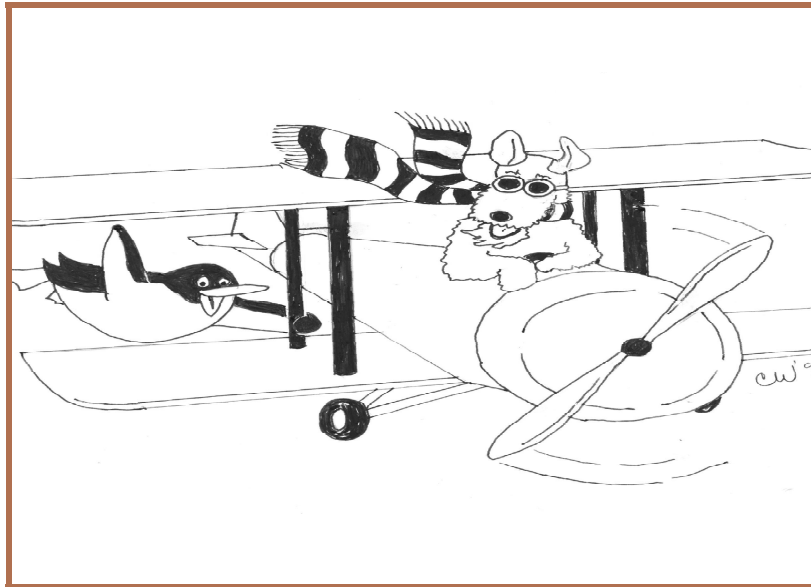


# T.Rex Firewall

## Installation and Administration Guide

Version 1.0



# FAS

2/10/2000

# Service and Support Information

## Product Version

This manual applies to T.Rex Version 1.0 software and the T.Rex Embedded Systems: T.Rex-ESm, T.Rex-ESs and T.Rex-ESi. To obtain technical support, customer service information or product sales use the contact information provided below.

## Telephone

Corporate:	281-759-3274
Technical Support:	281-759-3274 or 800-240-5754
Fax:	281-759-8558

## Electronic Mail

Technical Support:	<a href="mailto:trex@opensourcefirewall.com">trex@opensourcefirewall.com</a>
Sales Information:	<a href="mailto:sales@opensourcefirewall.com">sales@opensourcefirewall.com</a>
Product Feedback:	<a href="mailto:trex@opensourcefirewall.com">trex@opensourcefirewall.com</a>

## Internet Access

World Wide Web:	<a href="http://www.opensourcefirewall.com">www.opensourcefirewall.com</a>
Anonymous FTP:	<a href="ftp://gw.opensourcefirewall.com">gw.opensourcefirewall.com</a>

## Mailing Address

1830 S. Kirkwood Suite 205  
Houston, TX 77077

## Support Hours:

Regular Support:	8:00 am to 5:00 PM, Monday through Friday, Central Time (USA)
	Requires a support contract.

# Legal Notices

## Copyright

The following copyright notice protects this document under the copyright laws of the United States of America and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright (c) 1995-2000 by Freemont Avenue Software, Inc  
1830 S. Kirkwood Suite 205  
Houston, TX 77077

All Rights Reserved. Printed in U.S.A.

## Trademarks

T.Rex is a trademark of Freemont Avenue Software, Inc. worldwide and a registered trademark of Freemont Avenue Software, Inc. within the United States and other countries. Other products and company names are registered trademarks or trademarks of their respective holders. These are acknowledged in the section below.

## First Edition (February 2000)

**Freemont Avenue Software, Inc. (FAS) provides this manual "AS IS" without any warranty of any kind, express or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.** Some jurisdictions do not allow excluding or limiting implied warranties, and some jurisdictions have special statutory consumer protection provisions that may supersede this limitation. As a result, this limitation of liability may not apply to you if prohibited by the laws of your jurisdiction.

FAS does not warrant that the contents of this publication, including examples, will meet your requirements or that the publication or examples are free of error. This document could contain technical inaccuracies or typographical errors. Changes to the information contained in this document are periodically made and will be incorporated into new editions of the publication.

Requests for copies of this document and for technical information about FAS products should be made to FAS.

## Acknowledgments

The following trademarks and acknowledgments apply to this information:

Adobe Acrobat and Adobe are registered trademarks of Adobe Systems Incorporated.

AIX is a registered trademark of International Business Machines Corporation.

CRYPTOCard is a trademark of CRYPTOCard, Inc.

DEFENDER Security Server is a registered trademark of Digital Pathways, Inc.

HP and HP-UX are registered trademarks of Hewlett Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

LSLI is a registered trademark of Livermore Software Laboratories, International a division of Freemont Avenue Software.

MD5 was copyrighted by RSA Data Security, Inc. and released to the public domain.

NFS and NIS are registered trademarks of Sun Microsystems, Inc.

PORTUS is a registered trademark of Livermore Software Laboratories, International a division of Freemont Avenue Software, Inc.

RealAudio is a trademark of Progressive Networks.

RISC System/6000 is a registered trademark of International Business Machines Corporation.

SecureNet Key is a registered trademark of Digital Pathways, Inc.

SPARC, SPARCstation, and UltraSPARC are registered trademarks of SPARC International, Inc.

.

Sun, Sun Microsystems, Solaris, are registered trademarks of Sun Microsystems, Inc.

T.Rex is a registered trademark of Livermore Software Laboratories, International a division of Freemont Avenue Software, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

The HTTP proxy is a modified version of the Apache HTTP server developed by the Apache Organization.

This product is distributed under license from Freemont Avenue Software, Inc. and contains copyrighted materials of R. J. Livermore, Freemont Avenue Software, Adobe Systems Incorporated, Regents of the University of California, Progressive Networks, and Digital Pathways.

SOCKS Copyright (C) 1989 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Socks LICENSE, v 1.11 1999/05/24

Copyright (c) 1997, 1998, 1999 Inferno Nettverk A/S, Norway. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. The above copyright notice, this list of conditions and the following disclaimer must appear in all copies of the software, derivative works or modified versions, and any portions thereof, as well as in all supporting documentation.
2. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by Inferno Nettverk A/S, Norway.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE

OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Preface

This Installation and Administration Guide is directed to you, the system administrator, the person responsible for loading and customizing the T.Rex Firewall System. This document contains helpful information that is either too technical or too specialized for the normal user of the product. This document is designed to help you install, customize and maintain the product. This document describes the details of the command line interface. There is also a Graphical User Interface that contains online documentation.

The Installation and Administration Guide is organized by topic. The Customer Support section is placed near the beginning of the document so that you can find support information and numbers easily.

### Conventions Used in This Document

The following conventions are used in this document:

**Bold** is used for command names and filenames when they appear in the body of a paragraph. Bold is also used in examples to show commands that should be typed exactly as shown. For example, **ls -l** means the user should type "ls -l" exactly as it appears in the text.

*Italic* is used to show variables for which a context specific substitution should be made. For example, the IP address **192.168.240.2** should be replaced by the actual IP address of the customers system.

### Comments

If you have any comments or suggestions for improvement, please contact FAS Customer Support at:

Telephone: (281)-759-3274  
FAX: (281)-759-8558  
e-mail: trex@opensourcefirewall.com

or send mail to:

Freemont Avenue Software, Inc.  
1830 S. Kirkwood, Suite 205  
Houston, Texas 77077

# Table of Contents

Functions in T.Rex Release 1.0 .....	1-xv
Chapter 1. Introduction .....	1-1
1.1 Product Version .....	1-1
1.2 Executive Overview .....	1-1
1.3 Functional Summary Overview .....	1-2
1.4 Product Overview .....	1-2
1.4.1 Main Technologies .....	1-2
1.4.2 Secured Platform .....	1-3
1.5 Vendor Information .....	1-3
1.5.1 Contact Information .....	1-3
1.6 Security Architecture .....	1-4
1.6.1 Rationale .....	1-4
1.6.2 Security Architecture .....	1-4
1.6.3 Product Default operations .....	1-4
1.6.4 Protection of the firewall system .....	1-5
1.6.5 Protection of attached networks and hosts .....	1-6
1.6.6 Protection of individual hosts .....	1-6
1.7 Product Features and Mechanisms .....	1-7
1.7.1 Services Provided .....	1-7
1.7.2 Electronic mail (e-mail) .....	1-8
1.7.3 Application Proxy Server .....	1-9
1.7.4 HTTP Proxy .....	1-11
1.7.5 Web Content Filtering .....	1-11
1.7.6 Java, JavaScript and ActiveX Blocking .....	1-11
1.7.8 Telnet Proxy Overview .....	1-12
1.7.9 FTP Proxy Overview .....	1-13
1.7.10 RPC and UDP Proxy .....	1-14
1.7.11 Network News (NNTP) .....	1-14
1.7.12 Dual DNS Servers .....	1-14
1.7.13 SOCKS Circuit Gateway .....	1-15
1.7.14 Real Audio .....	1-15
1.8 Product Audit/Event Reporting and Summaries .....	1-15
1.8.1 Manner in which logs are stored and safeguarded .....	1-18
1.9 Product Testing Methodology .....	1-18
1.10 Product Performance Attributes .....	1-19
1.11 Product Operational Assumptions .....	1-20
1.12 Product Operational/management requirements .....	1-21
Chapter 2. Customer Support .....	2-1
Chapter 3. T.Rex Installation .....	3-1
3.1 Installation Overview .....	3-1
Before You Begin .....	3-1
3.2 OS Installation .....	3-2
3.3 AIX Installation .....	3-4
3.3.1 Install T.Rex Firewall Software on AIX .....	3-4
3.3.2 Configure AIX system .....	3-4
3.4 HP-UX Installation .....	3-6
3.4.1 Install T.Rex Firewall Software on HP-UX .....	3-6
3.4.2 Configure HP-UX system .....	3-6
3.5 Linux Installation .....	3-8



3.6 Solaris Installation .....	3-9
3.6.1 Install T.Rex Firewall Software on Solaris .....	3-9
3.6.2 Configure Solaris system .....	3-10
3.7 Common T.Rex configuration .....	3-12
IP Addresses Reserved for private networks .....	3-13
3.8 Setup Dual Domain Name Servers .....	3-17
3.8.1 Dual DNS Overview .....	3-17
3.6.2 Set up Internal DNS .....	3-17
3.6.3 Set up External DNS .....	3-17
3.6.4 Setup a Caching-only Name Server .....	3-18
3.7 Setup Mail Server .....	3-19
3. 8 Adding Administrative Users .....	3-19
3.8.1 Defining admin users with gwuser .....	3-19
3.8.2 Defining admin users with SMIT .....	3-19
3.8.3 Gateway Shell .....	3-21
3.9 Changes made during installation .....	3-21
3.9.1 Adding Proxy User ID .....	3-21
3.9.2 Sequestered E-mail Directory .....	3-21
3.9.3 Setup cron to periodically run sendmail .....	3-22
3.9.4 System Logs .....	3-22
3.10 OS Maintenance Requirements .....	3-25
3.10.1 AIX Maintenance .....	3-26
3.10.2 Solaris Maintenance .....	3-28
3.10.3 HP-UX Maintenance .....	3-29
Chapter 4. T.Rex User Administration .....	4-1
4.1 Managing User Access to T.Rex .....	4-1
4.2 Groups and Users .....	4-1
4.3 Online Help for group Administration .....	4-2
4.4 Adding a Group .....	4-3
4.5 Deleting a Group .....	4-4
4.6 Listing a Group .....	4-4
4.7 Modifying a Group .....	4-5
4.8 Adding a User .....	4-6
4.9 Deleting a User .....	4-9
4.10 Listing a User .....	4-9
4.11 Modifying a User .....	4-9
4.12 Examples .....	4-10
4.12.1 Adding a Group .....	4-10
4.12.3 Adding a User .....	4-10
4.12.4 Listing a User .....	4-11
4.12.5 Listing all users in a group .....	4-11
4.12.6 Deleting a user .....	4-11
4.12.7 Adding an Administrative User .....	4-12
4.13 Gwuser Configuration File .....	4-12
4.14 Configuring Restricted Shells .....	4-13
4.13 Generate Encrypted shared keys .....	4-15
4.13.2 Generate Encrypted user Key for OOBASRV .....	4-16
Chapter 5. Using Security Tokens .....	5-1
5.1 Programming the SecureNet Key (SNK) .....	5-1
5.2 Clearing SNK memory .....	5-1
5.3 Programming the CRYPTOCARD .....	5-2
5.4 Clearing CRYPTOCARD memory .....	5-2

5.5 Octal Conversion Chart .....	5-4
Chapter 6. Aproxy Administration .....	6-1
6.2 Aproxy Configuration File .....	6-3
6.3 Sample aproxy.conf file .....	6-7
6.4 To Proxy SQL .....	6-8
6.4.1 To Proxy SQL*NET .....	6-8
6.5 To Proxy pcAnywhere .....	6-8
6.6.1 API Code .....	6-9
6.6.2 Example Exit Programs .....	6-9
6.6.3 Activation of a new Exit .....	6-10
Chapter 7. FTP Proxy Administration .....	7-1
7.1 FTP Proxy Overview .....	7-1
7.2 FTP Proxy Configuration File .....	7-2
7.3 Sample ftproxy.conf file .....	7-5
Chapter 8. HTTPD Proxy Administration .....	8-1
8.1 HTTPD Proxy Overview .....	8-1
8.1.1 Multiple Functions .....	8-1
8.1.2 On the firewall .....	8-1
8.1.3 Behind the firewall .....	8-2
8.1.4 SOCKD versus HTTPD .....	8-2
8.1.5 Malicious Java and ActiveX .....	8-2
8.1.6 Java, JavaScript, ActiveX and Cookie blocking .....	8-3
8.1.7 Web Content Filtering .....	8-4
8.2 HTTPD Installation .....	8-5
8.3 Configuring HTTPD Proxy .....	8-6
8.4 Sample HTTPD Configuration for Firewall .....	8-18
8.5 Sample HTTPD Configuration for proxy behind firewall .....	8-21
8.6 Installing the WebBlocker GUI .....	8-21
8.6.1 Solaris .....	8-22
8.6.2 AIX 4 .....	8-22
8.7 Selecting URL Categories to Block .....	8-23
8.7.1 Adding URL lists for blocking .....	8-23
8.8 Controlling FTP access via HTTP .....	8-24
8.8.1 Requiring user authentication for FTP .....	8-24
8.8.2 Blocking ftp sites except for a select few .....	8-25
Chapter 9. RealAudio Proxy Administration .....	9-1
9.1 RealAudio Proxy Overview .....	9-1
9.2 Update the /etc/services file .....	9-1
9.3 Update the /etc/inetd.conf File .....	9-1
9.4 Raproxy Configuration File .....	9-1
9.5 RealAudio Player Configuration .....	9-3
9.5.1 How to configure RealPlayer Version 6 .....	9-4
9.6 Sample raproxy.conf file .....	9-5
Chapter 10. RPCproxy .....	10-1
10.1 Overview .....	10-1
10.2 RPCproxy Configuration File .....	10-3
10.3 Sample RPCproxy configuration file .....	10-8
Chapter 11. Secure Mail Wrapper .....	11-1
11.1 Secure Mail Wrapper Overview .....	11-1
11.2 Secure Mail Wrapper Configuration File .....	11-3
11.2 Sample Secure Mail Wrapper Configuration File .....	11-6
11.3 Sample e-mail Blocking .....	11-7

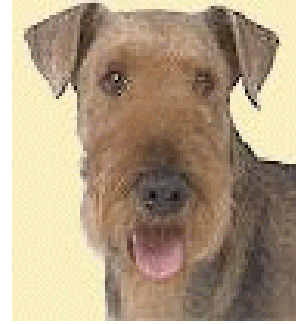
11.4	Secure Mail Wrapper Aliases File .....	11-8
	Sample Aliases File .....	11-9
11.5	Aliases Data Base Administration .....	11-13
	11.5.1 Creating the Aliases Data Bases .....	11-14
	11.5.2 Adding new entries to the Aliases Data Bases .....	11-17
	11.5.3 Deleting an entry from the Aliases Data Base .....	11-18
11.6	Listing Aliases and Reverse Aliases Data Bases .....	11-19
Chapter 12.	Mail Setup .....	12-1
	12.1 Overview .....	12-1
	12.2 Sendmail Configuration File Changes .....	12-1
	12.2.1 Additional Changes for Sendmail 8.6+ .....	12-2
	12.3 Changes to sendmail configuration files of protected hosts .....	12-2
Chapter 13.	Telnet Proxy Administration .....	13-1
	13.1 Telnet Proxy Overview .....	13-1
	13.2 X-Window Forwarding .....	13-1
	13.3 Telnet Proxy Configuration File .....	13-2
Chapter 14.	Webgate Administration .....	14-1
	14.1 Webgate Overview .....	14-1
	14.2 Webgate Configuration File .....	14-2
	14.3 Sample webgate.conf file .....	14-6
	14.4 Workload Balancing Example .....	14-6
	14.5 Refreshing Webgate .....	14-7
Chapter 15.	SOCKS Configuration .....	15-1
	15.1 SOCKS Overview .....	15-1
	15.2 SOCKS Server Installation .....	15-1
	15.3 Sockd configuration file .....	15-1
	15.3.1 SOCKS Server Settings .....	15-1
	15.3.2 SOCKS Client Rules .....	15-2
	15.4 Sample sockd configuration file .....	15-5
	15.5 SOCKS Client installation .....	15-8
	15.6 Sample socks configuration file .....	15-9
	15.7 SOCKS Configuration file rules .....	15-9
	15.8 Domain Name Service for SOCKS .....	15-12
Chapter 16.	DNS Configuration .....	16-1
	16.1 DNS Overview .....	16-1
	16.2 Internet Firewall DNS .....	16-1
	16.2.1 Caching-Only DNS Firewall .....	16-1
	16.2.2 Full Function DNS .....	16-2
	16.3 Internal Firewall .....	16-2
	16.4 Hosts File .....	16-2
Chapter 17.	Performance Monitor .....	17-1
	17.1 Performance Monitor Overview .....	17-1
	17.2 Monitor Options .....	17-1
	17.2.1 Interactive key commands .....	17-1
	17.2.2 Monitor Options .....	17-2
	17.3 Sample Monitor Display .....	17-3
Chapter 18.	T.Rex Integrity Monitor .....	18-1
	18.1 T.Rex Monitors Overview .....	18-1
	18.2 T.Rex Monitors Invocation .....	18-1
	18.3 T.Rexmon Configuration File .....	18-2
	18.4 SYN Flood Discussion .....	18-4
	18.5 General Discussion .....	18-5

18.6	When an attack is detected	18-5
18.7	Example /etc/firewall/T.Rexmon.conf File	18-7
Chapter 19	Firewall Logging Facilities	19-1
19.1	Activating System Logging	19-1
19.1.1	Sample /etc/syslog.conf file	19-1
19.1.2	Create the system log file	19-2
19.1.3	Activating syslogd	19-2
19.1.4	Activating syslog changes	19-2
19.1.5	syslog maintenance	19-2
19.2	SYSLOG Format	19-3
19.3	Examining the system log	19-3
19.3.1	Displaying the syslog in real time	19-3
19.3.2	Selecting messages by type and date	19-4
19.3.2	Displaying Security Alerts in real time	19-4
19.3.4	Reporting SOCKD activity	19-6
19.3.5	Sample Sockdsum Output	19-7
19.3.6	Searching for details on SOCKD activity	19-9
19.3.7	Reporting rftp activity	19-10
19.4	Reporting FTP activity	19-11
19.5	Genproxy Activity Reports	19-13
19.6	Reporting Secure Mail Wrapper activity	19-14
19.6.1	Sample smrpt output	19-15
19.6.2	Reporting Exceptions	19-17
19.7	Reporting tnproxy activity	19-21
19.8	Tracking HTTP Usage	19-24
19.8.1	Displaying http access in real time	19-24
19.8.2	HTTPSUM Report	19-24
19.8.3	Sample httpsum output	19-25
19.9	Tracking Administrator logins	19-26
19.10	Tracking Failed Login Attempts	19-27
19.11	Reporting use of the su command	19-28
19.11.1	Reporting all su attempts to root	19-28
19.10.2	Reporting all failed su attempts	19-29
19.12	Logging DNS activity	19-29
19.12.1	Activating named debugging during system startup	19-30
19.12.2	Controlling named debugging with Signals	19-31
19.12.3	Sample named Startup Debugging Information	19-31
19.12.4	Sample named Lookup Debugging Information	19-32
Chapter 20	Accounting and Security	20-1
20.1	Accounting Overview (AIX)	20-1
	Accounting Commands	20-1
	Sample Daily Summary	20-2
Chapter 21	T.Rex Admin Server	21-1
21.1	Tadminsrv Overview	21-1
21.2	Padminsrv Configuration File	21-1
21.3	Sample padmin.conf File	21-2
21.4	Padminsrv Backup Lists	21-2
21.4.1	T.Rex Backup List	21-2
21.4.2	System Backup List	21-3
21.4.3	Apache Backup List	21-4
21.4.4	Custom Backup List	21-4
Chapter 22	Remote logging with Plog	22-1

22.1 Plog Overview	22-1
22.2 Plog Configuration	22-1
22.3 Example plog.conf file	22-2
22.4 Plogd Overview	22-2
22.5 Plogd Configuration	22-3
22.6 Example plogd.conf file	22-4
Chapter 23. Protected Telnet	23-1
23.1 Ptelnet Overview	23-1
23.2 T.Rex User Administration	23-1
23.2.1 Create gwuser record	23-1
23.2.2 Creating a Remote Firewall Administrator	23-1
23.2.4 Creating a user with Encrypted telnet support	23-2
23.3 Generate shared private key for ptelne	23-2
23.4 Enable administrator logins	23-3
23.4.1 AIX Administrators	23-3
23.4.2 HP-UX Administrators	23-3
23.4.3 Solaris Administrators	23-3
23.5 ptelnet Installation	23-3
23.6 Ptelnet for Windows 95 or Windows NT	23-4
23.7 Ptelnet for AIX	23-4
23.8 Ptelnet for HP-UX	23-4
23.9 Ptelnet for Solaris	23-5
23.10 Install Ptelnet on UNIX	23-6
23.11 Copy T.Rex.key file	23-6
23.12 Remote Firewall Login Using Ptelnet	23-6
Chapter 24. External Security Server Support	24-1
24.1 Overview	24-1
24.2 Configuring DSS Support	24-1
Chapter 25. Integrity Checking and Auditing	25-1
Trusted Computing Base (TCB) (AIX only)	25-1
Chapter 26. Hot Backup	26-1
26.1 Overview	26-1
26.1.1 Manual Takeover	26-1
26.1.2 Automated Takeover	26-1
26.2 fwpulse	26-1
26.2.1 Pre-Takeover Exit	26-2
26.2.2 Post-Takeover Exit	26-2
26.2.3 Reverse Takeover	26-2
26.3 Fwpulsed	26-3
26.4 Hot Backup Configuration File	26-3
26.5 Auto Takeover Decision Tree	26-5
Chapter 27. Testing T.Rex	27-1
27.1 Testing for IP Packet Forwarding	27-1
27.2 Testing the Domain Name Server	27-1
27.3 Testing Ports with portscan	27-1
27.4 Portscan Services file	27-2
27.5 Sample Services file	27-2
27.6 Sample portscan output	27-4
Chapter 28. Sources for Security Information	28-1
28.1 Computer Emergency Response Team (CERT)	28-1
28.2 Computer Incident Advisory Capability (CIAC)	28-1
28.3 Mailing Lists	28-2

28.4 Usenet News Groups .....	28-3
Chapter 29. T.Rex Messages .....	29-1
29.1 APROXY Messages .....	29-1
29.3 Fwpulse Messages .....	29-17
29.4 Genproxy Messages .....	29-17
29.5 RAPROXY Messages .....	29-22
29.6 RPCproxy Messages .....	29-26
29.7 Sendmail Wrapper Messages .....	29-28
29.8 TNPROXY Messages .....	29-38
29.9 Webgate Proxy Messages .....	29-40
29.10 Gwsh2 Messages .....	29-45
29.11 Adam utility Messages .....	29-46
Chapter 30. Miscellaneous .....	30-1
Chapter 31. Adding Software to T.Rex .....	31-1
Chapter 32. Recommended Reading .....	32-1
Appendix A. Online Documentation .....	32-1
Appendix B. Sample Configuration Files .....	32-1
Files .....	32-1
Step 13. Turn off unwanted Daemons and activate T.Rex daemons. ....	32-5
Step 13.1 Overview .....	32-6
Step 13.2 AIX Systems .....	32-7
/etc/rc.tcpip .....	32-7
/etc/inetd.conf .....	32-10
/etc/rc.net (for AIX) .....	32-10
/etc/rc.nfs (for AIX 4.1) .....	32-10
/var/spool/cron/crontabs .....	32-13
/etc/syslog.conf .....	32-14
/etc/security/login.cfg .....	32-15
/etc/inetd.conf .....	32-15
/usr/lib/security/mkuser.default file .....	32-15
Index .....	32-1

# Functions in T.Rex Release 1.0



T.Rex 1.0 is a full function application proxy firewall. T.Rex supports the following software, hardware and applications.

## Operating System Support

<b>AIX 4.x</b>	T.Rex is supported on AIX 4.2 and AIX 4.3 (64-bit AIX).
<b>HP-UX 10</b>	T.Rex can be compiled and run on HP-UX, but is currently not supported by FAS.
<b>Linux</b>	T.Rex is supported on Red Hat Linux 6.1 (Intel and SPARC) and Caldera 2.3.
<b>Solaris</b>	T.Rex is supported on Solaris 2.6, 7 on SPARC and INTEL architectures. Solaris 2.6 Desktop Intel Platform Edition is required for Intel X86 compatible systems.

## LAN Interfaces

T.Rex works with any LAN Card supported by the Operating System. This includes:

Ethernet, Fast Ethernet, Gigabit Ethernet, FDDI, Token Ring, ATM (155 Mbs, 622 Mbs)

## VPN

T.Rex supports IPsec standards for VPN, including the Internet Security Association and Key Management Protocol or ISAKMP. The IPSEC card uses Digital Signature Standard (DSS) and Secure Hash Algorithm (SHA) in conjunction with X.509 v3 certificates to verify the identity of the sender and provide proof of authorship.

Processor intensive tasks such as encryption are offloaded on to specialized hardware providing unequaled system performance.

## VPN Encryption Standards

DES, DES-56, DES-128, Triple DES

## Hardware Based Encryption

T.Rex supports hardware based VPN using the RedCreek Communications IPsec Card. The

support is currently available only on Linux.

## Other Uses of Encryption

T.Rex provides secure remote administration using DES encryption along with strong user authentication.

## Authentication Supported

RADIUS, SecureID, CryptoCard, ActiveCard

## Advanced Application Proxy

The advanced application proxy (**aproxy**) is a third generation proxy that can support nearly all TCP/IP connection oriented client server applications. It also provides an Application Program Interface (API) and Workload Balancing.

### API

Aproxy provides an Application Program Interface (API) to permit customization for any number of applications. The API provides each application with three exits:

**Authentication exit::** permits customized access controls per application,  
**Client exit::** provides access to the buffer after receiving data from the client,  
**Server exit:** provides access to the buffer after reading data from the server.

### Workload Balancing

Aproxy also supports workload balancing between multiple servers on an application by application basis.

### High Performance

Aproxy is a standalone proxy that does not run under inetd. This makes it simpler to configure. It also exhibits high levels of performance since it pre-forks processes reducing systems overhead up to 90%. The number of processes are automatically adjusted to match the workload.

### Printer Support

The Ignore RST parameter has been added to the permit statement to improve printer support.

DNS lookup can be suppressed for peer hosts that do not have a DNS server.

## FTP Proxy

Can control commands by User and Group as well as time of day and day of week. All file transfers are logged by filename. Ftpoxy provides transparent support for inbound **anonymous ftp** users.

The ftpoxy blocks receiving of files when the file names begin with the pipe symbol "|". Some ftp clients will try to execute the file which could have disastrous results.



Ftp access controls can be used to restrict access to ftp sites, and or require user authentication.

## HTTP Proxy

The HTTPD proxy is a derivative of the Apache httpd server and has modified in the following manner:

- It has been compiled with functions that limit it to acting as proxy server only.
- A module has been included that is used to permit URL content filtering by category with customizable controls for re-direction, time-based access and more.
- Caching performance has been enhanced to exploit parallel I/O when the cache is distributed across multiple disks for added performance.

It operates in stealth mode which means it provides transparent web access to the protected network but can not be seen by an external host. It fully exploits SMP systems to maximize systems performance.

The HTTP proxy implements the latest protocols including HTTP.1.1 (RFC2068). This includes support for persistent connections, and chunked encoding.

### Java, Java Script, ActiveX and Cookie Blocking

HTTPD allows selective blocking of Java applets, JavaScripts, ActiveX and cookies. The HTTPD Proxy supports permit and deny commands for all these functions. This facility permits the administrator to allow some systems to send java applets but can deny access from all other systems.

### SSL Tunneling:

Httpd now supports SSL tunneling for applications other than SSL (port 443) and snews (port 563).

### Performance Features:

Httpd allows specification of larger receive data buffers to improve systems throughput. Httpd directives can be used to increase the TCP send buffer size. This is useful to increase performance on high speed high latency networks, such as high speed transcontinental lines.

When presented with a load spike httpd quickly adapts by spawning children at a faster rate.

## Webgate Proxy

Webgate is a high speed reverse http proxy designed to secure one or more web servers behind the firewall.

Multiple Web servers can reside behind the firewall each having its own name and IP address. Secure transmission of sensitive data is assured by the use of SSL.

The Web servers can be isolated on a network separated from the organizations secured

networks, thus providing higher levels of security.

Webgate now runs as a stand-alone daemon with preallocated processes. This eliminates approximately 90% of the system overhead. Webgate automatically restarts any failed process. The number of processes can dynamically be increased without disturbing existing work.

Webgate dynamically adjusts the number of pre-forked processes depending on the current workload.

Common Log Format extensions have been added to append Agent and Referrer data to the CLF records.

The Webgate recovery time has been reduced for hot backup situations.

Webgate can be directed to write Multi-Homed http access log format records to provide support for WebTrends when supporting multiple domains.

### **Workload Balancing**

Webgate can be used to balance the workload between multiple web servers, allowing an array of web servers to appear as one. If one of the web servers fails or is varied offline for maintenance then it automatically skips over the offline server until it detects its presence again.

## **SMTP**

The Secure Mail Wrapper program (smwrap) receives mail from remote hosts. This program is designed reduce exposure to SMTP based attacks and scrub internal network information from out bound mail.

### **Block SMTP based Attacks**

Smwrap protects against multiple attacks directed at mail servers, and mail clients. This includes checks for unauthorized use, requests to obtain access to private information, and multiple Denial of Service (DoS) attacks.

Smwrap guards against: Unauthorized sender/receiver, Bogus Helo command, use of VRFY and EXPN commands, anonymous mail relaying, commands imbedded in Header fields, password file access, Root user access, sendmail debug exploits, address spoofing.

#### **DoS Attacks blocked:**

Helo buffer overflow, SMTP command buffer overflow, SMTP header overflow, SMTP Header Parsing Attack, Maximum number of recipients exceeded, Maximum message size exceeded, harmful header address characters, MIME header buffer overflow, MIME field overflow, and more.

### **Mail Blocking**

Smwrap prevents annoying e-mail messages, commonly called "SPAM" from entering protected networks. The feature also blocks harassing messages making "Cyber-stalking"

more difficult. The Administrators can enter a list of senders, addresses, sites, or domains they want to target for blocking. Like Call Blocking on your telephone smwrap allows you to choose who you want to get e-mail from. Blocked e-mail can be deleted, sequestered or redirected to a specified recipient. If the mail is sequestered or redirected it can be kept as evidence along with the log information.

The **aliasreq** command permits control of who is allowed to send mail outbound through the firewall. If a user is not registered in the alias data base then any attempt to send mail through the firewall will be rejected and a Security Alert will be issued.

The Version 1.0 Mail Wrapper translates internal e-mail addresses to external e-mail addresses. This translation includes all internal addresses that are part of a Carbon Copy (Cc:) or To: addresses. Translation support for addresses generated by Novell's GroupWise and MS Mail Exchanger is also provided. .

Smwrap supports translation of out-bound headers generated by **Microsoft Outlook**. Smwrap will not translate the e-mail addresses that Outlook encloses within double quotes on out bound mail.

The MS Internet Mail Exchange program can be configured to produce non-standard e-mail addresses in To: and Cc: fields. Smwrap can accommodate translations of several new forms of To: and Cc: addresses on out-bound e-mail.

Smwrap deletes partially completed store and forward files from the hermes directory when there is an unexpected EOF from the remote client or I/O error.

### RealAudio Proxy

The RealAudio Proxy (raproxy) allows users behind the firewall to safely access to RealAudio servers through the firewall. The raproxy allows the systems administrator to control RealAudio access through the use of permit and deny commands in a manner consistent with the other proxies. Raproxy supports RealAudio Version 3.0 for servers that are not HTTP based. This includes G2 level multi-media. The HTTP proxy supports RealAudio V3.0 for Web browsers.

### RPC and UDP Proxy

The RPC-UDP Proxy (rpcproxy) provides controlled access for client server applications that use RPC, TCP and UDP protocols. The RPC proxy supports applications such as NFS, and tftp.

### Telnet Proxy

The telnet proxy provides extensive controls over the use of the telnet protocol. It also has an interface to allow X-Window applications to be used through the firewall. TN3270 is supported.

### X11 support

The xforward proxy has been tuned to minimize the cpu time required to support the X-Window applications.

To allow use of automated telnet scripts that use xforwarding a "port" argument can be added to the xforward command. To use this feature type port = nn after the xforward command.

xforward -port nn

Where nn is an integer from 10 to 99.

## SOCKS

The socks daemon supports both Version 4 and Version 5 of the SOCKS protocols. The socks V5 protocol supports both TCP/IP and UDP.

## Workload Balancing

The proxies support workload balancing for HTTP and most TCP/IP client server applications.

## NAT

T.Rex being a n application proxy automatically supports NAT.

## URL Content Filtering

The HTTPD proxy has built-in URL content filtering. FAS sells an annual subscription service that automatically updates the blocking lists used by the content filter.

- The Blocking List contains 33 pre-defined categories which you can be individually activated.
- The categories is open-ended and the format of the list is defined allowing any administrator to add, delete or modify the list contents.
- Time based access can be specified for each category. Some categories may be blocked all the time while others are blocked only during normal business hours.
- The administrator can specify custom error messages that is displayed to the user when they attempt to access a blocked URL. Unique messages can be used for each category.
- A script is available to automatically download updates to the blocking lists.
- All HTTP access is logged and report programs are available to analyze Web access.
- Access can be customized by client address. Some workstations can be excluded from any blocking, others can denied any access.

## Blocking of ActiveX, cookies

This is integrated into the HTTP proxy server.

## Automated installation process

The installation process has been automated to reduce the time and effort required to install and configure the T.Rex firewall. New users simply run the install\_T.Rex command, updates can be applied with the update\_T.Rex command. .

## Online Documentation

The Installation and Administration Guide is available as a PDF document that can be browsed and searched using the Adobe Acrobat reader.

## Report programs

Report programs are included that can produce **57 reports** which summarize activity by application.

Aproxy	5 reports: summary, top host by: bytes sent, bytes received, connection requests, received connections
FTP	5 reports: summary, top user by: bytes sent, bytes received, connection time, cpu time.
HTTP	32 reports:
Mail	7 reports: summary, top user by: messages received, messages sent, bytes received, bytes sent, exception reports.
Socks	3 reports: top host by bytes sent, bytes received, received connects.
Telnet	5 reports: summary, top user by: bytes sent, bytes received, connection time, cpu time.

## Real Time monitoring

Real time displays of the syslog, Security Alerts and HTTP activity can be displayed on the firewall console or on a remote host.

## Systems monitoring

The firewall monitor manages specialized daemons the monitor firewall activity.

**IP Spoofing** The automated IP Spoofing Monitor (spooftmon) alerts the system and systems administrators to attempted IP spoofing attacks. The IP spoof monitor can support up to 100 alias IP addresses on AIX 4.2+.

**SYN Flood** The SYN flood monitor checks for SYN flood attacks. A kernel extension is available for AIX 4.2 that improves resistance to SYN Flood denial of service attacks.

**Process** The firewall monitor program scans the process table for process names specified in thefwmon.conf file. Procmon counts the number for each of the specified processes and issues a Security Alert if the number falls outside of the specified range.

### disk space monitor

The diskmon procedure monitors disk utilization for specified file systems. When a file systems utilization crosses a user defined threshold an appropriate message is delivered to a list of recipients. Four utilization thresholds are mapped to the following message levels: Notice, Warn, Crit, and Alert. As the messages increase in severity the message router will send the message to additional persons.

Diskmon notifies the systems administrators of pending disk shortages in time for them to implement preventative measures.

### Performance Monitor

**AIX** The T.Rex firewall ships with a real-time performance monitor for AIX 4. The AIX performance monitor provides a real time display of cpu utilization, memory usage, LAN activity by interface, disk activity as well as other information. The monitor can display top processes sorted by cpu, or memory usage. SMP support is included. The monitor can also be used archive resource utilization to disk using weighted averages over a specified period of time. The real-time displays can be customized to use different colors schemes.

**Solaris** A comprehensive performance monitor is also available for Solaris.

**Linux** Under investigation.

### DEFENDER Security Server

T.Rex can be configured to act as an agent for the DEFENDER Security Server (DSS). Configuring T.Rex as a DSS agent bypasses the built-in security server of T.Rex and makes use of a similar challenge response system provided on the DSS.

### Enhanced User and Group Administration

T.Rex Version 1.0 simplifies user administration by assigning a user to a predefined group from which the user will inherit its permissions.

### Year 2000 Compliant

T.Rex version 1.0 is year 2000 compliant.

### GUI Administration tool

**Admin. Client** Hoplite provides the Graphical User Interface for **remote firewall administration**. Hoplite allows the systems administrator to manage one or more firewalls from a single location. Hoplite also includes a backup and recovery function for T.Rex configuration data. Hoplite provides strong user authentication and encrypted communications to insure the security and integrity of the firewall. The Hoplite client can run on MS WIN/95 , MS/NT 4.0, AIX 4.1.4+, Solaris 2.5.1+ (SPARC and Intel).

**Admin. Server** T.Rex 1.0 introduces the T.Rex Administrative Server (tadminsrv) to support GUI based remote administration using the Hoplite client.

### Encrypted Telnet

The ptelnet client provides encrypted telnet sessions. It also provides a secure command line interface for performing remote administration. The ptelnet client runs on AIX, HP-UX, Linux, MS Windows, MS NT, and Solaris (SPARC and Intel).

## Error Messages

There are more than 300 unique error messages to assist with problem determination.

## Automatic activation of Secure Computing Base .

The automated installation process activates the secure computing base. On HP-UX systems the trusted Systems/Secure Password Facility is activated.

## High Availability Option

The addition of new configuration commands allows for improved automated recovery when fwpulse detects its partner system has failed. Auto recovery has been added to simplify the process when the failed system comes back online. The takeover and recovery processes can be customized to the local environment through the use of pre and post takeover scripts.

The maximum number of network addresses supported by fwpulse for the takeover process has been increased from 16 to the maximum number of IP addresses supported by the OS ( a very large number).

## 64-bit accounting routines

The proxies and their associated reporting programs use 64-bit arithmetic to allow numbers as large as 128 terabytes. This allows transmission of multiple large files ( > 2 GB) in a single session with accurate accounting.

## Persistent Out Of band Authentication (OOBA)

## Non-disruptive procedures to refresh HTTP proxies and syslog daemon

Non-disruptive procedures to refresh the httpd and webgate proxies and the syslog daemon are provided. This allows configuration changes to be made to these proxies and daemon without disrupting operations.

## Remote Logging Facility

The plog daemon supports remote logging. This allows HTTP access logs to be send to another machine for log analysis by an third party tool.

Also provided is an plogd program that receives the logs from T.Rex. This program also automatically rotates logs and is supported on NT.

## Third Party Applications Supported

This is a partial list of applications which work with T.Rex.

Apache, Chameleon, gopher, Hummingbird Exceed, Informix, LDAP, Lotus Notes, Oracle, POP3, Microsoft IE, Microsoft Proxy Server, Microsoft Mail Exchange, Microsoft Outlook, Netmanage, Netscape Communicator, Netscape Secure Commerce Server, NNTP, NTP, OASIS, Pcan anywhere, RealAudio, RPCs Assuren et/Digital Pathways DSS Server, SNMP, Sybase, SOCKS clients, tftp, UDP WAIS, ...



# Chapter 1. Introduction



“An Airedale can do anything any other dog can do and then whip the other dog if it has to.”

Theodore Roosevelt

## 1.1 Product Version

Vendor Name: Freemont Avenue Software, Inc.

Product Version: T.Rex 1.0

Date of Publication: January 4, 2000

## 1.2 Executive Overview

The T.Rex Firewall is a highly integrated enterprise security suite that combines functions that normally require the installation of multiple products.

- Access control
- Authentication
- Extensible Application controls via application specific APIs.
- Hardware assisted Virtual Private Network (VPN)
- Network Address Translation (NAT)
- Content Filtering (URL, Java, ActiveX, JavaScripts, SPAM)
- Fault tolerant High Availability Option (99.999% availability)
- Workload Balancing
- Non-disruptive hardware and software modifications
- Extensive auditing and reporting tools that can produce more than 52 unique reports
- Real-time performance monitor
- Network scanning and intrusion detection tools
- Totally automated operations to minimize administration overhead.

T.Rex provides unequalled scalability to meet the requirements of small, large and ultra-large organizations. It's industry leading performance supports more than a gigabit/second throughput. Hardware assisted IPsec VPN provides encrypted communications without sacrificing system throughput. It provides support for hundreds of services and applications.

T.Rex provides organizations with the ability to define a single security policy that can be distributed across multiple firewalls from a single administration workstation.

T.Rex is an advanced hybrid firewall designed to repel the most sophisticated attacks from skilled and determined crackers. Application specific proxies block application based attacks that pass unnoticed through the best of the stateful packet filters. The proxy API's also allow local customization to fine tune security controls for third party applications. The T.Rex fault-tolerant architecture provides multiple levels of error detection, reporting and recovery. The "fail-safe" architecture blocks the flow of traffic when an error occurs thus preventing accidental violations of the security policy. Unlike packet filter firewalls that can fail-open in the event of a hardware or software error T.Rex will fail-shut blocking unauthorized traffic.

T.Rex logs and controls all traffic between secured and unsecured networks. T.Rex can be configured to match the security policies of an organization instead of imposing its own policy upon an organization. T.Rex is easy to install, configure, maintain and use. T.Rex works with shrink wrapped applications to provide easy and transparent access from secured to unsecured (Internet) networks.

T.Rex is available on AIX, HP-UX, Linux, and Solaris (SPARC and Intel) systems. In addition it is available on a variety of embedded systems using PowerPC, UltraSPARC and Intel architectures. T.Rex runs on single and multi-processor systems as well as clusters of SMP systems.

## **1.3 Functional Summary Overview**

The firewall functional summary follows the standard format used by most firewall vendors to describe distinguishing features and advantages of their products. This format has been derived through an open process including firewall vendors, agencies of the computer security and firewall customers.

## **1.4 Product Overview**

The T.Rex Firewall enterprise security suite resides on a hardened multi-homed bastion gateway. It provides an extensive array of technologies to allow T.Rex to be configured to enforce the local security policies.

### **1.4.1 Main Technologies**

The main technologies found in T.Rex include:

- Application specific proxies for
  - E-mail (SMTP, POP3),
  - File Transfer (FTP),
  - World Wide Web (HTTP, SHHTTP, SSL)
  - Terminal Services (Telnet, TN3270),
  - X Window System (X11),
  - Real Audio & Real Video
- Advanced Application proxy with extensible application controls via an API.
- A generalized RPC and UDP proxy
- Hardware assisted Virtual Private Network (VPN)
- Network Address Translation (NAT)
- Socks V4 & V5 Circuit Gateway
- E-mail controls
- Stateful Packet Filtering
- Integrated Content Filtering (URL, Java, ActiveX, JavaScripts, SPAM)
- Integrated fault tolerant High Availability Option (99.999% availability)
- Integrated Workload Balancing

High Speed Caching  
Split DNS  
Intrusion Monitoring and Detection  
Graphical User Interface  
Network Scanners  
Integrated Authentication Servers  
Built-in monitors for  
    Detecting attacks,  
    checking system and network integrity  
    performance and capacity,  
Automated operations including automatic log management and report generation

T.Rex provides unequaled scalability to meet the requirements of small, large and ultra-large organizations. It's industry leading performance supports more than a gigabit/second throughput. Hardware assisted IPsec VPN provides encrypted communications without sacrificing systems throughput. It provides support for hundreds of services and applications.

T.Rex provides organizations with the ability to define a single security policy that can be distributed across multiple firewalls from a single administration workstation.

The Advanced Application Proxy (Aproxy) supports hundreds of TCP client server applications. Its API allows local customization on an individual application basis to provide fine grained control of information flow between client and server. Aproxy supports countless applications including:

- USENET New (NNTP),
- Lotus Notes,
- Sybase SQL,
- DB2,
- Oracle SQL\*Net,
- Informix,
- Microsoft Exchange
- PCAnywhere,
- etc.

## **1.4.2 Secured Platform**

The automated Installation procedure for T.Rex automatically hardens the Operating System. Unessential services and services that pose security risks are removed. Non-secure TCP/IP services are disabled. All communications are controlled by programs shipped with the firewall.

## **1.5 Vendor Information**

Freemont Avenue Software is a privately held Texas corporation that has been designing and writing Internet security software since 1994.

### **1.5.1 Contact Information**

Contact Name:	Freemont Avenue Software, Inc.
Contact Business Hours:	09:00 - 17:00 CST Monday through Friday
Contact Telephone number:	281-795-3274
Contact FAX Number:	281-759-8558

Contact Email Address: [trex@opensourcefirewall.com](mailto:trex@opensourcefirewall.com)  
Contact Web URL: [www.opensourcefirewall.com](http://www.opensourcefirewall.com)  
Contact Postal Address: 1830 S. Kirkwood Suite 205  
Houston, TX 77077

## 1.6 Security Architecture

### 1.6.1 Rationale

When describing a networked computer security system, there are several aspects of its design that must be taken into consideration. A security system such as a firewall must be able to protect not only the systems connected to it, it must be able to protect itself. Generally, the mechanisms whereby this is accomplished are different. The firewall system's security is dependent on whatever security mechanisms the firewall has built into itself. The systems connected to the firewall's security are dependent on whatever security mechanisms the firewall provides to them. In some cases these mechanisms may be based on a common design feature. In others they may be a result of a combination of features. In this section we explain how the firewall protects itself and the systems connected to it. In cases where additional protections are provided, or additional protective relationships are provided, we will explain the design principles and operation of these protective relationships.

### 1.6.2 Security Architecture

T.Rex secures hosts on protected networks by providing strict controls on available functions, who can use them and how they can be used. T.Rex protects itself from attack by the design of its functions and special controls and monitoring built into the product.

T.Rex is designed to repel all known methods of attack. This is accomplished as follows:

- Programs with known security problems are eliminated.
- Standard daemons are replaced with secure proxies developed by FAS.
- IP packet forwarding is disabled. Thus, protected networks are not IP addressable from unprotected networks.
- The only way to get information through the firewall is by using one of the secure proxies provided with T.Rex.

Details can be found in the following sections.

### 1.6.3 Product Default operations

T.Rex takes the most secure approach to system security. That is all functions are denied except those which are explicitly allowed. All services are explicitly enabled through the creation of permit rules. If a systems administrator forgets to do something the service is denied. IP forwarding is turned off so that packets do not flow through the firewall without explicit rules coded by the administrator.

The gateway routing protocols have been deactivated and replaced with static routing. Protocols that can dynamically update the routing table, such as RIP and OSPF are blocked to prevent routing attacks. Routes are explicitly defined by the systems administrator using GUI tools.

All traffic through the firewall is logged by the proxies. This includes both successful and unsuccessful attempts to use the firewall. The proxies log date, time, source IP address and hostname, destination IP address and hostname, the number of bytes sent and received, the elapsed time and the number of cpu seconds used on the firewall.

Remote administration is disabled by default. Client software is provided to enable secure encrypted remote administration using either a GUI interface or a command line interface. The systems administrator must explicitly configure the clients to allow remote administration.

Authorization to use functions can be configured to match your organization's security policy. All external users of telnet and ftp must pass strong user authentication, using security tokens. The systems administrator can control the level of authentication required for internal users. At one extreme T.Rex can be configured to allow all protected users free access to the Internet without having to provide a user ID or password. This is the simplest to use and the easiest to administer. At the other extreme the systems administrator can control which functions are available to an individual based on user ID. There are many possible ways to configure user access. In all cases the default is to deny access unless the access is explicitly permitted.

In addition to controlling access by user ID, restrictions can be placed on the source and destination addresses for individual functions. For example, it is possible to allow an individual to ftp into the secured network (after passing strong user authentication) and to only allow then to ftp to a specific host or specific subnet. Since permit and deny rules are applied after identifying the user, absolute control of services is possible based on user ID, source and destination addresses.

The permit and deny controls are easy to understand and administer. Use of wild cards on user names, group names and IP addresses provides simple yet powerful control of system access.

#### **1.6.4 Protection of the firewall system**

The firewall installation program automatically hardens the operating system to protect it from attack. All unnecessary program are deactivated or removed. The inetd and inittab files are automatically modified to disable unwanted functions and to active firewall proxies and daemons. The following services have been deactivated biff, bootps, finger, instsrv, pcnfsd, rexd, rexecd, rlogin, rshd, rstatd, rwalld, sprayd, talk, tftp. The following TCP/IP daemons have also been deactivated: lpd, routed, gated, portmap, timed and rwhod.

Specialized daemons are activated that continuously monitor the status of the firewall and ensure that it is operating correctly. The process monitor checks for processes that should be running as well as processes that should not be running. If the rules are violated the process monitor will issue an Alert and take the appropriate specified action. The firewall checks for and reports attacks including SYN flood attacks, IP spoofing, port scanning, Denial of Service attacks etc.

The proxies are designed so that they can not be used to compromise the firewall system. The proxies run without root privileges, so can not be used to access or alter critical data on the firewall. The root directories of the proxies have been modified so that the proxies can not access the majority of the file system on the firewall. The proxies do not open files for output. Thus the proxy programs are unable to access or modify any data on the firewall. The proxy is confined to a prison cell without windows and the door is welded shut. If the proxies contain a bug the error could not be used to compromise the integrity of the firewall.

The proxies have been designed with extensive error checking and reporting. Each function checks its inputs for validity and if errors are detected they are reported and the function is

terminated. The firewall software contains more than 300 documented error messages. All error messages are complete sentences. The messages provide the systems administrator with all the information necessary to determine the cause of the error and correct the problem. For example, an error in a configuration file will cause a message to be issued that contains: a description of the error, the name of the configuration file, the line number containing the error, the invalid parameter and what was expected.

The users of the proxies are not actually logged onto the firewall. Thus they can not issue commands that execute on the firewall or access or modify any data on the firewall itself.

The firewall performs automated integrity checking of every file in the Trusted Computing Base (TCB). The firewall monitor daemon (fwmon) is started at system boot time and continuously monitors the integrity of the firewall system. T.Rex uses the MD5 algorithm to detect unauthorized changes to the TCB. Each TCB file has a unique 128 bit signature. There is no known method to alter a program or file in a manner that will produce an identical signature. Unexpected changes to the system will cause a security alert and if appropriate deactivate the appropriate function(s). The fwmon program also certifies the correct configuration of the firewall, correcting and reporting any deviations.

The root user can only login from the system console. It is not possible to login as root from a remote system even if it is on a protected network. T.Rex administrators can login from secured hosts after they have passed strong user authentication. They run under restricted shells and can only execute functions defined in their restricted shell. Even if the administrator knows the root password the system does not allow the administrator to switch user to root. As a result, the only way a person can alter programs on the firewall disk is to have physical access to the system console. Thus, logical security can be related to physical security.

The firewall logs all events in the syslog and can also log events on other systems. Options are available to notify systems administrators regarding Security Alerts using e-mail and or paging software.

### **1.6.5 Protection of attached networks and hosts**

The firewall proxies provide the only method to send information between the unprotected and protected networks. Only the functions explicitly permitted are allowed to pass through the firewall.

All the proxies check for IP address spoofing. The secured networks are defined to the firewall as well as the IP addresses of the secured network interfaces. If an IP packet arrives with a secured source IP address on an unsecured network interface then the connection will be broken, a Security Alert will be issued, and the event will be logged.

### **1.6.6 Protection of individual hosts**

The primary method for protecting hosts is based on individual user identification and authentication. Basing trust on IP addresses is too weak. How do you know who is using a host with a trusted IP address? If the user is on an unsecured host then strong user authentication is mandatory. Once the user is identified then rules to permit or deny access to a host can be applied based upon user ID, group ID, source and destination addresses. Applying the permit and deny rules after discovering the users identity provides T.Rex with unequalled control.

The firewall also provides Out Of Band Authentication for client server applications that make use of the application proxy (aproxy) server. This enables secure use of shrink wrapped client server

software without requiring modification of the client server code. The user will be subjected to strong user authentication using a challenge response system. This is substantially more secure than allowing access based on an IP address. This does require the installation of a small OOBA server program to run on the users workstation or PC to listen to and respond to the challenge response system. With this system one can provide source access based on string user identification to applications such as Lotus Notes, Sybase, etc.

It is possible to control which hosts an individual user can access based upon their user ID, group ID, source and destination IP address and the function they are trying to use( telnet, ftp, Lotus notes etc.).

Encrypted data transmissions between systems is under development and will be available in a product update early this year.

## **1.7 Product Features and Mechanisms**

### **1.7.1 Services Provided**

T.Rex supports hundreds of application services and protocols. Support is provided for all major Internet services including Web Browsers, Web Servers, e-mail, FTP, Telnet, Real Audio, all TCP/IP applications as well as RPC and UDP applications.

For security reasons the following services have been deactivated on the firewall: biff, bootps, finger, instsrv, pcnfsd, rexd, rexecd, rlogin, rshd, rstatd, rwalld, sprayd, talk, tftp. The gateway routing protocols have been deactivated and replaced with static routing. The following TCP/IP daemons have also been deactivated: lpd, routed, gated, portmap, timed and rwhod.

The following is a partial list of supported applications, additional applications are supported by the general purpose application proxy (aprox) and the RPC and UDP proxies.

<b>E-Mail</b>	SKIP URL Filtering	DNS Ident IRC NTP RAS PCAnywhere SMB SNMP SNMP-trap Syslog Telnet TN3270 encrypted telnet printer (lp) X11 whois finger ping Out Of Band Authentication
SMTP (sendmail,...) POP3 IMAP Microsoft Exchange	<b>Data Base</b>  Informix Lotus Notes Microsoft SQL Server Oracle SQL*Net Sybase IBM DB2	
<b>Information Retrieval</b>		
Archie FTP anonymous FTP Gopher HTTP NNTP WAIS	<b>RPC Services</b>  Lockmanager mountd NFS NIS	
<b>Security and Authentication</b>		
HTTPS Kerberos LDAP Authentication Server RADIUS SecurID	<b>Multimedia</b>  Read Audio Real Media	<b>Circuit Gateways</b>  SOCKS 4 SOCKS 5
	<b>Standard TCP and UDP</b>	

The firewall includes the ability to restrict or allow connections based on one or more criteria. Access control for various Internet services is done using permit and deny commands based on a number of variables (depending on the service) including user-id, time of day, day of week, source and/or destination IP address, port. All activity is logged and options are available to issue alerts upon specified conditions. See below for a discussion of telnet, ftp, circuit gateways, NNTP and the generalized proxy server.

### 1.7.2 Electronic mail (e-mail)

E-mail is handled by the secure mail wrapper (smwrap). Smwrap is a small secure application that receives mail from internal and external hosts. From the outside, all e-mail appears to originate from the firewall and all in-bound e-mail must be directed to the firewall. This program is designed to reduce exposure to SMTP based attacks and scrub internal network information from out bound mail.

#### Block SMTP based Attacks

Smwrap protects against multiple attacks directed at mail servers, and mail clients. This includes checks for unauthorized use, requests to obtain access to private information, and multiple Denial of Service (DoS) attacks.



### Smwrap guards against:

- Unauthorized sender/receiver,
- Bogus Helo command,
- use of VRFY and EXPN commands,
- anonymous mail relaying,
- commands imbedded in
- Header fields, password file access,
- Root user access,
- sendmail debug exploits,
- address spoofing.

### DoS Attacks blocked:

- Helo buffer overflow,
- SMTP command buffer overflow,
- SMTP header overflow,
- SMTP Header Parsing Attack,
- Maximum number of recipients exceeded,
- Maximum message size exceeded,
- Harmful header address characters,
- MIME header buffer overflow,
- MIME field overflow,
- Idle session termination.

### Mail Blocking

Smwrap prevents annoying e-mail messages, commonly called "SPAM" from entering protected networks. The feature also blocks harassing messages making "Cyber-stalking" more difficult. The Administrators can enter a list of senders, addresses, sites, or domains they want to target for blocking. Like Call Blocking on your telephone smwrap allows you to choose from whom you want to get e-mail. Blocked e-mail can be deleted, sequestered or redirected to a specified recipient. If the mail is sequestered or redirected it can be kept as evidence along with the log information.

The **aliasreq** command permits control of who is allowed to send mail outbound through the firewall. If a user is not registered in the alias data base then any attempt to send mail through the firewall will be rejected and a Security Alert will be issued.

Smwrap translates internal e-mail addresses to external e-mail addresses. This translation includes all internal addresses that are part of a Carbon Copy (Cc:) or To: addresses. Translation support for addresses generated by Novell's GroupWise and MS Mail Exchanger is also provided.

Smwrap deletes partially completed store and forward files from the hermes directory when there is an unexpected EOF from the remote client or I/O error.

## 1.7.3 Application Proxy Server

Aproxy provides generic proxy services for connection oriented TCP/IP applications with access control and logging. Aproxy is a standalone proxy that is started at system boot time. It reads the aproxy configuration file and parses the rules into a table stored in RAM. Based on the rules aproxy pre-forks an initial number of processes and listens to the specified ports. When a client connects to a port to which aproxy is listening it checks the rules and determines if the connection

shall be permitted. If NOT permitted the connection is closed and the attempt is logged. If the requesting host is allowed to connect the permit rule determines the IP address and port number of the host on the other side. Permissions are granted based on the source IP address and port. Entries in the configuration file also determine the destination host and port number.

Aproxy supports one-to-one and many-to-one connections. Many-to-one connections can be permitted using multiple permit statements or by using wildcards to specify a range of source addresses on a permit rule. Aproxy is designed so that it can not be used to compromise the security and integrity of the firewall. The program runs in a non-privileged state chrooted to a restricted directory. The only file I/O it performs is to read configuration files and to log user activity via the syslog daemon. The user is unable to issue any commands or shells that run on the firewall. The user is not logged on to the firewall. Commands and data simply pass through the firewall.

In addition to controlling client-server connectivity aproxy also provides an API and server workload balancing.

## **API**

Aproxy has an Application Programming Interface (API) to allow it to be customized for any number of applications. The API enables fine grained control over any applications.

Each permit statement provides three application exits. One for authorization and authentication. The second, provides access to the data read from the client. The third provides access to data read from the server.

## **Work Load Balancing**

Aproxy provides workload balancing to distribute application traffic across an array of servers. Aproxy controls and balances traffic at the application (port) level. Aproxy supports two load balancing methods: round robin and ratio. If one of the servers is twice as fast aproxy can be configured to distribute it twice as much work.

If one of the servers is off line for any reason aproxy will automatically stop scheduling that server until it comes back on-line.

## **Out Of Band System Authentication (OOBA)**

Aproxy can perform Out Of Band System Authentication to provide secure and transparent access to applications through the firewall. This can be done without requiring application modifications. When a system connects to a port on the firewall supported by the genproxy server the following steps occur.

1. Aproxy searches the rules data base for a match.
2. If a match is found and OOBA is specified the aproxy connects to the remote system using the port number following OOBA.
3. If there is no response from the remote host the connection is broken. If the connection is made the genproxy program issues a 8 digit random number as a challenge. The remote host then must generate an 8 digit hexadecimal response that matches the expected response generated by genproxy.

4. If the response is correct the connection between the client and server systems is made. Otherwise the connection is broken and a Security Alert is issued.

### 1.7.4 HTTP Proxy

The HTTP proxy provides secure access to the World Wide Web (WWW). The HTTP proxy is a derivative of the Apache Web Server that can only function as a proxy. Functions not required to support the proxy have been removed. Other changes have been added to enhance security, support content filtering and provide for a high speed parallel I/O cache. The httpd proxy provides multiple functions including:

- Basic browsing services,
- HTTP 1.1 protocol support,
- Enforcement of management policies,
- URL Content Filtering,
- Selective blocking of Java, JavaScripts, ActiveX and cookies,
- Caching of frequently accessed documents to improve Web response time,
- A Fault Tolerant design,
- HTTP Access logging in Common Log Format.

A HTTP summary report program is provided to produce meaningful reports of HTTP activity.

One or more copies of the HTTP proxy can also be run behind the firewall on one of the supported systems (AIX 4.2+, Solaris 2.6 for SPARC and X86, as well as Linux). Use of HTTPD behind the firewall still requires the use of HTTPD on the firewall. Use of cascaded proxies allows caching to be moved closer to the end users to deliver better performance, and to optimize internal network bandwidth. This mode of operation off-loads 40-50% of the HTTP proxy service from the firewall.

**WAIS:** This is supported by HTTP browsers such as Mosaic and NetScape. It is also supported by the Apache HTTP proxy server which ships with T.Rex.

**Gopher:** The HTTP proxy server supports Gopher.

### 1.7.5 Web Content Filtering

The WWW provides unprecedented access to information on the Web. Due to the variety of information available management is concerned about downloading of information deemed inappropriate for their organization. The HTTP proxy server is capable of blocking Web access based on several different criteria. This provides management with an effective tool for selectively blocking access to information deemed inappropriate for the organization. Web access can be blocked a number of ways. Blocking can be performed on a URL, host name, IP address or by key words that may appear in the URL.

### 1.7.6 Java, JavaScript and ActiveX Blocking

Despite assurances from SUN and Microsoft about the security of these programming and scripting languages, Java, JavaScript and ActiveX are inherently insecure and dangerous to today's web browsing community. Holes in the Java class loaders and the bytecode verifier can be exploited so that malicious applets can execute arbitrary machine code. ActiveX scripts can run any application on the browsing machine and make calls to the operating systems.

The httpd proxy allows selective blocking of Java applets, JavaScripts, ActiveX and cookies. The HTTPD proxy provided with T.Rex supports permit and deny commands for all these functions. Depending on the organizations security policy it is possible to permit all access to Java, block all access for Java, or some combination in between. The same controls are available for JavaScript and ActiveX.

### 1.7.7 Reverse HTTP Proxy

The webgate proxy is a reverse proxy designed to protect web servers behind the firewall. Webgate supports workload balancing between two or more web servers. This allows for higher system throughput and it isolates web browsers from planned and unplanned outages of a single web server.

### 1.7.8 Telnet Proxy Overview

The telnet proxy (tnproxy) provides telnet access through the T.Rex firewall without compromising the security and integrity of the firewall. Tnproxy supports access control by user ID, source IP-address and destination IP-addresses. Strong user authentication is required for unprotected hosts and is optional for protected hosts. Tnproxy also supports transparent Internet access for internal users. If transparent access is specified then internal users are not prompted for user IDs or passwords at the firewall. All successful and unsuccessful connection attempts are logged, including date, time, source and destination IP- addresses, and the number of bytes sent and received. Utilities are provided which summarize tnproxy activity.

The tnproxy program runs in a non-privileged state chrooted to a restricted directory. The only file I/O it performs is to read configuration files and to log user activity via the syslog daemon. The user is unable to issue any commands or shell scripts that run on the firewall. The user is not logged on to the firewall. Commands and data simply pass through the firewall. Standard telnet is also available to allow systems administrators to login to the firewall to perform administrative functions.

Permit and deny rules control access based on user or group ID and client host IP-address and destination IP-address. Tnproxy is invoked by the Internet daemon inetd when a telnet client on another host wants to telnet to another system through the T.Rex firewall. The tnproxy reads the securenets file to determine if the source host is on a protected network. The method of user authentication is determined by the security policies defined to the proxy and the security status of the clients host. If the user is connecting from a secured host and tnproxy has been configured with transparent access the user is not required to present a user id or password.

Secured host requiring user authentication: If the user is connecting from a secured host and tnproxy has been configured to authenticate internal users it then reads the gwuser file to determine if the user has telnet privileges at the current time. The user is prompted for a password.

Unsecured host: If the user is connecting from an unsecured host tnproxy asks for the users id. It then reads the gwuser file to determine if the user has telnet privileges at the current time and what form of user authentication to use, such a CRYPTOCARD, SecureNet Key or Security Dynamics. If a challenge response system is specified the user is challenged with a random 8 digit decimal number. The user must respond with a 8 digit hexadecimal number generated by a preprogrammed security token. There is only one chance in 4,294,967,296 of guessing the correct response.

In all cases if the user fails authentication or is denied telnet access a message is sent to the user, the connection is broken and the attempt is logged. After passing the authentication test the user is

prompted for the desired destination. Tnproxy uses the rules defined in its configuration file to determine if the connection is permitted or denied. If permitted the connection is made and logged. If denied a message is sent to the user, the connection is broken and the error is logged.

#### Tnproxy Access Control Rules

Connections between source and destination hosts are permitted or denied based upon permit and deny rules. The deny command will deny access to a user or a group of users if there is a match up of the source and destination IP-addresses. The permit command will permit access to user or a group of users if there is a match up of the source and destination IP-addresses. Wildcards "\*" can be used to define users, groups or IP addresses.

#### Advanced Controls:

The permit and deny rules can be used to control user access for multiple networks or subnets attached to the firewall.

### **1.7.9 FTP Proxy Overview**

The FTP proxy (ftproxy) provides ftp access through the firewall without compromising the security and integrity of the firewall. Ftpoxy supports access control by user ID, source IP- address and destination IP-addresses. Strong user authentication is required for unprotected hosts and is optional for protected hosts. All successful and unsuccessful connection attempts are logged, including times, source and destination IP-addresses, and the number of bytes sent and received. Utilities are provided which summarize ftpoxy activity.

The ftpoxy program runs in a non-privileged state chrooted to a restricted directory. The only file I/O it performs is to read configuration files and to log user activity via the syslog daemon. The user is unable to issue any commands or shell scripts that run on the firewall. The user is not logged on to the firewall. Commands and data simply pass through the firewall. System administrators can still use the ftp client on the firewall to send and receive data from the firewall system.

Permit and deny rules control access based on user or group ID as well as client host IP-address.

Ftpoxy is invoked by the Internet daemon inetd when a ftp client on another host wants to ftp to another system through the firewall. The ftpoxy reads the securenets file to determine if the source host is on a protected network. The method of user authentication is determined by the security policies defined to the proxy and the security status of the client host.

Secured host with transparent access: If the user is connecting from a secured host and ftpoxy has been configured with transparent access the user is not required to present a user id or password. They simply specify the remote host they want to connect to.

Secured host requiring user authentication: If the user is connecting from a secured host and ftpoxy has been configured to authenticate internal users it then reads the gwuser file to determine if the user has ftp privileges at the current time. The user is prompted for a password.

Unsecured host: If the user is connecting from an unsecured host ftpoxy asks for the users id. It then reads the gwuser file to determine if the user has ftp privileges at the current time and what form of user authentication to use, such a CRYPTOCARD, SecureNet Key or Security Dynamics. If a challenge response system is specified the user is challenged with a random 8 digit decimal number. The user must respond with a 8 digit hexadecimal number generated by a preprogrammed security token. There is only one chance in 4,294,967,296 of guessing the correct

response.

In all cases if the user fails authentication or is denied ftp access a message is sent, the connection is broken and the attempt is logged. After passing the authentication test the user is prompted for the desired destination. ftpoxy uses the rules defined in its configuration file to determine if the connection is permitted or denied. If permitted the connection is made and logged. If denied a message is sent to the user, the connection is broken and the error is logged.

#### Ftpoxy Access Control Rules

Connections between source and destination hosts are permitted or denied based upon permit and deny rules. The order in which rules are coded is very important, since the first rule that matches is enforced. Subsequent rules are ignored. The deny command will deny access to a user or a group of users if there is a match up of the source and destination IP-addresses. The permit command will permit access to user or a group of users if there is a match up of the source and destination IP-addresses. Wildcards "\*" can be used to define users, groups or IP addresses. If you want to permit all secured users to have ftp access to the unsecured networks all that is required is a single permit rule.

Advanced controls:

The permit and deny rules can be used to control user access for multiple networks or subnets attached to the firewall.

### **1.7.10 RPC and UDP Proxy**

The RPCproxy provides secure access to RPC and UDP applications. Permit rules allow either TCP or UDP based RPC applications based on the clients IP address, and the target address and port numbers.

Rpcproxy maintains stateful connections for UDP sessions. Only those UDP packets associated with an authorized client/server pair are allowed to pass through the firewall. Only the client is allowed to initiate the first transfer. Sessions are automatically terminated when the client and server are idle for a period equal to the timeout parameter.

### **1.7.11 Network News (NNTP)**

T.Rex provides a generalized configurable proxy server that allows Internal hosts to gain transparent access to specified external NNTP servers. The configurable proxy server allows connection-oriented TCP/IP applications to communicate through the firewall. The proxy server provides access control and logging for connections between internal and external systems. Permit commands stored in a configuration file allow for precise control of one-to-one and many-to-one connections. Permit commands control access by source IP address and port, and destination IP address and port.

### **1.7.12 Dual DNS Servers**

T.Rex supports separate external and internal Domain Name Servers (DNS). Users on internal hosts can resolve host names into IP address for both the internal and external systems. Users of external systems cannot access or even see the internal DNS. The internal network is not IP addressable and is completely hidden from the outside.

The caching-only name server that runs on the firewall is used by internal hosts to resolve external names. Some of the proxies use both the external and internal DNS servers to provide authorized users with a seamless name space.

**whois:** A whois client is available on the T.Rex firewall.

**finger:** A finger client is available on the firewall allowing protected users to finger external systems. The finger daemon is deactivated so that unprotected users can not finger the protected network. A socksified version of finger is available to allow users on protected hosts to finger external systems in a transparent manner.

**ping:** A ping client is available on the firewall to allow the administrator to ping external systems. External users are only able to ping the external firewall ports. Since we have disabled IP packet forwarding, the internal systems are not addressable by external hosts and thus are invisible to the outside world. A socksified ping client is also available for transparent pinging.

### 1.7.13 SOCKS Circuit Gateway

SOCKS is a public domain software package that allows hosts behind a firewall to gain full access to the Internet in a transparent manner. This is done in a secure fashion that does not require direct IP reachability. The SOCKS server runs on the firewall and socks clients run on the protected hosts. Socksified versions of ftp, telnet, finger, whois and Mosaic are available. Commercial Web browsers such as the NetScape Navigator and MS IE are also socksified, FAS has installed and tested the SOCKS daemon on the T.Rex firewall. FAS provides a binary version of the SOCKS server and sample configuration files at no extra cost to their customers. FAS has also built and tested socksified versions of the telnet, ftp and finger clients for the AIX, HP-UX and Solaris. Binary versions of these clients are provided to FAS customers at no cost.

There is also a collection of socket client applications that run on MS Windows behind a firewall running the SOCKS proxy server. The following applications currently work behind SOCKS: NetScape Navigator, telnet, ftp, finger, and Mosaic. These can be found on the Internet.

### 1.7.14 Real Audio

The RealAudio Proxy (raproxy) allows users behind the T.Rex firewall to gain access to RealAudio applications. Raproxy provides access controls to permit and deny access based upon source and destination IP addresses. Raproxy can also limit the bandwidth used for multi-media applications.

## 1.8 Product Audit/Event Reporting and Summaries

Since the firewall system deals with security, its important to know how to track who logs in, who uses various services, and who tries to get in without permission. The firewall administrator can select multiple levels of logging to be performed at the firewall. Most of the daemons will write log data to one or more systems logs using up to eight different priority levels. The systems administrator can select which message levels will be recorded and where they will be stored. Some of the proxies such as httpd and webgate have specialized logging facilities.

The /etc/syslog.conf controls which messages are recorded and where they are logged. The following facilities can be specified in the syslog configuration file: kern, user, mail, daemon, auth, syslog, lpr, news, uucp and \* for all messages. The priorities which can be specified are (from high to low): emerg/panic, alert, crit, err, warn, notice, info, and debug. The destination of the message

can be a specified file on a system, or a user id on a system. To send the message to another system follow the file name or the user id by the character string @hostname. All proxy servers write messages to the system log. Each proxy server message contains the name of the proxy server to identify the source of the message. This can be used to search for messages issued by a specific program on a specified date.

A special log daemon is provided that can send one or more log files to remote systems within the protected network. The remote system can process the data in real-time, batch mode, or simply archive the data for later use.

The Hoplite tools provides a remote console that can display the systems logs in real time. Selection criteria allow the administrator to filter what is displayed.

An automated log management program rotates logs using rules that control the frequency of rotation, the number of levels to keep, and the disposition of the logs that roll off of the active queue. This program requires no operator intervention and keeps the system from running out of disk space.

A suite of report programs is available that can produce more than 52 unique reports. There are multiple reports for each of the proxy services. The report programs are written in C and can produce multiple reports with one pass of the log data. This allows data to be analyzed at the rate of 300 MB per minute, a requirement for processing large amounts of log data.

Report scripts are available to produce adhoc queries and reports.

UNIX provides simple commands that allow the systems administrator to customize reports based on system log data. The data can be examined by date, type of service or type of message.

### **Aproxy Reports**

The aproxsum program produces the following reports showing how aproxy is being used:

- aproxy summary statistics
- top hosts by bytes sent
- to hosts by connect requests
- top hosts by bytes received
- to hosts by received connects

### **E-mail Reports**

The smrpt program produces reports showing how e-mail is being used. It produces the following reports:

- e-mail summary report,
- top users sorted by the number of messages received,
- top users sorted by the number of bytes received,
- top users sorted by the number of messages sent,
- top users sorted by the number of bytes sent.

The smrptx program produces the following e-mail exception reports:

VRFY: All attempts to use the VRFY command to obtain user IDs on the firewall are reported. Excessive use of the VRFY command from an unprotected host can be the sign of a cracker.



EXPN: All attempts to use the EXPN command to extract user information are reported.

Warning: All smwrap and smwrapd warning messages are printed. This helps find configuration and systems problems.

Error: All the Error messages issued by either smwrap or smwrapd are reported.

Security Alert: All the Security Alerts issued by smwrap or smwrapd.

Sequestered: All the files that can not be delivered and the reason they were sequestered are reported.

## HTTP Reports

The Smart Web Analysis Tool (swat) produces up to 27 reports including:

Summary Report	Directory Report
Monthly Report	File Type Report
Weekly Report	File Size Report
Daily Report	Status Code Report
Daily Summary	Request Report
Hourly Report	Redirection Report
Hourly Summary	Failure Report
Quarter-Hour Report	Failed User Report
Five-Minute Report	Failed Referrer Report
Browser Report	Redirected Referrer Report
Browser Summary	Referrer Site Report
Domain Report	
Host Report	

The HTTP reports can be generated in three formats:

HTML	- to allow web browser access
Delimited Text	- for input to spreadsheets
Text	- to be e-mailed

The swat program works with the http access log produced by the httpd proxy or the log produced by the reverse http proxy (webgate). Thus you can tell how your users are browsing the web and if you have a web server you can analyze how the Internet community is using your web server.

## FTP Reports

The ftprpt program produces reports that show how ftp is being used.

FTP summary statistics  
top users sorted by bytes sent  
top users sorted by bytes received  
top users sorted by cpu time used

## Telnet Reports

The tnprpt program produces reports that show how telnet is being used.

Tnproxy summary statistics  
top users sorted by bytes sent  
top users sorted by bytes received  
top users sorted by cpu time used

### **Exception Reports**

All Security Alerts in chronological order.  
All hardware, software and configuration errors listed in chronological order

### **SOCKS Reports**

If you are using the SOCKS Circuit Gateway the socksum program produces reports on how socks is being used.

## **1.8.1 Manner in which logs are stored and safeguarded**

The syslog on the firewall system is only available to the firewall administrator. Even the proxies do not have access to the file system that contains the syslog. If a hacker was given the password for the root user they could not login remotely as root. Remote logins by the administrator require special encrypted clients and strong user authentication. The use of strong user authentication stymies attempts to access the gateway machine by unauthorized individuals. Logged events can also be sent to another system for safe keeping.

## **1.9 Product Testing Methodology**

The firewall product is developed at Livermore Software Laboratories in Houston Texas. The lead developers have decades of experience working with bet-your-life mission-critical software. They have applied the same design, development and testing procedures they used to develop software for supporting manned space flight.

Our Quality Assurance procedures rigorous reviews at all levels, automated testing procedures that include scripts and program driven testing. Procedures test for normal and abnormal cases to ensure the system works as designed, and can also detect, report and recovery from attacks, as well as hardware and software failures without any loss of security or integrity. The product testing procedures include:

- Peer design reviews,
- Structured Code walk-throughs,
- Unit testing,
- System testing,
- Basic Function testing,
- Acceptance testing,
- Installation Testing,
- Upgrade Testing,
- De-installation Testing,
- Regression Testing,
- Boundary Testing,
- Testing for previously detected errors,
- Stress Testing,
- Performance Testing,
- Penetration Testing (Saint, ISS, and other tools),
- Denial of Service Testing, and
- Verification that all CERT Advisors are accounted for.

In addition to examining the output of the tools we verify the firewall has properly logged the attacks and issued the appropriate alerts.

Stress testing tools are also used to assure the firewall functions properly under high levels of stress. Hardware and software errors are also simulated to assure the error is detected, reported and that automated recovery procedures are executed.

Before a new release can be announced: all functions must work as specified, all known methods of attack are defeated, all attempts to bypass the security have failed, there are no major problems, the documentation is correct, and the product meets our goals for performance and reliability.

## 1.10 Product Performance Attributes

Firewall performance and throughput vary greatly based on the specific traffic mix and the size of the system hosting the firewall. The T.Rex firewall is efficiently designed to provide maximum system throughput without compromising system or network security. Its support of SMP machines allows it to be scaled up to support large numbers of users and multiple high speed networks. The performance numbers we provide are based on measurements with all the security features activated. Some of the other firewall vendors have published throughput numbers with the firewall features deactivated.

The T.Rex firewall supports networks with more than 30,000 users.

Web browsing activity tends to dominate firewall resource requirements as it normally accounts for the largest amount of traffic in terms of transactions and data volume. Use of the Integrated web cache and URL content filter will improve web response times and conserve the network bandwidth. To maximize web performance use to following guidelines.:

Product	TREX-ES12sx	TREXS-ES12m	TREX RS/6000 F50	TREX-ES12i
CPU	two Sun UltraSPARC 400 MHz	two Motorola PowerPC 604e 400 MHz	two IBM 604e-X5 332 MHz	two Pentium III Xeon 550 MHz
RAM	256 MB	256 MB	256 MB	256 MB
NICs	four 10/100 Ethernet	four 10/100 Ethernet	four 10/100 Ethernet	four 10/100 Ethernet
OS	Sun Solaris 2.6	Motorola AIX 4.2	IBM AIX 4.3	Red Hat Linux 6.1
MAX HTTP bps	135 mbps	110 mbps	135 mbps	135 mbps
CPU Utilization	42%	90%	50%	28%
MAX FTP bps	148	114	148	148
CPU Utilization	43%	90%	52%	53%
VPN hardware assisted	N/A	N/A	N/A	100 Mbps
CPU Utilization	N/A	N/A	N/A	5%

Faster models are available.

## 1.11 Product Operational Assumptions

The firewall runs on multiple hardware platforms and operating systems and is available on embedded systems platforms.

### Platforms

HP-PA 9000/700 and 800  
IBM RS/6000 and PowerPC  
Intel Pentium III and Xeon  
Motorola MATX Embedded systems  
Sun Ultra SPARC systems.

### Operating Systems

AIX 4.2, 4.3  
HP-UX 10.10 and 10.20  
Red Hat Linux 6.1  
Solaris 2.6, 7

### Embedded Systems

TREX-ES11i through TREX-ES28i

The firewall will run on Compaq Servers, Dell Servers, IBM Netfinity Servers, and any other Pentium III servers supported by Red Hat Linux.6.1 etc.

The system should have two or more network adapters, a display, mouse, keyboard, a 3.5" floppy diskette drive, hard drive, CD-ROM and an optional tape backup system.

**Memory Requirements:** The T.Rex system should be configured with at least 64 MB of ECC memory. Most T.Rex systems have 128 MB or more. More may be required depending on the number of concurrently active users.

**ECC Memory:** FAS strongly recommends the use of ECC memory to ensure the integrity, security, reliability and availability of the T.Rex system. ECC memory dynamically detects and corrects parity errors in the memory. If non-parity memory is used on the firewall system a hardware error in the memory can go undetected and possibly alter the behavior of the system. An undetected parity error could result in data corruption, a system failure or in the worst case it could make your network vulnerable to attack. Use of ECC memory eliminates this type of problem.

**Disk Space Requirements:** The T.Rex system should be configured with at least four GB of disk space. The operating systems (AIX, Linux, Solaris) require less than 400 MB of disk space. T.Rex software requires approximately 20 MB of disk space. The system requires less than 0.6 KB per user ID. Most of the disk space will be used by various system logs and httpd cache. The amount of space required for system logs depends on upon user activity. The size of each log entry varies by application. The following table shows the average number of log bytes by application. The HTTP access log usually consumes the most disk space of any application.

Estimated space required for system log entries.

- e-mail Every e-mail message generates two syslog entries using an average of 200 bytes per message.
- httpd 150 bytes per GET or POST operation. On the average there are 3 to 4 GETS every time the user selects a page entry.
- ftp 200 bytes per connection, plus 120 bytes per file transferred.

For best HTTP performance:

- Provide 32 MB of RAM for every 100 concurrently active Web Browsers.
- Dedicate two UW SCSI disk drives for every Megabit per second of Web browsing data traffic to handle the I/O requirements of the cache.
- Use separate UW SCSI disks for the syslog and cache to avoid I/O contention.

**Communication Hardware:** T.Rex is a multi-homed firewall that supports two or more communication adapters. There must be at least one communication adapter connected to a secure network and at least one communication adapter connected to an unsecured network. Any combination of communication adapters supported by TCP/IP will work with T.Rex. This includes Ethernet, Fast Ethernet (100 mbps) and Gigabit Ethernet, Token Ring, ATM (155 and 630 Mbps) and S/390 ESCON Channel Adapters.

**Security Tokens:** Use of telnet from unsecured networks requires strong user authentication. This requires the user to possess a security token (smart card) to pass the challenge presented at login time. T.Rex uses a challenge-response system that defeats eavesdroppers who might have picked up valid user IDs and passwords from the network. Challenges and valid responses change with every login attempt. Without access to the key it is virtually impossible to break in. T.Rex Version 1 supports the following security tokens:

#### **CRYPTOCARD**

CRYPTOCards are available from [www.cryptocard.com](http://www.cryptocard.com).

#### **SecureNet Key Card**

SecureNet Keys are available from Digital Pathways, 201 Ravendale Dr., Mt. View, CA. 94043. Their phone number is (415)-964-0707

## **1.12 Product Operational/management requirements**

The firewall is administered through a Graphical User Interface called Hoplite that can run on the firewall or a remote workstation. The GUI client is a Java Application that requires the use of the Java Virtual Machine. Remote Administration uses triple DES encryption to secure communications between the client and the server and it requires strong user authentication.

Hoplite also provides real-time Alert Panels and a Log Viewer. The log viewer allows the administrator to filter the display of log data to show what is of interest. Context sensitive help is available on-line.

A command line interface is also provided for system and firewall administration. A special encrypted version of telnet (ptelnet) is provided for secure remote communications.

GUI System administration tools are available for systems management.

Most administration chores are automated to minimize administration overhead. Log management utilities automatically manage system and proxy logs. Automatic log rotation and archiving prevent the system from running out of disk space.

### **1.13 Product Customer Support**

The T.Rex firewall is a licensed program product of FAS, distributed and supported worldwide. Multiple levels of support are available to meet the requirements of any organization. See Chapter 2 for more information.

For installation, configuration and training services refer to [www.opensourcefirewall.com](http://www.opensourcefirewall.com)

### **1.14 Product Inter-Operability Considerations**

There are no special requirements placed on the users of the firewall. They will need to know how to run thier favorite web browser or ftp, or telnet or mail client.

The VPN tunnel follows IPSec standards and will interoperate with the Red Creek Ravlin Ipsec products, etc.

notes:

## Chapter 2. Customer Support



T.Rex Version 1 is backed by a system designed to provide fast courteous service. If you need assistance beyond what is offered in the *Installation and Administration Guide* or the *User's Guide* then follow the steps listed below.

- Try to duplicate the problem and document exactly what was done.
- Have the following information ready:

- The T.Rex version number
- The T.Rex License number
- The machine make and model including the types of network adapters
- External and internal IP addresses
- The exact error message(if you received one)
- The version number of the operating system.

If possible we recommend that you be at the computer when you call.

The FAS T.Rex Firewall System support number is:

Telephone: 281-759-3274  
FAX: 281-759-8558  
e-mail: [trex@opensourcefirewall.com](mailto:trex@opensourcefirewall.com)

Regular support hours:

8:00 am to 5:00 PM, Monday through Friday, Central Time

When you call our customer support representative may ask you for your name and phone number, and the name of your company or organization.

T.Rex Version Number \_\_\_\_\_

T.Rex Support Contract # \_\_\_\_\_



Notes:

# Chapter 3. T.Rex Installation

## 3.1 Installation Overview

This chapter explains how to install and configure the T.Rex software. After completing the instructions in this chapter you will have installed T.Rex and done the initial configuration. The system has been designed to make installation as painless as possible, however, should you run into any problems, feel free to call our customer support number (see page 5).

**T.Rex should be installed on a fresh copy of the OS to ensure that the underlying OS has not been compromised prior to the T.Rex installation.** You should also check to make certain that all required maintenance has been performed on the OS prior to the T.Rex installation. See Section 3.8 for details.

There are minor difference in the installation process for each vendors version of UNIX supported. In order to minimize confusion we have created separate sections for each version of UNIX and Linux as shown below:

AIX	Chapter 3.3
HP-UX	Chapter 3.4
Linux	Chapter 3.5
Solaris	Chapter 3.6

## Before You Begin

Before you begin the installation you will need the following information.

1. The time zone of your system.
2. The IP address(s) of the protected networks.
3. The IP addresses(s) of the firewall's protected network interface(s).
4. The host and domain names for the gateway.
5. The IP address(s) for the unprotected network interface(s).
6. The IP address of the protected (internal) DNS.
7. The type of LANs (Ethernet: thick, thin; Token/Ring ...).
8. The IP address of your internal e-mail server (if you use one).
9. The organizations policy for Internet access. For example, will secured users have unrestricted access to the Internet or will secured users be given granted controlled access?
10. The IP address of an external NNTP (News) server, if you plan to allow access to NNTP servers.
11. The IP address of an internal NNTP server if you plan to use controlled NNTP access. This will allow the systems administrator to control access to News feeds.

If you plan to allow telnet or FTP access from the unsecured (Internet) network you will need to do the following:

1. Select the type of security token to be used by the initial user(s).
2. Decide if these users will be granted both telnet and FTP access from the external network.

## 3.2 OS Installation

### Before installing T.Rex

The default installation of most operating systems is inappropriate for a firewall. The reasons are many including poor allocation of file systems and disk space, installation of extraneous and potentially dangerous software, and failing to install necessary server applications such as `inetd` and `named`. Further, there is a need to limit the number of user id's that are allowed to access the system. For these reasons it is **strongly recommend** that a clean installation be performed that overwrites all of the data on the hard drive. This will help eliminate problems that may jeopardize security and minimize headaches associated with file systems filling up.

During installation you will be confronted with many options. It is important to choose the following:

1. Choose a server setup. Customization is preferred to add optional packages such as the Java run-time environment, man pages, etc. and avoid installing unnecessary packages such as compilers. If you are unsure of what to install, then do not customize. Packages can always be added later and unwanted server packages such as NFS will be disabled by the T.Rex installation.
2. Choose other/DNS for name service (NIS is not supported by T.rer for security reasons).
3. Customize the file system structure and disk allocation. The file system needs will vary depending on the type of operating system and the types of traffic that your firewall services. Below is a list of recommended file systems, minimum allocation size, and what their purpose is with regards to the firewall. This should help in determining what is best for your unique situation.

/	100 MB for most operating systems and 30 MB for AIX. Directories like <code>/etc</code> that are not defined as file systems will be under <code>/</code> . This is a fairly static file system typically holding configuration information.
/usr	400MB This file system holds the executable and is relatively static. Leave room to patch old packages and add new ones (100MB or so).
/var	500MB This file system holds the logs and sendmail's mail queue and dead_letter files. Firewall's for e-commerce or other high volume organizations will want considerably more space allocated to <code>/var</code> as the logs will grow at a tremendous rate.
/home	500MB This is for user's storage area, in particular, the proxy user id hermes. Temporary storage for the secure mail wrappers queue and the <code>httpd</code> cache reside here. Allocate more space if the firewall is to serve many browsers.

NOTE: For best performance, it is recommended that the `httpd` cache be spread out over several hard drives with the remaining file systems on hard drivers separate from the cache drives.

/tmp	100MB This is used as temporary storage by many applications. Being a separate file system helps to avoid race-conditions.
------	----------------------------------------------------------------------------------------------------------------------------

NOTE: Solaris calls the swap space `/tmp`, so that there is no distinction between the two.

swap at least equal to the amount of RAM. This is used as temporary storage for the kernel. For Linux the maximum size of swap is 128 MB. Additional 128Mb swap partitions can be defined to increase the swap space available to the kernel.

Once installation is complete, patch the operating system and various applications to the most current revision. This protects your firewall against well known attacks, Y2K issues, and bugs. Now you are ready to install T.Rex.

Note that many patches from the various operating system vendors will undo T.Rex security modifications. This is why patching before the T.Rex install is suggested. When future patches are released, add them one at a time and verify that configuration and startup files have not been modified. Typically, a patch will rename these files with a ".old" suffix so they may be inspected for differences. Do not add patches to software that are not currently installed or needed.

## 3.3 AIX Installation

The following describes how to copy and install the diskettes or CD-ROM on your AIX 4.x system.

### 3.3.1 Install T.Rex Firewall Software on AIX

#### (A1a) For CD-ROM

Login as root. Insert the T.Rex CD-ROM and issue the following command.

```
# mount -oro -v cdrfs /dev/cd0 /mnt      ( or your device name & mount point)
# cd /mnt/aix/T.Rex
```

Read the README.txt, license.txt and RELNOTES.txt.  
When ready to install enter the following command.

```
# ./install_T.Rex
```

Remove CD-ROM, and go to step 3.2.2.

```
# umount /mnt or umount file system /dev/cd0
```

#### (A1b) For Floppy disk

Login to the system as root and issue the following commands:

```
# cd /usr
# mkdir T.Rex
# cd T.Rex
```

Insert the T.Rex Installation diskette(s) and issue the following command.

```
# tar xvf /dev/fd0      ( or your device name, if different)
```

Remove floppy diskette.

Read the following files: README.txt, license.txt, and RELNOTES.txt.  
When you are ready to install type in the following command.

```
# ./install_T.Rex
```

The install\_T.Rex process creates a log of all changes made to the system. You can review these changes in the /usr/T.Rex/install\_T.Rex.log file.

### 3.3.2 Configure AIX system

#### (A2) Configure LAN adapters

Use IBM's **SMIT** utility to configure all the LAN adapters on the firewall. Login to the T.Rex system as *root*, since many of the installation steps require root access. Configure the system to support the network adapters. This requires two steps. First, use SMIT to configure the LAN adapter. Second, use SMIT to configure the network interface. Detailed instructions for installing and

configuring network adapters can be found using IBM's InfoExplorer, under the topic "TCP/IP Network Adapter Cards." You should be familiar with the use of IBM's System Management Interface Tool (SMIT).

For your convenience we have summarized the configuration process found in InfoExplorer. In this example we assume you are installing an Ethernet adapter.

1. Enter SMIT `chgenet` to use SMIT.
2. Select an available Ethernet adapter, such as `ent0`.
3. Specify whether the Ethernet is thick (dix) or thin (bnc).
4. exit SMIT.

### **(A3) Configure the network interface using SMIT.**

1. Enter SMIT `chinet`.
2. Select an available Network Interface from the list presented.
3. Specify the Internet address of the Network Adapter in the standard dotted decimal format (eg `192.168.240.1`).
4. Specify the optional network mask if desired (eg. `255.255.255.0`).
5. Specify yes for the Address Resolution Protocol (ARP). This resolves IP addresses into Ethernet addresses.

**Repeat this process for each Network Adapter.**

### **(A4) Configure fully qualified host name**

Use the SMIT utility to define the fully qualified hostname ( `host.domain`).

### **(A5) Set up static routing**

Routing must be done statically on the T.Rex firewall system. Set up static routing, so that all default traffic is routed to the external network and route each internal network with a separate route statement. You can use the SMIT utility program to define static routing or directly edit the `/etc/rc.net` file as shown in Appendix B. If you choose the manual approach you must also remove any statements from the start up file that run a dynamic routing protocol.

**Warning:** If you used SMIT to select the `/etc/rc.bsdnet` option then the static routes defined in `rc.net` will not be used. .

Go to Chapter 3.5 to perform configuration of T.Rex files.

## 3.4 HP-UX Installation

The following describes how to copy and install the diskettes or CD-ROM on your HP-UX 10.x system.

### 3.4.1 Install T.Rex Firewall Software on HP-UX

#### (H1a) For CD-ROM

Login as root. Insert the T.Rex CD-ROM and issue the following commands:

```
# mount -r -s cdfs /dev/dsk/c201d2s0 /cdrom
# cd /cdrom
# cd cd hpux/T.Rex
```

Read the README.txt, license.txt and RELNOTES.txt files.  
When ready to install enter the following command.

```
# ./install_T.Rex
```

Remove the CD and go to step 3.3.2

#### (H1b) For Floppy disk

Login as root and issue the following commands:

```
# cd /usr
# mkdir T.Rex
# cd T.Rex
```

Insert the T.Rex Installation diskette(s) and issue the following command.

```
# tar xvf /dev/floppy/c0t1d0          ( or your device name, if different)
```

Remove floppy diskette or CD-ROM.  
Read the following files: README.txt, license.txt, and RELNOTES.txt.  
When you are ready to install type in the following command.

```
# ./install_T.Rex
```

The install\_T.Rex process creates a log of all changes made to the system. You can review these changes in the /usr/T.Rex/install\_T.Rex.log file.

### 3.4.2 Configure HP-UX system

#### (H2) Change root password

Execute the **passwd** command to select a new root password of six to eight characters.

If the system has been changed from non-trusted to trusted system mode (see `install_T.Rex.log`), enter only the first eight characters of the old password when prompted. Although all password characters (not just the first eight) are significant in trusted system mode, passwords longer than eight will not work with vuelock.

### **(H3) Configure LAN adapters**

Use HP's System Administration Management(**SAM**) utility to configure all the LAN adapters on the firewall.

### **(H4) Define fully qualified host name**

Edit the `/etc/rc.config.d/netconf` file to define the fully qualified host name and static routes.

Go to Chapter 3.5 to perform configuration of T.Rex files.



## 3.5 Linux Installation

More to come.

## 3.6 Solaris Installation

The following describes how to copy and install the diskettes or CD-ROM on your Solaris 2.5 system.

### 3.6.1 Install T.Rex Firewall Software on Solaris

#### (S1a) For CD\_ROM

Login to the system as root. Insert the CD-ROM. The vold daemon automatically mounts the CD-ROM. To determine the mount point of the CD-ROM issue the mount command. The mount point will look something like: /cdrom/unamed\_cdrom.

CD to the mounted directory.

```
# cd /cdrom
#cd sparc/T.Rex
```

Read README.txt, license.txt, and RELNOTES.txt.  
When ready to install enter the following command.

```
#./install_T.Rex
```

Unmount the CD-ROM and go to step 3.4.2.

If the volume manager does not appear to be running issue the following command.

```
# ps -e | grep vold
```

A response like the following indicates the volume manager is running:

```
207 ? 0:01 vold
```

If it is not running then mount the CD-ROM as follows:

```
#mount -r -F hsfs /dev/dsk/c0t5d0s2 /cdrom
#cd /cdrom
#cd sparc/T.Rex
```

Read README.txt, license.txt, and RELNOTES.txt.  
When ready to install enter the following command.

```
#./install_T.Rex
```

Unmount the CD-ROM and go to step 3.4.2.

```
# eject cdrom
```

#### (S1b) For floppy disk

```
# cd /usr
# mkdir T.Rex
# cd T.Rex
```

Insert T.Rex diskette(s)

If the volume manager is running:

```
# volcheck
# tar xvf /vol/dev/aliases/floppy0
# eject
```

If the volume manager is not running:

```
# tar xvf /dev/fd0 ( or your device name if different)
```

Remove floppy diskette or CD-ROM.

Read the following files: README.txt, license.txt, and RELNOTES.txt.

When you are ready to install type in the following command.

```
# ./install_T.Rex
```

The install\_T.Rex process creates a log of all changes made to the system. You can review these changes in the /usr/T.Rex/install\_T.Rex.log file.

## 3.6.2 Configure Solaris system

### (S2) Configure LAN adapters

Create /etc/hostname.<interface> with hostname or IP address for the primary LAN adapter.

### (S3) Define fully qualified host name to system

Edit the following files to define the fully qualified host name (host.domain):  
/etc/hosts, /etc/nodename, /etc/hostname.\*,  
/etc/net/ticlts/hosts, /etc/net/ticots/hosts, and /etc/net/ticotsard/hosts.

### (S4) Define default route

Edit /etc/defaultrouter to define the IP address of the external gateway (router) to send packets.

### (S5) Define additional static routes

Edit /etc/init.d/inetsvc to add other static routes. Use the following format:

```
/usr/sbin/route add -net destination lan_adapter metric
```

- (S6)** Define any required alias IP addresses for the LAN adapters. This allows a single LAN adapter to respond to multiple IP addresses. The Genproxy Administration Chapter shows how this facility can be used.

Edit the `/etc/init.d/rootusr` file to add the necessary `ifconfig` commands. The following example shows the addition of two aliases to the `le0` LAN adapter.

```
ifconfig le0:1 204.71.109.32 netmask 255.255.255.0 up
ifconfig le0:2 204.71.109.33 netmask 255.255.255.0 up
```

To list all:

```
ifconfig -a
```

## 3.7 Common T.Rex configuration

This section describes the common installation steps for all UNIX platforms supported by T.Rex.

### (C1) Edit `/etc/hosts`

Edit the `/etc/hosts` file so that the host name and domain name matches the value specified to the DNS. If you have one or more mail servers then specify their host names and IP addresses in this file.

### (C2) Edit `/etc/resolv.conf`

Edit the `/etc/resolv.conf` file so that the domain names matches the value specified in the previous section. The T.Rex firewall runs a caching only DNS that is referenced using the local IP address 127.0.0.1. The T.Rex proxies will use this address to access the caching only DNS to resolve host names on the Internet. This is the preferred way to run T.Rex, and will offer the highest level of performance for name resolution.

It is also possible to code the IP address of the internal DNS, thus allowing applications running on the firewall to resolve internal names. If this option is chosen then the internal DNS will have to use a `forwarders` command coded in the `/etc/named.boot` file to point at the caching only DNS on the firewall. If this is not done then the proxies on the firewall will be unable to resolve external hosts. The advantage of this configuration is it allows the proxies to resolve all internal host names. The disadvantage is external host name resolution must pass through the internal DNS and be forwarded to the firewalls DNS. This doubles the DNS workload.

**Warning: Never code a `forwarders` command on the firewalls DNS** that forwards DNS requests to and internal DNS.. If you do this then an external user can resolve internal host names.

### (C3) Edit `/etc/firewall/resolv.inside.conf`

Edit the `/etc/firewall/resolv.inside.conf` file so that the domain names matches the value specified in the `/etc/resolv.conf` file. Specify the IP address(es) of the internal DNS. This file is used by the T.Rex proxies to resolve internal host names.

### (C4) Edit `/etc/named.local`

Edit the `/etc/named.local` file so that the host name and domain name matches the value specified in the preceding section.

### (C5) Specify Secure Networks

Edit the **`/etc/firewall/securenets` file** to define the protected ("secure") networks to the T.Rex proxy servers. You can define one or more networks as secure. Each network is defined on its own line in the standard IP dot-decimal format. Simply code the IP address of the networks you want to treat as secure. A host on a secure network can be given transparent access to the unsecured networks by the `ftproxy` and `tnproxy`. If a host is not included in the description of the secure networks then it is treated as a unsecured host and strong user authentication will be required by the `ftproxy` and `tnproxy`.

**Rules** for coding addresses.

When specifying IP addresses do not pad the number with leading zeros. For example, use "192.168.0." and not "192.168.00." If you add leading zeros then T.Rex will be not able to match the entry and the network will be treated as an external network.

**Truncated addressees:** You can code truncated addresses provided they end in a '.' or a '\*'. If a truncated address ends with a period then the comparison will be made using the character string up to and including the period. If the truncated string ends with an asterisk '\*' then the comparison will be made using the characters preceding the asterisk.

The following example shows how to code the IP addresses.

```
10.2.  
10.3*  
172.23.  
172.24.131.  
172.24.132.  
172.24.133.  
172.24.220.72
```

The first line "10.2." will treat all addresses starting with 10.2. as secure. Thus, all addresses of the form 10.2.x.y will be treated as secure. The address 10.21.x.y will be considered un-secure.

The second line "10.3\*" shows the use of an asterisk as a wildcard. In this example, the following networks will be secure: 10.3.x.y, 10.30.x.y, 10.31.x.y thru 10.39.x.y.

**Error:** If you code "10.3" this will be considered an error, since it is a truncated address (less than three periods) and does not end in a period or an asterisk.

If you code a line as "10.3.4.1" then only one address 10.3.4.1 will be considered secure. Addresses from 10.2.4.10 thru 10.3.4.19 will not pass this test. However, if you want to include all address that start with "10.3.4.1", including 10.3.4.1, 10.3.4.10 thru 10.3.4.19 then code "10.3.4.1\*".

The **Internet Assigned Number Authority (IANA)** has reserved the following three blocks of the IP address space for private networks. These IP addresses can be used behind the T.Rex firewall if you need additional IP addresses.

## IP Addresses Reserved for private networks

```
10.0.0.0 - 10.255.255.255  
172.16.0.0 - 172.31.255.255  
192.168.0.0 - 192.168.255.255
```

You may have noticed that we used reserved IP addresses in the previous example to avoid the use of IP addresses that might have been assigned to some organization.

## (C6) Specify Secure PORTS

Edit the **/etc/firewall/secureports file** to define the protected ports to the T.Rex proxy servers. Protected ports are the IP addresses of the communications adapters that are attached to the protected networks. You can define one or more ports as secure. Each port is defined on its own line in the standard IP dot-decimal format. Simply code the complete IP address of the network adapter you want to treat as secure. This information is used by the T.Rex proxy servers to check for IP address spoofing. If a connection request is made from a host using a secure IP address the local IP address has to be one of the secure port numbers. Otherwise a**Security Alert** will be issued and the connection will be broken.

The following example shows how to code the IP addresses.

192.168.24.1

With a dual-homed host one port will be secure and the other will be unsecured. With a multi-homed host ( 3 or more one or more ports can be secure.

## **(C7) Edit /etc/firewall/tnproxy.conf**

Edit the /etc/firewall/tnproxy.conf file to specify access controls that match your organizations security policies. Refer to the tnproxy chapter for details.

## **(C8) Edit /etc/firewall/ftproxy.conf**

Edit the /etc/firewall/ftproxy.conf file to specify access controls that match your organizations security policies. Refer to the ftproxy chapter for details.

## **(C9) Edit /etc/firewall/genproxy.conf**

Edit the /etc/firewall/genproxy.conf file to specify access controls that match your organizations security policies. Refer to the genproxy chapter for details.

## **(C10) Edit /etc/firewall/webgate.conf**

Edit the /etc/firewall/webgate.conf file to specify access controls that match your organizations security policies. Refer to the webgate chapter for details.

## **(C11) Edit /etc/inetd.conf**

Edit the /etc/inetd.conf file to activate genproxy and webgate on behalf of services you want to use. Refer to the genproxy chapter for details.

## **(C12) Edit /etc/sockd.conf**

Edit the /etc/sockd.conf file to specify access controls that match your organizations security

policies. Refer to the socks chapter for details.

### **(C13) Edit /etc/sockd.route**

Edit the /etc/sockd.route file to specify socks routing. Refer to the socks chapter for details.

### **(C14) Edit /etc/firewall/aliases**

Edit the /etc/firewall/aliases file to add mail aliases used to translate e-mail addresses for delivery. . Refer to Chapter 8 Sendmail Wrapper Administration for details.

### **(C15) Build the T.Rex Aliases Data Bases**

Run the adam utility to create the Aliases Data Bases.

```
# /usr/local/etc/adam -b
```

Refer to Chapter 8 Sendmail Wrapper Administration for details.

### **(C16) Edit sendmail aliases file**

Edit the aliases file used by sendmail to handle system generated e-mail. On AIX this is the /etc/aliases file. On HP-UX and Solaris the name of the file is /etc/mail/aliases. Refer to the Mail Setup chapter for details.

### **(C17) Build the aliases Data Base for sendmail**

Run the sendmail or newaliases program to create the Aliases Data Base used by sendmail for batch delivery.. On AIX the file is /etc/aliases on HP-UX and Solaris is it /etc/mail/aliases.

```
# newaliases          (for AIX)
```

or

```
# sendmail -bi
```

Refer to the Mail setup chapter for details.

### **(C18) Edit /etc/motd**

Edit /etc/motd to provide your company/organization name for the login announcement.

### **(C19) Add gateway users for telnet and ftp proxies.**

If your organization wants to control telnet or ftp access to the external network or you want to allow external users to enter the protected network then you have to add user information to the gwuser



data base. See Chapter 5 T.Rex User Administration for details.

## **(C20) Reboot the system.**

You are now ready to reboot the system. Simply issue the following command.

**For AIX:**

```
# shutdown -r
```

or

```
# sync  
# reboot
```

**For HP-UX**

```
# shutdown -r -y now
```

**For Linux**

```
#
```

**For Solaris:**

```
# shutdown -y -i6
```

or

```
# sync  
# reboot
```

## 3.8 Setup Dual Domain Name Servers

### 3.8.1 Dual DNS Overview

T.Rex supports separate external and internal Domain Name Servers (DNS). Users on internal hosts can resolve host names into IP address for both the internal and external systems. Users of external systems cannot access or even see the internal DNS. The internal network is not IP addressable and is completely hidden from the outside. Yet, e-mail, telnet and other authorized applications can access the internal systems.

T.Rex runs a caching-only name server used to resolve external host names. Internal hosts can use the caching-only DNS to resolve external names. Special client code is provided with T.Rex that uses a file other than `/etc/resolv.conf` to point at the internal DNS. This allows programs running on the firewall to resolve both internal and external names.

The `ftproxy`, `genproxy`, `tnproxy` and `webgate` are designed to resolve host names using both the external Domain Name Server and the Internal Domain Name Server. The external names are found using the standard `/etc/resolv.conf` to point at the external DNS. The `/etc/firewall/resolv.inside.conf` file is used to point at the internal DNS.

### 3.6.2 Set up Internal DNS

The internal DNS is pointed to by the file `/etc/firewall/resolv.inside.conf`. This file looks just like your `/etc/resolv.conf` file, but points to an internal name server rather than an external name server. Edit the file to make the following two changes:

- (1) change the sample domain name ***lsli.com*** to match your own, and
- (2) change the sample IP addresses ***192.168.24.12*** to that of your internal DNS.

```
domain lsli.com
nameserver 192.168.24.12
```

### 3.6.3 Set up External DNS

T.Rex is configured with a caching-only Domain Name Server that can resolve the names of **external** hosts to IP addresses. The `/etc/resolv.conf` file contains the IP address of the DNS to be used for name resolution. T.Rex provides a sample `/etc/resolv.conf` file to point at the caching-only DNS that runs on the firewall.

The proxy servers (`ftproxy`, `genproxy`, `tnproxy`, ...) provided with T.Rex can resolve external and internal host names since they query both the external and internal Domain Name Servers. A separate file `/etc/firewall/resolv.internal.conf` is used to point at the internal DNS.

The automated install procedure automatically creates the `/etc/resolv.conf` file and the only item that needs to be modified is the domain name. You can edit this file with your favorite editor or use the GUI based utilities provided with the OS.

**AIX:** On AIX use IBM's System Management Interface Tool (SMIT) or the UNIX `namerslv` command to make changes. If the T.Rex firewall has direct IP addressability to the Internet the `/etc/resolv.conf` does not have to be changed.

AIX allows you to change the default order for name resolution by creating the net services configuration file `/etc/netsvc.conf`. Simply add the following line to the file to specify DNS (bind) is used before the `/etc/hosts` file.

```
hosts=bind,local
```

**HP-UX:** Edit the `/etc/resolv.conf` file. (The HP-UX SAM utility does not have this function)

**Solaris:** Edit the `/etc/resolv.conf` file. Solaris also uses the `/etc/nsswitch.conf` file to specify how BIND, NIS+, and the `/etc/hosts` files are used. T.Rex provides a sample `/etc/nsswitch.conf` file that causes DNS to be used first for name resolution followed by the `/etc/hosts` file. The sample file has the following line:

```
hosts: dns [NOTFOUND=continue UNAVAIL=continue] files
```

### 3.6.4 Setup a Caching-only Name Server

A *Caching-only Name Server* should be setup on the T.Rex firewall for two reasons. First, it will reduce the time to establish a telnet and ftp connection. Second, it is required if you want to run the SOCKS daemon on the T.Rex firewall and allow SOCKS clients on the protected network to resolve Internet hosts names into IP addresses.

A Caching-only Server is very simple to configure. Only three files need to be defined for the caching-only DNS: `/etc/named.boot`, `/etc/named.ca` and `/etc/named.local`. These files are automatically created by the auto install process.

#### **`/etc/named.boot` file**

The `/etc/named.boot` file tells named to maintain a cache of name server responses. The cache statement in the example, tells named to initialize the cache using the contents of the file `/etc/named.ca`. A listing of the `/etc/named.boot` file can be found in Appendix B.

#### **`/etc/named.ca` file**

The automated install process creates the `/etc/named.ca` file. This file can be used as is. A current copy of this file is made available by the InterNIC registration services under anonymous FTP at host **ftp.rs.internic.net** as file **/domain/named.root**. The example was current when the manual was printed. The InterNIC registration services periodically updates their source file with new server names and addresses so you will need to download their file on a periodic basis.

A listing of the `/etc/named.ca` file can be found in Appendix B.

## **/etc/named.local file**

The /etc/named.local file is created by the auto install process. Edit the sample file by changing the sample host name "**gw.lsl.com**" to match the host name and domain name of your firewall. A listing of the /etc/named.local file can be found in Appendix B.

## **3.7 Setup Mail Server**

If you want to use the T.Rex Firewall system as the mail gateway then go the chapter on sendmail.

At this point the system is ready to go.

## **3.8 Adding Administrative Users**

Administrators have three methods available for administering the firewall. They can login at the firewall console, login remotely using ptelnet or the Hoplite client. Use of ptelnet or Hoplite requires the administrator to be defined in the gwuser data base as well as to the OS.

To add an administrative user the root user must use the gwuser command to add the administrator ID to the /etc/firewall/gwuser data base. The administrator user must also be added to the system using the system administration utility supplied by the OS vendor.

AIX: Use **SMIT** to add the user to the UNIX system.

HP-UX: Use the **sam** utility to add the user to the UNIX system.

Solaris: Use **admintool** to add the user to the UNIX system.

Both of these must be done to allow the administrator to login to the system.

### **3.8.1 Defining admin users with gwuser**

Administrators are added to the T.Rex system using either Hoplite or the gwuser command. The Hoplite client provides a fill in the blanks GUI. The ptelnet client provides the administrator with the an encrypted telnet session that provides access to all the commands defined in this manual. The following chapter documents the command line interface for gwuser and gwgroup. Hoplite provides online documentation for its use.

### **3.8.2 Defining admin users with SMIT (AIX)**

Users can be added to the T.Rex system using the IBM SMIT utility. Users are added just like any other AIX system. During the installation process the mkuser.default file was updated so that the users defined by the SMIT utility will have their initial program set to /usr/local/etc/gwsh2. Do not

override this option otherwise a cracker could break in as a systems administrator. When specifying the Initial Program fill in the complete path name of the gateway shell provided with T.Rex. Each userid on the system must have this as their shell in */etc/passwd*, or else not have a real login shell at all. This is required since telnetd is enabled and anyone can get a login prompt from either side of the gateway. If you give someone a regular shell, then someone with that password can easily get in from the outside.

**Initial Program            /usr/local/etc/gwsh2**

When all the administrative users have been added to the system you can display the *passwd* file and check that all the user IDs have gwsh2 specified.

**# cat /etc/passwd**

```
root:!:0:0:!/bin/ksh
daemon:!:1:1:!/etc:
bin:!:2:2:!/bin:
sys:!:3:3:!/usr/sys:
adm:!:4:4:!/usr/adm:
uucp:!:5:5:!/usr/lib/uucp:
guest:!:100:100:!/usr/guest:
nobody:!:4294967294:4294967294:!/
lpd:!:104:9:!/
jim:!:6:0:R. James Livermore:/u/jim:/usr/local/etc/gwsh2
charles:!:200:1:Charles M. Livermore:/u/charles:/usr/local/etc/gwsh2
ellana:!:201:1:Ellana T. Livermore:/u/ellana:/usr/local/etc/gwsh2
jay:!:202:1:Jay P. Lyall:/u/jack:/usr/local/etc/gwsh2
carlton:!:7:0:Carlton T. Doorman:/u/carlton:/usr/local/etc/gwsh2
nick:!:203:1:Nick Trio:/u/nick:/usr/local/etc/gwsh2
```

Note:

The root id uses the */bin/ksh*. This is OK since the system has been configured so that root can only login at the system console. A remote login to root is not possible, from either the protected or unprotected networks. If you specified the */usr/local/etc/gwsh2* for the root id then the user would receive the 8 digit challenge as if they were trying to login from an unprotected system. This would require the root user to have access to a CRYPTOCARD or a SecureNet Key card to login as root from the console. This is optional. Beware, if you do this and there is a problem with your card you will have to boot from a CD-ROM or tape or diskette to fix it.

The other IDs from daemon through lpd do not allow logins, so they do not have an initial program specified. These IDs have logins restricted in the */etc/security/passwd* file which can only be read or modified by the root id. Do not change these login IDs.

The login IDs from jim through nick all have the */usr/local/etc/gwsh2* specified.

Notice that the encrypted passwords are not stored in the */etc/passwd* file. The character "!" is stored instead. The encrypted passwords are stored in the */etc/security/passwd* file which can only

be read by the root id. This provides an extra level of security since all users can read the `/etc/passwd` file, but not the `/etc/security/passwd` file. Thus an ordinary user can't copy the encrypted passwords to another system where a password cracking program can be applied.

### 3.8.3 Gateway Shell

The Gateway Shell Version 2 - `gwsh2`

The `gwsh2` program is installed in the `/usr/local/etc` directory by the auto install program. Do not put the `gwsh2` into another directory, otherwise entries in the files `/etc/security/login.cfg` and `/usr/lib/security/mkuser.default` will be incorrect. Each userid on the system except root must have this as their shell in `/etc/passwd`, or else not have a real login shell at all. This is required since `telnetd` is enabled and anyone can get a login prompt from either side of the gateway. If you give someone a regular shell, then someone with that password can easily get in from the outside. The program `gwsh2` also needs to be `setuid` to root (`-r-sr-sr-x`). Inside the `gwsh`, there is a `setuid(getuid())` call so that the user **does not get root privileges when his real shell is run**.

**Note:** For security reasons the **ONLY** way to get a real shell and do administration is to login as root at the system console.

## 3.9 Changes made during installation

The automated installation process makes numerous system changes some of which are mentioned below.

### 3.9.1 Adding Proxy User ID

The proxies provided with T.Rex (`tnproxy`, `ftproxy`, `smwrap`, etc) run as unprivileged users in a chrooted directory. This enhances the security and integrity of T.Rex since the proxies are unable to access or modify programs or configuration data on the firewall system. This feature requires the definition of a user ID for the proxies. The automated installation process adds the user ID `hermes` and its directory to the system. The automated install process also adds the following directories required by `syslog` in a chrooted environment.

<code>/home/hermes/dev/null</code>	(AIX)
<code>/home/hermes/dev/ftp</code>	(Solaris)

You may decide to use another user ID for your proxies or decide to use separate user IDs for each proxy. If you decide to use other user IDs then you should use the system utility provided with the OS to make the changes. You will also need to change the user ID in the proxy configuration files.

### 3.9.2 Sequestered E-mail Directory

If the `smwrap` or `smwrapd` programs detect suspicious e-mail they will sequester the file from the valid mail and issue an alert. The `/etc/firewall/smwrap.conf` file specifies the name of the directory to store sequestered mail (see chapter 8.2). This directory must be a sub directory of the proxy

user directory and be writeable by the smwrap user ID (hermes). If you decide to run the smwrap and smwrapd daemons using an ID other than hermes then you will have to create the sequestered directory under the directory used by the new user ID. After adding the directory, change the owner of the directory to be that of the proxy user ID. You can use names other than hermes and sequestered. However, if you do so then you must make the appropriate changes in the /etc/firewall/smwrap.conf file as described in chapter 8.

```
# mkdir -p /home/hermes/sequestered
# chown hermes /home/hermes/sequestered
```

### 3.9.3 Setup cron to periodically run sendmail

The smwrapd periodically checks for mail spooled by the smwrap program. When smwrapd finds mail to deliver it passes the mail to sendmail for final delivery. If the receiving host is unable to receive mail when sendmail is called for delivery, sendmail places the mail in its own queue. The queued mail will sit there forever unless sendmail is periodically activated by cron to process the queued mail.

T.Rex provides a sample cron table that periodically executes sendmail as a batch program to deliver any queued mail.

### 3.9.4 System Logs

The auto install process automatically configures the syslog to record all proxy activity on the firewall system. However, you may want to alter the syslog to suit your own requirements.

#### **/etc/syslog.conf file**

The /etc/syslog.conf controls which messages are logged and where they are logged. Each line of the file specifies the type of message to log and the destination where the message will be logged. The type of message consists of two parts separated by a period. The first part specifies the facility that generated the message. The second part specifies priority of the message.

### Type of message

The following facilities can be specified in the syslog configuration file: kern, user, mail, daemon, auth, syslog, lpr, news, uucp and \* for all messages. The priorities which can be specified are (from high to low): emerg/panic, alert, crit, err, warn, notice, info, debug. For more information see the IBM InfoExplorer which contains an on-line description of the commands that can be used in the syslog configuration file.

### Destination

The destination of the message can be a specified file on a system, or a user id on a system. To send the message to another system follow the file name or the user id by the character string @hostname.

The following example shows the sample syslog.conf file for AIX which will log the following:

```
!      all error messages will be written to the file /var/adm/syslog,
!      all authorization messages regardless of priority will be written to the file /var/adm/syslog,
```

! all debug messages will be written to /var/adm/syslog.

The sample syslog.conf file for HP-UX will use /var/adm/syslog/syslog.log. The sample syslog.conf file for Solaris uses /var//adm/messages.

## Edit the /etc/syslog.conf file

This example will log all types of messages with a priority of debug or higher.

```
# file: /etc/syslog.conf          system: gw
# function: specify messages to logged by syslogd.
# created: by rjl@lsli          3/15/94
#
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995-2000
# All Rights Reserved
# Licensed Materials are Property of FAS
#
auth.info /var/adm/syslog
*.notice /var/adm/syslog
*.alert /dev/console
*.emerg *
```

If you want to log all system messages on another internal host then simply add the following line to /etc/syslog.conf, replacing *hostname* with the name of your protected host.

```
*.debug @hostname
```

## The syslog file

The auto install process automatically creates the syslog file. However, if the log is removed you will have to recreate the file using the touch command. Otherwise nothing will be logged. The standard location for the syslog is shown for each supported system.

System	Syslog Location
AIX	/var/adm/syslog
HP-UX	/var/adm/syslog/syslog.log
Solaris	/var/adm/messages

Chapter 6 shows how to read and interpret the syslog.

### /etc/inetd.conf

The file */etc/inetd.conf* contains the Internet server configuration data base. Services can be deleted by inserting a # at the beginning of the line. T.Rex replaces the standard /etc/inetd.conf file with a version that disables dangerous programs, and enables secure proxies supplied with T.Rex. The new version of inetd.conf is listed in Appendix B.



## **/etc/services**

The /etc/services file is updated by appending the following entries:

```
http      80/tcp
ssl       442/tcp
socks     1080/tcp
radmin    9010/tcp
generic   9011/tcp
```

### 3.10 OS Maintenance Requirements

**Warning: T.Rex should be installed on a fresh copy of the OS to ensure that the underlying OS has not been compromised prior to the T.Rex installation.**

During the last three years there have been many CERT advisories issued. Fortunately most of the advisories do not apply to systems running T.Rex, as T.Rex has disabled or replaced the functions discussed in the CERT advisory. However, there have been several CERT Advisories that require the installation of maintenance on either AIX, HP-UX, Linux, or Solaris. Therefore, you should check to make certain that all required maintenance has been performed on the OS prior to the T.Rex installation. The following section lists the maintenance that should be applied to the applicable OS. This list is current as of the date of publication. FAS will notify you should any additional OS maintenance be required.

### 3.10.1 AIX Maintenance

#### Problem: **Ping of Death**

In 1996 the CERT Coordination Center received reports of denial-of-service attacks using large ICMP datagrams. Systems receiving oversized ICMP datagrams may crash, freeze or reboot, resulting in a denial of service. Hackers have used a tool that sends oversized ping packets that caused buffer overruns in the kernel. This is known as the ping-of-death, as it causes most systems to die.

The following versions of AIX should have their respective APARs installed to protect against the ping-of-death.

AIX 4.1.X	IX59453
AIX 4.2.X	IX61858

To determine if you have this APAR on your system, run the following command.

<code>instfix -ik IX59453</code>	(for AIX 4.1)
<code>instfix -ik IX61858</code>	(for AIX 4.2)

You can also run the following command

```
lspp -h bos.net.tcp.client
```

This should produce the following results for AIX 4.1 and 4.2 respectively: 4.1.4.16 or 4.2.0.6.

#### Problem: **BIND Vulnerabilities**

Three vulnerabilities have been identified in BIND.

- (1) Improperly or maliciously formatted inverse query on a TCP stream.
- (2) Improperly or maliciously formatted DNS message.
- (3) Self-referential CNAMEs.

#### Damage Assessment:

- (1) If exploited, a remote user may cause a buffer overrun or gain root access.
- (2) & (3) These two vulnerabilities could lead to Denial-of-Service.

#### Solution:

Install the version of named 4.9.7 that ships with T.Rex in the /usr/sbin directory or install one of the applicable patches from IBM.

Apply patches or work-arounds as listed below.

The version of bind shipped with AIX is vulnerable and the following APARs will be available soon:

AIX 4.1.x:	IX76958 (fix for Topic 1 only)
AIX 4.2.x:	IX76959 (fix for Topic 1 only)

AIX 4.3.x: IX76960 (fix for Topic 1 and 3 only)  
AIX 4.3.x: IX76962 (fix for Topic 1, 2, and 3. This is bind 8.1.2.)

Until the official fixes are available, a temporary patch can be found at:

<ftp://aix.software.ibm.com/aix/efixes/security>

File	sum	md5
named.415.tar.Z	64980 157	0e795380b84bf29385d2d946d10406cb
named.421.tar.Z	44963 157	15a9a006abf4a9d0a0d3210f16d619e5
named4.430.tar.Z	48236 115	8377b14f74e207707154a9677906f20a
named8.430.tar.Z	51175 160	e2db14b7055a7424078456bfbfd9bf2d

### 3.10.2 Solaris Maintenance

**Problem: Ping of Death**

In 1996 the CERT Coordination Center received reports of denial-of-service attacks using large ICMP datagrams. Systems receiving oversized ICMP datagrams may crash, freeze or reboot, resulting in a denial of service. Hackers have used a tool that sends oversized ping packets that caused buffer overruns in the kernel. This is known as the ping-of-death, as it causes most systems to die.

Sun Microsystems has provided the following list of patches in response to this advisory:

103630-09 5.5.1  
103631-09 5.5.1\_x86  
103169-12 5.5  
103170-12 5.5\_x86  
101945-51 5.4  
101946-45 5.4\_x86

**Problem: BIND Vulnerabilities**

Three vulnerabilities have been identified in BIND.

- (1) Improperly or maliciously formatted inverse query on a TCP stream.
- (2) Improperly or maliciously formatted DNS message.
- (3) Self-referential CNAMEs.

**Damage Assessment:**

- (1) If exploited, a remote user may cause a buffer overrun or gain root access.
- (2) & (3) These two vulnerabilities could lead to Denial-of-Service.

**Solution:**

Install the version of named 4.9.7 that ships with T.Rex in the /usr/sbin directory or install one of the applicable patches from Sun Microsystems.

Sun Microsystems will produce patches for Solaris 2.5.1 and Solaris 2.6.

### 3.10.3 HP-UX Maintenance

Hewlett Packard's HP-UX patches/Security Bulletins/Security patches are available via email and/or WWW (via the browser of your choice) on HP's Electronic Support Center (ESC).

To subscribe to automatically receive future NEW HP Security Bulletins from the HP ESC Digest service via electronic mail, do the following:

1) From your Web browser, access the URL:

<http://us-support.external.hp.com> (US,Canada,Asia-Pacific, and Latin-America)

<http://europe-support.external.hp.com> (Europe)

Login with your user ID and password, or register for one (remember to save the User ID assigned to you, and your password). Once you are on the Main Menu, Click on the Technical Knowledge Database, and it will connect to a HP Search Technical Knowledge DB page. Near the bottom is a hyperlink to our Security Bulletin archive. Once in the archive there is another link to our current security patch matrix. Updated daily, this matrix is categorized by platform/OS release, and by bulletin topic.

To subscribe to receive future Security Bulletins by email, look for the subscription section on the Technical Knowledge Database page.

#### **Before installing any patches to HP-UX please contact FAS.**

Problem: **Ping of Death**

In 1996 the CERT Coordination Center received reports of denial-of-service attacks using large ICMP datagrams. Systems receiving oversized ICMP datagrams may crash, freeze or reboot, resulting in a denial of service. Hackers have used a tool that sends oversized ping packets that caused buffer overruns in the kernel. This is known as the ping-of-death, as it causes most systems to die.

For HP9000 Series 700 and 800 systems, apply the appropriate patch. see Hewlett-Packard Security Bulletin #000040 (HPSBUX9610-040) for further details. The bulletin is available from the HP Support Line and <ftp://ftp.cert.org/pub/vendors/hp/>.

Problem: **Bind**

Three vulnerabilities have been identified in BIND.

- (1) Improperly or maliciously formatted inverse query on a TCP stream.
- (2) Improperly or maliciously formatted DNS message.
- (3) Self-referential CNAMEs.

Damage Assessment:

- (1) If exploited, a remote user may cause a buffer overrun or gain root access.

(2) & (3) These two vulnerabilities could lead to Denial-of-Service.

**Solution:**

Install the version of named 4.9.7 that ships with T.Rex in the /usr/sbin directory or install one of the applicable patches from the Hewlett-Packard Company.

Apply patches or work-arounds as listed below.

HP is Vulnerable. Patches in process. Watch for the release of the associated HP Security Bulletin.

# Chapter 4. T.Rex User Administration

## 4.1 Managing User Access to T.Rex

T.Rex provides multiple options for managing user access to unprotected networks. You can select the method that best suits your organizations security policies. If your organization wants to provide unrestricted transparent access from protected networks to unprotected networks T.Rex can be configured this way. If the organization wants to control what functions are available to individual users this can also be done.

Unrestricted transparent access from the protected network to the unprotected network requires the least amount of administration. In this case no users need to be defined to T.Rex. You will not be required to use the `gwuser` utility described in this chapter.

If your organization wants to control and monitor access to the unprotected network on a user or group basis then the users need to be defined to the T.Rex firewall using the `gwuser` utility described below. The user record combined with the proxy server configuration rules allow the systems administrator to provide precise controls over user access. User access can be controlled by function (protected or unprotected telnet or ftp) by source and destination addresses and time of day and day of week. The user record controls what functions a user has access to and the method of user authentication. This chapter describes how to use the `gwuser` utility to manage user records. The proxy servers provide control by source and destination addresses as well as time. Proxy configuration controls are described in the appropriate proxy chapters.

If you have a requirement to allow users to enter your protected network from the unprotected network then you will have to define all these users to T.Rex. In many cases this is a small subset of the total user base. External users need to be defined since they will be forced to undergo strong user authentication during the connection process. This requires the systems administrator to define the user, the users privileges and the method to be used for authentication. Users can be added to the T.Rex system using the **gwuser** program supplied with T.Rex.

## 4.2 Groups and Users

All users defined to T.Rex must be associated with a previously defined group. Groups are created by the root user with the default values for the group. When an user record is defined it will inherit all the values assigned to the group. By changing a single group record the administrator can alter the permissions of an entire group of users. This allows a significant reduction in the time required for user administration.

Two functions `gwgroup` and `gwuser` are provided for management of groups and users.

### **gwgroup**

The `gwgroup` program allows the system administrators to add, delete or list groups of T.Rex users. T.Rex supports multiple systems administrators. Each group of users can be assigned a group administrator. These administrators log on to the firewall under the control of a restricted shell. The only command they need to use is the `gwuser` command. The only users allowed to use the `gwuser` command are those who are assigned by the root user to be group administrators. The group administrator can only add, delete or list users belonging to groups assigned to that administrator.



These administrators are not granted root privileges.

The same command is used for adding, deleting and listing users. For simplicity we show the command syntax for each function (add, delete and list) separately.

## **gwuser**

The gwuser program allows the system administrators to add, delete or list T.Rex users. T.Rex supports multiple systems administrators. Each systems administrator can be assigned one or more user groups to administer. These administrators log on to the firewall under the control of a restricted shell. The only command they need to use is the gwuser command. The only users allowed to use the gwuser command are those who are assigned by the root user to be group administrators. The group administrator can only add, delete or list users belonging to groups assigned to that administrator. These administrators are not granted root privileges.

The same command is used for adding, deleting and listing users. For simplicity we show the command syntax for each function (add, delete and list) separately.

## **4.3 Online Help for group Administration**

The gwgroup program contains on-line help. Simply type in the gwgroup command without any parameters or followed by a "?" or a "-h". If you want detailed help on a specific option type the command with the option followed by either a "?" or a "-h".

The following example shows how list the primary options. You will receive high level help by typing in gwgroup without any options or by following gwgroup with either a '?' or a '-h'.

```
# /usr/local/etc/gwuser -h
```

gwgroup usage:

```
help:      gwgroup [ -a | -m | -d | -l ] [ ? | -h ]
add:       gwgroup -a group [ options ]
modify:    gwgroup -m group options
delete:    gwgroup -d group
list:      gwgroup -l [ -a ] [ groups ]
```

If you want help with listing groups type in the following.

```
# /usr/local/etc/gwuser -l ?
```

gwgroup list group usage:

```
gwgroup -l [ -a ] [ groups ]
    -l List gwuser database group records.
    -a List all information.
```

groups Group name(s); if not specified, list all groups.  
If wild card specified, enclose name in quotes.

## 4.4 Adding a Group

**gwgroupp**     **-a group** [ -n "*name*" ] [ -s CRYPTO | SNK | SDI ]  
                 [-f [UNPRO\_FTP] [PRO\_FTP | PRO\_FTPSA]  
                 [ GET\_FTP] [ PUT\_FTP] { DEL\_FTP} ]  
                 [-t [UNPRO\_TN] [PRO\_TN | PRO\_TNSA ] [ADM\_TN]] ]  
                 [ -times ALL | "*days, hours*" ]

**-a group**        This is used to add a group to the gwuser data base. The group name is an alphanumeric character string from 1 to 16 characters in length. Valid characters are a-z and 0-9. Special characters and blank spaces are not allowed. The -a argument is mutually exclusive with the -d and -l arguments.

**-n "*name*"**        This is used to add a descriptive name for the group. This description can be from 1 to 32 characters in length.

**-s *token***        The token specifies the type of security token used for strong user authentication. Valid values are CRYPTO, SNK and SDI (AIX only). If this parameter is not coded the default value is NONE. If the security token is NONE then external access will be denied.

**-f *FTP\_access***    FTP access is used to grant the use of FTP for members of the group. The available options are:

**UNPRO\_FTP**        Group members on unprotected network can use FTP after passing strong user authentication.

**PRO\_FTP**          Group members on protected networks are allowed to use FTP. This option is mutually exclusive with PRO\_FTPSA.

**PRO\_FTPSA**        Group members on protected networks are allowed to use FTP after passing string user authentication. This option is mutually exclusive with PRO\_FTP.

**GET\_FTP**          This allows group members on protected networks to GET files and group members on unprotected networks to PUT files. The view point is that of the protected users.

**PUT\_FTP**          This allows group members on protected networks to PUT files and group members on unprotected networks to GET files. The view point is that of the protected users.

**DEL\_FTP**          This allows group members on protected networks to DEL files. The view point is that of the protected users.

The default value is NO\_FTP.

**-t *Telnet\_access***

Telnet access is used to grant the use of the telnet proxy at the individual user level. Five options are available: NO\_TN, PRO\_TN, UNPRO\_TN, PRO\_TNSA and ADM\_TN.

**NO\_TN** The user is not allowed to use tnproxy. This is the default value. NO\_TN is mutually exclusive with the other options.

**PRO\_TN** The user is allowed to use tnproxy from a protected host.

**UNPRO\_TN** The user is allowed to use tnproxy from a unprotected host if they have a security token (CRYPTOCARD, or SNK card).

**PRO\_TNSA** The user is allowed to use tnproxy from a protected host only if they pass strong user authentication.

**ADM\_TN** This allows system administrators to login to the firewall system with strong user authentication. ADM\_TN also requires use of special telnet clients that support encrypted data streams. This is an optional feature of T.Rex that is licensed separately. This feature is currently restricted and is available in limited countries.

ADM\_TN requires the specification of either PRO\_TNSA or UNPRO\_TN or both.

The current encrypted telnet clients designed to work with the Administrator option require the use of a shared private key. This shared private key is generated with the **tnkey** command described in section 5.6

**-times ALL | [days hours] ...**

The times that members of the group are allowed to access the firewall using telnet or ftp. If ALL is specified then the group members have access 168 hours each week. To specify Monday through Friday 8 AM till 5 PM do the following.

**-times m-f 8a-5p**

To allow access Monday through Friday 8 am till 5 PM and on Saturday from 9 am to 5 PM do the following.

**-times m-f 8a-5p sa 9a-5p**

## 4.5 Deleting a Group

**gwgroup -d *group\_name***

**-d *group\_name*** This is used to remove a group from the gwuser data base.. This argument is mutually exclusive with gwgroup -a and gwgroup -m. Before a group can be deleted all users must be removed from the group.

## 4.6 Listing a Group

**gwgroup -l [-a] [ group ]**

- l [group]** This is used to list one or more group entries in the gwuser data base. If there is no group name argument then all entries will be printed. You can use wildcards for group names. For example, if you enter gwgroup -l st\* then all entries starting with the characters st will be listed. This argument is mutually exclusive with gwgroup -a, and gwgroup -d.
- a** If the -a parameter is specified then all fields in the group record will be listed. If -a is not specified then only a summary listing is printed.

## 4.7 Modifying a Group

**gwgroup** **-m group** [ -n "name" ] [ -s CRYPTO | SNK | SDI | NONE ]  
[ -f [NO\_FTP ] [UNPRO\_FTP] [PRO\_FTP | PRO\_FTPSA]  
[ GET\_FTP] [ PUT\_FTP] { DEL\_FTP } ]  
[ -t [ NO\_TN } [UNPRO\_TN] [PRO\_TN | PRO\_TNSA ] [ADM\_TN]] ]  
[ -times ALL | NONE } [ hours ]

- m group** This is used to modify a group in the gwuser data base. The -a argument is mutually exclusive with the -d and -l arguments.
- n "name"** This is used to change the descriptive name for the group. This description can be from 1 to 32 characters in length.
- s token** The token specifies the type of security token used for strong user authentication. Valid values are CRYPTO, SNK, SDI (AIX only) and NONE. If the security token is NONE then external access will be denied.
- f FTP\_access** FTP access is used to grant the use of FTP for members of the group. The available options are:
- |                  |                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NO_FTP</b>    | Group members can not use FTP. This option is mutually exclusive with all other FTP options.                                                                        |
| <b>UNPRO_FTP</b> | Group members on unprotected network can use FTP after passing strong user authentication.                                                                          |
| <b>PRO_FTP</b>   | Group members on protected networks are allowed to use FTP. This option is mutually exclusive with PRO_FTPSA.                                                       |
| <b>PRO_FTPSA</b> | Group members on protected networks are allowed to use FTP after passing string user authentication. This option is mutually exclusive with PRO_FTP.                |
| <b>GET_FTP</b>   | This allows group members on protected networks to GET files and group members on unprotected networks to PUT files. The view point is that of the protected users. |
| <b>PUT_FTP</b>   | This allows group members on protected networks to PUT files and group members on unprotected networks to GET files. The view point is that of the protected users. |

**DEL\_FTP** This allows group members on protected networks to DEL files. The view point is that of the protected users.

The default value is NO\_FTP.

#### **-t Telnet\_access**

Telnet access is used to grant the use of the telnet proxy at the individual user level. Five options are available: NO\_TN, PRO\_TN, UNPRO\_TN, PRO\_TNSA and ADM\_TN.

**NO\_TN** The user is not allowed to use tnproxy. This is the default value. NO\_TN is mutually exclusive with the other options.

**PRO\_TN** The user is allowed to use tnproxy from a protected host.

**UNPRO\_TN** The user is allowed to use tnproxy from a unprotected host if they have a security token (CRYPTOCARD, or SNK card).

**PRO\_TNSA** The user is allowed to use tnproxy from a protected host only if they pass strong user authentication.

**ADM\_TN** This allows system administrators to login to the firewall system with strong user authentication. ADM\_TN also requires use of special telnet clients that support encrypted data streams. This is an optional feature of T.Rex that is licensed separately. This feature is currently restricted and is available in limited countries.

ADM\_TN requires the specification of either PRO\_TNSA or UNPRO\_TN or both.

The current encrypted telnet clients designed to work with the Administrator option require the use of a shared private key. This shared private key is generated with the **tnkey** command described in section 5.6

## **4.8 Adding a User**

```
gwuser -a user [-g group] [-n "actual name"] [-ph "phone number"]
          [-p password] [-s CRYPTO | SNK | SDI ]
          [-k nnn nnn nnn nnn nnn nnn nnn nnn ]
          [-t [ UNPRO_TN ] [ PRO_TN | PRO_TNSA ] [ ADM_TN ]
          [-f [ UNPRO_FTP ] [ PRO_FTP | PRO_FTPSA ]
          [ GET_FTP ] [ PUT_FTP ] [ DEL_FTP ] ]
```

**-a user** This is used to add a user to the T.Rex gateway. The user name is an alphanumeric character string from 1 to 16 characters in length. Valid characters are a-z and 0-9. Special characters and blank spaces are not allowed. The -a argument is mutually exclusive with the -d and -l arguments.

**-g group** The group parameter specifies the group the user belongs to. The group name is an alphanumeric character string from 1 to 16 characters in length. Valid characters are a-z

and 0-9. Special characters and blank spaces are not allowed. If group is not specified the default group is the first group listed for the systems administrator in the /etc/firewall/adm.conf file.

**-n "name"** The users real name. It should be enclosed within double quotes and can have a maximum length of 32 characters.

**-ph "phone #"** The users phone number. It should be enclosed within double quotes and can have a maximum length of 32 characters.

**-p password** The password parameter is used to specify the password for the user. If the tnproxy or ftpoxy are configured to request user authentication for users on protected hosts then this is the password that users must use for authentication.

If a password is specified it must be an alphanumeric string of at least nn bytes and no more than 16 bytes. There must be at least one character and at least one numeric value. Valid characters are a-z and 0-9. Special characters are not allowed.

External users must undergo strong user authentication and possess a security token (aka a smart card). This password is not used for access from an unprotected host.

**-s security\_token**

The security token specifies the type of security token used for strong user authentication. Valid values are CRYPTO, SNK and SDI (AIX only). If this parameter is not coded the **default** value found in the group record is used. If the default value in the group record is NONE external access will be denied.

**-k nnn nnn nnn nnn nnn nnn nnn nnn**

The key parameter specifies the authentication key used for strong user authentication. The key must be specified for users that are assigned security tokens that use a challenge response system. This includes the following security tokens: CRYPTOCARD, SecureNet Key (SNK), and Watchword. The key consists of eight three digit octal numbers separated by a blank. You can use any number between and including 001 and 177 (octal of course). The digits 8 and 9 are not valid in the octal numbering system. They are the same numbers that must be programmed into the security token specified by the -s parameter.

There are a small number of **weak keys** that should be avoided when programming a security token. The gwuser utility automatically checks for weak keys and will issue an error message should one be entered. The following weak keys should be avoided.

```
001 001 001 001 001 001 001 001
037 037 037 037 037 037 037 037
```

Other weak keys are larger than 177 and are automatically rejected.

**-t Telnet\_access**

Telnet access is used to grant the use of the telnet proxy at the individual user level. Five options are available: NO\_TN, PRO\_TN, UNPRO\_TN, PRO\_TNSA and ADM\_TN.

**NO\_TN** The user is not allowed to use tnproxy. This is the default value. NO\_TN is mutually exclusive with the other options.

<b>PRO_TN</b>	The user is allowed to use tnproxy from a protected host.
<b>PRO_TNSA</b>	The user is allowed to use tnproxy from a protected host only if they pass strong user authentication. This is mutually exclusive with PRO_TN.
<b>UNPRO_TN</b>	The user is allowed to use tnproxy from a unprotected host if they have a security token (CRYPTOCARD, or SNK card).
<b>ADM_TN</b>	This allows system administrators to login to the firewall system with strong user authentication. ADM_TN also requires use of special telnet clients that support encrypted data streams. This is an optional feature of T.Rex that is licensed separately. This feature is currently restricted and is available in limited countries.  ADM_TN requires the specification of either PRO_TNSA or UNPRO_TN or both.

The current encrypted telnet clients designed to work with the Administrator option require the use of a shared private key. This shared private key is generated with the **tnkey** command described in section 5.6

**-f FTP\_access** FTP access is used to grant the use of FTP at the individual user level. The available options are: .

<b>NO_FTP</b>	The user can not use FTP. This option is mutually exclusive with all other FTP options.
<b>UNPRO_FTP</b>	The user on an unprotected network can use FTP after passing strong user authentication.
<b>PRO_FTP</b>	The user on a protected networks is allowed to use FTP. This option is mutually exclusive with PRO_FTPSA.
<b>PRO_FTPSA</b>	The user on a protected networks is allowed to use FTP after passing string user authentication. This option is mutually exclusive with PRO_FTP.
<b>GET_FTP</b>	This allows the user on a protected networks to GET files or on an unprotected networks to PUT files. The view point is that of the protected users.
<b>PUT_FTP</b>	This allows a user on a protected networks to PUT files or on an unprotected networks to GET files. The view point is that of the protected users.
<b>DEL_FTP</b>	This allows a user on a protected networks to DEL files. The view point is that of the protected users.

If the FTP option is not specified the default value defined for the user group will be used.

## 4.9 Deleting a User

### **gwuser -d user**

**-d user** This is used to remove a user from the T.Rex gateway. This argument is mutually exclusive with gwuser -a and gwuser -l.

## 4.10 Listing a User

### **gwuser -l [user] [-a] [-g groups]**

- l [user]** This is used to list one or more user entries in the /etc/firewall/gwuser file. If there is no user argument then all entries will be printed. You can use wildcards for user names. For example, if you enter gwuser -l j\* then all entries starting with the characters j will be listed. This command will only list users who belong to the group(s) managed by the systems administrator who issued the command. This argument is mutually exclusive with gwuser -a, and gwuser -d.
- a** If the -a parameter is specified then all fields in the user record will be listed. If -a does not follow the user parameter then only a summary listing is printed.
- g groups** List users in the following groups. If not specified list users in all groups. If a wildcard is specified enclose the group name in quotes.

## 4.11 Modifying a User

One or more fields in a user record can be modified with the following command. The modify user command has two additional options ( lock and unlock) not found in the add user command. Descriptions of the options can be found in the add user section.

**gwuser -m user [-g group] [-n "actual name"] [-ph "phone number"]**  
**[-p password] [-s CRYPTO | SNK | SDI]**  
**[-k nnn nnn nnn nnn nnn nnn nnn nnn]**  
**[-t [ UNPRO\_TN ] [ PRO\_TN | PRO\_TNSA ] [ ADM\_TN ]**  
**[-f [ UNPRO\_FTP ] [ PRO\_FTP | PRO\_FTPSA ]**  
**[ GET\_FTP ] [ PUT\_FTP ] [ DEL\_FTP ] ]**  
**[-lock | -unlock]**

- lock** Lock out user access for the record. If the -a parameter is specified then all fields in the user record will be listed. If -a does not follow the user parameter then only a summary listing is printed.



## 4.12 Examples

### 4.12.1 Adding a Group

The following example shows how to add a new group using the default values specified in the /etc/firewall/gwuser.conf file.

```
#gwgroup -a mash4077 -t PRO_TN
```

### 4.12.2 Listing a Group

The following example demonstrates how to list all the information stored for group mash4077. The default security token is CRYPTO. The default times members of the group are allowed to use the firewall are Monday to Friday 8 am to 5 pm.. Users are allowed to use telnet from a protected host. Telnet from an unprotected host is not allowed unless the group member has that function enabled in their user record. FTP is not allowed unless it is explicitly granted in a users record. .

```
#gwgroup -l mash4077 -a
```

```
Group ID:      mash4077
Real name:     Null
Token type:    CRYPTO
Telnet access: PRO_TN
FTP access:    NO_FTP
Access times:  Day/Hours
                000000000011111111112222
Day            012345678901234567890123
Sunday         .....
Monday         .....YYYYYYYYYY.....
Tuesday        .....YYYYYYYYYY.....
Wednesday      .....YYYYYYYYYY.....
Thursday       .....YYYYYYYYYY.....
Friday         .....YYYYYYYYYY.....
Saturday       .....
```

```
Record date:   Wed May 21 09:41:32 1997
```

```
Number of group records listed = 1.
```

### 4.12.3 Adding a User

The following example demonstrates adding a user to the T.Rex system and selectively listing user information to validate the process. In this example the user radar is added to the group mash4077. He will be able to use the telnet proxy from a protected host but not from a unprotected host. He will

not be given use of the ftproxy.

```
# gwuser -a radar -g mash4077 -n "Walter O'Reilly" -p Keokuk1931 -t PRO_TN
```

#### 4.12.4 Listing a User

```
#gwuser -l radar -a
```

```
User ID:      radar
Real name:    Walter O'Reilly
Phone number: Null
Group:        mash4077
Password:     #####
Password expires: Mon Aug 18 17:59:21 1997
Token type:    Default
Token key:     155 012 126 066 110 124 155 054
Telnet access: PRO_TN
FTP access:    Default
Start date:    Now
Stop date:     Never
Access times:  Default
Record date:   Tue May 20 17:59:21 1997
User status:   Unlocked
```

Number of user records listed = 1.

Notice that radar has been assigned to the group mash4077 and has not been assigned a security token. He will assigned a token type t receive the Since he has no security token he will not be allowed to use the tnproxy or ftproxy from an unprotected host. Since no security token was assigned the security token has been set to zeros. If radar tries to use the ftproxy he will be denied service. The tnproxy will be available from a protected host but not from an unprotected host. If radar is allowed to login to the firewall system he will be given a restricted shell. Before radar can login to the system the systems administrator must add his ID to the system.

#### 4.12.5 Listing all users in a group

```
#gwuser -l -g mash4077
```

```
henry
klinger
macahay
margaret
potter
radar
trapperjohn
```

#### 4.12.6 Deleting a user

The user henry can be deleted from the system using the following command.

### **#gwuser -d henry**

The preceding command will delete the user record from the /etc/firewall/gwuser file.

## **4.12.7 Adding an Administrative User**

The following example demonstrates adding an administrative user to the T.Rex system. This user is allowed to login from a remote systems using ptelnet, as well as use the Hoplite Client for remote Administration.

```
# gwuser -a potter -g mash4077 -n "Colonel William T. Potter" -p HorseHockey1 -s CRYPTO -t PRO_TNSA
UNPRO_TN ADM_TN -f UNPRO_FTP PRO_FTP -times ALL
```

After creating the administrator record you should generate the shared private key for the ptelnet and Hoplite clients. To generate a private key for just Colonel Potter issue the following command.

```
# /usr/local/etc/tnkey potter > /tmp/T.Rex.key
```

The following example shows how to generate a file containing private keys for all the users.

```
# /usr/local/etc/tnkey `/usr/local/etc/gwuser -l` > /tmp/T.Rex.key
```

The shared key /tmp/T.Rex.key must be placed on the client machine per the instructions found in the ptelnet chapter.

## **4.13 Gwuser Configuration File**

The gwuser configuration contains the default values for gwuser when defining a new group or user. The product ships with the following values in the configuration file.

```
# file: /etc/firewall/gwuser.conf          system: T.Rex gateway
#
# function: The /etc/firewall/gwuser.conf file is used by the gwuser
#           database administration utility.
#
# (C) Freemont Avenue Software, Inc. 1995-2000.
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The following entries are mandatory, but the values can be changed
```

```

# Default group for new user
default_group = staff

# Default token type for new group (CRYPTO, SNK, SDI (AIX only), NONE)
default_token = CRYPTO

# Default telnet access for new group (NO_TN, UNPRO_TN, PRO_TN, PRO_TNSA,
# ADM_TN, or combination; PRO_TN and PRO_TNSA are exclusive; ADM_TN also
# requires PRO_TNSA, UNPRO_TN, or both)
default_telnet = NO_TN

# Default FTP access for new group (NO_FTP, UNPRO_FTP, PRO_FTP, PRO_FTPSA,
# GET_FTP, PUT_FTP, DEL_FTP, or combination; GET_FTP, PUT_FTP, and DEL_FTP
# also require PRO_FTP, PRO_FTPSA, or UNPRO_FTP; PRO_FTP and PRO_FTPSA are
# exclusive)
default_ftp = NO_FTP

# Default time access for new group (ALL, NONE, MTuWThFSaSu, 1-12, A, P; a
# hyphen indicates a range, otherwise just the indicated day or hour applies;
# hours must be followed by A or P and modify preceding days; if hours are
# not specified, 12A-12P applies; minutes are not allowed)
default_times = M-F 8A-5P

# Password min length (4 - 16)
password_min_length = 6

# Password min alpha characters (0 - 16; password_min_alpha
# plus password_min_other must not exceed password_min_length)
password_min_alpha = 3

# Password min non-alpha characters (0 - 16; password_min_alpha
# plus password_min_other must not exceed password_min_length)
password_min_other = 1

# Password max change interval in days (UNLIMITED, 1 - 365)
password_max_age = 90

# Number of unique passwords before reuse (NONE, 1 - 12)
no_unique_passwords = 4

# Number of failed logins before locked (UNLIMITED, 1 - 12)
no_login_attempts = 3

```

## 4.14 Configuring Restricted Shells

When user's telnet to the T.Rex firewall they will be connected to the tnproxy which will connect them to another remote system without logging on to the firewall system. There are occasions when certain users will need to login to the firewall. One such use would be to allow systems administrators to login to run the gwuser program. In order to login to the firewall a systems administrator will telnet to an assigned port. This port will be associated to the telnetd allowing a user to login to T.Rex. When users login to T.Rex they are assigned a restricted shell, that limits the commands the administrator can issue on the firewall system.

T.Rex allows different shells for different users. The gwuser command is used to define which shell a user is assigned when they login to the T.Rex firewall. This chapter shows how to control the use of the restricted shell.

The idea of a restricted shell is to restrict the commands the user can perform on the system (for instance, you don't necessarily want the user editing and compiling code on your gateway). The gateway shell (gwsh2) determines which shell a user receives. When a user receives a restricted shell the only commands allowed are those defined in the restricted shell. All other commands will be rejected.

## Sample Restricted Shell

The following example demonstrates how to code a restricted shell. This sample can be modified to serve your needs. A soft copy of this example is included with the product in the file restrict.sh.

```
#!/bin/sh
# guest shell:
PATH="/bin:/usr/ucb"
fromhost="/usr/user/rmthost"
echo Currently logged in from $fromhost
echo imported TERM=$TERM
if test "$TERM" = "network"
then
    echo =====
    echo Your terminal type was not exported.
    stty rows 24 columns 80 -tabs
fi
echo -n 'Please enter what you would like set as TERM ['$TERM'] '
read term
if test "$term" = ""
then
    term=$TERM
fi
TERM=$term; export TERM
echo Your TERM variable has been set to $TERM
DISPLAY="$fromhost":0; export DISPLAY
echo Your DISPLAY variable has been set to $DISPLAY
prompt=`hostname`
trap "" 2          # ignore SIGINT
while true
do
    echo -n "$prompt: "
    read command arg1 arg2
    if test $? -ne 0
    then
        # end-of-file
        echo
        exit 0
    fi
    if test "$command" = ""
    then
        # null command
        continue
    fi
    case $command in
        passwd) /bin/passwd;;

        finger) /usr/ucb/finger $arg1;;
```

```

tn3270) /usr/ucb/tn3270 $arg1;;
tn) /usr/ucb/tn $arg1 $arg2;;
tnctest) telnet -e 3270 $arg1;;
display) DISPLAY=$arg1; export DISPLAY;;
showdisp) echo $DISPLAY;;
ping) /etc/ping $arg1 56 20;;
exit|logout) exit 0;;
xant) /usr/user/xant $arg1;;
xarchie) /usr/user/xarchie;;
gopher) /usr/user/gopher;;
xgopher) /usr/user/xgopher $arg1;;
xmosaic) /usr/user/xmosaic;;
irc) /usr/user/irc;;
stty) /bin/stty $arg1 $arg2;;
traceroute) /usr/user/traceroute $arg1;;
dig) /usr/user/dig $arg1 $arg2 $arg3;;
whois) /usr/ucb/whois $arg1;;
weather) /usr/bin/telnet 141.212.196.79 3000;;
*) echo $command not valid
esac
done
exit 0

```

This is a basic shell. It starts up by determining the user's remote Internet address (using the `rmthost` command, also part of this distribution), sets up various environment variables (like terminal type), then allows the user to issue the following commands:

`passwd`, `finger`, `tn3270`, `tn`, `tnctest`, `display`, `showdisp`, `ping`, `exit`, `xant`, `xarchie`, `gopher`, `xgopher`, `xmosaic`, `irc`, `stty`, `traceroute`, `dig`, `whois`, and `weather`.

You may want to eliminate some of the commands shown in this example. For example, if you don't support `tn3270` or `xarchie` those commands should be removed. The following commands allow no arguments to be passed to them: `passwd`, `exit/logout`, `showdisp`, `xarchie`, `gopher`, `xmosiac`, `irc`, and `weather`.

`finger` allows the user to pass one argument it. `telnet` allows up to two arguments to be passed to it. You might want to be cautious about the number of arguments that can be passed (especially if you don't want users running a bunch of programs in background by using the "&").

It's easy to add commands to this shell. You can even have different shells for different people.

If you want to pass arguments to the command use something like this (this example allows passing of two arguments to the command).

`command) /bin/command $arg1 $arg2;;`

## 4.13 Generate Encrypted shared keys

Use of remote T.Rex administration requires the generation and distribution of shared private keys. The encrypted shared key is generated using the **tnkey** command on the firewall, and uses information in the `gwuser` database to generate the key to allow a user to initiate an encrypted telnet session with T.Rex. This commands requires root authority to run.

**tnkey** [-a ] [-o] user [user....]

where:

- a**      Generate keys for all users in the gwuser database.
- o**      Generate encrypted key file for use with oobasrv. This allows oobasrv to perform automated persistent OOBA.
- user**    The administrative user ID for whom the key is being generated. More than one user can be specified at one time.

tnkey generates keys that are 48 hexadecimal digits long. The keys generated are output to stdout. You can capture the output in a file using shell redirect (>).

#### 4.13.1 Generate Key for ptelnet and hoplite

The following example shows how to generate the shared private key for user ID jim. This key is used by ptelnet and hoplite. The key is redirected to the file /tmp/.T.Rex.key.

```
# /usr/local/etc/tnkey jim > /tmp/.T.Rex.key
```

The saved output file should be put in the /etc directory of the user's \$HOME directory and given a filename **.T.Rex.key**. The T.Rex encrypting telnet client looks for .T.Rex.key first in \$HOME and then in /etc. If the file is not found, the client will not be able to initiate an encrypted session with the T.Rex tnproxy.

Note: A key will not be generated if the user entry in the gwuser database contains a null security token. An informational message will be displayed in this situation.

You can generate keys for multiple users as follows:

```
# /usr/local/etc/tnkey `usr/local/etc/gwuser -l "*" ` > /tmp/.T.Rex.key
```

#### 4.13.2 Generate Encrypted user Key for OOBASRV

The following example shows how to generate the shared private key for user ID jim. This key is used by the oobasrv program for automated persistent OOBA. The key is redirected to the file /tmp/oobasrv.usr.

```
# /usr/local/etc/tnkey jim > /oobasrv.usr
```

Notes:



# Chapter 5. Using Security Tokens

The current version of T.Rex supports the SecureNet Key from Digital Pathways. and the CRYPTOCARD from CRYPTOCARD, Inc.

## 5.1 Programming the SecureNet Key (SNK)

Here are the instructions in brief for programming the SecureNet Key ( The SecureNet Key<sup>1</sup> is a small device (looks like a small calculator) for authenticating users who are coming into the gateway from a non-secure network. You will get a full set when you order the keys):

1. Turn the key **on** (it will display **"E0"**).
2. Type **"5"** then **"ent"** to specify hexadecimal mode (it will display **"E1"**).
3. Type each three digit octal number (**001 through 177**) ( the card will prompt you with numbers **1,2, .. 8**). After typing in all eight octal numbers press the **"ent"** key. The SNK will display the check sum.
4. Press the **"ent"** key again and the SNK will display **"E2"**.
5. Type a full 4 digit PIN number for the user **xxxx** and press the **"ent"** key. The SNK will then display **"E3"**.
6. Type the 4 digit PIN number again **xxxx** and press the **"ent"** key. The SNK will display EP. Congratulations you are done programming the key..

Now the SecureNet Key is ready for use. The user telnets to the gateway, gives his user-id and password, and is presented an 8 character challenge. The user turns on his SecureNet Key, enters his PIN and "ent" (the card will say "Ed"), enters the challenge and "ent", then types the response (in hex) presented on the LCD of the SecureNet Key. If it is correct, the system lets the user in.

## 5.2 Clearing SNK memory

There will be occasions when you will want to clear the SNK memory and reprogram the card.

1. Turn the key **on** if it displays **"EP"** continue. If it displays **"E0"** memory has already been cleared and you can following the instructions in section 4.1 above.
2. Type the number **"3"** and press the **"ent"** key. If the SNK displays **"Error"** then repeat steps 1 & 2 until the SNK displays **"E0"** ( this process usually takes 4 to 5 iterations) . When **"E0"** is displayed the memory has been cleared and you can use the procedure in section 4.1 to reprogram the SNK. If the SNK displays **"Ed"** go to step 3.
3. Type in **"00000000"** and press the **"ent"** key. Ignore the response number displayed by the key.
4. Press the **"on"** key, type in **"3"** and press the **"ent"** key. The display should now read **"Ed"**.
5. Type in **"00000000"** and press the **"ent"** key. If the SNK displays **"E0"** then the memory has

---

<sup>1</sup> The SecureNet Keys are available from Digital Pathways, 201 Ravendale Dr., Mt. View, CA 94043 at a cost of approximately \$60 each.

been cleared and you can use the procedure in section 4.1 to reprogram the SNK. Otherwise go back to step 1 and repeat the procedure.

Detailed instructions are available in the SNK User's Guide published by Digital Pathways.

## 5.3 Programming the CRYPTOCARD

This is the quick and easy way to format your CRYPTOCARD with what we consider a good base level of functionality. If you would like to explore the other functions the card is capable of handling then read their instructions.

If you would like to understand the meaning of these input directions, in other words, "what am I doing?", then follow along with the card's instructions. This is a quick cookbook work sheet for those who hate reading the "technical-ese."

Press the following buttons on the CRYPTOCARD (Input is in **BOLD**)

1. **ON/OFF** (displays "**Locked**")
2. **ENT** (displays "**Options?**")
3. **100 ->** (goes to option 2)
4. **134 ->** (goes to option 3)
5. **101 ->** (goes to option 4)
6. **ENT** (displays "**key1?**")
7. enter eight three digit octal keys (**001** through **177**) , with a -> after each 3-digit set. . This key must match the value in the user record found in the gwuser file.
8. **ENT** (displays eight character hex number)
9. **ENT** (displays "**User ID?**")
10. Enter eight three digit octal numbers for the user ID using a character to octal conversion chart, with a -> after each 3-digit set. Shorter user ID's may be used with a octal 040 (space) entered for unused characters. Section 4.5 has an ASCII to Octal conversion chart.
11. **ENT** (displays user ID)
12. **ENT** (displays "**new PIN?**")
13. enter PIN number (this number **must** be changed after card activation so don't use your favorite one here)
- 14] **ENT** (displays "**Card OK**")

When you turn your card on again you will have to put a new PIN number in.

## 5.4 Clearing CRYPTOCARD memory

To erase the memory in your CRYPTOCARD after it is fully programmed, and then reprogram it you can do the following:

1. Turn on your card and before anything appears on the screen type **225371**. This is

elaborated under the topic of "First Quality Assurance Mode" in the CRYPTOCARD Operation Guide. You will have to type fast for this procedure to work.

2. If procedure 1 fails or you can't type the numbers in fast enough simply take both batteries out of the card. This works all the time.

## 5.5 Octal Conversion Chart

The following chart can be used to convert ASCII characters to octal format used to enter user IDs into the CRYPTOCARD.

Character	Octal	Character	Octal
A	101	a	141
B	102	b	142
C	103	c	143
D	104	d	144
E	105	e	145
F	106	f	146
G	107	g	147
H	110	h	150
I	111	i	151
J	112	j	152
K	113	k	153
L	114	l	154
M	115	m	155
N	116	n	156
O	117	o	156
P	120	p	160
Q	121	q	161
R	122	r	162
S	123	s	163
T	124	t	164
U	125	u	165
V	126	v	166
W	127	w	167
X	130	x	170
Y	131	y	171
Z	132	z	172

A blank is represented by an octal 040.

# Chapter 6. Aproxy Administration

## 6.1 Aproxy Overview

Aproxy provides generic proxy services for connection oriented TCP/IP applications with access control and logging. Aproxy is a standalone proxy that is started at system boot time. It reads the aproxy configuration file and parses the rules into a table stored in RAM. Based on the rules aproxy pre-forks an initial number of processes and listens to the specified ports. In addition to controlling client-server connectivity aproxy also provides an API and server workload balancing.

### API

Aproxy has an Application Programming Interface (API) to allow it to be customized for any number of applications.

Each permit statement provides three application exits. One for authorization and authentication. The second, provides access to the data read from the client. The third provides access to data read from the server.

### Work Load Balancing

Aproxy provides workload balancing to distribute application traffic across an array of servers. Aproxy controls and balances traffic at the application (port) level. Aproxy supports two load balancing methods: round robin and ratio. If one of the servers is twice as fast aproxy can be configured to distribute it twice as much work.

### Improved Application Availability

Aproxy can improve the reliability and availability of application servers in the following ways:

#### Content distribution:

Distribution of data and functions across multiple servers allows individual servers to be brought down for preventative maintenance and returned to service without disrupting service.

#### Intelligent workload distribution:

Workload balancing spreads the workload across multiple servers preventing server overload. If an individual server fails aproxy stops dispatching work to it until the server comes back online.

#### Hardware and Software redundancy:

The T.Rex High Availability option uses two redundant systems that dynamically share the workload and automatically detect and recover from hardware or software failures. Should one of the redundant units fail the other assumes all the functions to prevent service disruptions.

### Improved performance

Aproxy is faster than genproxy since it pre-forks processes reducing systems overhead up to 90%. The number of processes are automatically adjusted to match the workload. Aproxy reads and parses the /etc/firewall/aproxy.conf file at start up time. This saves the overhead of parsing the configuration file for every application connection.

There is a refresh command to for Aproxy to re-read the configuration file, after changes have been made.

### **Other improvements**

Aproxy is simpler to configure than genproxy since it does not run under control of inetd. With genproxy you have to edit both inetd.conf and genproxy.conf to add support for a new application. With Aproxy you only need to modify aproxy.conf.

Aproxy only listens to the IP-Address and port-number specified in the permit statement. Since genproxy ran under control of inetd it would listen to all IP address for a given port number. This makes Aproxy stealthier than genproxy.

Aproxy uses permit and deny statements to control access to server applications. If NOT permitted the connection is closed and the attempt is logged. If the requesting host is allowed to connect the permit rule determines the IP address and port number of the host on the other side. Permissions are granted based on the source IP address, local IP address and local port number. If there is a match the connection is made to the host and port number specified after the redirect keyword. To allow the server to distinguish between multiple clients there is an optional keyword "using" forces Aproxy to bind to a specific local IP address when connecting to the server.

Aproxy supports one-to-one and many-to-one connections. Many-to-one connections can be permitted using multiple permit statements or by using wildcards to specify a range of source addresses on a permit rule.

### **Out Of Band Authentication (OOBA)**

Aproxy provides optional **Out Of Band Authentication** for the client requesting proxy services. Use of ooba causes the genproxy program to connect to an ooba server running on the clients host. Aproxy will challenge the ooba server using with a random 8 digit decimal number. The ooba program on the client host must generate an 8 digit hexadecimal character string using the DES algorithm and the same key stored in the user record specified in the permit statement. Aproxy reads the response. If the response is correct Aproxy will issue a message to the ooba server, close the ooba connection then connect the client to the requested service. If the response is not correct an error message is sent to the ooba server, the ooba connection is broken and the primary Aproxy connection is broken.

Aproxy can be used to support many different client server applications, such as Informix, Sybase, and Oracle data bases, Lotus Notes, and others. Combined with the OOBA functions this can be done in a secure manner without modifications to the client server code.

### **Multiple Servers Support**

Multiple servers for the same function can reside behind the firewall each having their own name and IP address. For example, it is possible to support two or more LOTUS Notes Servers behind the same firewall all using the same port number.

To support multiple servers with the same port number the firewall must listen to a separate IP address for each of the servers. This can be done using one or more LAN adapters. A single LAN adapter can be configured to listen to multiple IP addresses by defining aliases with the appropriate command. The following example shows the addition of two alias IP addresses to be used by Ethernet adapter 0. You can use `netstat -i` to confirm the changes.

#### **For AIX:**

The following commands should be added to the `/etc/rc.net` file to activate the aliases when the system is booted.

```
#ifconfig en0 204.71.109.32 alias
#ifconfig en0 204.71.109.33 alias
```

#### **For HP-UX:**

The following commands should be added to the `/sbin/init.d/net` file following the `ifconfig` line. This will activate the aliases when the system is booted.

```
#ifalias lan0 204.71.109.32
#ifalias lan0 204.71.109.33
```

#### **.For Solaris:**

The following commands should be added to the `/etc/init.d/rootusr` file. This will activate the aliases when the system is booted.

```
#ifconfig le0:1 204.71.109.32 netmask 255.255.255.0 up
#ifconfig le0:2 204.71.109.33 netmask 255.255.255.0 up
```

## **6.2 Aproxy Configuration File**

The `aproxy` program is controlled by rules defined in the `/etc/firewall/aproxy.conf` file. Connections between external and internal hosts are permitted or denied based on the **permit** and **deny** rules. In most cases the `deny` rule need not be coded, since the default action is to deny all requests except those which are permitted. The order in which the rules are coded is very important, since the first rule encountered that has a match with the source address, local address and destination port number is enforced.

## **Command Syntax**

Each rule is contained on a line that can be up to 1023 bytes long. Spaces and tabs separate fields. Optional fields are enclosed in square brackets.

## Mandatory Commands

The `gproxpath` command is mandatory and must be coded before the deny and permit rules.

### **gproxpath = pathname**

The `gproxpath` value specifies the directory that `aproxy` will run under.

## Optional Commands

### **bufsize = nnnn**

The data buffer size in bytes. The default size is 4096 bytes which is good for most applications. If the buffer is too small or too large performance can be reduced. If you make the data buffer larger than 4 KB you may have to tune the TCP/IP parameters.

### **checkprocs = nnn**

The time interval in seconds that `aproxy` waits before checking on the number of child processes. The default value is 5 seconds.

### **maxprocs = nnn**

The maximum number of `aproxy` processes permitted.

### **maxuse = nnn**

The maximum number of times a process is to be used before it is retired and replaced by a fresh process. This automatically recovers any resources that may have been tied up by a long running process. For best performance set `maxuse` to 1000 uses. The default value is 500 uses.

### **nprocs = nnn nnn nnn**

The `nprocs` command is used to specify the number of processes to start, the minimum number of spare idle processes and the maximum number of spare idle processes. The first number is the number of pre-allocated processes to be started. If this parameter is not coded the default value is 20.

### **timeout = nnnn**

The timeout value specifies the number of seconds the `aproxy` program will wait for a message before it exits. The value `nnnn` must be a positive integer less than 1000000000. The default value is 60 seconds.

## Access Control Rules



## **deny from src\_addr to loc\_addr loc\_port**

<b>from</b>	The from keyword must immediately follow the deny command. This is used to make the command syntax consistent.
<b>src_addr</b>	The source address field is required and must immediately follow the <b>deny</b> keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk '*' can be used as a wild card for source IP addresses. For example, 192.168.240.* would deny all from 192.168.240.0 through 192.168.240.255
<b>to</b>	The to keyword is required and must immediately precede the <b>loc_addr</b> field.
<b>loc_addr</b>	The local host address field is required and must immediately follow the <b>to</b> keyword. The local address gives the IP address of the firewall that is to receive the connection from the source host. This address is in the standard dot decimal format. An asterisk '*' can be used as a wild card for source IP addresses. For example, 192.168.240.* would deny all connections attempted to addresses from 192.168.240.0 through 192.168.240.255 from the source address(es) previously coded.
<b>loc_port</b>	The local port field is required and must immediately follow the <b>loc_addr</b> field. The loc_port can be either the port number or the name of the service found in the <b>/etc/services</b> file.

## **permit from src\_addr to loc\_addr loc\_port redirect (dest\_addr1, dest\_addr2, ...) dest\_port [using loc\_addr2] [ignore rst [sleep\_secs] ] [userexit exitname] [ooba userid ooba\_port]**

<b>from</b>	The from keyword must immediately follow the permit command. This is used to identify the new command syntax. .
<b>src_addr</b>	The source address field is required and must immediately follow the <b>from</b> keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk '*' can be used as a wild card for source IP addresses. For example, 192.168.240.* would permit all from 192.168.240.0 through 192.168.240.255
<b>to</b>	The to keyword is required and must immediately precede the <b>loc_addr</b> field.
<b>loc_addr</b>	The local host address field is required and must immediately follow the <b>to</b> keyword. The local address gives the IP address of the firewall that is to receive the connection from the source host. This address is in the standard dot decimal format. The permit command does not support wild cards for the local address.

<b>loc_port</b>	The local port field is required and must immediately follow the <b>loc_addr</b> field. The <b>loc_port</b> can be either the port number or the name of the service found in the <b>/etc/services</b> file.
<b>redirect</b>	The <b>redirect</b> keyword is required and must immediately precede the <b>dest_addr</b> field.
<b>dest_addr</b>	The destination address field is required and must immediately follow the <b>redirect</b> keyword. The destination address gives the IP address of the target host in the standard dot decimal format. Wildcards can not be used with destination IP addresses. More than one destination can be coded if they are enclosed within parentheses.
<b>dest_port</b>	The destination port field is required and must immediately follow the <b>dest_addr</b> field. The <b>dest_port</b> can be either the port number or the name of the service found in the <b>/etc/services</b> file.

#### **using loc\_addr2**

The optional keyword **using** specifies the local IP address to be used for the connection to the destination host. This feature would normally be used if a LAN adapter has been configured to support multiple IP address, and wants to control which IP address is to be used. This feature makes it possible for the server to determine which client is requesting a connection. This parameter is optional and must follow the destination port.

#### **ignorerst sleep\_sec**

The optional keyword **ignorerst** is used to tell aproxy to ignore connect resets from the destination computer. This is used when a print server is sending print jobs to a printer. The printer server will normally ignore the reset command when coming from a printer. Without this parameter aproxy would close the connection and make the server think the printer had died. This parameter is optional and must follow the destination port or the **using loc\_addr2** parameter.

#### **userexit exitname**

The optional keyword is used to define a user exit. The exit name is a non-blank character string with a maximum length of 15 characters. Each permit statement can have a unique exit name. Exit names can also be shared by multiple permit statements. Each exit is given control at three points. Control is passed to the exit for authentication. Control is passed to the client exit when data is read from the client. Control is passed to the server exit when data is read from the server. API exits must be explicitly coded for control to be passed to an exit routine. Rules for coding exits are listed in at the end of this chapter.

**ooba** The optional keyword **ooba** is used to specify **Out Of Band Authentication** for the client requesting proxy services. Use of **ooba** causes the genproxy program to connect to an ooba server running on the clients host. Genproxy will challenge the ooba server and read its response. If the response is correct genproxy will connect the client to the requested service. Otherwise the connection is broken.

If the **ooba** keyword is specified it must be followed by a **userid** and the **ooba\_port** number.

<b>userid</b>	This is the userid that contains the DES key to be used for OOBA. This userid must be in the /etc/firewall/gwuser data base. The OOBA server must use the same DES key to generate the correct response to the challenge.
<b>ooba_port</b>	This is the port number assigned to the ooba server on the client host.

## 6.3 Sample aproxy.conf file

The following sample configuration file is shipped with T.Rex Version 1. Edit the file to meet your unique requirements.

```
# file: /etc/firewall/aproxy.conf system: T.Rex gateway
# function: The /etc/firewall/aproxy.conf file is used to control Aproxy.
# Aproxy permits or denies access to the requested service
# based on "permit" and "deny" rules found in this file.
#
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The first permit statement allows all systems to access the application on port 7000.
# The second permit statement distributes the workload for port 7200 across three servers.
# The next two permit from statements allow all systems to access the applications on port 2781
# by connecting to one of two addresses. The using statement allows the server at 10.10.0.15 to
# determine which external address the client connected to.
#
# The fifth permit allows any client to connect to port 9876. The user exit pgm9876 is given control
# during processing.
#
# The host at IP address 10.11.0.17 is allowed to connect to
# the Network News Server at IP address 198.65.128.14.
# Any other host attempting to establish an NNTP connection through the
# T.Rex firewall will be denied access.
#
gproxpath = /home/hermes
timeout = 300
permit from * to 204.71.109.32.1 7000 redirect 10.10.0.15 7000
permit from * to 204.71.109.32.1 7200 redirect (10.10.0.5,10.10.0.6,10.10.0.7) 7200
permit from * to 204.71.109.32.1 2781 redirect 10.10.0.15 2781 using 10.10.0.2
permit from * to 204.71.109.33.1 2781 redirect 10.10.0.15 2781 using 10.10.0.3
permit from * to 204.71.109.33.1 9876 redirect 10.10.0.16 9876 userexit pgm9876
```

deny from \* to 204.71.109.32.1 NNTP

## 6.4 To Proxy SQL

Aproxy can proxy connections between SQL clients and SQL Servers provided by Oracle, Sybase and other relational data base. To enable SQL client server connections do the following.

1. Define the SQL service in the /etc/services file if it is not already defined.
2. Edit the /etc/firewall/aproxy.conf file and add a permit statement for the service.
3. Run the /usr/local/etc/refresh\_aproxy command to make the new rules take effect.

### 6.4.1 To Proxy SQL\*NET

If the Oracle Server is running as a multiple process server then aproxy can be used to connect the client to the server. If the Oracle server is set up as a multi-threaded server (mts) then there is no simple aproxy solution. To set up aproxy to handle Oracle's SQL\*NET do the following.

1. Add the following line to /etc/services.

```
sqlnet 1521/tcp
```

The Oracle server listens to the well known port 1521, so it must be specified in the /etc/services file.

2. Add the following line to /etc/firewall/aproxy.conf

```
permit from * to 204.71.109.32 sqlnet redirect 10.10.0.15 sqlnet
```

Strong user authentication for SQL\*Net can be enforced by specifying Out Of Band Authentication as follows: .

```
permit from * to 204.71.109.32 sqlnet redirect 10.10.0.15 sqlnet ooba & 7500
```

This requires installation of the ooba client on the user's machine. The user will need a security token programmed with the same private key stored in the user's record in the gwuser data base.

3. Run the /usr/local/etc/refresh\_aproxy command to make the new rules take effect.

## 6.5 To Proxy pcAnywhere

Aproxy can be used to let pcAnywhere connect a remote client to a local PC. Use of Out Of Band Authentication (OOBA) is recommended to ensure strong user authentication if one allows connections from an insecure network such as the Internet. The following example shows how this is done.

```
permit from * to 204.71.109.32.1 5631 redirect 10.10.0.15 5631
permit from * to 204.71.109.32.1 5632 redirect 10.10.0.15 5632
```

pcAnywhere also requires the use of RPCproxy to let some UDP packets through. To do so add the following permit statements to rpcproxy.conf and refresh the proxy to incorporate the new rules. See the Rpcproxy chapter for details.

```
permit udp from * * to 204.71.109.32.1 5631 redirect 10.10.0.15 5631
permit udp from * * to 204.71.109.32.1 5632 redirect 10.10.0.15 5632
```

## 6.6 User Exits

Aproxy allows customization of the proxy by providing exits that are given control during processing. This section provides the details necessary to create and incorporate exits into the proxy.

### 6.6.1 API Code

Aproxy ships with sample code to assist in the development of the user exits. The samples include the following.

A makefile.

An AIX export file.

A libuser\_exits.a (or .so).

It contains the object from user\_exits.c. It references libuser.a for all the user exit functions. This is a dynamic library and has to be this way. This file must be included in the user's LIBPATH or LD\_LIBRARY\_PATH before it can load. The library is in the /usr/local/etc/exits directory.

A user\_exits.h header file.

This header must be included into the exit program(s). This header contains structures and defines used by the Aproxy user exit interface. It also describes the prototype of each of the user exits.

The user\_exits.c file.

This program defines an extern table whose entries represent sets of user exit functions. Each entry is identified by a string of characters associated with a permit statement.

The libuser.a library.

A user provided library that contains the user exit functions. This is setup as a static library. The user will have to change the makefile if a shared library is needed. A libuser.a is supplied with aproxy and contains example user exit functions.

### 6.6.2 Example Exit Programs

addcrlf.c

Example user exit function that scans the input buffer and replaces all occurrences of '\n' with '\r\n'.

rmcrlf.c

Example user exit function that scans the input and strips all occurrences of '\r'.

auth.c

Example user exit that simulates an authentication sequence.

These 4 files are simple example programs that act as client/server applications to be enabled thru the firewall by aproxy to demonstrate the user exit capabilities.

serverget.c

Example program that pairs with clientput.c to transfer a file from the client to the server.

clientput.c

Example program that pairs with serverget.c to transfer a file from the client to the server.

serverput.c

Example program that pairs with clientget.c to transfer a file from the server to the client.

clientget.c

Example program that pairs with serverput.c to transfer a file from the server to the client.

Note that the example programs (clientput.c, serverget.c, clientget.c and serverput.c) don't work on Windows.

### 6.6.3 Activation of a new Exit

To activate a new user exit do the following:

1. Write the functions.
2. Add an extern declaration for each function in user\_exits.c
3. Add entries in the u\_exits table. Each entry contains:
  - id-string that's up to 15 characters long
  - pointer to the Auth exit or NULL if none
  - pointer to the CtoSBuf exit or NULL if none
  - pointer to the StoCBuf exit or NULL if none
4. Run make in the directory
5. Add permit statements in /etc/firewall/aproxy.conf that each contains the "userexit" keyword with an exitname that matches an id-string in one of the u\_exits entries in user\_exits.c

## SOLARIS

Start the Aproxy program like this.

```
LD_LIBRARY_PATH=/usr/local/etc/exits /usr/local/etc/exits
```

# Chapter 7. FTP Proxy Administration

## 7.1 FTP Proxy Overview

The FTP proxy (ftproxy) provides ftp access through the T.Rex firewall without compromising the security and integrity of the firewall. Ftp proxy supports access control by user ID, group ID, source IP-address and destination IP-addresses. Strong user authentication is required for unprotected hosts. User identification and authentication is optional for protected hosts. All successful and unsuccessful connection attempts are logged, including times, source and destination IP-addresses, the number of bytes sent and received, the number of commands issued and the processor time used on the firewall. Utilities are provided which summarize ftp proxy activity.

The ftp proxy program runs in a non-privileged state chrooted to a restricted directory. The only file I/O it performs is to **read** configuration files. User activity is logged by the syslog daemon. The user is not logged on to the firewall and is unable to issue any commands or shells that run on the firewall. The ftp proxy can not be used to read or write any data on the firewall. Commands and data simply pass through the firewall. System administrators can still use the ftp client on the firewall to send and receive data from the firewall system.

**Anonymous ftp** is supported by automatically redirecting the anonymous ftp request to a designated ftp server that is configured to support anonymous ftp. The anonymous ftp service is provided in a transparent manner to the external user, making it appear as if the T.Rex firewall is the anonymous ftp server. This implementation preserves the high degree of security and integrity of the firewall since the ftp proxy does not allow users to read or write data on the firewall system. In order to activate anonymous ftp server the systems administrator has to permit the external user anonymous, and specify the name or IP address of the anonymous ftp server in the ftp proxy configuration file.

Ftp proxy is invoked by the Internet daemon **inetd** when a ftp client on another host wants to ftp to another system through the T.Rex firewall. The ftp proxy reads the **/etc/firewall/securenets** file to determine if the source host is on a protected network. Ftp proxy checks for IP address spoofing. A Connection to a secure host must be made using one of the secure network adapters defined in the **/etc/firewall/secureports** file. Spoofing attempts cause the connection to be broken and a Security Alert to be issued. What happens next depends upon whether the user is on a protected or unprotected host.

**Protected host:** If user authentication is not required for protected users then ftp proxy does not require a user id or password, and the following check is skipped. If user authentication is required for protected hosts ftp proxy asks for the user's id. It then reads the **/etc/firewall/gwuser** file to determine if the user has ftp privileges at the current time and what form of user authentication is to be used. The user is prompted for a password. If the user fails authentication or is denied ftp access a message is sent, the connection is broken and the attempt is logged.



**Unprotected host:** If the client machine is not a protected host then strong user authentication is required. The user is prompted for their user id. It then reads the **/etc/firewall/gwuser** file to determine if the user has ftp privileges at the current time and what form of user authentication is to be used. If a challenge response system is specified in the user record then the following procedure is required: (1) the user issues **quote auth** command, (2) the ftp proxy issues a challenge, (3) the user issues **quote resp nnnnnnnn** command.

After identification and authentication of the user, additional access control is granted based on permit and deny rules. Control is based on host IP-address and optionally by user-id or group-id. After passing the authentication test the user is prompted for the desired destination. The user replies with a **quote site hostname** command. Ftp proxy reads the **/etc/firewall/ftproxy.conf** file to determine if the connection is permitted or denied. If permitted the connection is made and logged. If denied a message is sent to the user, the connection is broken and the error is logged.

## 7.2 FTP Proxy Configuration File

The ftp proxy program is controlled by rules defined in the **/etc/firewall/ftproxy.conf** file. Connections between source and destination hosts are permitted or denied based upon **permit** and **deny** rules. The order in which rules are coded is very important, since the first rule that matches is enforced. Subsequent rules are ignored. There are four mandatory commands which must be coded before the access control rules. These commands govern the general behavior of the ftp proxy. There must be one or more access control rules following the mandatory commands.

### Command Syntax

Each rule is contained on a line that can be up to 1023 characters long. Spaces and tabs and equal signs "=" separate fields. Comment lines begin with the "#" character. Blank lines are ignored. The format of each command is shown below:

### Mandatory Commands

The groupid, ftproxpath, userid, and timeout commands are mandatory and must be coded before the deny and permit rules.

**groupid = name**

The groupid value is used to specify the group name the ftp proxy will run under.

**ftproxpath = pathname**

The ftproxpath value specifies the directory that ftproxy will run under.

**userid = name**

The userid value is used to specify the user name the ftproxy program will run under. The ftproxy program will not allow the userid to be root or 0.

**timeout = nnnn**

The timeout value specifies the number of seconds the ftproxy program will wait for a message before it exits.

## Optional Commands

**auth = yes/no**

The auth command is used to specify whether an internal user has to supply a userid and password to use the ftproxy. The auth command is optional. If auth is not coded in the ftproxy.conf file the proxy uses yes as the default value. If auth = no is specified the ftproxy uses the IP address of the users host in place of the users ID. If auth= no is coded the permit and deny rules should be coded to reflect the fact that IP addresses are being used to identify users.

If auth is specified then either yes or no must follow the equal sign. Otherwise ftproxy will issue an error message.

**external = yes/no**

The external command is used to permit or deny ftp access for unprotected (external) hosts. If **external = yes** is specified then ftp access is permitted if a user passes the challenge response system. If **external = no** is coded then any attempt to use ftp from an unprotected host will cause a Security Alert. If this command is not coded the default value is no.

**anonymousftp = server\_name / IP\_address**

The anonymous ftp command is used to specify the host name or IP address of the system configured to provide anonymous ftp service. To permit anonymous ftp service through the firewall, one has to allow external users (see preceding command) and code a permit command to allow the anonymous user access from unprotected network(s).

## Access Control Rules

***deny [users = usrlist] [groups = grplist] from src\_addr to dest\_addr***

The deny command will deny access to a list of users or a group of users if there is a match up of the source and destination IP-addresses.

**users = usrlist:** The users parameter is optional. The userlist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The user-id's are names of users on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading */*. The format of the file allows one or more user-id's per line. Blanks, tabs and commas can be used as separators. The *#* character marks the rest of the line as a comment.

**groups = grplist:** The groups parameter is optional. The grplist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The group-id's are names of groups on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading */*. The format of the file allows one or more group-id's per line. Blanks, tabs and commas can be used as separators. The *#* character marks the rest of the line as a comment.

**from:** The from keyword is required and must immediately precede the source IP-address.

**src\_addr:** The source address field is required and must immediately follow the from keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk *\** can be used as a wild card for source IP addresses. For example, 192.168.240.\* would permit all from 192.168.240.0 through 192.168.240.255

**to:** The to keyword is required and must immediately precede the destination IP-address.

**dest\_addr:** The destination address field is required and must immediately follow the to keyword. The destination address gives the IP address of the requesting host in the standard dot decimal format. An asterisk *\** can be used as a wild card for destination IP addresses. For example, 192.168.240.\* would permit all from 192.168.240.0 through 192.168.240.255

***permit [users = usrlist] [groups = grplist] from src\_addr to dest\_addr***

The permit command will permit access to a list of users or a group of users if there is a match up of the source and destination IP-addresses.

**users = usrlist:** The users parameter is optional. The userlist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The user-id's are names of users on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading `/`. The format of the file allows one or more user-id's per line. Blanks, tabs and commas can be used as separators. The `#` character marks the rest of the line as a comment.

**groups = grplist:** The groups parameter is optional. The grplist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The group-id's are names of groups on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading `/`. The format of the file allows one or more group-id's per line. Blanks, tabs and commas can be used as separators. The `#` character marks the rest of the line as a comment.

**from:** The from keyword is required and must immediately precede the source IP-address.

**src\_addr:** The source address field is required and must immediately follow the from keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk `*` can be used as a wild card for source IP addresses. For example, `192.168.240.*` would permit all from 192.168.240.0 through 192.168.240.255

**to:** The to keyword is required and must immediately precede the destination IP-address.

**dest\_addr:** The destination address field is required and must immediately follow the to keyword. The destination address gives the IP address of the requesting host in the standard dot decimal format. An asterisk `*` can be used as a wild card for destination IP addresses. For example, `192.168.240.*` would permit all from 192.168.240.0 through 192.168.240.255

## 7.3 Sample ftproxy.conf file

```
# file: /etc/firewall/ftproxy.conf
# function: The ftproxy.conf file is used to control the execution of the
#           ftproxy program. Connections between external and internal
#           hosts are permitted or denied based on the permit and deny rules.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
```

```

# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The format of the commands are as follows:
#
# timeout nnn the number of seconds genproxy will wait for a message
# before it exits.
#
# deny src_host src_port
#
# permit src_host src_port to dest_host dest_port
auth = yes
external = yes
groupid = staff
userid = hermes
ftproxpath = /home/hermes
anonymousftp = secure.ftp.server.FAS.com
timeout = 600
permit users = * from 10.11.0.* to *
permit users = anonymous from * to *
deny users = * from * to 10.11.*

```

# Chapter 8. HTTPD Proxy Administration

## 8.1 HTTPD Proxy Overview

### 8.1.1 Multiple Functions

The HTTPD proxy server provides multiple functions, including:

- The basic services for Web Browsing,
- Enforcement of management policies regarding use of the web,
- URL content filtering,
- Selective blocking of Java, JavaScripts, ActiveX and cookies,
- Caching of frequently used documents provides faster Web response,
- Conserves network bandwidth,
- pre-forking to eliminate up to 90% of system overhead,
- non-disruptive upgrades with on-the-fly scalability without downtime,
- fault tolerant mission critical design,
- automatic fail-over when configured as a redundant system,
- logs WWW access in Common Log Format that can be processed by third party products like WebTrends,
- Report generation programs are provided,

T.Rex V 1 provides a high performance HTTPD proxy server that is a derivative of the Apache server. The proxy has been compiled so that it can be used only as a proxy and not as a server. Functions not required to support the proxy have been removed. Other changes have been made to enhance security, and support URL content filtering. It also operates in stealth mode which means it provides transparent web access to the protected networks but can not be seen by an external host since it does not listen for connections on the external network interfaces.

### 8.1.2 On the firewall

Httpd runs on the T.Rex firewall and provides the users with secure and transparent access to the World Wide Web. The end user simply configures their web browser to go the system running the HTTP proxy. After that all access is transparent.

Httpd provides a complete log of all URL's accessed by every browser. Httpd supports caching of

frequently accessed pages. This is the preferred mode of operation as it reduces web traffic on your Internet link and reduces the average response time since more than 50% of requests will be found in the cache.

The HTTP proxy supports standard HTTP access (port 80) and HTTPS which sits on top of the Secure Sockets Layer also known as SSL (port 443). Configure the system with enough memory to support the additional workload, and enough disk to handle the additional log data, and any page caching you may want to do. In most cases it will be less expensive to upgrade the firewall system to handle the additional work than it would be to add another system to act as a proxy server behind the firewall.

### **8.1.3 Behind the firewall**

Httpd can also be run behind the firewall on one of the supported systems (AIX 4.1.4+, HP-UX 11.x, Solaris 2.5.1+). To run httpd behind the firewall still requires httpd on the firewall. Simply add the directive "ProxyRemote = xxx.xxx.xxx.xxx" to instruct the httpd proxy to connect to the httpd proxy on the firewall for Internet access. This will allow the caching and content filtering to be off loaded from the firewall to another server.

This is an improvement over the older CERN proxy that used SOCKS to access the Internet. Using two httpd proxy servers back to back is approximately 10 times more efficient than using the socksified CERN proxy.

### **8.1.4 SOCKD versus HTTPD**

Many web browsers such as NetScape have been socksified and can be configured by the end user to go directly to SOCKS daemon on the firewall. If you allow this your users can bypass the HTTPD proxy. This will eliminate the performance benefits of the caching httpd proxy, and will bypass the content filtering capabilities of httpd. Use of socks instead of httpd also means there will not be a complete log of URL requests. For these reasons we suggest that the sockd be configured to deny access to ports 80 and 443, thus forcing the use of httpd.

### **8.1.5 Malicious Java and ActiveX**

Despite assurances from SUN and Microsoft about the security of these programming and scripting languages, Java and ActiveX are inherently insecure and dangerous. Holes in the Java class loaders and the bytecode verifier can be exploited allowing malicious applets to execute arbitrary machine code. In February 1996 Princeton researchers demonstrated Java applets that exploited DNS weaknesses that forced browsers to unwittingly make network connections.

Some web browsers offer user restrictions on Java allowing Java applets and Java Script to be deactivated and not read. Java and ActiveX Scripts can insert malicious code that will skirt these

controls and run all scripts without the users knowledge.

**Hostile or Malicious Applets** are proliferating on the Internet at an alarming pace. There are several sites that distribute source code for malicious applets that can read, alter, and delete files or disrupt the normal operating of the browsing machine. Recorded hostile applets can:

- lock the browser.
- lock the browsing machine's keyboard.
- make the browser begin barking and then exit.
- open a window asking for a user name and password and then return the information to the offending machine.
- kill other applets, running programs, and defend against attempt to use other programs such as ThreadDeath to shut the hostile applet down.
- forge and send e-mail
- create a myriad of windows, calculations, noise, and run other programs and report back the results.
- disseminate not only Windows based, but also UNIX based virus at an alarming rate.

Applets run with the full permission level as the browsing user, so that if a user browses with full root access any applet it encounters will have the same access permissions.

It is possible to filter applet classes using packet filtering, but weaknesses in the Java Remote Method Invocation class loader and security manager make it possible to dynamically load hostile classes or re-classified hostile classes.

**JavaScript and ActiveX** scripts are more dangerous than applets. Scripts are harder to separate from the HTML transmission and can run before a browser can refuse them. Like applets scripts run with the full access of the browsing user. ActiveX scripts can run any application on the browsing machine and make calls to the operating systems. In February 1997 German television reported of an ActiveX control that opened Quicken, the personal finance program, and attempted to transfer funds, add transactions, and modify the application's reports.

The **IEEE Symposium on Security and Privacy** assessed Java Security as follows:

**"We conclude that the Java system in its current form can not easily be made secure. Significant redesign of the language, the byte code format, and the runtime system appear to be necessary steps toward building a higher-assurance system."**

The only reliable way to secure protected systems from the risks of Java, JavaScript and ActiveX is to block access at the firewall.

## 8.1.6 Java, JavaScript, ActiveX and Cookie blocking



T.Rex allows selective blocking of Java applets, JavaScripts, ActiveX and cookies. The HTTPD Proxy provided with T.Rex supports permit and deny commands for all these functions.

**Example 1:** Deny downloads of all Java, JavaScript, ActiveX and cookies.

```
ProxyBlockContent    java
ProxyBlockContent    scripts
ProxyBlockContent    activeX
ProxyBlockContent    cookies
```

**Example 2:** Deny access to Java from a specific location.

```
ProxyBlockContent    java    http://www.xyz.com*
```

**Example 3:** Deny Java, JavaScript, ActiveX and cookies from all systems except for a single trusted server.

```
ProxyPermitContent    java    http://www.FAS.com*
ProxyBlockContent    java
ProxyBlockContent    scripts
ProxyBlockContent    activeX
ProxyBlockContent    cookies
```

## 8.1.7 Web Content Filtering

The rapid increase in Internet usage by large organizations has raised issues of security, legality and employee productivity. Organizations are setting firm policies regarding Internet usage. In addition, school districts across the nation backed by the federal government are connecting to the Internet. Parents, teachers and administrators do not want their children exposed to inappropriate material on the Internet.

The httpd proxy solves all of these problems and more. Httpd actively monitors, and selectively blocks access to information deemed inappropriate by the organization. **Httpd reduces users network response time by caching frequently requested documents.** Information is only requested from the source when the source data has been updated or the cache data has expired. On average more than 50% of the requested data will be found in the cache. This reduces the average response time and reduces the total network traffic on the Internet connection. Frequently accessed information can automatically be cached during off hours providing users a fresh copies without delays.

**Httpd increases user productivity by filtering content and controlling access to information.**

Httpd provides flexible methods for blocking access to offensive and inappropriate information. Httpd blocks selected URLs using a high speed search engine to scan a list of objectionable material. Search times are measured in microseconds for data bases containing tens of thousands of entries. A service providing monthly data base updates is available to keep the URL lists current.

**Httpd increases network security by blocking selected content types.** JAVA, JavaScripts, and ActiveX applications can be permitted or blocked from selected or all remote sites.

The httpd proxy provides secure and transparent access to the WWW regardless of whether it is doing the content filtering on the firewall or behind the firewall.

The Httpd proxy gives the network administrator the flexibility to block sites prohibited by the organizations security policy. The Httpd Proxy comes with the ability to block sites from more than 20 different categories. The network administrator simply selects which categories should be blocked.

## 8.2 HTTPD Installation

The **install\_httpd** script automates the installation of the HTTP proxy server. The script automatically determines if you are installing on the T.Rex system or on a protected host behind the firewall.

Perform the following steps on either the firewall or a protected system behind the firewall.

1. Login or su to root.

**# cd /usr**

2. Insert the T.Rex CD
3. Mount the CD

AIX	<b># mount -r -v cdrfs /dev/cd0 /mnt</b>
HP-UX	<b># mount -r /dev/dsk/c0t2d0 /cdrom</b>
Solaris	<b># mount -r /dev/dsk/c0t2d0 /cdrom</b>

4. Change to the directory for your machine type, AIX, HP, SPARC, Intel

```
#cd aix
```

5. Change to the directory of the httpd.

```
# cd httpd
```

6. Run the install script

```
# ./install_httpd
```

After completing the software installation you are ready to configure the httpd.conf file.

## 8.3 Configuring HTTPD Proxy

### /etc/apache/httpd.conf file

The install\_httpd script automatically creates the /etc/apache/httpd.conf file shown below, and adds the user ID hermes if it doesn't exist. You can change some of the macros found in the sample file.

**AllowCONNECT**      list of ports

The AllowCONNECT directive specifies a list of port numbers to which the proxy CONNECT method may connect. Today's browsers use this method when a https connection is requested and proxy tunneling over http is in effect. By default, only the default https port (443) and the default snews port (563) are enabled. Use the AllowCONNECT directive to over ride the default ports and allow connections to the listed ports only. For example:

```
ALLOWCONNECT 443, 563, 7500
```

**CacheDefaultExpire**    *nn*

If the document is fetched via a protocol that does not support expiry times, then use nn hours as the expiry time. CacheMaxExpire does not override this setting.

Default:            1

**CacheDirLevels**      ***n***

The CacheDirlevels directive sets the number of levels of subdirectories in the cache. Cached data will be saved this many directory levels below CacheRoot.

Default:            3

**CacheForceCompletion**      **percentage**

The CacheForceCompletion directive specifies that an http transfer that has been canceled will complete the transfer to the cache if the percentage specified has already been transferred. The percentage must be a number between 0 and 100. Use of 100 will cause the cause a document to be cached only if the transfer was allowed to complete. A number between 60 and 90 is recommended.

Default:            The default value is 90 percent.

**CacheRoot**            **/home/hermes/cache**

The CacheRoot directive specifies the home directory for the cache. If you change the location of the cache don't forget to create the cache directory with hermes as owner.

Default:            The default value is no caching.

**CacheSize**            ***nnnnn***

The CacheSize directive sets the number of Kilo-Bytes (1024) used by the cache. The value 40960 will set the limit at 40 MB. During operation the size of the cache may grow larger than this limit. When Garbage Collection occurs older files will be deleted until usage is less than or equal to this value. The value used for CacheSize should be 20 to 40 percent less than the maximum space available in the caches file system.

Default:            The default size is 5 KB, which smaller than the average size of one URL!

The amount of space required for the cache can be calculated using the following formula.

$$CS = DR * (1 - PIC) * PC * ET$$

Where:

CS = Cache Size required.

DR = Download Rate

PIC = Percent In Cache

PC = Percent Cacheable

ET = Expiration Time

Example:

DR = 400,000 bits per second = 50 KBps (daily average).

PIC = 0.4 = 40%

PC = 0.6 = 60% will be added to the cache

ET = 24 hours = 86,400 seconds

$$CS = 50 * (1 - 0.4) * 0.6 * 86400 = 1,555,200 \text{ KB} = 1.6 \text{ GB}$$

In this example, a 1.6 GB cache will be required if data is to remain in the cache 24 hours. You will also have to plan for cache overflow between garbage collection intervals. Using the following example, 324MB of additional space will be required to hold the overflow. Thus 2 GB should be allocated to the cache file system. To keep files in the cache for 5 days would require approximately 8.3 GB.

## **CacheGcInterval      *nn***

The Garbage Collection interval tells httpd to check the cache every n hours and delete files if the space usage is greater than the value set by CacheSize. The time interval is expressed in hours and can have a decimal point. For example, "CacheGcInterval 1.5 " will force garbage collection every 90 minutes.

Default:              If this value is not set the cache will grow until the file system is full.

The larger CacheGcInterval is set the more space beyond the CacheSize will be needed for the cache between garbage collections. The amount of extra space required can be calculated using the following formula.

$$ES = DR * (1 - PIC) * PC * GCI$$

Where:

ES = Extra Space required.

DR = Download Rate

PIC = Percent In Cache

PC = Percent Cacheable

GCI = Garbage Collection Interval

Example:

DR = 1,000,000 bits per second = 125 Kbps (for peak hour).

PIC = 0.4 = 40%

PC = 0.6 = 60% will be added to the cache

CGI = 1 hour = 3600 seconds

$ES = 125 * (1 - .4) * 0.6 * 3600 = 162,000 \text{ KB} = 162 \text{ MB}$

In this example the cache will grow at the rate of 162 MB per hour. The data rate is representative of 10 GETs per second with an average URL size of 12.5 KB. The data rate on the Internet link will be 75 KB per second or 600 Kilobits per second which is 40% of a T1 link. If CacheGcInterval is set at 2 then 324 MB of space beyond the CacheSize will be required.

**CacheLastModifiedFactor**     *nnn*

If the origin HTTP server did not supply an expiry date for the document, then estimate one using the formula

$\text{expiry-period} = \text{time-since-last-modification} * \text{nnn}$

For example, if the document was last modified 10 hours ago, and nnn is 0.1, then the expiry period will be set to  $10 * 0.1 = 1$  hour.

If the expiry-period would be longer than that set by CacheMaxExpire, then the latter takes precedence.

Default:            0.1

**CacheMaxExpire**            *nn*

This command specifies how many hours cached documents are retained. This expiration time is enforced even if the document has a expiration date that is further in the future.

Default:           The default is 24 hours.

**ErrorLog**           ***/var/adm/httpd.error.log***

The ErrorLog directive sets the name of the file that httpd will use to log errors. If the name does not start with a slash "/" then the file location is relative to the value specified by ServerRoot. For security reasons the directory where the error logs are stored should only be writeable by root.

Default:           The default file name is logs/error.log.

**Group** **staff**

The Group directive set the groupid that httpd runs under. You may choose another group if you wish. However, avoid using "nobody".

Default:           #-1

**HostnameLookups** **on|off|double**

The HostnameLookups directive tells httpd if it should do a DNS lookup to allow recording of the client host name in the http access log. The "double" value tells httpd to do a double-reverse DNS lookup. If double is specified the hostname found in the first reverse lookup is used to find the IP address of the client. At least one of the IP addresses found in the forward lookup must match the original IP address. This option is for the paranoid. The default value is off, since it improves performance by eliminating DNS lookups of client machines. If you are running httpd on the firewall the external DNS will not resolve the internal clients. So off is the best option.

Default:           off

**Listen**           ***IP\_address:port***

The Listen directive instructs httpd to listen to one or more IP address and port numbers. Multiple Listen commands can be coded to allow httpd to listen to more than one IP address. Httpd does not listen to other IP

addresses on this system which are not coded in a Listen statement.

Default: none

For example, to make httpd accept connections on two specified interfaces and port numbers use:

```
Listen 192.168.0.1:80
```

```
Listen 10.10.0.1:8080
```

## **ListenBacklog** *nnn*

The ListenBacklog directive specifies the maximum length of the queue of pending connections. Generally no tuning is necessary here.

Default: The default value is 511.

## **LocationMatch** regex

The <LocationMatch> directive provides for access control by URL, in an identical manner to <Location>. However, it takes a regular expression as an argument instead of a simple string. For example:

```
<LocationMatch "/(extra|special)/data">
```

would match URLs that contained the substring "/extra/data" or "/special/data".

## **MaxClients** *nnn*

The MaxClients directive sets the upper limit on the number of server processes that can be run. MaxClients is designed to prevent Web browsers from consuming all the computers resources.

Default: The default value is 256.

Connection attempts that would exceed the maximum value are queued, up to the number based on the ListenBacklog directive. As soon as a process frees up a queued request will be serviced.

## **MaxRequestsPerChild** *nn*

The maximum number of requests each child process is allowed to handle



before it dies and is replaced by a new process. This increases system reliability by recovering any storage lost to memory leaks, even if the leak exists in system libraries. Busy servers should have this value set at 1000 or more to minimize system overhead.

Default:           The default value is 0 which means the child will not automatically expire.

**MaxSpareServers       nn**

The MaxSpareServers directive specifies the maximum number of *idle* processes to keep in reserve to handle transient workload spikes. An idle process is one that is not handling a request. If the number of idle processes exceeds the maximum value then the parent will kill off the excess processes.

Default:           The default value is 10.

**MinSpareServers       nn**

The MinSpareServers sets the minimum number of *idle* child processes httpd will hold in reserve to handle transient workload spikes . If there are fewer idle child processes than specified the parent process will spawn another process.

Default:           The default value is 5.

**NoCache               word|host|domain list**

The NoCache directive specifies a list of words, hosts and/or domains, separated by spaces. HTTP and non-passworded FTP documents from matched words, hosts or domains are not cached by the proxy server. The proxy module will also attempt to determine IP addresses of list items which may be hostnames during startup, and cache them for match test as well.

Example:

NoCache joes-garage.com some-host.co.uk bullwinkle.wotsamattau.edu

'bullwinkle.wotsamattau.edu' would also be matched if referenced by IP address. Note that 'wotsamattau' would also be sufficient to match 'wotsamattau.edu'. Also note that "NoCache \*" disables caching completely.

Default:           None.

**PidFile           httpd.pid**

The PidFile directive sets the name of the file that stores the Process ID of the httpd proxy. If the name does not begin with a slash "/" then the file position will be relative to the value of ServerRoot.

Default:           The default value is logs/httpd.pid.

**Port               nn**

The Port directive specifies a port number from 0 to 65535. The ServerName and Port number form the canonical address of the server. When the Listen directive is coded with a port number the Port directive does not affect which address(es) httpd listens to.

Default:           The standard port for http is 80.

**ProxyBlockContent   cookies|Java|Scripts|ActiveX**

The ProxyBlockContent directive specifies selective blocking of Java Applets, JavaScripts, ActiveX and cookies. A directive must be coded for each type you want blocked.

Default:           none.

**ProxyBlockHeader   name|ALL**

The ProxyBlockHeader directive specifies selective blocking of named request headers. Zero or more ProxyBlockHeader directives can be coded. To block User-Agent header data simply code:

ProxyBlockHeader User-Agent

You can block all non-essential request headers by coding:

ProxyBlockHeader ALL

Default: none.

**ProxyBlockList**      ***filename** [time\_range] [redirectURL/filename]*

The ProxyBlockList directive specifies the name of a file containing URLs that you want to be blocked. The filename must be fully qualified, such as "/etc/firewall/lists/sex". A ProxyBlockList command is required for each file or category you want to block. The format of the blocking list is documented in section 8.7.1

The **time\_range** that the blocking list is in effect can be specified by coding the start time and end time after the filename parameter. The start time and stop time can be specified in hours "hh" or hours and minutes. "hh:mm". The start\_time and end\_time are separated with a minus sign '-'. The times can be specified using either the 12 hour or 24 clock. For example:

8:00am-6:30pm

8:00-18:30

If no time\_range is specified then the list is always blocked.

The **redirectURL** allows you to specify the URL that the browser will be directed to when the target URL is blocked by the list. A sample redirectURL may look like this. <http://www.FAS.com/nono.html>. You can also specify a file on the firewall rather than a URL on a remote server. This file is usually placed in the /etc/apache/htdocs directory.

Default: none.

**ProxyDebugHeader**      ***on***

The ProxyDebugHeader directive specifies the logging of a message in the error\_log every time a header is blocked. This is to be used for debugging purposes only, and should not remain on during production, as it is very verbose.

Default: none.

**ProxyPermitContent**      ***cookies/Java/Scripts/ActiveX [URL]***

The ProxyPermitContent directive allows selective permitting of cookies, Java, JavaScripts or ActiveX. The URL can be a fully qualified URL or a URL containing asterisk as wildcards.

Default: none.

For example, to permit Java and JavaScripts from Motorola do the following:

```
ProxyPermitContent Java http://*.mot.cm*
ProxyPermitContent Scripts http://*.mot.cm*
```

**ProxyPermitHeader**    **name**

The ProxyPermitHeader directive allows the named header to pass when ProxyBlockHeader ALL is coded. Zero or more ProxyPermitHeader directives can be coded in conjunction with the ProxyBlockHeader ALL directive.

Default: none.

**ProxyPrivClient**    **x.x.x.x**

The ProxyPrivClient directive allows selected client IP addresses to be excluded from site blocking.

**SendBufferSize**    **nnnn**

The SendBufferSize directive sets the TCP buffer size to the number of bytes specified. This is useful to increase performance on high speed high latency networks, such as high speed transcontinental lines.

Default: The OS default is used.

**ServerRoot**    **/etc/apache**

The ServerRoot directive sets the directory where the httpd proxy's config, error and log files are kept unless their fully qualified path names are coded as shown below.

Default:           The default is /usr/local/apache.

## **StartServers**

**nn**

The StartServers directive sets the number of child server processes created on startup. The number of processes is dynamically adjusted depending on the load. There is usually little reason to adjust this parameter unless the server has a very large workload.

Default:           5

## **TransferLog   /var/adm/httpd.access.log**

The TransferLog directive sets the name of the file httpd will use to log all http access activity. If the name does not start with a slash "/" then the file position will be relative to the value of ServerRoot.

Default:           none.

## **User**

**hermes**

The User directive sets the userid that httpd will run under. The default value is hermes. If you change this value **do not** use a userid that is able to access files which are not intended for external users. Do not use "nobody" or "root". You must use a valid user ID, that has write access to the cache directory.

Default:           #-1

## **allow from**

**IP\_address/mask**

The **allow from** command permits browsers from an IP\_address to use the proxy.

The allow from command must be coded after a <Directory proxy:\*> command and before a </Directory> command.

**<Directory proxy:\*>**

```
allow from 10.0.0.0/255.255.255.0
allow from 192.168.0.0/255.255.255.0
</Directory>
```

## 8.4 Sample HTTPD Configuration for Firewall

```
#####
# file: /etc/apache/httpd.conf
#
#       Sample configuration file for httpd.
#
# Port: The port the standalone listens to. For ports < 1023, you will
#       need httpd to be run as root initially.
Port 80

# HostnameLookups:
#       Log the names of clients or just their IP numbers.
#       e.g.   www.opensourcefirewall.com (on) or 206.50.87.4 (off)
#       The default is off because it improves performance by eliminating
#       DNS lookups.
HostnameLookups off

# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.

# User/Group:
#       The name of the user/group to run httpd as.
#       The default user name is hermes. Do not use "nobody" as some
#       systems like HP-UX will not be able to use shared memory.
User hermes
Group staff

# ServerRoot:
#       The directory the server's config, error, and log files
#       are kept in
ServerRoot /etc/apache

# ErrorLog:
#       The location of the error log file. If this does not start
#       with /, ServerRoot is prepended to it.
ErrorLog /var/adm/httpd.error.log

# TransferLog:
#       The location of the transfer log file. If this does not
#       start with /, ServerRoot is prepended to it.
TransferLog /var/adm/httpd.access.log
```

```

# PidFile: The file the server should log its pid to
PidFile httpd.pid

# ServerName allows you to set a host name which is sent back to clients for
# your server if it's different than the one the program would get (i.e. use
# "www" instead of the host's real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don't understand
# this, ask your network administrator.

ServerName gw

# Server-pool size regulation. Rather than making you guess how many
# server processes you need, httpd dynamically adapts to the load it
# sees --- that is, it tries to maintain enough server processes to
# handle the current load, plus a few spare servers to handle transient
# load spikes (e.g., multiple simultaneous requests from a single
# Netscape browser).

# It does this by periodically checking how many servers are waiting
# for a request. If there are fewer than MinSpareServers, it creates
# a new spare. If there are more than MaxSpareServers, some of the
# spares die off. These values are probably OK for most sites ---
MinSpareServers 5
MaxSpareServers 10

# Number of servers to start --- should be a reasonable ballpark figure.
StartServers 5

# Limit on total number of servers running, i.e., limit on the number
# of clients who can simultaneously connect --- if this limit is ever
# reached, clients will be LOCKED OUT, so it should NOT BE SET TOO LOW.
# It is intended mainly as a brake to keep a runaway server from taking
# Unix with it as it spirals down...

MaxClients 150

# MaxRequestsPerChild: the number of requests each child process is
# allowed to process before the child dies.
# The child will exit so as to avoid problems after prolonged use when
# httpd (and maybe the libraries it uses) leak. On most systems, this
# isn't really needed, but a few (such as Solaris) do have notable leaks
# in the libraries.

MaxRequestsPerChild 30

```



```

# Content Filtering Directives.
#      Uncomment the following lines to block URLs found in the
#      desinated files.
#ProxyBlockList /etc/firewall/lists/art
#ProxyBlockList /etc/firewall/lists/banners
#ProxyBlockList /etc/firewall/lists/chat
#ProxyBlockList /etc/firewall/lists/criminal
#ProxyBlockList /etc/firewall/lists/cults
#ProxyBlockList /etc/firewall/lists/culture
#ProxyBlockList /etc/firewall/lists/dating
#ProxyBlockList /etc/firewall/lists/entertainment
#ProxyBlockList /etc/firewall/lists/gambling
#ProxyBlockList /etc/firewall/lists/games
#ProxyBlockList /etc/firewall/lists/hacker
#ProxyBlockList /etc/firewall/lists/hate
#ProxyBlockList /etc/firewall/lists/humor
#ProxyBlockList /etc/firewall/lists/investing
#ProxyBlockList /etc/firewall/lists/job_search
#ProxyBlockList /etc/firewall/lists/news
#ProxyBlockList /etc/firewall/lists/personal
#ProxyBlockList /etc/firewall/lists/sex          http://www.yourserver.com/nono.html
#ProxyBlockList /etc/firewall/lists/sports
#ProxyBlockList /etc/firewall/lists/terrorist
#ProxyBlockList /etc/firewall/lists/travel

#      Uncomment the following lines to selectively permit or block:
#      cookies, Java, JavaScripts, and ActiveX
#ProxyPermitContent cookies      http://*.microsoft.com*
#ProxyPermitContent Java        http://*.mot.com*
#ProxyPermitContent Scripts     http://*.mot.com*
#ProxyBlockContent  cookies
#ProxyBlockContent  Java
#ProxyBlockContent  Scripts
#ProxyBlockContent  ActiveX

# Cache
#      The following lines control the location, size and garbage collection
#      intervals.  If you change the CacheRoot don't forget to create the
#      cache directory with hermes as the owner.
#      The cache size is expressed in Kilobytes ( 40960 = 40 MB).
#      The gargbage collection interval is expressed in hours.
#      The Maximum Expiration time is expressed in hours.
CacheRoot /home/hermes/cache
CacheSize      40960
CacheGcInterval 4
CacheMaxExpire 48

```

```
# Listen:
#     Allows you to bind httpd to specific IP addresses and/or
#     ports, in addition to the default. [IP_address:port]
#     Change the IP address(es) and port numbers for your location.
Listen 192.168.0.1:80
Listen 10.1.0.1:80

# Directory:
#     Define the networks that are allowed to use the httpd proxy
#     server.
<Directory proxy:*>
allow from 192.168.0.0/255.255.255.0
allow from 10.1.0.0/255.255.255.0
</Directory>
```

## 8.5 Sample HTTPD Configuration for proxy behind firewall

The httpd.conf file for a proxy running behind the firewall looks just like the preceding example with the following line added.

**ProxyRemote -= httpd:// 192.168.0.1:80**

Change the IP address and port numbers to match the internal IP address of the firewall and the port number listened to by the firewall's httpd proxy.

## 8.6 Installing the WebBlocker GUI

The WebBlocker provides a Graphical User Interface (GUI) to simplify administration of the blocking lists. The WebBlocker GUI is implemented in Java. Therefore a Java environment has to be installed on the system running the WebBlocker before the GUI can be used.

Use your favorite Internet Browser to download the Java executables and libraries. If a browser is not installed on the proxy server, first download the file to another machine and then ftp the file to the proxy server.

Instructions for downloading the JAVA executables are provided for Solaris and AIX.

### 8.6.1 Solaris

If the WebBlocker is running on Solaris use this procedure to download java.

Go to URL "<http://java.sun.com/products/jdk/1.1/jre/index.html>"

Click "Select Operating System" and select the Solaris "SPARC" or "x86" version

Click "Download Software"

Note the size of the download and make sure you have enough file system space to receive it.

If the downloaded file is on another machine, ftp from the proxy server to that machine and 'get' the file over to the firewall.

cd to /usr/local/etc

execute the downloaded file which will create a jre1.1.3 directory and put the Java executables and libraries in it. E.g., assuming the downloaded file is /tmp/jre1.1.3-solaris2-sparc.bin, issue the following command:

```
sh /tmp/jre1.1.3-solaris2-sparc.bin
```

### 8.6.2 AIX 4

Go to URL "<http://ncc.hursley.ibm.com/javainfo/download/index.html>"

Click on "Please register here" and register for an access to IBM's Java Developer's area.

You will receive an email that confirms your username and password.

With that info, go to URL "<http://ncc.hursley.ibm.com/javainfo/Developer/>" and sign on with your username (your email address) and your password.

Click on "AIX JDK 1.1.2" (or a later update).

Click on Java112.tar.Z (making sure that you have enough file system space to receive the file). If you are in the US, you may want to instead click on "US mirror" to minimize download time.

If the downloaded file is on another machine, ftp from the proxy server to that machine and 'get' the file over to the proxy server.

Uncompress the download file using the uncompress command. E.g., assuming the downloaded file is /tmp/Java112.tar.Z, issue the commands

```
uncompress /tmp/Java112.tar
```

```
cd to /usr/local/etc
```

un-tar the downloaded file by issuing the command

```
tar -xvf /tmp/Java112.tar
```

## 8.7 Selecting URL Categories to Block

The T.Rex\_webblock program provides a GUI for selecting categories of information to be blocked by the WebBlocker. Simply execute the command "T.Rex\_webblock" to invoke the program. The GUI displays a list of categories that can be selected by clicking on a button. The list of categories is extendable. Files containing sites to be blocked are located in the subdirectory /etc/firewall/lists. The GUI will list each file in that subdirectory so the administrator can select the files to be used for blocking. An optional redirect-URL can be entered next to each file. If a redirect-URL is specified, attempted accesses to sites listed in the corresponding file will display the specified redirected-URL instead. If no redirect-URL is specified, attempted accesses to blocked sites will display a message that the access is denied.

When the administrator is done selecting and de-selecting files for blocking, clicking on the "OK" button will save all the changes into the httpd config file (/etc/httpd.conf). Clicking on the "Cancel" button will exit the graphical interface without saving any changes.

### 8.7.1 Adding URL lists for blocking

An administrator can build additional lists to be used for blocking. The lists are in this format:

## **site\_name ip\_address path**

where

**site\_name** is the Internet name of the site, e.g. badstuff.com. The site name is used to identify the host name as many Web servers use virtual hosts that share the same IP address.

**ip\_address** is the IP address of the site, e.g. 192.168.0.2 Note that some sites have more than one IP addresses. In order to completely block those sites, more than one entry will have to be entered for them, each with a different IP address.

**path** is the path within the site, e.g. badtopic/badfiles/bad.html If path is blank, all the files within the site will be blocked.

To add a URL list for blocking, first build the list using the above format. Give the file a name that identifies its contents. Put the file in the /etc/firewall/lists directory. When the T.Rex graphical interface is invoked after this point, the new list will appear in the graphics panel, ready to be selected.

Sites can be added or deleted from existing categories by editing the file.

FAS offers a service where we provide lists that are updated on a daily basis. As of May 15, 1999 the lists blocked tens of millions of URLs contained in more than 50,000 locations. We also provide an automated procedure for downloading updates to the blocking lists.

note: List creation is an ongoing process. We welcome your contributions.

## **8.8 Controlling FTP access via HTTP**

Downloading of ftp files via HTTP can be controlled through the use of the LocationMatch directive. This directive is very flexible and can be used many different ways:

### **8.8.1 Requiring user authentication for FTP**

LocationMatch can be used to request a user id and password. User authentication only has to be done once and will persist so long as the browser is connected the httpd proxy. The user's ID and password has to be stored in the gwuser data base for this option to work.

```
#LocationMatch:
```

```
# Require user ID and password for ftp access
```

```
<LocationMatch ftp:/*>
require valid-user
order allow,deny
allow from 10.0.0.0/255.255.0.0
</LocationMatch>
```

### 8.8.2 Blocking ftp sites except for a select few

LocationMatch can be used to block access to all ftp sites except for a select few.

```
#LocationMatch:
# Block ftp sites except for a few specified sites
<LocationMatch ftp:*>
order allow,deny
deny from all
</LocationMatch>
<LocationMatch ftp://ftp.ibm.com*|ftp.microsoft.com*>
require valid-user
order allow,deny
allow from 10.0.0.0/255.255.0.0
</LocationMatch>
```

# Chapter 9. RealAudio Proxy Administration

## 9.1 RealAudio Proxy Overview

The RealAudio Proxy (raproxy) allows users behind the T.Rex firewall to gain access to RealAudio applications. Raproxy provides access controls to permit and deny access based upon source and destination IP addresses. In order to use the Real Audio Player the firewall must be configured to run raproxy and the client application has to be configured to use the proxy.

## 9.2 Update the /etc/services file

The T.Rex installation procedure adds the following entry to the /etc/services file. The rap service name tells inetd which port to use for raproxy. You can pick another port number so long as it is not already in use. This port number must be specified to the RealAudio player on the clients computer.

```
rap    7070/tcp
```

## 9.3 Update the /etc/inetd.conf File

The raproxy runs under control of the Internet daemon (inetd) so you must uncomment the following line in the /etc/inetd.conf file.

```
rap    stream tcp nowait root /usr/local/etc/raproxy raproxy rap
```

## 9.4 Raproxy Configuration File

The raproxy program is controlled by rules defined in the **/etc/firewall/raproxy.conf** file. Connections between external and internal hosts are permitted or denied based on the **permit** and **deny** rules. In most cases the deny rule need not be coded, since the default action is to deny all requests except those which are permitted. The order in which the rules are coded is very important, since the first rule encountered that has a match with the source address and destination address is enforced.

## Command Syntax

Each rule is contained on a line that can be up to 1023 bytes long. Spaces and tabs separate fields. All fields are required except for the optional shell command on the deny rule. Optional fields are enclosed in square brackets.

## Mandatory Commands

The `gproxpath` command is mandatory and must be coded before the deny and permit rules.

### **raproxpath = pathname**

The `raproxpath` value specifies the directory that raproxy will run under.

## Optional Commands

### **timeout = nnn**

The timeout value specifies the number of seconds the raproxy program will wait for a message before it exits.

## Access Control Rules

### **deny from src\_addr to dest\_addr**

<b>from</b>	The from keyword must immediately follow the permit command. This is used to identify the new command syntax. .
<b>src_addr</b>	The source address field is required and must immediately follow the <b>from</b> keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk <b>*</b> can be used as a wild card for source IP addresses. For example, 192.168.240.* would deny all from 192.168.240.0 through 192.168.240.255
<b>to</b>	The to keyword is required and must immediately precede the <b>dest_addr</b> field.
<b>dest_addr</b>	The destination address field is required and must immediately follow the <b>to</b>



keyword.

## **permit from *src\_addr* to *dest\_addr***

<b>from</b>	The from keyword must immediately follow the permit command. This is used to identify the new command syntax. .
<b>src_addr</b>	The source address field is required and must immediately follow the <b>from</b> keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk "*" can be used as a wild card for source IP addresses. For example, 192.168.240.* would permit all from 192.168.240.0 through 192.168.240.255
<b>to</b>	The to keyword is required and must immediately precede the <b>dest_addr</b> field.
<b>dest_addr</b>	The destination address field is required and must immediately follow the <b>redirect</b> keyword. The destination address gives the IP address of the target host in the standard dot decimal format. Wildcards can not be used with destination IP addresses.
<b>dest_port</b>	The source port field is required and must immediately follow the <b>dest_addr</b> field. The dest_port can be either the port number or the name of the service found in the <b>/etc/services</b> file.

## **9.5 RealAudio Player Configuration**

The RealAudio Players behind the firewall must be configured to use the raproxy. Start the RealAudio Player and configure it as follows:

Specify:

transport use TCP, and

attempt to use TCP for all content.

#### Select Proxies

click on use proxy.

Enter the IP address and the port number for the firewall in the Proxies form.

Enter the IP address and port number of the HTTP proxy.

### 9.5.1 How to configure RealPlayer Version 6

Use the following settings on your PC.

PNA    Use the firewall's internal IP address    Port: 7070

RTSP   Don't fill it in.

HTTP   Use the firewall's internal IP address    Port: 8080

In the last box: do not use for local host...

Under "Specify transports" check : Use HTTP only.

## 9.6 Sample raproxy.conf file

The following sample configuration file is shipped with T.Rex Version 2. Edit the file to meet your unique requirements.

```
# file: /etc/firewall/raproxy.conf system: T.Rex gateway
# function: The /etc/firewall/raproxy.conf file is used to control the
# RealAudio proxy server. The raproxy program will permit or deny access to the
# proxy service based on the use of the "permit" and "deny" rules found in
# this file.
#
# (C) Livermore Software Laboratories, International 1996
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
#
raproxpath = /home/hermes
groupid = staff
userid = hermes
timeout = 300
permit from 192.168.0.* to *
deny from * to 192.168.0.*
```

# Chapter 10. RPCproxy

## 10.1 Overview

The RPCproxy is designed to provide secure access to UDP and RPC applications through the T.Rex firewall. The RPCproxy can control UDP packets so that:

- (1) only specified machines can start UDP communications with a remote server;
- (2) RPCproxy will only accept packets from a remote server if the specified client machine has started a conversation.

RPCproxy is started as a stand alone daemon at system boot time. It does not run under control of inetd. RPCproxy immediately reads its configuration file which contains a set of permit rules that govern flow of UDP packets. The commands are parsed and a set of control structures is generated in memory.

The permit command allows the system administrator to specify the following:

TCP or UDP for RPC based applications.

The IP address of the client host.

The source port on the client host.

The local IP address on the firewall that will accept connections.

The local port number or range of port numbers to which the client is allowed to connect.

The IP address and port number for the server host.

The proxy will select a free port number to uniquely identify the instance of the communication. The local port number on the firewall to be used for connecting to the server, (if this is done only one concurrent connection can be made using that port).

When the first packet from a client program is accepted by RPCproxy a set of control structures is created to manage the flow of UDP packets between the client and server. These structures are used to associate the unique combination of client: IP address, port number; firewall IP address, firewall local port1; server IP address, server port number and the firewall local port2.

Subsequent UDP packets arriving at the firewall will be examined for a match with one of the control structures generated from a permit rule. If the UDP packets source IP address and port number; destination IP address and port number matches one of the control structures then the packet will be redirected to a new destination address and port number.

For UDP applications, only UDP packets that match a permit rule can create a set of association control structures. The RPCproxy will not accept any UDP packets that do not have an associated set of structures. UDP packets coming from any system that does not match one of the associated

structures will be rejected and logged..

A UDP packet received from one of the server systems will be rejected unless it has already been sent a UDP packet from a client that matches the association.

A timeout value for an association can be specified in the configuration file. If an instance of an association has not received a packet for the duration of the timeout then the association structures will be removed.

The RPCproxy will stop UDP packets from hosts that don't meet the preceding criteria. This is about as secure as you can make UDP without encryption.

## RPC Support

RPC applications connect to the portmapper port on the firewall to discover the port number assigned to a particular function. The RPCproxy relays the portmapper request to the destination address coded in the permit rule. The portmapper on the destination host returns the RPC application port number to the RPCproxy. The RPCproxy relays this information to the client program, and begins listening to the specified port. When the RPC client receives the port number, it connects to that port on the firewall. The RPC proxy receives the connection and compares it to the rules stored in the control structures. If the connection is permitted the proxy establishes a connection to the server address and port.

The portmapper application uses UDP. However, RPCs can use either UDP or TCP/IP protocols. The permit rule allows specification of either tcrpc or udprpc based on the capabilities of the client server applications. If the RPC application can use TCP/IP then the TCPRPC option should be selected as it provides greater security. **The problem with the UDP protocol is there is no way to detect the injection of bogus packets into the data stream.** Use of TCP/IP makes this very difficult.

Portmapper is a UDP application, and is subject to the UDP access controls previously discussed.

## Special considerations

### NFS

Some NFS clients (e.g. AIX) do not use the portmapper to obtain the NFS port number.. These clients use the well known port 2049 for NFS traffic. For rpcproxy to work with these clients in UDP, do not code udprpc on the permit line for port 2049. This ensures that port 2049 is enabled without rpcproxy having to rely on a preceding portmapper request (which in this case does not exist) to enable the port. For rpcproxy to work with these clients in TCP, use genproxy to permit port 2049 traffic instead (rpcproxy still needs to be configured to allow portmapper traffic and mountd traffic).

### Maximum Open Files

Most systems limit the maximum number of open files for a single process. Usually there is a soft and a hard maximum value. Some systems allow the soft maximum to be increased up to the value of the hard limit. The rpcproxy automatically increases the maximum number of open files to the systems hard limit. This is done for the following reasons. The RPC proxy runs as a single process and uses multiple file descriptors for each transaction. Each RPC transaction requires two sockets, one for each side of the firewall. Applications such as NFS use multiple RPC transactions per connection (mountd, lockmgr, llockmgr, status, etc), resulting in as many as 10 open file descriptors per NFS request. Increasing the max open limit prevents rpcproxy from aborting a transactions due to a resource shortage. The maximum value set for AIX is 2000, and Solaris is 1000. This should be enough for most.

## Rule Order

The permit rules found in the RPCproxy configuration are scanned from top to the bottom until match is found. Once a match is found none of the remaining rules will be examined. For example, in the following case the first rule will apply and the second rule will always be ignored.

```
permit from * * to x.x.x.x 111 redirect y.y.y.y *
permit from * 192.168.0.28 to x.x.x.x 111 redirect z.z.z.z *
```

## Conflicting Rules

Some applications such as NFS require multiple permit statements, one for the portmapper, one for NFS, and another for mountd and the lockmgr. When writing the permit rules remember the permit rule must redirect all three applications to the same server. The following two rules have conflicting server addresses. The portmapper is directed to 192.168.2.9, while the NFS requests are directed to 192.168.2.21.

```
permit tcprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 111 redirect 192.168.2.9 **
permit tcpudp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 2049 redirect 192.168.2.21 **
```

## 10.2 RPCproxy Configuration File

The UDProxy is controlled by parameters defined in the `/etc/firewall/rpcproxy.conf` file.

## Command Syntax

Each command is contained on a line that can be up to 1023 characters long. Spaces, tabs and equal signs "=" separate fields. Comment lines begin with the **#** character. Blank lines are ignored. The format of each command is shown below:

## Permit Rule

There are several forms of the permit rule.. The simple form, used for UDP applications, the RPC form used for either TCP or UDP based RPC applications, TCP form and the tftp form. The simple form is identified with the leading keywords **"permit from"**. The RPC form has two flavors and is coded as **"permit tcprpc from"** or **"permit udprpc from"**. The RPC rule allows specification of a range of port numbers by adding the **hiport** parameter. This feature simplifies administration of RPC applications, by eliminating the requirement of a rule for each port number. The **hiport** number is mutually exclusive with the simple "permit from" rule.

Every simple "permit from" rule must be positioned before all the RPC rules ( "permit udprpc" or "permit tcprpc"). If a "permit from" rule is encountered after a "permit tcprpc" or a "permit udprpc" rule then an error message will be issued and the proxy will stop.

```
permit {tcprpc | udprpc| tcp | tftp | udp } from { src_addr | *} [mask ip_mask] {src_port | *} to  
      loc_addr loc_port [- hiport] redirect dest_addr {dest_port | *}  
      [using loc_addr2] { loc_port2 | *} [priv]
```

**tcprpc** indicates the permit is for a TCP port that is to be opened after a successful portmapper request from the client (*src\_addr*) returned the RPC application port number. The port is subject to timeouts.

**udprpc** indicates the permit is for a UDP port that is to be opened after a successful portmapper request from the client (*src\_addr*) returned the RPC application port. The port is subject to timeouts.

**tftp** indicates the permit is for the tftp protocol. Only outbound tftp is supported. The from *src* address can not be a wildcard **"\*"**.

**tcp** indicates the permit is for the tcp protocol. It differs from **tcprpc** in that the ports permitted by **tcp** will not expire. This command is intended for the RPC portmapper when used in **tcp** mode.

**udp** indicates the permit is for the udp protocol. It differs from **udprpc** in that the ports permitted by **udp** will not expire. This command is intended for the udp traffic. This is used to enable access to portmapper using the udp protocol.

Without 'tcp rpc' or 'udp rpc' the specified port will be opened at program startup and will not be subject to timeouts.

<b>from</b>	The <b>from</b> keyword must immediately follow the tcp rpc or udp rpc command.
<b>src_addr</b>	The source address field is required and must immediately follow the <b>from</b> keyword. The source address gives the fully qualified IP address of the requesting host in standard dot decimal format. An asterisk "*" can be used as a wild card to match any source IP addresses. The IP address and asterisk are mutually exclusive.
<b>mask</b>	The mask keyword is optional. If specified it must immediately follow the src_addr, and precede the ip_mask.
<b>ip_mask</b>	The IP mask is a optional field and must immediately follow the mask keyword. If the masking bit is one then an exact match is required. If the masking bit is zero then bit in the IP address is ignored. Specification of a source mask of 255.255.255.255 requires an exact match with the src_addr for the rule to apply. Specification of an IP mask of 0.0.0.0 permits a match no matter what source IP address is applied.
<b>src_port</b>	The source port field is required and must immediately follow the <b>src_addr</b> or the <b>ip_mask</b> if coded. An asterisk "*" can be used to permit a match with any source port.
<b>to</b>	The to keyword is required and must immediately precede the <b>loc_addr</b> field.
<b>loc_addr</b>	The local host address field is required and must immediately follow the <b>to</b> keyword. The local address gives the IP address of the firewall that is to receive the connection from the source host. This address is in the standard dot decimal format. The use of wildcards is not allowed.
<b>loc_port</b>	The local port field is required and must immediately follow the <b>loc_addr</b> field. The loc_port can be either the port number or the name of the service found in the <b>/etc/services</b> file. The use of a wild card is not allowed.
<b>hiport</b>	The high port number is optional, and allows a range of port numbers to be permitted by a single permit command. The high port can be specified by coding a minus sign "-" after the local port. The high port number must follow the minus sign. All ports between and including the loc_port and the hiport will be permitted. The high port option is not available with the simple "permit from" rule.
<b>redirect</b>	The <b>redirect</b> keyword is required and must immediately precede the <b>dest_addr</b> field.
<b>dest_addr</b>	The destination address field is required and must immediately follow the <b>redirect</b> keyword. The destination address gives the IP address of the target host in the standard dot



decimal format. Wildcards can not be used with destination IP addresses.

**dest\_port** The destination port field is required and must immediately follow the **dest\_addr** field. The **dest\_port** can be either the port number or the name of the service found in the **/etc/services** file. If an asterisk "\*" is coded the proxy will use the **loc\_port** number for the destination port.

#### using **loc\_addr2**

The **using loc\_addr2** parameter is used to specify the local IP address to be used for the connection to the destination host. This feature would normally be used if a LAN adapter has been configured to support multiple IP address, and wants to control which IP address is to be used. This feature makes it possible for the server to determine which client is requesting a connection. This parameter is optional and must follow the destination port.

**loc\_port2** The **loc\_port2** is used to specify the local port on the firewall used for the connection to the remote server. The **loc\_port2** field is required and must immediately follow the **dest\_port** field. The **loc\_port2** can be either the port number or the name of the service found in the **/etc/services** file. If an asterisk "\*" is coded then the proxy will use a system assigned port number.

**priv** The **priv** keyword is optional and is used to specify the use of a privileged port for **loc\_port2**.

#### **timeout = nnn**

The timeout value specifies the number of seconds the RPC port will stay idle before it is closed. The default is 1800 seconds (30 minutes). This value applies to RPC ports only (those that are specified with **tcprpc** or **udprpc**).

#### **qlen = nnn**

The length of the listen queues for TCP RPC ports. This is the number of requests that can be queued up for each TCP RPC port. Default is 100.

#### **Notes:**

To permit RPC applications, the portmapper (port 111) must be permitted and not subject to timeouts.

RPCproxy intercepts portmapper requests on behalf of RPC clients, and begins listening to

the port. Since the permit command allows a range of port numbers to be specified this greatly simplifies proxy administration. One permit rule can control access to more than a hundred ports. Since the proxy is not controlled by inetd the administrator does not have to code any lines in the /etc/inetd.conf file. However, needs to specify a range of RPC port numbers that does not conflict with any ports assigned to inetd.

## 10.3 Sample RPCproxy configuration file

The following sample RPCproxy configuration file is shipped with T.Rex V 1. Edit the file to suit your needs.

```
# file: /etc/firewall/rpcproxy.conf
# function: The rpcproxy.conf file is used to control the execution of the
#           rpcproxy program. RPC and UDP connections between external and internal
#           hosts are permitted or denied based on the permit and deny rules.
#
# (C) Freemont Avenue Software, Inc. 1996-2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
#
# The format of the commands are as follows:
#
# timeout nnn    the number of seconds rpc port will stay idle before it is
#                closed.
#
# permit from src_IP src_port to loc_IP loc_port redirect dest_IP dest_port loc_port2
#
# timeout = 300
# userid = hermes
#
# permit outbound tftp
#
# permit tftp from 10.10.0.20 * to 10.10.0.1 tftp redirect 192.168.0.7 * *
#
# permit udp portmapper/rpcbind
#
# permit udp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 111 redirect 192.168.2.9 * *
# permit udp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.100 111 redirect 192.168.2.100 * *
#
# permit tcp portmapper/rpcbind
```

```

#
#permit tcp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 111 redirect 192.168.2.9 * *
#permit tcp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.100 111 redirect 192.168.2.100 * *
#
# permit udp rpc applications
#
#permit udprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 2049 redirect 192.168.2.9 * *
#permit udprpc from 10.10.0.0 * to 10.10.0.100 2049 redirect 192.168.2.100 * *
#permit udprpc from * * to 10.10.0.1 33907 redirect 192.168.2.9 * *
#permit udprpc from * * to 10.10.0.100 41553 redirect 192.168.2.100 * *
#
# permit tcp rpc applications
#
#permit tcprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 2049 redirect 192.168.2.9 * *
#permit tcprpc from 10.10.0.0 * to 10.10.0.100 2049 redirect 192.168.2.100 * *
#permit tcprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 659 redirect 192.168.2.9 * *

```

### **example 1: NFS access for AIX clients**

To allow NFS access from AIX clients on network 10.10.0.x to the NFS server with the IP address 192.168.2.9 you must code the following three rules to permit: portmapper, NFS, mountd, and lockmgr. The last rule permits both mountd and lockmgr. If the proxy is running on an AIX system then you must code `priv` at the end of the line to allow access to port numbers below 1024.

```
timeout = 300
userid = hermes
permit udp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 111 redirect 192.168.2.9 *
permit udp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 2049 redirect 192.168.2.9 *
permit udprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 600-1200 redirect 192.168.2.9 * * priv
```

### **example 2: NFS access for other clients**

To allow NFS access from other clients (not AIX) on network 10.10.0.x to the NFS server with the IP address 192.168.2.9 you must code the following four rules to permit: portmapper, tcp based NFS, udp based NFS, mountd, and lockmgr. The last rule permits both mountd and lockmgr.

```
timeout = 300
userid = hermes
permit udp from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 111 redirect 192.168.2.9 *
permit tcprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 2049 redirect 192.168.2.9 *
permit udprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 2049 redirect 192.168.2.9 *
permit udprpc from 10.10.0.0 mask 255.255.255.0 * to 10.10.0.1 31000-45000 redirect 192.168.2.9 * *
```

### **Note:**

If you have coded a `udprpc` permit for NFS and a non-AIX client is already using NFS then any request by a AIX client for NFS services will fail with a message indicating that port 2049 is already in use. If this should occur then you will have to use the `udp` form of the permit rule shown in example 1.

# Chapter 11. Secure Mail Wrapper

## 11.1 Secure Mail Wrapper Overview

The T.Rex Sendmail Wrapper is a set of object oriented programs that provide a secure SMTP interface between a protected network and an unprotected network. It consists of a smwrap daemon that communicates with remote hosts using SMTP, a smwrapd daemon that manages e-mail deliveries, an aliases data base builder and several reporting programs.

### smwrap

The smwrap program is a small secure application designed to receive mail from protected and unprotected hosts. It runs in a non-privileged state in a chrooted directory. This significantly reduces the risks associated with running large privileged programs such as sendmail on a system that has direct contact with unprotected networks.

smwrap is invoked by the Internet daemon (inetd) when another system tries to send mail to the T.Rex firewall. The smwrap program stores the e-mail in a unique file in its spool directory. The SMTP envelope is stored at the beginning of the file. The smwrap program creates the store and foreword e-mail files with a name prefix of "smwrap" followed by a string of random characters. When the entire file is complete smwrap closes the file and changes its name so that the leading character is a capital "S". When the session with the SMTP program on the remote system is complete the smwrap program exits. The smwrapd program periodically scans the directory for file names beginning with the characters "Smwrap". Thus smwrapd will only process complete files. Should the system fail in the middle of receiving a file the smwrapd will not attempt to deliver an incomplete file.

**Detect, Report and Repel crackers using e-mail:** smwrap is designed to detect and report cracker activity using SMTP. With the wrapper in place a cracker can not use sendmail to access or modify data on your protected hosts. The VRFY command will not provide the cracker with a list of valid user IDs, but echoes every name the user tries to verify. Attempts to use this SMTP sub-command are logged. The wrapper also detects and reports spoofed host names and addresses and prohibits communications with such systems. The systems administrator is alerted whenever mail is received that is directed to a file instead of a user or when parameters are being passed to sendmail.

The use of e-mail is controlled using parameters stored in a smwrap control file. Parameters include maximum size and number of recipients. Specifying the maximum size of a e-mail document makes it more difficult for a hacker to flood your disk with large files. Placing an upper limit on the number of recipients for each piece of mail limits the effectiveness of "denial-of-service" attacks. The wrapper can also be directed to reject mail from specific hosts or users.

**Mail Blocking:** T.Rex prevents annoying e-mail messages, commonly called "SPAM" from entering protected networks. The feature also blocks harassing messages making "Cyber-stalking" more

difficult. The Administrators can enter a list of senders, addresses, sites, or domains they want to target for blocking. Like Call Blocking on your telephone T.Rex allows you to choose who you want to get e-mail from. Blocked e-mail can be deleted, sequestered or redirected to a specified recipient. If the mail is sequestered or redirected it can be kept as evidence along with the log information.

**Information hiding:** smwrap hides internal network information by automatically translating e-mail addresses and by scrubbing selected header information from outbound mail. Smwrap uses high speed table lookups to translate external e-mail addresses to internal addresses. A reverse translation is made for outbound mail. When outbound mail is directed to multiple users using "Cc:" each internal e-mail address is translated to an external name.

**Multiple domain support:** The aliases data base used by smwrap supports e-mail for multiple domains. This can be done using several different options. For example, one can explicitly code each users external address and their associated internal address. This method would be used when mail for a single domain is to be delivered to more than one internal system. If all the mail for a domain is to be delivered to a single mail server then it is possible to specify a single entry that will route all mail for a domain to a single server.

**Other SMTP mail servers:** smwrap works with other SMTP compatible mail servers such as sendmail, HP's Open Mail, Microsoft Exchange Server Internet Mail Connector, Microsoft Outlook, and others.

## smwrapd

The smwrapd daemon is started when the system boots and runs continuously. Its behavior is controlled by parameters defined in the sendmail wrapper configuration file. The program periodically scans the e-mail spool directory looking for mail to deliver. When it finds mail waiting for delivery it passes the file to a mail delivery program. This can be the sendmail program. Sendmail can be safely used this way since it is not receiving commands from another system. When all the files in the spool directory have been passed to the mailer the smwrapd takes a nap for the duration specified by the wakeup parameter.

The smwrapd performs multiple validity checks before passing the file on to the mailer. If the file is not in the proper format or has suspicious characteristics it is sequestered from the valid mail and a message is logged. Sequestered mail is stored in a special directory for later review by the systems administrator.

## adam

The adam program builds the aliases data bases used by the smwrap program to translate external e-mail addresses to internal e-mail addresses. It also builds the reverse aliases data base to translate internal addresses to the external addresses. This eliminates the need for the /etc/aliases file used by sendmail, and also eliminates the requirement for writing complex sendmail rules to translate names and addresses.

High levels of performance are maintained using sophisticated hashing algorithms to perform the table searches. Even if the table contains tens of thousands of entries a table search normally requires only one or two disk I/Os.

Using this data base the sendmail wrapper is able to accept e-mail for multiple domain names. This provides much more flexibility than the standard sendmail and /etc/aliases files.

## **smrpt**

The sendmail report program (smrpt) reads the system log and produces reports showing how e-mail is being used. It produces five reports showing the overall usage of e-mail, the top users sorted by the number of messages received, bytes received, messages sent and bytes sent. Parameters are available to control the amount of detail reported.

## **smrptx**

The sendmail exception report program (smrptx) reads the system log and produces a set of reports.

**VRFY:** All attempts to use the VRFY command to obtain user IDs on the firewall are reported. Excessive use of the VRFY command from an unprotected host can be the sign of a cracker.

**EXPN:** All attempts to use the EXPN command to extract user information are reported.

**Warning:** All smwrap and smwrapd warning messages are printed. This helps find configuration and systems problems.

**Error:** All the Error messages issued by either smwrap or smwrapd are reported.

**Security Alert:** All the Security Alerts issued by smwrap or smwrapd.

**Sequestered:** All the files that can not be delivered and the reason they were sequestered are reported.

## **11.2 Secure Mail Wrapper Configuration File**



The Sendmail Wrapper client and daemon programs are controlled by parameters defined in the **/etc/firewall/smwrap.conf** file.

## Command Syntax

Each command is contained on a line that can be up to 1023 characters long. Spaces, tabs and equal signs "=" separate fields. Comment lines begin with the **#** character. Blank lines are ignored. The format of each command is shown below:

### **aliasreq**

The aliasreq command tells smwrap to only allow outbound mail for users that are defined in the alias data base created with the adam utility. If this command is specified and a user tries to send mail through the firewall the mail will be rejected and a Security Alert will be issued. This command is mutually exclusive with the noaliases command.

### **deny from *src\_addr* to *rcpt\_addr* [block|forward|sequester]**

The deny rule is used to block mail from a specified e-mail address to a specified recipient.

<b>from</b>	The from keyword must immediately follow the deny command.
<b>src_addr</b>	The source e-mail address is required and must immediately follow the from keyword. The source address can have the form <i>userid@host.domain</i> , or <i>@host.domain</i> , <i>@ domain</i> , or <i>*</i> .
<b>to</b>	The to keyword must immediately follow the <i>src_addr</i> .
<b>rcpt_addr</b>	The recipient e-mail address is required and must immediately follow the to keyword. The recipient address can have the form <i>userid@host.domain</i> , or <i>@host.domain</i> , <i>@ domain</i> , or <i>*</i> .
<b>block</b>	The disposition of the blocked mail is specified by one of three mutually exclusive options: block, forward and sequester. The default disposition is block. In this case a Security Alert is issued and the mail is rejected. Nothing is stored on the disk.
<b>forward</b>	If forward is coded the mail will be forwarded to the specified e-mail address coded immediately after the forward keyword. A Security alert will also be issued.
<b>sequester</b>	If sequester is coded the blocked e-mail will be stored in the directory

specified by the undelivpath. A Security Alert will also be issued.

**groupid = *name***

The groupid value is used to specify the group name the smwrap and smwrapd programs should run under.

**maxchildren = nn**

The maxchildren value specifies the maximum number of processes that can be spun off for processing the spool directory. This allows control over the level of concurrence. Too small a value may slow down mail delivery, too large a value may cause the system to thrash. Values between 5 and 10 are reasonable. A value larger than 12 will produce a Warning message.

**maxbytes = nnnnnnnn**

The maxbytes value specifies the maximum size of a single piece of mail. A value of 0 will produce an error message. A value larger than 10000000 will produce a warning message.

**maxreceipts = nnn**

The maxreceipts value specifies the maximum number of recipients permitted for a single piece of mail. This is used to control denial-of-service attacks that use large numbers of recipients to keep sendmail busy.

**noaliases po = host.name**

The noaliases command is used if you do not want smwrap to translate e-mail addresses using the aliases data base. This command directs inbound mail to the internal mail server specified by the post office parameter "po". This command is mutually exclusive with aliasreq.

**smtppath = pathname**

The smtppath value specifies the fully qualified path name of the program that will be delivered the mail. This will most likely be /usr/lib/sendmail.

**smwpath = directory**

The smwppath value specifies the directory name of the sendmail wrapper daemon smwrapd.

#### **spoolpath = pathname**

The spoolpath value specifies the directory that will store the mail files. Both the smwrap and smwrapd programs must have write permissions to this directory. To enhance system security smwrap changes its root directory to the value found in spoolpath. This prevents smwrap from addressing the rest of the file system on the firewall. As a result, smwrap does not permit the use of "/", "/dev", "/etc", "/lib", "/sbin", "/unix", "/usr", "/var", etc. Specification of these prohibited directories will produce an error message and terminate program execution.

#### **undelivpath = pathname**

The undelivpath value specifies the directory that will store sequestered mail. This is mail that can not be delivered. The reason can be found by running the smrptx program.

#### **userid = name**

The userid value is used to specify the user name the smwrap and smwrapd programs should run under. This user ID must have write access to the spool directory and the sequestered mail directory. The smwrap and smwrapd programs will not allow the userid to be root or 0.

The smwrapd passes mail to the sendmail program ( or other program of choice) for distribution. It tells sendmail whom the message is**from**. Unless you tell sendmail the smwrap userid is trusted, sendmail will use the smwrap userid instead of the desired from address. See chapter 13 for details on**trusted user** specification.

#### **timeout nnnn**

The timeout value specifies the number of seconds the smwrap program will wait for a message before it exits.

#### **wakeup nnn**

The wakeup value specifies the number of seconds the smwrapd will sleep after scanning the e-mail spool directory.

## **11.2 Sample Secure Mail Wrapper Configuration File**

The following sample sendmail wrapper configuration file is shipped with T.Rex version 1.

```
# file: /etc/firewall/smwrap.conf
# function:      The smwrap.conf file is used to control the execution of the
#                smwrap and smwrapd programs.
#
# (C) Freemont Avenue, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU.  SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
timeout    = 600
userid     = hermes
groupid    = staff
spoolpath  = /home/hermes
maxbytes   = 1500000
maxreceipts = 100
maxchildren = 10
smwpdpath  = /usr/local/etc
smtpdpath  = /usr/lib/sendmail
undelivpath = /home/hermes/sequestered
wakeup     = 60
```

## 11.3 Sample e-mail Blocking

Mail Blocking is activated by adding deny statements to the end of the smwrap.conf file. The following example shows blocking of in-bound mail from selected sites and specified users at specific sites. For example, all mail from the domains: 18576.com, bigfoot.com, hotmail.com and savetrees.com will be blocked. Only users email4all and runnersssss will be blocked from aol.com.

```
# sample e-mail blocking
deny from @18576.com          to *
deny from email4all@aol.com    to *
deny from runnersssss@aol.com  to *
deny from @bigfoot.com         to *
deny from @hotmail.com         to *
deny from @infoblasterstvc.com to *
deny from @mailr.com           to *
deny from @mailrod.com         to *
deny from @powertips.com       to *
deny from @savetrees.com       to *
deny from ballman@t-1net.com    to *
deny from 800@tollfree.net     to *
deny from @webspecials.com     to *
```

## 11.4 Secure Mail Wrapper Aliases File

The Sendmail Wrapper Aliases file is used by the adam utility to build the aliases and reverse aliases Data Bases. This file must have the following path name: **/etc/firewall/aliases**.

### Translation Table Syntax

**Defining entries for each user:** Each line in the table contains two entries. The first entry is the external mail address. The second entry is the internal mail address. Each line in the table can be up to 511 characters long. However, the maximum length of an address or its aliases is 200 bytes each. Spaces, tabs and colons ":" separate the two addresses. Comment lines begin with the "#" character. Blank lines are ignored. The format of each entry is shown below:

```
name@host.domain
name@host
name
```

**Defining domain mail server(s):** Mail servers can be defined for a domain using character strings that begin with the '@' character followed by the domain name. When using commands that begin with the '@' character the user name is not translated. Only the character strings to the right of the '@' character are exchanged. The format is shown below:

@domain.name @mail.server

## Sample Aliases File

The following sample aliases file can be used to create the aliases and reverse alias data bases for a firewall supporting a single domain. This sendmail wrapper aliases file is shipped with T.Rex Version 1.

```
# file:      /etc/firewall/aliases      system:  gw.your.domain
# function:
# created:by rjl@lsli.com      4/03/94
#
#
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1994 - 2000
# All Rights Reserved
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS, FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
#
# NOTE:
#      /usr/local/etc/adam -b must be run after updating the
#      aliases file. There will be a delay while the command updates
#      the aliases data bases. When the command will print a
#      summary of what it did.
#
#      Replace root@system2 with the internal e-mail address of the
```

```

#           systems administrator who should receive mail for
#           MAILER-DAEMON, postmaster, and root. In this example
#           the entry for root should appear before MAILER-DAEMON and
#           postmaster.
#
# Aliases for root, mailer-daemon, and postmaster
root:       root@system2
MAILER-DAEMON: root@system2
postmaster: root@system2

# Aliases to handle mail to msgs and news
nobody:    /dev/null

# Replace the following examples with your own external and internal names
# external_addr: internal_addr
# Aliases to handle mail to msgs and news
nobody:    /dev/null
# Aliases in the form of first initial & last name
bing:      bing@crosby
bud:       bud@abbot
buster:    buster@keaton
charlie:   charlie@chaplin
chico:     chico@marx
curly:     curly@howard
ernie:     ernie@kovaks
george:    george@burns
gracie:    gracie@allen
groucho:   groucho@marx
harpo:     harpo@marx
harld:     harld@lloyd
john:      john@bulushi
jackie:    jackie@gleason
larry:     larry@fine
lou:       lou@costello
lucy:      lucy@ball
may:       may@west
moe:       moe@howard
shemp:     shemp@howard
spanky:    spanky@mcfarland
wc:        wc@fields
zeppo:     zeppo@marx

```

## Multiple Domain Aliases File

The following aliases file can be used to create the aliases and reverse alias data bases for a firewall supporting a multiple domains.

```
# file:      /etc/firewall/aliases
# function:   The aliases file is used to control the translation of
#            external to internal addresses as well as internal to external #
#            addresses.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU.  SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# External Address Internal Address

user1@domain1.com      abc001@host1
user2@domain1.com      abc002@host1
user3@domain2.com      def002@host2
user4@domain3.com      ghi123@host17
...
```

To support this example MX records would be used to direct mail for domains domain1.com, domain2.com, and domain3.com to the T.Rex firewall. The smwrap program will use the aliases data base to translate the external e-mail address to the internal e-mail address. Thus, mail sent to user2@domain1.com would be directed to abc002@host1. Internal mail heading for the outside would undergo the reverse translation.

### Duplicate names are not allowed :

The aliases data base does not support the use the duplicate addresses.



However, the user names can be the same so long as their domain names are different or their host names are different. The following example is valid.

user1 @domain1	abc001 @host1
user1 @domain2	xyz001 @host5

### Invalid Example

The following is not valid because the reverse aliases data base would not support the translation of abc001 @host1 to both user1 @domain1 and user101 @domain3.

user1 @domain1	abc001 @host1
user101 @domain3	abc001 @host1

## Multiple Domain Servers - Aliases File

The following aliases file can be used to create the aliases and reverse aliases data bases for a firewall supporting a multiple internal mail servers, each of which supports a single domain.

```
# file:      /etc/firewall/aliases
# function:   The aliases file is used to control the translation of
#            external to internal addresses as well as internal to external #
#            addresses.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU.  SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
```

```
#
# domain name          Server name

@domain1.com          @host1.domain1.com
@domain1.com          @domain1.com
@domain2.com          @host2.domain2.com
@domain2.com          @domain2.com
@domain3.com          @host3.domain3.com
@domain3.com          @domain3.com
@domain4.com          @host4.domain4.com
@domain4.com          @domain4.com
...
```

In the preceding example each domain has two lines. The first line is used to send in-bound mail to the mail server using its fully qualified host name. All out-bound mail with a "From:" address containing the fully qualified host name of the server will have its "From:" address translated to the domain name found in the first column. The second line generates a second entry in the out-bound translation table. This is used if the mail server sends mail to the firewall with the external e-mail address. In this case the "From:" address will remain unchanged.

## 11.5 Aliases Data Base Administration

The aliases and reverse aliases data bases are created by running the Aliases Database Administrator utility **adam**. After creating the aliases file in the `/etc/T.Rex` directory simply run the adam program as shown below. The adam utility will read the `/etc/firewall/smwrap.conf` file to determine where to place the smwrap aliases data bases. They must be located in the sendmail wrapper spool directory since the smwrap program will change its root directory to the same directory and will be unable to read the data bases if they are located elsewhere.

The adam utility and its possible parameters are listed below.

**adam [-a] [-b] [-da name|filename] [-dr name|filename] [-h] [-p [all|aliases|rev] ]**

- a** This is used to add new records using the `/etc/firewall/newaliases` file as input. This option is used to add additional records to an existing aliases data base that was previously built using the **-b** option. The format of the newaliases file is exactly the same as the aliases file.
- b** This is used to build the aliases and reverse aliases data bases using the `/etc/firewall/aliases` file as input. .

**-da name|filename**

This option is used to delete aliases from the aliases data base. You can delete a single alias entry by providing its name. You can also delete a list of aliases contained in a file by providing the fully qualified file name after the `-da`. For example, "`-da /etc/firewall/delete.aliases`" would cause adam to delete all the aliases found in the file `/etc/firewall/delete.aliases`. The file has a simple format of one name per line.

**-dr name|filename**

This option is used to delete aliases from the aliases data base. You can delete a single alias entry by providing its name. You can also delete a list of aliases contained in a file by providing the fully qualified file name after the `-da`. For example, "`-da /etc/firewall/delete.aliases`" would cause adam to delete all the aliases found in the file `/etc/firewall/delete.aliases`. The file has a simple format of one name per line.

**-p [all|aliases|rev]**

This option is used to print the aliases data base, the reverse aliases data base or both. Since the records are not stored in alphabetical order they are sorted by the adam utility. The current implementation supports the printing of data base with up to 10,000 entries. If there are more than 10,000 entries this function will not work.

**-h** print usage information.

## 11.5.1 Creating the Aliases Data Bases

To create the aliases and reverse aliases data bases using the `/etc/firewall/aliases` file simply key in the following command. The adam utility will print a summary report for the build of the aliases data base and the build of the reverse aliases data base. If the utility finds any duplicate keys while building the reverse aliases data base the duplicates will be ignored and a listing of the duplicates will precede the last summary report.

The sample output was generated on a T.Rex Entry Level System (ELS) that is restricted to a maximum of 50 users. The ELS version was used to show the how adam reports the number of available user IDs.

## **# /usr/local/etc/adam -b**

Aliases Database Administrator (adam) V 2.2

Summary of Build for aliasesDB

```
input file      = /etc/firewall/aliases
data base name  = /home/hermes/aliasesDB
number of lines read from input file = 60
number of records stored in DB      = 28
maximum number authorized            = 50
number of unused records available   = 22
largest key size                     = 13
average key size                     = 5.3
total key size                       = 149
largest record size                  = 16
average record size                  = 11.2
total record size                     = 314
```

Starting to build Reverse Aliases Data Base.

```
Duplicate record ignored, key = robert@hope,
                           data = mailer-daemon,
                           record number = 51, input line = 59.
Duplicate record ignored, key = robert@hope,
                           data = root,
                           record number = 51, input line = 60.
```

...

Summary of Build for raliasesDB

```
input file      = /etc/firewall/aliases
data base name  = /home/hermes/raliasesDB
number of lines read from input file = 60
number of records stored in Rev DB    = 51
number of unique records stored in DB = 26
```

maximum number authorized	= 50
number of unused records available	= 24
number of duplicates rejected	= 2
largest key size	= 16
average key size	= 8.1
total key size	= 411
largest record size	= 10
average record size	= 5.1
total record size	= 258

Elapsed time = 0 seconds.

## 11.5.2 Adding new entries to the Aliases Data Bases

The following example shows the addition of new aliases to the aliases and reverse aliases data bases. In this example the T.Rex ELS is restricted to a maximum of 50 users. Before the update there were 23 users in the aliases DB. When the request to add the twentyeight new record was made it was rejected and an Error message was printed.

```
.      # /adam -a
```

```
Aliases Database Administrator (adam) V 2.2
```

```
Updating Aliases Data Base.
```

```
Error: The record 'newuser28@lsli.com newuser28@marx.lsli.com'
      on line 23 of /etc/firewall/newaliases was not added as the maximum
      number of allowed users (50) has been reached.
```

```
Summary of Aliases DB update
```

```
input file           = /etc/firewall/newaliases
data base name       = /u/hermes/aliasesDB
number of lines read from input file = 23
beginning number of records in DB   = 28
number of records added to DB       = 22
total number of records in DB       = 50
maximum number authorized            = 50
number of unused records available   = 0
largest key size                    = 18
average key size                    = 18.4
total key size                      = 405
largest record size                 = 23
average record size                 = 23.6
total record size                   = 520
```

```
Updating Reverse Aliases Data Base.
```

```
Summary of Reverse Aliases DB update
```

```
input file           = /etc/firewall/newaliases
```

```

data base name = /u/hermes/raliasesDB
number of lines read from input file      = 28
beginning number of unique records in DB = 26
number of records added to DB             = 56
number of unique records added to DB      = 28
total number of unique records in DB      = 50
maximum number authorized                  = 50
number of unused records available         = 0
number of duplicates rejected              = 0
largest key size                           = 23
average key size                           = 27.0
total key size                             = 1513
largest record size                        = 18
average record size                        = 26.5
total record size                          = 1485

```

Elapsed time = 0 seconds.

### 11.5.3 Deleting an entry from the Aliases Data Base

The following example shows the deletion of a single name from the aliases data bases. Deleting a name from the Aliases DB does not delete the name from the reverse Aliases DB.

```
. # /adam -da jim
```

Aliases Database AdMministrator (adam)

Record deleted. key = jim.

Elapsed time = 0 seconds.

## 11.5.4 Deleting an entry from the Reverse Aliases Data Base

The following example shows the deletion of a single name from the aliases data bases.

```
.      # /adam -dr jim@system2
```

Aliases Database AdMministrator (adam)

Record deleted. key = jim@system2.

Elapsed time = 0 seconds.

## 11.6 Listing Aliases and Reverse Aliases Data Bases

```
# adam -p all
```

Aliases Database AdMministrator (adam)

Aliases Data Base

Data Base Name: /home/hermes/aliasesDB.

```
record_key  data
```

```
MAILER-DAEMON jim@system2
      alex alex@system2
      clivermore charles@system2
      elivermore ellana@system2
      ellana ellana@system2
      firstcat alex@system2
      fletch fletch@system2
      hermes ellana@system2
```



```

        jay jay@system2
        jim jim@system2
    jlivermore jim@system2
jlivermore@gw.lsl.i.com jim@system2
        jlyall jay@system2
    nobody /dev/null
        T.Rex jim@system2
    T.Rexinfo ellana@system2
    postmaster jim@system2
        root jim@system2
        suds jay@system2
        ted ted@system2

```

Number of records read = 20

Reverse Aliases Data Base

Data Base Name: /home/hermes/raliasesDB.

```

record_key data

    /dev/null nobody
        alex alex
    alex@system2 alex
        charles clivermore
    charles@system2 clivermore
        ellana elivermore
    ellana@system2 T.Rexinfo
    ellana@system2 elivermore
    ellanaTLivermore ellanabana
        fletch fletch
    fletch@system2 fletch
        jay jay
    jay@system2 jay
    jay@system2 suds
        jim jlivermore
    jim@system2 jlivermore
        ted ted
    ted@system2 ted

```

Number of records read = 20

Elapsed time = 0 seconds.

# Chapter 12. Mail Setup

## 12.1 Overview

The T.Rex Firewall System can also be setup to act as a mail gateway. There are several advantages to using the T.Rex Firewall System as the electronic post office. Outside users only receive mail from the gateway so they do not see internal host names, thus enhancing the network security. You can use aliases on your user accounts so that mail coming from the outside can be addressed to a standard name such as **elvis@rroll.com**. The gateway will translate the external name and address to the internal user id and address, such as **ghost@gracelnd.rroll.com**. Since the outside world does not see the internal name and address a company employee can change departments without impacting the external mail. Only the aliases file on the gateway is updated. If the user changes organizations the gateway can still route the mail, and there is no need to retain the users old account just to enable use of the **forward** file.

The systems administrator should be familiar with configuring the mail. If you are not then I would recommend the books referenced in the bibliography.

T.Rex provides a sendmail wrapper that enhances security and increases system flexibility. The sendmail wrapper receives mail from remote hosts and places it in the sendmail wrapper directory. The sendmail wrapper daemon periodically wakes up to see if there is any mail to pass on to sendmail for delivery to a remote system. Since the sendmail wrapper uses it's own aliases data base to translate external e-mail address to internal e-mail addresses sendmail should not use the /etc/aliases data base. The sendmail wrapper also translates internal e-mail addresses to external e-mail names. Thus you do not have to code special sendmail rules in the sendmail.cf file to rewrite header addresses. See chapter 8 for more information regarding the sendmail wrapper and aliases name processing.

If there is an /etc/aliases file and the newaliases command has been run to create the /etc/aliasesDB/DB.dir and DB.pag data bases then you should remove the Data Base files to prevent sendmail from doing its own translations.

## 12.2 Sendmail Configuration File Changes

The following changes are made to the /etc/sendmail.cf file by the automated install process.

- (1) Specify the use of **MX** records.
- (2) Remove the sendmail and OS release and revision numbers from the received by header.
- (3) Specify the smwrap userid as a trusted user with the **T** macro.

## 12.2.1 Additional Changes for Sendmail 8.6+

If you are running sendmail 8.6+ then you should activate the following Option in the /etc/sendmail.cf file.

**O TryNullMXList**

Failure to do so will cause the following problems. If out-bound mail can not be delivered to the external destination after repeated retries, it will be sent to the postmaster rather than returned to the sender.

This change should be implemented on AIX 4.2+ and Solaris 2.5.1+, since they run Sendmail 8.6.

To avoid mail loop problems when mail was sent to a bogus address add the following macro to the sendmail.cf file. Substitute your own domain name.

**Cwdomain.name**

## 12.3 Changes to sendmail configuration files of protected hosts

In order to route outgoing mail through the T.Rex secure gateway each **internal** UNIX system running sendmail will require the following changes to their /etc/sendmail.cf file.

### Sendmail 8.7+

For newer versions of sendmail use the DS macro to specify the relay host to forward all outgoing mail not in the local domain. Substitute the name of your gateway for gw.isli.com.

**DSsmtp:gw.isli.com**

### Older versions of Sendmail.

1. Specify the relay mailer definition to use TCP.

**DMtcp**

2. Specify the firewall's name as your Internet Relay as follows. Substitute the name of your gateway for gw.isli.com.

**DRgw.isli.com**



notes:

...

# Chapter 13. Telnet Proxy Administration

## 13.1 Telnet Proxy Overview

The telnet proxy (tnproxy) provides telnet access through the T.Rex firewall without compromising the security and integrity of the firewall. Tnproxy supports access control by user ID, source IP-address and destination IP-addresses. Strong user authentication is required for unprotected hosts and is optional for protected hosts. Tnproxy also supports transparent Internet access for internal users. If transparent access is specified then internal users are not prompted for user IDs or passwords at the firewall. All successful and unsuccessful connection attempts are logged, including times, source and destination IP-addresses, and the number of bytes sent and received. Utilities are provided which summarize tnproxy activity.

The tnproxy program runs in a non-privileged state chrooted to a restricted directory. The only file I/O it performs is to **read** configuration files and to log user activity via the syslog daemon. The user is unable to issue any commands or shells that run on the firewall. The user is not logged on to the firewall. Commands and data simply pass through the firewall. Standard telnet is also available to allow systems administrators to login to the firewall to perform administrative functions.

Access control is granted based on permit and deny rules. Control is based on host IP-address and optionally user-id or group-id.

Tnproxy is invoked by the Internet daemon **inetd** when a telnet client on another host wants to telnet to another system through the T.Rex firewall. The tnproxy reads the **/etc/firewall/securenets** file to determine if the source host is on a protected network. Tnproxy asks for the user's id. It then reads the **/etc/firewall/gwuser** file to determine if the user has telnet privileges at the current time and what form of user authentication to be used. The user is prompted for a password. If the user fails authentication or is denied telnet access a message is sent, the connection is broken and the attempt is logged. After passing the authentication test the user is prompted for the desired destination. Tnproxy reads the **/etc/firewall/tnproxy.conf** file to determine if the connection is permitted or denied. If permitted the connection is made and logged. If denied a message is sent to the user, the connection is broken and the error is logged.

## 13.2 X-Window Forwarding

The tnproxy provides controlled user access to X-Window applications through the T.Rex firewall. X-window applications can be run on any client host with the display sent to a virtual terminal on the firewall. When the xforward program receives the connection request it opens a window on the user's display requesting permission for the connection. If permission is granted by the user then the xforward program passes data between the virtual display on the firewall and the user's real display. If permission is denied the connection is broken.

The X-forwarding connection is established as follows. After the user has been authenticated by tnproxy they can issue the xforward command. Then the user connects to the system where the X-client program resides. After logging on to the X-client system the user executes the program with the **-display firewall\_name:nn.0** parameters. The value to be used for **nn** is displayed in a pop up control window on the user's display. It is normally 10 or higher. The client program then displays its output on the virtual display at the firewall. When the client program connects to the xforward program xforward then pops up a window on the user's display requesting permission for the connection. If the user grants permission then xforward passes data between the virtual display on the firewall and the user's real display.

The xforward program runs as a child process of the tnproxy. The X-Window session can be broken by either end. The user can break the X-window session by clicking on the EXIT button found in the xforward control window. The user can also kill the X-client application using the tnproxy session.

To allow use of automated telnet scripts that use xforwarding a "port" argument can be added to the xforward command. To use this feature type port = nn after the xforward command.

xforward -port nn

Where nn is an integer from 10 to 99.

If port is not specified xforward will start with port 10 and will increment and retry a higher port number if one is in use.

## 13.3 Telnet Proxy Configuration File

The tnproxy program is controlled by rules defined in the `/etc/firewall/tnproxy.conf` file. Connections between source and destination hosts are permitted or denied based upon **permit** and **deny** rules. The order in which rules are coded is very important, since the first rule that matches is enforced. Subsequent rules are ignored. There are four mandatory commands which must be coded before the access control rules. These commands govern the general behavior of the tnproxy. There must be one or more access control rules following the mandatory commands.

### Command Syntax

Each rule is contained on a line that can be up to 1023 characters long. Spaces and tabs and equal signs "=" separate fields. Comment lines begin with the "#" character. Blank lines are ignored. The format of each command is shown below:

### Mandatory Commands

The groupid, tnproxpath, userid, and timeout commands are mandatory and must be coded before the deny and permit rules.

**groupid = name**

The groupid value is used to specify the group name the tnproxy will run under.

**tnproxpath = pathname**

The tnproxpath value specifies the directory that tnproxy will run under.

**userid = name**

The userid value is used to specify the user name the tnproxy program will run under. The tnproxy program will not allow the userid to be root or 0.

**timeout nnnn**



The timeout value specifies the number of seconds the tnproxy program will wait for a message before it exits.

## Optional Commands

### **auth = yes/no**

The auth command is used to specify whether an internal user has to supply a userid and password to use the tnproxy. The auth command is optional. If auth is not coded in the tnproxy.conf file the proxy uses yes as the default value. If auth = no is specified the tnproxy uses the IP address of the users host in place of the users ID. If auth= no is coded the permit and deny rules should be coded to reflect the fact that IP addresses are being used to identify users.

If auth is coded then either yes or no must follow the equal sign. Otherwise tnproxy will issue an error message.

### **xforward = yes/no**

The xforward command is used to specify whether an authorized tnproxy user can use X-Window forwarding. The xforwarding command is optional. If xforwarding is not coded in the tnpxy.conf file the proxy uses no as the default value.

If xforward is coded then either yes or no must follow the equal sign. Otherwise tnproxy will issue an error message.

## Access Control Rules

### **deny [users = usrlist] [groups = grplist] from src\_addr to dest\_addr**

The deny command will deny access to a list of users or a group of users if there is a match up of the source and destination IP-addresses.

**users = usrlist:** The users parameter is optional. The userlist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The user-id's are names of users on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading/. The format of the file allows one or more user-id's per line. Blanks, tabs and commas can be used as separators. The # character marks the rest of the line as a comment.

**groups = grplist:** The groups parameter is optional. The grplist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The group-id's are names of groups on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading/. The format of the file allows one or more group-id's per line. Blanks, tabs and commas can be used as separators. The # character marks the rest of the line as a comment.

**from:** The from keyword is required and must immediately precede the source IP-address.

**src\_addr:** The source address field is required and must immediately follow the from keyword. The source address gives the IP address of the requesting host in the standard

dot decimal format. An asterisk '\*' can be used as a wild card for source IP addresses. For example, 192.168.240.\* would permit all from 192.168.240.0 through 192.168.240.255

**to:** The to keyword is required and must immediately precede the destination IP-address.

**dest\_addr:** The destination address field is required and must immediately follow the to keyword. The destination address gives the IP address of the requesting host in the standard dot decimal format. An asterisk '\*' can be used as a wild card for destination IP addresses. For example, 192.168.240.\* would permit all from 192.168.240.0 through 192.168.240.255

### **permit [users = usrlist] [groups = grplist] from src\_addr to dest\_addr**

The permit command will permit access to a list of users or a group of users if there is a match up of the source and destination IP-addresses.

**users = usrlist:** The users parameter is optional. The userlist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The user-id's are names of users on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading /. The format of the file allows one or more user-id's per line. Blanks, tabs and commas can be used as separators. The # character marks the rest of the line as a comment.

**groups = grplist:** The groups parameter is optional. The grplist can contain one or more user-id's or filenames with commas as separators. Spaces and tabs are not allowed in the list. The group-id's are names of groups on the source host and not the destination host. The filenames must be fully qualified pathnames beginning with a leading /. The format of the file allows one or more group-id's per line. Blanks, tabs and commas can be used as separators. The # character marks the rest of the line as a comment.

**from:** The from keyword is required and must immediately precede the source IP-address.

**src\_addr:** The source address field is required and must immediately follow the from keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk '\*' can be used as a wild card for source IP addresses. For example, 192.168.240.\* would permit all from 192.168.240.0 through 192.168.240.255

**to:** The to keyword is required and must immediately precede the destination IP-address.

**dest\_addr:** The destination address field is required and must immediately follow the to keyword. The destination address gives the IP address of the requesting host in the standard dot decimal format. An asterisk '\*' can be used as a wild card for destination IP addresses. For example, 192.168.240.\* would permit all from 192.168.240.0 through 192.168.240.255

## **13.3 Sample tnproxy configuration file**

The following sample tnproxy configuration file is shipped with T.Rex V2. Edit the file to suit your own needs.

```
# file: /etc/firewall/tnproxy.conf
# function: The tnproxy.conf file is used to control the execution of the
#           tnproxy program. Connections between external and internal
```

```

#           hosts are permitted or denied based on the permit and deny rules.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The format of the commands are as follows:
#
# timeout nnn    the number of seconds genproxy will wait for a message
#                before it exits.
#
# deny  src_host src_port
#
# permit src_host src_port to dest_host dest_port
# auth      = yes
# groupid   = staff
# userid    = hermes
# tnproxpath = /home/hermes
# timeout   = 300
# permit groups = staff from 192.168.240.* to *
# deny groups  = * from * to *

```

notes:

# Chapter 14. Webgate Administration

## 14.1 Webgate Overview

Webgate provides gateway services for Web Servers protected behind a firewall. Webgate runs as a stand alone proxy. Webgate reads its configuration file and determines whether or not the function is permitted. If NOT permitted the connection is closed and the attempt is logged. If the requesting host is allowed to connect the permit rule determines the IP address and port number of the host on the other side. Permissions are granted based on the source IP address and port. Entries in the configuration file also determine the destination host and port number.

Webgate supports one-to-one and many-to-one connections. Many-to-one connections can be permitted using multiple permit statements or by using wildcards to specify a range of source addresses on a permit rule.

### Multiple Secure Web Servers

Multiple Web servers can reside behind the firewall each having their own name and IP address. All protocols not necessary to business operations are blocked. Secure transmission of sensitive data is assured by the use of SSL. The Web servers can be isolated on a network separated from the organizations secured networks, thus providing higher levels of security. Communications between the secured web servers and the organizations secured network must pass through the T.Rex firewall. Thus, the firewall can control communications between the web serves and other file serves and data base servers.

To support multiple secure web servers the firewall must listen to a separate IP address for each web server. This can be done using one or more LAN adapters. A single LAN adapter can be configured to listen to multiple IP addresses by defining aliases with the ifconfig command. The following example shows the addition of two alias IP addresses to be used by Ethernet adapter 0. You can use netstat -i to confirm the changes.

#### For AIX:

The following commands should be added to the /etc/rc.net file to activate the aliases when the system is booted.

```
#ifconfig en0 204.71.109.32 alias
#ifconfig en0 204.71.109.33 alias
```

#### For HP-UX:

The following commands should be added to the /sbin/init.d/net file following the ifconfig line. This will activate the aliases when the system is booted.

```
#ifalias lan0 204.71.109.32
#ifalias lan0 204.71.109.33
```

#### For Solaris:

The following commands should be added to the `/etc/init.d/rootusr` file. This will activate the aliases when the system is booted.

```
#ifconfig le0:1 204.71.109.32
#ifconfig le0:2 204.71.109.33
```

The Webgate can be configured to connect to separate web servers for each IP address. See the matching example in the `webgate.conf` file.

### Common Access Log format

The Webgate program logs all requested URLs in the specified log file on the firewall. The access log must be made by the Webgate program since the Web Servers behind the firewall do not have direct access to the client machines, thus can not determine the IP address of the client for logging purposes.

## 14.2 Webgate Configuration File

The Webgate program is controlled by rules defined in the `/etc/firewall/webgate.conf` file. Connections between external and internal hosts are permitted or denied based on the **permit** and **deny** rules. In most cases the deny rule need not be coded, since the default action is to deny all requests except those which are permitted. The order in which the rules are coded is very important, since the first rule encountered that has a match with the source address and source port number is enforced.

## Command Syntax

Each rule is contained on a line that can be up to 1023 bytes long. Spaces and tabs separate fields. All fields are required except for the optional shell command on the deny rule. Optional fields are enclosed in square brackets.

## Mandatory Commands

The `gproxpath` command is mandatory and must be coded before the deny and permit rules.

**gproxpath = pathname**

The `gproxpath` value specifies the directory that Webgate will run under.

## Optional Commands

### **checkprocs = nnn**

The time interval in seconds that webgate waits before checking on the number of child processes. The default value is 5 seconds.

### **DNS = yes|no**

The DNS command is used to specify the use of DNS lookups for logging of hostnames instead of IP addresses. The default value is no.

### **httplog = filename**

The httplog value specifies the name of the file used to log all HTTP activity. If the httplog command is not coded the **default value** used for the file name is **http.access.log**. The http access log is created automatically by Webgate in the directory specified in the genprox command. The httplog must reside in the directory specified by gproxpath since Webgate changes its root directory making all files outside of that directory inaccessible.

### **Logformat = CLFExt | WebTrendsMultiHomed**

The logformat value specifies the format that the http access log will be written. The default value is the Common Log Format with Extensions (CLFExt). If webgate is supporting multiple domains the Multi-Homed WebTrends format can be specified.

### **maxprocs = nnn**

The maximum number of webgate processes permitted.

### **maxuse = nnn**

The maximum number of times a process is to be used before it is retired and replaced by a fresh process. This automatically recovers any resources that may have been tied up by a long running process. For best performance set maxuse to 1000 uses. The default value is 500 uses.

### **nprocs = nnn nnn nnn**

The nprocs command is used to specify the number of processes to start, the minimum number of spare idle processes and the maximum number of spare idle processes. The first number is the number of pre-allocated processes to be started. If this parameter is not coded the default value is 20.

### **timeout = nnn**

The timeout value specifies the number of seconds the Webgate program will wait for a message before it exits.

## Access Control Rules

**deny from *src\_addr* to *loc\_addr* *loc\_port* [*shell cmd*]**

<b>from</b>	The from keyword must immediately follow the permit command. This is used to identify the new command syntax. .
<b>src_addr</b>	The source address field is required and must immediately follow the <b>from</b> keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk <b>'*'</b> can be used as a wild card for source IP addresses. For example, 192.168.240.* would deny all from 192.168.240.0 through 192.168.240.255
<b>to</b>	The to keyword is required and must immediately precede the <b>loc_addr</b> field.
<b>loc_addr</b>	The local host address field is required and must immediately follow the <b>to</b> keyword. The local address gives the IP address of the firewall that is to receive the connection from the source host. This address is in the standard dot decimal format. An asterisk <b>'*'</b> can be used as a wild card for source IP addresses. For example, 192.168.240.* would permit all connections to be made to addresses from 192.168.240.0 through 192.168.240.255
<b>loc_port</b>	The local port field is required and must immediately follow the <b>loc_addr</b> field. The <b>loc_port</b> can be either the port number or the name of the service found in the <b>/etc/services</b> file.
<b>shell cmd</b>	The shell command field is optional. If all the conditions on the line are met and the shell command is specified then the command string will be executed. The shell command can be used to notify the systems administrator of an unsuccessful attempt to use the proxy server.

### PERMIT rule:

The permit command was designed to support multiple IP addresses thus uses three IP addresses as parameters.

**permit from *src\_addr* to *loc\_addr* *loc\_port* redirect (*dest\_host1*,  
*dest\_host2*, ...) *dest\_port* [*vlog*]**

<b>from</b>	The from keyword must immediately follow the permit command. This is used to identify the new command syntax. .
<b>src_addr</b>	The source address field is required and must immediately follow the <b>from</b> keyword. The source address gives the IP address of the requesting host in the standard dot decimal format. An asterisk '*' can be used as a wild card for source IP addresses. For example, 192.168.240.* would permit all from 192.168.240.0 through 192.168.240.255
<b>to</b>	The to keyword is required and must immediately precede the <b>loc_addr</b> field.
<b>loc_addr</b>	The local host address field is required and must immediately follow the <b>to</b> keyword. The local address gives the IP address of the firewall that is to receive the connection from the source host. This address is in the standard dot decimal format. An asterisk '*' can be used as a wild card for source IP addresses. For example, 192.168.240.* would permit all connections to be made to addresses from 192.168.240.0 through 192.168.240.255
<b>loc_port</b>	The local port field is required and must immediately follow the <b>loc_addr</b> field. The loc_port can be either the port number or the name of the service found in the <b>/etc/services</b> file.
<b>redirect</b>	The redirect keyword is required and must immediately precede the <b>dest_addr</b> field. More than one destination host can be specified after the redirect command if they are enclosed within parentheses.
<b>dest_addr</b>	<p>The destination address field is required and must immediately follow the <b>redirect</b> keyword. The destination address gives the IP address of the target host in the standard dot decimal format. Wildcards can not be used with destination IP addresses.</p> <p>If more than one host is specified they must be included within a pair of parenthesis and separated by commas. When multiple destinations are specified webgate will allocate resources in a round robin fashion.</p>
<b>dest_port</b>	The destination port field is required and must immediately follow the <b>dest_addr</b> field. The dest_port can be either the port number or the name of the service found in the <b>/etc/services</b> file.
<b>vlog</b>	The optional keyword vlog is used to specify verbose logging of connect and disconnect messages in the syslog. This is in addition to the records written to the http access. This keyword must appear as the last parameter in the permit statement. The connect and disconnect message will be written with the LOG_NOTICE priority. One can use this feature to keep track of SSL connections which are not recorded in the http access log since the data is encrypted. The syslog records will only be written for those permit statements that apply.



## 14.3 Sample webgate.conf file

The following sample configuration file is shipped with T.Rex Version 1. Edit the file to meet your unique requirements.

```
# file:      /etc/firewall/webgate.conf      system:      T.Rex gateway
# function:  The /etc/firewall/webgate.conf file is used to control the web
# gateway program.
# The webgate program will permit or deny access to the web server(s) based
# on the use of the "permit" and "deny" rules found in this file.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU.  SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The permit from statements allow all systems to access the secured web
servers using
# HTTP with and without encryption (SSL)..
#
#
gproxpath = /home/hermes
httplog    = http.access.log
timeout    = 300
nprocs     = 80 20 40
maxprocs   = 200
maxuse     = 1000
checkprocs = 10
permit from * to 204.71.109.32.1 80    redirect 10.10.0.15 80
permit from * to 204.71.109.32.1 443 redirect 10.10.0.15 443 vlog
permit from * to 204.71.109.33.1 80    redirect 10.10.0.16 80
permit from * to 204.71.109.33.1 443 redirect 10.10.0.16 443
```

Notice the that verbose logging has been specified for IP address 204.71.109.32.1 port 443 (SSL).

## 14.4 Workload Balancing Example

The following configuration file demonstrates the use of workload balancing between two web servers. The maximum number of servers in the list is limited by the maximum length of a command line. This should allow workload balancing between at least 60 web servers.

```
# file:      /etc/firewall/webgate.conf      system: T.Rex gateway
# function:  The /etc/firewall/webgate.conf file is used to control the web
# gateway program.
# The webgate program will permit or deny access to the web server(s) based on
```

```

# the use of the "permit" and "deny" rules found in this file.
#
# (C) Freemont Avenue Software, Inc. 1995 - 2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The permit from statements allow all systems to access the secured web
servers using
# HTTP with and without encryption (SSL)..
#
#
gproxpath = /home/hermes
httplog    = http.access.log
timeout    = 300
nprocs     = 80 20 40
maxprocs   = 200
maxuse     = 1000
checkprocs = 10
permit from * to 204.71.109.32.1 80 redirect (10.10.0.15, 10.10.0.16) 80
permit from * to 204.71.109.32.1 443 redirect (10.10.0.15, 10.10.0.17) 443

```

## 14.5 Refreshing Webgate

Webgate can be refreshed using the refresh.webgate script found in the /usr/local/etc directory.

```
# /usr/local/etc/refresh.webgate
```

This command kills all idle processes, allows active ones to finish, and restarts Webgate using the parameters specified in the current copy of webgate.conf. It also appends the date to the existing http access log and starts a new http access log.

```

#
kill -USR1 `cat /etc/firewall/webgate.pid`
mv /home/hermes/http.access.log /home/hermes/http.access.log.`date +%y%m%d`
/usr/local/etc/webgate

```

Since the webgate.conf file will be read and parsed all configuration changes will take affect. For example, if a new alias IP address and an associated permit statement have been added then webgate will begin listening to the new address.

# Chapter 15. SOCKS Configuration

## 15.1 SOCKS Overview

SOCKS is a public domain software package that allows hosts behind a firewall to gain full access to the Internet in a transparent manner. This is done in a secure fashion that does not require direct IP reachability. The SOCKS server runs on the T.Rex firewall and socks clients run on the protected hosts. SOCKSified versions of ftp, telnet, finger, whois and Mosaic are available. FAS has installed and tested the SOCKS daemon on the T.Rex firewall. FAS provides a binary version of the SOCKS server and sample configuration files at no cost to T.Rex customers. FAS has also built and tested socksified versions of the telnet and ftp clients for AIX, HP-UX and Solaris. Binary versions of these clients are provided to T.Rex customers at no cost.

There are commercial products from other vendors that have been socksified, such as the Netscape Navigator, and the Internet Explorer.

## 15.2 SOCKS Server Installation

The SOCKS server on the T.Rex firewall system is automatically installed by the `install_firewall` script.

## 15.3 Sockd configuration file

The `sockd` configuration file contains directives that control both logging and access control. These directives have been grouped into two parts server setting and access control rules.

### 15.3.1 SOCKS Server Settings

The socks server settings control the generic behavior of the socks daemon. Each keyword is separated from its value by a colon `‘:’`.

#### **logoutput: syslog|file\_name**

The `logoutput` directive tells `sockd` where to write the log messages. The log can be written to `syslog` or some other file or a combination.

**syslog** This keyword tells `sockd` to use the systems `syslog` to record messages.

**file\_name** A fully qualified pathname can be used to force logging to a file other than the standard `syslog`.

To have `sockd` messages written to `syslog` code:  
`logoutput: syslog`

#### **internal: loc\_addr port = 1080**

The `internal` directive tells `sockd` which IP addresses `sockd` will accept client connections. One or more `internal` directives can be coded. `Sockd` will bind each pair of IP addresses and port numbers.

**loc\_addr** The local host address field is required and must immediately follow the **internal:** keyword. The local address gives the IP address of the firewall that is to receive

the connection from the client. This address is in the standard dot decimal format.

**port** The local port keyword is required and must immediately follow the **loc\_addr** field. An equal sign '=' separates the keyword and the numeric value of the port.

**external: ip\_addr**

The external directive tells sockd which IP address is to be used for outgoing connections.

**method: none|username|rfc931**

The method directive tells sockd which methods to use for user **authentication**. The acceptable methods are none, username or rfc931. Rfc931 requires the client machine to run the ident daemon.

**user.privileged name**

The user.privileged directive specifies the user name sockd will use for performing privileged operations.

**name** valid user ID

**user.notprivileged name**

The user.unprivileged directive specifies the user name sockd will use most of the time. operations.

**name** valid user ID

**connecttimeout: nn**

The connecttimeout directive tells sockd how many seconds to wait for the client to send a command after making the connect. If connecttimeout is set to zero '0' it will wait forever. A value of 10 should be more than adequate on a LAN.

**lotimeout: nn**

The lotimeout directive specifies the number of seconds an established connection can be idle before it times out.

**srchost: [nunknown] [nomismatch]**

The srchost directive is optional and is used to deny connections from clients that do not have DNS entries or have mismatched DNS entries.

**nunknown** This keyword will deny connections from source hosts that do not have a DNS entry. The default is to accept the connection.

**nomismatch** This keyword will deny connections from source hosts that have a mismatch between the DNS entry and the IP address.

## 15.3.2 SOCKS Client Rules

Rules prefixed with the keyword **client** are checked first and are used to see if the client is allowed to connect to the sockd server. The client keyword is followed by either a **pass** or **deny** keyword which determines whether the connection is permitted or denied.

```
client pass|deny {  
    from: src_addr/mask1 to: dest_addr/mask2  
    [log: connect,disconnect,data,ioperation] [user: name|method]  
}
```

**from:** The from keyword must immediately follow the client deny or client pass command.

**src\_addr** The source address field is required and must immediately follow the **from:** keyword. The source address gives the IP address of the requesting host in the standard dot decimal format.

**mask1** The mask is separated from the source IP address by a forward slash and specifies the significant number of bits to be used for the comparison. For example, an address of 10.0.0.0/8 would compare only the first eight bits of the 10 block. Thus all addresses from 10.0.0.0 through 10.255.255.255 would match.

**to:** The to keyword is required and must immediately follow the first address/mask.

**dest\_addr** The destination address is required and must immediately follow the to: keyword.  
**mask2** The mask is separated from the destination IP address by a forward slash and specifies the significant number of bits to be used for the comparison. For example, an address of 206.50.87.0/24 would compare the first twentyfour bits of the 206.50.87.0 address block. Thus all addresses from 206.50.87.0 through 206.50.87.255 would match.

**log:** The log: keyword is optional and is used to specify control the level of logging.  
**connect, disconnect, data ioperation**

Acceptable keywords to follow the log directive are connect, disconnect, data and ioperation.

**user:** The user keyword is optional. If coded the sockd daemon will only accept users matching one of the names given as a value. If a method is given then one can specify name, rfc931 or none.

### 15.3.3 SOCKS Rules

```
pass|deny {  
    from: src_addr/mask1 to: dest_addr/mask2  
    command: cmd_list  
    [log: connect,disconnect,data,ioperation] [user: name|method]  
    [method: none|username|rfc931]  
    protocol: [tcp] [udp]  
    proxyprotocol: socks_v4|socks_v5
```

```

    user
}

```

**from:** The from keyword must immediately follow the client deny or client pass command.

**src\_addr** The source address field is required and must immediately follow the **from:** keyword. The source address gives the IP address of the requesting host in the standard dot decimal format.

**mask1** The mask is separated from the source IP address by a forward slash and specifies the significant number of bits to be used for the comparison. For example, an address of 10.0.0.0/8 would compare only the first eight bits of the 10 block. Thus all addresses from 10.0.0.0 through 10.255.255.255 would match.

**to:** The to keyword is required and must immediately follow the first address/mask.

**dest\_addr** The destination address is required and must immediately follow the to: keyword.

**mask2** The IP mask is separated from the destination IP address by a forward slash. If the masking bit is one then an exact match is required. If the masking bit is zero then bit in the IP address is ignored. Specification of a source mask of 255.255.255.255 requires an exact match with the src\_addr for the rule to apply. Specification of an IP mask of 0.0.0.0 permits a match no matter what source IP address is applied.

**command: cmd\_list** The rule applies to the given commands. Valid commands are: **bind**, **bindreply**, **connect**, **udpassocait** and **udpreply**. The commands can be used instead of or in conjunction with the protocol directive.

**log:** The log: keyword is optional and is used to specify control the level of logging. Acceptable keywords to follow the log directive are connect, disconnect, data and iooperation.

**connect**  
**disconnect**  
**data iooperation**

**method: none|username|rfc931**

The method directive tells sockd which methods to use for user **authentication** for this directive. The acceptable methods are none, username or rfc931. Rfc931 requires the client machine to run the ident daemon.

**protocol: [tcp] [ udp]**

The rule applies for the given protocol(s). Either tcp or upd or both can be coded.

**proxyprotocol: socks\_v4|socks\_v5**

The rule applies to requests using the given proxy prootcol. Valid proxy protocols are socks\_v4 and socks\_v5

**user:** The user keyword is optional. If coded the sockd daemon will only accept users matching one of the names given as a value. If a method is given then one can specify name, rfc931 or none.

## 15.4 Sample sockd configuration file

The following sample configuration file is shipped with T.Rex. Edit the file to specify your internal addresses instead of the sample IP addresses provided. Notice that the networks and hosts permitted in the `/etc/sockd.conf` example match the example in the `/etc/securenets` file. This is the secure way to run SOCKS.

**WARNING:** Secure operation of the SOCKS server on T.Rex requires proper configuration. Do not code permit statements using external IP addresses as source addresses. Doing this could let an intruder in from the outside. Also make certain the source address mask has the appropriate bits turned on.

```
# file:                /etc/sockd.conf                system:                T.Rex gateway
# function:            The /etc/sockd.conf file is used to control access to the
#                      SOCKS proxy server (sockd). The sockd will permit or deny access
#                      to service based on the use of "pass" and "deny" commands in the
#                      action field.
#
# (C) Freemont Avenue Software, Inc. (FAS) 1994-2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# SERVER SETTINGS
# The server will log messages to syslog.
logoutoutput: syslog

# The server will bind to the address 10.0.0.1, port 1080 and will only
# accept connections going to that address.
internal: 10.0.0.1 port = 1080

# All outgoing connections from the server will use the ipaddress
# 206.50.87.2
external: 206.50.87.2

# list over acceptable methods, order of preference
#method: username none #rfc931

#or if you want to use rfc931 (ident) too
#method: username rfc931 none

# When doing something that can require privilege, it will use the
# userid "hermes".
user.privileged: hermes

# When running as usual, it will use the unprivileged userid of "hermes".
user.notprivileged: sockd

# how many seconds can pass from when a client connects til it has
# sent us it's request? Adjust according to your network performance
```

```

# and methods supported.
connecttimeout: 10 # on a lan, this should be enough if method is "none".

# how many seconds can the client and it's peer idle without sending
# any data before we dump it? Unless you disable tcp keep-alive
# for some reason, it's proably best to set this to 0, which
# is "forever".
iotimeout: 600 # 10 minutes.

# do you want to accept connections from addresses without
# dns info? what about addresses having a mismatch in dnsinfo?
#srchost: nunknown nomismatch

#=====
# CLIENT RULES.
#=====
# The rules prefixed with "client" are checked first and say who is allowed
# and who is not allowed to speak/connect to the server. I.e the
# ip range containing possibly valid clients.
# It is especially important that these only use ipaddresses, not hostnames,
# for security reasons.
#
# The rules that do not have a "client" prefix are checked later, when the
# client has sent its request and are used to evaluate the actual
# request.
#
# The "to:" in the "client" context gives the address the connection
# is accepted on, i.e the address the socksserver is listening on, or
# just "0.0.0.0/0" for any address the server is listening on.
#
# The "to:" in the non-"client" context gives the destination of the clients
# socksrequest.
#
# "from:" is the source address in both contexts.
#

# the "client" rules. All our clients come from the net 10.0.0.0/255.0.0.0.
# Allow all clients from the 10.0.0.0 network to pass
client pass {
    from: 10.0.0.0/255.0.0.0 to: 0.0.0.0/0
    user: rfc931 # match all idented users that also are in passwordfile
}

# drop everyone else as soon as we can and log the connect, they are not
# on our net and have no business connecting to us. This is the default
# but if you give the rule yourself, you can specify details.
client block {
    from: 0.0.0.0/0.0.0.0 to: 0.0.0.0/0.0.0.0
    log: connect error
}

```



```

# the rules controlling what clients are allowed what requests
#

# you probably don't want people connecting to loopback addresses,
# who knows what could happen then.
block {
    from: 0.0.0.0/0.0.0.0 to: 127.0.0.0/255.0.0.0
    log: connect error
}

# the people at the 172.16.0.0/255.240.0.0 are bad, no one should talk to them.
block {
    from: 0.0.0.0/0.0.0.0 to: 172.16.0.0/255.240.0.0 port = any
    log: connect error
}

# unless you need it, you could block any bind requests.
#block {
#    from: 0.0.0.0/0.0.0.0 to: 0.0.0.0/0.0.0.0
#    command: bind
#    log: connect error
#}

# or you might want to allow it, for instance "active" ftp uses it.
# Note that a "bindreply" command must also be allowed, it
# should usually by from "0.0.0.0/0.0.0.0", i.e if a client of yours
# has permission to bind, it will also have permission to accept
# the reply from anywhere.
#pass {
#    from: 10.0.0.0/255.0.0.0 to: 0.0.0.0/0.0.0.0
#    command: bind
#    log: connect error
#}

# some connections expect some sort of "reply", this might be
# the reply to a bind request or it may be the reply to a
# udppacket, since udp is packetbased.
# Note that nothing is done to verify that it's a "genuine" reply,
# that is in general not possible anyway. The below will allow
# all "replies" in to your clients at the 10.0.0.0/8 net.
#pass {
#    from: 0.0.0.0/0.0.0.0 to: 10.0.0.0/255.0.0.0
#    command: bindreply udpreply
#    log: connect error
#}

# pass any http connects to the example.com domain if they
# authenticate with username.
# This matches "example.com" itself and everything ending in ".example.com".
#pass {
#    from: 10.0.0.0/255.0.0.0 to: .example.com port = http

```

```

#      log: connect error
#      method: username
#}

# block any other http connects to the example.com domain.
#block {
#      from: 0.0.0.0/0.0.0.0 to: .example.com port = http
#      log: connect error
#}

# everyone from our internal network, 10.0.0.0/255.0.0.0 is allowed to use
# tcp and udp for everything else.
pass {
    from: 10.0.0.0/255.0.0.0 to: 0.0.0.0/0.0.0.0
    protocol: tcp udp
}

# last line, block everyone else. This is the default but if you provide
# one yourself you can specify your own logging/actions
block {
    from: 0.0.0.0/0.0.0.0 to: 0.0.0.0/0.0.0.0
    log: connect error
}

```

The preceding example shows simple rules which allow every user from the permitted nodes access to all other nodes through the SOCKS proxy server. More sophisticated rules can be coded. For example, individual users or lists of users on a source host can be permitted or denied access to socks services. Access to destination hosts can also be controlled as well as access to port numbers. A shell command can also be executed if the conditions of a command line are met. Even though SOCKS allows the specification of complex rules it is best to keep the rules as simple as possible. Complex rule sets are harder to maintain and harder to keep accurate.

## 15.5 SOCKS Client installation

Four socks clients have been compiled and tested on AIX, HP-UX and Solaris on both SPARC and Intel platforms and are included on the **SOCKS Installation Diskette 1**. The four clients are: **rfinger**, **rftp**, **rtelnet** and **rwhois**. These programs are functionally equivalent to finger, ftp, telnet, whois. The user interfaces are exactly the same. The four clients run on a protected host and use the socks proxy server running on T.Rex to provide transparent access to the Internet. The four sample programs are "versatile" clients, which means they can be used to connect directly to internal hosts as well as to outside hosts via the SOCKS server.

The SOCKS client programs read the socks configuration file **/etc/socks.conf** to determine whether a request is to be permitted or denied. Requests are permitted or denied based upon the requesters user-id, the requested service and the destination host. The configuration file also

determines whether the SOCKS client will use a direct host connection or use the SOCKS proxy server.

The sample socks client programs can be installed on a protected system using the following script.

```
# ./install_socks_clients
```

## 15.6 Sample socks configuration file

The following sample configuration file is shipped with T.Rex.

```
# file:          /etc/socks.conf          system: protected hosts
# function:      The /etc/socks.conf file is used by all socks client
#               programs to determine whether to use direct or proxy connection to
#               a given destination. It is also used to control access based
#               on destination host, port number, and the effective user-id
#               of the requesting local user.
#
# (C) Freemont Avenue Software, Inc. 1994-2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The location of the socks server(s) is defined using the sockd line.
#
# For this example, the socks daemon is at IP address 172.16.110.1. The
# use of zeros for the destination address and destination mask sends all
# request to the socks proxy server.
#
sockd    @=172.16.110.1 0.0.0.0 0.0.0.0
```

## 15.7 SOCKS Configuration file rules

The **/etc/socks.conf** file is used by all socks client programs to determine whether to use direct or proxy connection to a given destination. It is also used to control access based on destination host, port number, and the effective user-id of the requesting local user. If the **/etc/socks.conf** file is not on the system then the clients will only try direct connections, making them behave like their standard counterparts. Absence of the **/etc/socks.conf** file would allow connections to other hosts behind the firewall, but not to hosts on the other side of the firewall.

Each line in the file may be up to 1023 bytes in length. Lines starting with the character **#** are treated as comments. Besides comment lines there are three types of command lines: **deny**, **direct** and **sockd**. The format of each line is shown below.

**deny** **[\*=userlist]** **dest\_addr** **dest\_mask** **[op dest\_port]** **[ shell\_cmd]**

The **deny** command tells the socks client to reject a request that matches the conditions. This can be used to reject connections to specified hosts or networks, or specified functions (port numbers). It can also be used to reject users requesting access.

**direct** **[\*=userlist]** **dest\_addr** **dest\_mask** **[op dest\_port]** **[ : shell\_cmd]**

The **direct** line tells the socks client when to use direct connections, instead of using the socks proxy server.

**sockd** **[@=serverlist]** **[\*=userlist]** **dest\_addr** **dest\_mask** **[op dest\_port]**  
**[ : shell\_cmd]**

The **sockd** line tells the socks client when to use a proxy connection and which proxy server to use when there is more than one.

The fields used in the preceding commands are explained below.

**\*=userlist** The **userlist** field is optional, and when present consists of one or more user-ids or filenames. Entries are separated by a comma. Tabs and spaces are not allowed. The user-ids should be ids of users on the local host and not ids of users on the destination host or the T.Rex gateway. Filenames must be the full pathname beginning with a leading **/**.

The user-id files can have one or more user-ids per line. Each user-id on a line can be separated by a comma, blanks or a tab. The character **#** marks the remainder of the line as a comment. Each line in the file can be up to 1023 characters long.

If the **\*=userlist** is absent the line applies to all users on the local host.

**@=serverlist** The **server list** can only be used in the **sockd** line. It consists of one or more SOCKS proxy servers which the socks client program will try to use. The SOCKS proxy servers will be tried in the order they appear in the list. Multiple server entries in the list are delimited by a comma. Blanks or tabs are not allowed in the list. The list can contain either domain names or IP addresses. It is probably more prudent to use IP addresses. If the **serverlist** is omitted then the SOCKS client will use the default proxy server, which is determined by the

environment variable SOCKS\_SERVER if it exists.

- dest\_addr** The destination address is a required field that must be immediately followed by the destination mask. The pair together specify the destination IP address or range of IP addresses. The addresses are specified in the standard IP dotted format.
- dest\_mask** The destination mask is a required field and must immediately follow the destination address. If the masking bit is one then an exact match is required. If the masking bit is zero then bit in the IP address is ignored. Specification of a destination mask of 255.255.255.255 requires an exact match with the P address for the rule to apply. Specification of a destination mask of 0.0.0.0 causes a match no matter what destination address is applied.
- op dest\_port** The operator destination port is an optional field. The valid codes for the operator are **eq**, **neq**, **gt**, **lt**, **le** and **ge**. This stands for equal, not equal, greater than, less than, less than or equal and greater than or equal. The **dst\_port** can be either the port number or the equivalent name of the service found in the /etc/services file. For example, one could code either port number 23 or telnet. If the **op dst\_port** pair is not coded then all destination ports are valid.
- :shell cmd** The shell command field is optional. If all the conditions on the line are met and the shell command is specified then the command string will be executed. The following substitutions will be performed before the string is passed to the Bourne shell for execution.
- %A** will be replaced by the client host's domain name if known otherwise it will be replaced by the IP address.
  - %a** will be replaced by the client host's IP address.
  - %c** will be replaced by the command that sockd is asked to execute (**connect** or **bind**).
  - %p** will be replaced by the process id of sockd.
  - %S** will be replaced by the service name of the destination port number if known otherwise by the port number. If the destination port number is 20 then **ftp** will replace %S in the command string.
  - %s** will be replaced by the destination port number.
  - %U** will be replaced by the user-id reported by identd.
  - %u** will be replaced by the user-id reported by the client program.
  - %Z** will be replaced by the destination host's domain name.
  - %z** will be replaced by the destination host's IP address.
  - %%** will be replaced by a single %.

Multiple shell commands can be strung together in the usual way. For example:

```
/usr/bin/finger @%A | /usr/bin/mail -s 'SOCKS: rejected %u@%A to %Z (%S)' root  
root@%A
```

will finger the client host and pipe the result into an e-mail message for superusers at the server host and the client host with an appropriate Subject line. This feature will be used most often with a deny rule, but it could also be used with a permit rule.

When the SOCKS client has to make a network connection it first checks the **/etc/socks.conf** file to determine the action to be performed. It reads the rules one line at a time until a match is found then it uses the selected rule. Therefore, if there are multiple rules the order is important. When a match is found the remaining lines are skipped. If no match is found then the request is denied.

## 15.8 Domain Name Service for SOCKS

SOCKS clients allow access to unprotected systems by hostname even though there is no direct IP addressability between the protected and unprotected networks. In order for the socks client program to resolve the external name into an IP address for use by the SOCKS proxy server it must be directed to search the caching-only DNS on the T.Rex firewall. The socks client program must know two things before it can resolve internal and external names. It must know its own domain name and the IP address of the caching-only DNS on the gateway. This information can be provided to socks clients three ways.

### 1. Specify Environmental variables

Specify the environment variable **SOCKS\_NS** with the IP address of the T.Rex firewall. Substitute the IP address of the secure network interface on your T.Rex firewall in place of the sample value given below.

**SOCKS\_NS=192.168.0.1**

If you are running a socks client on a system that does not have the IP address of the internal DNS specified in the **/etc/resolv.conf** file then you should use the **SOCKS\_DNAME** to specify your domain name, as shown below.

**SOCKS\_DNAME=your.domain.com**

### 2. Use shell scripts to specify environmental variables

As an alternative you may choose to specify the environmental variables in scripts to be executed by your users. The sample script shown below can be used to run the socks rtelnet client.

```
#!/bin/sh
SOCKS_DNAME=your.domain.name
export SOCKS_DNAME
SOCKS_NS=192.168.0.1
rtelnet "$@"
```

### 3. Compile the default values into the client programs

If you use the socks source code to build your own clients simply modify the #define statements found in the ../include/socks.h file as shown below:

```
/*=====
 * file: ../socks/include/socks.h
 *
 *-----
 * Default SOCKS server host; you MUST set this for you site.
 * This is overridden at run time by the contents of environment
 * variable SOCKS_SERVER if it exists.
 *=====*/
#define SOCKS_DEFAULT_SERVER "gw.your.domain.com"

/*=====
 * Default Domain Name Server (DNS) for the socks clients. This
 * define must contain the IP address of the T.Rex firewall.
 * This variable can be overridden at run time by the contents of
 * environment variable SOCKS_NS if it exists.
 *=====*/
#define SOCKS_DEFAULT_NS "192.168.0.1"

/*=====
 * Default Domain Name. Leave it undefined if your systems use
 * /etc/resolv.conf to point at the Internal DNS.
 * This define can be overridden at run time by the environment
 * variable SOCKS_DNAME.
 *=====*/
/*#define SOCKS_DEFAULT_DNAME "domain.com"*/

...
```

## SOCKS Source Code

A complementary diskette with the SOCKS source code is provided at no charge. This source code is shipped with T.Rex at no charge. SOCKS source code can also be found at the anonymous ftp server **ftp.nec.com** in directory **/pub/security/socks.cstc**. Source code is available for the socks server and socks clients. Basic clients are available for finger, ftp, telnet and whois.

FAS has already built the SOCKS proxy server and customized it for the T.Rex system. FAS plans to keep its customers supplied with tested updated versions of the SOCKS proxy server on a timely basis. T.Rex customers should not have to expend the time and effort to customize, build and test the SOCKS proxy server for the T.Rex system. FAS has made a number of small changes that fix compile and run time problems. These changes are not part of the sample source.

FAS has already built and tested the SOCKS client programs: finger, ftp, telnet and whois for AIX, HP-UX and Solaris systems.



# Chapter 16. DNS Configuration

## 16.1 DNS Overview

T.Rex is designed to provide domain name service for both internal and external access without leaking internal host names or IP addresses to external users. To achieve this goal T.Rex uses separate external and internal Domain Name Servers (DNS). Internal hosts use the DNS behind the firewall for all name resolution. The internal DNS is configured to resolve all internal host names. The internal DNS also resolves external host names by forwarding the request to the firewall. The DNS on the firewall then resolves the external host name and returns the answer to the internal DNS. With this dual DNS configuration internal users can resolve both internal and external host names in a seamless manner. Users of external systems cannot access or even see the internal DNS. The internal network is not IP addressable and is completely hidden from the outside. Yet, e-mail, telnet and other authorized applications can access the internal systems.

## 16.2 Internet Firewall DNS

T.Rex can serve as an Internet Firewall or an Intranet firewall. As an Internet firewall it runs a caching DNS that is primed by the Internet root name servers. As a Intranet firewall the firewalls DNS is used to resolve hosts on the external side of the firewall. Since it does not have direct access to the Internet so it does not make use of the Internet's root name servers. Instead it forwards request for Internet name resolution to another name server that has access to the Internet.

### 16.2.1 Caching-Only DNS Firewall

If the Internet Service Provider (ISP) provides name service for the organizations external computers the firewall will run a caching-only DNS. A caching-only name server does not use local name server data base files. It acquires it's answers from remote name servers on the Internet. Once it learns the answer to a query it caches the answer and uses it for future queries. All name serves cache the answers to their queries. A caching-only name server is 100% dependent on this method.

The T.Rex proxies are aware of the dual DNS architecture. Ftp proxy and tnproxy will query both the external and internal DNS to resolve a name if the first DNS did not provide the answer. These proxies use the address of the external DNS found in the `/etc/resolv.conf` file and the address of the internal DNS found in the `/etc/firewall/resolv.inside.conf` file. This allows the proxies to resolve both internal and external names in a seamless manner.

Delivery of e-mail to internal mail servers such as POP3 servers, or MS Mail Exchange servers requires the servers to be defined in the `/etc/hosts` file. See section 16.4 for details.

The caching-only DNS requires only three files in the `/etc` directory:

named.boot,  
named.ca, and  
named.local.

All three files are automatically created by the T.Rex installation. The `named.boot` and `named.ca` file

do not require any modification. The named.local file needs to be edited to reflect the fully qualified host named of the firewall.

## 16.2.2 Full Function DNS

In some cases you may want to use the DNS on the firewall as the primary DNS for your domain. This can be done by adding two files one for forward lookup and the other for reverse lookup. None of the internal hosts data should be contained in these files.

## 16.3 Internal Firewall

Internal firewalls that do not have direct Internet connectivity should be configured a primary DNS for the external side of the firewall.

## 16.4 Hosts File

Delivery of in-bound e-mail requires entries for each internal mail server, as the host name can not be resolved by the external DNS, and the mail program is unaware of the internal DNS server. The /etc/hosts file entries for the mail server should be coded as follows.

```
# file: /etc/hosts
# mail servers
10.11.12.17    pop3.lsl.com          pop3
10.11.12.18    msexch.lsl.commsexch
```

# Chapter 17. Performance Monitor

## 17.1 Performance Monitor Overview

The performance monitor displays the most useful performance statistics on a single screen. The monitor displays CPU utilization, memory usage, paging activity, process events, disk I/O activity, and network I/O. This monitor is available with the T.Rex-ESm systems, and with T.Rex on AIX 4.

The performance monitor has two output modes: on the screen or to a file. Both modes can be active at the same time. The screen output is used for real-time monitoring or ad hoc trouble shooting. The file mode is used for general performance monitoring and capacity planning. The monitor can display its output in an X-window or on an ASCII display.

## 17.2 Monitor Options

The performance monitor can be started as follows:

```
/usr/local/etc/monitor [-options]
```

The monitor shows various system variables and updates the screen on a periodic basis.

The monitor can be stopped using a ctrl-C or by typing in the 'q' command.

### 17.2.1 Interactive key commands

Other interactive key commands are:

- a toggles between normal and alternative display.
- d toggles to full disk display in alternative display.
- n toggles to full network display in alternative display.
- s toggles to SMP cpu display in alternative display.
- t toggles to top processes display
- u toggle between username and userid in top.
- r toggle between showing running processes versus all processes.
- c,p,m,v

Sorts "top" processes by cpu usage, page faults, main memory usage or virtual

memory usage.

- R toggle to include root sessions in user count/activity.
- + increase the sample time by one second.
- decrease the sample time by one second.
- > increase the sample time by 5 seconds.
- < decrease the sample time by 5 seconds.
- ? show help message.

All commands work in both the normal and alternative display mode. The 'd' and 'n' commands do not automatically switch between these display modes. The 't' command is special: it remembers the top display status for both display modes.

Typing the space bar or enter key will cause the monitor to update the sample being displayed on the screen. The default period between sample updates is 10 seconds.

## 17.2.2 Monitor Options

These options are entered when you start the monitor. All the options and their arguments can be abbreviated to their first character except for the **-smp** option, since it would overload the **sample** option.

**-alternative** Shows both the system events and the top processes. This option requires the use of X-window aixterm or xterm.

**-compress [*prog*]**

When the filename for logmode changes, this option will compress the old logfile using the *prog* as pipe. The default value for *prog* is /usr/local/bin/gzip.

**-disk** Show detailed disk information.

**-Disk** Sort disk information on read+write rate.

**-Highlight [*none/reverse/bold/c#/m#/v#*]**

This sets the highlight mode of headers.

c sets the color to the number specified.

m set metric color name.

v set metric color value.

- interval time** Set the time interval to time second for logmode. The default is 900 second or 15 minutes.
- log filename** Switch on log mode. This will dump the log data to the specified file on a time interval basis.
- L** Dump screen samples to the logfile if logmode is on.
- net** Show detailed network statistics.
- run** Show only running processes or show those processes that have gained cpu time in the interval.
- Rootinclude** By default root sessions are ignored in user count/activity calculations. This option will include them.
- sample time** Set the delay between screen updates to *time* seconds. The default delay is 10 seconds after first 2 second delay.
- smp** Show Symmetric Multi-Processor cpu information.
- Sync** Synchronize sample and interval time on day boundary. This option is implied by the **-log** option. If Sync is specifed and the time intevlas is set at 900 seconds (15 min) then the intervals will end at 0:00, ):15, \_:30, 0:45, 1:00 etc.
- top [nn]** Shows top cpu processes and shows only summary of system variables. If nn is specified it will show the top nn processes.
- Toplog** When log mode is on the top processes will also be written to the log file.

## 17.3 Sample Monitor Display

The default monitor mode will show the following information:

Hostname, date and time between display delays.

Percentage distribution of cpu-load by system, wait, user and idle.

Runnable processes value/second and load average values of 1, 5 and 15 minutes.

Processes waiting to be swapped in.

Real and virtual memory usage including totals. Real memory usage is shown for processes and file pages. AIX will map files in real memory to maximize performance. Thus real memory will always be heavily used.

Paging information.

Various process and system events.

Disk activity read and write bytes/second and the disk utilization percentage for each hard drive.

Network activity read and write bytes/second and the utilization percentage for each NIC.

## Sample Report

```
AIX monitor v1.12: gw.lsl.com                      Wed Jun 10 13:09:40 1998
Refresh: 1.00 s

Sys  2.0% Wait  3.9% User  2.0% Idle 92.2%
0%           25%           50%           75%           100%
=WW.....
Runnable processes 0.00 load average: 0.00, 0.01, 0.02

Memory    Real      Virtual    Paging (4kB)    Process events    File/TTY-IO
free      0.4 MB    79.4 MB    2.9 pgfaults    111 pswitch       23 iget
procs     27.0 MB    32.6 MB    2.9 pgin        119 syscall       3 namei
files     4.6 MB           0.0 pgout       48 read          11 dirblk
total     32.0 MB    112.0 MB    1.9 pgsin       5 write          17764 readch
                                0.0 pgsout      0 fork           12108 writech
DiskIO     Total Summary
read       8.2 kByte/s
write      0.0 kByte/s
transfers  2.9 tps
active     1/2 disks

TOPdisk read write busy
hdisk0    8      0 kB/s  4%
cd0       0      0 kB/s  0%

Netw  read  write
lo0   0.3    0.3 kB/s
en0   14.4   1.0 kB/s
en1   0.7   12.5 kB/s
en2   0.0    0.0 kB/s
```

## CPU Usage

The CPU utilization is broken down by System, Wait, User and Idle percentages. The line below the numeric values is a bar graph showing CPU usage.

## Run Queue Lengths:

Immediately below the bar graph is run queue length averaged for the last sample interval, the previous minute, the previous 5 minutes, and the previous 15 minutes. These four averages will tell you if the load is increasing or decreasing with time.

## Real and Virtual Memory Usage

This table shows how much memory is allocated to processes, and files and how much is free. The Virtual column refers to the amount of page space used by processes and how much is free. Notice that files do not use page space.

## Paging Column

The paging column shows paging in and out of page space as well as paging in and out of the filesystems

## Process Events

The process events shows:

pswitch:	process switches
syscall	system calls
read	read calls
write	write calls
fork	forks
exec	exec calls
rcvint	
xmtint	

# Chapter 18. T.Rex Integrity Monitor

## 18.1 T.Rex Monitors Overview

The T.Rex Monitors provide additional security by monitoring the vital activities on the firewall. The T.Rex Monitors consist of **fwmon**, **procmon**, **spoofmon** and **synmon**. Procmon, spoofmon and synmon are automatically started by fwmon when the system is booted.

**fwmon** periodically checks on the correct configuration of the firewall to make sure that operating system parameters are properly set. For example, the IP forwarding and source routing options in the operating system need to be disabled on the firewall. fwmon will periodically check on these parameters and disable the options if they are found enabled.

**procmon** monitors the process table for processes whose names are specified in the `/etc/firewall/T.Rexmon.conf` file. procmon counts the number of occurrences for each of the specified processes and issues an alert if the number falls outside of the specified range. One use of procmon is to monitor critical processes to make sure they are running. A second use is make certain some processes are not running.

**spoofmon** monitors TCP/IP connection requests to the firewall. By examining output from the device driver it is able to determine which communication interface received the connection request. This enables T.Rex to determine if the source IP address has been spoofed. If a protected source IP address was received by the wrong communication adapter then the address has been spoofed. spoofmon communicates the connection request information to the proxies which will reject and log any source IP address spoofing attempts.

When spoofmon is active, ftproxy and tnproxy depend on it for connection information used to detect source IP address spoofing. If spoofmon is stopped for any reason (eg. killed) ftproxy and tnproxy will detect the condition and refuse all connections until spoofmon is restarted. If spoofmon is not running the proxies will write error messages to the system log. The spoofmon program can be restarted by issuing the following command.

```
# /usr/local/etc/fwmon
```

**synmon** monitors the `so_q0` queues for each active port on the firewall. The `so_q0` queues hold incomplete connection requests that are pending. By monitoring the queues status and by giving the administrator the ability to manage the queues, synmon provides an effective defense against SYN flood (a form of denial of service) attacks. **Synmon runs as a kernel extension and is currently available only on AIX 4.1.4, 4.2, and 4.3.**

## 18.2 Firewall Monitor Invocation

The firewall monitors are contained in the directory `/usr/local/etc`. T.Rex auto-install process updates the system configuration files so that the monitors are automatically started at system boot time. The location of the scripts is operating system dependent as shown by the following table:

OS	startup script
AIX	<code>/etc/rc.tcpip</code>



HP-UX	/sbin/init.d/T.Rex
Linux	
Solaris	/etc/init.d/T.Rex

On startup, the T.Rex Monitors read the /etc/firewall/fwmon.conf file to determine the parameters to use for monitoring and managing the firewall system.

synmon also has the capability to re-read the /etc/firewall/fwmon.conf file without having to be stopped and then re-started. After a change is made to the /etc/firewall/fwmon.conf file, issue the following command to ask synmon to re-read the parameters:

```
# kill -USR1 `cat /etc/firewall/synmon.pid`
```

## 18.3 Fwmon Configuration File

The firewall monitors are controlled by commands specified in the /etc/firewall/fwmon.conf file. These commands control the monitoring and management of T.Rex, including how it will respond to various forms of attack.

### Command Syntax

Each parameter is contained on a line that can be up to 1023 characters long. Spaces and tabs and equal signs "=" separate fields. Comment lines begin with the "#" character. Blank lines are ignored. The format of each parameter is shown below:

### Mandatory Parameters

The mandatory synmon parameters are intended for use during normal network activities. They are applied to all the active ports on the firewall. The queue length should be set low to enable early detection of a SYN flood attack while preserving system resources for normal activities.

#### **syn\_chk\_interval = n**

This specifies the number of seconds to wait between iterations of the synmon activities of monitoring and managing the partially completed connection queue (so\_q0 queues). If syn\_chk\_interval is set equal to zero then synmon is disabled and syn\_qlen, syn\_himark and syn\_lowmark are not required.

#### **syn\_qlen = n**

This specifies the maximum number of entries to allow in the so\_q0 queue for every active port on the firewall.

This value should be low, the AIX default is 10 and it is a good starting value to use. The kernel code uses the following formula to compute the absolute maximum queue length.

$$3/2 * (\text{value specified}) + 1$$

For example, if the value specified is 10, the maximum is 16. If the value specified is 1000, the maximum is 1501.

### **syn\_himark = n**

This specifies that synmon should alert the system administrator to a possible attack and start cleaning up a so\_q0 queue if there are more than n entries in the queue. This should be a value that is somewhat less than the absolute maximum queue length. Set it to around 90% of the absolute maximum queue length.

### **syn\_lomark = n**

This specifies the number of entries in an so\_q0 queue to be kept during queue cleanup. If this number is too low, it makes it less likely for valid connections from slower networks to connect to the firewall during an attack. If this number is too high, synmon cleanup may not be able to open enough slots on the queue for new (hopefully valid) incoming connection requests. For normal network load, set this to around 15% to 20% of the absolute maximum queue length.

## **Optional Parameters**

### **procmon\_interval [ *check\_interval* [ *alert\_interval* ]**

The procmon\_interval specifies the time intervals used by procmon.

**check\_interval** This specifies the number of seconds to rest between procmon checks. The default value is 3600 seconds or 1 hour.

**alert\_interval** This specifies the minimum number of seconds between alerts from procmon. The default value is the check\_interval.

### **procmon process\_name [ n1 [ n2 ] ]**

Procmon specifies the processes to be monitored. The presence of a procmon statement in /etc/firewall/T.Rexmon.conf will activate procmon. If no procmon statement is specified, procmon will be disabled. Procmon is not supported by AIX 3.2.5 or HP-UX.

**process\_name** The name of a process to be monitored. Such as smwrapd.

**n1** The minimum number of occurrences for the named process. The default value is 1. If the minimum and maximum values is 0 then an alert will be issued if the process is running.

**n2** The maximum number of occurrences allowed for the named process. The default value is n1.

### **procmon\_email email\_addr**

procmon\_email specifies the e-mail address for procmon to send alerts to.

**email\_addr** e-mail address to receive procmon alerts. The default is root on the local machine.

**spoofmon = yes|no**

The default value for spoofmon is yes, and this is the normally way to run T.Rex. For most systems and workloads this overhead is less than 5% of the processor. There are situations when spoofmon can be safely disabled thus reducing the cpu time required to perform the check. spoofmon can be safely disabled under the following circumstances:

The firewall is behind a router that blocks all packets containing protected source IP addresses.

The tnproxy and ftproxy deny all connections from internal IP addresses to other internal IP addresses. In other words no one is allowed to bounce off the inside of the firewall.

**syn\_verbose = yes|no**

Specifying yes will produce more syslog output from synmon. This may be helpful during an attack for monitoring the port status. Default is no.

**syn\_trusted = pathname**

This specifies an input file where synmon can obtain a list of trusted IP addresses, and optionally, their netmasks. Connection requests from any of the trusted IP addresses will not be purged from the queue during synmon cleanup. This is intended to allow the administrator to selectively give priority to certain IP addresses during an attack so that a connection from these hosts will be more likely to get completed. All the addresses in the /etc/firewall/securenets file are automatically trusted so outbound connections have priority.

**syn\_port = p , qlen = l , himark = hi , lomark = lo**

All these parameters have to be on the same line. This specifies a port that synmon should manage differently from others. More than one of these lines can be specified for different ports.

**p** gives the port number that these parameters apply to.

**l** is similar to the syn\_qlen parameter above but only applies to port p.

**hi** is similar to the syn\_himark parameter above but only applies to port p.

**lo** is similar to the syn\_lomark parameter above but only applies to port p.

## 18.4 SYN Flood Discussion

(... 3 way handshake ...)

A SYN flood attack makes use of the fact that a TCP connection request is put into an so\_q0 queue on the server and kept for a pre-defined timeout period which, on most systems, has a default of 75 seconds. After a SYN packet is received, the server places it in the so\_q0 queue for the requested port and sends out a SYN/ACK packet to the client and will wait for a SYN/ACK packet from the client before accepting the connection request. If the client's IP address is spoofed and is not

reachable from the server, the connection request will stay on the `so_q0` queue for the entire timeout period. If just enough entries with unreachable client addresses are sent to a server, the server's `so_q0` queue for that port will be full for the entire timeout period.

There is one `so_q0` for each active port on the server. A SYN flood attack can be directed to a specific port or ports on the server while leaving other ports untouched. The rate of the attack is limited by the slowest link between the attacker and the target. So unless the attacker is on a high speed link to the target, the attack will usually be concentrated on a few selected ports. An unfortunate implication of this is that the higher the speed your link to the Internet is, the higher the SYN flood rate is possible to your site.

There is no complete solution known to-date against a SYN flood attack since the attack makes use of the current TCP protocol in such a way that it is impossible to distinguish a hostile packet from a friendly one. The best defense so far is to detect the existence of an attack and enable flexible management of system resources to survive the attack.

`synmon` monitors the `so_q0` queues and alerts the system administrator to unusual activity loads when they are detected. It also provides the capability for the administrator to dynamically manage the system resources to minimize the effects from an attack.

## 18.5 General Discussion

Using `synmon` Under normal circumstances

The example `/etc/firewall/T.Rexmon.conf` file contain `synmon` parameter values that should be good starting points for most installations.

The `syn_chk_interval` value is set to 10 seconds so that, every 10 seconds, `synmon` will perform a complete sweep of all the `so_q0` queues to look for signs of unusually high number of connection requests. This should provide a sufficiently early alert to any SYN flood attacks while requiring only a small percentage of system resources.

`syn_qlen` is set to 10 which is the default on most operating systems. Unless your network load requires you to make a change to this operating system parameter (called `somaxconn`), you should leave `syn_qlen` set to 10. This will provide for a maximum of 16 entries on each `so_q0` queue (see explanation in `syn_qlen` section above).

`syn_himark` is set to 14 so that an alert can be issued just before the queue filled up. It can be set to the maximum of 16 if your network is busy so as to avoid getting superfluous alerts.

`syn_lomark` is set to 2 so that `synmon` will leave 2 entries on a queue during cleanup. This parameter is not that important under normal loads when detection is the main focus.

## 18.6 When an attack is detected

First determine which ports are being attacked using the alert messages generated by `synmon` in the `syslog`.

The `syn_chk_interval` of 10 is roughly equivalent to reducing the timeout for incomplete connection requests from the system default of 75 seconds to 10 seconds. You may want to lower this value somewhat if you think the valid connections can usually be connected within the shorter time.

Increase the queue length for the ports being attacked so that more connection requests can be queued, increasing the chance for valid attempts to complete a connection. However, it takes system resources to support additional entries in the queues. Do not use the `syn_qlen` parameter but use the `syn_port` line so that only the attacked ports get increased queue sizes.

The basic idea is, given an attack rate of  $n$  packets a second, we make adjustments so that

$$n * \text{syn\_chk\_interval} < \text{syn\_port qlen}$$

Unfortunately,  $n$  is not easily determined unless you have a sniffer tool that can measure the rate. This makes the adjustment of the parameters a non-trivial task. Increment `syn_qlen` slowly (100 at a time) and keep it no greater than 1000. If you start noticing slower keystroke response from the system, stop. Then start decrementing `syn_chk_interval`, keeping it no less than 3. At each adjustment, get a rough estimate of the effect by sending multiple requests to connect to the port and measuring the success rate. You should also make `syn_port` `hmark` relative low (around 50% of `syn_port qlen`) to make sure `synmon` does a cleanup every time it wakes up.

System resources can get used up rather rapidly by the increased queue sizes. So if you are being attacked at a high rate on multiple ports, pick the most important one (or two, keep it as few as possible) to give more resources to. Specify `syn_port` lines only for this port. If too much resources are allocated to queue management, the system may slow to a crawl, which makes the situation worse than having one or two ports shutdown by an attack.

After you made the necessary adjustments in `/etc/firewall/fwmon.conf`, issue the following command to have `synmon` read the new parameters:

```
kill -USR1 `cat /etc/firewall/synmon.pid`
```

## 18.7 Example /etc/firewall/T.Rexmon.conf File

The sample fwmon.conf file will cause an alert to issued if any of the following conditions apply:

(1) the following daemons are found running: fingerd, ftpd, talkd, telnetd, tftpd, .rexecd, rlogind, rshd, and sendmail,.

(2) the following daemons are not running: named, fwmon, smwrapd, spoofmon, synmon. If you are running T.Rex on a system other than AIX synmon is not support and its entry should be removed.

```
# file: /etc/firewall/fwmon.conf
# function: The fwmon.conf file is used to control the execution of the
#           Firewall monitor program.
#
# (C) Freemont Avenue Software, Inc. 1996-2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
spoofmon           = yes
syn_verbose        = no
syn_chk_interval   = 10
syn_qlen           = 16
syn_himark         = 23
syn_lomark         = 2
# Please read the T.Rexmon chapter in the T.Rex Administration Guide
# carefully before adding syn_port entries.
# Example:
# syn_port 25 qlen=1000 himark=900 lomark=100
# syn_port 80 qlen=1000 himark=900 lomark=100
procmon_email jim@lsli2
procmon_interval   600
procmon named      1 1
procmon fwmon      1 1
procmon smwrapd    1 10
procmon spoofmon   1 1
procmon synmon     1 1
procmon syslogd    1 1
procmon fingerd    0 0
procmon ftpd       0 0
procmon talkd      0 0
procmon telnetd    0 0
procmon tftpd      0 0
procmon rexecd     0 0
procmon rlogind    0 0
procmon rshd       0 0
procmon sendmail   0 0
```

notes:

# Chapter 19. Firewall Logging Facilities

Since the firewall system deals with security, its important to know how to track who's logging in, who's using FTP, and who's trying to get in without permission. This chapter shows you how to activate system logging, how to configure what is logged, and how to produce and analyze activity reports.

## 19.1 Activating System Logging

The `/etc/syslog.conf` controls which messages are logged and where they are logged. Each line of the file specifies the **type of message** to log and the **destination** where the message will be logged. The type of message consists of two parts separated by a period. The first part specifies the facility that generated the message. The second part specifies priority of the message.

### Type of message (facility.priority)

**facility:** The following facilities can be specified in the syslog configuration file: kern, user, mail, daemon, auth, syslog, lpr, news, uucp and \* for all messages.

**priority:** The priorities which can be specified are (from high to low): emerg/panic, alert, crit, err, warn, notice, info, debug.

For more information see the IBM InfoExplorer which contains an on-line description of the commands that can be used in the syslog configuration file.

### Destination

The destination of the message can be a specified file on a system, or a user id on a system. To send the message to another system follow the file name or the user id by the character string `@hostname`.

The following example shows a `syslog.conf` file which will log the following:

all authorization messages at info level and higher will be written to the syslog,  
all notices will be written to the syslog,  
all alerts are written to the console,  
all emergency messages are written to all users.

### 19.1.1 Sample `/etc/syslog.conf` file

```
# file:      /etc/syslog.conf      system:      gw
# function:  specify messages to logged by syslogd.
# created:   by rjl@lsli.com      3/15/94
#
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1994-2000
# All Rights Reserved
#
auth.info    /var/adm/syslog
*.notice     /var/adm/syslog
*.alert      /dev/console
*.emerg      *
```

If you want to log all system messages on another internal host then simply add the following line to `/etc/syslog`



replacing *hostname* with the name of your protected host.

```
*.debug @hostname
```

## 19.1.2 Create the system log file

If the system log file doesn't exist you need to create the file using the touch command. Otherwise nothing will be logged.

OS	Command
AIX	<b># touch /var/adm/syslog</b>
HP-UX	<b># touch /var/adm/syslog/syslog.log</b>
Linux	<b># touch /var/log/messages</b>
Solaris	<b># touch /var/adm/messages</b>

## 19.1.3 Activating syslogd

The syslogd should be automatically activated at system boot time. by a command in the /etc/rc.tcpip file. The default file shipped with AIX has the following command that activates the syslogd.

```
start /etc/syslogd "$src_running"
```

## 19.1.4 Activating syslog changes

After changing /etc/syslog.conf you need to refresh the syslogd to activate the changes. This can be done with the following command.

For AIX:

```
# refresh -s syslogd
```

For Solaris:

```
# kill syslod.pid  
#/usr/sbin/syslogd
```

Where *syslogd.pid* is the process ID of syslogd.

## 19.1.5 syslog maintenance

To do this, issue the following commands. Periodically you will want to move the existing /var/adm/syslog file to an archive and start with a fresh /var/adm/syslog.

```
#mv /var/adm/syslog /var/adm/syslog.`date +%y%m%d`  
#touch /var/adm/syslog  
#refresh -s syslogd
```

Don't forget to refresh the syslogd or it will continue to write to the original file instead of the new one. In the previous example we simply concatenated the date to the name of the syslog file to help identify it.

## 19.2 SYSLOG Format

The format of the information in the system log is as follows:

date time host prog\_name [process\_id]: message text.

The following example shows typical lines found in the system log.

```
Jun  8 14:39:14 gw ftpoxy[17523]: permit user = jim source = k2.lsl.com (192.168.24.2) connect to info.cert.org (192.88.209.5)
Jun  8 14:41:08 gw ftpoxy[17523]: GET CA-95:07a.REVISED.satan.vul from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:42:30 gw ftpoxy[17523]: GET cert_faq from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:43:46 gw ftpoxy[17523]: GET FAQ.virus-1 from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:46:30 gw ftpoxy[17523]: GET virus-1.README from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:49:04 gw ftpoxy[17523]: GET md5.1.ps from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:49:18 gw ftpoxy[17523]: GET md5-announcement.txt from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:49:36 gw ftpoxy[17523]: GET md5c.c from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:49:57 gw ftpoxy[17523]: GET md5.h from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:50:05 gw ftpoxy[17523]: GET mddriver.c from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:50:15 gw ftpoxy[17523]: GET rfc1321.txt from info.cert.org (192.88.209.5) user = jim at k2.lsl.com (192.168.24.2).
Jun  8 14:50:31 gw ftpoxy[17523]: exit user = jim source = k2.lsl.com (192.168.24.2) dest = info.cert.org (192.88.209.5) cmds = 70 in = 227752 out = 0 duration = 718 sec cpu = 0.21
Jun  8 14:58:36 gw ftpoxy[15224]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 129.7.1.3 aka jane.uh.edu.
Jun  8 15:18:37 gw ftpoxy[13717]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 129.7.1.3 aka jane.uh.edu.
Jun 11 00:08:48 gw ftpoxy[17606]: permit user = jim source = k2.lsl.com (192.168.24.2) connect to manix (192.168.0.6)
Jun 11 00:09:06 gw ftpoxy[17606]: GET testlog from manix (192.168.0.6) user = jim at k2.lsl.com (192.168.24.2).
Jun 11 00:09:17 gw ftpoxy[17606]: exit user = jim source = k2.lsl.com (192.168.24.2) dest = manix (192.168.0.6) cmds = 7 in = 3774448 out = 0 duration = 37 sec cpu = 2.83
Jun 11 00:09:51 gw ftpoxy[17607]: permit user = jim source = k2.lsl.com (192.168.24.2) connect to manix (192.168.0.6)
Jun 11 00:10:12 gw ftpoxy[17607]: GET testlog from manix (192.168.0.6) user = jim at k2.lsl.com (192.168.24.2).
Jun 11 00:10:25 gw ftpoxy[17607]: exit user = jim source = k2.lsl.com (192.168.24.2) dest = manix (192.168.0.6) cmds = 8 in = 3740716 out = 0 duration = 44 sec cpu = 2.71
Jun 11 00:11:20 gw ftpoxy[17608]: permit user = jim source = k2.lsl.com (192.168.24.2) connect to manix (192.168.0.6)
Jun 11 00:11:43 gw ftpoxy[17608]: GET testlog from manix (192.168.0.6) user = jim at k2.lsl.com (192.168.24.2).
Jun 11 00:12:18 gw ftpoxy[17608]: exit user = jim source = k2.lsl.com (192.168.24.2) dest = manix (192.168.0.6) cmds = 13 in = 11221372 out = 0 duration = 65 sec cpu = 6.86
```

## 19.3 Examining the system log

All of the proxy servers and daemons supplied with T.Rex write messages to the system log. Each message contains the name of the program to identify the source of the message. This can be used to search for messages issued by a specific program on a specified date. The messages written by T.Rex proxies and daemons use the following names: ftpoxy, genproxy, httpd, T.Rexmon, procmon, rpcproxy, smwrap, smwrapd, synmon, tnproxy and webgate.

UNIX provides simple commands that allow the systems administrator to customize reports based on system log data. The data can be examined by date, type of service or type of message. The following sections show how you can use basic UNIX commands to examine system activity.

T.Rex also provides a number of utility programs that produce reports for each proxy.

### 19.3.1 Displaying the syslog in real time

The **display\_syslog** script displays the system log in real-time in an X-window using black letters on a yellow background. The window has a 500 line buffer and a scroll bar to view the history of alerts. The script is found in the /usr/local/etc/directory. You can invoke this command as follows:

```
# /usr/local/etc/display_alerts
```

You can also use the X-window display command to redirect the syslog messages to another system that is running X-window.

If you are not using X-window the following command will display all security alerts on the system console as they occur.

For AIX:

```
tail -f /var/adm/syslog
```

For Solaris:

```
tail -f /var/adm/messages
```

### 19.3.2 Selecting messages by type and date

The following command will allow you to view all the genproxy messages entries for June 10. The output is directed into the more command so that you can scroll through the data.

```
# cat /var/adm/syslog | grep "Jun 10" | grep "genproxy" | more
```

```
Jun 10 09:32:18 gw genproxy[18290]: connect src_host = s4.lsl.com (192.168.24.4) src_port = 119 dest_host = 198.65.128.14 dest_port = 119
Jun 10 09:41:41 gw genproxy[18290]: disconnect src_host = s4.lsl.com (192.168.24.4) src_port = 119, dest_host = 198.65.128.14 dest_port = 119 input = 89400 bytes output = 3239 bytes, duration = 00:09:25 cpu = 0.28 sec
Jun 10 11:23:25 gw genproxy[11681]: connect src_host = s3.lsl.com (192.168.24.3) src_port = 119 dest_host = 198.65.128.14 dest_port = 119
Jun 10 11:25:04 gw genproxy[11681]: disconnect src_host = s3.lsl.com (192.168.24.3) src_port = 119, dest_host = 198.65.128.14 dest_port = 119 input = 178398 bytes output = 307 bytes, duration = 00:01:39 cpu = 0.31 sec
```

This example shows four messages issued by the general proxy server genproxy. There are two pairs of messages representing the connection and disconnection of the genproxy server. These sessions were initiated by two internal news readers requesting access to an external news server. The disconnect messages show the name and IP address of the computer that was the source of the request, the name and IP address of the destination host (server) the source port number and the destination port number. The number of input bytes and output bytes are recorded, as well as the duration of the session and the amount of cpu time used to do the work.

A program is provided to summarize and report all genproxy activity.

In general, any monitoring you can normally do under UNIX can be done on the T.Rex Firewall System.

### 19.3.2 Displaying Security Alerts in real time

The **display\_alerts** script displays security alerts in real-time in an X-window using red letters on a white background. The window has a 500 line buffer and a scroll bar to view the history of alerts.

The script is found in the /usr/local/etc/directory. You can invoke this command as follows:

```
# /usr/local/etc/display_alerts
```

You can also use the X-window display command to redirect the security alerts to another system that is running X-window.

If you are not using X-window the following command will display all security alerts on the system console as they occur.

For AIX:

```
tail -f /var/adm/syslog | grep "Security"
```

For Solaris:

```
tail -f /var/adm/messages | grep "Security"
```

### 19.3.3 Analyzing Security Alerts

The following example shows security alerts generated by FTP and telnet proxies.

```
Jun  8 14:58:36 gw ftpproxy[15224]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 129.7.1.3 aka jane.uh.edu.
Jun  8 15:18:37 gw ftpproxy[13717]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 129.7.1.3 aka jane.uh.edu.
Jun  8 15:26:10 gw ftpproxy[13211]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 129.7.1.3 aka jane.uh.edu.
Jun  8 16:05:50 gw tnpnproxy[11203]: Security Alert: multiple authentication failures: source host = jane.uh.edu (129.7.1.3) user(s) =
bin root nobody guest
Jun  9 08:11:15 gw ftpproxy[17632]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 129.7.1.3 aka jane.uh.edu.
Jun  9 08:12:35 gw tnpnproxy[17633]: Security Alert: multiple authentication failures: source host = jane.uh.edu (129.7.1.3) user(s) =
root guest nobody anonymous
Jun  9 09:03:04 gw ftpproxy[15657]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 198.65.130.22 aka lsli.sccsi.com.
Jun  9 09:03:38 gw ftpproxy[15659]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 198.65.130.22 aka lsli.sccsi.com.
Jun  9 15:14:04 gw tnpnproxy[13759]: Security Alert: multiple authentication failures: source host = lsli4.lsli.com (192.168.24.4)
user(s) = ellana ellana exit
Jun  9 16:23:50 gw ftpproxy[11762]: Security Alert: unauthorized ftp attempt to 198.65.130.22 from 146.83.244.123 aka mogul.tasco.cl.
Jun 10 11:19:32 gw tnpnproxy[11678]: Security Alert: multiple authentication failures: source host = lsli2.lsli.com (192.168.24.2)
user(s) = nobody root guest bin
Jun 10 16:02:57 gw tnpnproxy[15954]: Security Alert: multiple authentication failures: source host = nuchat.sccsi.com (198.65.128.16)
user(s) = ellana ellana jim jim
Jun 13 17:30:33 gw ftpproxy[12934]: Security Alert: ftp attempt to unsecured port 198.65.130.22 using secure source IP address
192.168.24.2 (k2.lsli.com).
Jun 13 17:30:47 gw tnpnproxy[12935]: Security Alert: telnet attempt to unsecured port 198.65.130.22 using secure source IP address
192.168.24.2 (k2.lsli.com)
```

Some of the ftpproxy entries may be legitimate attempts to retrieve information from your organization or a real break-in attempt. Only you can tell for sure. Notice you have the date, time and IP address of the host used to ftp to your gateway.

Some of the telnet attempts look like actual break-in attempts. For example, on June 8 at 16:05 someone tried to login as bin, root, nobody and guest. None of these ID represent legitimate login IDs. A similar incident was recorded on June 10 at 11:11.

On June 13 there were two attempts to use IP address spoofing to gain entry to the protected network. The ftp and telnet proxies both detected that the packets contained source IP addresses for protected hosts but the packets were received from an unsecured port! This is a break-in attempt.

If you want to find out more information regarding an unauthorized ftp attempt you can use the nslookup, whois traceroute, and dig commands.

### 19.3.4 Reporting SOCKD activity

T.Rex provides the **sockdsum** program to summarize the use of SOCKS on the T.Rex firewall. The sockdsum program reads the system log and produces reports showing how the SOCKS program is being used. It produces the following reports:

#### **Socks Daemon Summary Report:**

The summary report provides the name of the logfile used and the total number of records of any type found in the logfile. Any incomplete records in the log are reported as truncated records and are not reported. The date and time stamp of the first and last record found in the log file are listed. This can be used to certify that you used the correct log for the report. The total number of records in the log written by sockd, the total number of connection requests and the number of hosts involved in socks activity, and the total number of bytes sent and received from the secure network are reported. The totals in the summary report are for all the sockd activity found in the specified log.

#### **Top Hosts by Bytes Sent:**

The Top Hosts by Bytes sent lists the hosts sorted by the number of bytes sent to the T.Rex firewall. The first column lists the bytes sent by the host. The second column lists the number of connection requests made by the host. Only hosts on a secure network are allowed to initiate socksified connections. Thus, all unsecured hosts should have a zero in this column. The third column lists the number of bytes received by the host. The fourth column shows the number of connection requests received by the host.

If the sockdsum program was run with the **-t nnn** parameter the Top Hosts reports will reflect the filtered output.

#### **Top Hosts by Connect Requests**

The Top Hosts by Connect Requests is similar to the Top Hosts by Bytes Sent but is sorted by the number of connection requests initiated by the host. Only secure hosts should have none zero values in column two.

#### **Top Hosts by Bytes Received**

The Top Hosts by Bytes Received is the same as the Top Hosts by Bytes Sent except it is sorted by the number of Bytes Received.

#### **Top Hosts by Received Connects**

The Top Hosts by Received Connects is the same as the Top Hosts by Bytes Sent except it is sorted by the number of connection requests received by the host. Secure hosts should have zero values in the Received Connects column.

Parameters are available to control the amount of detail reported. The syntax of the sockdsum command is as follows:

**sockdsum [-l logfile] [-t nnn]**

- l logfile** This is used to specify the name of the log file. If not specified then it defaults to /var/adm/syslog.
- t nnn** This is used to limit the top user reported to the value specified.

### 19.3.5 Sample Sockdsum Output

The following sockdsum report was produced using the following command to limit the number of systems reported to the top 12 in each category.

**# sockdsum -t 12**

#### SOCKS Daemon Summary Report

log file name: /var/adm/syslog

```
total records in log file      = 3265
number of truncated records    = 0
date of first record          = Oct 16 18:01:01
date of last record           = Oct 23 10:15:49
total number of sockd records = 1573
number of unique hosts        = 39
number of connect messages    = 520
total bytes sent               = 132257
total bytes received           = 5145995
total bytes sent & received    = 5278252
```

#### Top hosts by bytes sent

bytes sent	initiated connects	bytes received	received connects	host
1943492	0	101	2	ftp3.netscape.com
1369279	0	13716	29	Fe3.rust.net
605230	0	27053	117	www.nj.com
203178	0	3588	2	Jane.UH.EDU
105722	0	9894	51	charlotte05.va.pubnix.com
96301	429	3675072	0	system7.lsli.com
86523	0	1498	9	www.netmanage.com
83898	0	9211	39	www2.netscape.com
81418	0	1533	9	155.136.64.225
80307	0	11168	50	fs1.houston.sccsi.com
60382	0	8928	33	www11.netscape.com
54110	0	220	1	www1.access.digex.net

totals:

4769840	429	3761982	342	12
---------	-----	---------	-----	----

#### Top hosts by connect requests

bytes sent	initiated connects	bytes received	received connects	host
96301	429	3675072	0	system7.lsli.com
35956	91	1470923	0	system12.lsli.com
1369279	0	13716	29	Fe3.rust.net
605230	0	27053	117	www.nj.com
203178	0	3588	2	Jane.UH.EDU
105722	0	9894	51	charlotte05.va.pubnix.com
1943492	0	101	2	ftp3.netscape.com
83898	0	9211	39	www2.netscape.com
81418	0	1533	9	155.136.64.225
86523	0	1498	9	www.netmanage.com
80307	0	11168	50	fs1.houston.sccsi.com
54110	0	220	1	www1.access.digex.net

totals:				
4745414	520	5223977	309	12

#### Top hosts by bytes received

bytes sent	initiated connects	bytes received	received connects	host
96301	429	3675072	0	system7.lsli.com
35956	91	1470923	0	system12.lsli.com
605230	0	27053	117	www.nj.com
1369279	0	13716	29	Fe3.rust.net
80307	0	11168	50	fs1.houston.sccsi.com
105722	0	9894	51	charlotte05.va.pubnix.com
83898	0	9211	39	www2.netscape.com
60382	0	8928	33	www11.netscape.com
42013	0	7353	29	www.lsli.com
42804	0	4848	4	zeppo.ncsa.uiuc.edu
23936	0	4318	17	www9.netscape.com
45670	0	4197	18	www8.yahoo.com

totals:				
2591498	520	5246681	387	12

#### Top hosts by receives

bytes sent	initiated connects	bytes received	received connects	host
605230	0	27053	117	www.nj.com

105722	0	9894	51	charlotte05.va.pubnix.com
80307	0	11168	50	fs1.houston.sccsi.com
83898	0	9211	39	www2.netscape.com
60382	0	8928	33	www11.netscape.com
1369279	0	13716	29	Fe3.rust.net
42013	0	7353	29	www.lsli.com
45670	0	4197	18	www8.yahoo.com
23936	0	4318	17	www9.netscape.com
23642	0	3983	15	198.95.249.77
46016	0	2467	12	noc.usvi.net
33385	0	534	10	cert.org
totals:				
2519480	0	102822	420	12

### 19.3.6 Searching for details on SOCKD activity

UNIX comes with a variety of commands that can be used to filter and display specific activity recorded in syslog. For example, if you want to examine sockd activity for a specific day on behalf of a particular host enter the following commands.

```
#cat /var/adm/syslog | grep "Dec 10" | grep "uno"
```

```
Dec 10 17:34:52 gw sockd[13585]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (ftp)
Dec 10 17:35:15 gw sockd[11282]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3227)
Dec 10 17:35:16 gw sockd[11282]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3227).
Dec 10 17:35:16 gw sockd[11282]: 0 bytes from uno.lsli.com, 747 bytes from vivaldi.inoc.dl.nec.com
Dec 10 17:35:26 gw sockd[11283]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3230)
Dec 10 17:35:28 gw sockd[11283]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3230).
Dec 10 17:35:28 gw sockd[11283]: 0 bytes from uno.lsli.com, 1437 bytes from vivaldi.inoc.dl.nec.com
Dec 10 17:36:08 gw sockd[13079]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3232)
Dec 10 17:36:10 gw sockd[13079]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3232).
Dec 10 17:36:10 gw sockd[13079]: 0 bytes from uno.lsli.com, 1269 bytes from vivaldi.inoc.dl.nec.com
Dec 10 17:36:50 gw sockd[13080]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3236)
Dec 10 17:36:52 gw sockd[13080]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3236).
Dec 10 17:36:52 gw sockd[13080]: 0 bytes from uno.lsli.com, 3070 bytes from vivaldi.inoc.dl.nec.com
Dec 10 17:37:12 gw sockd[13081]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3237)
Dec 10 17:37:14 gw sockd[13081]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3237).
Dec 10 17:37:14 gw sockd[13081]: 0 bytes from uno.lsli.com, 1269 bytes from vivaldi.inoc.dl.nec.com
Dec 10 17:37:45 gw sockd[13082]: connected -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3241)
Dec 10 17:39:45 gw sockd[13082]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (3241).
Dec 10 17:39:45 gw sockd[13082]: 0 bytes from uno.lsli.com, 177610 bytes from vivaldi.inoc.dl.nec.com
Dec 10 17:41:08 gw sockd[13585]: terminated -- Connect from jim(jim)@uno.lsli.com to vivaldi.inoc.dl.nec.com (ftp).
Dec 10 17:41:08 gw sockd[13585]: 187 bytes from uno.lsli.com, 2417 bytes from vivaldi.inoc.dl.nec.com
```

The sockd records beginning on Dec 10 at 17:34:52 shows data collected on the firewall produced during an **ftp** session initiated by user jim on the client host uno.lsli.com. Each ftp sub command such as: dir, cd, dir, bin and get issued by the user produces an sockd message line. Some of the ftp sub commands such as dir produce output that the sockd records. The get command recorded at 17:39:45 shows the total number of bytes retrieved. The process id on the ftp server is displayed within the parenthesis. This process id matches the system log entry recorded by the socks ftp client program (rftp) on the client host.



### 19.3.7 Reporting rftp activity

The following command was issued on host uno.lsl.com to show the information written in the client host's system log by the socks ftp client (rftp). This example shows the rftp client information that corresponds with the sockd entries for the rftp session recorded in group five above.

```
#cat /var/adm/syslog | grep "rftp" | grep "Feb 10"
```

```
Dec 10 17:36:04 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (ftp) using sockd at 192.168.220.1
Dec 10 17:36:27 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (3227) using sockd at 192.168.220.1
Dec 10 17:36:38 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (3230) using sockd at 192.168.220.1
Dec 10 17:37:20 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (3232) using sockd at 192.168.220.1
Dec 10 17:38:02 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (3236) using sockd at 192.168.220.1
Dec 10 17:38:23 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (3237) using sockd at 192.168.220.1
Dec 10 17:38:57 lsl2 rftp[15886]: connect() from jim(jim) to vivaldi.inoc.dl.nec.com (3241) using sockd at 192.168.220.1
Dec 10 17:40:57 lsl2 rftp[15886]: Received remote file cops-1.02.tar.Z as cops-1.02.tar.Z -- 177610 bytes
```

During this session the user jim downloaded the remote file cops-1.02.tar.Z. The file was stored with the same name and contained 177,610 bytes. The first message shows the start of the ftp session. Subsequent messages were produced for each ftp subcommand such as: dir, cd, bin, and get.

## 19.4 Reporting FTP activity

T.Rex provides the **ftprpt** utility for reporting FTP proxy usage. The ftprpt program reads the system log and produces reports showing how ftp is being used. It produces the following reports:

Ftproxy summary report,  
top users sorted by the number of bytes sent,  
top users sorted by the number of bytes received,  
top users sorted by connect time,  
top users sorted by cpu time used.

Parameters are available to control the amount of detail reported. The syntax of the ftprpt command is as follows:

**ftprpt [-l logfile] [-t nnn]**

**-l logfile** This is used to specify the name of the log file. If not specified then it defaults to /var/adm/syslog.

**-t nnn** This is used to limit the top users reported to the value specified.

### Sample ftprpt output

The following example shows the reporting of the top ten ftp users. command.

```
$ ftprpt -l /var/admsyslog -t 10
```

Ftproxy V 2.1 Summary Report

log file name: test1

```
total records in log file      = 2702
number of truncated records   = 0
date of first record          = Dec 29 16:48:38
date of last record           = Jan  4 18:01:52
total number of ftproxy records = 2702
total number of GET commands   = 502
total number of PUT commands   = 2137
total number of ftp commands   = 13383
number of unique addresses     = 11
number of ftproxy messages     = 2702
number of ftp sessions         = 17
total bytes sent (PUT)         = 4643070976
total bytes received (GET)     = 686093504
total bytes sent and received = 5329164288
total duration                 = 42535 sec
total cpu time                 = 2263.09
```

```
Mega Bytes per cpu second    = 2.25
Average file size for PUT    = 2172705.2
Average file size for GET    = 1366720.1
Average connect time per user = 3866.8
Average connect time per session = 2502
```

Top users by bytes sent

bytes sent	bytes received	number PUTS	number GETS	number commands	connect time	cpu time	address
4312667648	634379136	412	249	1789	15298	1548.68	groucho@marx.lsli.com
192667568	31379136	111	105	415	1328	291.68	gracie@allen.lsli.com
116873096	1075827	1535	4	9991	17362	204.29	bob@hope.lsli.com
11267572	3769135	31	24	394	1328	48.63	ace@ventura.lsli.com
3716534	379135	22	46	539	1298	52.18	ted@airdale.lsli.com
3176772	9937135	19	25	43	217	54.61	harpo@marx.lsli.com
2656772	1379135	6	31	94	298	48.36	wc@fields.lsli.com
45056	9503	1	0	33	181	0.53	zeppo@marx.lsli.com
0	51796	0	16	58	598	0.82	10.20.30.40@marx.lsli.com
0	568361	0	1	15	1698	2.21	phill@silvers.lsli.com

```
totals:
4643071488 682928320 2137 501 13371 39606 2251.99 10
```

# Top users by bytes received

bytes sent	bytes received	number PUTS	number GETS	number commands	connect time	cpu time	address
4312667648	634379136	412	249	1789	15298	1548.68	groucho@marx.lsl.i.com
192667568	31379136	111	105	415	1328	291.68	gracie@allen.lsl.i.com
3176772	9937135	19	25	43	217	54.61	harpo@marx.lsl.i.com
11267572	3769135	31	24	394	1328	48.63	ace@ventura.lsl.i.com
0	3165168	0	1	12	2929	11.10	george@allen.lsl.i.com
2656772	1379135	6	31	94	298	48.36	wc@fields.lsl.i.com
116873096	1075827	1535	4	9991	17362	204.29	bob@hope.lsl.i.com
0	568361	0	1	15	1698	2.21	phill@silvers.lsl.i.com
3716534	379135	22	46	539	1298	52.18	ted@airdale.lsl.i.com
0	51796	0	16	58	598	0.82	10.20.30.40@marx.lsl.i.com
totals:							
4643026432	686084032	2136	502	13350	42354	2262.56	10

# Top users by connect time

bytes sent	bytes received	number PUTS	number GETS	number commands	connect time	cpu time	address
116873096	1075827	1535	4	9991	17362	204.29	bob@hope.lsl.i.com
4312667648	634379136	412	249	1789	15298	1548.68	groucho@marx.lsl.i.com
0	3165168	0	1	12	2929	11.10	george@allen.lsl.i.com
0	568361	0	1	15	1698	2.21	phill@silvers.lsl.i.com
11267572	3769135	31	24	394	1328	48.63	ace@ventura.lsl.i.com
192667568	31379136	111	105	415	1328	291.68	gracie@allen.lsl.i.com
3716534	379135	22	46	539	1298	52.18	ted@airdale.lsl.i.com
0	51796	0	16	58	598	0.82	10.20.30.40@marx.lsl.i.com
2656772	1379135	6	31	94	298	48.36	wc@fields.lsl.i.com
3176772	9937135	19	25	43	217	54.61	harpo@marx.lsl.i.com
totals:							
4643026432	686084032	2136	502	13350	42354	2262.56	10

# Top users by cpu time

bytes sent	bytes received	number PUTS	number GETS	number commands	connect time	cpu time	address
4312667648	634379136	412	249	1789	15298	1548.68	groucho@marx.lsl.i.com
192667568	31379136	111	105	415	1328	291.68	gracie@allen.lsl.i.com
116873096	1075827	1535	4	9991	17362	204.29	bob@hope.lsl.i.com
3176772	9937135	19	25	43	217	54.61	harpo@marx.lsl.i.com
3716534	379135	22	46	539	1298	52.18	ted@airdale.lsl.i.com
11267572	3769135	31	24	394	1328	48.63	ace@ventura.lsl.i.com
2656772	1379135	6	31	94	298	48.36	wc@fields.lsl.i.com
0	3165168	0	1	12	2929	11.10	george@allen.lsl.i.com
0	568361	0	1	15	1698	2.21	phill@silvers.lsl.i.com
0	51796	0	16	58	598	0.82	10.20.30.40@marx.lsl.i.com
totals:							
4643026432	686084032	2136	502	13350	42354	2262.56	10

## 19.5 Genproxy Activity Reports

T.Rex provides the **genproxsum** program to summarize the use of the general proxy server. The genproxsum program reads the system log and produces reports showing how the general proxy program is being used. It produces the following reports:

- genproxy summary statistics
- top hosts by bytes sent
- to hosts by connect requests
- top hosts by bytes received
- to hosts by received connects

### Sample Genproxsum Output

The following example shows the top genproxy uses. The multiple protected hosts were allowed to use a news reader to access an external News Server. The genproxy.conf used to generate the sample output contained the following rules:

```
permit from 192.168.220.* NNTP to 198.65.128.14 NNTP
deny * nntp
```

```
$ genproxsum
```

#### General Proxy Summary Report

```
log file name: /var/adm/syslog
```

```
total records in log file      = 10012
total bytes in log file        = 1132372
number of truncated records    = 1
date of first record           = Apr  1 10:33:03
date of last record            = Apr 17 13:09:01
number of genproxy records     = 492
total size of genproxy records = 72185
number of unique hosts         = 6
number of connect messages     = 176
number of disconnect messages  = 176
number of Security Alerts      = 26
total bytes sent                = 0
total bytes received            = 0
total duration                  = 28822
```

#### Top hosts by bytes sent

bytes sent	initiated connects	bytes received	received connects	PORT	duration	IP address	hostname
4272311	134	110983	0	119	23706	192.168.220.4	zeppo.lsli.com
740213	34	60558	0	119	4924	192.168.220.3	unknown
207068	8	341	0	119	192	192.168.220.2	groucho.lsli.com
171547	0	5012709	172	119	28708	198.65.128.14	198.65.128.14
222	0	1932	2	23	55	129.7.1.3	129.7.1.3

113	0	204951	2	23	59	192.168.220.2	192.168.220.2
totals:							
5391474	176	5391474	176		57644	6	

Top hosts by connect requests

bytes sent	initiated connects	bytes received	received connects	PORT	duration	IP address	hostname
4272311	134	110983	0	119	23706	192.168.220.4	zeppo.lsli.com
740213	34	60558	0	119	4924	192.168.220.3	unknown
207068	8	341	0	119	192	192.168.220.2	groucho.lsli.com
171547	0	5012709	172	119	28708	198.65.128.14	198.65.128.14
222	0	1932	2	23	55	129.7.1.3	129.7.1.3
113	0	204951	2	23	59	192.168.220.2	192.168.220.2
totals:							
5391474	176	5391474	176		57644	6	

Top hosts by bytes received

bytes sent	initiated connects	bytes received	received connects	PORT	duration	IP address	hostname
171547	0	5012709	172	119	28708	198.65.128.14	198.65.128.14
113	0	204951	2	23	59	192.168.220.2	192.168.220.2
4272311	134	110983	0	119	23706	192.168.220.4	zeppo.lsli.com
740213	34	60558	0	119	4924	192.168.220.3	unknown
222	0	1932	2	23	55	129.7.1.3	129.7.1.3
207068	8	341	0	119	192	192.168.220.2	groucho.lsli.com
totals:							
5391474	176	5391474	176		57644	6	

Top hosts by receives

bytes sent	initiated connects	bytes received	received connects	PORT	duration	IP address	hostname
171547	0	5012709	172	119	28708	198.65.128.14	198.65.128.14
113	0	204951	2	23	59	192.168.220.2	192.168.220.2
222	0	1932	2	23	55	129.7.1.3	129.7.1.3
4272311	134	110983	0	119	23706	192.168.220.4	zeppo.lsli.com
740213	34	60558	0	119	4924	192.168.220.3	unknown
207068	8	341	0	119	192	192.168.220.2	groucho.lsli.com
totals:							
5391474	176	5391474	176		57644	6	

## 19.6 Reporting Secure Mail Wrapper activity

The **smrpt** program reads the system log and produces reports showing how e-mail is being used. It produces the following reports:

e-mail summary report,  
top users sorted by the number of messages received,

top users sorted by the number of bytes received,  
top users sorted by the number of messages sent,  
top users sorted by the number of bytes sent.

Parameters are available to control the amount of detail reported. The syntax of the `smrpt` command is as follows:

**`smrpt [-d domain] [-l logfile] [-m nnnnnn] [-t nnn]`**

- d domain** This is used to list the e-mail addresses for a single domain.
- l logfile** This is used to specify the name of the log file. If not specified then it defaults to `/var/adm/syslog`.
- m nnnnnn** This is used to specify the maximum number of e-mail addresses to sort. The default value is 16,000 e-mail addresses, which is large enough for all but the very largest organizations.
- t nnn** This is used to limit the top users reported to the value specified.

### 19.6.1 Sample `smrpt` output

The following example shows the reporting of the top ten e-mail users. `command`.

```
$ smrpt -l /var/admsyslog -t 10
```

Sendmail Wrapper Summary Report

log file name: `/var/adm/syslog`

```
total records in log file      = 11513
number of truncated records    = 0
date of first record           = Dec 29 17:12:05
date of last record            = Mar 24 12:19:57
total number of smwrap records = 236
number of unique addresses     = 29
number of E-mail messages     = 47
total E-mail bytes             = 31765
```

Top senders by bytes sent

bytes sent	msg sent	bytes received	msg received	address
9210	7	0	0	jay@gw.lsli.com
5703	13	0	0	jim@gw.lsli.com
2901	3	3103	3	root@system2.lsli.com
2855	5	0	0	steve@nuchat.sccsi.com
2825	5	0	0	jay@jetson.uh.edu
2058	2	10142	18	jim@system2.lsli.com
2003	1	3047	3	ted@system2.lsli.com
1990	1	571	1	ellana@system2.lsli.com
490	2	2256	3	ellana@system2

374	2	0	0	system2
totals:				
30409	41	19119	28	10

#### Top senders by messages sent

bytes sent	msg sent	bytes received	msg received	address
5703	13	0	0	jim@gw
9210	7	0	0	jay@gw
2855	5	0	0	steve@nuchat.sccsi.com
2825	5	0	0	jay@jetson.uh.edu
2901	3	3103	3	root@system2
2058	2	10142	18	jim@system2
490	2	2256	3	ellana@system2
374	2	0	0	system2
2003	1	3047	3	ted@system2
1990	1	571	1	ellana@system2
totals:				
30409	41	19119	28	10

#### Top receiver by bytes received

bytes sent	msg sent	bytes received	msg received	address
2058	2	10142	18	jim@system2
2901	3	3103	3	root@system2
2003	1	3047	3	ted@system2
490	2	2256	3	ellana@system2
0	0	2080	2	alex@system2
0	0	1996	2	postmaster@system2
0	0	1509	1	jay@system2
0	0	1401	3	jim1@system2
0	0	1156	2	jim2@system2
0	0	1156	2	jim3@system2
totals:				
7452	8	27846	39	10

#### Top receiver by messages received

bytes sent	msg sent	bytes received	msg received	address
2058	2	10142	18	jim@system2
2901	3	3103	3	root@system2
2003	1	3047	3	ted@system2
490	2	2256	3	ellana@system2
0	0	1401	3	jim1@system2

0	0	1996	2	postmaster@system2
0	0	1156	2	jim2@system2
0	0	2080	2	alex@system2
0	0	1156	2	jim3@system2
0	0	1509	1	jay@system2
totals:				
7452	8	27846	39	10

## 19.6.2 Reporting Exceptions

The smrptx program reads the system log and produces the following exception reports:

**VRFY:** All attempts to use the VRFY command to obtain user IDs on the firewall are reported. Excessive use of the VRFY command from an unprotected host can be the sign of a cracker.

**EXPN:** All attempts to use the EXPN command to extract user information are reported.

**Warning:** All smwrap and smwrapd warning messages are printed. This helps find configuration and systems problems.

**Error:** All the Error messages issued by either smwrap or smwrapd are reported.

**Security Alert:** All the Security Alerts issued by smwrap or smwrapd.

**Sequestered:** All the files that can not be delivered and the reason they were sequestered are reported.

Parameters are available to control the amount of detail reported. The syntax of the smrptx command is as follows:

**smrptx [-d] [-l logfile] [-t nnn]**

- d** This is used to produce detailed exception reports
- l logfile** This is used to specify the name of the log file. If not specified then it defaults to /var/adm/syslog.
- t nnn** This is used to limit the top users reported to the value

Sendmail Wrapper Exception Report



log file name: /var/adm/syslog

```
total records in log file      = 11477
number of truncated records   = 0
date of first record          = Dec 29 17:12:05
date of last record           = Mar 24 07:21:11
number of smwrap records      = 236
number of smwrapd records     = 163
total messages delivered      = 9
number of VRFY attempts       = 10
number of Error messages      = 39
number of EXPN attempts       = 3
number of Security Alerts     = 0
number of Sequestered files   = 3
number of Warning messages    = 13
```

Listing of all VRFY attempts to port 25

```
Mar  9 12:34:46 lsli2 smwrap[17681]: vrfy request from lsli2 (192.168.0.2).
Mar  9 12:34:51 lsli2 smwrap[17681]: VRFY JIM request from lsli2 (192.168.0.2).
Mar  9 12:35:14 lsli2 smwrap[17681]: VRFY root request from lsli2 (192.168.0.2).
Mar  9 16:51:45 lsli2 smwrap[17913]: VRFY jack request from lsli2 (192.168.0.2).
Mar 10 09:40:18 lsli2 smwrap[15586]: vrfy ellana request from gw (192.168.0.1).
Mar 23 15:50:06 lsli2 smwrap[14521]: vrfy alex request from gw (192.168.0.1).
Mar 23 15:50:13 lsli2 smwrap[14521]: vrfy jay request from gw (192.168.0.1).
Mar 23 15:50:19 lsli2 smwrap[14521]: vrfy ellana request from gw (192.168.0.1).
Mar 23 15:50:39 lsli2 smwrap[14521]: vrfy goofy request from gw (192.168.0.1).
Mar 23 15:51:24 lsli2 smwrap[20666]: vrfy goofy request from gw (192.168.0.1).
Mar 23 15:51:24 lsli2 smwrap[20666]: vrfy goofy request from gw (192.168.0.1).
```

Listing of all EXPN attempts to port 25

```
Mar  9 16:51:55 lsli2 smwrap[17913]: EXPN request from lsli2 (192.168.0.2).
Mar  9 16:52:00 lsli2 smwrap[17913]: EXPN jim request from lsli2 (192.168.0.2).
Mar 10 09:40:25 lsli2 smwrap[15586]: expn request from gw (192.168.0.1).
Mar 10 09:40:25 lsli2 smwrap[15586]: expn request from gw (192.168.0.1).
```

Listing of all Warning messages from smwrap & smwrapd

```
Mar 10 12:44:33 lsli2 smwrap[18468]: Warning maxchildren = '12' is > 10. See line 8 of
/etc/security/smwrap.conf.
Mar 10 17:08:16 lsli2 smwrap[12596]: Warning maxchildren = '12' is > 10. See line 8 of
/etc/security/smwrap.conf.
Mar 10 17:43:02 lsli2 smwrapd[9820]: Warning maxchildren = '12' is > 10. See line 8 of
/etc/security/smwrap.conf.
Mar 11 08:42:55 lsli2 smwrap[15729]: Warning maxchildren = '12' is > 10. See line 8 of
/etc/security/smwrap.conf.
Mar 11 08:58:19 lsli2 smwrapd[14242]: Warning maxchildren = '12' is > 10. See line 8 of
/etc/security/smwrap.conf.
Mar 11 09:04:14 lsli2 smwrapd[14796]: Warning maxchildren = '12' is > 10. See line 8 of
/etc/security/smwrap.conf.
Mar 13 10:24:28 lsli2 smwrapd[14080]: Warning: maxbytes = '0' on line 6 of
/etc/security/smwrap.conf. Unable to deliver mail.
Mar 13 11:36:10 lsli2 smwrapd[17568]: Warning: maxreceipts = '00' on line 7 of
/etc/security/smwrap.conf. Unable to deliver mail. Exiting.
Mar 13 12:07:57 lsli2 smwrapd[12795]: Warning: maxchildren = '0' on line 8 of
/etc/security/smwrap.conf. Unable to deliver mail. Exiting.
Mar 13 17:54:37 lsli2 smwrapd[13086]: Warning: maxbytes = '000000' on line 6 of
```

```

/etc/security/smwrap.conf. Unable to deliver mail. Exiting.
Mar 22 13:46:20 lsli2 smwrapd[16060]: Warning: maxbytes = '000000' on line 6 of
/etc/security/smwrap.conf. Unable to deliver mail. Exiting.
Mar 22 13:51:11 lsli2 smwrapd[16844]: Warning: maxreceipts = '00' on line 7 of
/etc/security/smwrap.conf. Unable to deliver mail. Exiting.
Mar 22 13:53:16 lsli2 smwrapd[16850]: Warning maxchildren = '1000' is > 12. See line 8 of
/etc/security/smwrap.conf.
Mar 22 13:53:16 lsli2 smwrapd[16850]: Warning maxchildren = '1000' is > 12. See line 8 of
/etc/security/smwrap.conf.

```

#### Listing of all Error messages from smwrap & smwrapd

```

Mar 14 09:43:46 lsli2 smwrapd[17460]: Error: cannot open 'smwrapABCDEF': Is a directory
Mar 14 09:44:46 lsli2 smwrapd[17462]: Error: cannot open 'smwrapABCDEF': Is a directory
Mar 14 09:45:57 lsli2 smwrapd[17469]: Error: cannot open 'smwrapABCDEF': Permission denied
Mar 14 10:40:05 lsli2 smwrapd[15251]: Error: cannot open 'smwrapGOOD1': Permission denied
Mar 14 10:40:05 lsli2 smwrapd[17558]: Error: cannot open 'smwrap2long': Permission denied
Mar 14 11:48:22 lsli2 smwrapd[16615]: Error: cannot open 'smwrap2long': Permission denied
Mar 14 12:07:45 lsli2 smwrapd[16388]: Error: cannot open 'smwrap22long': Permission denied
Mar 14 12:12:25 lsli2 smwrapd[16407]: Error: cannot open 'smwrap22long': Permission denied
Mar 14 16:37:52 lsli2 smwrapd[16555]: Error: cannot open 'smwrapbadfrom': Permission denied
Mar 14 16:51:28 lsli2 smwrapd[17856]: Error: cannot open 'smwrapGOOD1': Permission denied
Mar 14 16:51:28 lsli2 smwrapd[17857]: Error: cannot open 'smwrapGOOD1': Permission denied
Mar 14 16:52:25 lsli2 smwrapd[17862]: Error: cannot open 'smwrapbadfrom': Permission denied
Mar 15 08:43:46 lsli2 smwrapd[12356]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 15 08:43:46 lsli2 smwrapd[12357]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 15 09:43:57 lsli2 smwrapd[12428]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 15 09:43:57 lsli2 smwrapd[12429]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 15 09:43:57 lsli2 smwrapd[12430]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 15 09:44:57 lsli2 smwrapd[14994]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 15 12:09:07 lsli2 smwrapd[17881]: Error: cannot open 'smwrapbadtoaddr': Permission denied
Mar 16 15:44:54 lsli2 smwrapd[10830]: Error: cannot fork. Unable to deliver smwrapGOOD1e: Resource
temporarily unavailable
Mar 16 16:45:05 lsli2 smwrapd[18753]: Error: cannot fork. Current # pids = 10. Unable to deliver
smwrapGOOD1j: Resource temporarily unavailable. Will retry.
Mar 22 11:24:56 lsli2 smwrapd[13753]: Error: cannot open 'smwrapbadguy': Is a directory
Mar 22 13:40:17 lsli2 smwrapd[16811]: Error: No timeout record found in /etc/security/smwrap.conf.
Mar 22 13:41:35 lsli2 smwrapd[16814]: Error: No userid record found in /etc/security/smwrap.conf.
Mar 22 13:42:22 lsli2 smwrapd[16048]: Error: user name 'ji' on line 3 of /etc/security/smwrap.conf
is not valid user.
Mar 22 13:43:02 lsli2 smwrapd[16050]: Error: No groupid record found in /etc/security/smwrap.conf.
Mar 22 13:43:36 lsli2 smwrapd[16052]: Error: group name 'syste' on line 4 of
/etc/security/smwrap.conf is not valid group.
Mar 22 13:44:12 lsli2 smwrapd[16054]: Error: No spoolpath record found in
/etc/security/smwrap.conf.
Mar 22 13:44:58 lsli2 smwrapd[16056]: Error: cannot chdir to spool directory /u/jim/smwra: No such
file or directory
Mar 22 13:45:41 lsli2 smwrapd[16058]: Error: No maxbytes record found in /etc/security/smwrap.conf.
Mar 22 13:49:50 lsli2 smwrapd[16839]: Error: non numeric value '-1000000' specified for maxbytes on
line 6 of /etc/security/smwrap.conf.
Mar 22 13:50:43 lsli2 smwrapd[16842]: Error: No maxreceipts record found in
/etc/security/smwrap.conf.
Mar 22 13:52:39 lsli2 smwrapd[16848]: Error: non numeric value '-10' specified for maxchildren on
line 8 of /etc/security/smwrap.conf.
Mar 22 14:24:16 lsli2 smwrapd[16860]: Error: No smwpdpath record found in
/etc/security/smwrap.conf.
Mar 22 14:27:00 lsli2 smwrapd[16864]: Error: No smtpdpath record found in /etc/security/smwrap.conf.
Mar 22 14:27:31 lsli2 smwrapd[16866]: Error: smtpdpath name '/zusr/lib/sendmail' on line 10 of
/etc/security/smwrap.conf not found.
Mar 22 14:28:10 lsli2 smwrapd[16868]: Error: No undelivpath record found in
/etc/security/smwrap.conf.
Mar 22 14:28:41 lsli2 smwrapd[16870]: Error: undelivpath name '/zu/jim/smwrap/undeliv' on line 11
of /etc/security/smwrap.conf not found.
Mar 22 14:39:44 lsli2 smwrapd[14090]: Error: smwpdpath name '/zusr/local/etc' on line 9 of
/etc/security/smwrap.conf not found.
Mar 22 14:39:44 lsli2 smwrapd[14090]: Error: smwpdpath name '/zusr/local/etc' on line 9 of
/etc/security/smwrap.conf not found.

```

Listing of all Security Alert messages from smwrap & smwrapd

Listing of all Sequestered file messages

```
Mar 22 12:11:43 lsli2 smwrapd[14125]: Sequestered file 'smwrapNOT' does not begin with 'FROM'.
Mar 22 12:11:43 lsli2 smwrapd[18734]: Sequestered file smwrapNOBODY contains no receipients or
data! (No such file or directory)
Mar 22 12:11:45 lsli2 smwrapd[15667]: Sequestered file smwrapbadtoaddr contains a bad RCPT.
Mar 22 12:11:45 lsli2 smwrapd[15667]: Sequestered file smwrapbadtoaddr contains a bad RCPT.
```

## 19.7 Reporting tnproxy activity

T.Rex provides the **tnprpt** utility for reporting telnet proxy usage. The tnprpt program reads the system log and produces reports showing how telnet is being used. It produces the following reports:

- tnproxy summary report,
- top users sorted by the number of bytes sent,
- top users sorted by the number of bytes received,
- top users sorted by connect time,
- top users sorted by cpu time used.

Parameters are available to control the amount of detail reported. The syntax of the tnprpt command is as follows:

**tnprpt [-l logfile] [-t nnn]**

**-l logfile** This is used to specify the name of the log file. If not specified then it defaults to /var/adm/syslog.

**-t nnn** This is used to limit the top users reported to the value specified.

### Sample tnprpt output

The following example shows the reporting of the top ten telnet users. command.

```
$ tnprpt -l /var/admsyslog -t 10
```

Ftproxy V 2.2 Summary Report

log file name: test1

total records in log file	=	164
number of truncated records	=	1
date of first record	=	Dec 29 09:20:29
date of last record	=	Jan 8 14:45:35
total number of tnproxy records	=	164
number of unique addresses	=	13
number of ftp sessions	=	60
total bytes sent	=	22441
total bytes received	=	462260
total bytes sent and received	=	484701
total duration	=	31179 sec
total cpu time	=	22.83

Mega Bytes per cpu second	=	0.02
Average connect time per user	=	2398.4
Average connect time per session	=	519.6

Top users by bytes sent

bytes sent	bytes received	connect time	cpu time	address
---------------	-------------------	-----------------	-------------	---------

4370	121056	5211	6.58	george@burns.lslsli.com
4222	70651	4973	4.05	bob@hope.lslsli.com
2358	37554	2879	2.77	gracie@allen.lslsli.com
2307	42514	4607	3.64	groucho@marx.lslsli.com
2215	15067	1056	0.71	steve@martin.lslsli.com
1516	26145	1060	0.40	jack@bennie.lslsli.com
1471	44927	1091	0.75	phil@silvers.lslsli.com
1446	11281	1030	0.72	ace@ventura.lslsli.com
817	9091	934	0.19	zeppo@marx.lslsli.com
534	59452	1213	1.60	harpo@marx.lslsli.com
totals:				
21256	437738	24054	21.41	10

Top users by bytes received

bytes sent	bytes received	connect time	cpu time	address
4370	121056	5211	6.58	george@burns.lslsli.com
4222	70651	4973	4.05	bob@hope.lslsli.com
534	59452	1213	1.60	harpo@marx.lslsli.com
1471	44927	1091	0.75	phil@silvers.lslsli.com
2307	42514	4607	3.64	groucho@marx.lslsli.com
2358	37554	2879	2.77	gracie@allen.lslsli.com
1516	26145	1060	0.40	jack@bennie.lslsli.com
2215	15067	1056	0.71	steve@martin.lslsli.com
435	13804	1397	0.83	wc@fields.lslsli.com
1446	11281	1030	0.72	ace@ventura.lslsli.com
totals:				
20874	442451	24517	22.05	10

Top users by connect time

bytes sent	bytes received	connect time	cpu time	address
4370	121056	5211	6.58	george@burns.lslsli.com
4222	70651	4973	4.05	bob@hope.lslsli.com
410	9009	4837	0.51	red@skelton.lslsli.com
2307	42514	4607	3.64	groucho@marx.lslsli.com
2358	37554	2879	2.77	gracie@allen.lslsli.com
435	13804	1397	0.83	wc@fields.lslsli.com
534	59452	1213	1.60	harpo@marx.lslsli.com
1471	44927	1091	0.75	phil@silvers.lslsli.com
1516	26145	1060	0.40	jack@bennie.lslsli.com
2215	15067	1056	0.71	steve@martin.lslsli.com
totals:				
19838	440179	28324	21.84	10

Top users by cpu time

bytes sent	bytes received	connect time	cpu time	address
4370	121056	5211	6.58	george@burns.lsli.com
4222	70651	4973	4.05	bob@hope.lsli.com
2307	42514	4607	3.64	groucho@marx.lsli.com
2358	37554	2879	2.77	gracie@allen.lsli.com
534	59452	1213	1.60	harpo@marx.lsli.com
435	13804	1397	0.83	wc@fields.lsli.com
1471	44927	1091	0.75	phil@silvers.lsli.com
1446	11281	1030	0.72	ace@ventura.lsli.com
2215	15067	1056	0.71	steve@martin.lsli.com
410	9009	4837	0.51	red@skelton.lsli.com
totals:				
19768	425315	28294	22.16	10

## 19.8 Tracking HTTP Usage

T.Rex ships with two HTTP proxies: httpd and webgate. Httpd provides World Wide Web access for internal browsers. Webgate is a reverse HTTP proxy to provides Internet browsers with transparent access web servers that are in your DMZ. Httpd and webgate log http access information in their own log file. Since both proxies use the Common Log Format (CLF) for recording http access activity the same reporting tools can be used to analyze and report web activity. Both logs can be processed with the Smart Web Analysis Tool (SWAT), WebTrends, Hit-List and others. SWAT is available as a separate program from FAS, and can produce 27 different reports from httpd and webgate logs. SWAT uses a 64-bit parallel processing architecture to run more than 60 times faster than other log analysis programs such as WebTrends, and Hit List.

### 19.8.1 Displaying http access in real time

The **display\_webgate** script displays the webgate http access log in real-time in an X-window using black letters on a yellow background. The window has a 500 line buffer and a scroll bar to view the history of alerts. The script is found in the /usr/local/etc/directory. You can invoke this command as follows:

```
# /usr/local/etc/display_webgate
```

You can also use the X-window display command to redirect the syslog messages to another system that is running X-window.

If you are not using X-window the following command will display all security alerts on the system console as they occur.

```
tail -f /home/hermes/http.access.log
```

**Caution:** We do not recommend using these commands other than for problem determination. If you have an active web site that receives more than 100 hits per second the display will not be able to keep up with the http access log.

### 19.8.2 HTTPSUM Report

The httpsum program can be used process the logs generated by the **httpd** proxy and the **webgate** program. Httpsum reads the http access log and produces four reports showing how http is being used. Httpsum is not very fast and we recommend using one of the previously mentioned tools if your logs are large or if you need more than four simple reports.

```
summary report
top servers by command bytes received
top servers by data bytes sent
```

**Webgate:** If you are running webgate on the firewall then you can summarize webgate activity by specifying the fully qualified pathname of the http access log generated by webgate. Webgate can be used to support multiple web servers behind the firewall. Thus the log can contain entries for multiple web servers.

**HTTPD:** If you are running the httpd proxy sever behind the firewall and have specified logging (see Chapter 15 for details) then you can run the httpsum report on the same system. Parameters are available to control the amount of detail reported.

## httpsum [-l proxylog] [-t nnn]

**-l proxylog** This is used to specify the name of the proxy log. If not specified it defaults to the name coded in the AccessLog line found in the /etc/httpd.conf file. This file may not exist because the HTTPD appends the date to the specified name.

**-t nnn** This is used to limit the reporting to the top nnn clients and servers.

### 19.8.3 Sample httpsum output

The following report shows the reporting of the top ten WWW clients and servers each sorted by the number of command bytes sent and the number of data bytes received.

```
$ httpsum -l home/spider/httpd/proxy-log.Nov1595 -t 10
```

HTTPD Summary Report Version 1.1

proxy-log file name: /home/spider/httpd/proxy-log.Nov1595

```
total records in log file      = 1318
number of truncated records   = 198
date of first record          = 09/Nov/1995:17:10:56
date of last record           = 15/Nov/1995:15:58:40
number of complete httpd records = 1107
number of complete gopher records = 3
number of client hosts        = 4
number of server hosts        = 123
number of GET requests        = 1313
total command bytes           = 281340
total data bytes              = 6726332
total bytes                   = 7007672
```

#### Top Clients by Command bytes sent

CMD bytes sent	Databytes received	Num Puts	Num Gets	hostname
142008	3944113	0	633	192.168.220.6
113632	2099104	0	553	192.168.220.14
22700	660902	0	112	192.168.220.15
3000	22213	0	15	lsli2

```
totals:
281340 6726332 0 1313 3
```

#### Top Clients by Data bytes received

CMD bytes sent	Databytes received	Num Puts	Num Gets	hostname
----------------	--------------------	----------	----------	----------



142008	3944113	0	633	192.168.220.6
113632	2099104	0	553	192.168.220.14
22700	660902	0	112	192.168.220.15
3000	22213	0	15	lsli2
totals:				
281340	6726332	0	1313	3

#### Top Servers by Command bytes received

CMD bytes received	Databytes sent	Num Puts	Num Gets	hostname
24000	440240	0	120	home.netscape.com
21600	377476	0	108	www.yahoo.com
20200	391839	0	99	www.women.com
17400	141212	0	69	www.lsli.com
15000	308870	0	75	www.microsoft.com
14140	8540	0	35	system2.lsli.com
10200	116182	0	51	cwsapps.texas.net
8800	259875	0	44	www.cs.uregina.ca
7400	133948	0	37	www.rain.org
7200	234955	0	36	www2.best.com

totals:				
145940	2413137	0	674	10

#### Top Servers by Data bytes sent

CMD bytes received	Databytes sent	Num Puts	Num Gets	hostname
3800	1486311	0	19	www0.iiijnet.or.jp
24000	440240	0	120	home.netscape.com
20200	391839	0	99	www.women.com
21600	377476	0	108	www.yahoo.com
15000	308870	0	75	www.microsoft.com
200	262144	0	1	info.internet.isi.edu
8800	259875	0	44	www.cs.uregina.ca
7200	234955	0	36	www2.best.com
200	212020	0	1	www.mcs.com
4400	164418	0	22	www.desire.co.uk

totals:				
105400	4138148	0	525	10

## 19.9 Tracking Administrator logins

When a systems administrator logs on to the firewall their activity is recorded in two places. If they use ptelnet for a remote login the login is recorded by tnproxy in the syslog. When they connect to the firewall using tnproxy the operating system also records their login. This activity can be reviewed using the "last" command. This information is stored in the UNIX binary file **/usr/adm/wtmp**, and must be created if you don't already have it. The last command will tell you who logged in, from what IP address, and how long they were on. The following is an example of the output from the "last" command filtered by grep to show only the data from March 31.

```
# last | grep "Mar 31"
```

```
ellana    ftp          lsli2        Thu Mar 31 13:32    still logged in.
jim       ftp          lsli2        Thu Mar 31 13:13 - 13:29 (00:16)
root     hft/0        Thu Mar 31 13:03    still logged in.
jim      pts/1        128.0.1.3    Thu Mar 31 12:08 - 12:08 (00:00)
jim      pts/1        128.0.1.3    Thu Mar 31 12:08 - 12:08 (00:00)
jim      pts/1        128.0.1.3    Thu Mar 31 12:07 - 12:07 (00:00)
jim      pts/1        128.0.1.3    Thu Mar 31 12:07 - 12:07 (00:00)
root     pts/0        Thu Mar 31 10:49 - 12:58 (02:08)
carlton  pts/0        128.0.1.3    Thu Mar 31 09:52 - 09:52 (00:00)
root     pts/1        Thu Mar 31 09:52 - 10:45 (00:53)
jim      pts/3        128.0.1.3    Thu Mar 31 09:19 - 09:46 (00:27)
jim      pts/3        128.0.1.3    Thu Mar 31 09:14 - 09:14 (00:00)
carlton  pts/3        128.0.1.3    Thu Mar 31 09:13 - 09:13 (00:00)
ellana   pts/3        128.0.1.3    Thu Mar 31 09:08 - 09:09 (00:01)
jack     pts/3        128.0.1.3    Thu Mar 31 09:06 - 09:07 (00:01)
carlton  pts/3        128.0.1.3    Thu Mar 31 08:35 - 08:36 (00:00)
carlton  pts/3        128.0.1.3    Thu Mar 31 08:35 - 08:35 (00:00)
charles  pts/2        128.0.0.2    Thu Mar 31 08:27 - 09:46 (01:18)
nick     pts/2        128.0.0.2    Thu Mar 31 08:26 - 08:26 (00:00)
root     pts/3        Thu Mar 31 08:23 - 08:24 (00:00)
jack     pts/2        128.0.0.2    Thu Mar 31 08:20 - 08:20 (00:00)
jack     ftp          lsli2        Thu Mar 31 08:15 - 08:17 (00:01)
```

Notice the four attempts that someone made to login as jim between 12:07 and 12:08. All four attempts were made from the system at IP address 128.0.1.3, which is on the unprotected side of this T.Rex firewall. These entries were created when the user jim logged in and failed to respond correctly to the challenge. As a result, each login lasted less than one minute. This could be the result of a user error. That is the user didn't key in the challenge or the response correctly, during the login process. The other possibility is an unauthorized attempt to enter the system.

Notice there were three successful uses of ftp by ellana, jim, and jack. All three users came from the protected system lsli2.

## 19.10 Tracking Failed Login Attempts

You can also see failed login attempts via Telnet using the command "who /etc/security/failedlogin". The following is an example of the output from the "who /etc/security/failedlogin" command.

```
# who /etc/security/failedlogin | grep "Mar 31"
```

```
charles    pts/2        Mar 31 08:20        (128.0.0.2)
charles    pts/2        Mar 31 08:20        (128.0.0.2)
```

UNKNOWN	pts/2	Mar 31 08:21	(128.0.0.2)
jim	pts/3	Mar 31 09:14	(128.0.1.3)
carlton	pts/4	Mar 31 09:21	(128.0.1.3)
carlton	pts/4	Mar 31 09:22	(128.0.1.3)
carlton	pts/4	Mar 31 09:22	(128.0.1.3)
carlton	pts/4	Mar 31 09:37	(128.0.1.3)
carlton	pts/4	Mar 31 09:37	(128.0.1.3)
carlton	pts/4	Mar 31 09:37	(128.0.1.3)
carlton	pts/0	Mar 31 09:52	(128.0.1.3)
jim	pts/1	Mar 31 10:53	(128.0.1.3)
jim	pts/1	Mar 31 12:06	(128.0.1.3)
jim	pts/1	Mar 31 12:08	(128.0.1.3)
UNKNOWN	pts/1	Mar 31 14:26	(128.0.1.3)
UNKNOWN	pts/1	Mar 31 14:29	(128.0.1.3)

This is useful if you want to see if someone is trying to attack your system.

The two login attempts by user UNKNOWN\_ at 14:26 and 14:29 were made by someone trying to login as root from IP address 128.0.1.3, which is on the unprotected network. The person who attempts to login as root will receive a message that says "3004-007 You have entered an invalid name or password".

There was also an unsuccessful login attempt for root made at 08:21. This attempt was made from a system (IP address 128.0.0.2) on the protected network. The root id on the T.Rex system is configured to so that the root user has to login from the console.

Notice the multiple unsuccessful attempts to login to carlton from the unprotected network. In each of these cases the user supplied an incorrect password. This could be a break in attempt.

## 19.11 Reporting use of the su command

The switch user (**su**) command allows an authorized user to become another user without logging off. On the T.Rex gateway the use of the **su** command should be restricted to system administrators. For maximum security we recommend that root logins be restricted to the system console and that su to root be disabled. Chapter 3 Step 11 shows you how to protect the root user id.

The system automatically logs su commands in the **/var/adm/sulog** file. The file contains the date and time of the event, a plus sign "+" to indicate a successful attempt, a minus sign "-" to indicate a failed attempt, the terminal ID, the user ID of the person who entered the command and the user ID they attempted to switch to.

### 19.11.1 Reporting all su attempts to root

The following command will list all entries in the sulog that contain the word root. If root appears as the target user ID then someone is trying to gain root access on your gateway. If the system has been configured properly any attempts to su to root will be rejected ( minus sign " - ").

```
#cat /var/adm/sulog | grep "root"
```

```
SU 07/15 07:50 - pts/2 jim-root
```

```

SU 07/15 16:34 - pts/5 jim-root
SU 07/18 14:02 + pts/0 root-adm
SU 07/24 12:06 + pts/2 root-jim
SU 07/24 12:14 - pts/0 root-root
SU 07/25 17:04 - pts/2 root-root
SU 07/29 08:52 - pts/2 root-root
SU 07/29 08:52 + pts/2 root-jay
SU 08/01 09:49 + pts/0 root-jay
SU 08/01 11:25 - pts/1 jay-root
SU 08/01 11:26 - pts/1 jay-root
SU 08/01 11:26 - pts/1 jay-root
SU 08/23 15:31 - pts/2 jim-root

```

On 7/15 user jim tried to su to root at 7:50 and 16:34, both tries were rejected. The user jim had logged in using terminal pts/2 and pts/5. You will also notice that root can not su to root since the system has been configured to disable any and all su to root.

### 19.10.2 Reporting all failed su attempts

The following command will list all failed su attempts.

```
#cat /var/adm/sulog | grep " - "
```

```

SU 07/15 07:50 - pts/2 jim-root
SU 07/15 16:34 - pts/5 jim-root
SU 07/19 13:36 - pts/2 jay-guest
SU 07/19 13:36 - pts/2 jay-guest
SU 07/24 12:14 - pts/0 root-root
SU 07/25 17:04 - pts/2 root-root
SU 07/29 08:52 - pts/2 root-root
SU 08/01 11:25 - pts/1 jay-root
SU 08/01 11:26 - pts/1 jay-root
SU 08/01 11:26 - pts/1 jay-root
SU 08/23 15:31 - pts/2 jim-root
SU 10/25 08:16 - pts/1 ellana-root
SU 10/29 13:40 - pts/2 ellana-root

```

## 19.12 Logging DNS activity

Most of the daemons that run on the T.Rex firewall automatically write events to the system log. The Domain Name Server **named** will write a limited amount of information to the system log as shown below:

```

Nov 11 10:52:46 gw named[5187]: started again.
Nov 12 11:47:12 gw named[5187]: named: 0826-035 Response from 192.112.36.4 is not a valid form.
Nov 18 17:01:42 gw named[5187]: started again.

```

named will write additional information to the **/etc/tmp/named.run** file if debugging is activated. The amount of information recorded by named is governed by the debugging level specified. named

supports eleven debugging levels, with level 1 recording the least amount of information and level 11 recording the most. The higher debugging levels should be reserved for problem resolution since they record **enormous** amounts of information. Level 1 should be adequate.

### 19.12.1 Activating named debugging during system startup

Named debugging can be activated two ways. You can activate named debugging at system boot time or via a signal command.

#### AIX

To activate named debugging at boot time add "-d 1" to the end of the start /etc/named line in the **/etc/rc.tcpip** file as shown below:

```
start /etc/named "$src_running" " -d 1"
```

To stop named on AIX issue the following command.

```
#kill 'cat /etc/named.pid'
```

To start named on AIX issue the following command.

```
#startsrc -s named
```

#### HP-UX

For HP-uX add "-d1" to named\_ARGS in /etc/rc.config.d/namesvrs.

To stop named on HP-UX issue the following command.

```
#!/sbin/init.d/named stop
```

To start named on HP-UX issue the following command.

```
#!/sbin/init.d/named start
```

#### Solaris

For Solaris add "-d1" to in.named start-up in /etc/init.d/inetsvc.

To stop named on Solaris issue the following command.

```
# kill 'cat /etc/named.pid'
```

To start named on Solaris issue the following command.

```
# /usr/sbin/in.named
```

## 19.12.2 Controlling named debugging with Signals

Signals can be used to start and stop named debugging as well as change the debugging level. Get the name from /etc/named.pid. To increase the debugging level by one issue the kill -ASSAYER 'cat /etc/named.pid' command first time the command is issued will start debugging at level 1. Each successive ASSA R signal will increase by one. To decrease the debugging level issue the kill -USR2 'cat /etc/named.pid' command.

## 19.12.3 Sample named Startup Debugging Information

The following output was written to the /etc/tmp/named.run file while a caching-only name server was initializing

```
Debug turned ON, Level 1
Version = named 4.8.3 Fri Sep 10 15:53:27 CDT 1993:/
      build@chance.austin.ibm.com:/CHANCE/tcpip/3.2.5/src/nls/sockcmd/named
      bootfile = /etc/named.boot
dqp->dq_addr 127.0.0.1 d_dfd 6
loopback address: x7f000001
dqp->dq_addr 198.65.130.22 d_dfd 7
dqp->dq_addr 192.168.0.1 d_dfd 8
dqp->dq_addr 192.168.1.1 d_dfd 9
dqp->dq_addr 0.0.0.0 d_dfd 10

ns_init(/etc/named.boot)
zone origin 0.0.127.IN-ADDR-ARPA, source = /etc/named.local
reloading zone
db_load(/etc/named.local, 0.0.127.IN-ADDR-ARPA, 1, 0)
zone[1] type 1: '0.0.127.IN-ADDR-ARPA' z_time 0, z_refresh 0
zone origin ., source = /etc/named.ca
reloading zone
db_load(/etc/named.ca, , 0, 0)
zone[0] type 3: '.' z_time 0, z_refresh 0
exit ns_init(), need maintenance immediately
Network and sort list:
net xc6418215 mask xffffff my_addr xc6418216 198.65.130.22
net xc0a80000 mask xffffff0 my_addr xc0a80001 192.168.0.1
net xc0a80100 mask xffffff0 my_addr xc0a80101 192.168.1.1
net xc6418200 mask xffffff0 my_addr xc6418216 198.65.130.22
database initialized
Ready to answer queries.
prime_cache: priming = 0
sysquery: send -> 128.9.0.107 10 (53), nsid=1 id=0 0ms
sendto error errno= 69

ns_maint(); now Thu Feb  9 19:16:29 1995
sched_maint: Next interrupt in 0 sec
exit ns_maint()
resend(addr=1 n=0) -> 192.33.4.12 10 (53) nsid=1 id=0 0ms
```

```
datagram from 192.168.0.1 port 1026, fd 8, len 26
req: nlookup(loopback) id 1 type=1
req: missed 'loopback' as '' (cname=0)
forw: forw -> 128.9.0.107 10 (53) nsid=2 id=1 0ms retry 4 sec
prime_cache: priming = 1
resend(addr=2 n=0) -> 128.8.10.90 10 (53) nsid=1 id=0 0ms
```

```
datagram from 192.168.0.1 port 1025, fd 8, len 20
req: nlookup(gw) id 1 type=1
req: missed 'gw' as '' (cname=0)
forw: forw -> 128.9.0.107 10 (53) nsid=3 id=1 0ms retry 4 sec
prime_cache: priming = 1
resend(addr=1 n=0) -> 192.33.4.12 10 (53) nsid=2 id=1 0ms
```

```
datagram from 192.168.0.1 port 1026, fd 8, len 26
req: nlookup(loopback) id 1 type=1
req: missed 'loopback' as '' (cname=0)
prime_cache: priming = 1
resend(addr=1 n=0) -> 192.33.4.12 10 (53) nsid=3 id=1 0ms
resend(addr=3 n=0) -> 128.102.16.10 10 (53) nsid=1 id=0 0ms
resend(addr=2 n=0) -> 128.8.10.90 10 (53) nsid=2 id=1 0ms
```

```
datagram from 192.168.0.1 port 1027, fd 8, len 27
req: nlookup(localhost) id 1 type=1
req: missed 'localhost' as '' (cname=0)
forw: forw -> 128.9.0.107 10 (53) nsid=4 id=1 0ms retry 4 sec
prime_cache: priming = 1
```

```
datagram from 192.168.0.1 port 1028, fd 8, len 27
req: nlookup(localhost) id 1 type=1
req: missed 'localhost' as '' (cname=0)
forw: forw -> 128.9.0.107 10 (53) nsid=5 id=1 0ms retry 4 sec
prime_cache: priming = 1
resend(addr=4 n=0) -> 192.52.195.10 10 (53) nsid=1 id=0 0ms
resend(addr=2 n=0) -> 128.8.10.90 10 (53) nsid=3 id=1 0ms
resend(addr=3 n=0) -> 128.102.16.10 10 (53) nsid=2 id=1 0ms
resend(addr=1 n=0) -> 192.33.4.12 10 (53) nsid=5 id=1 0ms
resend(addr=1 n=0) -> 192.33.4.12 10 (53) nsid=4 id=1 0ms
```

...

## 19.12.4 Sample named Lookup Debugging Information

The following named level 1 debugging information was recorded during a successful nslookup for the name sun.com.

```
datagram from 192.168.220.1 port 1553, fd 8, len 25
req: nlookup(sun.com) id 1 type=1
req: found 'sun.com' as 'com' (cname=0)
forw: forw -> 198.41.0.4 10 (53) nsid=315 id=1 238ms retry 4 sec
```

```
datagram from 198.41.0.4 port 53, fd 7, len 294
resp: nlookup(sun.com) type=1
resp: found 'sun.com' as 'sun.com' (cname=0)
resp: forw -> 128.63.16.6 10 (53) nsid=316 id=1 0ms

datagram from 128.63.16.6 port 53, fd 7, len 41
send_msg -> 192.168.220.1 (UDP 8 1553) id=1
```



# Chapter 20. Accounting and Security

## 20.1 Accounting Overview (AIX)

The AIX accounting system allows collection and reporting on resources used by individual users and groups of users. Since the accounting files contain detailed information on how individuals are using the system these files are key to investigating the security of the system. This information can be used to bill users for resource utilization and to monitor how the system is used. The accounting system can also be used for capacity planning, and setting resource limits and quotas.

The systems administrator can setup the system to automatically collect and analyze the accounting information. Automated procedures are available to report on:

- (1) The amount of time each user spends logged in to the system.
- (2) The amount of cpu, memory and I/O resources used.
- (3) The amount of disk space occupied by each user's files.
- (4) The number of times a specific command is given.

### Accounting Commands

AIX provides a robust set of commands for controlling and reporting accounting information, such as:

<b>acctcms</b>	The acctcms provides a summary of resource usage by command. Acctcms provides information on the number times each command was used and how much cpu and memory it used. The acctcms command also provides information on total systems usage.
<b>acctcom</b>	The acctcom command provides detailed information by process.
<b>acctmerg</b>	The acctmerg command creates daily reports from raw accounting data. The acctmerg program can convert records between ASCII and binary formats and can be used to merge records from multiple sources into single records for each user.
<b>monacct</b>	The monacct command produces a monthly accounting report.
<b>prdaily</b>	Prints the previous day accounting summary.
<b>turnacct</b>	Switches accounting on or off.
<b>sa</b>	Summary of accounting
	<b>sa -k</b> Summarize activity by user
	<b>sa -a</b> Summarize activity by Command

**ac** Prints the total connect time for all users or for all users specified by the user parameter.

**ac -d** Connect time by day

**ac -p** print the connect time by person by day

**ac jim** print the total connect time for the user jim

**sag** The sag command displays a graph of system activity

## Sample Daily Summary

The prdaily command creates a summary report of the previous day's accounting data. This command is called by the runacct command to automatically produce daily summary reports in the /var/adm/acct/sum directory. For each day of the month there will be a file with the name rptmmd.

Sat Dec 10 23:10:10 CST 1994 DAILY COMMAND SUMMARY Page 1

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
TOTALS	1248	976.48	5.63	2271.93	173.32	0.00	0.25	1.72e+08	118.00
sendmail	405	723.01	2.74	417.21	263.86	0.00	0.66	3.146e+07	0.00
grep	54	92.82	0.69	1.45	133.90	0.01	47.78	4.446e+07	0.00
aixterm	11	50.34	0.17	39.28	297.38	0.02	0.43	679840.00	0.00
bsh	125	21.73	0.27	3.26	79.08	0.00	8.43	781725.00	0.00
telnet	6	19.47	0.43	120.57	44.85	0.07	0.36	1.175e+06	0.00
ksh	53	10.41	0.10	163.14	102.72	0.00	0.06	802178.00	0.00
telnetd	7	9.38	0.34	122.93	27.34	0.05	0.28	993282.00	0.00
dctrl	2	5.45	0.03	0.27	205.00	0.01	9.69	401024.00	0.00
more	18	4.90	0.06	1.38	84.80	0.00	4.20	324784.00	0.00
slipdl	3	3.17	0.27	1366.36	11.74	0.09	0.02	13999.00	0.00
sadc	14	2.75	0.10	0.22	26.81	0.00	46.74	1.645e+07	0.00
acctprc2	1	2.66	0.02	0.03	142.00	0.02	62.61	64944.00	0.00
errpt	2	2.59	0.00	0.03	284.51	0.00	27.56	494720.00	0.00
awk	25	2.29	0.02	0.04	127.68	0.00	41.07	105525.00	0.00
cat	51	1.81	0.04	0.80	49.91	0.00	4.52	7.054e+07	0.00
acctprcl	1	1.74	0.01	0.02	167.00	0.01	48.19	119168.00	0.00
acctcms	4	1.57	0.00	0.01	177.24	0.00	69.39	131792.00	0.00
who	8	1.31	0.02	0.04	85.29	0.00	41.84	198980.00	0.00
ls	12	1.19	0.01	0.04	84.33	0.00	36.24	139152.00	0.00
ftpd	5	1.12	0.02	20.02	46.66	0.00	0.12	649968.00	0.00
doswrite	8	1.03	0.01	1.33	84.47	0.00	0.92	333592.00	118.00
cp	44	1.02	0.02	0.04	47.80	0.00	54.30	24372.00	0.00
rm	41	1.01	0.02	0.05	58.00	0.00	32.68	66048.00	0.00
ps	4	0.99	0.01	0.02	90.38	0.00	43.75	59716.00	0.00
df	25	0.84	0.01	0.03	66.94	0.00	39.67	16100.00	0.00
errclear	2	0.75	0.00	0.02	168.41	0.00	24.64	183168.00	0.00
chmod	38	0.72	0.01	0.02	51.96	0.00	85.48	0.00	0.00
sockd	9	0.72	0.02	9.30	35.71	0.00	0.22	461696.00	0.00
du	24	0.70	0.01	0.03	58.57	0.00	41.07	216.00	0.00
nslookup	11	0.70	0.00	0.21	78.82	0.00	4.14	204104.00	0.00
tsm	1	0.68	0.00	0.26	175.00	0.00	1.52	30768.00	0.00
sed	29	0.63	0.01	0.07	58.54	0.00	14.34	33186.00	0.00
ping	6	0.57	0.00	0.87	75.86	0.00	0.87	73872.00	0.00
dspmsg	18	0.54	0.00	0.01	64.81	0.00	78.05	74234.00	0.00
pr	5	0.53	0.00	0.01	88.78	0.00	60.53	27076.00	0.00
chgrp	13	0.45	0.00	0.00	66.92	0.00	89.66	3757.00	0.00
whois	7	0.43	0.00	0.66	58.93	0.00	1.11	84768.00	0.00
date	20	0.42	0.00	0.01	53.57	0.00	71.43	142.00	0.00
ln	25	0.41	0.00	0.02	46.32	0.00	43.04	0.00	0.00
tty	10	0.31	0.00	0.01	66.78	0.00	43.90	1530.00	0.00
lscons	14	0.31	0.00	0.03	49.08	0.00	18.90	154.00	0.00
sort	5	0.30	0.00	0.03	77.47	0.00	12.00	4136.00	0.00
dhf	1	0.26	0.00	0.04	101.00	0.00	5.88	0.00	0.00
chown	13	0.24	0.00	0.00	38.67	0.00	75.00	7644.00	0.00
cut	10	0.23	0.00	0.01	54.19	0.00	29.09	324.00	0.00
xset	4	0.19	0.00	0.01	48.73	0.00	27.27	6732.00	0.00
finger	3	0.18	0.00	0.48	85.00	0.00	0.43	32784.00	0.00
acctconl	1	0.16	0.00	0.00	126.00	0.00	26.32	15192.00	0.00
gwhsh	1	0.16	0.00	0.00	61.00	0.00	62.50	34632.00	0.00
acctmerg	6	0.16	0.00	0.00	67.44	0.00	33.33	20021.00	0.00

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL COMMAND SUMMARY		MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
			TOTAL CPU-MIN	TOTAL REAL-MIN					
TOTALS	10383	16520.01	40.62	101446.67	406.71	0.00	0.04	7.957e+08	474.00
mosaic	6	8852.25	3.67	59.11	2412.88	0.61	6.21	2.07e+07	0.00
info_gr	2	3219.06	1.16	149.57	2775.31	0.58	0.78	7.959e+06	0.00
sendmail	2403	1877.33	7.87	2022.28	238.69	0.00	0.39	1.048e+08	0.00
aixterm	72	714.26	2.34	32839.99	304.68	0.03	0.00	4.949e+06	0.00
grep	288	315.83	2.39	4.62	132.06	0.00	51.75	1.444e+08	0.00
bsh	1261	225.80	2.70	92.64	83.69	0.00	2.91	7.346e+06	0.00
telnet	63	183.73	3.06	5387.59	60.04	0.05	0.06	9.533e+06	0.00
vi	35	180.55	0.55	116.85	325.35	0.02	0.47	2.714e+06	0.00
find	17	107.27	3.69	5.36	29.11	0.22	68.70	105080.00	0.00
msmit	2	105.80	0.13	2.48	832.51	0.06	5.12	574080.00	0.00
ksh	348	93.56	0.82	38464.74	114.05	0.00	0.00	6.133e+06	0.00
telnetd	71	67.16	2.63	5821.66	25.51	0.04	0.05	8.448e+06	0.00
man	4	44.68	0.17	4.43	262.73	0.04	3.84	1.536e+07	0.00
more	151	44.64	0.56	158.41	79.83	0.00	0.35	2.511e+06	0.00
slipdl	13	43.67	2.20	12379.13	19.82	0.17	0.02	58346.00	0.00
dctrl	20	39.33	0.21	1.09	185.07	0.01	19.55	4.01e+06	0.00
sh	46	27.28	0.21	204.24	128.84	0.00	0.10	1.036e+06	0.00
errpt	20	25.40	0.09	0.18	286.05	0.00	50.29	4.836e+06	0.00
acctprc2	10	24.86	0.18	0.28	141.64	0.02	62.64	602552.00	0.00
sadc	112	24.33	0.85	1.84	28.66	0.00	46.24	1.334e+08	0.00
awk	268	23.00	0.21	0.58	108.92	0.00	36.65	1.13e+06	0.00
gwhsh	41	21.00	0.17	18.87	123.68	0.00	0.90	1.774e+06	0.00
ftpd	30	20.66	0.68	357.92	30.50	0.02	0.19	2.612e+07	0.00
acctprcl	10	16.58	0.10	0.20	171.11	0.01	48.19	1.117e+06	0.00
ls	123	16.50	0.16	0.42	101.68	0.00	38.43	1.355e+06	0.00
acctcms	41	14.80	0.09	0.12	168.16	0.00	76.13	1.201e+06	0.00
tail	4	14.22	0.94	3233.58	15.13	0.23	0.03	80216.00	0.00
cat	361	13.15	0.27	3.86	49.61	0.00	6.87	2.684e+08	0.00
who	88	12.30	0.14	0.36	87.64	0.00	39.52	1.611e+06	0.00
cp	454	10.53	0.22	0.42	48.24	0.00	51.66	644480.00	0.00
rm	421	9.61	0.19	0.45	50.28	0.00	42.11	98834.00	0.00
sed	303	8.36	0.13	0.55	63.94	0.00	23.60	360176.00	0.00
errclear	19	8.17	0.05	0.25	156.10	0.00	21.09	1.529e+06	0.00
chmod	383	7.55	0.14	0.18	52.25	0.00	80.09	16512.00	0.00
df	249	7.29	0.12	0.31	61.50	0.00	38.46	160356.00	0.00
ps	33	6.90	0.09	0.19	80.30	0.00	45.90	366612.00	0.00
pr	53	6.43	0.06	0.11	104.62	0.00	56.73	294628.00	0.00
dspmsg	192	6.21	0.09	0.13	68.94	0.00	70.33	838212.00	0.00
du	239	6.17	0.13	0.31	49.29	0.00	40.56	2155.00	0.00
date	177	4.92	0.07	0.10	66.25	0.00	71.43	17993.00	0.00
ln	249	4.80	0.09	0.20	52.55	0.00	45.17	0.00	0.00
ping	75	4.35	0.07	6.08	65.52	0.00	1.09	817972.00	0.00
chgrp	132	4.19	0.06	0.08	66.53	0.00	77.81	37966.00	0.00
chown	132	3.43	0.06	0.09	54.23	0.00	71.68	77434.00	0.00
tsm	3	2.82	0.01	0.67	200.44	0.00	2.11	137504.00	0.00
whois	48	2.81	0.04	6.72	63.42	0.00	0.66	545375.00	0.00
tty	100	2.80	0.05	0.10	62.09	0.00	45.77	15300.00	0.00
sockd	38	2.77	0.06	15.18	48.40	0.00	0.38	820696.00	0.00
nslookup	44	2.50	0.03	0.70	71.67	0.00	4.98	771480.00	0.00
doswrite	12	2.12	0.02	2.28	88.35	0.00	1.05	914768.00	243.00

## Exceptional usage by login ID

The `prdaily` command can also be used to report exceptional usage by login id. To produce an exceptional usage report by login ID supply the arguments `-l mmdd`.

```
#/usr/sbin/acct/prdaily -l 1210
```

Logins with exceptional Prime/Non-prime Time Usage  
CPU > 20 or KCORE > 500 or CONNECT > 120

Login UID	Name	CPU (mins)		KCORE-mins		CONNECT-mins		disk Blocks	# of Procs	# of Sess	# Disk Samples	fee
		Prime	Nprime	Prime	Nprime	Prime	Nprime					
0	root	1	0	28466	12787	23587	10234	0	0	2	0	0
4	adm	0	0	1226	6379	207	17600	0	0	0	0	0
200	jim	0	0	1265	0	468	0	0	0	0	0	0
203	jay	0	0	5761	0	1986	0	0	0	173	0	0

## Connect time by Login ID by Day

The connect time by Login ID by Day can be produced by the `ac -p` command.

```
#ac -p
```

```
wtmp begins Thu Oct 27 06:29
root      14.17
ellana    0.24
jay       0.02
jim       0.00
ted       0.74
total     15.17
```

## Reporting commands used by login ID

It is possible to capture every command issued by a Login ID and report: command name, user name, tty name, start time, end time, real seconds, cpu seconds, mean memory size in KB.

```
# acctcom | grep "ted"
```

who	ted	pts	14:18:19	14:18:19	0.08	0.03	30.00
awk	ted	pts	14:18:19	14:18:19	0.12	0.05	168.00
sh	ted	pts	14:18:19	14:18:19	0.27	0.08	75.00
hostname	ted	pts	14:18:29	14:18:29	0.05	0.02	0.00
true	ted	pts	14:18:30	14:18:30	0.02	0.01	118.00
true	ted	pts	14:18:49	14:18:49	0.02	0.01	120.00
true	ted	pts	14:18:51	14:18:51	0.02	0.01	104.00
true	ted	pts	14:18:54	14:18:54	0.02	0.02	64.00
true	ted	pts	14:19:01	14:19:01	0.02	0.01	114.00
true	ted	pts	14:19:12	14:19:12	0.02	0.02	0.00
true	ted	pts	14:19:38	14:19:38	0.02	0.02	0.00
true	ted	pts	14:20:14	14:20:14	0.02	0.02	0.00
true	ted	pts	14:20:23	14:20:23	0.02	0.01	102.00

ntelnet	ted	pts	14:20:37	14:20:37	0.22	0.08	58.00
true	ted	pts	14:20:37	14:20:37	0.02	0.02	0.00
true	ted	pts	14:21:06	14:21:06	0.02	0.01	112.00
#sh	ted	pts	14:16:15	14:21:08	293.12	0.58	195.00
who	ted	pts	14:22:15	14:22:15	0.11	0.05	0.00
awk	ted	pts	14:22:15	14:22:15	0.12	0.05	174.00
sh	ted	pts	14:22:15	14:22:15	0.25	0.08	48.00
hostname.	ted	pts	14:22:26	14:22:26	0.03	0.03	91.00
true	ted	pts	14:22:26	14:22:26	0.02	0.01	106.00
who	ted	pts	14:25:49	14:25:49	0.08	0.08	69.00
awk	ted	pts	14:25:49	14:25:49	0.20	0.06	127.00
sh	ted	pts	14:25:49	14:25:49	0.33	0.08	71.00
hostname.	ted	pts	14:25:53	14:25:53	0.03	0.02	18.00
true	ted	pts	14:25:53	14:25:53	0.02	0.02	0.00
#sh	ted	pts	14:25:38	14:29:53	255.88	0.48	169.00
ntelnet	ted	pts	14:22:36	14:58:12	2136.00	0.78	57.00
true	ted	pts	14:58:12	14:58:12	0.02	0.01	116.00
#sh	ted	pts	14:21:16	14:58:18	2222.00	0.56	130.00

There are many other reports available as part of the AIX system.

notes:

# Chapter 21. T.Rex Admin Server

## 21.1 Tadminsvr Overview

The T.Rex Admin Server (tadminsvr) supports the remote administration client (Hoplite). Padminsvr is automatically installed during the T.Rex installation process. Padminsvr allows administrative users to modify T.Rex configuration files using the Hoplite client. Hoplite users must undergo strong user authentication before they can administer the firewall. Security is maintained by encrypting all data traffic between Hoplite and Padminsvr.

## 21.2 Padminsvr Configuration File

Padminsvr uses the /etc/firewall/padmin.conf file to control connections with the Hoplite client.

### Command Syntax

Each command is contained on a line that can be up to 1023 characters in length. Spaces, tabs and equal signs separate fields. Comment lines begin with a "#" character. Blank lines are ignored. The format of each command follows:

#### **timeout = *nnn***

The timeout value specifies the number of seconds the padminsvr will stay idle before it closes the Hoplite connection.

#### **permit *ip\_address* [*ip\_mask*]**

The permit rule controls the IP addresses from which the Hoplite server can connect.

**ip\_address**     The IP\_address field is required and must immediately follow the **permit** keyword. The IP\_address gives the fully qualified IP address of the Hoplite client in standard dot decimal format.

**ip\_mask**        The IP mask is a optional field and must immediately follow the IP\_address. If the masking bit is one then an exact match is required. If the masking bit is zero then bit in the IP address is ignored. Specification of a source mask of 255.255.255.255 requires an exact match with the src\_addr for the rule to apply. Specification of an IP mask of 0.0.0.0 permits a match no matter what source IP address is applied.

#### **rootuser *username* [*ip\_address* [*ip\_mask*]]**

The root user command specifies which Hoplite users have root user authority. Root authority is required for some functions such as backup and restore.

**username**        The admintrator's user name.

**ip\_address**     The IP\_address field is optional and must immediately follow the **username**. The

IP\_address gives the fully qualified IP address of the Hoplite client in standard dot decimal format. If the IP\_address is not coded then no restrictions are placed on the administrative users location.

**ip\_mask** The IP mask is a optional field and must immediately follow the IP\_address. If the masking bit is one then an exact match is required. If the masking bit is zero then bit in the IP address is ignored. Specification of a source mask of 255.255.255.255 requires an exact match with the src\_addr for the rule to apply. Specification of an IP mask of 0.0.0.0 permits a match no matter what source IP address is applied.

## 21.3 Sample padmin.conf File

The following sample padmin.conf file is shipped with T.Rex.

```
# Set timeout for padmin server (in seconds)
timeout 300
#
# Permit remote login clients
# Syntax:
#   permit ip_address [ip_mask]
# Examples:
#   permit 10.0.0.1
#   permit 192.168.0.0 255.255.255.0
#
# Specify usernames who can act as root users
# Syntax:
#   rootuser username [ip_address [ip_mask]]
# Examples:
#   rootuser admin0
#   rootuser admin1 192.168.0.11
```

## 21.4 Padminsvr Backup Lists

Padminsvr allows the hoplite utility to selectively backup configuration files used by: T.Rex, the system, the Apache http server plus a customized list. There are four files in the /etc/T.Rex directory that are used by hoplite . Each file contains a list of files to be backed up on the hoplite computer.

### 21.4.1 T.Rex Backup List

The sample backup list for T.Rex files is listed below.

```
./etc/firewall/aliases
./etc/firewall/apache.bulist
./etc/firewall/custom.bulist
./etc/firewall/ftproxy.conf
./etc/firewall/fwpulse.conf
./etc/firewall/genproxy.conf
./etc/firewall/gwuser.conf
./etc/firewall/padmin.conf
```



```
./etc/firewall/T.Rex.bulist
./etc/firewall/T.Rexkey*
./etc/firewall/T.Rexmon.conf
./etc/firewall/raproxy.conf
./etc/firewall/resolv.inside.conf
./etc/firewall/rpcproxy.conf
./etc/firewall/securenets
./etc/firewall/secureports
./etc/firewall/smwrap.conf
./etc/firewall/sockd.conf
./etc/firewall/sockd.route
./etc/firewall/system.bulist
./etc/firewall/tnproxy.conf
./etc/firewall/webgate.conf
./etc/firewall/gwuser.dir
./etc/firewall/gwuser.pag
```

## 21.4.2 System Backup List

The system backup list contains files modified during the T.Rex installation.

```
./etc/aliases
#Solaris file
./etc/mail/aliases
#Solaris file
./etc/defaultrouter
./etc/hosts
#Solaris files
./etc/hostname.*
# HPUX file
./etc/rc.config.d/netconf
#Solaris file
./etc/net/*/hosts
./etc/inetd.conf
./etc/inittab
./etc/motd
./etc/named.*
#Solaris file
./etc/nodename
./etc/netsvc.conf
#Solaris file
./etc/nsswitch.conf
./etc/profile
./etc/rc.net
#Solaris file
./etc/init.d/inetsvc
./etc/rc.tcpip
./etc/resolv.conf
./etc/sendmail.cf
#Solaris file
./etc/mail/sendmail.cf
./etc/services
```

```
./etc/syslog.conf  
#
```

### **21.4.3 Apache Backup List**

Only two files used by the http proxy need to be backed up. These are.

```
./etc/apache/httpd.conf  
./etc/apache/mime.types
```

### **21.4.4 Custom Backup List**

You can make your own custom backup list using the same format as shown above.

# Chapter 22. Remote logging with Plog

## 22.1 Plog Overview

The plog program is designed to transmit one or more files to another system in near real time. Plog can be used to transmit http access logs from the firewall to another system. The remote system can be running real-time log analysis programs to display web browsing activity. The plogd program must be installed on the remote system to receive the data from plog.

Plog buffers log data to improve transmission performance. Webgate can support hundreds to thousands of web Operations per second. The average length of a log record is 150 bytes. A web server supporting 500 Web Operations per second will generate 75,000 bytes of log data per second. Plog monitors the writing of data to the specified log and copies the data to a buffer. At designated time intervals plog writes the contents of the buffer to the remote system. Buffering the writes significantly improves the performance of remote logging.

Plog maintains a history of what data has been sent to the remote server. If communications with the remote server is interrupted for any reason plog can resume sending data starting with the point of the interruption. There is no loss of data or unnecessary re-transmission.

Plog communicates with the plogd program on the remote server. Both plog and plogd must be configured to allow the two systems to communicate with each other.

## 22.2 Plog Configuration

The plog program is controlled by commands in the /etc/firewall/plog.conf file. There can be one or more remotelog commands in the plog.conf file. Each command is used to transmit a single file to a remote server. Multiple commands can be used to transmit multiple files to remote servers.

### Command Syntax

Each command is contained on a line that can be up to 1023 characters in length. Spaces, tabs and equal signs separate fields. Comment lines begin with a "#" character. Blank lines are ignored. The format of each command follows:

```
remotelog      full_path_name server_address [port_number [buffer_size [time_interval]]]
```

### Required parameters

**full\_path\_name** The full path name of the file to be transmitted.

**server\_address** The IP address of the remote server.

### Optional Parameters

**port\_number** The port number that plogd is listening to. The default port is 743.

buffer\_size      The buffer size in bytes. The default value is 16384 bytes.

time\_interval    The time interval between transmissions. The default value is 5 seconds.

## 22.3 Example plog.conf file

```
# file: /etc/firewall/plog.conf
# function: The plog.conf file is used to control the execution of the
#           plog program.
#
# (C) R. J. Livermore 1999-2000
# (C) Freemont Avenue Software, Inc. 1999-2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The format of the commands are as follows:
#
# remotelog <full_path_name> <server_address>
#           [<port number> [<buffer size> [<time interval>]]]
#
# The following example sends the running contents of /var/adm/syslog to
# IP address 192.168.0.8 at port 743 using a 16384 byte buffer. New contents
# are sent in bursts every 5 seconds.
#
remotelog /var/adm/syslog 192.168.0.8 743 16384 5
```

## 22.4 Plogd Overview

The plogd daemon is designed to receive log information from one or more systems. Plogd can be used to receive http access logs from multiple firewalls running plog. The system can run log analysis programs to display web browsing activity.

## 22.5 Plogd Configuration

The plogd program can be run on a MS/ NT system. The plogd program is controlled by commands in the plogd.cfg file. There can be one or more log commands in the plogd.cfg file. Each log command is used to permit receipt of data from a remote firewall system. Multiple commands can be used to receive multiple files from one or more firewalls.

### Command Syntax

Each command is contained on a line that can be up to 1023 characters in length. Spaces, tabs and equal signs separate fields. Comment lines begin with a "#" character. Blank lines are ignored. The format of each command follows:

**log**    **log\_file\_name** **from\_address** **to\_port** [**rotate\_date**]

#### Required parameters

<b>Log_file_name</b>	The name of the file to be logged.
<b>From_address</b>	The IP address of the sending system.
<b>to_port</b>	The port number that plogd is listening to. The default port is 743.
<b>rotate_date</b>	The rotate_date parameter tells plogd to create a new file when the date changes. This parameter is optional.

## 22.6 Example plogd.conf file

```
# file: plogd.cfg
# function: The plogd.cfg file is used to control the execution of the plogd program.
#
#
# (C) R. J. Livermore 1999-2000
# (C) Freemont Avenue Software, Inc.. 1999-2000
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The format of the commands are as follows:
#
# remotelog <full_path_name> <server_address>
#           [<port number> [<buffer size> [<time interval>]]]
#
# The following example sends the running contents of /var/adm/syslog to
# IP address 192.168.0.8 at port 743 using a 16384 byte buffer. New contents
# are sent in bursts every 5 seconds.
#
log gw1.http.access.log 192.168.0.5 743 rotate_date
```

# Chapter 23. Protected Telnet

## 23.1 Ptelnet Overview

The ptelnet client provides encrypted transmissions between telnet clients and the T.Rex firewall. This ensures that all the data transmitted between the client and the firewall remains private. This includes user IDs, passwords and all commands and data. In the United States and Canada the ptelnet client uses the DES encryption algorithm. Today the ptelnet client is only available in the USA and Canada. The ptelnet client can communicate with standard telnet daemons and the tnproxy supplied with T.Rex.

Ptelnet supports normal users and firewall administrators.

**Normal users** can request encrypted transmission between unsecured hosts and the firewall. After a user has connected to the telnet proxy and passed strong user authentication they can connect to another system using the **safetn** command. The safetn command activates data encryption between the ptelnet client and the firewall. The safetn does not support encryption between secured hosts and the firewall. This restriction is in place because the encryption takes place only between the ptelnet client and the firewall. The data transmission between the firewall and the server host is not encrypted. In order to prevent a false sense of security we do not activate encryption for users using ptelnet on a secured client.

**T.Rex administrators** can login to T.Rex from a remote system using ptelnet. Administrator logins to T.Rex can be made from either a protected or unprotected host. The administrator uses ptelnet to connect to the firewall. After providing a user ID and passing the authentication test the administrator requests a connection to the firewall. If the administrator is authorized to login to the firewall, encryption will automatically be activated. T.Rex administrators will always be subjected to strong user authentication and will always have an encrypted data stream.

## 23.2 T.Rex User Administration

To use ptelnet with T.Rex the systems administrator must create a gwuser record on the firewall and create a shared private key to be stored on the ptelnet client system. The steps are defined as follows:

### 23.2.1 Create gwuser record

Logon to the firewall and execute /usr/local/etc/gwuser program for each secure telnet user. The following two examples, demonstrate definition of a firewall administrator and a regular user that requires encrypted telnet.

### 23.2.2 Creating a Remote Firewall Administrator

Run the gwuser program to define a T.Rex user that can logon on to the firewall as a systems administrator. This user must have the following telnet options specified:

PRO_TNSA	strong user authentication for protected telnet.
ADM_TN	administrator login.
token type	Either CRYPTOCARD or SecureNet Key support to support strong user authentication.

Chapter 5 contains detailed information on the gwuser command. The following example shows the creating of a gwuser record for a firewall administrator.

```
# gwuser -a james -t PRO_TNSA UNPRO_TN ADM_TN -s CRYPTO \
-k 077 044 001 170 047 177 072 101
```

These are the minimum options required to specify a T.Rex systems administrator. With this example, the systems administrator will be assigned the default restricted shell. This shell controls the commands available to the administrator.

### 23.2.4 Creating a user with Encrypted telnet support

Run the gwuser program to define a T.Rex user that can encrypt telnet sessions between an unsecured host and the firewall. This user must have the following telnet options specified:

UNPRO_TN	user is allowed telnet access from an unprotected host.
token type	Either CRYPTOCARD or SecureNet Key support to support strong user authentication.

Chapter 5 contains detailed information on the gwuser command. The following example shows the creating of a gwuser record for a firewall administrator.

```
# gwuser -a james -t UNPRO_TN -s CRYPTO -k 077 044 001 170 047 177 072 101
```

These are the minimum options required to specify a T.Rex systems administrator.

## 23.3 Generate shared private key for ptelnet

Ptelnet uses shared private encryption keys that can be generated using the **tnkey** utility. The tnkey program is run on the firewall and generates an encrypted file that contains the DES key used by ptelnet. The syntax of the tnkey is as follows:

**tnkey [-q] [ user [user...] ]**

**-q** Suppresses printing of username as comment after each key. By default, tnkey prints the username as a comment after each key.

**user** One or more user IDs can be specified. User IDs are separated by blanks.

The normal output of the command is to the screen. The output can also be directed to a file using the '>' character followed by the file name. Once created the file can be moved to the client system



via ftp or by diskette or tape.

Example 1:

Create a file with a DES key for each user found in the data base. The output is directed to the T.Rex.key file in the tmp directory. Notice the gwuser program is used to create a list of users as arguments to tnkey.

```
# /usr/local/etc/tnkey `/usr/local/etc/gwuser -l` > /tmp/T.Rex.key
```

Example 2:

To create a file with a DES key for each user belonging to the group "staff" issue the following command.

```
# /usr/local/etc/tnkey `/usr/local/etc/gwuser -l -g staff` > /tmp/T.Rex.key
```

## **23.4 Enable administrator logins**

### **23.4.1 AIX Administrators**

On AIX, execute "chuser rlogin=true ttys=ALL root"

Look in /etc/inetd.conf and make real sure that no ports, other than port 23 are open for telnet access. Edit the file and refresh inetd if necessary.

### **23.4.2 HP-UX Administrators**

On HP-UX, edit /etc/security and add the following entries (one per simultaneous login), but don't remove the existing "console" entry, or root logins will be disallowed at the console:

```
pts/0  
pts/1  
pts/2  
pts/3  
pts/4
```

### **23.4.3 Solaris Administrators**

On Solaris, edit /etc/default/login and comment out the "CONSOLE=/dev/console" entry

## **23.5 ptnet Installation**

The following sections describe the installation process for the ptnet client on MS Windows/95 MS NT, AIX, HP-UX and Solaris.

## 23.6 Ptelnet for Windows 95 or Windows NT

Perform the following steps on the Windows 95 or Windows NT client system:

1. Copy ptelnet.exe to \T.Rex\ptelnet.exe.
2. Copy /tmp/T.Rex.key from the firewall to c:\T.Rex\T.Rex.key
3. If the host you intend to telnet to with the Windows ptelnet client is an AIX V4 machine do the following on the AIX host :
  - 3.1 Check to see if the "ansi" terminfo entry exists and install it if necessary.
  - 3.2 Use "IsIpp -l | more" to determine if the fileset bos.terminfo.ansi.data is installed. If not, install it. You can find this fileset on the AIX installation media provided by IBM.

## 23.7 Ptelnet for AIX

Perform the following steps on a protected AIX 3.2.5 or later system.

Login as root.

```
# mkdir /usr/ptelnet_client
# cd /usr/ptelnet_client
```

### Install from Diskette:

Insert the ptelnet client diskette.

```
# tar -xvf /dev/fd0 ./aix ( or your device name if different)
#cd aix
```

### Install from CD-ROM:

Insert CD-ROM.

```
# mount -r -v cdrfs /dev/cd0 /mnt(or your device name and mount point)
# tar -xvf /mnt/aix/ptncInt.tar
# umount /mnt
```

Remove diskette or CD-ROM.

**Go to step 20.10.**

## 23.8 Ptelnet for HP-UX

Perform the following steps on a protected HP-UX 10.x system.

Login as root.

```
# mkdir /usr/ptelnet_client
# cd /usr/ptelnet_client
```

#### **Install from Diskette:**

Insert the ptelnet client diskette.

```
# tar -xvf /dev/floppy/c0t1d0 ./hpux ( or your device name if different)
# cd hpux
```

#### **Install from CD-ROM:**

Insert CD-ROM.

```
# mount -r /dev/dsk/c0t1d0 /cdrom (or your device name and mount point)
# tar -xvf /cdrom/hpux/PTNCLNT.TAR;1
# umount /cdrom
```

Remove diskette or CD-ROM.

**Go to step 20.10.**

## **23.9 Ptelnet for Solaris**

Perform the following steps on a protected Solaris 2.5.1 system. Replace the following references to <directory> with "solaris-sparc" or "solaris-x86" depending on the type of machine.

Login as root.

```
# mkdir /usr/ptelnet_client
# cd /usr/ptelnet_client
```

Determine if the volume manager is running:

```
# ps -e | grep vold
```

A response like the following indicates that the volume manager is running

```
207 ? 0:01 vold
```

#### **Install from Diskette:**

Insert the ptelnet client diskette.

If the volume manager is running do the following:

```
# volcheck
# tar -xvf /vol/dev/aliases/floppy0 ./<directory>
# eject
# cd <directory>
```

If the volume manager is not running

```
# tar -xvf /dev/fd0 ./<directory> ( or your device name if different)
# cd <directory>
```

### **Install from CD-ROM:**

Insert CD-ROM.

If the volume manager is running:

```
# tar -xvf /cdrom/T.Rex2/<directory>/ptnclnt.tar
# eject
```

If the volume manager is not running:

```
# mount -r -F hsfs /dev/dsk/c0t5d0s2 /cdrom (or your device name and mount point)
# tar -xvf /cdrom/<directory>/ptnclnt.tar
# umount /cdrom
```

Remove diskette or CD-ROM.

**Go to step 20.10.**

## **23.10 Install Ptelnet on UNIX**

```
# ./install_ptelnet_clinet
```

## **23.11 Copy T.Rex.key file**

Copy the T.Rex.key file created by the tnkey utility into the /etc directory and give it the following name "/etc/.T.Rex.key".

## **23.12 Remote Firewall Login Using Ptelnet**

The following example shows how to use ptelnet to login to the firewall as an administrator.

```
$ /usr/local/etc/ptelnet gw
Trying 192.168.10.1...
Connected to gw.lsl.com.
Escape character is '^]'.
gw.lsl.com telnet Version 3.2 ready:
-> c gw
```

Trying 206.50.87.2 port 23...  
Authentication is required before connection to firewall is permitted.  
Username: jim  
Challenge: 68858744  
Enter Response: e13a8013

AIX Version 4  
(C) Copyrights by IBM and by others 1982, 1996.  
login: jim  
jim's Password:

```
*****
* This is Freemont Avenue Software's Firewall System *
* This is a private machine and network storing proprietary information. *
* Access must be approved by FAS management. *
* Users may not copy, add, delete, modify or use information on this *
* system or network without express permission of FAS management. *
* *
* External users are being monitored. Unauthorized attempts to login *
* are recorded. Intruders will be prosecuted. *
* *
*****
Last unsuccessful login: Wed Feb 24 08:59:10 CST 1999 on /dev/tty0
Last login: Wed Feb 24 08:59:15 CST 1999 on /dev/tty0
```

# Chapter 24. External Security Server Support

## 24.1 Overview

Although T.Rex comes with a built-in security server that supports strong user authentication there will be occasions when an external security server is used instead. For example, another security server may be installed to support strong user authentication for internal systems, or modem pools.

## 24.2 Configuring DSS Support

T.Rex can be configured to act as an agent for the DEFENDER Security Server (DSS). Configuring T.Rex as a DSS agent bypasses the built-in security server of T.Rex and makes use of a similar challenge response system provided on the DSS.

DSS support is activated by creating the `/etc/firewall/dssagent.conf` file and editing five command lines. If the `dssagent.conf` file is present the `tnproxy` and `ftproxy` will rely on DSS to provide the challenge- response for strong user authentication. There are five mandatory commands which must be coded. These commands tell the T.Rex authentication manager where to find the DSS server (IP address and port number) and identify the T.Rex firewall as a DSS agent.

## Command Syntax

Each rule is contained on a line that can be up to 1023 characters long. Spaces and tabs and equal signs "=" separate fields. Comment lines begin with the `#` character. Blank lines are ignored. The format of each command is shown below:

## Mandatory Commands

**agentkey = 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88**

The agent key is used to authenticate T.Rex as a DSS agent. This key must match the T.Rex agent key defined on DSS. Use the hex values defined on DSS in place of the 11, 22, ... 88 number shown above.

**agentid = T.Rex**

The agentid must match the name defined on DSS.

**dss\_address = nnn.nnn.nnn.nnn**

The `dss_address` specifies the IP address of the DSS server.

**dss\_port = nnnn**

The `dss_port` specifies the port number that DSS listens to for agent requests.

**dss\_timeout = nnn**

The `dss_timeout` specifies the number of seconds the T.Rex authentication manager will wait before timing out.

### **Sample `/etc/firewall/dssagent.conf` file**

```
agentkey = 0x13, 0x27, 0x16, 0x45, 0x53, 0x61, 0x74, 0x28
agentid  = T.Rex
dss_address = 10.11.0.25
dss_port   = 2626
dss_timeout = 60
```

# Chapter 25. Integrity Checking and Auditing

## Trusted Computing Base (TCB) (AIX only)

The T.Rex system is designed to protect networks from intruders. When T.Rex is properly configured it is extremely difficult for an intruder to penetrate. It is also very difficult for a legitimate authorized user to access data or commands unless they are explicitly allowed. Effective implementation of security on T.Rex depends upon the correct operation of trusted programs and the integrity of related configuration files. To ensure the integrity of the T.Rex system, programs and automated procedures are provided. When the systems administrator activates TCB checking T.Rex provides the following functions:

1. Every file in the Trusted Computing Base (see TCB below) including AIX system files, T.Rex files and SOCKS files will automatically and periodically be validated.
2. Any changes to the TCB will be reported.
3. Some errors are automatically corrected when detected. Especially when the error would result in the possible reduction in the security level of the product.
4. The entire file system will be checked for the addition of any files with superuser authority that are not in the TCB data base.
5. A program to automatically check the SOCKS files and make certain the configuration is valid is also included. Conflicts between the SOCKS and T.Rex definitions of secure networks are flagged.
6. A program to validate the proper pointers to the External DNS and Internal DNS is included. This prevents incorrect configurations which could leak out DNS information.

### Trusted Computing Base (TCB)

The Trusted Computing Base (TCB) is the part of the system that enforces the security policies of the system. The TCB consists of the kernel, system configuration files that control system operation and any programs that run with system privileges. All programs that can suid to root, all programs that can setgid to administration groups and any program that is run exclusively by root or a member of the system group are part of the TCB.

The systems administrator can mark files as part of the TCB using the change TCB command **chtcb**. The systems administrator should only add software to the TCB that can be fully trusted. The TCB programs should have been fully tested. If the source code is available the program source should be carefully reviewed. If the program comes from an external source make certain it comes from a reliable source and has been thoroughly tested.

### Trusted Computing Base (TCB) Checking programs

To ensure the integrity of the operating system the TCB files must be properly protected. The **tcback** command audits the state of the TCB files to ensure their security. The **tcback** command uses the information in the **/etc/security/sysck.cfg** file to audit the TCB files.

The **tcback** program can be used to perform the following functions:



1. Check the installation of each file described in the `/etc/security/syschk.cfg` file.

## Password Checking Program

### **pwdck** command

The password check command "pwdck" verifies the correctness of the information in the `/etc/passwd` and the `/etc/security/passwd` files. The pwdck command can be used to verify the correctness of ALL users or a list of users.

The pwdck command checks the following information in the **`/etc/passwd` file**.

It validates the user name. The name must be a unique string of 8 characters or less. It can not begin with a plus sign "+" and minus sign "-" or a tilde "~" or a colon ":". The program ensures that the password field in the `/etc/passwd` file is an exclamation point "!".

The pwdck command checks the following information in the **`/etc/security/passwd` file**.

It ensures that the password attribute exists and is not blank.

# Chapter 26. Hot Backup

## 26.1 Overview

Some organizations require high levels of availability that can only be delivered using redundant systems, running with uninterruptable power supplies. The cost of a single outage for some organizations far exceeds the cost of a redundant firewall. In order to support organizations that have very high availability requirements hot backup support has been integrated into T.Rex. One simply needs to have two hardware systems connected to the same networks with a licensed copy T.Rex installed on each one.

T.Rex Hot Backup support is based on a pair of T.Rex firewall systems. Each system runs the fwpulse program to monitor the status of the other system. When one system detects a problem with its sibling it issues an alert. The alert can be mailed to one or more systems administrators or e-mailed to a pager system. The fwpulse program can be configured to perform automatic takeovers or manual takeovers.

### 26.1.1 Manual Takover

After problem notification the systems administrator must examine the status of the firewall and makes a decision if the secondary system should assume the responsibilities of the primary system. If the primary system has failed then the administrator simply issues the takeover command and the backup system automatically assumes the roles of the failed system.

### 26.1.2 Automated Takeover

In most cases fwpulse will be configured to perform automatic takeover. In this case fwpulse will perform the necessary configuration changes to takeover the failed systems IP addresses. Fwpulse will also issue warnings in the syslog and e-mail messages to the designated administrators.

The two T.Rex firewalls can be clones with identical hardware and software configurations or they can be heterogeneous systems with different hardware architectures and operating systems. For example, one T.Rex firewall can be running AIX 4.2 on a PowerPC while the other can be running Solaris 2.6 on a Sun UltraSPARC.

Three programs are used to implement the host backup function: fwpulse, fwpulsed and takeover.

## 26.2 fwpulse

The fwpulse program uses the parameters in the `/etc/firewall/fwpulse.conf` file to monitor the status of a firewall. If the firewall is found to be un-responsive, fwpulse notifies the system administrator of the situation via e-mail. It will repeat the notice once an hour if the firewall continues to be un-responsive.

fwpulse is intended to be run on a machine which runs alongside a firewall and which is configured as a firewall. fwpulse accepts no arguments and runs as a daemon. It must be run as root.

Each time fwpulse wakes up, it checks the status of each IP address on the remote firewall. It makes a syslog entry every time it finds that an IP address is not responsive. It will also send an e-

mail to alert a system administrator to the situation. It will repeat the e-mail once every half hour if the situation persists.

If fwpulse has been configured for automatic takeover it will also perform the take over functions.

Fwpulse uses a user configurable private port to monitor a remote firewall. If the remote port is not responsive for any reason, fwpulse will the router specified in the configuration file to see if it responds. Fwpulse can also be configured to check for a critical application port to see if it is responding. Querying the router as well as a critical application port provides fwpulse with more complete status on the remote firewall. This helps fwpulse avoid false alarms and unnecessary takeovers caused by transient network conditions or problems other than the remote firewall.

Fwpulse provides user exits to allow the administrator to customize the takeover procedure. Customization permits scripts to be run before and after the takeover process.

### **26.2.1 Pre-Takeover Exit**

If the `/etc/firewall/takeover` file exists, fwpulse executes it before performing the takeover. Fwpulse will only perform the takeover if the return code from the exit is zero.

### **26.2.2 Post-Takeover Exit**

After fwpulse has completed the takeover process it will check for the existence of the `/etc/firewall/post_takeover` file. If it exists fwpulse will execute it.

### **26.2.3 Reverse Takeover**

During automatic takeover fwpulse creates the `/etc/firewall/reverse_takeover` script file. This file can be used by the systems administrator when the remote firewall is ready to resume its firewall responsibilities. The `reverse_takeover` script provides user exits in the files `/etc/firewall/pre_reverse_takeover` and `/etc/firewall/post_reverse_takeover`. The `reverse_takeover` script will check for the existence of the `/etc/firewall/pre_takeover` file. If it exists `reverse_takeover` will execute it before running the `reverse_takover`. The reverse takeover will proceed only if the `pre_reverse_takeover` script returns a zero return code.

`Reverse_takeover` will execute the `/etc/firewall/post_reverse_takover` script , if it exists, after completing the reverse takeover

After automatic takeover is complete, fwpulse will exit. Since the remote firewall is no longer running, fwpulse does not need to run. Fwpulse should not be restarted until the remote firewall is ready to resume its firewall responsibilities. When fwpulse restarts it first checks to see if the `/etc/firewall/reverse_takeover` file exists. If `reverse_takeover` exists fwpulse will run it to restore the local system to its pre-takeover condition. Then fwpulse will resume its normal functions.

The configuration changes made by the fwpulse takeover functions are only temporary and will be reset with a system re-boot.

All communications between the two firewall systems are time-stamped and encrypted. This protects against spoofing attacks and replay attacks.

## 26.3 Fwpulsed

The fwpulsed program uses the parameters in the /etc/firewall/fwpulse.conf file to support the remote monitoring and automatic failover functions of fwpulse.

Fwpulsed responds to inquiries from the remote fwpulse using a private port. Fwpulse returns firewall status to the remote fwpulse program enabling it to determine the status of the local firewall. The port number used by fwpulsed is configurable. All communications are time stamped and encrypted to foil hackers..

If the firewall is only partially disabled, fwpulse can use fwpulsed to gracefully shutdown its network interfaces as part of the automated takeover by the remote system.

Fwpulsed runs under control of inetd. The /etc/services file and /etc/inetd.conf files will have to be updated to activate fwpulsed. The /etc/firewall/fwpulse.conf file also needs to be properly configured for fwpulsed to work properly..

## 26.4 Manual Takeover

The takeover program uses the parameters in the /etc/firewall/fwpulse.conf file to configure a machine to take over the responsibilities of another firewall.

Takeover is intended to be run on a machine which runs alongside a firewall. It must be run as root. Make sure the machine you run takeover on is configured as a firewall before you run takeover. The machine can be a running firewall currently handling part of the firewall activities. Takeover sets up network aliases on the machine so it can respond to the IP addresses specified in the /etc/firewall/fwpulse.conf file.

The configuration that takeover performs is only temporary in that a reboot will reset the configuration.

## 26.4 Hot Backup Configuration File

Each command is contained on a line that can be up to 1023 characters long. The format of each command is shown below:

### Mandatory Commands

```
network ip_address = a.b.c.d net_if = if_name [ netmask = mask ]  
[alias | critical_port number] [router = a.b.c.d]
```

Two or more network statements are required. One for the unprotected side and the other for the protected side. There is no practical limit on the number of network commands.

**ip\_address = a.b.c.d** Specifies a fully qualified dotted decimal IP address of the firewall.

**net\_if = if\_name** Specifies the name of a network interface on the firewall. On an AIX system typical Ethernet interface names would be en0 and lan0. On a

Solaris machine, the interface names should be coded as an alias of a network interface. For example, `le0:1`.

**router = *a.b.c.d*** Specifies the fully qualified dot decimal IP address of a router or a third machines which is connected the same network as in the preceding `ip_address`.

**netmask = *x.x.x.x*** Specifies the network mask to be used for the IP address. This option is available on AIX and Solaris only. HP-UX does not support netmasks for alias IP addresses.

**alias** Specifies that this network statement describes an alias address. An alias address is not monitored by fwpules but is only used in the takeover activities. This parameter and the `critical_port` parameter are mutually exclusive.

**critical\_port *number*** Specifies the port number of the most critical application running on this network address. This port number is used by fwpulse to gather additional information regarding the remote firewalls status. This parameter and the `alias` parameter are mutually exclusive.

Note: On Solaris the “network” statement for the primary network interfaces ( the ones without a colon “:”) should be placed after all “network” statements for alias interfaces (interfaces with names containing a colon).

**private\_port *number nnn nnn nnn nnn nnn nnn nnn***

Specifies the private port and the encryption key used by fwpulse and fwpulsed for communication via the private port. The encryption key is specified as eight three digit octol numbers which are used as a seed to generate the encryption key.

**email *email\_address***

Specifies the e-mail address which fwpulse should send warning notices to. Be sure that this e-mail address does not need to be routed through the firewall since the firewall is most likely non functional when a notice needs to be sent.

## Optional Commands

**hostname *hostname***

Specifies the fully qualified hostname for the firewall. If specified, the takeover program will reset the hostname on the machine to the hostname specified. This command is optional.

**wakup *nnn***

The wakeup value is the number of seconds the fwpulse program should wait before testing the pulse of the sibling. The default value is 300 seconds or 5 minutes.

**autotakeover**

Specifies that fwpulse will perform an automatic takeover when the remote firewall is unresponsive. The default is manual takeover.

### **threshold seconds**

Specifies the maximum number of seconds to wait for a response before concluding that the remote firewall is not responding. The default value is 3 seconds.

### **startup\_delay seconds**

Specifies the number of seconds upon startup during which fwpulse should ignore any non-response. This delay is needed during start-up to prevent unnecessary takeovers while the remote firewall is being started. The default value is 300 seconds.

## **26.5 Auto Takeover Decision Tree**

Fwpulse performs its monitoring activities in this order:

First, attempt to connect to remote private port.

Second, if the first step fails attempt to ping router.

Third, if the second step succeeded attempt to connect to the critical application port..

Connect to private port	ping router	Connect to Application Port	Conclusion and Action
Success	-	-	Interface is OK. No action.
Failed	Failed	-	Local machine not functioning properly, or network isn't working. Write warning message to syslog. Send e-mail. Do not attempt takeover.
Failed	Success	Success	Conclude fwpulse down. Write Warning message to syslog and send e-mail alert. Do not attempt a takeover.
Failed	Success	Failed	Remote firewall is down. Write Warning message to syslog. Send e-mail alert. <b>Perform autotakeover.</b>

## **26.6 Sample Fwpulse Configuration File**

```
# file: /etc/firewall/fwpulse.conf
# function: The fwpulse.conf file is used to control the fwpulse program
#           in a 2 firewall setup where one firewall can check on the
#           other firewall and issue a warning to the system administrator
#           when the other firewall does not respond.
```

```

#      It is also used by the takeover program to configure a machine
#      to takeover the ip addresses of the other firewall when instructed
#      by the operator.
#
# (C) Freemont Avenue Software, Inc. 1996-1999
#
# NOTICE TO USERS OF SOURCE CODE EXAMPLES
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AGENT OF FAS) ASSUME THE ENTIRE COST OR ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# The format of the /etc/firewall/fwpulse.conf file are as follows:
#
# hostname      <hostname>
# network       ip_address=a.b.c.d      net_if=en0
# network       ip_address=e.f.g.h      net_if=en1
# email         admin@internal.machine
# wakeup        <secs>
#
# On a Solaris machine, code the net_if parameter as the alias name of a
# network interface, e.g. le0:1.
#
#
hostname      gw.lsl.com
threshold     3
private_port  742 013 076 132 155 042 073 126
networkip_address = 128.1.2.8 net_if = en1 critical_port
networknet_if = en2 ip_address = 10.10.0.1
email admin@machine1.lsl.com

```

# Chapter 27. Testing T.Rex

This chapter shows you how to use the test programs that ship with the T.Rex firewall system.

## 27.1 Testing for IP Packet Forwarding

During the T.Rex installation process the netinet module is replaced with a version that has IP Packet forwarding **disabled**. To ensure that IP Packet forwarding has been properly disabled use the ping command to try and ping hosts on the opposite sides of the firewall. Suppose host 192.168.220.2 is on a protected network and that 198.65.130.21 is on an unprotected network.

### Addressing the unprotected network from the protected network

From the protected system enter the ping command. You should not see any response. Wait five or six seconds then interrupt the command with **CNTL c**. You should then see a message indicating that 100% of the packets sent were lost.

```
$ ping 198.65.130.21
PING 198.65.130.21: (198.65.130.21) 56 data bytes.
6 packets transmitted, 0 packets received, 100% packet loss.
```

### Addressing the protected network from the unprotected network

Repeat the process from the unprotected side. Again you should see 100% packet loss.

```
$ ping 192.168.220.2
PING 192.168.220.2: (192.168.220.2) 56 data bytes.
6 packets transmitted, 0 packets received, 100% packet loss.
```

## 27.2 Testing the Domain Name Server

When properly installed the T.Rex firewall will be running **acaching-only** Domain Name Server that can resolve host names on the unprotected network. The DNS should not be able to resolve internal names. To make certain external users can't use DNS to lookup protected hosts use the nslookup command.

From either an unprotected host or the firewall itself issue the nslookup command using the name of an host on a protected network.

```
$ nslookup fido
Server: lsli-port.sccsi.com
Address: 198.65.130.21
```

```
*** No address information is available for fido
```

## 27.3 Testing Ports with portscan



The portscan utility can be used to test which ports on the T.Rex firewall are active. Use this program to validate that applications you want to run are listening to their respective ports and that dangerous applications are not active and listening.

The portscan program is in the /usr/T.Rex directory. You can copy it to another directory if you choose.

The syntax of the portscan command is as follows:

**portscan [-l low\_port] [-h high\_port] [-f services\_file\_name] host**

**-l low\_port** This is used to specify the low port for the scan process. The port number must be a positive integer, less than 64000. The default value is 0.

**-h high\_port** This is used to specify the high port for the scan process. The port number must be a positive integer larger than the low port number. The default value is 32000. If you run with the default ports it will take less than 55 seconds to scan all 32000 ports using an ethernet LAN.

**-f services\_file\_name** This is the fully qualified path name of a file containing the service entries. This file must have the same format as the /etc/services file. If a file is not specified then portscan will use the hosts/etc/services file.

**host** The host name of the system to be scanned. This must be the name of the host or the IP address of the host in dot-decimal format ( eg. gw.lsl.com or 198.65.130.22).

## 27.4 Portscan Services file

The format of the services file is the same as the /etc/services file. Each service is listed on a separate line. The format of each line is as follows:

**ServiceName PortNumber/Protocol Aliases or comments**

**ServiceName** The service name specifies the official Internet service name. This name can be from 1 to 16 characters long. .

**PortNumber** The socket port number used for the service (0 - 64000).

**Protocol** The transport protocol used for the service ("tcp" or "udp").

The items on each line can be separated by one or more blanks or tabs. Comments begin with a '#' and continue to the end of the line. The PortNumber and the protocol can be separated by a '/' or a '.'.

## 27.5 Sample Services file

A sample services file would look like the following:

```
# file: /etc/services.gw used to test portscan
#
```

```

# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995
# All Rights Reserved
# Licensed Materials - Property of Freemont Avenue Software
#
#
echo          7/tcp
echo          7/udp
discard      9/tcp          sink null
discard      9/udp          sink null
systat       11/tcp         users
daytime      13/tcp
daytime      13/udp
netstat      15/tcp
qotd         17/tcp          quote
chargen      19/tcp          ttytst source
chargen      19/udp          ttytst source
ftp-data     20/tcp
ftp          21/tcp
telnet       23/tcp
smtp         25/tcp          mail
time         37/tcp          timserver
time         37/udp          timserver
rlp          39/udp          resource # resource location
nameserver   42/udp          name # IEN 116
whois        43/tcp          nickname
domain       53/tcp          nameserver # name-domain server
domain       53/udp          nameserver
mtp          57/tcp          # deprecated
bootps       67/udp          # bootp server port
bootpc       68/udp          # bootp client port
tftp         69/udp
rje          77/tcp          netrjs
finger       79/tcp
http         80/tcp          #WWW server
link         87/tcp          ttylink
supdup       95/tcp
hostnames 101/tcp          hostname # usually from sri-nic
iso_tsap     102/tcp
x400         103/tcp
x400-snd     104/tcp
csnet-ns     105/tcp
pop          109/tcp          postoffice
sunrpc       111/tcp
sunrpc       111/udp
auth         113/tcp          authentication
sftp         115/tcp
uucp-path    117/tcp
nntp         119/tcp          readnews untp # USENET News Transfer Protocol
ntp          123/tcp
NeWS         144/tcp
snmp         161/udp          # snmp request port
snmp-trap    162/udp          # snmp monitor trap port
smux         199/tcp          # snmpd smux port
src          200/udp          # System Resource controller
exec         512/tcp          # rexec
biff         512/udp          comsat
login        513/tcp          # rlogin - dangerous on a firewall
who          513/udp          whod
shell        514/tcp          # rshd - dangerous on a firewall
syslog       514/udp
printer      515/tcp          spooler # line printer spooler
talk         517/udp
ntalk        518/udp
efs          520/tcp          # for LucasFilm
route        520/udp          router routed

```

timed	525/udp	timeserver
tempo	526/tcp	newdate
courier	530/tcp	rpc
conference 531/tcp	chat	
netnews	532/tcp	readnews
netwall	533/udp	# -for emergency broadcasts
uucp	540/tcp	uucpd # uucp daemon
new-rwho	550/udp	
remotefs	556/tcp	rfs_server rfs # Brunhoff remote filesystem
rmonitor	560/udp	
monitor	561/udp	
securid	755/udp	# added by rjl for security dynamics
socks	1080/tcp	# socks
instsrv	1234/tcp	# network install service
ingreslock	1524/tcp	
writesrv	2401/tcp	# temporary port number
securidprop	5510/tcp	# added by rjl for security dynamics

## 27.6 Sample portscan output

The following example shows the normal output of the portscan command. The default output will go the users terminal. The output can also be directed to a file if you want to save it, as shown below.

```
$ /usr/T.Rex/portscan -h 6000 gw > portscan.ouput.941210
```

To display the output on you screen enter the command:

```
$ /usr/T.Rex/portscan -h 6000 gw
```

```
portscan: trying stream ports between 0 and 6000 on gw
```

service	name	port/tcp	message
	echo	7/tcp	is alive on gw.lsli.com (198.65.130.22)
	discard	9/tcp	is alive on gw.lsli.com (198.65.130.22)
	daytime	13/tcp	is alive on gw.lsli.com (198.65.130.22)
	chargen	19/tcp	is alive on gw.lsli.com (198.65.130.22)
	ftp	21/tcp	is alive on gw.lsli.com (198.65.130.22)
	telnet	23/tcp	is alive on gw.lsli.com (198.65.130.22)
	smtp	25/tcp	is alive on gw.lsli.com (198.65.130.22)
	time	37/tcp	is alive on gw.lsli.com (198.65.130.22)

WARNING! DNS (53/tcp) is running. Make certain DNS is running as a caching-only server on the firewall, and can only resolve external hosts.

domain	53/tcp	is alive on gw.lsli.com (198.65.130.22)
smux	199/tcp	is alive on gw.lsli.com (198.65.130.22)
	1026/tcp	is alive on gw.lsli.com (198.65.130.22)
writesrv	2401/tcp	is alive on gw.lsli.com (198.65.130.22)
cppbrowse	4242/tcp	is alive on gw.lsli.com (198.65.130.22)
	6000/tcp	is alive on gw.lsli.com (198.65.130.22)

```
ports scanned = 6001, number active = 14
```

## Sample portscan output (with warnings)

The following output shows the warning messages generated when dangerous applications are running on the target host. The command is run using all the default values. The warnings are for **rexecd**, **rlogin**, **rshd** and **uucp**, none of which should be running on a firewall.

### \$ portscan duo

portscan: trying stream ports between 0 and 32000 on duo

service	name	port/tcp	message
	echo	7/tcp	is alive on duo.lsli.com (192.168.220.2)
	discard	9/tcp	is alive on duo.lsli.com (192.168.220.2)
	daytime	13/tcp	is alive on duo.lsli.com (192.168.220.2)
	chargen	19/tcp	is alive on duo.lsli.com (192.168.220.2)
	ftp	21/tcp	is alive on duo.lsli.com (192.168.220.2)
	telnet	23/tcp	is alive on duo.lsli.com (192.168.220.2)
	smtp	25/tcp	is alive on duo.lsli.com (192.168.220.2)
	time	37/tcp	is alive on duo.lsli.com (192.168.220.2)

WARNING! DNS (53/tcp) is running. Make certain DNS is running as a caching-only server on the firewall, and can only resolve external hosts.

domain	53/tcp	is alive on duo.lsli.com (192.168.220.2)
smux	199/tcp	is alive on duo.lsli.com (192.168.220.2)

WARNING! Use of exec on port 512 is considered UNSAFE!

exec	512/tcp	is alive on duo.lsli.com (192.168.220.2)
------	---------	------------------------------------------

WARNING! Use of login on port 513 is considered UNSAFE!

login	513/tcp	is alive on duo.lsli.com (192.168.220.2)
-------	---------	------------------------------------------

WARNING! Use of shell on port 514 is considered UNSAFE!

shell	514/tcp	is alive on duo.lsli.com (192.168.220.2)
-------	---------	------------------------------------------

WARNING! Use of uucp on port 540 is considered UNSAFE!

uucp	540/tcp	is alive on duo.lsli.com (192.168.220.2)
	1026/tcp	is alive on duo.lsli.com (192.168.220.2)
writesrv	2401/tcp	is alive on duo.lsli.com (192.168.220.2)
cppbrowse	4242/tcp	is alive on duo.lsli.com (192.168.220.2)
	6000/tcp	is alive on duo.lsli.com (192.168.220.2)
spc	6111/tcp	is alive on duo.lsli.com (192.168.220.2)
	7685/tcp	is alive on duo.lsli.com (192.168.220.2)

ports scanned = 32001, number active = 20

4 WARNING(S) ISSUED.

notes:

## Chapter 28. Sources for Security Information

Installation and configuration of a firewall does not solve all your security problems. The person(s) responsible for network security needs to understand the current security exposures for all the systems installed on their network not just the firewall. Since new methods of attack are continually being created network security administration requires constant vigilance. The resources listed in this chapter will help you remain current with the latest security issues.

### 28.1 Computer Emergency Response Team (CERT)

The Computer Emergency Response Team (CERT) was set up by the Defense Advanced Research Projects Agency (DARPA) to monitor computer security and break-in activities reported on the Internet. CERT helps the Internet community respond to security problems associated with the net and provides a hotline for reporting Internet security incidents. If you believe your system has been compromised you can contact CERT at (412)-268-7090. CERT personnel answer the phone between the hours of 8:30 am and 5:00 pm EST (GMT-5)/EDT(GMT-4).

CERT's Internet **e-mail** address is **cert@cert.org**.

CERT also acts as a security information clearing house and provides this information on a web server at <http://www.cert.org/>.

#### Quarterly Summaries

Each quarter the CERT Coordination Center issues the CERT summary to draw attention to the types of attacks reported to their incident response team. The summary contains pointers to sources of information for dealing with the problems. Past CERT Summaries can be found at the following URL: <http://www.cert.org/summaries/>

#### CERT/CC Current Activity

The CERT/CC Current Activity web page contains a summary of the most current high-impact security incidents at the following URL: [http://www.cert.org/current/current\\_activity.html](http://www.cert.org/current/current_activity.html)

### 28.2 Computer Incident Advisory Capability (CIAC)

CIAC provides on-call technical assistance and information to Department of Energy (DOE) sites faced with computer security incidents and is an element of the Computer Security Technology Center (CSTC) which supports the Lawrence Livermore National Laboratory (LLNL).

CIAC was established in 1989 to serve the DOE Community. CIAC is one of two oldest response teams and is recognized nationally and internationally for its contributions to the Internet community. CIAC is a founding member of FIRST, the Forum of Incident Response and Security Teams, a global organization established to foster cooperation and coordination among computer security teams worldwide.

To protect and ensure authenticity all of CIAC electronic publications, bulletins and advisories are signed with a PGP encryption key. Use this key to validate CIAC publications sent to you.

The CIAC contains a wealth of computer security information including bulletins, tools, documents

and a CIAC mailing list. To subscribe to the CIAC Mailing list send an e-mail to [majordomo@rumpole.llnl.gov](mailto:majordomo@rumpole.llnl.gov). In the BODY (not subject) of the message put: [subscribe ciac-bulletin](#)

The CIAC can be reached at:

E-mail:	<a href="mailto:ciac@llnl.gov">ciac@llnl.gov</a>
Web Site:	<a href="http://ciac.llnl.gov/">http://ciac.llnl.gov/</a>
FAX:	+1 925 423-8002
Anonymous FTP:	<a href="http://ciac.llnl.gov">ciac.llnl.gov</a> or <a href="http://ftp.ciac.org">ftp.ciac.org</a>
Phone:	+1 925 422-8193 (08:00 - 18:00 U.S. Pacific Standard Time (GMT-8), 16:00 - 02:00 GMT)

## 28.3 Mailing Lists

The following table identifies several security related mailing lists.

Security Mailing List	Description
<a href="mailto:alert@iss.net">alert@iss.net</a>	<b>The Internet Security Systems Alert List.</b> This list contains alerts and product announcements from ISS. To subscribe to this list go to <a href="http://iss.net/vd/mailist">http://iss.net/vd/mailist</a> .
<a href="mailto:bugtraq@netspace.org">bugtraq@netspace.org</a>	<b>BUGTRAQ Mailing List.</b> This is one of the best sources for information on recent bugs found in UNIX operating systems. To subscribe to this list send an e-mail with the command <a href="#">SUBSCRIBE BUGTRAQ</a> in the body.
<a href="mailto:firewall-wizards@nfr.net">firewall-wizards@nfr.net</a>	<b>The Firewall Wizards Mailing List.</b> This a moderated forum for advanced firewall administrators. You can subscribe at <a href="http://www.nfr.net/forum/firewall-wizards.html">http://www.nfr.net/forum/firewall-wizards.html</a> .
<a href="mailto:linux-security-request@redhat.com">linux-security-request@redhat.com</a>	<b>The Linux Security List.</b> To subscribe to the Linux Security list send an e-mail with the command <a href="#">subscribe</a> in the subject line.
<a href="mailto:majordomo@lists.gnac.net">majordomo@lists.gnac.net</a>	<b>The Firewall Mailing List.</b> This is an un-moderated list that focuses on firewall security. To subscribe send an e-mail with the command <a href="#">subscribe firewalls</a> in the body.

Security Mailing List	Description
<a href="mailto:majordomo@uow.edu.au">majordomo@uow.edu.au</a>	<b>The Intrusion Detection Systems List.</b> The members of this list discuss real-time intrusion detection techniques. To subscribe send an e-mail with the command <a href="#">subscribe ids</a> in the body.
<a href="mailto:listserv@listserv.ntbugtraq.co">listserv@listserv.ntbugtraq.co</a>	<b>The NTBUGTRAQ List.</b> This list tracks security exposures related to the Microsoft NT system. To subscribe send an e-mail with the command <a href="#">ntbugtraq firstname lastname</a> in the body.

## 28.4 Usenet News Groups

The following table identifies several security related news groups. There are other news groups but these seem to be the most useful.

Security News Group	Description
alt.2600	This group contains hacking and cracking exploits. The signal to noise ratio is poor but there is an occasional gem to be found.
alt.2600.crackz	This group concentrates on cracking and is a distribution point for cracks and wares.
alt.computer.security	This group is concerned with general computer security.
alt.hackers.malicious	This group focuses on tools and methods that cause damage to their targets. You can find Denial of Service attacks, cracking and viruses here.
comp.lang.java.security	This group has interesting security information pertaining to the Java language. Java security defects often show up here first.
comp.security	This group has general discussions on computer security.
comp.security.firewalls	There are noteworthy discussions here regarding firewalls
comp.security.unix	This group has worthwhile discussions regarding UNIX security, that is also applicable for Linux.





notes:

## Chapter 29. T.Rex Messages

This chapter documents messages generated by T.Rex which are not standard UNIX messages. The messages are grouped by the program name. All syslog messages start with date, time, gateway name, proxy name and the process ID. This is followed by the message text. Messages that start with the word Error: indicate that T.Rex has detected an error. Messages that start with Security Alert: indicate some one has tried to gain unauthorized access to system resources.

### 29.1 APROXY Messages

All syslog messages issued by the application proxy begin with: **date time gateway\_name aproxy[pid]:**

For the sake of brevity only the message text is listed below.

#### **Error: could not malloc . Text explaining reason**

The aproxy was unable to acquire memory for control blocks. There is a sever memory shortage.

**Action:** Try again later.

#### **Error: open failed for aproxy.conf file. Text explaining reason.**

Aproxy was unable to open its configuration file.

**Action:** Examine the error message to determine why /etc/firewall/aproxy.conf could not be opened. Does the file exist and have the correct permissions.

#### **Error: no value specified for timeout on line nnn of /etc/firewall/aproxy.conf**

Aproxy was expecting a numeric value following the keyword timeout.

**Action:** Edit the aproxy.conf file and insert a numeric value after the keyword timeout. The timeout must be a positive integer less than 1000000000.

#### **Error: non numeric value 'xyz' specified for timeout on line nnn of /etc/firewall/aproxy.conf**

Aproxy was expecting a numeric value following the keyword timeout.

**Action:** Edit the aproxy.conf file and insert a numeric value after the keyword timeout. The timeout must be a positive integer less than 1000000000.

#### **Error: DNS = yes option is no longer supported .**

Aproxy does not support DNS =yes.

**Action:** Remove this line from the /etc/firewall/aproxy.conf file.

**Error: no value specified for trace on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects to find the value “yes” or “no” after the keyword trace.

**Action:** To activate an aproxy trace code the parameter “yes”. To deactivate the trace code “no”. Otherwise remove the trace line.

**Error: invalid value specified for trace on line nnn of /etc/firewall/aproxy.conf. Should be yes or no.**

Aproxy found a keyword other than yes or no after the trace command.

**Action:** Edit the aproxy.conf file so that either yes or no follow the trace command.

**Error: no value specified for debug on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects to find a positive integer following the debug keyword.

**Action:** To activate an aproxy trace code the parameter “yes”. To deactivate the trace code “no”. Otherwise remove the trace line.

**Error: non numeric value ‘xyz’ specified for debug on line nnn of /etc/firewall/aproxy.conf .**

Aproxy expects to find a positive integer following the debug keyword.

**Action:** To activate an aproxy trace code the parameter “yes”. To deactivate the trace code “no”. Otherwise remove the trace line.

**Error: no value specified for gproxpath on line nn of /etc/firewall/aproxy.conf.**

Aproxy expected to find a valid path name following the gproxpath keyword.

**Action:** Edit the aproxy.conf file and insert a valid path name after the gproxpath keyword. Check the Aproxy Administration Chapter for details.

**Error: gproxpath name ‘/x/y/x’ on line nnn of /etc/firewall/aproxy.conf is longer than 200 characters .**

Aproxy requires the path name to be less than 200 characters long.

**Action:** Change the pathname to an acceptable value and restart aproxy. Check the Aproxy Administration Chapter for details.

**Error: gproxpath name is ‘/x/y/z’ is not allowed.**

Aproxy does not allow the pathname to begin with any of the following character strings:

“/dev”, “/etc”, “/lib”, “/sbin”, “/unix”, “/usr”, or “/var”. It also does not allow use of the root directory “/”

**Action:** Change the pathname to an acceptable value and restart aproxy. Check the Aproxy Administration Chapter for details.

**Error: no value specified for nprocs on line nn of /etc/firewall/aproxy.conf.**

Aproxy expects three positive integers to follow the nprocs command. These numbers specify the number of processes to start, the minimum number of spare idle processes and the maximum number of spare idle processes.

**Action:** Insert the three numbers after the nprocs keyword as specified in the Aproxy Administration Chapter.

**Error: non numeric value specified for nprocs on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects three positive integers to follow the nprocs command. These numbers specify the number of processes to start, the minimum number of spare idle processes and the maximum number of spare idle processes.

**Action:** Insert the three numbers after the nprocs keyword as specified in the Aproxy Administration Chapter.

**Error: nprocs must be greater than 0 on line nnn of /etc/firewall/aproxy.conf.**

The initial number of processes to start must be an integer larger than 0.

**Action:** Edit the line and restart aproxy.

**Error: non numeric value 'xyz' specified for minspare on line nnn of /etc/firewall/aproxy.conf.**

The second parameter following nprocs specifies the minimum number of spare processes and must be an integer greater than 0.

**Action:** Edit the line and restart aproxy.

**Error: minspare must be greater than 0 on line nnn of /etc/firewall/aproxy.conf.**

The minimum number of spare processes must be an integer larger than 0.

**Action:** Edit the line and restart aproxy.

**Error: non numeric value 'xyz' specified for maxspare on line nnn of /etc/firewall/aproxy.conf.**

The third parameter following nprocs specifies the maximum number of spare processes and must be an integer greater than minspare.

**Action:** Edit the line and restart aproxy.

**Error: maxspare must be greater than minspare (nnn) on line nnn of /etc/firewall/aproxy.conf.**

The maximum number of spare processes must be an integer larger than minspare.

**Action:** Edit the line and restart aproxy.

**Error: no value specified for maxuse on line nn of /etc/firewall/aproxy.conf.**

Aproxy expects a positive integer to follow the maxuse command. This number specifies the maximum number of transaction a process will handle before being retired and replace by a fresh

process.

**Action:** Insert the number after the maxuse keyword and restart aproxy.

**Error: non numeric value 'xyz' specified for maxuse on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects a positive integer to follow the maxuse command. This number specifies the maximum number of transaction a process will handle before being retired and replace by a fresh process.

**Action:** Edit the line and restart aproxy.

**Error: maxuse must be greater than 0 on line nnn of /etc/firewall/aproxy.conf.**

The maximum number of uses for a process before retirement must be an integer larger than 0.

**Action:** Edit the line and restart aproxy.

**Error: no value specified for maxprocs on line nn of /etc/firewall/aproxy.conf.**

Aproxy expects a positive integer to follow the maxprocs command. This number specifies the maximum number of concurrent processes that will be started by aproxy.

**Action:** Insert the number after the maxprocs keyword and restart aproxy.

**Error: non numeric value 'xyz' specified for maxprocs on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects a positive integer to follow the maxprocs command. This number specifies the maximum number of concurrent processes that will be started by aproxy.

**Action:** Edit the line and restart aproxy.

**Error: maxprocs must be greater than 0 on line nnn of /etc/firewall/aproxy.conf.**

The maximum number of concurrent processes must be an integer larger than 0.

**Action:** Edit the line and restart aproxy.

**Error: no value specified for checkprocs on line nn of /etc/firewall/aproxy.conf.**

Checkprocs specifies the time interval in seconds that aproxy waits before checking on the number of child processes. Checkprocs must be a positive integer.

**Action:** Insert the number after the checkprocs keyword and restart aproxy.

**Error: non numeric value 'xyz' specified for checkprocs on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects a positive integer to follow the checkprocs command. This number specifies the time interval in seconds that aproxy waits before checking on the number of child processes.

**Action:** Edit the line and restart aproxy.

**Error: checkprocs must be greater than 0 on line nnn of /etc/firewall/aproxy.conf.**

The checkprocs time interval must be an integer larger than 0.

**Action:** Edit the line and restart aproxy.

**Error: no value specified for bufsize on line nn of /etc/firewall/aproxy.conf.**

Bufsize specifies the aproxy buffer size in bytes. Bufsize must be a positive integer.

**Action:** Insert the number after the bufsize keyword and restart aproxy.

**Error: non numeric value 'xyz' specified for bufsize on line nnn of /etc/firewall/aproxy.conf.**

Aproxy expects a positive integer to follow the bufsize command. This number specifies the aproxy buffer size in bytes.

**Action:** Edit the line and restart aproxy.

**Error: bufsize must be greater than 0 and less than 65535 on line nnn of /etc/firewall/aproxy.conf.**

The bufsize value must be an integer larger than 0 and less than 64535.

**Action:** Edit the line and restart aproxy.

**Error: Could not realloc. Text explaining error.**

Aproxy tried to re-allocate memory to hold the permit rules and failed.

**Action:** There may be a temporary memory shortage. Try to restart aproxy. If this does not work examine the system log for other error messages. If running AIX use smit to process the error log. If the problem persists try to re-boot the system.

**Error: invalid record on line nnn of /etc/firewall/aproxy.conf, unknown keyword. Service denied.**

Aproxy encountered an invalid keyword.

**Action:** Edit the aproxy.conf file and restart aproxy. Check the Aproxy Administration Chapter for details.

**Error: could not malloc io buffer.**

Aproxy could not allocate the I/O buffer. There may be a temporary memory shortage.

**Action:** Examine the system performance monitor to see if the system is low on storage.

**Error: no gproxpath record found.**

Gproxpath specifies the directory that aproxy will run under. It is a mandatory command.

**Action:** Edit the /etc/firewall/aparoxy file and add this command. Check the Aproxy Administration Chapter for details.

**Error: permit statement on line nn of /etc/firewall/aproxy.conf is missing parameters.**

Aproxy found an invalid permit line that contains no parameters.

**Action:** Edit the file and restart aproxy.

**Error: permit/deny statement on line nn of /etc/firewall/aproxy.conf is missing parameters.**

Aproxy found an invalid permit or deny line that contains no parameters.

**Action:** Edit the file and restart aproxy.

**Error: from address too long on line nn of /etc/firewall/aproxy.conf.**

The from IP address is too long.

**Action:** Edit the file and restart aproxy.

**Error: permit/deny statement on line nn of /etc/firewall/aproxy.conf is missing “to” keyword.**

The “to” keyword should follow the from IP address.

**Action:** Edit the file and restart aproxy.

**Error: to address too long on line nn of /etc/firewall/aproxy.conf.**

The to IP address is too long.

**Action:** Edit the file and restart aproxy.

**Error: unknown service ‘xyz’ can not decode as port, on line nnn of /etc/firewall/aproxy.conf .**

The service name following the to address could not be found in the /etc/services file.

**Action:** Either add the service names or correct the spelling. Then restart aproxy.

**Error: permit/deny statement on line nn of /etc/firewall/aproxy.conf is missing the “redirect” keyword.**

The redirect keyword is missing from the permit statement.

**Action:** Correct the statement and restart aproxy.

**Error: unbalanced ‘(’ and ‘)’ after redirect on line nn of /etc/firewall/aproxy.conf**

Multiple IP address can be specified if enclosed between ‘(’ and ‘)’. If either one is coded then both must be coded.

**Action:** Fix the error and restart aproxy.

**Error: Syntax error after redirect on line nn of /etc/firewall/aproxy.conf.**

There were no IP addresses found between the “()”.



**Action:** Correct the error and restart aproxy.

## 29.2 FTPROXY Messages

All syslog messages issued by the ftp proxy begin with: **date time gateway\_name ftproxy[pid]:**  
For the sake of brevity only the message text is listed below.

### **User QUIT. source = *hostname (IPaddress)***

This message is issued if the user enters the QUIT command before completing user authentication. Ftproxy breaks the connection.

**Action:** none.

### **Error: keyword 'USER' not specified. source = *hostname (IPaddress)*.**

The client code did not follow the FTP protocol. The keyword **USER** was expected but not received. Ftproxy sends error message 501 to the client and lets the client try again.

**Action:** none.

### **Error: keyword 'PASS' not specified. source = *hostname (IPaddress)*.**

The client code did not follow the FTP protocol. The keyword **PASS** was expected but not received. The user is allowed four tries before the connection is broken.

**Action:** none.

### **Error: no password specified for user\_name. source = *hostname (IPaddress)*.**

The client did not send a password following the keyword PASS. The user is allowed four tries before the connection is broken.

**Action:** none.

### **Access denied to *user\_name* source = *hostname (IPaddress)*.**

The user record does not allow this user FTP access. The user is allowed four tries before the connection is broken.

**Action:** If the systems administrator wants to allow this user access to FTP services then they must update the users record with the gwuser command.

### **Access denied to *user\_name* source = *hostname (IPaddress)*. NULL password in user record.**

The user record contains a NULL password and user authentication is required for protected users. The user is allowed four tries before the connection is broken.

**Action:** If the systems administrator wants to allow this user access to FTP services then they must add a password to the users record with the gwuser command.

### **Unable to read response to challenge. *system error***

The ftproxy was unable to read the response to the challenge. The system error should indicate the reason why. This message is issued when the user record specifies the use of a CRYPTOCARD or

SecureNet Key card for strong user authentication and the user is trying to connect from a unsecured host.

**Action:** Examine the system error message to determine what to do.

**Invalid IP address 'IPaddress'. length = nn.**

An invalid IP address was detected. This could be a system error. The connection is broken.

**Action:** Report the error to FAS.

**open failed for */etc/firewall/securenets* file.**

The securenets file is damaged or missing. All IP addresses are treated as unsecured.

**Action:** If the file is missing you must recreate the file as described in chapter 3.

**Error: open failed for *file\_name* file.**

The ftproxy was unable to open its configuration file. The connection is broken.

**Action:** Determine the reason the ftproxy file could not be opened. If necessary recreate it using the instructions in chapter 3.

**Error: no value specified for auth on line *nn* of *file\_name*.**

Line number nn of the ftproxy configuration file is in error. The connection is broken.

**Action:** Fix the line in error using the instructions in chapter 6.

**Error: invalid auth value '*xyz*' on line *nn* of *file\_name*.**

Line nn of the ftproxy configuration file contains an invalid character string after the auth = keywords. The only valid values are yes or no. The connection is broken.

**Action:** Fix the line in error using the instructions in chapter 6.

**Error: no value specified for timeout on line *nn* of *file\_name*.**

Line number nn is in error. The connection is broken.

**Action:** Fix the line in error using the instructions in chapter 6.

**Error: non numeric value '*xyz*' specified for timeout on line *nn* of *file\_name*.**

Line nn of the ftproxy configuration file contains an invalid character string after the timeout = keywords. The value must be a positive decimal integer. The connection is broken.

**Action:** Fix the line in error using the instructions in chapter 6.

**Error: no value specified for ftproxpath on line *nn* of *file\_name*.**

Line nn of the ftproxy configuration file is missing a path name following the ftproxpath keyword. The connection is broken.

**Action:** Fix the invalid line using the instructions in chapter 6.

**Error: ftproxpath name 'xyz' is too long on line *nn* of *file\_name*.**

The path name must be less than 255 characters.

**Action:** Fix the error on line *nn* of the ftproxy configuration file.

**Error: ftproxpath name 'xyz' on line *nn* of *file\_name* is not allowed.**

A forbidden path name was coded for ftproxpath. Certain path names such as "/", "etc", "/usr" are not allowed. The connection is broken.

**Action:** Specify a path name that does not contain system code.

**Error: No value specified for userid on line *nn* of *file\_name*.**

Expected a userid after the keyword userid and found nothing. The connection is broken.

**Action:** Fix the error on line *nn* of the ftproxy configuration file.

**Error: User name 'xyz' is too long on line *nn* of *file\_name*.**

The user name is longer than 8 characters, which is the maximum user name length allowed by the operating system.

**Action:** Specify a valid user name for the UNIX system. T.Rex allows 16 character user names because T.Rex user's don't actually logon to the system.

**Error: user name 'xyz' on line *nn* of *file\_name* is not a valid user**

The user name that the ftproxy program runs under must be a valid user name defined to the operating system.

**Action:** Fix the error on line *nn* of the ftproxy configuration file by using a valid user id on the system.

**Error: user name '*name*' on line *nn* of *file\_name* is rejected because user id = 0.**

The user name produces the root user id of zero. The ftproxy will not run as root for security reasons. The connection is broken.

**Action:** Assign another user id for ftproxy to run under.

**Error: no value specified for grouid on line *nn* of *file\_name*.**

Expected a groupid after the keyword groupid and found nothing. The connection is broken.

**Action:** Assign a valid groupid after the keyword groupid. See chapter 6 for details.

**Error: Group name 'xyz' is too long on line *nn* of */etc/firewall/ftproxy.conf*.**

The group name is longer than 8 characters, which is the maximum user name length allowed by the operating system.

**Action:** Specify a valid group name for the UNIX system.

**Error: Group name 'xyz' on line nn of /etc/firewall/ftproxy.conf is not a valid group**

The group name that the ftproxy program runs under must be a valid group name defined to the operating system.

**Action:** Fix the error on line nn of the ftproxy configuration file by using a valid group name defined to the operating system.

**Error: timeout command not found in /etc/firewall/ftproxy.conf.**

The ftproxy did not find the timeout command in the ftproxy configuration file. The connection is broken.

**Action:** Add the timeout command as defined in chapter 6.

**Error: userid command not found in /etc/firewall/ftproxy.conf.**

The ftproxy did not find the userid command in the ftproxy configuration file. The connection is broken.

**Action:** Add the userid command as defined in chapter 6.

**Error: groupid command not found in /etc/firewall/ftproxy.conf.**

The ftproxy did not find the groupid command in the ftproxy configuration file. The connection is broken.

**Action:** Add the groupid command as defined in chapter 6.

**Error: ftproxpath command not found in /etc/firewall/ftproxy.conf.**

The ftproxy did not find the ftproxpath command in the ftproxy configuration file. The connection is broken.

**Action:** Add the ftproxpath command as defined in chapter 6.

**Error: can not get peername.**

The peername() function failed. The connection is broken.

**Action:** Check for link failure.

**Error: getsockname failed: *system error message*.**

The getsockname() call failed. The system is unable to determine the local IP address. The connection is broken.

**Action:** Examine the system error message.

**Security Alert: ftp attempt to unsecured port *IPaddress* using secure source IP address *IPaddress*.**

A remote host with a secure source IP address tried to connect to an unsecured port. This looks like an IP address spoofing attempt. The connection is broken.

**Action:** Examine the /etc/firewall/securenets file and the /etc/firewall/secureports file to make certain the two files are not misconfigured. A securenet should not be attached to an unsecured port.

**Security Alert: unauthorized ftp attempt to *IPaddress* from *IPaddress* aka as *host\_name*.**

A remote host was rejected. The connection is broken.

**Action:** This may be a break in attempt.. Check it out.

**Error: Can Not display MOTD from *IPaddress* to *IPaddress* (*host\_name*)**

Ftproxy was unable to display the Message Of The Day file /etc/motd. The connection is broken.

**Action:** Check to see if the file exists and is readable. If the file does not exist create one.

**Error: Can Not display FTP Proxy msg from *IPaddress* to *IPaddress* (*hostname*)**

The ftproxy was unable to display "220 *hostname* FTP Proxy (T.Rex Version) to *IPaddress* (*hostname*).\" The connection is broken.

**Action:** Check for link failure.

**Authentication failure: source host = *hostname* (*IPaddress*) user(s) = *list\_of\_user names* .**

The user failed the authentication test multiple times. The user id used for each attempt is listed. If it is the same user id then the person was failing to provide the correct password or was failing the strong user authentication. The connection is broken.

**Action:** Check to see if this is an attempted break in or simply a user forgetting their password.

**authenticate user = *username*.**

The user was authenticated. This is normal.

**Action:** none.

**Unable to write '230 User authenticated msg'.**

The ftproxy was unable to write the authentication message to the ftp client. The connection is broken.

**Action:** Check for a link failure.

**Error: unable to get quote site dest**

The ftproxy was unable to read the 'quote site *hostname*' from the ftp client. The connection is broken.

**Action:** Check for a link failure.

**Error: expecting site destination.**

Ftproxy was expecting the **site** keyword to follow the **quote** keyword. The connection is broken.

**Action:** Check the FTP client for proper configuration. Some PC FTP client software may require the setting of preferences for their GUI interfaces.

**Error: hostname (hostname) unknown, error\_number**

The ftproxy could not find the hostname using either the External DNS or the Internal DNS. The connection is broken.

**Action:** none. The user will have to try again.

**permit user = user\_name source = hostname (IPaddress) connect to destination\_host (IPaddress)**

The user is permitted to connect to the destination host. This is a normal message.

**Action:** none.

**Error: select: system error message**

The select() call failed and returned a negative value. The connection is broken.

**Action:** Examine the system error message to determine the reason for the select failure.

**exit user = user\_name source = hostname (IPaddress) cmds = nn in = num\_bytes out = num\_bytes duration = seconds**

The system timed out waiting for additional activity. The number of user commands, the number of input bytes and output bytes are recorded along with the duration in seconds. This is a normal message. The connection is broken.

**Action:** none.

**Error: chdir directory\_name failed: system error message**

Ftproxy was unable to change to its user directory. The connection is broken.

**Action:** Examine the system error message to determine the cause of the chdir failure. Check to see if the directory exists.

**Error: chroot directory\_name failed: system error message**

Ftproxy was unable to change its root directory to its user directory. The connection is broken.

**Action:** Examine the system error message to determine the cause of the chroot failure. Check to see if the directory exists.

**Error: cannot setgid *groupid*: system error message**

Ftproxy was unable to set its groupid. The connection is broken.

**Action:** Examine the system error message to determine the cause of the setgid failure.

**Error: cannot setuid *userid*: system error message**

Ftproxy was unable to set its userid. The connection is broken.

**Action:** Examine the system error message to determine the cause of the setuid failure.

**Error: open failed for *filename*: system error message.**

Ftproxy failed to open the ftproxy configuration file. The connection is broken.

**Action:** Check to see if the file exists. If the file is damaged or missing then rebuild it.

**Error: permit is missing parameters on line *nn* of *filename*.**

There are no parameters following the permit command. The connection is broken.

**Action:** Correct the error.

**Error: deny is missing parameters on line *nn* of *filename*.**

There are no parameters following the deny command. The connection is broken.

**Action:** Correct the error.

**Permission denied to connect user *user\_name* from *IPaddress* to *IPaddress***

The connection was denied by default. That is there was no permit command to allow it. The connection is broken.

**Action:** You may want to check for a possible breakin attempt.

**Error: either users or groups must follow permit on line *nn* of *filename*.**

Ftproxy expects either **users** or **groups** to follow the **permit** command. The connection is broken.

**Action:** Correct the error by adding either users or groups after the permit command.

**Error: missing user parameter on line *nn* of *filename*.**

Ftproxy expected a user name to follow the keyword **users**. Instead it found nothing. The connection is broken.

**Action:** Correct the error by adding a user name or an asterisk.

**Error: missing group parameter on line *nn* of *filename*.**



Ftproxy expected a group name to follow the keyword **groups**. Instead it found nothing. The connection is broken.

**Action:** Correct the error by adding a group name or an asterisk.

**Error: missing from parameters on line *nn* of *filename*.**

Ftproxy expected the keyword **from** followed by an IPaddress. Instead it found nothing. The connection is broken.

**Action:** Correct the error.

**Error: missing source IP address on line *nn* of *filename*.**

Ftproxy expected to find the source IP address following the from keyword. Instead it found nothing. The connection is broken.

**Action:** Correct the error.

**Error: missing to parameters on line *nn* of *filename*.**

Ftproxy expected the keyword **to** followed by an IPaddress. Instead it found nothing. The connection is broken.

**Action:** Correct the error.

**Error: missing destination IP parameters on line *nn* of *filename*.**

Ftproxy expected to find the destination IP address following the to keyword. Instead it found nothing. The connection is broken.

**Action:** Correct the error.

**open failed for filename: system error message.**

The ftproxy was unable to open the gateway user file. The connection is broken.

**Action:** Use the system error message to determine the reason for the failure.

**Invalid record for user *user\_name* in *filename*. Not version 2.**

The ftproxy found an invalid record type in the file. The connection is broken.

**Action:** Try to list the record using the gwuser utility.

**Invalid record for user *user\_name* in *filename*. Internal shell name is NULL.**

The ftproxy found an invalid record type in the file. The connection is broken.

**Action:** Try to list the record using the gwuser utility. Rebuild the user record.

**Invalid record for user *user\_name* in *filename*. External shell name is NULL.**

The ftproxy found an invalid record type in the file. The connection is broken.

**Action:** Try to list the record using the gwuser utility. Rebuild the user record.

**read error searching for user *user\_name* in *filename*.**

The ftproxy encountered a read error while searching for the user's record. The connection is broken.

**Action:** Try to list the record using the gwuser utility. Rebuild the user record.

## 29.3 Fwpulse Messages

All syslog messages issued by fwpulse begin with: **date time gateway\_name genroxy[pid]:**  
For the sake of brevity only the message text is listed below.

### **Error: fwpulse cannot fork. Fwpulse is not active.**

Fwpulse can not create a daemon process. Therefore it is not active.

**Action:** Try starting it again. If it fails again you may have to re-boot the system. Notify FAS.

### **Firewall IP address xxx.xxx.xxx.xxx not responding to ping**

Fwpulse was unable to ping the IP address of the LAN Adapter with the specified address.

**Action:** Check to see if the other firewall is active. See if its other LAN adapters are active. If the firewall is down use the takeover command on the firewall that issued the error message.

### **Error: fwpulse failed to create temp file**

Fwpulse was unable to create a temporary file used to send an e-mail message to the systems administrator.

**Action:** Check to see if the /tmp file system is full.

### **Error: fwpulse attempt to mail attention notification to *e-mail\_address* failed.**

Fwpulse was unable to send an e-mail notification to the systems administrator. The system() call failed.

**Action:** Check the syslog and the error log.

### **Security Alert: firewall system *name* is not responding to fwpulse. It needs attention.**

Fwpulse is unable to contact its sibling firewall. The specified firewall does not appear to be operational.

**Action:** Check to see if the other firewall is active. See if its other LAN adapters are active. If the firewall is down use the takeover command on the firewall that issued the error message.

## 29.4 Genproxy Messages

All syslog messages issued by genproxy begin with: **date time gateway\_name genroxy[pid]:**  
For the sake of brevity only the message text is listed below.

**Error: times() error.**

The times() function returned an error. Execution continues without cpu timing.

**Action:** none.

**Error: unknown service 'xyz' can not decode as port.**

A request for a service not found in the /etc/services file was made. The connection is broken.

**Action:** Examine the genproxy configuration file to see if the service name is misspelled. If the service name is correct but not in the /etc/services file then it must be added to /etc/services.

**Error: can not get remote host: *system error message*.**

Genproxy was unable to obtain the peer's hostname and IP address. The connection is broken.

**Action:** Examine the system error message to determine the cause of the error.

**Error: getsockname failed: system error message**

Getsockname() returned a negative value. Thus, genproxy was unable to obtain the local IP address. The connection is broken.

**Action:** Examine the system error message to determine the cause of the error.

**Security Alert: attempt to use unsecured port *IPaddress* using secure source IP address *IPaddress* aka *host\_name*.**

A remote host with a secure IP address tried to connect to an unsecured port. This looks like an IP spoofing attempt. The connection is broken.

**Action:** Examine the /etc/firewall/securenets file and the /etc/firewall/secureports file to make certain the two files are not misconfigured. A securenet should not be connected to an unsecured port.

**deny host = *host\_name (IPaddress)***

The source host was denied access. The connection is broken.

**Action:** This could be a mistake or an attempt to gain unauthorized access to services.

**deny host = *host\_name (IPaddress)* service = *service\_name***

The source host was denied access. The connection is broken.

**Action:** This could be a mistake or an attempt to gain unauthorized access to services.

**Error, no 'to' server given**

The destination server was given. The connection is broken.

**Action:** Examine the genproxy configuration file for a command line missing a to server.

**Error: not sure what port to connect to.**

The genproxy could not determine which port to connect to.

**Action:** Examine the genproxy configuration file for a command line missing a port number.

**Error: cannot connect to server *IPaddress port nnn: system error message.***

Genproxy was unable to connect to the server machine.

**Action:** Examine the system error message to determine the cause of the error.

**connect src\_host = hostname (IPaddress) src\_port = nnn des\_host = IPaddress dest\_port = nnn**

Genproxy connect the source host to the server host using the indicated ports.

**Action:** none.

**connection timeout: nnn sec.**

The genproxy connection timed out after nnn seconds due to inactivity. The connection is broken.

**Action:** none.

**Error: while reading from source host.**

Genproxy received a read error trying to read data from the source host. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: write to server returned nnnn expected nnnn.**

Genproxy encountered an error writing to the server. The number of bytes written did not equal the number requested. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: while reading from server host.**

Genproxy received a read error trying to read data from the server host. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: write to source returned nnnn expected nnnn.**

Genproxy encountered an error writing to the source host. The number of bytes written did not

equal the number requested. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: times() error. cpu timing stopped.**

The times() function returned an error. CPU timing for genproxy is stopped. The rest of the statistics are gathered and reported.

**Action:** none.

**disconnect src\_host = hostname (IPaddress) src\_port = nnn, dest\_host = IPaddress dest\_port = nnn  
input = nnnn bytes output = nnnn bytes, duration = hh:mm:ss cpu = sec.**

Genproxy has complete its work and disconnected the source host from the server host. The statistics show the source hostname and port number, the destination hosts IP address and port number, the number of bytes of input and output from the source hosts perspective. The duration of the session and the total cpu time used on the firewall are also recorded.

**Action:** none.

**Invalid IP address 'xyz'. length = nn.**

An invalid IP address has been passed to genproxy. The connection is broken.

**Action:** none.

**open failed for filename: system error message.**

Genproxy was unable to open the securenets file. All addresses will be treated as unsecured.

**Action:** Determine the reason for the open failure. If the file is damaged or missing rebuild it.

**Security Alert: cannot connect to OOBA server ip\_address port number system error message**

Genproxy was unable to connect to the Out Of Band Authentication server on the client host. This condition could occur if the OOBA server is not installed, not running or is listening to a different port than specified by the permit statement in the /etc/firewall/genproxy.conf file.

**Action:** Determine why genproxy could not connect to the OOBA server on the client host.

**Error: unable to write OOBA challenge to IPaddress port number. system error message**

Genproxy was unable to write the OOBA challenge to the Out Of Band Authentication server on the client host. This condition could occur if the connection is broken shortly after the connection is established.

**Action:** Determine why the connection was broken.

**Error: unable to read response to challenge from IPaddress port number system error message**

Genproxy was unable to read the response from the Out Of Band Authentication server on the client host. This condition could occur if the connection is broken shortly after the sending the challenge.

**Action:** Determine why the connection was broken.

## 29.5 RAPROXY Messages

### **Error: times function failed.**

The times function failed. Execution continues but cpu times will not be reported.

**Action:** none.

### **Error: sysconf(\_SC\_CLK\_TIC) function failed**

The sysconf function failed. Execution continues but cpu times will not be reported.

**Action:** none.

### **Error: unknown service 'name' can not be decode as port**

The service name could not be found in the /etc/services file. The connection is broken.

**Action:** none.

### **Can not get remote host**

The peername call failed to get the host anme of the remote client.. The connection is broken.

**Action:** none.

### **Error: getsockname failed.**

The peername call failed to get the hostnanme of the remote client.. The connection is broken.

**Action:** none.

**RealAudio player at src\_host\_nmae (IP address) disconnect from destination\_name (IP address)  
reads = nnnn, writes = nnnn, bytes ead = nnnn, bytes written = nnnn.**

This is the naormal completion message.

**Action:** none.

### **Error: sending version to client 1**

The raproxy was unable to write the version number to the client. The connection is broken.

**Action:** none.

### **Error: sending version to client 2**

The raproxy was unable to write the version number to the client. The connection is broken.



**Action:** none.

**Error: sending version to client 3**

The raproxy was unable to write the version number to the client. The connection is broken.

**Action:** none.

**Error: bad proxy protocol version.**

The raproxy recieved an invalid version number from the client. The connection is broken.

**Action:** none.

**Error: malloc() out of memory.**

The raproxy was unable to acquire memory for buffers. The connection is broken.

**Action:** Try again later.

**Error: reading hostname from client**

The raproxy was unable to read the clients hostname. The connection is broken.

**Action:** Check the version of the RealAudio player

**Error: reading server port from the client**

The raproxy was unable to read the server port number from the client. The connection is broken.

**Action:** Check the version of the RealAudio player

**Error: reading backport offset from client**

The raproxy was unable to read the backport offset from the client. The connection is broken.

**Action:** Check the version of the RealAudio player

**Error: reading connection type from client client**

The raproxy was unable to read the connection type from the client. The connection is broken.

**Action:** Check the version of the RealAudio player

**Error: not enough proxy information**

The raproxy was unable to read the essential proxy information. The connection is broken.

**Action:** Check the version of the RealAudio player

**Error: open failed for 'raproxy.conf'**

The raproxy was unable to open the raproxy configuration file. The connection is broken.

**Action:** Check the /etc/T.Rex directory for the existence of the raproxy.conf file. If the file does not exist create one. If the file exists check the file permissions.

**Error: permit is missing parameters on line nnn of raproxy.conf.**

The permit command does not have any parameters. The connection is broken.

**Action:** Examine the raproxy.conf file and the RealAudio Proxy Administration chapter.

**Error: deny is missing parameters on line nnn of raproxy.conf.**

The deny command does not have any parameters. The connection is broken.

**Action:** Examine the raproxy.conf file and the RealAudio Proxy Administration chapter.

**Permission denied to connect player from src\_host to Destination\_host**

The client was denied access to connect their player to the server because there was no matching permit command. The connection is broken.

**Action:** Examine the permit commands in the raproxy.conf file to see if the client should be allowed access.

**Error: missing 'from' keyword on line nn of raproxy.conf. found 'string'**

An invalid permit or deny command was found. The connection is broken.

**Action:** Examine the sited line of raproxy.conf file and compare it to the specification found in the RealAudio Proxy Administration chapter.

**Error: missing source IP address on line nn of raproxy.conf.**

An invalid permit or deny command was found. The connection is broken.

**Action:** Examine the sited line of raproxy.conf file and compare it to the specification found in the RealAudio Proxy Administration chapter.

**Error: missing 'to' keyword on line nn of raproxy.conf.**

An invalid permit or deny command was found. The connection is broken.

**Action:** Examine the sited line of raproxy.conf file and compare it to the specification found in the RealAudio Proxy Administration chapter.

**Error: missing destination IP parameters on line nn of raproxy.conf.**

An invalid permit or deny command was found. The connection is broken.

**Action:** Examine the sited line of `raproxy.conf` file and compare it to the specification found in the RealAudio Proxy Administration chapter.

## 29.6 RPCproxy Messages

### **Boosting buffer size to nnn.**

RPCproxy has dynamically increased the size of its buffers. .

**Action:** None.

### **Error: malloc (nnn) failed, out of memory.**

RPCproxy was unable to acquire storage for buffers. The connection is broken.

**Action:** None.

### **Error: no sockets to listen on, exiting.**

RPCproxy was unable to open a socket, and has stopped running.

**Action:** Check the system to see if the network adapters are online.

**exit UDP src\_ip = xxx.xxx.xxx.xxx, src\_port = nnn, local\_ip = yyy.yyy.yyy.yyy local\_port =nnn  
dest\_ip = zzz.zzz.zzz.zzz, dest\_port = nnn bytes\_from\_src = nnn bytes\_to\_src = nnn  
duration = nnn sec. .**

This is a summary UDP activity for a given set of address and port numbers.

**Action:** None.

### **Error: failed to open socket (port nnn). system error.**

RPCproxy was unable to open a socket for port number nnnn. The system error message follows.

**Action:** None.

notes:

## 29.7 Sendmail Wrapper Messages

All syslog messages issued by smwrap begin with: **date time gateway\_name smwrap[pid]:**  
For the sake of brevity only the message text is listed below.

### **timeout after nn seconds - exiting.**

The sendmail wrapper timed out because the remote host did not send any data for nn seconds.

**Action:** none.

### **Error: cannot get remote host.**

The call to getpeername failed. The connection is broken.

**Action:** none.

### **Error: cannot chroot to xyz: *system error message*.**

Smwrap could not change its root directory to xyz. Smwrap must change its root directory to the value specified by the spoolpath parameter. The connection is broken.

**Action:** Examine the system error message to determine the cause of the failure.

### **Error: NULL spool directory returned.**

A null entry for the spoolpath was encountered. The connection is broken.

**Action:** Examine the smwrap configuration file and correct the error.

### **Error: cannot set uid to nn: *system error message*.**

The setuid failed. The connection is broken.

**Action:** Examine the smwrap configuration file and correct the error.

### **Error: gethostname failed: *system error message*.**

The call to gethostname failed. Smwrap is unable to determine the name of its own host! The connection is broken.

**Action:** Examine the system error message to determine the cause.

### **EOF from peer: will retry.**

Smwrap received a End Of File from the peer host. Smwrap will wait one second and retry one more time.

**Action:** none.

**peer dropped connection: unexpected EOF.**

Smwrap received an End Of File from its peer host. The connection is broken.

**Action:** none.

**peer dropped connection: *system error message*.**

Smwrap received an error trying to read data from the peer host. The connection is broken.

**Action:** Examine the system error message to determine the cause of the error.

**Error: out of memory.**

Smwrap was unable to acquire storage for additional data structures. The temporary file is unlinked, and the peer connection is broken.

**Action:** none.

**exiting: number of recipients exceeds max (nn).**

The peer host sent more RCPT commands than the maximum value specified in the sendmail wrapper configuration file. The temporary file is unlinked and the peer connection is broken.

**Action:** You may want to determine if this was a legitimate attempt to send a piece of mail to multiple mail boxes or a denial of service attack.

**Error: cannot open temporary file. *system error message*.**

Smwrap could not open a temporary file to hold the mail. The connection is broken.

**Action:** Examine the system error message to determine the cause of the error. If the message implies you are out of disk space and there is plenty of free space on the disk then run **thef** command to see if the file system containing the smwrap spool file is out of space. If it is then you may have to use smit to increase the size of that specific file system.

**exiting too much data > nnnnnn.**

Smwrap received a mail file that is larger than the maximum size specified in the sendmail wrapper configuration file. The file is unlinked. The connection is broken.

**Action:** If this is a legitimate attempt to send a large piece of mail you may want to increase the maximum size.

**host = *hostname (IPaddress)* bytes = *nnnn* from *user* to *to\_address***

Smwrap received the mail from the specified host that is destined for the specified to\_user. The size of the message body is nnnn bytes.

**Action:** none.

**SMTP QUIT with no message. *hostname (IPaddress)*.**

Smwrap receive the SMTP keyword QUIT prior to receiving the message body. The hostname and IPaddress of the host are listed.

**Action:** none.

**EXPN *name* request from *hostname (IPaddress)***

Smwrap received a EXPN command from the peer host. Smwrap always echoes the name back to the peer host. Thus the remote user can't determine if there is a used ID by that name on the system.

**Action:** If you see multiple occurrences of EXPN for different usernames coming from the same host this may be an indication that someone is trying to find a way to breakin.

**VERFY *name* request from *hostname (IPaddress)***

Smwrap received a VRFY command from the peer host. Smwrap always echoes the name back to the peer host. Thus the remote user can't determine if there is a used ID by that name on the system.

**Action:** If you see multiple occurrences of VRFY for different usernames coming from the same host this may be an indication that someone is trying to find a way to breakin.

**Error: create failed for tracefile *filename* system error message.**

Smwrap tried to create a file to hold dynamic trace information, but the request failed. Execution continues without the trace function.

**Action:** Examine the system error message to determine the cause of the error.

**Host name lookup failed for *hostname***

A call to gethostbyname failed using the result of a gethostbyaddr call. In otherwords the DNS could not find the name of the peer host. The peer hostname is treated as unknown and execution continues.

**Action:** You could be talking to a bad guy, or the Domain Name Server could have an error.

**Security Alert: possible spoof, *hostname (IPaddress)* != expected\_name lookup mismatch.**

This is a possible hostname spoof, or an bad DNS entry. The peer hostname is treated as unknown and execution continues.

**Action:** You could be talking to a bad guy, or the Domain Name Server could have an error.

**Security Alert: Connection received using IP options (ignored): *options***

The peer host tried to connect using IP options. Smwrap automatically disables these options and continues.

**Action:** none.

**Error: invalid SMWRAP pointer passed.**

An internal error has been detected. The connection is broken.

**Action:** Call FAS.



**Error: open failed for *file\_name*: *system error message*.**

Smwrap was unable to open the sendmail wrapper configuration file. The connection is broken.

**Action:** Examine the system error message to determine why the open failed. If the file is missing restore it using a backup or recreate it using the instructions found in the Sendmail Wrapper Administration chapter.

**Error: No value specified for userid on line *nn* of *filename*.**

Expected a userid after the keyword userid and found nothing. The connection is broken.

**Action:** Fix the error on line *nn* of the sendmail wrapper configuration file.

**Error: User name '*xyz*' is too long on line *nn* of *file\_name*.**

The user name is longer than 8 characters, which is the maximum user name length allowed by the operating system.

**Action:** Specify a valid user name for the UNIX system.

**Error: user name '*xyz*' on line *nn* of *file\_name* is not a valid user**

The user name that the sendmail wrapper program runs under must be a valid user name defined to the operating system.

**Action:** Fix the error on line *nn* of the sendmail wrapper configuration file by using a valid user id on the system.

**Error: user name '*xyz*' on line *nn* of *filename* is rejected because user id = 0.**

The user name produces the root user id of zero. The smwrap will not run as root for security reasons. The connection is broken.

**Action:** Assign another user id for smwrap to run under.

**Error: No value specified for groupid on line *nn* of *filename*.**

Expected a groupid after the keyword groupid and found nothing. The connection is broken.

**Action:** Assign a valid groupid after the keyword groupid. See chapter 8 for details.

**Error: Group name '*xyz*' is too long on line *nn* of *filename*.**

The group name is longer than 8 characters, which is the maximum group name length allowed by the operating system.

**Action:** Assign a valid groupid after the keyword groupid. See chapter 8 for details.

**Error: group name '*xyz*' on line *nn* of *filename* is not valid group.**

The group name that smwrap runs under must be a valid group name defined to the operating

system. The connection is broken.

**Action:** Fix the error on line *nn* of the smwrap configuration file by using a valid group name defined to the operating system.

**Error: no value specified for maxbytes on line *nn* of *filename*.**

Expected a positive decimal number after the keyword maxbytes and found nothing. The connection is broken.

**Action:** Assign a valid number after the keyword maxbytes or remove the keyword maxbytes if you don't want to specify a limit on mail file size.

**Error: none numeric value 'xyz' specified for maxbytes on line *nn* of *filename*.**

Expected a positive decimal number after the keyword maxbytes and found characters other than 0-9. The connection is broken.

**Action:** Assign a valid number after the keyword maxbytes or remove the keyword maxbytes if you don't want to specify a limit on mail file size.

**Error: maxbytes = 'nn' on line *nn* of *filename*. Unable to deliver mail.**

The value specified for maxbytes is too small. Smwrap will not be able to deliver any mail. The connection is broken.

**Action:** Assign a valid number after the keyword maxbytes.

**Caution: maxbytes = nnnnnnnn on line *nn* of *filename* exceeds 10000000.**

The maximum value for a mail message exceeds the indicated value.

**Action:** none.

**Error: no value specified for maxreceipts on line *nn* of *filename*.**

Expected a positive decimal number after the keyword maxreceipts and found nothing. The connection is broken.

**Action:** Assign a valid number after the keyword maxreceipts.

**Error: none numeric value 'xyz' specified for maxreceipts on line *nn* of *filename*.**

Expected a positive decimal number after the keyword maxreceipts and found characters other than 0-9. The connection is broken.

**Action:** Assign a valid number after the keyword maxreceipts.

**Error: maxreceipts = 'nn' on line *nn* of *filename*. Unable to deliver mail. Exiting.**

The value specified for maxreceipts is too small. Smwrap will not be able to deliver any mail. The connection is broken.

**Action:** Assign a valid number after the keyword maxreceipts.

**Notice: maxreceipts 'nnn' on line *nn* of *filename* exceeds RFC 821 req. of 100. Continuing**

The value specified for maxreceipts exceeds the requirements documented in RFC 821. Program execution continues.

**Action:** none.

**Error: no value specified for timeout on line *nn* of *filename*.**

Expected a positive decimal number after the keyword timeout and found nothing. The connection is broken.

**Action:** Assign a valid timeout value after the keyword timeout.

**Error: none numeric value 'xyz' specified for timeout on line *nn* of *filename*.**

Expected a positive decimal number after the keyword timeout and found characters other than 0-9. The connection is broken.

**Action:** Assign a valid number after the keyword timeout.

**Error: no value specified for wakeup on line *nn* of *filename*.**

Expected a positive decimal number after the keyword wakeup and found nothing. The wakeup value is used by the send mail wrapper daemon smwrapd to determine how frequently it scans the mail directory. The connection is broken.

**Action:** Assign a valid wakeup value after the keyword wakeup.

**Error: none numeric value 'xyz' specified for wakeup on line *nn* of *filename*.**

Expected a positive decimal number after the keyword wakeup and found characters other than 0-9. The connection is broken.

**Action:** Assign a valid number after the keyword wakeup.

**Error: no value specified for spoolpath on line *nn* of *filename*.**

Expected a valid pathname after the keyword spoolpath and found nothing. The connection is broken.

**Action:** Assign a valid path name value after the keyword spoolpath.

**Error: spoolpath name 'xyz..' on line *nn* of *filename* is longer than 200.**

The pathname for the spool file exceeds 200 characters. The connection is broken.

**Action:** Assign a valid path name value after the keyword spoolpath.

**Error: spoolpath name 'xyz' on line *nn* of *filename* is not allowed.**

To enhance security the smwrap program changes its root directory to the value found in spoolpath. This prevents the smwrap program from accessing the rest of the file system on the firewall. As a result, smwrap does not allow the systems administrator to specify spoolpaths such as "/", "/dev", "/etc", "/lib", "/sbin", etc.

**Action:** Select a valid directory such as /home/hermes.

**Error: no value specified for maxchildren on line *nn* of *filename*.**

Expected a positive decimal number after the keyword maxchildren and found nothing. The connection is broken.

**Action:** Assign a valid number after the keyword maxchildren. Values between 5 and 10 are reasonable. A value larger than 12 will produce a warning message.

**Error: non numeric value 'xyz' specified for maxchildren on line *nn* of *filename*.**

Expected a positive decimal number after the keyword maxchildren and found characters other than 0-9. The connection is broken.

**Action:** Assign a valid number after the keyword maxchildren. Values between 5 and 10 are reasonable. A value larger than 12 will produce a warning message.

**Error: maxchildren = 'n' on line *nn* of *filename*. is too small Unable to deliver mail. Exiting.**

The value specified for maxchildren is too small. Smwrap will not be able to deliver any mail. The connection is broken. .

**Action:** Assign a valid number after the keyword maxchildren. Values between 5 and 10 are reasonable. A value larger than 12 will produce a warning message.

**Warning: maxchildren = 'nn' is > 12. See line *nn* of *filename*.**

The maxchildren specifies the maximum number of children that can be spun off to process the entries in the spool directory. Specification of too large a number could reduce system throughput. Execution continues.

**Action:** Reduce the number to 12 to eliminate this warning message.

**Error: no value specified for smwppath on line *nn* of *filename*.**

Expected a valid pathname after the keyword smwppath and found nothing. The connection is broken.

**Action:** Assign a valid path name value after the keyword smwppath.

**Error: smwppath name 'xyz..' is too long on line *nn* of *filename*.**

The pathname for the smwrap file exceeds 255 characters. The connection is broken.

**Action:** Assign a valid path name value after the keyword smwppath.

**Error: smwrapdpath name 'xyz' on line *nn* of *filename* not found.**

The smwrapd program was not found in the directory specified by smwrapdpath. The connection is broken.

**Action:** Make certain the smwrapd program is located in the directory specified by smwrapdpath. The value should contain only the directory name and should not include the program name.

**Error: no value specified for smtpdpath on line *nn* of *filename*.**

Expected a valid pathname after the keyword smtpdpath and found nothing. The connection is broken.

**Action:** Assign a valid path name for the program used to deliver the smtp mail. For example, /usr/lib/sendmail.

**Error: smtpdpath name 'xyz.' is too long on line *nn* of *filename*.**

The pathname for the sendmail program exceeds 255 characters. The connection is broken.

**Action:** Assign a valid path name value after the keyword smtpdpath.

**Error: smwrapdpath name 'xyz' on line *nn* of *filename* not found.**

The sendmail program was not found in the directory specified by smtpdpath. The connection is broken.

**Action:** Make certain the sendmail program is located in the directory specified by smwrapdpath. The value should contain the fully qualified path name of the sendmail program including the program name.

**Error: no value specified for undelivpath on line *nn* of *filename*.**

Expected a valid pathname after the keyword undelivpath and found nothing. The connection is broken.

**Action:** Assign a valid path name for the directory used to store mail that cannot or should not be delivered. This directory should be contained in the spoolpath directory.

**Error: undelivpath name 'xyz.' on line *nn* of *filename* is longer than 200.**

The pathname for the undelivered mail exceeds 200 characters. The connection is broken.

**Action:** Assign a valid path name value after the keyword undelivpath.

**Error: undelivpath name 'xyz' on line *nn* of *filename* not found.**

The pathname for the undelivered mail did not exist. The connection is broken.

**Action:** Assign a valid path name value after the keyword undelivpath.

**Error: invalid record on line *nn* of *filename*, keyword 'xyz' unknown. The connection is broken.**

Smwrap detected an invalid keyword. The connection is broken.

**Action:** Examine the spelling of the unknown command. Refer to chapter 8 for coding of keywords used by the sendmail wrapper.

**Error: no userid record found in *filename*.**

The smwrap program did not find the `userid` command in the smwrap configuration file. The connection is broken.

**Action:** Add the `userid` command as defined in chapter 8.

**Error: no groupid record found in *filename*.**

The smwrap program did not find the `groupid` command in the smwrap configuration file. The connection is broken.

**Action:** Add the `groupid` command as defined in chapter 8.

**Error: no maxbytes record found in *filename*.**

The smwrap program did not find the `maxbytes` command in the smwrap configuration file. The connection is broken.

**Action:** Add the `maxbytes` command as defined in chapter 8.

**Error: no maxreceipts record found in *filename*.**

The smwrap program did not find the `maxreceipts` command in the smwrap configuration file. The connection is broken.

**Action:** Add the `maxreceipts` command as defined in chapter 8.

**Error: no timeout record found in *filename*.**

The smwrap program did not find the `timeout` command in the smwrap configuration file. The connection is broken.

**Action:** Add the `timeout` command as defined in chapter 8.

**Error: no spoolpath record found in *filename*.**

The smwrap program did not find the `spoolpath` command in the smwrap configuration file. The connection is broken.

**Action:** Add the `spoolpath` command as defined in chapter 8.

**Error: no smwppath record found in *filename*.**

The smwrap program did not find the `smwppath` command in the smwrap configuration file. The connection is broken.

**Action:** Add the smwppath command as defined in chapter 8.

**Error: no smtp path record found in *filename*.**

The smwrap program did not find the smtp path command in the smwrap configuration file. The connection is broken.

**Action:** Add the smtp path command as defined in chapter 8.

**Error: no undeliv path record found in *filename*.**

The smwrap program did not find the undeliv path command in the smwrap configuration file. The connection is broken.

**Action:** Add the undeliv path command as defined in chapter 8.

## 29.8 TNPROXY Messages

### **Access denied to *user\_name***

The user tried to access the telnet proxy and the gwuser record does not allow telnet access from the protected network.

**Action:** If you want the user to have access to telnet from the protected network then use the gwuser command to add PRO\_TN privileges.

### **Access denied to *user\_name*. NULL password.**

The user tried to access telnet from the protected network and user authorization was in effect and the user's record contained a NULL password.

**Action:** If the user requires telnet access and authorization is specified in the tnproxy configuration file then you must add a password to the gwuser record.

### **Error: open failed for *filename*: *system error message*.**

The tnproxy was unable to open the securenets file to determine if the source host is on a secured network. The system is treated as unsecured, and the user will have to undergo strong user authentication.

**Action:** Examine the system error message to determine the reason for the error.

### **authenticate user = *user\_name***

The user was authenticated.

**Action:** none.

### **Error: gethostbyname failed for '*host\_name*', *error\_number***

Tnproxy was unable to convert the hostname into an IP address.

**Action:** Lookup the error message number returned by gethostbyname to determine the reason for the failure.

### **Error: open failed for *file\_name*: *system error message***

Tnproxy was unable to open the gwuser file to read the user record.

**Action:** Examine the system error message to determine the cause for the open failure.

### **User *user\_name* not in *file\_name***

Tnproxy did not find the user ID in the gwuser file. The user will be prompted for their password. After receiving the password tnproxy will display a message to the user saying they have entered an invalid login name or password.

**Action:** none.

### **Error: read error searching for user *user\_name* in *file\_name*.**



The system returned a read error while trying to read the users record in the gwuser file.

**Action:** Examine the system error log for error messages regarding the disk drive. Try listing the users record with the gwuser program to see if it is able to read the file.

## 29.9 Webgate Proxy Messages

All syslog messages issued by webgate begin with: **date time gateway\_name webgate[pid]:**  
For the sake of brevity only the message text is listed below.

### **Error: times() error.**

The times() function returned an error. Execution continues without cpu timing.

**Action:** none.

### **Error: unknown service 'xyz' can not decode as port.**

A request for a service not found in the /etc/services file was made. The connection is broken.

**Action:** Examine the webgate configuration file to see if the service name is misspelled. If the service name is correct but not in the /etc/services file then it must be added to /etc/services.

### **Error: fcntl(0,FSETOWN) failed.**

Unable to set trap for out of band signal. Execution continues.

**Action:** none.

### **Error: can not get remote host: *system error message*.**

Webgate was unable to obtain the peer's hostname and IP address. The connection is broken.

**Action:** Examine the system error message to determine the cause of the error.

### **Error: getsockname failed: system error message**

Getsockname() returned a negative value. Thus, webgate was unable to obtain the local IP address. The connection is broken.

**Action:** Examine the system error message to determine the cause of the error.

### **Security Alert: attempt to use unsecured port *IPaddress* using secure source IP address *IPaddress* aka *host\_name*.**

A remote host with a secure IP address tried to connect to an unsecured port. This looks like an IP spoofing attempt. The connection is broken.

**Action:** Examine the /etc/firewall/securenets file and the /etc/firewall/secureports file to make certain the two files are not misconfigured. A securenet should not be connected to an unsecured port.

### **Error: cannot chroot to *path\_name***

The webgate proxy was unable to change its root directory to the path name specified in webgate.conf. The connection is broken.

**Action:** Examine the /etc/firewall/webgate.conf file to see if the specified directory exists.

**Error: NULL gproxpath directory entry returned.**

The pathname returned for genproxpath was NULL. The connection is broken.

**Action:** Correct the genproxpath command in the /etc/firewall/webgate.conf file.

**deny host = *host\_name (IPaddress)***

The source host was denied access. The connection is broken.

**Action:** This could be a mistake or an attempt to gain unauthorized access to services.

**deny host = *host\_name (IPaddress)* service = *service\_name***

The source host was denied access. The connection is broken.

**Action:** This could be a mistake or an attempt to gain unauthorized access to services.

**Error: no 'to' server given**

The destination server was given. The connection is broken.

**Action:** Examine the webgate configuration file for a command line missing a to server.

**Error: unknown service *service\_name***

The destination service is unknown. Execution continues using the source port number.

**Action:** none.

**Error: not sure what port to connect to.**

The webgate could not determine which port to connect to.

**Action:** Examine the webgate configuration file for a command line missing a port number.

**Error: cannot connect to server *IPaddress* port *nnn*: *system error message*.**

Webgate was unable to connect to the server machine.

**Action:** Examine the system error message to determine the cause of the error.

**connect src\_host = *hostname (IPaddress)* src\_port = *nnn* des\_host = *IPaddress* dest\_port = *nnn***

Webgate connects the source host to the server host using the indicated ports.

**Action:** none.

**Error: trace\_init returned NULL, function is not active: *system error message*.**

The webgate trace could not be activated. Execution continues without the trace.

**Action:** none.

**connection timeout: nnn sec.**

The webgate connection timed out after nnn seconds due to inactivity. The connection is broken.

**Action:** none.

**Error: select returned an error: *return\_code*.**

The select statement returned a error. The connection is broken.

**Action:** none.

**Error: while reading from source host.**

Webgate received a read error trying to read data from the source host. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: write to server returned *mmmm* expected *nnnn*.**

Webgate encountered an error writing to the server. The number of bytes written did not equal the number requested. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: while reading from server host.**

Webgate received a read error trying to read data from the server host. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: write to source returned *mmmm* expected *nnnn*.**

Webgate encountered an error writing to the source host. The number of bytes written did not equal the number requested. Statistics are recorded and the connection is broken.

**Action:** none.

**Error: times() error. cpu timing stopped.**

The times() function returned an error. CPU timing for webgate is stopped. The rest of the statistics are gathered and reported.

**Action:** none.

**disconnect** *src\_host = hostname (IPaddress) src\_port = nnn, dest\_host = IPaddress dest\_port = nnn*  
**input = nnnn bytes output = nnnn bytes, duration = hh:mm:ss cpu = sec.**

Webgate has complete its work and disconnected the source host from the server host. The statistics show the source hostname and port number, the destination hosts IP address and port number, the number of bytes of input and output from the source hosts perspective. The duration of the session and the total cpu time used on the firewall are also recorded.

**Action:** none.

**open failed for tracefile *filename* system error message.**

Webgate was unable to open the trace file. Tracing is deactivated. execution continues without a trace. .

**Action:** Determine the reason for the open failure.

**Invalid IP address 'xyz'. length = nn.**

An invalid IP address has been passed to webgate. The connection is broken.

**Action:** none.

**Error: open failed for *filename*: system error message.**

Webgate was unable to open the securenets file. All addresses will be treated as unsecured.

**Action:** Determine the reason for the open failure. If the file is damaged or missing rebuild it.

**Urgent Signal trapped and sent to server.**

Webgate received an urgent signal from the client and passed it to the server.

**Action:** none.

**Security Alert: cannot connect to OOBA server *ip\_address* port *number* system error message**

Webgate was unable to connect to the Out Of Band Authentication server on the client host. This condition could occur if the OOBA server is not installed, not running or is listening to a different port than specified by the permit statement in the /etc/firewall/genproxy.conf file.

**Action:** Determine why genproxy could not connect to the OOBA server on the client host.

**Error: unable to write OOBA challenge to *IPaddress* port *number*. system error message**

Webgate was unable to write the OOBA challenge to the Out Of Band Authentication server on the client host. This condition could occur if the connection is broken shortly after the connection is established.

**Action:** Determine why the connection was broken.

**Error: unable to read response to challenge from *IPaddress* port *number* system error message**

Webgate was unable to read the response from the Out Of Band Authentication server on the client

host. This condition could occur if the connection is broken shortly after the sending the challenge.

**Action:** Determine why the connection was broken.

## 29.10 Gwsh2 Messages

### **3004-007 You entered an invalid name or password**

If the user attempts to login with either an invalid name or invalid password they will get the standard error message from AIX. They should also get this error message if they try to login as the root user, since the root id is restricted to logins from the console.

### **3004-501 Cannot su to "root": account is not accessible.**

If an account has been configured so that another user **CAN NOT SU** to the user then an attempt to su will generate this command. This should be the case for the root user id and possibly others.

## 29.11 Adam utility Messages

### **Error: Invalid leading character 'x'**

The adam utility expects a minus sign '-' to precede all input parameters.

**Return Code:** -1

**Action:** Refer to chapter 8.4 for details on adam usage.

### **Error: No parameters following '-da'.**

### **Error: No parameters following '-dr'**

The adam utility expect either a user name or filename to follow.

**Return Code:** -1

**Action:** Refer to chapter 8.4 for details on adam usage.

### **Error: parameters following '-da' is longer than 200 characters.**

The fully qualified file name should follow -da. It is limited to a maximum length of 200 characters.

**Return Code:** -1

**Action:** Refer to chapter 8.4 for details on adam usage.

### **Error: Aliases name or file name must follow '-da'.**

An aliases name or fully qualified file name should follow -da.

**Return Code:** -1

**Action:** Refer to chapter 8.4 for details on adam usage.

### **Error: parameters following '-dr' is longer than 200 characters.**

The fully qualified file name should follow -da. It is limited to a maximum length of 200 characters.

**Return Code:** -1

**Action:** Refer to chapter 8.4 for details on adam usage.

### **Error: Invalid paramter '-dx'.**

Either a or r must follow -d.

**Return Code:** -1

**Action:** Refer to chapter 8.4 for details on adam usage.

### **Error: no parameter after -p.**

The -p parameter must be followed by "all" or "aliases" or "rev".



**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: missing input parameters.**

The were no input parameters following the program name adam.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: addaliases rc = -1**

The add aliases function failed.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: addraliases rc = -1**

The add reverse aliases function failed.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: bldaliases rc = -1**

The build aliases function failed.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: bldraliases rc = -1**

The build reverse aliases function failed.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: delaliases rc = -1**

The delete aliases function failed.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: delraliases rc = -1**

The delete reverse aliases function failed.

**Return Code:** -1  
**Action:** Refer to chapter 8.4 for details on adam usage.

**Error: open failed for new aliases file.**

The addaliases function was unable to open the new aliases file for input.

**Return Code:** -1

**Action:** Check to see if the file exists. .

**Error: unable to open aliases DB.**

The addaliases function was unable to open the aliases Data Base.

**Return Code:** -1

**Action:** Check to see if the file exists. .

**Error: unable to open aliases DB for add aliases function.**

The addaliases function was unable to open the aliases Data Base.

**Return Code:** -1

**Action:** Check to see if the file exists. .

**Error: inval;id keyname found on line nnn of filename . Length is > 200 characters.**

Aliases names must be less than 200 charcters. Correct the error found on the indicated line.

**Return Code:** -1

**Action:** Check to see if the file exists. .

**Error: NULL secondary name@host.domain found on line nnn of filenmae**

The specified line contained the primary name but the the secondary name (alias). Correct the error found on the indicated line.

**Return Code:** -1

**Action:** Check to see if the file exists.

**Error: invalid aliases name found on line nnn of filename. Length is > 200 characters.**

Aliases names must be less than 200 characters. Correct the error found on the indicated line.

**Return Code:** -1

**Action:** Check to see if the file exists.

notes:

## Chapter 30. Miscellaneous

### **Long response time for telnet to complete a connection between the T.Rex system and an internal system.**

If the users experience an abnormally long delay when trying to telnet to or from the T.Rex firewall system there is a good chance that T.Rex is unable to communicate with the external name server. The long delay is caused by telnet waiting for a response from the external name server.

In order for T.Rex to function correctly it has to communicate with both an internal domain name server and an external domain name server. See chapter 3 step 11 for details on setting up the pointers to the domain name servers.

### **Users are receiving blank mail.**

If your users are receiving blank mail there is a good chance the /var file system has filled up and the sendmail program is unable to write the body of the letter to the spool file. Use the df command to see if the /var file system is 100% full. If so find out why. There are many ways to fill up the /var directory some that you might check for are:

- The syslog has grown very large and needs to be off loaded to tape.
- There are multiple copies of old syslogs that need to be compressed or removed.
- A daemon is running with a debug option on and is filling up the disk. For example, if the named has debug turned on it could create a large file in the /var/tmp directory.

**Increase size of /var:** The size of the /var file system can be increased if more space is required to hold large syslogs.

### **on AIX**

Run the smit utility

```
#smit
```

Make the following selections using smits GUI.

Select Physical and Logical Storage.

Select File System

Select Add/Change/Show/Delete File System

Select Journaled File System

Select Change/Show Characteristics of a Journaled File System

Select the /var file system

Increase the number of 512 byte blocks in the box labeled:  
SIZE of file system (512 byte blocks)

Double check the new value then Select the DO button.



## Chapter 31. Adding Software to T.Rex

In many cases you can use the `aproxy` program to allow connection oriented TCP/IP client server applications to communicate through the firewall. Should there be a need to run a server application on the firewall it can be done. However, one should make certain the application itself is secure, and does not do anything to violate the security and integrity of the firewall system. For example, an server added to the firewall system should run as an unprivileged user in a chrooted directory. This will minimize its ability to compromise the firewalls security.

Applications that need to communicate with external and internal systems must be modified to work, or the firewall will have to be configured as follows. If the application needs to communicate with a limited number of systems behind the firewall then you should add entries to the `/etc/hosts` file. If the number of systems is large then you may want to re-configure the DNS as follows. Change the `/etc/resolv.conf` file to point at the internal DNS. Then configure the internal DNS with a forwarders command pointed at the firewall. Another option is to modify the application to use code to do dual domain lookups in a manner similar to the T.Rex proxies. This modification will use the internal DNS (specified in `/etc/firewall/resolv.inside.conf`) as well as at the external DNS specified by `/etc/resolv.conf`. You will have to modify the following code that is in the public domain. :

```
gethostnamadr.c
res_comp.c
res_debug.c
res_init.c
res_mkquery
res_query.c
res_send.c
resolv.h
```

Simply modify your make file to include these in the build. Instead of using the system libraries for things like `gethostbyname()` it will use these and the new `resolv.ntelnet` to look at the new nameserver. You will also need to add a `-DNRT` to the command line when compiling these.

The source code for these functions and header file can be found on the Internet at <ftp.uu.net>. Simply ftp the following packages.

```
packages/bsd-sources/lib/libc/net/gethostnamadr.c.Z
packages/bsd-sources/lib/libc/net/res_comp.c.Z
packages/bsd-sources/lib/libc/net/res_debug.c.Z
packages/bsd-sources/lib/libc/net/res_init.c.Z
packages/bsd-sources/lib/libc/net/res_mkquery.c.Z
packages/bsd-sources/lib/libc/net/res_send.c.Z
```

You will need to uncompress them (using the UNIX `uncompress` command) before you can use them. Make certain you ftp them in binary mode.

Put these files in the same directory as your client code, and edit them to make sure they use a local copy of `resolv.h` rather than the standard one in `/usr/include`.

notes:

## Chapter 32. Recommended Reading

T.Rex if properly used provides a secure gateway to external unsecured networks. However, a single system such as the T.Rex Firewall System does not eliminate the need for good security policies throughout your network. If you are interested in further understanding of security for UNIX systems then we recommend the following books. There may be many other good books on this topic, but we can recommend these for a start.

Anonymous, Maximum Linux Security: A Hacker's Guide to Protecting Your Linux Server and Workstation, SAMS, 1999, ISBN 0-672-31670-6.

Anonymous, Maximum Security: A Hacker's Guide to Protecting Your Internet Site and Network, SAMS: Second Edition, 1998, ISBN 0-672-31341-3.

Albitz, Paul and Liu Cricket, DNS and BIND, O'Reilly & Associates, Inc., 1993

Christoph Braun, UNIX System Security Essentials, Addison Wesley, 1994, ISBN 0-201-42775-3

Casey Cannon, Scott Trent, Carolyn Jones, Simply AIX 4.3 Second Edition, Prentice Hall, 1999, ISBN 0-13-021344-6.

D. Bret Chapman and Elizabeth D. Zwicky, Building Internet Firewalls, O'Reilly & Associates, Inc. 1995, ISBN 1-56592-124-0

Cheswick, William R. and Steven M. Bellovin, Firewalls and Internet Security: Repelling the Wily Hacker, Addison-Wesley Professional Computing Series, 1994, ISBN 0-201-63357-4

Bryan Costales with Eric Allman & Neil Rickert, sendmail, O'Reilly and Associates, 1993.

David Curry, UNIX System Security, A Guide for Users and System Administrators Addison-Wesley Professional Computing Series, 1992.

James DeRoest, AIX Version 4: System and Administration Guide McGraw-Hill, 1997, ISBN 0-07-036688-8

Rik Farrow, UNIX System Security: How to Protect Your Data and Prevent Intruders Addison-Wesley, 1994, ISBN 0-201-57030-0

Aleen Frisch, Essential System Administration, O'Reilly and Associates, Inc., 1993

Simon Garfinkel and Gene Spafford, Practical UNIX Security, O'Reilly and Associates, 1991.

Arthur E. Hunt, Seymour Bosworth, Douglas B. Hoyt, Computer Security Handbook Third Edition Wiley, 1995, ISBN 0-471-01907-0

Craig Hunt, TCP/IP Network Administration, O'Reilly and Associates, 1992

Ben Laurie & Peter Laurie, Apache: The Definitive Guide Second Edition O'Reilly, 1999, ISBN 1-56592-528-9.

Nemeth, Garth, Snyder, Seebass, Hein, UNIX System Administration Handbook, Prentice-Hall PTR, 1995, ISBN 0-13-151051-7

Uday O. Pabrai, Vijay K. Gurbani, Internet & TCP/IP Network Security: Securing Protocols and Applications, McGraw-Hill, 1996, ISBN 0-07-048215-2

Manuel Alberto Ricart, Apache Server Survival Guide, SAMS.NET, 1996, ISBN 1-57521-175-0

Charlie Scott, Paul Wolfe and Mike Erwin, Virtual Private Networks, O'Reilly, 1998, ISBN 1-56592-319-7

William Stallings, Network and InterNetwork Security, Prentice Hall, 1995

David Stang and Sylvia Moon, Network Security Secrets, IDG Books, 1993

Frank Waters, AIX Performance Tuning, Prentice-Hall PTR, 1996, ISBN 0-13-386707-2

This book explains how to analyze and tune AIX systems. It documents how to interrogate, set and tune all the major AIX performance parameters.

Janice Winsor, Solaris Advanced System Administrator's Guide, Macmillan Technical Publishing, 1998, ISBN 1-57870-039-6



# Appendix A. Online Documentation

This document is available online in a PDF file that can be read using the Adobe Acrobat program. Acrobat allows you to search by topic, keyword and even allows you to write your own notes. The automated installation process loads the PDF file on to the system.

## How to find acrobat

The acrobat reader is available free of charge from Adobe. You can download a free copy using your favorite web browser. Go to **www.adobe.com** and follow their instructions for downloading.

## How to start Acrobat

ON UNIX acrobat is a Motif based application requiring you to be running X Window.

.... more later ...

notes:

# Appendix B. Sample Configuration Files

## Files

Installation of T.Rex requires the creation or modification of the following files on the firewall system. Instructions regarding creation or modification for each file are included in the detailed information for each step.

### Modified System Configuration Files

1. /etc/aliases
2. /etc/inetd.conf
3. /etc/named.boot
4. /etc/named.ca
5. /etc/named.local
6. /etc/passwd
7. /etc/resolv.conf
8. /etc/sendmail.cf
9. /etc/syslog.conf

### AIX Only

10. /etc/rc.net
11. /etc/rc.tcpip
12. /etc/security/login.cfg
13. /usr/lib/security/mkuser.default

### T.Rex Configuration Files

There are nine T.Rex configuration files to be added to the /etc/T.Rex directory.

1. /etc/firewall/aliases
2. /etc/firewall/flexkey
3. /etc/firewall/ftproxy.conf
4. /etc/firewall/genproxy.conf
5. /etc/firewall/resolv.inside.conf
6. /etc/firewall/securenets
7. /etc/firewall/secureports
8. /etc/firewall/smwrap.conf
9. /etc/firewall/tnproxy.conf
10. /etc/firewall/webgate.conf

## T.Rex Program Files

1. /usr/local/etc/adam
2. /usr/local/etc/ftproxy
3. /usr/local/etc/ftprrpt
4. /usr/local/etc/genproxsum
5. /usr/local/etc/genproxy
6. /usr/local/etc/gwsh2
7. /usr/local/etc/gwuser
8. /usr/local/etc/http\_3.0A
9. /usr/local/etc/httpsum
10. /usr/local/etc/md5
11. /usr/local/etc/portscan
12. /usr/local/etc/portsmon
13. /usr/local/etc/restrict.sh
14. /usr/local/etc/rmthost2
15. /usr/local/etc/smrpt
16. /usr/local/etc/smrptx
17. /usr/local/etc/smwrap
18. /usr/local/etc/smwrapd
19. /usr/local/etc/sockd
20. /usr/local/etc/sockdsum
21. /usr/local/etc/tnproxy
22. /usr/local/etc/tnprpt
23. /usr/local/etc/webgate
24. /usr/local/etc/xforward

## /etc/firewall/resolv.inside.conf

The internal DNS is pointed to by the file `/etc/firewall/resolv.inside.conf`. This file looks just like your `/etc/resolv.conf` file, but points to an internal name server rather than an external nameserver. The format is exactly the same, otherwise.

```
# file: /etc/firewall/resolv.inside.conf
# function: T.Rex uses this file to point at the internal name server
# created: by Jim Livermore 6/14/95
#
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995-2000
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# Substitute the IP address of your internal DNS in place of 192.168.24.12
#
domain lsli.com
nameserver 10.10.1.12
```

## /etc/resolv.conf

T.Rex is configured with a caching-only Domain Name Server that can resolve the names of external hosts to IP addresses. T.Rex provides a sample /etc/resolv.conf file to point at the caching-only DNS that runs on the firewall.

```
# file:          /etc/resolv.conf      system: gw
# function:      provides domain name, and points to caching-only Domain Name
#               Server used to resolve external hostnames.
# created:       by Jim Livermore 6/14/95
#
# (C) COPYRIGHT Jim Livermore 1995-2000
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995-2000
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AN AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# Replace goode.com with the domain name of your organization.
# The standard internal IP address (127.0.0.1) is used to specify the
# address of the firewall system, since it is running a caching-only DNS.
#
domain goode.com
nameserver      127.0.0.1
```

## /etc/named.boot file

The /etc/named.boot file tells named to maintain a cache of name server responses. The cache statement in the example, tells named to initialize the cache using the contents of the file /etc/named.ca.

```
; file: /etc/named.boot                host: gw
; This file is set up for a caching-only DNS
;
; (C) COPYRIGHT Freemont Avenue Software, Inc. 1994-2000
;
; NOTICE TO USERS OF SOURCE CODE EXAMPLES
;
; FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
; KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
; WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
; ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
; IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
; (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
; SERVICING, REPAIR OR CORRECTION.

primary 0.0.127.IN-ADDR.ARPA           /etc/named.local
cache .                                /etc/named.ca
```

## /etc/named.ca file

T.Rex ships with a sample named.ca file in the examples directory. A current copy of this file is made available by the InterNIC registration services under anonymous FTP at host ftp.rs.internic.net as file /domain/named.root. The example was current when the manual was printed. The InterNIC registration services periodically updates their source file with new server names and addresses so you will need to download their file on a periodic basis.

```
; file: /etc/named.ca          host: gw
; This is the cache initialization file for the gateway system.
; This file contains the root servers of each domain in the USA.
;
; (C) COPYRIGHT Freemont Avenue Software, Inc. (FAS) 1994-2000
;
; NOTICE TO USERS OF SOURCE CODE EXAMPLES
;
; FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
; KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
; WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
; ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
; IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
; (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
; SERVICING, REPAIR OR CORRECTION.
;
; This file is made available by InterNIC registration services under
; anonymous FTP at ftp.rs.internic.net as file /domain/named.root.
;
;
; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>"
; configuration file of BIND domain name servers).
;
; This file is made available by InterNIC registration services
; under anonymous FTP as
;
; file          /domain/named.root
; on server     FTP.RS.INTERNIC.NET
; -OR- under Gopher at RS.INTERNIC.NET
; under menu    InterNIC Registration Services (NSI)
; submenu       InterNIC Registration Archives
; file          named.root
;
; last update:   Nov 8, 1995
; related version of root zone: 1995110800
;
;
; formerly NS.INTERNIC.NET
;
.           3600000   IN   NS       A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000   A     198.41.0.4
;
.           3600000   NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000   A     128.9.0.107
;
.           3600000   NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000   A     192.33.4.12
;
.           3600000   NS      D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000   A     128.8.10.90
;
.           3600000   NS      E.ROOT-SERVERS.NET.
```

```

E.ROOT-SERVERS.NET.      3600000      A      192.203.230.10
;
.                        3600000      NS      F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A      192.5.5.241
;
.                        3600000      NS      G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A      192.112.36.4
;
.                        3600000      NS      H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A      128.63.2.53
;
.                        3600000      NS      I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A      192.36.148.17
; End of File

```

## /etc/named.local file

Edit the sample file by changing the sample host name "*gw.lsl.com*" to match the host name and domain name of your firewall. The contents of the sample file are listed below.

```

; file:      /etc/named.local      system: gw
; function:   Is used to convert the loop-back address 127.0.0.1 into
;            the name localhost.
; created:    by jlivermore@gw.lsl.com      7/28/94
;
; (C) COPYRIGHT Freemont Avenue Software, Inc. (FAS) 1994-2000
;
; NOTICE TO USERS OF SOURCE CODE EXAMPLES
;
; FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
; KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
; WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
; ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
; IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
; (AND NOT FAS OR ANY AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
; SERVICING, REPAIR OR CORRECTION.
;
; The SOA record identifies gw.lsl.com. as the server for this zone
; with root@gw.lsl.com as the point of contact for this zone.
;
0.0.127.in-addr.arpa. IN SOA gw.lsl.com. root.gw.lsl.com. (
                        1.1      ; serial
                        36000      ; refresh every 10 hours
                        3600      ; retry after 1 hour
                        3600000 ; expire after 1000 hours
                        36000      ; default ttl is 10 hours
                        )
IN NS      127.0.0.1.
1 IN PTR localhost.

```

## Step 13. Turn off unwanted Daemons and activate T.Rex daemons.

## Step 13.1 Overview

It is wise to turn off any daemons you're not using or that are inherently dangerous, such as the "r" commands (rlogin, rsh, rwho), finger, NFS, NIS, tftp and sendmail. For essential functions like e-mail T.Rex provides sendmail wrapper programs that isolate the sendmail program from remote hosts. As a result sendmail does not run as a daemon.

The following tables show the daemons that are Deactivated, and Activated by T.Rex. The tables also show the names of the files used to control the daemons. Since AIX and HP-UX use different files to control their daemons there are separate columns of each platform.

### Deactivated Daemons

The following daemons are deactivated during the installation of T.Rex. The columns labeled AIX File and HP-UX file show the name of the file that is modified to deactivate the specified daemon.

Daemon name	AIX file	HP-UX file
bootpd	inetd.conf	inetd.conf
fingerd	inetd.conf	inetd.conf
ftpd	inetd.conf	inetd.conf
gated	gated.conf	netbsdrc
lpd	rc.tcpip	N/A
netstat	inetd.conf	inetd.conf
portmap	rc.tcpip	netnfsrc
sendmail	rc.tcpip	netbsdrc
timed	rc.tcpip	?
rlogind	inetd.conf	inetd.conf
rlpdemon	N/A	inetd.conf
remshd	inetd.conf	inetd.conf
rexecd	inetd.conf	inetd.conf
remshd	N/A	inetd.conf
routed	rc.tcpip	?
rshd	inetd.conf	N/A
rwhod	rc.tcpip	netbsdrc
snmp	rc.tcpip	?
talkd	inetd.conf	
telnetd	inetd.conf	inetd.conf
tftpd	inetd.conf	inetd.conf
uucpd	inetd.conf	inetd.conf

### NFS Services

name	AIX file	HP-UX file
ypbind	rc.nfs	netnfsrc
ypserv	rc.nfs	netnfsrc
ypupdated	rc.nfs	?
biod	rc.nfs	netnfsrc
nfsd	rc.nfs	netnfsrc
pcnfsd	N/A	netnfsrc
rpc.mountd	rc.nfs	?
rpc.rexd	inetd.conf	netnfsrc
rpc.rstatd	inetd.conf	netnfsrc
rpc.rusersd	inetd.conf	netnfsrc
rpc.walld	inetd.conf	netnfsrc
rpc.rquotad	?	netnfsrc



rpc.sprayd	inetd.conf	netnfsrc
rpc.lockd	rc.nfs	?
rpc.yppasswd	rc.nfs	netnfsrc

## Standard Daemons

The following daemons are not changed or deactivated.

name	AIX file	HP-UX file
inetd	rc.tcpip	netlinksrc
named	rc.tcpip	netbsdsrc
syslogd	rc.tcpip	netlinksrc

## T.Rex Deamons

The following daemons are provided by T.Rex and are activated by the indicated files.

name	AIX file	HP-UX file
ftproxy	inetd.conf	inetd.conf
genproxy	inetd.conf	inetd.conf
nntp	inetd.conf	inetd.conf
T.Rexmon	rc.tcpip	netbsdsrc
smwrap	inetd.conf	inetd.conf
smwrapd	rc.tcpip	netbsdsrc
sockd	inetd.conf	inetd.conf

If you are installing T.Rex on an AIX system go to step 13.2. If you are installing T.Rex on an HP-UX system go to step 13.3.

## Step 13.2 AIX Systems

### */etc/gated.conf*

The file */etc/gated.conf* shipped by IBM has the gateway routing protocols commented out, and they should remain commented out. All lines in the */etc/rc.gated.conf* file should begin with a #.

### */etc/rc.tcpip*

On AIX the */etc/rc.tcpip* file starts the TCP/IP daemons: syslogd, lpd, routed or gated, sendmail, portmap, inetd, named, timed, rwhod, and snmpd. The first daemon started by rc.tcpip is the syslog daemon since the other daemons use this function for error and event recording. The sample file shipped with T.Rex deactivates the following daemons by having their start line commented out: lpd, routed, gated, sendmail,

portmap, timed, rwhod, snmp

The following daemons are activated by rc.tcpip: syslogd, inetd, named, smwrapd, T.Rexmon

```
# file: /etc/rc.tcpip for AIX
# function: The standard rc.tcpip is used to start the TCP/IP daemons:
#           syslogd, lpd, routed, gated, sendmail, portmap, inetd,
#           named, timed, rwhod, and snmpd.
#
#           The T.Rex version deactivates the following functions by
#           commenting out their entries.
#           lpd, routed, gated, sendmail, portmap, timed, rwhod, snmp
#
# created: by Jim Livermore 6/14/95
#
# (C) Jim Livermore 1995-2000
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995-2000
# All Rights Reserved
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AN AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# start -
#       starts daemons using either src or command-line method
# args:
#       $1: pathname of daemon
#       $2: non-null if we should use src to start the daemon
#       $3: any arguments to pass it
#
start()
{
    # just return if the daemon doesn't exist
    #
    [ -x $1 ] || return 0

    # start the daemon using either src or command-line method
    #
    cmd=`basename $1`
    if [ -n "$2" ] ; then
        startsrc -s $cmd -a "$3"
    else
        $1 $3
        echo "\t$cmd"
    fi
}

# check the bootup_option flag in the configuration database
option=`lsattr -E -l inet0 -a bootup_option -F value`
if [ "$option" = "no" ]
then
#=====
#
# Check to see if srcmstr is running; if so, we try to use it;
# otherwise, we start the daemons without src
#
i=3 # make sure init has time to start it
while [ $i != 0 ] ; do
    if [ -n "`ps -e | awk 'NF == "srcmstr" { print $1; exit }'`" ] ; then
        src_running=1 # set flag
    fi
    i=i-1
done
```

```

        break
    fi
    i=`expr $i - 1` # decrement count
done

# If srcmstr is running, ensure that it is active before issuing the
# startsrc commands
#
if [ -n "$src_running" ] ; then
    echo "Checking for srcmstr active...\c"
    i=10 # try ten times to contact it
    while [ $i != 0 ] ; do
        lssrc -s inetd >/dev/null 2>&1 && break # break out on success
        sleep 1 # otherwise wait a second and try again
        echo ".\c"
        i=`expr $i - 1` # decrement count
    done
    if [ $i = 0 ] ; then
        echo "\n\nERROR: srcmstr is not accepting connections.\n"
        exit 1
    fi
    echo "complete"
fi

else
    src_running=""
fi

#####
# Start up the daemons
#####
echo "Starting tcpip daemons:"

# Start up syslog daemon (for error and event logging)
start /etc/syslogd "$src_running"

#####
# The following daemons are NOT started by T.Rex
# print route, gate, sendmail and portmapper daemons.
#start /usr/lpd/lpd "$src_running"
#start /usr/sbin/routed "$src_running" -q
#start /usr/sbin/gated "$src_running"
#
#qpi=30m # 30 minute interval
#
#start /usr/lib/sendmail "$src_running" "-bd -q${qpi}"
# Start up Portmapper
mount | grep ' /usr *nfs' 2>&1 > /dev/null
if [ "$?" -ne 0 ]
then
    REMOTE_USR="N"
    #start /usr/sbin/portmap "$src_running"
fi
#####
# Start up inetd - socket-based daemons
# Start up Domain Name Server as a caching only DNS
#####
start /usr/sbin/inetd "$src_running"
start /usr/sbin/named "$src_running"
#####
# Deactivate: timed, rwhod, snmpd
#####
#start /usr/sbin/timed "$src_running"
#start /usr/sbin/rwhod "$src_running"
#start /usr/sbin/snmpd "$src_running"

#####
# Start up the sendmail wrapper daemon smwrapd
# Start up the T.Rex monitor daemon T.Rexmon

```

```
#####
start /usr/local/etc/smwrapd
start /usr/local/etc/T.Rexmon
```

## **/etc/inetd.conf**

## **/etc/rc.net (for AIX)**

Edit the **/etc/rc.net** file so that it contains the route add commands. The route add commands should be added at the end of Part II of the sample rc.net file. The following example shows the a portion of the **/etc/rc.net** file with the route commands added.

```
#
LOGFILE=/tmp/rc.net.out      # LOGFILE is where all stdout goes.
...
#
# Now we set any static routes.
#
# /usr/sbin/route add 0 gateway                >>$LOGFILE 2>&1
# /usr/sbin/route add 192.168.25.0 gateway    >>$LOGFILE 2>&1
#####
# The following command was added by FAS to specify the default
# route for all Internet traffic.  You should replace the sample
# IP address (198.65.130.22) with the IP address of your
# internet gw or router.  If you have multiple internal
# networks then you will have to provide route add commands
# for each internal gateway.
#
#####
/usr/sbin/route -n add      0 198.65.130.22 5 >>$LOGFILE2>&1
/usr/sbin/route -n add 192.168.25.0 192.168.24.15 5 >>$LOGFILE 2>&1

#####
# Part III - Miscellaneous Commands.
#####
....
```

**Note:** The use of the **-n** argument prevents the system from attempting to print symbolic host and network names in case the local name servers are down.

## **/etc/rc.nfs (for AIX 4.1)**

AIX 4.1 uses the **rc.nfs** file to start NIS, NFS and related daemons. These daemons are not secure and are deactivated by T.Rex. The contents of the sample **rc.nfs** file are listed below.

```
# file: /etc/rc.nfs      for AIX 4.1
# function: The standard rc.nfs is used to start NIS, NFS and related TCP/IP #      daemons:
#
# created: by Jim Livermore 7/14/95
#
```

```

# (C) COPYRIGHT Jim Livermore 1995-2000
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1995-2000
# All Rights Reserved
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AN AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#
# start() has the following logic
#   1) if srcmstr is running, use it to start the daemon
#   2) if srcmstr is NOT running, lookup the daemon path and arguments from
#       srcmstr's config database in odm (SRCsubsys).
#   3) if the config info can not be found, attempt to start the daemon
#       from the default parameters passed into start().
#
start()
{
daemon=$1                                # Subsystem name
default_path=$2                          # full path w/ cmdname
shift;shift
default_arg=$*                           # default arguments
=====
# Get the path to the daemon from the SRC ODM config info
=====
daemon_path=`odmget -q subsysname=$daemon SRCsubsys 2>/dev/null | \
awk ' $1 == "path" { print $NF }' 2>/dev/null | sed 's/"//g' `

=====
# If daemon_path not set (length zero) then try synonymname
=====
if [ -z "$daemon_path" ] ; then
    daemon_path=`odmget -q synonym=$daemon SRCsubsys 2>/dev/null | \
awk ' $1 == "path" { print $NF }' 2>/dev/null | sed 's/"//g' `
fi

=====
# get the arguments to the daemon from the SRC ODM config info
=====
cmdargs=`odmget -q subsysname=$daemon SRCsubsys 2>/dev/null | \
awk ' $1 == "cmdargs" { print $NF }' 2>/dev/null | sed 's/"//g' `

if [ -n "$src_running" -a -n "$daemon_path" ] ; then
    #
    #if srcmstr is running and there is an entry in SRCsubsys - use src
    #
    startsrc -s $daemon
else
    #if srcmstr not running, start manually
    if [ -n "$daemon_path" ] ; then
        if [ -n "$cmdargs" ] ; then
            $daemon_path $cmdargs &                # issue cmd
        else
            $daemon_path $default_arg & # issue cmd
        fi
    else
        $default_path $default_arg &                #issue cmd
    fi
fi

}

#
# determine if srcmstr is running

```

```

#
if [ -n "`ps -e | awk '$NF == "srcmstr" {print $1} ``" ] ; then
    src_running=1
else
    src_running=""
fi

# Check the mount of /. If it is remote, do not start statd,lockd.
REMOTE_ROOT="N"
/usr/sbin/mount | /usr/bin/grep ' / *jfs ' 2>&1 > /dev/null
if [ "$?" != 0 ]
then
    REMOTE_ROOT="Y"
fi

# Check the mount of /usr. If it is remote, do not start statd.
REMOTE_USR="N"
/usr/sbin/mount | /usr/bin/grep ' /usr *jfs ' 2>&1 > /dev/null
if [ "$?" != 0 ]
then
    REMOTE_USR="Y"
fi

# Uncomment the following lines and change the domain
# name to define your domain (domain must be defined
# before starting NIS).
#if [ -x /usr/bin/domainname ]; then
#    /usr/bin/domainname ibm
#fi

#
# Clear all servers' rmtab files in case we went down abnormally.
#
if [ -s /sbin/helpers/nfsmnthelp ]; then
    /sbin/helpers/nfsmnthelp B 0
fi
#####
# Do NOT run NIS on T.Rex. NIS can not be made secure.
#####
#dspmsg cmdnfs.cat -s 8 2 "starting NIS services:\n"
#if [ -x /usr/lib/netsvc/yp/ypserv -a -d /var/yp/`domainname` ]; then
#    start ypserv /usr/lib/netsvc/yp/ypserv
#fi

#if [ -x /usr/lib/netsvc/yp/ypbind ]; then
#    start ypbind /usr/lib/netsvc/yp/ypbind
#fi

#if [ -x /usr/sbin/keyserv ]; then
#    start keyserv /usr/sbin/keyserv
#fi

#if [ -x /usr/lib/netsvc/yp/rpc.yppupdated -a -d /var/yp/`domainname` ]; then
#    start yppupdated /usr/lib/netsvc/yp/rpc.yppupdated
#fi
#####
# Do NOT run NFS clients or servers on T.Rex.
#####
#dspmsg cmdnfs.cat -s 8 1 "starting nfs services:\n"
#if [ -x /usr/sbin/biod ]; then
#    start biod /usr/sbin/biod 8
#fi

#
# If nfs daemon is executable and /etc/exports, become nfs server.
#
#if [ -x /usr/sbin/nfsd -a -f /etc/exports ]; then

```

```

#       > /etc/xtab
#       /usr/sbin/exportfs -a
#       start nfsd /usr/sbin/nfsd 8
#       start rpc.mountd /usr/sbin/rpc.mountd
#fi
#####
# The rpc.statd and rpc.lockd are deactivated by T.Rex.
#####
#
# start up status monitor and locking daemon if present
#
# if [ -x /usr/sbin/rpc.statd -a $REMOTE_ROOT = "N" -a $REMOTE_USR = "N" ]; then
#     start rpc.statd /usr/sbin/rpc.statd
# fi
#
# if [ -x /usr/sbin/rpc.lockd -a $REMOTE_ROOT = "N" ]; then
# start rpc.lockd /usr/sbin/rpc.lockd
# fi

#
# Uncomment the following lines to start up the NIS
# yppasswd daemon.
# DIR=/etc
# if [ -x /usr/lib/netsvc/yp/rpc.yppasswdd -a -f $DIR/passwd ]; then
#     start rpc.yppasswdd /usr/lib/netsvc/yp/rpc.yppasswdd /etc/passwd -m
# fi

```

## **/var/spool/cron/crontabs**

The smwrapd periodically checks for mail spooled by the smwrap program. When smwrapd finds mail to deliver it passes the mail to sendmail for final delivery. If the receiving host is unable to receive mail when sendmail is called for delivery, sendmail places the mail in its own queue. The queued mail will sit there forever unless sendmail is periodically activated by cron to process the queued mail.

T.Rex provides a sample cron table that periodically executes sendmail as a batch program to deliver any queued mail. The sample crontabs file is shown below:

```

# file:          /var/spool/cron/crontabs          system: gw
# function:      Provides commands required for basic system needs
#               including periodic execution of sendmail as a batch program.
# created:       by Jim Livermore 6/14/95
#
# (C) COPYRIGHT Jim Livermore 1995-2000
# (C) COPYRIGHT Freemont Avenue Software, Inc. (FAS) 1995-2000
#
# FAS PROVIDES THE SOURCE CODE EXAMPLES, "AS IS" WITHOUT WARRANTY OF ANY
# KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
# ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOURCE CODE EXAMPLES
# IS WITH YOU. SHOULD ANY PART OF THE SOURCE CODE PROVE DEFECTIVE, YOU
# (AND NOT FAS OR AN AGENT OF FAS) ASSUME THE ENTIRE COST OF ALL NECESSARY
# SERVICING, REPAIR OR CORRECTION.
#

```

```
# added two lines to run sendmail every fifteen minutes, since it no longer
# runs as a daemon
#
#0 3 * * * /etc/skulker
#45 2 * * 0 /usr/lib/spell/compress
#45 23 * * * ulimit 5000; /usr/lib/smdemon.cleau > /dev/null
0 11 * * * /usr/bin/errclear -d S,O 30
0 12 * * * /usr/bin/errclear -d H 90
01 4 * * * /etc/lpp/diagnostics/bin/test_batt 1>/dev/null 2>/dev/null
01 3 * * * /etc/lpp/diagnostics/bin/run_ela 1>/dev/null 2>/dev/null
0,30 * * * * /usr/lib/sendmail -q > /dev/null 2>&1
15,45 * * * * /usr/lib/sendmail -q > /dev/null 2>&1
```

## **/etc/syslog.conf file**

The **/etc/syslog.conf** controls which messages are logged and where they are logged. Each line of the file specifies the type of message to log and the destination where the message will be logged. The type of message consists of two parts separated by a period. The first part specifies the facility that generated the message. The second part specifies priority of the message.

## **Type of message**

The following facilities can be specified in the syslog configuration file: kern, user, mail, daemon, auth, syslog, lpr, news, uucp and \* for all messages. The priorities which can be specified are (from high to low): emerg/panic, alert, crit, err, warn, notice, info, debug. For more information see the IBM InfoExplorer which contains an on-line description of the commands that can be used in the syslog configuration file.

## **Destination**

The destination of the message can be a specified file on a system, or a user id on a system. To send the message to another system follow the file name or the user id by the character string @hostname.

The following example shows the sample syslog.conf file for AIX which will log the following:

```
!      all error messages will be written to the file /var/adm/syslog,
!      all authorization messages regardless of priority will be written to the file
      /var/adm/syslog,
!      all debug messages will be written to /var/adm/syslog.
```

The sample syslog.conf file for HP-UX will use **/var/adm/syslog/syslog.log**. This example will log all types of messages with a priority of debug or higher.

```
# file: /etc/syslog.conf                system: gw
# function:      specify messages to logged by syslogd.
# created:      by Jim Livermore        3/15/94
#
```



```
# (C) COPYRIGHT Jim Livermore 1994-2000
# (C) COPYRIGHT Freemont Avenue Software, Inc. 1994-2000
# All Rights Reserved
#
auth.info      /var/adm/syslog
*.notice       /var/adm/syslog
*.alert /dev/console
*.emerg *
```

If you want to log all system messages on another internal host then simply add the following line to */etc/syslog.conf*, replacing *hostname* with the name of your protected host.

```
*.debug      @hostname
```

### ***/etc/security/login.cfg***

The original */etc/security/login.cfg* file was replaced with a modified version supplied with T.Rex. The new version adds an entry to the "shells=" line for */usr/local/etc/gwsh2* (its near the bottom of the file). This allows "gwsh2" to be used as the legitimate shell for users that you allow to log on to the T.Rex firewall. This entry must be added before you can use *smit* to specify *gwsh2* in the Initial Program field for any user. This field is used to create the complete path name of the gateway shell to be stored in the */etc/passwd* file.

### ***/etc/inetd.conf***

The file */etc/inetd.conf* contains the Internet server configuration data base. Services can be deleted by inserting a # at the beginning of the line. T.Rex replaces the standard */etc/inetd.conf* file with a version that disables dangerous programs, and enables secure proxies supplied with T.Rex. The new version of *inetd.conf* is listed in Appendix B.

### ***/usr/lib/security/mkuser.default* file**

On AIX the */usr/lib/security/mkuser.default* file contains the default values for users created by the *mkuser* command. The */usr/lib/security/mkuser.default* file is automatically updated during the install process so that the first program to be run is the gateway shell */usr/local/etc/gwsh2*. This is required since *telnetd* is enabled and anyone can get a login prompt from either side of the gateway. If you give a user a regular shell, then someone with that password can easily get in from the outside. The */usr/lib/security/mkuser.default* file is automatically updated during the install process.

# Index

ActiveX ..... 1-xvii, 1-xx, 1-1, 1-2, 1-11, 1-12, 8-1-8-5, 8-13, 8-14, 8-20  
adam ..... 3-15, 11-2, 11-4, 11-8, 11-9, 11-13-11-15, 11-17-11-19, 29-46, 29-47, 32-2  
AIX ... 1-iv, 1-xv, 1-xxi, 1-xxii, 1-2, 1-11, 1-15, 1-19, 1-20, 3-1, 3-2, 3-4, 3-15, 3-16, 3-18, 3-19, 3-21-3-23,  
3-25-3-27, 4-3, 4-5, 4-7, 4-13, 6-3, 6-9, 8-2, 8-5, 8-6, 8-22, 8-23, 10-2, 10-3, 10-10, 12-2,  
14-1, 15-1, 15-8, 15-14, 17-1, 17-3, 17-4, 18-1-18-3, 18-7, 19-2, 19-4, 19-5, 19-30, 20-1,  
20-5, 23-3, 23-4, 23-7, 25-1, 26-1, 26-3, 26-4, 29-5, 29-45, 30-1, 32-1, 32-2, 32-1, 32-6-32-8,  
32-10, 32-14, 32-15  
aliases ..... 3-10, 3-11, 3-15, 6-3, 11-1-11-3, 11-5, 11-8-11-15, 11-17-11-20, 12-1, 14-1, 14-2, 21-2, 21-3,  
23-5, 26-3, 27-2, 29-46-29-48, 32-1  
anonymous ftp ..... 1-2, 1-xvi, 1-7, 3-18, 7-1, 7-3, 15-14, 28-2, 32-4  
Apache ..... 1-iv, 1-xvii, 1-xxiv, 1-11, 8-1, 8-6, 8-14-8-16, 8-18, 21-2, 21-4, 32-1, 32-2  
aproxy ..... 1-xvi, 1-xxi, 1-3, 1-6, 1-9, 1-10, 1-16, 6-1-6-4, 6-6-6-10, 29-1-29-7, 31-1  
CD-ROM ..... 1-20, 3-4, 3-6, 3-9, 3-10, 3-20, 23-4-23-6  
CERT ..... 1-18, 3-25, 3-26, 3-28, 3-29, 19-3, 19-9, 28-1  
CIAC ..... 28-1, 28-2  
cookies ..... 1-xvii, 1-xx, 1-11, 1-12, 8-1, 8-4, 8-13, 8-14, 8-20  
CRYPTOCARD ..... 1-iv, 1-xvi, 1-12, 1-13, 1-21, 3-20, 4-4, 4-6-4-8, 5-1-5-4, 23-2, 29-8  
DNS ... 1-xvi, 1-3, 1-8, 1-14, 1-15, 3-1, 3-2, 3-12, 3-17, 3-18, 3-26, 3-28, 3-29, 8-2, 8-10, 8-18, 8-19, 14-3,  
15-2, 15-6, 15-12, 15-13, 16-1, 16-2, 19-29, 25-1, 27-1, 27-4, 27-5, 29-1, 29-13, 29-30, 31-1,  
32-1-32-3, 32-9  
DSS ..... 1-xv, 1-xxii, 1-xxiv, 24-1, 24-2  
Dual DNS ..... 1-14, 3-17, 16-1  
FTP .... 1-2, 1-xvi, 1-xvii, 1-xxi, 1-2, 1-5, 1-7, 1-8, 1-13-1-15, 1-17, 1-19, 1-21, 1-22, 3-1, 3-15, 3-18, 3-21,  
3-27, 3-29, 4-1, 4-3-4-6, 4-8-4-13, 7-1-7-3, 7-6, 8-12, 8-22-8-25, 15-1, 15-7, 15-8, 15-11,  
15-14, 19-1, 19-3, 19-5, 19-9-19-11, 19-21, 19-27, 23-3, 27-3-27-5, 28-2, 29-8, 29-12, 29-13,  
31-1, 32-4  
ftproxy ..... 1-xvi, 1-13, 1-14, 3-12, 3-14, 3-17, 3-21, 4-7, 4-11, 7-1-7-3, 7-5, 16-1, 18-1, 18-4, 19-3, 19-5,  
19-11, 19-21, 21-2, 24-1, 29-8-29-16, 32-1, 32-2, 32-7  
ftprpt ..... 1-17, 19-11, 19-21, 32-2  
fwpulse ..... 1-xxiii, 21-2, 26-1-26-6, 29-17  
genproxsum ..... 19-13, 32-2  
genproxy ... 1-10, 3-11, 3-14, 3-17, 6-2, 6-6, 7-6, 10-2, 13-5, 19-3, 19-4, 19-13, 21-2, 29-17-29-20, 29-43,  
32-1, 32-2, 32-7  
gwgroupp ..... 3-19, 4-1-4-5, 4-10  
gwuser . 1-12, 1-13, 3-15, 3-19, 4-1-4-7, 4-9-4-16, 5-2, 6-7, 6-8, 7-1, 7-2, 8-24, 13-1, 21-2, 21-3, 23-1-23-3,  
29-8, 29-15, 29-16, 29-38, 29-39, 32-2  
hoplite ..... 1-xxii, 1-16, 1-21, 3-19, 4-12, 4-16, 21-1, 21-2  
HP-UX ... 1-iv, 1-xv, 1-xxii, 1-xxiii, 1-2, 1-15, 1-20, 3-1, 3-6, 3-15, 3-16, 3-18, 3-19, 3-23, 3-25, 3-29, 6-3,  
8-2, 8-5, 14-1, 15-1, 15-8, 15-14, 18-2, 18-3, 19-2, 19-30, 23-3, 23-4, 26-4, 32-6, 32-7,  
32-14  
HTML ..... 1-17, 8-3, 8-14, 8-20, 8-22, 8-24, 28-1, 28-2  
httpd .. 1-xvii, 1-xx, 1-xxiii, 1-11, 1-12, 1-15, 1-17, 1-20, 1-21, 3-2, 8-1, 8-2, 8-4-8-6, 8-8, 8-10-8-13, 8-15,  
8-16, 8-18, 8-19, 8-21, 8-23, 8-24, 19-3, 19-24, 19-25, 21-4  
httpd.conf ..... 8-6, 8-18, 8-21, 8-23, 19-25, 21-4  
httpsum ..... 19-24, 19-25, 32-2  
IANA ..... 3-13  
IEEE ..... 8-3  
inetd.conf ..... 3-14, 3-23, 6-2, 9-1, 10-7, 21-3, 23-3, 26-3, 32-1, 32-6, 32-7, 32-10, 32-15  
Internet Assigned Number Authority ..... 3-13

ISS ..... 1-18, 28-2  
 Java ..... 1-xvii, 1-1, 1-2, 1-11, 1-12, 1-21, 3-2, 8-1-8-5, 8-13-8-15, 8-20-8-22, 28-3  
 JavaScript ..... 1-11, 1-12, 8-3, 8-4  
 Linux ..... 1-xv, 1-xvi, 1-xxii, 1-2, 1-11, 1-19, 1-20, 3-1, 3-3, 3-8, 3-16, 3-25, 19-2, 28-2, 28-3, 32-1  
 NCSA ..... 19-8  
 NNTP ..... 1-xxiv, 1-3, 1-7, 1-8, 1-14, 3-1, 6-7, 6-8, 19-13, 27-3, 32-7  
 nsswitch.conf ..... 3-18, 21-3  
 OOBA ..... 1-xxiii, 1-7, 1-10, 4-16, 6-2, 6-5-6-8, 29-20, 29-43  
 Out Of Band Authentication ..... 1-xxiii, 1-6, 1-8, 6-2, 6-6, 6-8, 29-20, 29-43  
 private ... 1-xviii, 1-1, 1-2, 1-8, 3-13, 4-4, 4-6, 4-8, 4-12, 4-15, 4-16, 6-8, 23-1, 23-2, 23-7, 26-2-26-6, 32-2  
 procmon ..... 1-xxi, 18-1, 18-3, 18-7, 19-3  
 ptelnet ..... 1-xxii, 1-21, 3-19, 4-12, 4-16, 19-27, 23-1-23-6  
 raproxy ..... 1-xix, 1-15, 9-1-9-3, 9-5, 21-3, 29-22-29-25  
 resolv.conf ..... 3-12, 3-17, 3-18, 15-12, 15-13, 16-1, 21-3, 31-1, 32-1-32-3  
 resolv.inside.conf ..... 3-12, 3-17, 16-1, 21-3, 31-1, 32-1, 32-2  
 RFC ..... 29-33  
 RPC ..... 1-xix, 1-2, 1-7, 1-8, 1-14, 10-1-10-4, 10-6-10-9, 27-4, 32-6, 32-7, 32-12, 32-13  
 rpcproxy ..... 1-xix, 1-14, 6-9, 10-1-10-3, 10-6, 10-8, 19-3, 21-3, 29-26  
 SecureNet Key ..... 1-iv, 1-12, 1-13, 1-21, 3-20, 4-7, 5-1, 23-2, 29-9  
 securenets . 1-12, 1-13, 3-12, 7-1, 13-1, 15-5, 18-4, 21-3, 29-9, 29-12, 29-18, 29-20, 29-38, 29-40, 29-43,  
 32-1  
 secureports ..... 3-14, 7-1, 21-3, 29-12, 29-18, 29-40, 32-1  
 security .... 1-iv, 1-xv, 1-xviii, 1-xix, 1-xxi, 1-xxii, 1-1-1-7, 1-9-1-13, 1-15, 1-17-1-21, 3-1-3-3, 3-14, 3-20,  
 3-21, 3-27, 3-29, 3-30, 4-1, 4-3-4-8, 4-10, 4-11, 4-16, 5-1, 6-8, 7-1, 7-3, 8-1-8-5, 8-10, 10-2,  
 11-3-11-6, 12-1, 13-1, 14-1, 15-6, 15-14, 18-1, 19-1, 19-3-19-5, 19-13, 19-17-19-20, 19-24,  
 19-27, 19-28, 20-1, 21-1, 23-1, 23-3, 24-1, 25-1, 25-2, 27-4, 28-1-28-3, 29-1, 29-10, 29-12,  
 29-17, 29-18, 29-20, 29-30, 29-31, 29-34, 29-40, 29-43, 31-1, 32-1, 32-2, 32-1, 32-15  
 smrpt ..... 1-16, 11-3, 19-14, 19-15, 32-2  
 smrptx ..... 1-16, 11-3, 11-6, 19-17, 32-2  
 smwrap ..... 1-xviii, 1-xix, 1-8, 1-9, 1-17, 3-21, 3-22, 11-1-11-7, 11-11, 11-13, 12-1, 19-3, 19-15, 19-17-  
 19-20, 21-3, 29-28-29-32, 29-34-29-37, 32-1, 32-2, 32-7, 32-13  
 smwrapd . 1-17, 3-21, 3-22, 11-1-11-3, 11-5-11-7, 18-3, 18-7, 19-3, 19-17-19-20, 29-33-29-35, 32-2, 32-7-  
 32-10, 32-13  
 SNK ..... 4-3-4-9, 4-13, 5-1, 5-2  
 sockd ..... 3-14, 3-15, 8-2, 15-1-15-5, 15-9-15-11, 19-6, 19-7, 19-9, 19-10, 20-2, 20-3, 21-3, 32-2, 32-7  
 sockd.conf ..... 3-14, 15-5, 21-3  
 sockd.route ..... 3-15, 21-3  
 sockdsum ..... 19-6, 19-7, 32-2  
 Solaris .... 1-iv, 1-xv, 1-xxii, 1-2, 1-11, 1-15, 1-19, 1-20, 3-1, 3-2, 3-9, 3-10, 3-15, 3-16, 3-18, 3-19, 3-21,  
 3-23, 3-25, 3-28, 6-3, 6-10, 8-2, 8-5, 8-19, 8-22, 10-3, 12-2, 14-1, 15-1, 15-8, 15-14, 18-2,  
 19-2, 19-4, 19-5, 19-30, 21-3, 23-3, 23-5, 26-1, 26-4, 26-6, 32-2  
 SPARC ..... 1-iv, 1-xv, 1-xxii, 1-2, 1-11, 1-20, 3-9, 8-5, 8-22, 15-8, 23-5  
 spoofmon ..... 1-xxi, 18-1, 18-4, 18-7  
 SQL ..... 1-3, 1-8, 6-8  
 SQL\*NET ..... 1-3, 1-8, 6-8  
 static routing ..... 1-4, 1-7, 3-5  
 SYN Flood ..... 1-xxi, 1-5, 18-1, 18-2, 18-4, 18-5  
 synmon ..... 18-1-18-7, 19-3  
 takeover ..... 1-xxiii, 26-1-26-6, 29-17  
 tftp ..... 1-xix, 1-xxiv, 1-5, 1-7, 10-4, 10-8, 27-3, 32-6  
 tnkey ..... 4-4, 4-6, 4-8, 4-12, 4-15, 4-16, 23-2, 23-3, 23-6  
 tnproxy .... 1-12, 1-13, 1-18, 3-12, 3-14, 3-17, 3-21, 4-4, 4-6-4-8, 4-11, 4-13, 4-16, 13-1-13-4, 16-1, 18-1,

	18-4, 19-3, 19-21, 19-27, 21-3, 23-1, 24-1, 29-38, 32-1, 32-2
tnprpt .....	1-17, 19-21, 32-2
UDP . . .	1-xix, 1-xx, 1-xxiv, 1-2, 1-7, 1-8, 1-14, 6-9, 10-1, 10-2, 10-4, 10-8-10-10, 15-7, 15-8, 19-33, 27-2-27-4, 29-26
URL . . .	1-xvii, 1-xx, 1-1, 1-2, 1-4, 1-8, 1-11, 1-19, 3-29, 8-1, 8-2, 8-5, 8-7, 8-9, 8-11, 8-14, 8-22-8-24, 28-1
Usenet .....	1-3, 27-3, 28-3
webgate . . . .	1-xvii, 1-xviii, 1-xxiii, 1-12, 1-15, 1-17, 3-14, 3-17, 14-1-14-7, 19-3, 19-24, 21-3, 22-1, 29-40-29-43, 32-1, 32-2

T.Rex Open Source Firewall  
Installation and Administration Guide  
First Edition

Reader's  
Comment  
Form

You may use this form to communicate your comments regarding this publication, its organization, or subject matter, with the understanding that Freemont Avenue Software, Inc. may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Send comments to:                      Freemont Avenue Software, Inc.  
                                                 1830 S. Kirkwood Suite 205  
                                                 Houston, Texas 77077  
                                                 Telephone:        (800)-240-5754  
                                                 :                    (281)-759-3274  
                                                 FAX:                (281)-759-8558  
                                                 e-mail:            [trex@gw.opensourcefirewall.com](mailto:trex@gw.opensourcefirewall.com)

notes: