NAME

esets_daemon − Main ESETS system control and scanning Daemon module.

SYNOPSIS

esets_daemon [**OPTIONS** ...]

DESCRIPTION

esets_daemon is the main ESETS system control and scanning daemon module. It reads all the ESETS scanner configuration from the main ESETS configuration file and provides all the main ESETS tasks: anti-virus scanning of the specified objects, anti-spam scanning of specified e-mails, maintenance of the agent daemon processes, maintenance of the samples submission system using ThreatSense.NET intelligent technology, logging, notification, etc. Every task of the ESETS main scanning daemon is described in individual section of this manual page (see below).

SIGNALS AND DAEMON PROCESS STRUCTURE

After start of **esets_daemon** daemon, so called main manager process is created that performs scanner initialization.

Once scanner initialized manager process creates so called scanning process running **num_thrd** scanning threads. Note that scanning process is replaced by the new one each time the limited number **max_session** of sessions performed, or on hard or soft restart signal.

Besides scanning process, all daemon agents processes are maintained by main manager **esets_daemon** process. To launch an appropriate daemon agent, it must be enabled by using parameter **agent_enabled** in an appropriate agent section of the configuration file. See daemon agent modules manual pages for details.

The following signals are predefined to perform restart of the esets_daemon and daemon agents processes:

HUP This signal causes restart of the **esets_daemon** and all daemon agent processes, i.e. after **esets_daemon** manager process obtained this signal it sends TERM signal to all child processes and then terminates all scanning threads. Once all child processes terminated, the **esets_daemon** manager process restarts itself.

USR1 This signal causes so called hard restart of the **esets_daemon** and all daemon agent processes, i.e. after **esets_daemon** manager process obtained this signal it pass this signal to all child processes and terminates immediately all scanning threads. Once all child processes terminated, the **esets_daemon** manager process restarts itself.

TERM This signal causes termination of all **esets_daemon** and daemon agent processes.

CONFIGURATION FILE OPTIONS

The main ESETS configuration mechanism is assumed to be that using ESETS main configuration file. Note that most principles of this mechanism are described in the esets.cfg(5) manual page while this section contains only additional information related with the list of configuration file options valid for this particular module. Therefore we recommend user to become familiar with the above mentioned documentation prior reading this section.

ESETS MODULE PRIVATE OPTIONS

daemon_sock_path = *path*

type: string

default: *path* = "*/tmp/esets.sock*"

Defines path of the UNIX socket used by ESETS agent modules to communicate with the ESETS main scanning daemon.

num_thrd = *value*

type: integer

default: *value* = 2

Defines number of ESETS daemon scanning threads.

restricted_user = *user*

type: string

default: *user* = "esets"

In products without on-access all executive daemons will run as this user (dropping root privileges). If empty, no daemon will change user id. User creation and certain directories ownership adjustment will be done automatically.

av_clean_file_cache_size = *size*

type: integer

default: size = 50000

Number of clean scan results in file cache. Zero disables the cache completely.

scheduler_tasks = *tasks*

type: string

default: *tasks* = "TODO"

Scheduled tasks. See section SCHEDULER below.

mail_domains = *mail_domain_pattern1:mail_domain_pattern2:...*

type: string

default: no default

Scheduler allows to execute a given action when a mail is sent from local domain (see **LOCAL SENDER**). Whether the mail domain is considered as local or not depends on a mail sender. The sender address is searched in patterns listed in this option. If the address matches, the event "local sender is triggered.

av_mail_notified_users = *list of addresses patterns and users*

type: string

default: no default

Define list of addresses patterns and users, that will be notified if infected mail is sent. List have following format **pattern1:user1:pattern2:user2:...patternN:userN**. After mail will be scanned and some infection was found inside, the address of mail sender and recipient will be searched in patterns. Script **mail_notification_script** will be executed for each pattern, that match sender or recipient address. The script is stored scripts subdirectory of ESETS configuration directory and default it sends an mail to a user mentioned after pattern. Feel free to modify and customize this script. It works only in mail agents that provide sender and receiver (values from mail are not taken into account). It is all that provides user specific configuration. Mail options are passed through environment variables. You can find their description inside the script, and they are the same as in part **LOCAL SENDER** and one new was also added. It defines a mail address of nofitied user.

log_format_summ = *format*

type: string

default: *format* = "vdb=%vdb%, agent=%agent%, name="%name%", virus="%virus%",

action="%action%", info="%info%", avstatus="%avstatus%", hop="%hop%""

Defines ESETS logging format of the first (summary) line of logging output. The format string is a character string composed of zero or more directives conversion specifications:

%vdb%
Virus signatures database build number.

%vdv%
Virus signatures database version.

%agent%
ESETS agent module name.

%userspec%
ESETS user specification ID.

%msgid%
E-mail msgid.

%sndr%
Sender address taken from an e-mail header.

%rcpt%
Recipient address taken from an e-mail header.

%name%
Scanned object name.

%virus%
Virus name.

%action%
Action taken by scanner.

%info%
Additional information from scanner.

%hop%
Handle object policy action taken against the scanned object.

%avstatus%
Anti-virus scanning status string.

%asstatus%
Anti-spam scanning status string.

log_format_part = *format*

type: string

default: *format* = "vdb=%vdb%, agent=%agent%, name="%name%", virus="%virus%", action="%action%", info="%info%""

Defines ESETS logging format of other than first (particular objects) lines of logging output. The format string is a character string composed of zero or more directives conversion specifications surrounded by special character '%'. You can use all specifications like for the 'log_format_summ' parameter.

report_enabled = *yes/no*

type: bool

default: no

Enables/disables date storing for reporting. Path of store is "/var/opt/eset/esets/cache/reports (base

Linux and BSD installation) or "/var/opt/esets/cache/reports" (Solaris installation)

report_domains = *list of domains*

type: string

default: *list of domains*=""

List of domains(separated by comma) for which the report data should be stored.

report_lifetime = *value*

type: integer

default: 365

Automatically delete entries older than value days. The @logs (Logs maintenance) task in Scheduler executes this action.

samples_enabled = *yes/no*

type: bool

default: no

Enables/disables samples submission system (Charon) cache initialization. Once it is enabled the samples will be collected as a part of scanning process. See section **SAMPLES SUBMISSION SYSTEM** of this manual page for details.

samples_send_target = *list*

type: string

default: *list*="ras:eset"

Defines target of samples submission system delivery. The string argument 'target' is assumed to be of the format "target1:target2:...", where target1, target2, etc. are specified targets. Two target types are supported in current version, i.e. ras - Remote Administration Server and eset - ESET Anti-Virus laboratory.

samples_send_files = *event*

type: string

default: *event*="update"

Defines event when suspected files shall be sent to ESET Anti-virus Laboratory. Three values are possible, i.e. none - never, add - on add of sample to cache, update - on modules update.

samples_send_stats = *event*

type: string

default: *event*="update"

Defines event when statistics regarding suspected files shall be sent to ESET Anti-virus Laboratory. Three values are possible, i.e. none - never, add - on add of sample to cache, update - on modules update.

samples_exclude = *excludes*

type: string

default: *excludes*="*.doc:*.rtf:*.xl?:*.dbf:*.mdb:*.sxw:*.sxc"

This option allows you to exclude certain files/folders from submission. For example, it may be useful to exclude files which may carry confidential information, such as documents or spreadsheets. The most common file types are excluded by default (.doc, etc.). You can add file types to the list of excluded files.

user_mail = *mail*

type: string

default: no default

This option provides an optional information for ESET Anti-Virus laboratory team to contact sample sender if necessary.

av_update_server = *server*

type: string

default: *server*=""

Defines address of ESET server where pre-compiled ESETS modules are downloaded from. If not specified, the autos-elect mode is activated.

av_prerelease_updates = *yes/no*

type: bool

default: no

Enables/disables prerelease updates. This option shouldn't be enabled on systems where mirror is enabled or where av_update_server is set.

av_update_username = *username*

type: string

default: *username*=""

Defines ESETS authentication username.

av_update_password = *password*

type: string

default: *password*=""

Defines ESETS authentication password.

av_update_quarantine_scan = *yes/no*

type: bool

default: no

Enables/disables scan of quarantined files automatically after each virus signature database update.

av_mirror_enabled = *yes/no*

type: bool

default: no

Enables/disables ESETS Anti-Virus mirror creation and maintenance.

av_mirror_pcu = *list*

type: string

default: *list*=""

Defines a list of so called ESETS anti-virus Program Component Update modules to be mirrored from ESET server. The string argument 'list' is assumed to be of the format "pcu1:pcu2:pcu3:...", where pcu1, pcu2, pcu3 etc. are specified modules to be mirrored. Use the WWW Interface to set this option.

proxy_addr = *addr*

type: string

default: *addr*=""

Defines address of the local domain proxy server. If this parameter defined the system access the Internet via proxy defined.

proxy_port = *port*

type: integer

default: no default

Defines port of the local domain proxy server.

proxy_username = *username*

type: string

default: no default

Defines the local domain proxy server authentication username. If this parameter defined the system access the proxy server as with authentication options.

proxy_password = *password*

type: string

default: no default

Defines the local domain proxy authentication password.

as_log_enabled = *yes/no*

type: bool

default: no

Enables/disables diagnostic logging of Antispam.

as_message_scan_size = *size*

type: integer

default: 0

Limit message size (in Bytes), which should be scanned by Antispam. 0 means unlimited.

as_approved_ip_list = *IPs_list*

type: string

default: *IPs_list*=""

List of IP addresses or ranges separated by commas, whose mails should be approved. Ranges can be specified following ways:

IP (e.g. 1.2.3.4)

IP/netmask
(e.g. 1.2.3.4/8)

StartingIP-EndingIP
(e.g. 1.2.3.4-1.2.3.8)

as_blocked_ip_list = *IPs_list*

type: string

default: *IPs_list*=""

List of IP addresses or ranges separated by commas whose mails should be. Ranges can be specified following ways:

IP (e.g. 1.2.3.4)

IP/netmask
(e.g. 1.2.3.4/8)

StartingIP-EndingIP
(e.g. 1.2.3.4-1.2.3.8)

as_ignored_ip_list = *IPs_list*

type: string

default: *IPs_list*=""

List of IP addresses or ranges separated by commas whose mails should be ignored. Ranges can be specified following ways:

IP (e.g. 1.2.3.4)

IP/netmask
(e.g. 1.2.3.4/8)

StartingIP-EndingIP
(e.g. 1.2.3.4-1.2.3.8)

as_blocked_body_domain_list= *domain_list*

type: string

default: *domain_list*=""

List of domains separated by commas, which will be blocked if they appear in mail body.

as_ignored_body_domain_list= *domain_list*

type: string

default: *domain_list*=""

List of domains separated by commas, which will be ignored if they appear in mail body.

as_blocked_body_ip_list= *IPs_list*

type: string

default: *IPs_list*=""

List of IPs separated by commas, which will be blocked if they appear in mail body.

as_ignored_body_ip_list= *IPs_list*

type: string

default: *IPs_list*=""

List of IPs separated by commas, which will be ignored if they appear in mail body.

as_approved_senders = *list of approved domains and users*

type: string

default: *list of approved domains and users*=""

List of senders addresses or senders domains separated by commas, whose mails never will be considered spam.

as_blocked_senders = *list of blocked domains and users*

type: string

default: *list of approved domains and users*=""

List of senders addresses or senders domains separated by commas, whose mails will be considered as spam.

as_approved_domain_to_ip_list = *domain_list*

type: string

default: *list of approved domains*=""

List of domains separated by commas, which are regularly resolved to IP addresses and will be used with approved list as_approved_ip_list.

as_blocked_domain_to_ip_list = *domain_list*

type: string

default: *list of blocked domains*=""

List of domains separated by commas, which are regularly resolved to IP addresses and will be used with blocked list as_blocked_ip_list.

as_ignored_domain_to_ip_list = *domain_list*

type: string

default: *list of blocked domains*=""

List of domains separated by commas, which are regularly resolved to IP addresses and will be used with ignore list as_ignored_ip_list.

as_rbl_service = *service1:response1,servic2:response2_1;response2_2;...*

type: string

default: *as_rbl_service*=""

This option allows to server define realtime blackhole lists. Services are separated by commas and responses are separated by semicolon.

as_rbl_max_ips = *value*

type: integer

default: 0

This option allows to limit the number IP addresses queried against the RBL server. The total number of RBL queries will be the number of IP addresses in the Received: headers (up to a maximum of RBL maxcheck IP addresses) multiplied by the number of RBL servers specified in RBL list. If the value is set to "0" an unlimited number of received headers are checked. IPs on the ignored IP list do not count towards the RBL IP addresses limit.

as_dnsbl_service = *service1:response1,servic2:response2_1;response2_2;...*

type: string

default: *as_dnsbl_service*=""

Defines a list of DNS Blocklist (DNSBL) servers to query with domains and IPs extracted from the message body. Services are separated by commas and responses are separated by semicolon.

as_dnsbl_max_ips = *value*

type: integer

default: 0

This option allows to limit the number of IP addresses queried against the DNS Blocklist server.

as_dnsbl_max_domains = *value*

type: integer

default: 0

This option allows to limit the number of domains queried against the DNS Blocklist server.

racl_server_addr = *address*

type: string

default: no default

Primary Remote Administration Server address.

racl_server_port = *port*

type: integer

default: 2222

Primary Remote Administration Server port.

racl_password = *password*

type: string

default: no default

Password for authentication to primary Remote Administration Server.

racl_secure_enabled = *yes/no*

type: bool

default: yes

If enabled, connection to primary Remote Administration Server with unsecured communication will be prohibited.

racl_alt_server_addr = *address*

type: string

default: no default

Secondary Remote Administration Server address.

racl_alt_server_port = *port*

type: integer

default: 2222

Secondary Remote Administration Server port.

racl_alt_password = *password*

type: string

default: no default

Password for authentication to secondary Remote Administration Server.

racl_alt_secure_enabled = *yes/no*

type: bool

default: yes

If enabled, connection to secondary Remote Administration Server with unsecured communication will be prohibited.

racl_interval = *value*

type: integer

default: 10

Interval between connecting to RA Server (minutes).

disklogs_lifetime = *value*

type: integer

default: 90

Disklog entries older than given age in days will be automatically deleted.

disklogs_optimize_percent = *value*

type: integer

default: 25

Optimize disklogs when the percentage of unused records exceeds given percent.

ESETS MODULE COMMON OPTIONS

To get detailed description of ESETS module common options, please refer to the section **ESETS MODULE COMMON OPTIONS** of the **esets.cfg(5)** manual page.

ESETS SCANNER COMMON OPTIONS

To get detailed description of ESETS scanner common options, please refer to the section **ESETS SCANNER COMMON OPTIONS** of the **esets.cfg(5)** manual page.

COMMAND LINE OPTIONS

This section contains list of command line options valid for this module.

Note that present version of ESETS implements in general three types of command line options parameters:

Integer parameters accept integer values, e.g.:

--integer-parameter 26

For integer parameter it is also possible to append unit K (1K = 1024), M (1M=1024*1024), G (1=1024*1024*1024), e.g.:

--integer-parameter 4K

Boolean (logical) parameters do not accept any arguments. Their appearance in the command line automatically signal enabling of an appropriate functionality related with the parameter, e.g.:

--bool-parameter

Thus in order to provide negation of an appropriate logical parameter prescribe this parameter with --no, e.g.:

--no-bool-parameter

String parameters accept string values. They can be put into quotation marks, e.g.:

--string-parameter "string value"

Besides options on the command line, there are also commands. All ESETS programs accept at least these commands for displaying the version number and a simple help screen.

-v, --version

Print version information to stdout and exit.

-h, --help

Print help screen to stdout and exit.

SPECIAL PRIVATE COMMAND LINE OPTIONS

This section contains list of private **esets_daemon** command line options that do not have an appropriate configuration file options partners.

--cfg-path *path*
type: string

default: *path* = */etc/opt/eset/esets/esets.cfg* (base Linux installation), *path* = */usr/local/etc/esets/esets.cfg* (BSD installation), *path* = */etc/opt/esets/esets.cfg* (Solaris installation).

Defines *path* of the ESETS main configuration file instead of the default.

--report *path*
type: string

default: *path* = ""

Path of template can be found: */etc/opt/eset/esets/scripts/report_template.eml* (base Linux installation), */usr/local/etc/esets/scripts/report_template.eml* (BSD installation), */etc/opt/esets/scripts/report_template.eml* (Solaris installation),

Defines *path* to the template with format strings. Options --report-domain and --report-time-period must be set too and --report-mail is optional. For reporting DOMAIN every month, create scheduler task like:

"0;Report for DOMAIN;;0;0 0 1 * * *;esets_daemon --report=*path* --report-domain=DOMAIN --report-mail=user@domain --report-time-period=1m | sendmail -t -oi"

Possible format strings:

%from_time%
Start date of reporting

%to_time%
End date of reporting

%domain%
Name of domain

%mail%
Mail

%num_clean%
Number of clean mails

%num_clean_percent%
Number of clean mails in percent

%num_threats%
Number of threats mails

%num_threats_percent%
Number of threats mails in percent

%num_not_scanned%
Number of not scanned mails

%num_not_scanned_percent%
Number of not scanned mails in percent

%num_spam%
Number of spam mails

%num_spam_percent%
Number of spam mails in percent

%num_total%
Number of all mails

%num_total_percent%
Number of all mails in percent

%threats_table_line_start%
The template start line which will be replaced by all threats. The line template is used for all threats and %threats_table_line_end% must be defined in template. A string of line temaplate can use formats:

%threats_column_no%
Number of line

%threats_column_date%
Date of occurence

%threats_column_from%
Mail of sender

%threats_column_to%
Mail of recipient

%threats_column_virus_name%
Name of threat

%threats_column_av_status%
Antivirus status of mail

%threats_column_as_status%
Antispam status of mail

%threats_column_action%
Action with mail

%threats_table_line_end%
End of line threats template

--report-domain *domain*
type: string

default: *domain* = ""

Defines *domain* for filling(%domain%) in template.

--report-time-period *time-period*
type: string

default: *time-period* = ""

Defines specific*time period* for filling(%from_time% and %to_time%) in template. Format is: [1..n][d,w,m] or YY/MM/DD-YY/MM/DD. n - number, d - previos day, w - previous week, m - previos mounth, YY - year, MM - month, DD - day"

--report-mail *mail*
type: string

default: *mail* = ""

Defines *mail* for filling(%mail%) in template.

ANTI-VIRUS AND ANTI-SPAM SCANNING

The main purpose of the **esets_daemon** daemon is to perform anti-virus scanning of objects specified by an appropriate agent during TCP/IP communication with the daemon. This functionality can be enabled using parameter *action_av* (to get description of the parameter see esets.cfg(5) manual page).

Besides anti-virus scanning task the **esets_daemon** daemon performs also anti-spam scanning task if enabled. This functionality can be enabled using parameter *action_as* (to get description of the parameter see esets.cfg(5) manual page). Note that anti-spam scanning can be enabled only upon e-mail objects, thus this functionality is relevant only for those agent modules related directly with operation of ESET Mail Security.

It is useful to know that both the anti-virus and anti-spam scanning functionality can be enabled on the per agent (resp. per user specific configuration) basis. This means the anti-virus and anti-spam scanning functionality can be enabled/disabled individually for agent modules as well as for individual user specific configuration used (see section **USER SPECIFIC CONFIGURATION** to get detailed information about user

specific configuration functionality).

Note also that unlike anti-virus scanning engine initialized always during the daemon initialization the anti-spam scanning engine is initialized only in case the parameter *action_as* was set to 'scan' anywhere in the configuration.

LOGGING

The **esets_daemon** daemon logging is provided via syslog facility defined by the parameter *syslog_facility*. From the logging structure point of view we divide logging output into so called summarizing (**summ**) lines, i.e. lines describing whole scanned object and so called particular (**part**) lines describing individual parts of the scanned object. In order to define format of **summ** or **part** log messages use parameter *log_format_summ* or *log_format_part*. Several so called syslog classes can be defined using parameter *syslog_class* to provide various types of logging messages output. Note that each syslog messages class is logged with its predefined priority (see the *syslog_class* parameter description for details).

SAMPLES SUBMISSION SYSTEM

Sample 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 so called samples submission system servers. Every suspected object delivered to the servers is treated by the ESET virus laboratory team and virus signature of the object is added into the virus signatures database, if necessary, to provide effective pro-active protection of the ESETS users against infiltrations.

In order to enable catching of samples, the samples submission system cache has to be initialized. For this purpose parameter **samples_enabled** has to be enabled. In order to enable process of samples delivery it is necessary to set parameter **samples_send_target** to one of the predefined values (eset, ras) and at least one of the parameters **samples_send_files**, **samples_send_stats** to value other than 'none'. To get more information concerned with setting of the sample submission system, please refer to the section **ESETS MODULE PRIVATE OPTIONS** of this manual page.

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.

SCHEDULER

Scheduler runs scheduled tasks when specified time elapses or event happens. Each task has 6 parameters: *id*, *name*, *flags*, *failstart*, *datespec* and *command*. In configuration file, all these parameters and tasks themselves are semicolon-separated. Any other semicolons (and backslashes) must be backslash escaped.

Id is a unique number written as hex. If zero, a new one is generated.

Name is non-unique task description.

Flags contain zero or more colon-separated flags with this meaning:

disabled
: task will be ignored

dontrunonbat
: task will not be run when system is on batteries

Failstart instructs what to do if task could not be run on scheduled date. No value means to run on next scheduled date. Otherwise it will be run if it was last run at least given number of minutes ago (0 means therefore as soon as possible). This parameter has no meaning for tasks triggered by event.

Datespec can be either

- a regular date specification with 6 (crontab like year-extended) fields meaning at which minute, hour, day-of-month, month, year and day-of-week to run the task. Each field can be a special value '*'

(asterisk) meaning 'all' or a comma separated list of numeric values or ranges. The asterisk or a range can be followed by /step, so that e.g. 3-9/2 means 3,5,7,9. Minutes and hours are numbered from 0, days-of-month and months from 1. Days-of-week are 1 for Monday, ..., 6 for Saturday, 0 and 7 for Sunday. If day-of-month and day-of-week are both restricted (not '*'), the task will be run when either field matches.

- the word 'repeat' and a number indicating the period in minutes for repeated task
- an event name with optional one-run interval in minutes meaning:

  start daemon startup

  startonce
  daemon startup but at most once a day

  dialup dial-up connection initiation (not on unix)

  engine successful engine update

  app successful application update (not on unix)

  login GUI startup

  threat threat detected(see **THREAT DETECTION and NOT SCANNED**)

  notscanned
  not scanned mail or file(see **THREAT DETECTION and NOT SCANNED**)

  licexp 30 days before license expiration(see **LICENSE EXPIRATION WARNING NOTIFICATION**)

  localsender
  send e-mail from local network(see **LOCAL SENDER**)

Note that tasks triggered by events 'threat', 'notscanned' and 'licexp' typically execute external script with set some enviroment variables. Script is typically used to send e-mail to system administrator.

Command can be an absolute path to a command followed by its arguments or a special command name meaning:

@update
anti-virus update

@asupdate
anti-spam update

@logs logs maintenance

@uscan
on-demand scan

@uscanr
readonly on-demand scan

@sscan
startup file scan

where '@uscan' and '@uscanr' are followed by a 'profile:include:-exclude' argument with profile name, includes and excludes.

and where '@sscan' is followed by one of these priorities: 'normal', 'lower', 'lowest', 'idle'.

LOCAL SENDER

Notification is provided by **esets_daemon** daemon. Task triggered by event will be executed if sender sent by email match the criteria defined in *mail_domains*. Proper setting of this parameter is individually for every agent module. Setting is the same as setting of client's (i.e. sender's) fully qualified email address or 'c_eml'. (see section **USER SPECIFIC CONFIGURATION** part of documentation of agent.

If task defines an using external script all the relevant information concerned with the scanned e-mail is

passed to the script process image via environment variables. The following variables are provided by **esets_daemon**:

ESETS_USERSPEC
User specification ID.

ESETS_SENDER
Fully qualified address of the e-mail message sender.

ESETS_RECIPIENT
Fully qualified address of the e-mail message recipient.

ESETS_RECIPIENT_DOMAIN
One of 'local', 'remote' of 'unknown'. Depends on whether recipient match *mail_domains* too.

THREAT DETECTION AND NOT SCANNED

The ESETS notification is provided by **esets_daemon** daemon via executing of the external script **daemon_notification_script** stored in scripts subdirectory of ESETS configuration directory. The action, eventually the file name and path of the external script can be redefined via **esets_daemon** private option *scheduler_tasks*. To get more information on this topic refer to the section**SCHEDULER** of this manual page.

If task is enabled, an external script is executed by **esets_daemon** daemon under condition that the scanned e-mail or file message is not clean. All the relevant information concerned with the scanned object is passed to the script process image via environment variables. The following variables are provided by **esets_daemon**:

ESETS_USERSPEC
User specification ID.

ESETS_MSGID
ID of the e-mail message.

ESETS_SENDER
Fully qualified address of the e-mail message sender. **It is highly recommended to not use this information to send notification to the sender of scanned e-mail, since the sender address of the infected e-mail could be modified by the infection and therefore will result in a SPAM effects.**

ESETS_RECIPIENT
Fully qualified address of the e-mail message recipient. **It is highly recommended to not use this information to send notification to the recipient of scanned e-mail, since the recipient address of the infected e-mail could be modified by the infection and therefore will result in a SPAM effects.**

ESETS_AV_STATUS
Status of AV scanning process.

ESETS_AS_STATUS
Status of AS scanning process.

ESETS_ACTION
Action taken as a consequence of scanning status.

ESETS_LOG
Short report concerned with the scanned e-mail message infection.

ESETS_VIRUS
The virus name.

These environment variables can be used to build the notification e-mail message. For further details refer to the example script **daemon_notification_script** stored in the ESETS configuration directory. The example script provides system administrator with short e-mail notification. **Note that it is highly recommended to not use the scenario of the sender and/or recipient notification due to a possibly modified**

sender and/or recipient addresses of the infected e-mail which could result in a SPAM effects.

LICENSE EXPIRATION WARNING NOTIFICATION

During the daemon initialization and also while it is running, the license validity related with the product is checked. A warning notification functionality can be enabled by proper setting of parameter *scheduler_tasks* that provides system administrator with appropriate logging output and also email notification once time to license expiration is less than 30 days and/or the license expired already. In this case a short message is written into the daemon logs each time the main manager process is initialized and choosen action is done. Default e-mail message with the license status is sent using external script to system administrator once per day. Path of external script is */etc/opt/eset/esets/scripts/license_warning_script*" (base Linux installation), */usr/local/etc/esets/scripts/license_warning_script*" (BSD installation), */etc/opt/esets/scripts/license_warning_script*" (Solaris installation). Path of external script or default action can be change using *scheduler_tasks*. To get more information on this topic refer to the section **SCHEDULER** of this manual page.

REPORTING BUGS

In order to report bugs, please visit http://www.eset.com/support <URL:http://www.eset.com/support> or use directly the support form at http://www.eset.eu/support/form <URL:http://www.eset.eu/support/form>.

COPYRIGHT

SEE ALSO

esets_cgp(1) esets_cli(1) esets_dac(1) esets_ftp(1) esets_gwia(1) esets_http(1) esets_icap(1) esets_imap(1) esets_mda(1) esets_mird(1) libesets_pac.so(1) esets_pipe(1) esets_pop3(1) esets_smfi(1) esets_smtp(1) esets_ssfi.so(1) esets_wwwi(1) esets_zmfi(1) esets(5) esets.cfg(5) esets_daemon(8) esets_inst(8) esets_lic(8) esets_quar(8) esets_scan(8) esets_set(8) esets_update(8) eset_efs_userguide.pdf eset_egs_userguide.pdf eset_ems_userguide.pdf