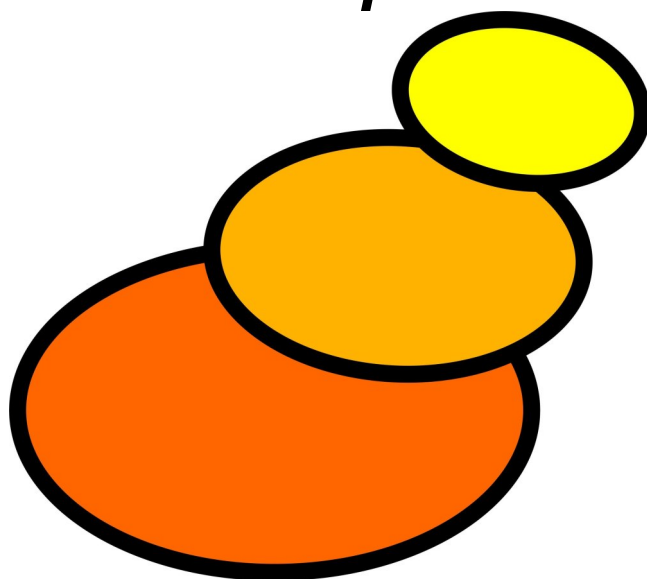


Babel Enterprise v1.0



Babel Enterprise v1.0

First Edition(v1.0) Edition

Published August 30th, 2006

Copyright © 2006 Artica Soluciones Tecnológicas S.L, Sancho Lerena, Esteban Sanchez y otros.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Revision History

Revision 1.0 15 Sep 2006

Submitted.

Revision 0.1 30 Aug 2006

First draft for review.

Table of Contents

Introduction to <i>Babel Enterprise</i>	v
1. <i>Babel Enterprise</i> fundamentals.	1
1.1. General architecture	1
1.2. Babel Agents	2
2. <i>Babel Enterprise</i> installation	3
2.1. Prerequisites	3
2.2. Installing Babel Server	3
2.2.1. Build and link Babel Server	3
2.2.2. Installation Names	4
2.2.3. Configuring your new Babel Server setup	5
2.2.4. Setting up SSH configuration	6
2.3. Installing Babel Console and database.....	6
2.3.1. Initial configuration	7
2.4. Installing Babel Agents	9
2.4.1. Installing Babel Agent for Unix	9
2.4.2. Installing Babel Agent for Windows	11
3. Babel administration	16
3.1. Users and profiles	16
3.1.1. Profiles and user roles.....	16
3.1.2. User management, group and profiles.	17
3.2. Agent management.	17
3.2.1. Setup agent.	18
3.2.2. Module configuration.	18
4. Auditing with Babel	20
4.1. Module types	20
4.1.1. Password audit	20
4.1.2. Banner checking	20
4.1.3. File permission and ownership	20
4.1.4. Networking Kernel parameters.....	21
4.1.5. Remote access	22
4.1.6. Minimization of internet services (Inetd or Xinetd).....	22
4.1.7. Minimization of system services / daemons.....	22
4.1.8. UID0 Users	23
4.1.9. SUID0 Files	23
4.1.10. Security patch (enumerate).....	23
4.1.11. Security patch (local).....	23
4.1.12. Filehash.....	24
4.1.13. Bigfiles.....	25
4.1.14. Installed software.....	25
4.1.15. Open ports	25
4.2. Auditing with <i>Babel Enterprise</i>	26
4.2.1. Your first audit	26
4.2.2. Analyzing your audit	26
4.3. Reviewing your audit data.....	28

A. GNU Free Documentation License.....	31
A.1. 0. PREAMBLE	31
A.2. 1. APPLICABILITY AND DEFINITIONS	31
A.3. 2. VERBATIM COPYING.....	32
A.4. 3. COPYING IN QUANTITY	32
A.5. 4. MODIFICATIONS.....	33
A.6. 5. COMBINING DOCUMENTS.....	34
A.7. 6. COLLECTIONS OF DOCUMENTS	35
A.8. 7. AGGREGATION WITH INDEPENDENT WORKS.....	35
A.9. 8. TRANSLATION	35
A.10. 9. TERMINATION.....	36
A.11. 10. FUTURE REVISIONS OF THIS LICENSE.....	36
A.12. Addendum	36
B. GNU General Public License	38
B.1. Preamble.....	38
B.2. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION.....	39
B.2.1. Section 0	39
B.2.2. Section 1	39
B.2.3. Section 2	39
B.2.4. Section 3	40
B.2.5. Section 4	41
B.2.6. Section 5	41
B.2.7. Section 6	41
B.2.8. Section 7	41
B.2.9. Section 8	42
B.2.10. Section 9	42
B.2.11. Section 10	42
B.2.12. NO WARRANTY Section 11	43
B.2.13. Section 12	43
B.3. How to Apply These Terms to Your New Programs	43

Introduction to *Babel Enterprise*

Babel Enterprise is an audit tool, focused on evaluating the security of the base Operating Systems. *Babel Enterprise* evaluate the security level, or hardening, from a wide range Operating Systems. *Babel Enterprise* run several audit probes and checks to give a *photograph* of the current security status of the system. *Babel Enterprise* also gives a *Security Indicator* of whole system.

Babel Enterprise design is made to manage system security in a big and complex enviroment with many different kind of Operating Systems, versions, technology and configurations, including human teams with different abilities and responsability. *Babel Enterprise* it's a multiuser, distributed management auditing system for the major Operating Systems of the real world. *Babel Enterprise* also allow to be installed which redundant components in all of its components.

Each time you run a new audit policy, you will be able to see and evaluate objectively each important change in your systems: modifications on existing elements, new or erased elements, so that it will know if the security of that system it's going to get better or it's going to get worse and, very important, what it's happening.

Babel Enterprise uses a pragmatic point of view and tries to evaluate mainly those points that represent a security risk and could be improved by intervention of the administrator. *Babel Enterprise* is Free Software, so it's very flexible because has an open API and all internal details to allow advanced users to adapt to any imaginable audit check, without limits.

Babel Enterprise it's a *non-intrusive* tool, so absolutely no change is made in your systems. *Babel Enterprise* only run so many tests as you need and give you the results in a detailed way, including a final numeric Security Indicator.

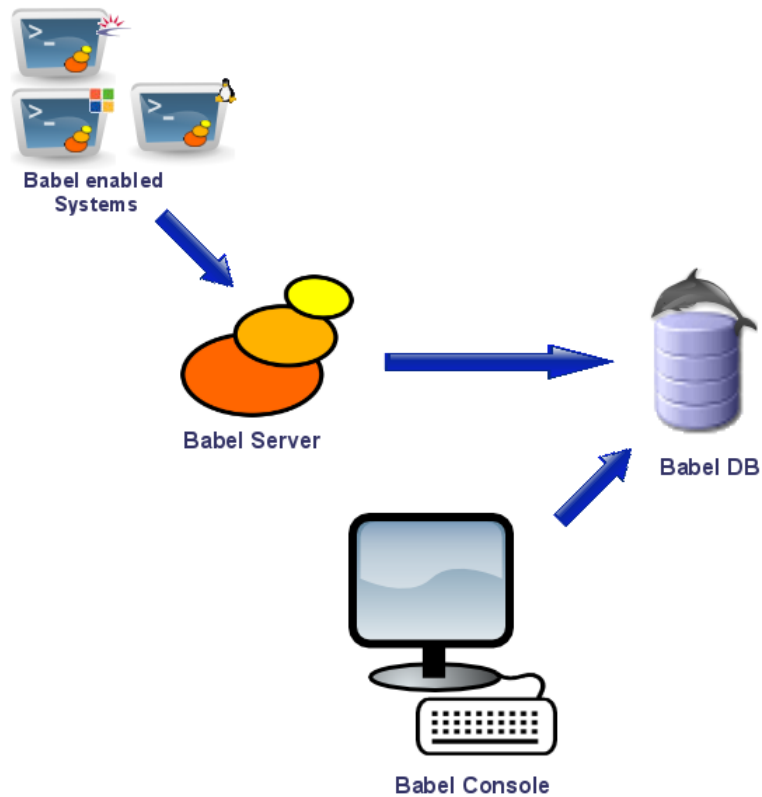
Babel Enterprise agents has versions to run in the last versions of Microsoft™, like Windows 2003, Windows XP, and the more common Unix systems, like Solaris™ 9, AIX™ 5.1, SUSE GNU/Linux 9 ES and Debian/Ubuntu Dapper, but *Babel Enterprise* agents could be very easily adapted for another versions and other similar systems, like BSD systems or HP-UX™).

Babel Enterprise is *Free Software*. It has a public source repository (Subversion). All documentation has been generated using Free Standards, like DocBook SGML, and edited using Free Software (eMacs, OpenJade). WEB page for the project is <http://babel.sourceforge.net>. Also we have a small community of power users in <http://www.openideas.info>.

Chapter 1. *Babel Enterprise* fundamentals.

1.1. General architecture

Babel Enterprise v1.0 has a distributed architecture based on several components:



- *Agent* This is the component who makes the audit and takes the information on the system who is audited. This agent implement the audit using different module for each kind of audit test. Each module could have two kinds of filters, white lists or black lists who helps administrator to refine results of audits, to avoid unnecessary stress on audited hosts and information not really useful for the security audit.
- *Server*. This component process data packets sent by *Babel Enterprise* agents and transform in useful and normalized data, stores in database, ready for the administrator review. Server checks every item to see if has changed from last policy run, and server is who finally scores the security index for this policy run.

Each server is always running as Daemon, ready to process data coming from agents. Server is critical because it's the element that process data, execute alerts and events.

- *WEB Console*, is the graphical user interface. It presents information and allow to view graphical reports and detailed information about policy runs. This enviroment allows also to manage *Babel Enterprise* infraestructure and see last status from each agent.
- *Database*. Stores all Babel information: agent information, user profiles, internal policy events, and of course, audit data. It's based on MySQL, and we could use a MySQL cluster to improve performance and scalability.

1.2. Babel Agents

Babel has been designed with a very modular architecture, very easy to expand and customize. Agents are defined by a basic module, with extension `.bem` (*Babel Executable Module*) who are independent from each other. Each module runs an specific security test (users, permissions, services, etc). Some of them have associated a small database in plain text. This is a white list or black list that allows administrator to customize the behaviour of agent, because helps to "filter" the gathered information by agent. These libraries or small local databases, have the same name that the module, but with `.lst` extension. For example, in the password module, the library contains these words used by "default" in our corporation that, at this moment, we want to check no one is using them.

Each version of *Babel Enterprise* has small differences between modules, for example, GNU/Linux has a module called `xined.bem`, while its equivalent in Solaris™ or AIX™ it's called `inetd.conf`. Internal development of each module has been adapted to native architecture, and Solaris™ modules don't run on AIX™ without adapting them. That is the reason because there are different *Babel Enterprise* Agent versions for each architecture.

Chapter 2. *Babel Enterprise* installation

2.1. Prerequisites

Babel Enterprise is not only a single app, its composed by several shellsript files (Unix Agents), a WEB application in PHP (Console), some code in C++ (Windows Agent), some code in ANSI C (Server) and some structure and data in SQL (Database), so, to get all this running you need to have some pieces of software installed in your system. This is a list of packages, libraries and software you need before install *Babel Enterprise*.

To build *Babel Server* you need to have binary and developer packages for:

- libxml2
- libmysql
- GNU C Extensions (standard in GNU/Linux, not in AIX™ or Solaris™)
- autotools (autoconf, automake, pkg-config, etc...)

In a Ubuntu/Debian GNU/Linux package names are

- libxml2-dev
- libmysqlclient14-dev
- autoconf
- automake1.9
- pkg-config

For running *Babel Console* you need a Web Server (Apache recommended) with PHP 4.3.x (or higher but PHP 5 Not tested yet), PHP4-MySQL and PHP4-session modules, PHP-GD v1.3 library and JpGraph (<http://www.aditus.nu/jpgraph>) for graphic generation.

Babel Agents for Unix requires SSH2 (scp command). Babel Agent for Windows doesn't require anything special for running, in order to build from sources, you will need the latest Dev-Cpp IDE version, with the MinGW tools. Download from <http://www.bloodshed.net/devcpp.html>.

2.2. Installing Babel Server

2.2.1. Build and link Babel Server

The simplest way to compile this package is:

1. 'cd' to the directory containing the package's source code and type './configure' to configure the package for your system. If you're using 'csh' on an old version of System V, you might need to type 'sh ./configure' instead to prevent 'csh' from trying to execute 'configure' itself.

Running 'configure' takes awhile. While running, it prints some messages telling which features it is checking for.

2. Type 'make' to compile the package.
3. Optionally, type 'make check' to run any self-tests that come with the package.
4. Type 'make install' to install the programs and any data files and documentation.
5. You can remove the program binaries and object files from the source code directory by typing 'make clean'. To also remove the files that 'configure' created (so you can compile the package for a different kind of computer), type 'make distclean'. There is also a 'make maintainer-clean' target, but that is intended mainly for the package's developers. If you use it, you may have to get all sorts of other programs in order to regenerate files that came with the distribution.

2.2.2. Installation Names

By default, 'make install' will install the package's files in '/usr/local/bin', '/usr/local/etc', etc. You can specify an installation prefix other than '/usr/local' by giving 'configure' the option '--prefix=PATH'. Make install will install these files by default

```
/usr/local/etc/babel/babel_server.conf  
/usr/local/bin/babel/babelserver
```

And create this directory for incoming data packets from Babel Agents.

```
/var/spool/babel/data_in
```

You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you give 'configure' the option '--exec-prefix=PATH', the package will use PATH as the prefix for installing programs and libraries. Documentation and other data files will still use the regular prefix.

In addition, if you use an unusual directory layout you can give options like '--bindir=PATH' to specify different values for particular kinds of files. Run 'configure --help' for a list of the directories you can set and what kinds of files go in them. If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving 'configure' the option '--program-prefix=PREFIX' or '--program-suffix=SUFFIX'.

There are a daemon launcher called `babelserver_daemon` inside `./contrib` directory, under the main distribution directory. Copy to `/etc/init.d` (or equivalent) to have a startup script ready to launch Babel server.

2.2.3. Configuring your new Babel Server setup

After install Babel Server in, you will need to edit the file `babel_server.conf`, where are defined the variables of the server configuration. File `babel_server.conf` is a text file, you need to edit to setup a few options:

incoming_dir: Define the directory where the server will receive the data sends by the agents. By default this directory is `/var/spool/babel/data_in`. In this directory the user "babel" must have permission to write.

user_name: User with permission in the database. By default the user is "babel". For avoid confusions it's better not to change this user.

password: Password for the database. By default this password is "babel". Would be a *very good idea* change this password for security reasons. Please change this password in MySQL too.

host_name: Name of the machine where is the MySQL Server which contain the database Pandora. By default is localhost.

logfile: Log file where the Pandora daemon writes logs. By default, this file is in `/opt/pandora_server/pandora_server.log`. You must rotate this logs or purge it each week.

verbosity: Level of the log detail in the error messages. 1 is short, 2 is with more detail, 3 debug, 4 more debug, until 10 maximum debug. By default the level is 2. Be sure that your production server runs with a maximum value of 2, because it could generate very BIG files in debug levels and have an important impact on performance.

After modify this configuration values you need to restart Babel server. After run the Babel Server process (daemon), the file `/var/run/babelserver.pid` is created with the process PID.

To automate Babel Server, copy the script `babelserver_daemon` in `/etc/init.d` and establish a link in the appropriate runlevel, like `/etc/rc2.d/S80babelserver_daemon`, for example.

If you modify the installation directory by default you must modify the Babel Server daemon script. It is very important to verify that really exists the file `/var/spool/babel/data_in` and that the user "babel" has permission to write in.

For security, you could use user "babel" to run Babel Server process without privileges.

2.2.4. Setting up SSH configuration

Babel Enterprise, uses SSH protocol to copy XML data packets, generated by the agents, to the server. You need to generate a SSH2 key in every agent, and copy the public key in `$BABEL_HOME/.ssh/authorized_keys`, so you need to create a user called "babel" without privileges. This user will be used by agents to copy data into Babel Server `/var/spool/babel/data_in` directory.

Please BE SURE that user "babel" exists (if not, create with `useradd`), and `$BABEL_HOME/.ssh/authorized_keys` exists and ownership of this file and directory is for babel user, and permissions set to 600.

Please be sure that directory `/var/spool/babel/data_in` exists and babel user is able to write in.

2.3. Installing Babel Console and database

Please look at MySQL install and management guide (<http://dev.mysql.com/doc>) to obtain information about how to create a MySQL database, how to manage mysql users and give him/her privileges to read/write in Pandora database. Remember that you must write the password of the root user in MySQL database to enter mysql command line. This user is not the same of the Operating System. The root password in MySQL is in blank by default (within almost all distributions), you must changed this password with the MySQL command `mysqladmin`. Please be careful with this.

You need a database with name "babel", you could rename it, but you need to reconfigure in server too.

To create the structure of babel database in MySQL Server you have the SQL script `"babel_dbstruct.sql"`. It creates tables and indexes needed to insert information into babel database. You MUST populate database with SQL script `"babel_dbdata.sql"`, it inserts data needed to run Web Console and default user (login: admin, pass: babel) to access babel Web Console. First create a database called "babel", and set an user to be able to access this database:

```
mysql> create database babel;
```

Later, execute the next commands using a user with enough privileges to create tables and indexes into babel Database into your MySQL Server:

```
cat babel_dbstruct.sql | mysql -D babel -u root -p
cat babel_dbdata.sql | mysql -D babel -u root -p
```

Note: if your system is Windows, use the command type instead of cat.

You can also use the source command, if you are connected to MySQL, from the MySQL prompt:

```
mysql> use babel
mysql> source path_to_babel_dbstruct.sql
mysql> source path_to_babel_dbdata.sql
```

This example is valid using root user in MySQL¹

If you have any problem with this commands, from the OS command line you can run this commands:

```
cat babel_dbstruct.sql | mysql -D babel -u root -p
cat babel_dbdata.sql | mysql -D babel -u root -p
```

Note: if you're using Windows, you must use type command instead of the cat one.

Now we will create an user "babel" and will be given to it privileges from the localhost:

```
mysql> grant all on babel.* to 'babel'@'localhost' identified by 'babel';
```

Keep in mind that users need access from babel WEB Console and from babel Server, if your deployment has many subcomponents in different physical machines, you need to setup a MySQL user with privileges to access from different locations.

If you get the error "Warning: mysql_connect() [function.mysql-connect]: client does not support authentication protocol requested by server; consider upgrading" when authenticating Web Console, you have to change the way the password is stored into the database:

```
mysql> set password for 'babel'@'localhost' = old_password('babel');
```

Please note this user will be used by several babel subcomponents(babel Server, babel Web Console) to access database.

2.3.1. Initial configuration

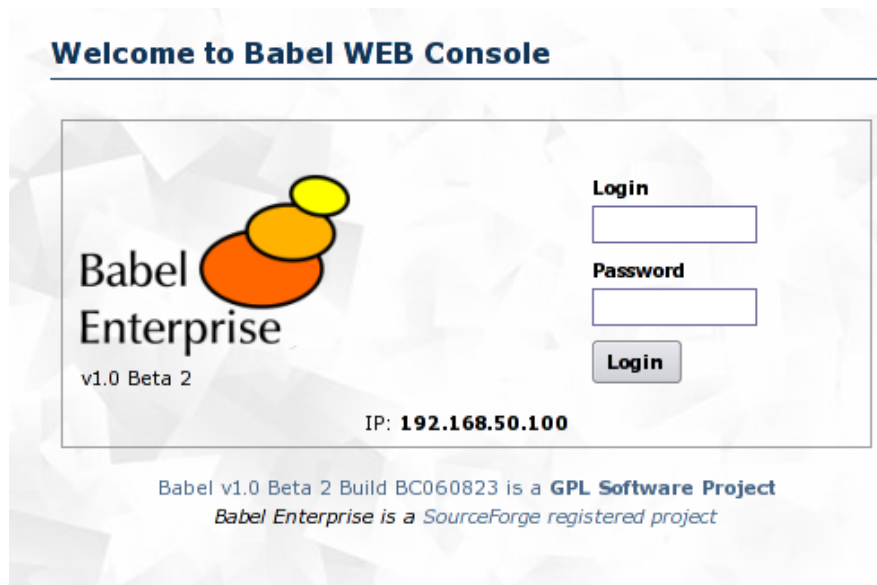
The only file you need to modify is include/config.php, where the following variables are included in .php code:

```
$dbname="babel"; // name of database for babel (default: babel)
$dbuser="babel"; // mysql user to access babel database
$dbpassword="babel"; // Password for mysql user to access babel database
$dbhost="babel"; // Hostname or IP where mySQL server runs
```

If database is defined and was correctly installed, you can now access:

```
http://hostname_babel_webconsole:port/installation_directory/index.php
```

The first time you log there is a default admin user "admin" and password "babel". It's worth to say that *YOU MUST CHANGE CREDENTIALS BEFORE LOGIN FIRST TIME*, change it or create another account, give it administrator privileges, and disable this one.



If you cannot see a screen like this, it's possible that you have problems with PHP installation. When you installed the Web, please check that PHP engine is running. First try to access to the server IP with a browser. You must see the Welcome Apache page.

Remember that after installing the PHP and the PHP module for Apache you must stop and start the Server Apache. As an example, Ubuntu with Apache2:

```
/etc/init.d/apache2 stop
/etc/init.d/apache2 start
```

To verify the PHP and Apache integration you can create the file `test.php` with the following lines:

```
<?PHP
```

```
echo "<h1>TEST</h1>";  
phpinfo();  
?>
```

Now, copy this file in the Apache HTTPDOC directory. This directory depend of the Operating System or Linux Distribution, for example in Ubuntu this directory is `/var/www` and in SUSE is `/srv/www/htdocs`).

To check this integration, please use your browser to open the following URL:

```
http://IP/test.php
```

Where IP is IP Address of your Apache server. If the integration is correct you will see in the browser a text string with big font: “TEST” and a big table with a lot of info about your PHP installation.

2.4. Installing Babel Agents

2.4.1. Installing Babel Agent for Unix

All Babel Unix agents (Suse GNU/Linux, Debian/Ubuntu GNU/Linux, Solaris™ and AIX™) are made with a combination of shellsript code agent and “.BEM modules”, coded also in shellsript.

Only Babel agent for Ubuntu GNU/Linux comes with a “daemon” launcher to put in `/etc/init.d`. All agents comes with the agent script, called `babel.sh`. This is shellsript code, you could edit it.

There are a set of directories too:

- doc - Some documentation
- util - Some binary tools for Babel Agent
- temp - Temporal directory where data packets are writed. After sending to babel server they are automatically deleted. You can change this behaviour editing `babel.sh`.
- modules - This is the most important directory, because it contains module scripts, called .BEM scripts.

Babel Agent home directory often is placed on `/opt/babel_agent` or in `/usr/local/bin/babel_agent`. Put the distribution files under this directory.

2.4.1.1. Configuring Babel Agent for Unix

2.4.1.1.1. Setting up SSH

Babel Agent contact Babel Server using a SSH scp command, so you need to create a SSH key for your user root. You need to run Babel Agent with user root because you need to access your system resources with all privileges (for example to test user password strength).²

After create the ssh key you need to copy to `authorized_keys` file in Babel server filesystem under the babel user (`/home/babel/.ssh/authorized_keys` by default in many systems). Please be sure that SSH auth are running before launching Babel agent.

2.4.1.1.2. Setting up main Babel agent parameters

Babel Agent main config is on `babel_agent.conf` file. If you open it with a text editor like *emacs*, you can see a configure options and text commented with “#” characters in the first column.

- `server_ip`. IP Address or name for Babel Server.
- `server_path`. Remote path in Babel Server where it process data XML files
- `temporal`. Temporal directory where data packets are writed. After sending to babel server they are automatically deleted. You can change this behaviour editing `babel.sh`.
- `interval`. Interval time in minutes. Babel Agents runs each interval minutes, so for example 10080 are seven days (a week). You need to setup in Babel Console too if change this value after the first policy run.
- `agent_name`. By default the agent name is taken from hostname, but if you want to call it with a different name, this is the option.

2.4.1.1.3. Setting up modules

Modules are defined by a BEM (script code) file, called for example `openport.bem` and a library or customization file, with the same name as BEM module but `.lst` extension. All files are text files so you can edit it with text editor like *emacs* or *vi*.

Library files generally defines the "valid" values if they are running with "white list", so this values will be ignored and not reported. For example, in service module, the library file contains service names that are considered as valid.

There are other kind of formats for library files, but all are very easy to understand. Refer to module documentation on how to custom and configure it.

To disable a module, simple move to a new directory called "disabled" or delete the `.bem` file.

2.4.1.2. Running Babel Unix agents

For Solaris™ and AIX™ there is no startup script. By default agent could be in `/usr/local/bin/babel_agent` directory, to execute it, type:

```
/usr/local/bin/babel_agent/babel.sh /usr/local/bin/babel_agent
```

So, this execute `babel_agent` and read config in `/usr/local/bin/babel_agent` directory. Check logfile at `/usr/local/bin/babel_agent/log/babel_agent.log` and `/usr/local/bin/babel_agent/log/babel_agent_error.log`.

When you run, you must see a banner like:

```
Babel Agent 1.0 Build 060404 (c) Sancho Lerena, and others 2004-2006
This program is licensed under GPL2 Terms. http://babel.sourceforge.net
Running in daeva at 2006/08/23 18:45:12
```

Babel Agent process all BEM modules and generate a XML in `temp` directory. When ends, it will copy using SSH to server, please check that SSH is working before worry about other possible problems.

For Ubuntu systems, a start-up script called `babel_agent` are launching `babel.sh` for you, first check code for be sure that `BABEL_PATH` variable is correct. You can use this startup script for Solaris™ and other GNU/Linux agents.

2.4.2. Installing Babel Agent for Windows

2.4.2.1. Build Windows Agent from sources

In order to build from sources, you will need the latest Dev-Cpp IDE version, with the MinGW tools. Download from <http://www.bloodshed.net/devcpp.html>

Open `BabelService.dev` with Dev-Cpp and construct the project. Everything should compile fine in a default installation.

2.4.2.2. Windows Agent installation

Before running or installation of Babel Windows service, you must create the configuration directory and extract the `BabelBin.zip` file into it. It doesn't matter where it is installed, because Babel Agent will adapt to any local directory. In the examples, the application will be installed in `C:\Babel\`

This directory will hold the configuration files, which are:

```
c:\Babel\babel_agent.conf    :: Babel Windows service main configuration (see above)
c:\Babel\filehash.lst       :: Files that the filehash module will hash
c:\Babel\registry.lst       :: Registry keys and values that the registry module will ch
c:\Babel\id_dsa              :: Private key to access the Babel server using SSH
c:\Babel\id_dsa.pub          :: Public key to access the Babel server using SSH
```

Notice: At this moment, the installation of the Babel Windows Agent must be done manually. We are working in a auto-install package.

To install the Babel Windows Agent execute this sentence in a Windows command line:

```
BabelService.exe --install
```

The Agent will be installed into the Windows services system. You can check it on Control Panel -> Administrative tools -> Services.

To run the Agent open the "Services" dialog (Control Panel -> Administrative tools-> Services), search the "Babel Service" service and run it clicking the play button. To stop the service, open the "Services" dialog, search the "Babel Service" and click the stop button.

To uninstall the Babel Windows Agent, execute this sentence in a Windows command line:

```
BabelService.exe --uninstall
```

2.4.2.3. Windows Agent testing

You can check the Pandora Windows Agent output in the C:\babel\babel-debug.dbg file, that is a plain text file and includes info about the execution flow of the Agent.

To test that SSH is working correctly, you can use the --test-ssh parameter in the executable file. This force babel to conect using internal SSH and copy a file called "ssh.test".

2.4.2.4. Windows Agent configuration

All setup is made in `babel_agent.conf`. This file is a list of keys/values pairs. Here is an example of this file.

```
## Begin of babel_agent.conf example
# The comments begin with the '#' character
# IP of the Babel server
server_ip      192.168.50.1

# Remote path to copy the data
server_path    /opt/babel/data_in/

# Local path to the temporal directory, please put " " between blank spaces
temporal      "C:\Documents and Settings\temp\"

# Interval between executions (in minutes)
interval      7000

# Name of the agent
agent_name    antiriad

# Modules activation
filehash enabled
patch  enabled
services enabled
openport enabled
registry enabled
registry_kernel enabled
password disabled
software enabled
## End of babel_agent.conf example
```

You need to generate OpenSSH key files and named it as `id_dsa`, `id_dsa.pub`. These files must be generated using SSH utils and the Babel server must be configured to allow accessing the babel user ("babel" by default) using this pair of public/private keys.

Most of setup files for Windows Agent are the same for Unix Agents, so configuration for this modules is not described here.

2.4.2.4.1. filehash.lst

Configure the filehash module with this file, including each file to hash in a line of this file. You can put a single file or a directory that will be hashed recursevely. It's possible to exclude a file or a disable the recursivity of a dir. For example

```
# Begin of filehash.lst example
# The comments begin with the '#' character
# Recursive directory
```

```

C:\Windows
# Excluded file
!C:\Windows\TASKMAN.EXE
# Excluded directory
!C:\Windows\System32
# Non-recursive directories
@C:\Windows\Help
@C:\Document and Settings\usuario\My Documents
# End of filehash.lst example

```

2.4.2.4.2. *software.lst*

Contains a list of "authorized" software packages. Any software package that doesn't be present in this list will be reported.

2.4.2.4.3. *registry.lst*

This file is used to configure the registry module. Simply include each registry and it's expected value in a line of this file. If the key or the value has any blank space, put it between a pair of ". Example:

```

# Begin of registry.lst example
# The comments begin with the '#' character
# Recursive directory
# No spaces, no need to put "
HKEY_CURRENT_USER\Software\GTK\2.0\asdf 124
# Spaces in the value Put "!!
HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0\Identifier "86 Famil
2"
# Spaces in the key and the value. Put "!!
"HKEY_LOCAL_MACHINE\SOFTWARE\BabelService\CmdLineParamCount" "afsdfer sdfda dsg"
# End of registry.lst example

```

2.4.2.4.4. *registry_kernel.lst*

Very similar to *registry.lst*. It contains a list of special items from registry and its respective values. Any value different from this OR not existant will be reported.

Notes

1. Remember if you're in Windows use the double slash ("/") with the path to the files, not the backslash ("\").
2. It's possible to run Babel Agent without root privileges, but it's very limited

Chapter 3. Babel administration

Babel administration is doing from WEB console, this application is made with PHP and doesn't require additional software installation. There is no need to install Flash, Java or ActiveX. Web console can be accessed from any standard platform who supports HTML and CSS, like Mozilla, Firefox, Konqueror or IE5+.

Web console could be installed in multiple servers, so we could have many consoles as we need, for load balancing purposes or because logistic / distribution problems in our network (different user groups, geographical issues, different administrative roles, etc). *Babel Enterprise* web console only needs to connect Babel MySQL server database.

Babel administration is divided in several main groups:

- Babel users and profiles
- Agents
- Server setup
- Database maintenance

3.1. Users and profiles

Babel allow to define different users for common daily operation. For each user you can assign a unlimited combination of a profile/groups, profiles defines the different "rights" over Babel.

3.1.1. Profiles and user roles

Management features in Babel allow to define different custom roles, specific for each user of Babel, and on this way, to create a hierarchy of users sorted by their profiles. You can define operator with read only access, or simple database operators only with access to purge database, and of course, each profile is applied only in a group of agents if you want.

To create a profile, go to "Profile management" > "Create profile", inside administration menu.

In this screen are all profiles defined. By default there are five profiles created:

- Operator (Read)
- Operator (Write)
- Chief Operator


- Group coordinator
- Babel Administrator

You can assign to a profile any of the following rights:

- View incidents. For future versions of Babel, could integrate profiles with a problem/incident management system, used to manage system upgrades and hardening. This right allow to read incidents.
- Edit incidents. Also for future versions of Babel. This rights allow to create and modify your own incidents.
- Manage incidents. Also for future versions. Allows a senior operator or operation group to create, delete or edit any incident in this group.
- View agents. Allow to read information reported by agents.
- Edit agents. Allow to modify agent configuration, like module descriptions, S.O type, interval and other parameters.
- Edit alerts. For next version, Babel allow to define alerts to automatically notify users when policy does not complaint.
- Manage users. Allow to manage users, group and profiles.
- Manage Database. Database management.
- Manage alerts. Allow to add or modify system commands used by alerts and ther parameters.
- Babel Management. Allow to manage general parameters of *Babel Enterprise* setup.

3.1.2. User management, group and profiles.

A given group contains agents. Group also may contain another groups. An user is associated with a profile and a group, in this way, each user could have different rights over different groups.

To delete a profile/group assignement, you can use the same button used to delete users, agents and many other objects in the *Babel Enterprise* WEB console. This icon is .

3.2. Agent management.

3.2.1. Setup agent.

Babel Enterprise autocreates an agent definition in the WEB console when it receives the first data packet from the agent, but it's very recommended that you review agent configuration in the WEB console, to adjust interval (in minutes), OS type and assign to a group. All of these parameters could be established in the main agent setup screen. This screen is like that:

Agent configuration

Update agent

Agent name: antirad

IP Address:

Group: Windows Server

Interval: 10080

OS: Windows

Description: Autocreated agent

Module definition: Learning mode ☒ Normal mode ☐

Disabled: Disabled ☒ Active ☐

Update

Also, after agent general setup, you could associate data modules to agent. When the first data packet arrives to *Babel Enterprise* server, they also are created and aggregated, so you don't need to create and associate, but could be interesting to modify module description in some cases to be more explicit.

3.2.2. Module configuration.

Babel Enterprise assigns a numeric qualification, a Security Risk Indicator, of security (hardening) in that operating system. It assigns different values for each "bad" value in each module. You could adjust the quantity for each module, modifying default values for each module in the module weight editor.

Defined module types

Type name	Weight	Type	Delete
crontab	0	All values	<input checked="" type="checkbox"/>
devices	0	All values	<input checked="" type="checkbox"/>
filehash	0	Bad values only	<input checked="" type="checkbox"/>
fileperm	1	Bad values only	<input checked="" type="checkbox"/>
inetd	1	Bad values only	<input checked="" type="checkbox"/>

In this editor, we can change weight for that modules who are of type "only bad values". There are two types of modules, modules who only "enumerate" data items, as for example module "Software". This module doesn't set if a software package is "bad" or "good", it only reports "there is a package called XXX installed in this system", so this kind of modules don't assign a numeric value for the global risk indicator for that system, it's impossible to be objective if you say that a whole system is worst than other because it has more software packages installed, so this kind of modules, called "All values" module, only enumerate items, doesn't qualify them. This kind of module is not used to generate the global *security index* for that system.

The other type of module, described as "Bad values only", has a value associated. For each element found for that type of module, the module value will be added to the total count of Security Indicator for this agent. For example, if you have module "kernel" with a value of three, and you have found four elements that are bad, total value added for this module in the global Security Indicator will be $3 \times 4 = 12$. Sum for all other modules will result the total Security Indicator for that module, so if you have a big Security Indicator, you have a insecure system.

This kind of customization allows to ponderate some modules more than others, because, for example, is more critical to have weak passwords than to have a few bad kernel parameters.

Chapter 4. Auditing with Babel

4.1. Module types

Babel Enterprise agents are able to audit any security aspect of your system, as if a security patch is installed, or if there is an user with a weak password. When something doesn't comply with a specific security policy module, module weight its used to sum to a global security indicator. Babel helps you to qualify along time your systems security.

For *Babel Enterprise* v1.0 there are a lot of security module implemented, and they are explained in this section. Not all modules are available for all platforms.

4.1.1. Password audit.

This module checks that users passwords will be secure, that's it, that in a quick brute force test, cannot be broken. This module also makes a dictionary attack test and again, checks that any user has a "typical" password. This checks also include a joey attack test (same password as user name). Dictionary can be edited by administrator for adding the more common passwords used in your organization.

This module checks only valid users, so these users that had been disabled dont be checked. So this module try to discover vulnerable accounts, if any account is discovered by this module, it will need to change immediatly.

4.1.1.1. Setting up password auditing

To customize, simple edit `password.lst` and add those "typical" password you have in your organization, they we'll be used in the next policy run.

4.1.2. Banner checking

This module checks existence of banners for remote services, like SSH, Telnet and FTP. Could be easy customized to check a specific content in this banner (your organization name or some kind of content...). It's very recommended to alter default banner of remote services, and include some information about access policies for this systems.

4.1.3. File permission and ownership

System file permissions, user and group ownership could be checked, using a customized list, you could monitor any change on them. This list could be customized to control any file of the system.

4.1.3.1. Setting up file permissions

Use your favorite text editor, like eMac for editing `fileperm.lst`. This file has a familiar format:

```
-rw-r----- /etc/shadow root shadow
-rw-r--r-- /etc/passwd root root
-rw-r--r-- /etc/hosts root root
```

In the first column, a common Unix syntax representing permissions are shown. Simple use it to compare with files in host filesystem. If host file has the SAME permissions, dont give any alert.

In the second column, you have the file used for comparation. In the third and fourth, you have user owner and group owner. The same rule is applied: if host filesystem has a different owner, an alert will be reported to Babel Server.

4.1.4. Networking Kernel parameters

This parameters, related with security and stability of system, also permits machine to ressis Denial of Service attacks. This module checks things like ARP refresh tables, ICMP redirects, ICMP Broadcast response, ICMP unreachable ignore, TCP buffer sizes and more.

4.1.4.1. Setting up kernel settings

To customize, simple edit `kernel.lst`. This file has a slightly different format for each Unix system, for example in AIX has this format:

```
somaxconn:4096
ipsendredirects:0
ipforwarding:0
tcp_keepidle:900
```

While in Linux has this format:

```
/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts:1
/proc/sys/net/ipv4/ip_forward:0
/proc/sys/net/ipv4/ipfrag_time:30
/proc/sys/net/ipv4/tcp_keepalive_intvl:35
```

Whatever Unix agent you need to setup. First parameter, separated with a “:” character, its the kernel setting, and the second parameter is the value.

4.1.5. Remote access

This module checks existence and setup of typical remote access, like SSH, Rlogin, FTP, Telnet, automatic services based on `.netrc`.

SSH: Checks that root user cannot make remote connections. In some kind of enviroments, root access are limited using PAM subsystem instead of generic configuration from OpenSSH. It look for existence of public key used for automatic authentication, because in some kind of policies is not permitted, because the risk of that practice. Also makes a port forwarding audit.

Telnet: Checks that root cannot enter

Rlogin: Checks existence and contents of `.rhosts` files in each home user directory and the existence of `hosts.equiv`. Administrator needs to check correct configuration of this files and validate (using a whilelist) if they need to be reconfigured or deleted.

.netrc resources, look for all command scripts based on `.netrc` standard and show to administrator to take an action with them.

FTP: Checks that root user cannot login with root user. Also it's verified that anonymous login are disabled.

4.1.6. Minimization of internet services (Inetd or Xinetd)

Using a while list of permitted services, *Babel Enterprise* checks existence of programmed services in the startup who be unnecessary for the system or has not been authorized. Minimization of services is key for securing a system.

4.1.6.1. Setting up Internet services

Simply edit `inetd.lst` or `xinetd.lst` and add your "valid" services there.

4.1.7. Minimization of system services / daemons

As with internet services, *Babel Enterprise* checks for services who are currently configured for running in system startup. It looks for services in default system V levels, inittab file or any other places your system could start different services.

4.1.7.1. Setting up system services/daemons

Very similar to internet services: Simply edit `services.lst` and add your "valid" services there. They will be searched in default System V directory structure, `/etc/inittab` and other places. In Windows agents, Service name is used, not description.

4.1.8. UID0 Users

All active users are checked, looking for all UID0 users different from root user.

4.1.9. SUID0 Files

This module makes a find for all filesystem looking for suspect SUID0 files, and are compared with a whitelist of identified "valid" SUID0 files. Any other file will be reported.

4.1.9.1. Setting up SUID0 files

There is a list of "valid" SUID0 files in `suid0.lst`. Enter there each suid0 file you know is good.

4.1.10. Security patch (enumerate)

Keep a centralized list of each patch installed in all systems managed with Babel, so you can quickly find who systems have (or not) installed new critical patches.

4.1.11. Security patch (local)

Allow to examine patch level of your machines and determine if a critical patch is not installed on them. This requires to maintain local libraries of critical patches on each system.

4.1.11.1. Setting up Patch

Instead gather all patches available on host and manage in a central repository, you may prefer to include a list of "needed" patch in each local Babel agent. For this purpose you have a "blacklist" of patches. If they are not found in system, it will be reported. Edit `patch.lst`. For each platform there are changes in format, for example in AIX:

```
IY48657:5.2:Sendmail remote buffer overflow
IY68464:5.2:Local root exploits in perl 5.8.x
IY70027:5.2:Attacks against TCP via ICMP
```

The first column is the patch ID, second the AIX major version and third is description for the patch.

For Solaris, is very similar:

```
108899-04|SunOS 5.8: /usr/bin/ftp patch
108919-27|CDE 1.4: dtlogin patch
```

But patches are applied for a specific version or application as shown in example.

4.1.12. Filehash

This is possibly, one of the audit modules more interesting. It keeps an inventory of MD5 hashes for each file you choose for the system audited. Its very useful to keep a centralized inventory of MD5 hashes of specific files on your systems, and be sure that are not modified. There are many similar methods, but with *Babel Enterprise* you could have this inventory centralized, avoiding local modifications of this inventory. You also could compare contents of different Operating System files with same HASH.

4.1.12.1. Setting up Filehash

Edit `filehash.lst` and add there the directories you want to be included in this md5 hash inventory. If this directory contains another directories, they also be included in search. Be careful, include here important files, not EVERY file of your system, or you'll be wasting CPU from host machine, and database storage in Babel.

Format for this file is very simply, include in each line a file or directory to be included. because Babel makes a "find" command under this directories. This is an example:

```
/etc/security/
/etc/ssh/
/etc/ssl/certs/
/usr/bin/paswd
/etc/passwd
```

```
/etc/shadow
```

4.1.13. Bigfiles

Babel checks existence in specific directories for files with a size bigger than 10MB (this limit could be customized). Any file above this limit in that directories, will be reported.

4.1.13.1. Setting up bigfiles

Edit `bigfiles.lst` and add there the directories you want to search. Be careful, use ONLY local filesystems, because Babel makes a "find" command under this directories.

You also could modify default size search value. By default it's configured to use 10MegaBytes. In order to change it, you need to edit `bigfile.bem` and change first variable in code.

4.1.14. Installed software

Babel Enterprise details all software packages installed on your system (this could be very exhaustive in UNIX systems), including installed version when this information be available. This information is very useful to make a software inventory oriented to security. This information, plus patch level of each security system componente, could be very valuable. In Unix systems we also can know library versions, so the total available information permits to draw a extensive map of software security in our system.

4.1.15. Open ports

Open port module report all ports (TCP or UDP) who are listening in the system, it also gives name of the proccess who open the port, and the user owner of this proccess (if this information is available). It can be customized using a whitelist of valid ports ignored.

4.1.15.1. Setting up OpenPorts

`openport.lst` has a format like

```
80/TCP
22/TCP
21/TCP
```

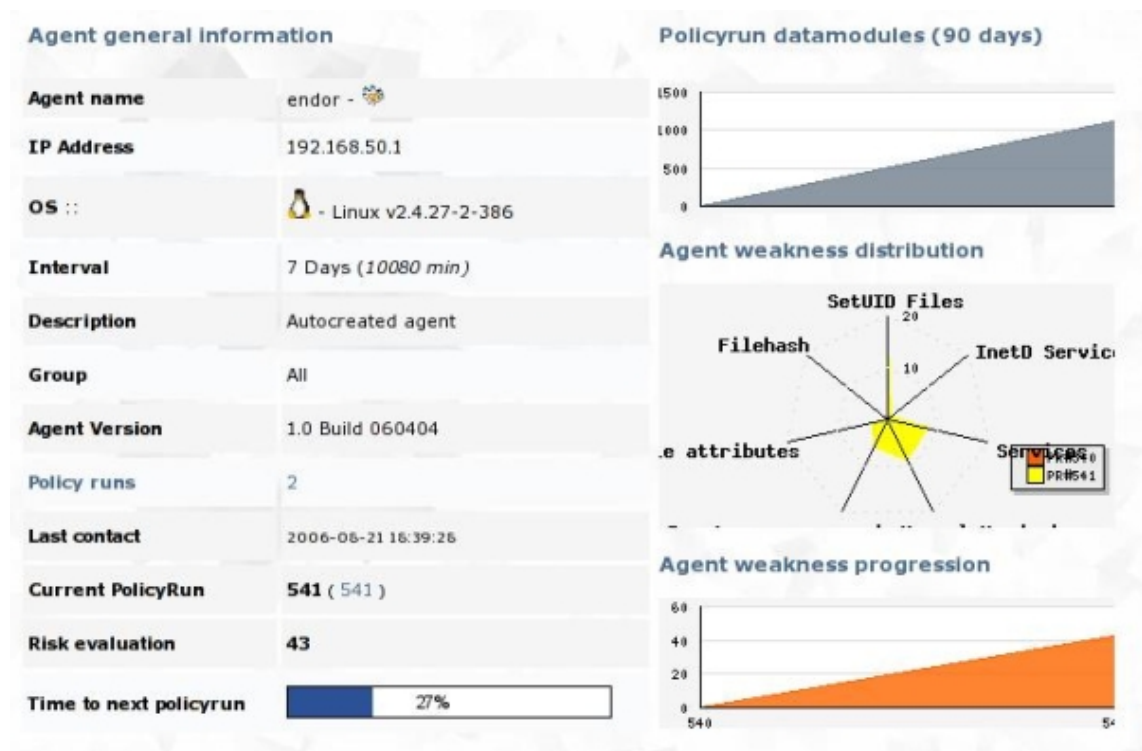
Ports listed in this file are ignored for OpenPort module. You can specify TCP or UDP protocols only.

4.2. Auditing with *Babel Enterprise*

4.2.1. Your first audit

Audits are programmed and configured in agent local configurations, so when you run an agent, a new audit is automatically launched. The first audit is called "learning" audit. None result is shown in Babel Console, so you must launch a second before see any information in console.

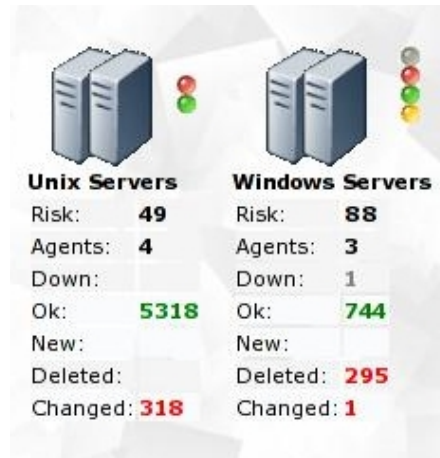
When you execute your first "valid" audit, enter in your agent main screen. You must see something similar to:



Agent weakness distribution graph is only shown when agent has five or more modules. Agent weakness progression graph show interesting data only when you have at least five or more audits and have any variations in security indicators. This screen is your "central command" for this Agent.

4.2.2. Analyzing your audit

In the first screen you have a group view. This screen shows you many parameters for all agents inside groups. You must see something similar to this screen:



. In this capture, Unix Servers group has four agetns. Thereare a total of individual data items of 5318, 318 of them have changed from last policy run. *Global Risk Indicator for this entire group is 49*. The same value for Windows Servers group shown a Risk of 88, so Unix Servers are more secure in this "snapshot".

You need to have many policy runs to see the effects of "improving" your security. If you work on secure your servers, in each policy run you must see the changes and the global risk indicator will decrease. A high Risk Indicator shows a bad security in your systems, with many security risks. Risk Indicator for a group is made with a arithmetical media for all agents inside it.

Of course, you may prefer a more detailed view, like this:

Agent	OS	Interval	Group	Risk	Modules	Policy runs	Status
endor		7 Days	All	43	9	2	
daeva		4 Days	Unix Servers	77	10	15	
ortega		9 Days	Unix Servers	35	12	2	
pacheco		8 Days	Unix Servers	38	6	2	
antiriad		7 Days	Windows Servers	119	8	459	
winbox01		1 Days	Windows Servers	61	3	17	
xpbox01		4 Days	Windows Servers	82	5	23	

Here you can see a brief info for all agents in *Babel Enterprise*. Interval is "translated" in a more human language, from minutes to days. Usually a good policy is to run a policy for week in each system. Critical systems could launch a daily policy run (use it with care to not include unnecessary information in policy).



show a changed value. A changed value in a filehash module could be dangerous!.



show a deleted value. For example, a software package that it's removed, or better, a file that has been deleted!.



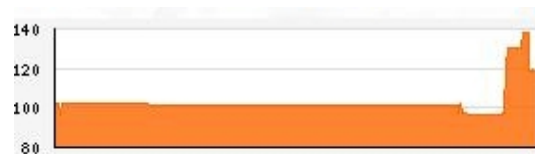
show a new value. For example, a software package that has not installed in last policy run, or a new file or problem in any audit module.



show that there are "no change" for at least one value.



Datamodules graph show total number of individual items (atomic pieces of information about your system) that has arrived in each policy run. It's a good indicator about the quantity of information coming in each policy run.



Perhaps the most important graph here is the "risk indicator" graph. It shown a progression in time for the global risk indicator of this agent. It shown if your system security is improving or not.

4.3. Reviewing your audit data

Search option gives you a way to manage your audit data. It allows to compare data from different

machines or search across the changes between audits.

Data search

Agent

Group

Module type

Search into:

☒ Item

☐ Data

☐ Description

☐ Search all policy runs

Search string

Search

Data search

Agent	Group	Policy run	Module name	Module type	Item	Data	Description
daeva	Unix Servers	539	Installed packages	software	openssh-client	4.2p1-7ubuntu3	Secure shell client, an rlogin/rsh/rcp repla
daeva	Unix Servers	539	Installed packages	software	openssh-server	4.2p1-7ubuntu3	Secure shell server, an rshd replacement
daeva	Unix Servers	539	Installed packages	software	ssh	4.2p1-7ubuntu3	Secure shell client and server (transitional)
antiriad	Windows Servers	529	Software	software	SSH Secure Shell	(null)	Installed software
endor	Unix Servers	541	Installed packages	software	ssh	3.8.1p1-8.sarge.4	Secure rlogin/rsh/rcp replacement (OpenSSH)

In agent main view, you see -by default- the last audit policy results in a graphical view like this:

Last policy evaluation. Policy run # 529

Standard policy - 2006-06-13 12:18:00

Name	Module type	Risk	Data	Status	Nochange	New	Deleted	Changed	Description
Filehash	filehash	0			135		135		Autocreated by BabelServer
Patch	patch	0			61				Autocreated by BabelServer
Services	services	91			91				Autocreated by BabelServer
Openport	openport	0			6				Autocreated by BabelServer
Registry	registry	0			1		1		Autocreated by BabelServer
Registry kernel	registry_kernel	22			21		23	1	Autocreated by BabelServer
Password	password	6			3				Autocreated by BabelServer
Software	software	0			136				Autocreated by BabelServer

Each audit module shows their data in a table. *risk* is the most important value. Data icon, open a new window with all data collected for this module. In a simple and graphical way, are four colors to show valid (green), modified (yellow), new data (blue) and deleted data (red), like the global agent view.

You also have three columns (if has data) to see with detail New, Deleted or Changed data for each module.

Appendix A. GNU Free Documentation License

A.1. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A.3. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. 3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material

on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- **C.** State on the Title Page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.

- **I.** Preserve the section entitled “History”, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** In any section entitled “Acknowledgements” or “Dedications”, preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section entitled “Endorsements”. Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version .

A.6. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms

defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

A.7. 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

A.9. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

A.10. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation (<http://www.gnu.org/fsf/fsf.html>) may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/> (<http://www.gnu.org/copyleft/>).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. Addendum

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have no Invariant Sections, write “with no Invariant Sections” instead of saying which ones are invariant. If you have no Front-Cover Texts, write “no Front-Cover Texts” instead of “Front-Cover Texts being LIST”; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>), to permit their use in free software.

Appendix B. GNU General Public License

B.1. Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and
2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

B.2. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

B.2.1. Section 0

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

B.2.2. Section 1

You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

B.2.3. Section 2

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

- b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: If the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

B.2.4. Section 3

You may copy and distribute the Program (or a work based on it, under Section 2 in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

B.2.5. Section 4

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.2.6. Section 5

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

B.2.7. Section 6

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

B.2.8. Section 7

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not

limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

B.2.9. Section 8

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

B.2.10. Section 9

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

B.2.11. Section 10

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

B.2.12. NO WARRANTY Section 11

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “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 PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

B.2.13. Section 12

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

B.3. How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type “show w”. This is free software, and you are welcome to redistribute it under certain conditions; type “show c” for details.

The hypothetical commands “show w” and “show c” should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than “show w” and “show c”; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program “Gnomovision” (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary

applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.