



we protect your digital worlds

ESET Gateway Security

*Installation Manual
and User's documentation*

Table of contents

1. Introduction	3
2. Terminology and abbreviations	5
3. Installation	9
4. Product's Roadmap	11
5. Integration with Internet Gateway services	15
5.1. Transparent HTTP/FTP proxy configuration	16
5.2. Manual HTTP proxy configuration	17
5.2.1. Manual proxy configuration of Mozilla Firefox	17
5.2.2. Manual proxy configuration of Squid Web Proxy Cache	18
5.3. Large HTTP Objects Handling	19
5.3.1. Method of deferred scan	19
5.3.2. Partial scan technique	19
5.4. ESETS plug-in filter for SafeSquid Proxy Cache	20
5.4.1. Operation principle	20
5.4.2. Installation and configuration	20
6. Important ESET Gateway Security mechanisms	23
6.1. Handle Object Policy	24
6.2. User Specific Configuration	24
6.3. Black-list and white-list	25
6.4. Samples Submission System	26
6.5. World WideWeb Interface	26
7. ESET Security system update	29
7.1. ESETS update utility	30
7.2. ESETS update process description	30
8. Let us know	33
A. ESETS setup process description	35
A.1. Setting ESETS for scanning of HTTP communication - transparent mode	36
A.2. Setting ESETS for scanning of FTP communication - transparent mode	36
Appendix A. PHP License	37

ESET Gateway Security

Copyright © 2007 ESET, spol. s r.o.

ESET Gateway Security was developed by ESET, spol. s r.o. For more information visit www.eset.com.

All rights reserved. No part of this documentation may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise without a permission in writing from the author.

ESET, spol. s r.o. reserves the right to change any of the described application software without prior notice.

This product includes PHP software, freely available from <http://www.php.net/software/>.



Chapter 1:

Introduction



Dear user, you have acquired ESET Gateway Security - probably the best security system running under the Linux/BSD OS. As you will soon find out, the system using the state-of-the-art ESET scanning engine, has unsurpassed scanning speed and detection rate, combined with a very small footprint that makes it the ideal choice for any Linux/BSD OS server. In the rest of this chapter we review a key features of the system.

- The ESET anti-virus scanning engine algorithms provide the highest detection rate and the fastest scanning times.
- The ESET Gateway Security is developed to run on the single-processor as well as on the multi-processor units.
- It includes unique advanced heuristics for Win32 worms and back-doors.
- Inbuilt archivers unpack archived objects without the need for any external programs.
- In order to increase speed and efficiency of the system, its architecture is based on the running daemon (resident program) where all the scanning requests are sent to.
- The system supports selective configuration specific for user or client/server identification.
- Six logging levels can be configured to get information about system activity and infiltrations.
- The ESET Gateway Security installation does not require external libraries or programs except for LIBC.
- The system can be configured to notify any person in case of detected infiltration.

To run efficiently, ESET Gateway Security requires just 16MB of hard-disk space and 32MB of RAM. It works smoothly under the 2.2.x, 2.4.x and 2.6.x Linux OS kernel versions and also under 5.x, 6.x FreeBSD OS kernel versions.

From lower-powered, small office servers to enterprise-class ISP servers with thousands of users, the system delivers the performance and scalability you expect from a UNIX based solution and the unequalled security of ESET products.



Chapter 2:

Terminology and abbreviations



In the following text we review terms and abbreviations used in this documentation. Note that in this documentation (PDF format only) a boldface font is reserved for product components names and in this chapter also for newly defined terms and abbreviations. Note also that terms and abbreviations defined in this chapter are emphasized later in this documentation (PDF format only).

ESETS

ESET Security is a common acronym for all security products developed by ESET, spol. s r.o. for Linux OS (resp. for BSD OS). It is also the name (or its part) of the software package containing the products.

RSR

Abbreviation of 'RedHat/Novell(SuSE) Ready'. Note that we support also so called RedHat Ready and Novell(SuSE) Ready variation of the product. The difference from the "standard" Linux version is that the *RSR* package meets criteria defined by FHS (File-system Hierarchy Standard defined as a part of Linux Standard Base) document required by the RedHat Ready and Novell(SuSE) Ready certificate. This means for instance that the *RSR* package is installed as an add-on application, i.e. the primary installation directory is '/opt/ezet/sets'.

ESETS daemon

Main *ESETS* system control and scanning daemon **esets_daemon**.

ESETS base directory

The directory where *ESETS* loadable modules containing for instance virus signatures database are stored. Further in this documentation we use abbreviation **@BASEDIR@** for the directory. The directory location is as follows:

```
Linux: /var/lib/esets
Linux RSR: /var/opt/ezet/esets/lib
BSD: /var/lib/esets
```

ESETS configuration directory

A directory where all files related with the ESET File Security configuration are stored. Further in this documentation we use abbreviation **@ETCDIR@** for the directory. The directory location is as follows:

```
Linux: /etc/esets
Linux RSR: /etc/opt/ezet/esets
BSD: /usr/local/etc/esets
```

ESETS configuration file

Main ESET File Security configuration file. The absolute path of the file is as follows:

```
@ETCDIR@/esets.cfg
```

ESETS binary files directory

The directory where the relevant ESET File Security binary files are stored. Further in this documentation we use abbreviation **@BINDIR@** for the directory. The directory location is as follows:

```
Linux: /usr/bin
Linux RSR: /opt/ezet/esets/bin
BSD: /usr/local/bin
```

ESETS system binary files directory

The directory where the relevant ESET File Security system binary files are stored. Further in this documentation we use abbreviation **@SBINDIR@** for the directory. The directory location is as follows:

```
Linux: /usr/sbin  
Linux RSR: /opt/eset/esets/sbin  
BSD: /usr/local/sbin
```

ESETS object files directory

The directory where the relevant ESET File Security object files and libraries are stored. Further in this documentation we use abbreviation **@LIBDIR@** for the directory. The directory location is as follows:

```
Linux: /usr/lib/esets  
Linux RSR: /opt/eset/esets/lib  
BSD: /usr/local/lib/esets
```





Chapter 3:

Installation



This product is distributed as a binary file:

```
esets.i386.ext.bin
```

where 'ext' is a Linux/BSD OS distribution dependent suffix, i.e. 'deb' for Debian, 'rpm' for RedHat and SuSE, 'tgz' for other Linux OS distributions, 'fbs5.tgz' for FreeBSD 5.xx and 'fbs6.tgz' for FreeBSD 6.xx distributions.

Note that the Linux *RSR* binary file format is:

```
esets-rsr.i386.rpm.bin
```

In order to install or update the product, use statement:

```
sh ./esets.i386.ext.bin
```

resp. for Linux *RSR* variation of the product, use statement:

```
sh ./esets-rsr.i386.rpm.bin
```

As a result the product's User License Acceptance Agreement is shown. Once you have confirmed the Acceptance Agreement, the installation package is placed into the current working directory and relevant information regarding the package's installation, un-installation or update is printed into terminal.

Once the package is installed and the main *ESETS* service is running, in Linux OS you can check its operation by using command:

```
ps -C esets_daemon
```

In case of BSD OS you can use a similar command:

```
ps -ax esets_daemon | grep esets_daemon
```

You will see the following (or similar) message on return:

PID	TTY	TIME	CMD
2226	?	00:00:00	esets_daemon
2229	?	00:00:00	esets_daemon

where at least two *ESETS daemon* processes running in the background have to be present. One of the processes is so-called process and threads manager of the system. The other serves as *ESETS* scanning process.

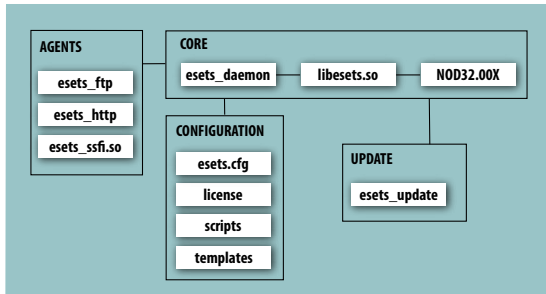
Chapter 4:

Product's Roadmap

Once the product package has been successfully installed, it is time to become familiar with its content.

The structure of ESET Gateway Security is shown in the figure 4-1. The system is composed of the following components.

Figure 4-1. Structure of ESET Gateway Security.



CORE

Core of ESET File Security consists of *ESETS daemon* **esets_daemon**. The daemon uses *ESETS* API library **libesets.so** and *ESETS* loading modules **nod32.00X** to provide base system tasks: scanning, maintenance of the agent daemon processes, maintenance of the samples submission system, logging, notification, etc.. Please refer to **esets_daemon(8)** manual page for details.

AGENTS

The purpose of *ESETS* agent modules is to integrate *ESETS* with the Linux/BSD Server environment. Please note a special chapter in this document devoted to the topic.

UPDATE

The update utility is a particular fraction of the system. It is developed to update *ESETS* loading modules containing for instance virus signatures database, archives support, advanced heuristics support etc. Please note a special chapter in this document devoted to the topic.

CONFIGURATION

Proper configuration is the most important condition for the system operation. Therefore we describe all the related components in the rest of this chapter. We also strongly recommend to read **esets.cfg(5)** manual page, an essential information source regarding *ESETS* configuration. After the product is successfully installed, all its configuration components are stored in *ESETS configuration directory*. The directory consists of the following files.

@ETCDIR@/esets.cfg

This is the most important configuration file as it maintains the major part of the product functionality. After exploring the file you can see that it is built from various parameters distributed within sections. Note the section names always enclosed in square brackets. In the *ESETS configuration file* there is always one global and several so-called agent sections.

Parameters in global section are used to define configuration options of *ESETS daemon* as well as default values of *ESETS* scanning engine configuration options. Parameters in agent sections are used to define configuration options of so-called agents, i.e. modules used to intercept various data flow types in the computer and/or its neighborhood and prepare this data for scanning. Note that besides the number of parameters used for the system configuration, there is also a number of rules determining organization of the file. To become familiar with this knowledge, please refer to *esets.cfg(5)*, *esets_daemon(8)* manual page and also to manual pages related to relevant agents.

@ETCDIR@/certs

This directory is used to store the certificates used by *ESETS* WWW Interface for authentication (see *esets_wwwi(8)* for details).

@ETCDIR@/license

This directory is used to store the product(s) license key(s) you have acquired from your vendor. Note that the *ESETS daemon* will always check only this directory to evaluate license key validity unless it is redefined by *ESETS configuration file* parameter 'lic_dir'.

@ETCDIR@/scripts/license_warning_script

This script, if enabled by *ESETS configuration file* parameter 'license_warn_enabled', is executed since 30 days (once per day) before product license expiration. It is used to send e-mail notification about the expiration status to system administrator.

@ETCDIR@/scripts/daemon_notification_script

This script, if enabled by *ESETS configuration file* parameter 'exec_script', is executed in case the infiltration has been detected by the anti-virus system. It is used to send e-mail notification about the event to system administrator.

@ETCDIR@/templates/http_*.html.example

These files are html templates used in various non-successful returning states of *esets_http* module. The meaning of individual files is as follows.

The following templates are used in case the object downloaded was found as infected.

```
content of http_header_infected.html
list of infiltrations found by the scanner
content of http_footer_infected.html
```

The following templates are used in case the object downloaded could not be scanned.

```
content of http_header_not_scanned.html
list of object scanned by the scanner
content of http_footer_not_scanned.html
```



Chapter 5:

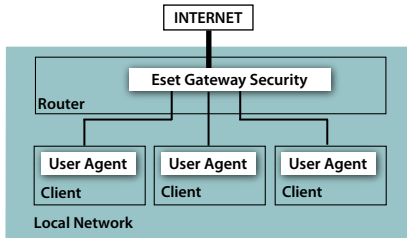
Integration with Internet Gateway services

The ESET Gateway Security protects organization's HTTP and FTP services against viruses, worms, trojans, spyware, phishing and other internet threats on the level of Internet Gateway Servers. Note that under the term Gateway Servers we understand layer-3 Gateways of ISO/OSI model, i.e. routers. In this chapter we review the process of the product integration with the services introduced.

5.1. Transparent HTTP/FTP proxy configuration

Configuration for transparent proxying is based on standard routing mechanism shown in the following figure.

Figure 5-1. Scheme of ESET Gateway Security as a transparent proxy.



The configuration is created naturally as kernel IP routing tables are defined on each local network client. These routing tables are used to set-up static routes to the default network gateway server (router). Note that it is done automatically in case of the DHCP network. Using this mechanism all the HTTP (resp. FTP) communication with the outbound servers is routed via network gateway server where ESET Gateway Server must be installed in order to scan the communication for infiltrations. For this purpose, a generic *ESETS* HTTP (resp. FTP) filter - **esets_http** (resp. **esets_ftp**) has been developed.

In order to configure ESET Gateway Security for scanning of HTTP (resp. FTP) messages routed through the network gateway server, enter the command:

```
esets_setup
```

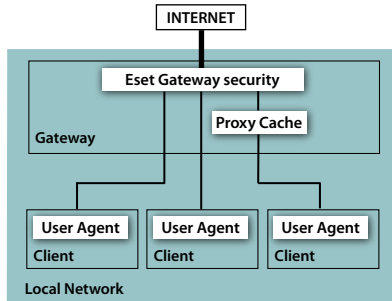
Follow instructions provided by the script. Once 'Available installations/un-installations' offer appears, choose 'HTTP' (resp. FTP) option that will provide you with the appropriate module 'install/uninstall' options. Choose the one called 'install'. This will automatically set-up the module to listen to predefined port and redirect IP packets originating from the selected network and with HTTP (resp. FTP) destination port to the port where **esets_http** (resp. **esets_ftp**) listens. This means that only requests originally sent to HTTP (resp. FTP) destination port will be scanned. If other ports are under interest, an equivalent redirection rules have to be assigned.

Note that the installer in default mode shows all steps it is going to perform and provide also the backup of the configuration that may be restored later at any time. The detailed installer utility steps for all possible scenarios are described also in the appendix A of this documentation.

5.2. Manual HTTP proxy configuration

The manual proxy configuration (see figure 5-2) is characteristic by explicit specifying of parent proxy listen address and port in the configuration of proxied user agent.

Figure 5-2. Scheme of ESET Gateway Security as a manual proxy.



The proxy server in this case usually modifies requests and/or responses transferred, i.e. works in the non-transparent mode. The manual proxying support of **esets_http** has been tested with the wide range of most common user agents, i.e. proxy caches (Squid Proxy Cache, SafeSquid), client browsers (Mozilla Firefox, Opera, Netscape, Konqueror). In general every HTTP user agent supporting manual parent proxy settings will cooperate with **esets_http** module. In the next we describe manual proxy configuration setting of **esets_http** with Mozilla Firefox and with Squid Web Proxy Cache that are one of the most common HTTP user agent applications.

5.2.1. Manual proxy configuration of Mozilla Firefox

The manual HTTP proxy configuration of **esets_http** with the Mozilla Firefox is described in general by the left side of the figure 4-2.

Note that this configuration allows to install ESET Gateway Security anywhere within the local network including gateway server and also user agent's computer.

In this example we configure **esets_http** to listen to port 8080 of the computer with local network IP address 192.168.1.10 by specifying the following parameters within [http] section of main *ESETS configuration file*:

```
agent_enabled = yes
listen_addr = "192.168.1.10"
listen_port = 8080
```

Note that the parameter 'listen_addr' can be specified also as the host name visible from the local network.

In order to set up Mozilla to use **esets_http** one has to select the 'Edit' menu from the menu bar and from there 'Preferences' (in older versions of Mozilla one has to select 'Tools' menu and from there 'Options'). Then click on 'Connection settings' found under 'General' settings and

select 'Manual Proxy Configuration'. In the last step, one has to fill up the fields 'HTTP Proxy' with the host name (resp. IP address) and related 'Port' with the port where **esets_http** listens to (in this example an IP address '192.168.1.10' and port 8080 shall be specified). For reread of newly created configuration, reload *ESETS daemon*.

It is good to note that the configuration described here is not very suitable for networks with higher number of client's computers. The reason is that in this case the HTTP cache (if any) is present only in user agent and thus the same source object is scanned multiple times when requested from different user agents.

5.2.2. Manual proxy configuration of Squid Web Proxy Cache

The manual HTTP proxy configuration of **esets_http** with the Squid Web Proxy Cache is described in general by the right side of the figure 4-2.

The significant difference from the previously described configuration is that the ESET Gateway Security is installed in HTTP Gateway between proxy cache (Squid Web Proxy in this example) and the Internet. Thus all the HTTP responses incoming to the network are first scanned for infiltrations and afterward stored in the network dedicated cache, i.e. all once requested source objects present within a proxy cache are already checked for viruses and no additional check is necessary when requested again.

In this example we configure **esets_http** to listen to port 8080 of the gateway server with local network IP address 192.168.1.10 by specifying the following parameters within [http] section of *ESETS configuration file*:

```
agent_enabled = yes
listen_addr = "192.168.1.10"
listen_port = 8080
```

Note that the parameter 'listen_addr' can be specified either as the host name visible from the local network or one may also use 0.0.0.0 address to let **esets_http** listen to all interfaces. In the later case one has to be careful as also users outside the local network are allowed to use HTTP scanner unless further security steps are provided to prevent from it.

In order to set up Squid to use **esets_http** as parent proxy one has to add the following lines into the Squid configuration file (*/etc/squid/squid.conf*):

```
cache_peer 192.168.1.10 parent 8080 0 no-query default
acl all src 0.0.0.0/0.0.0.0
never_direct allow all
```

In the lines above we have set up the Squid to use http proxy listening on IP address 192.168.1.10 at port 8080 as a parent proxy. All requests processed by Squid will be thus passed to this destination. The rest of the lines define behavior of Squid to report error message in case the parent proxy is down or becomes unreachable. There is an alternative way to set up Squid in order to try direct connections when the parent proxy is unreachable. In this case the parameters to add into Squid configuration file are as follows:

```
cache_peer 192.168.1.10 parent 8080 0 no-query
prefer_direct off
```

For reread of newly created configuration, reload *ESETS daemon*.

5.3. Large HTTP Objects Handling

Under the normal conditions **esets_http** handles every object transferred in a way that the object is first transferred from the HTTP server (resp. client) to **esets_http**, second, it is scanned for infiltration and last, it is transferred to the HTTP client (resp. server). Concerning large files (the large objects whose transfer time is larger than timeout defined by the parameter `lo_timeout`) this becomes not very suitable scenario as the user agent's timeout or user's impatience can cause interrupts or even canceling of the objects transfer. Therefore other methods to process the large objects must be implemented.

5.3.1. Method of deferred scan

The **esets_http** implements standard so-called 'deferred scan' method of large files handling. This means if object transferred becomes large the **esets_http** starts to send the object transparently to an awaiting HTTP end-point (i.e. client or server). After the last part of the object has arrived to **esets_http**, the object is scanned for infiltrations. If the object has been found as infected the last part of the object (current version of ESET Gateway Security defines last part as last 4KB of object's data) is not sent to the awaiting end-point and the connection with the end-point is dropped. In parallel, the e-mail notification is sent to the Gateway administrator with the relevant information about the dangerous file transfer. Note that the notification is sent only in case of server to client data transfer. The URL of the source object is stored in this case in the **esets_http** cache to block the source transfer if requested again.

In this place we would like to point out that the 'deferred scan' technique described above presents potential risk for the computer whose user agent requested the infected large file for the first time. The risk resides in that even data transfer of an infected object has been deferred some parts of already transferred data can contain executable danger code. That is why the ESET developed modification of the 'deferred scan' technique called 'partial scan' technique.

5.3.2. Partial scan technique

The 'partial scan' technique has been developed to safeguard 'deferred scan' method. Operation principle of the 'partial scan' technique is based on the idea that scanning time of a large object is negligible as compared to overall process time of the object. Note that this condition is fulfilled in case of HTTP transfer of large object as significantly higher time is needed to transfer the object than to scan it for infiltrations. This assumption allows us to perform more than only one scan during the large object transfer.

Once parameter `lo_partscan_enabled` is enabled in [http] section of *ESETS configuration file* the large object is scanned for infiltrations during its transfer in some predefined intervals and data scanned are sent to awaiting end-point (i.e. to client or to server). Using this method there is no way to pass any infiltration to the computer whose user agent has requested the large infected object as each portion of the data sent is already ensured to be secure.

It has been proved that in the common circumstances (by means the speed of the Gateway local network connection is orderly higher than the speed of the Gateway connection to the Internet) the process time of the large object transfer with the 'partial scan' technique used is approximately the same as when the standard 'deferred scan' method used.

5.4. ESETS plug-in filter for SafeSquid Proxy Cache

In the previous sections we have described integration of ESET Gateway Security with the Internet Gateway HTTP and FTP services using `esets_http` and `esets_ftp`. Although the methods described are applicable for most common user agents including very well known content filtering internet proxy - SafeSquid (<http://www.safesquid.com>) the ESET Gateway Security offers for this special case also an alternative way to protect the Gateway services using `esets_ssfi.so` module developed for this purpose.

5.4.1. Operation principle

The `esets_ssfi.so` module is a plug-in with the purpose to access all objects processed by SafeSquid proxy cache using special interface developed by SafeSquid people for this purpose. Once the plug-in accesses the object it is scanned for infiltrations using *ESETS daemon*. If the object is infected the SafeSquid blocks an appropriate resource and sends predefined template page instead. Note that `esets_ssfi.so` is supported by SafeSquid Advanced version 4.0.4.2 and higher.

5.4.2. Installation and configuration

In the following we assume that the SafeSquid is installed in `/opt/safesquid` directory. To integrate the module you have to provide links from the SafeSquid modules directory to the appropriate installation locations of ESET Gateway package.

```
ln -s @LIBDIR@/ssfi/esets_ssfi.so /opt/safesquid/modules/esets_ssfi.so
ln -s @LIBDIR@/ssfi/esets_ssfi.xml /opt/safesquid/modules/esets_ssfi.xml
```

Once SafeSquid plug-in installed it is ready to work, however, some more fine tunings in the SafeSquid configuration are needed. In the next we at least configure SafeSquid to use *ESETS* predefined blocking templates in case the transferred source object is infected (resp. not scanned).

Log in to the SafeSquid Web Administration Interface, select the 'Config' menu from the main interface page and browse down the sections in 'Select a Section to Configure' until 'ESET Gateway Security' section found. Next, add the new templates definitions by pressing 'Add' at the bottom of the 'ESET Gateway Security' section and define the following parameters in the list that appears

```
Comment: ESET Gateway Security blocking templates
Profiles: antivirus
Infected template: esets_infected
Not scanned template: esets_not_scanned
```

After submitting the list of templates go to 'Templates' page of the main 'Config' menu. You shall see a parameter 'Path' that defines SafeSquid templates directory path (in the next we assume the parameter is `/opt/safesquid/templates`). Ensure that an appropriate directory exists and if not, create it. To access the *ESETS* predefined templates from within this directory you have to provide an appropriate links using the following shell statements:

```
ln -s @LIBDIR@/ssfi/templates/ssfi_infected.html /opt/safesquid/ssfi_infected.html
ln -s @LIBDIR@/ssfi/templates/ssfi_not_scanned.html /opt/safesquid/ssfi_not_scanned.html
```

You have also to add the new templates definitions in the SafeSquid configuration by pressing 'Add' in the 'Templates' section. In the list that appears the following parameters have to be defined for infected *ESETS* blocking page:

```
Comment: ESET Gateway Security infected template
Name: esets_infected
File: ssfi_infected.html
Mime type: text/html
Response code: 200
Type: File
Parsable: Yes
```

resp. for not-scanned *ESETS* blocking page the list is as follows:

```
Comment: ESET Gateway Security not scanned template
Name: esets_not_scanned
File: ssfi_not_scanned.html
Mime type: text/html
Response code: 200
Type: File
Parsable: Yes
```

For reread of newly created configuration, reload SafeSquid and also *ESETS daemon*.



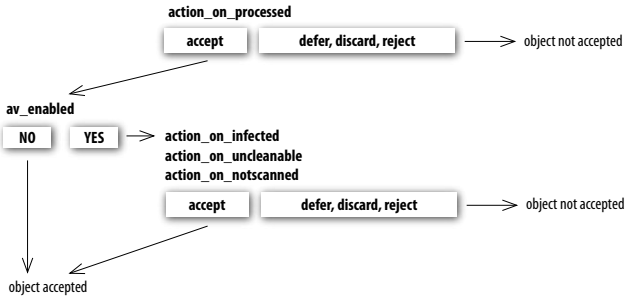
Chapter 6:

Important ESET Gateway Security mechanisms

6.1. Handle Object Policy

The Handle Object Policy (see figure 6-1) is a mechanism that provides handling of the scanned objects depending on their scanning status. The mechanism is based on so-called action configuration options ('action_on_processed', 'action_on_infected', 'action_on_uncleanable', 'action_on_notscanned') combined with Anti-Virus enabling configuration option ('av_enabled'). For detailed information on the options, please refer to the esets.cfg(5) manual page.

Figure 6-1. Scheme of Handle Object Policy mechanism.



Every object processed is at first handled with respect to the setting of the configuration option 'action_on_processed'. Once the option is set to 'accept', the object is handled according to the setting of configuration option 'av_enabled'. Once 'av_enabled' is enabled the object is scanned for virus infiltrations and set of action configuration options 'action_on_infected', 'action_on_uncleanable' and 'action_on_notscanned' is taken into account to evaluate further handling of the object. If action 'accept' has been taken as a result of the three above options or 'av_enabled' is disabled the access to the object is allowed, otherwise the object is blocked.

6.2. User Specific Configuration

User Specific Configuration mechanism is implemented in the product in order to provide administrator with enhanced configuration functionality. It allows to define *ESETS* anti-virus scanner parameters selectively for client/server identification.

Please note that the detailed description of this functionality can be found in esets.cfg(5) manual page and manual pages referenced there. Thus in this section we will only provide short example of user specific configuration definition.

Let's say we use **esets_http** to control HTTP traffic on port 8080 of the gateway server with local network IP address 192.168.1.10. The module is subjected to configuration section [http] in *ESETS configuration file*. The section is as follows:

```
[http]
agent_enabled = yes
listen_addr = "192.168.1.10"
```

```
listen_port = 8080
action_on_processed = accept
```

In order to provide individual parameters setting one has to define 'user_config' parameter with the path to the special configuration file where the individual setting will be stored. In the next example we create reference to the special configuration file 'esets_http_spec.cfg' located within the *ESETS configuration directory*.

```
[http]
agent_enabled = yes
listen_addr = "192.168.1.10"
listen_port = 8080
action_on_processed = accept
user_config = "esets_http_spec.cfg"
```

Once special configuration file referenced from within [http] section we have to create the file in the *ESETS configuration directory* and provide it with an appropriate individual settings. The next example shows individual setting of parameter 'action_on_processed' for client's IP address 192.168.1.40.

```
[192.168.1.40]
action_on_processed = reject
```

Note that the section header name of the special section contains identification of the HTTP client for which we have created individual setting. The section body then contains individual parameters specified for this identification. Thus with this special configuration an HTTP traffic of all local network clients will be processed, i.e. scanned for infiltrations, with exception of the client determined by IP address 192.168.1.40 that will be rejected, i.e. blocked in any case.

6.3. Black-list and white-list

In the next example we demonstrate the black-list and also white-list creation for the **esets_http** configured as an HTTP proxy scanner. Note that we use configuration described in the previous section for this purpose.

Thus in order to create black-list used by **esets_http** we have to create the following group section within the special configuration file 'esets_http_spec.cfg' introduced in the previous section.

```
[black-list]
action_on_processed = reject
```

The next step is to add some HTTP server into the 'black-list' group. For this purpose we have to create special section

```
[aaa.bbb.ccc.ddd]
parent_id = "black-list"
```

where 'aaa.bbb.ccc.ddd' is an IP address of the server added into the 'black-list'. Note that with this setting all HTTP traffic related with the specified server will be rejected, i.e. the server will be blocked.

If we want to create the 'white-list' used by `esets_http` we have to create the following group section within the special configuration file 'esets_http_spec.cfg' introduced in the previous section.

```
[white-list]
action_on_processed = accept
av_enabled = no
```

Adding of HTTP servers into the list is self-explanatory.

6.4. Samples Submission System

Samples submission system is an intelligent ThreatSense.NET technology that provides catching of the infected objects found by advanced heuristics method and delivering these objects to the samples submission system server. All virus samples caught by the sample submission system will be processed by the team of ESET virus laboratory department and consequently added into the ESET virus database, if necessary.

NOTE: ACCORDING TO OUR LICENSE AGREEMENT, BY ENABLING SAMPLE SUBMISSION SYSTEM YOU ARE AGREEING TO ALLOW THE COMPUTER AND/OR PLATFORM ON WHICH THE `ESETS_DAEMON` IS INSTALLED TO COLLECT DATA (WHICH MAY INCLUDE PERSONAL INFORMATION ABOUT YOU AND/OR THE USER OF THE COMPUTER) AND SAMPLES OF NEWLY DETECTED VIRUSES OR OTHER THREATS AND SEND THEM TO OUR VIRUS LAB. THIS FEATURE IS TURNED OFF BY DEFAULT. WE WILL ONLY USE THIS INFORMATION AND DATA TO STUDY THE THREAT AND WILL TAKE REASONABLE STEPS TO PRESERVE THE CONFIDENTIALITY OF SUCH INFORMATION.

In order to turn on Samples Submission System, the samples submission system cache has to be initialized. This can be achieved by enabling configuration option 'samples_enabled' in [global] section of *ESETS configuration file*. In order to enable process of samples delivery to ESET virus laboratory servers it is yet necessary to enable parameter 'samples_send_enabled' in the same section.

User may decide to provide the ESET virus laboratory team with the additional optional information using configuration options 'samples_provider_mail' and/or 'samples_provider_country'. This information will help us to get overview on the infiltration spreading throughout the Internet.

In order to get detailed information on the Samples Submission System, refer to `esets_daemon(8)` manual page.

6.5. World WideWeb Interface

WWW Interface allows user-friendly ESETS configuration, administration and license management.

This module is a separate agent and must be explicitly enabled. For quickstart, set all of these options in *ESETS configuration file* and restart *ESETS daemon*:

```
[wwwi]
agent_enabled = yes
listen_addr = address
listen_port = port
username = name
```

```
password = pass
```

(enter all four values as your own ones) and direct your browser to 'https://address:port' (note the *https*) and login with 'name/pass'. There are basic usage instructions on the help page. For more technical details about **e sets_wwwi** see the **e sets_wwwi(1)** manual page.

Chapter 7:

ESET Security system update

7.1. ESETS update utility

In order to keep the ESET Gateway Security effective, it is necessary to keep its virus signatures database up to date. The `esets_update` utility has been developed for this purpose (see `esets_update(8)` manual page for details). In order to launch update one has to define configuration options 'username' and 'password' in [update] section of *ESETS configuration file*. Note that in case you access the Internet via HTTP proxy additional configuration options 'proxy_addr', 'proxy_port' and optionally 'proxy_username' and 'proxy_password' have to be specified there as well. To trigger an update, enter command:

```
@SBINDIR@/esets_update
```

To provide the highest security for the user, the ESET team collects the virus definitions continuously from all over the world. The new patterns can appear within the database in very short intervals. It is therefore recommended, to trigger an update on a regular basis. Note that *ESETS daemon* is able to provide the periodic update of the system once 'av_update_period' configuration option specified in [update] section of *ESETS configuration file* and the daemon is up and running.

7.2. ESETS update process description

The update process is composed of two stages. First, the mirror of all relevant so-called pre-compiled modules have to be created from the origin ESET server. The pre-compiled modules are downloaded by default into directory

```
@BASEDIR@/mirror
```

Note that the mirror directory path can be redefined using configuration option 'mirror_dir' in section [update] of *ESETS configuration file*.

The *ESETS* modules are divided into two categories; engine category and component category. The modules of component category are currently only for use on the MS Windows OS. Currently the following types of engine category modules are supported: base scanning modules (prefix engine) containing virus signatures database, archives support modules (prefix archs) supporting various file system archive formats, advanced heuristics modules (prefix advheur) containing implementation of so-called advanced heuristics method of virus and worm detection, packed worm scanner modules (prefix pwscan) used on MS Windows OS, utilities modules (prefix utilmod) used on MS Windows OS and ThreatSense.NET technology support modules (prefix charon). These modules are always necessary and therefore are all downloaded by default at each download process. On the other hand the component category modules are platform dependent and language localization dependent and thus the download of component category modules is optional.

After download of the pre-compiled modules the 'update.ver' file is created in the mirror directory as well. This file contains the information about the modules currently stored in the newly created mirror. The newly created mirror thus serves as fully functional modules download server and can be used to create subordinate mirrors, however, some more conditions have to be fulfilled yet. First, there must be a http server installed on the computer where the modules are going to be downloaded from. Second, the modules to be downloaded by other computers have to be placed at the directory path

```
/http-serv-base-path/nod_upd
```

where 'http-serv-base-path' is a base http server directory path, as this is the first place where update utility looks the modules for.

Second part of the update process is the compilation of modules loadable by the ESET Gateway Security scanner from those stored in the local mirror. Typically the following *ESETS* loading modules are created: base module (nod32.000), archives support module (nod32.002), advanced heuristics module (nod32.003), packed worm scanner module (nod32.004), windows utilities module (nod32.005) and ThreatSense.NET support module (nod32.006) in the directory:

```
@BASEDIR@
```

Note that it is exactly the directory where *ESETS daemon* loads modules from and thus can be redefined by using configuration option 'base_dir' in section [global] (resp. [update]) of *ESETS configuration file*.





Chapter 8:

Let us know



Dear user, this guide should have given you a good knowledge about the ESET File Security installation, configuration and maintenance. However, writing a documentation is a process that is never finished. There will always be some parts that can be explained better or are not even explained at all. Therefore, in case of bugs or inconsistencies found within this documentation, please report a problem to our support center

<http://www.eset.com/support>

We are looking forward to help you solve any problem concerning the product.



Appendix A. *ESETS* setup process description

A.1. Setting *ESETS* for scanning of HTTP communication - transparent mode

The HTTP communication scanning is performed using **esets_http** daemon. In the [http] section of *ESETS configuration file* set these parameters:

```
agent_enabled = yes
listen_addr = "192.168.1.10"
listen_port = 8080
```

where 'listen_addr' is the address of local network interface named *if0*. Then restart *ESETS daemon*. The next step is to redirect all HTTP requests to **esets_http**. In case of IP-filtering provided by ipchains administration tool an appropriate rule is:

```
ipchains -A INPUT -p tcp -i if0 --dport 80 -j REDIRECT 8080
```

If IP-filtering mechanism is provided by iptables administration tool, the rule is:

```
iptables -t nat -A PREROUTING -p tcp -i if0 \
--dport 80 -j REDIRECT --to-ports 8080
```

resp. if ipfw tool used (in case BSD OS) the rule is as follows:

```
ipfw add fwd 192.168.1.10,8080 tcp from any to any 80 via if0 in
```

A.2. Setting *ESETS* for scanning of FTP communication - transparent mode

The FTP communication scanning is performed using **esets_ftp** daemon. In the [ftp] section of *ESETS configuration file* set these parameters:

```
agent_enabled = yes
listen_addr = "192.168.1.10"
listen_port = 2121
```

where 'listen_addr' is the address of local network interface named *if0*. Then restart *ESETS daemon*. The next step is to redirect all FTP requests to **esets_ftp**. In case of IP-filtering provided by ipchains administration tool an appropriate rule is:

```
ipchains -A INPUT -p tcp -i if0 --dport 21 -j REDIRECT 2121
```

If IP-filtering mechanism is provided by iptables administration tool, the rule is:

```
iptables -t nat -A PREROUTING -p tcp -i if0 \
--dport 21 -j REDIRECT --to-ports 2121
```

resp. if ipfw tool used (in case BSD OS) the rule is as follows:

```
ipfw add fwd 192.168.1.10,2121 tcp from any to any 21 via if0 in
```



Appendix A. PHP License



The PHP License, version 3.01 Copyright (c) 1999 - 2006 The PHP Group. All rights reserved. Redistribution and use in source and binary forms, with or without modification, is 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. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.
4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo".
5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes PHP software, freely available from <http://www.php.net/software/>".

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM ``AS IS'' AND ANY EXPRESSED 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 PHP DEVELOPMENT TEAM OR ITS 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, 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.