NAME

esets_dac − ESETS Dazuko powered file Access Control module.

SYNOPSIS

esets_dac [**OPTIONS**]

DESCRIPTION

The **esets_dac** is so called on-access control module, i.e. module that controls file system objects before they are accessed by user and/or operating system. There are two different projects - old Dazuko2.x and new DazukoFS, which is meant to replace the old project. To keep access control over the file system **esets_dac** uses so called Dazuko kernel module (see also www.dazuko.org <URL:www.dazuko.org>).

Scanning of file system objects is performed upon customizable file access event. The following events are supported (DazukoFS supports only open event):

open This file access event is triggered when a file is opened.

create This file access event is triggered when a file opened for write is closed.

exec This file access event is triggered when a file is executed.

By using this mechanism all opened, created and executed regular files are scanned by **esets_daemon** for viruses. Based on the scanning result the access to the files is allowed or denied.

SIGNALS AND DAEMON PROCESS STRUCTURE

In order to start **esets_dac**, the agent manager process must be enabled using parameter **agent_enabled** in section [dac] of the main ESETS configuration file. The manager process is responsible for the agent initialization and also for the maintenance of **num_proc** so called on-access control processes each running **num_thrd** on-access control threads. Note that each thread performs control sessions until termination, resp. until hard termination of the agent processes.

TERM This signal causes termination of the agent processes, i.e. after **esets_dac** manager process obtained this signal it terminates all daughter threads and processes as soon as possible and terminates.

USR1 This signal causes so called hard termination of the agent processes, i.e. after **esets_dac** manager process obtained this signal it terminates immediately all daughter threads and processes and terminates.

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

agent_enabled = *yes/no*

type: bool

default: no

Enables/disables operation of **esets_dac** daemon manager process.

num_proc = *value*

type: integer

default: *value* = 1

Defines number of **esets_dac** daemon processes.

num_thrd = *value*

type: integer

default: *value* = 2

Defines number of **esets_dac** daemon threads per process.

ctl_incl = *list*

type: string

default: *list* = ""

Defines list of directories to be under control of scanner. The string argument 'list' is assumed to be of the format "dir1:dir2:dir3:...", where dir1, dir2, dir3 etc. are specified directories. This option is required and there is no default. Note that DazukoFS requires mount right directories (see DazukoFS documentation).

event_mask = *mask*

type: string

default: *mask* = ""

Defines file system object events user wishes to guard. The string argument is assumed to be of the format "event1:event2:...", where event1, event2 etc. are one of: *open*, *create* or *exec*.

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.

ESETS ON-ACCESS SCANNER COMMON OPTIONS

Due to the on-access scanner optimization reason, the following options have been introduced to redefine default values of those described by section **ESETS SCANNER COMMON OPTIONS**. Note that these common options are used only by on-access scanner agent modules, e.g. **libesets_pac.so**, **esets_dac** etc.

av_scan_obj_archives = *yes/no*

type: bool

default: no

Enables/disables scanning of archives (.ZIP, .RAR, .ARJ, etc.).

av_scan_obj_mime = *yes/no*

type: bool

default: no

Enables/disables scanning of MIME archives, i.e. e-mail messages in raw format.

av_scan_obj_mailbox = *yes/no*

type: bool

default: no

Enables/disables scanning of various mailboxes.

av_scan_obj_rtp = *yes/no*

type: bool

default: no

Enables/disables scanning of runtime-packers.

av_scan_obj_sfx = *yes/no*

type: bool

default: no

Enables/disables scanning of self-extracting archives.

av_scan_adv_heur = *yes/no*

type: bool

default: no

Enables/disables use of advanced heuristics method while scanning.

av_exec_scan_adv_heur = *yes/no*

type: bool

default: no

Enables/disables use of advanced heuristics method for scanned regular file in case it is catched by on-access scanner within 'ON_EXEC' event.

av_create_scan_adv_heur = *yes/no*

type: bool

default: yes

Enables/disables use of advanced heuristics method for scanned regular file in case it is catched by on-access scanner within 'ON_CREATE' event.

av_create_scan_obj_rtp = *yes/no*

type: bool

default: yes

Enables/disables scanning of runtime-packers in case they are catched by on-access scanner within 'ON_CREATE' event.

av_create_scan_obj_sfx = *yes/no*

type: bool

default: yes

Enables/disables scanning of self-extracting archives in case they are catched by on-access scanner within 'ON_CREATE' event.

av_create_scan_def_arch = *yes/no*

type: bool

default: yes

Enables/disables use of AV scanner limit parameters 'av_create_scan_archive_max_level' and 'av_create_scan_archive_max_size'. If enabled, it effectively redefines default values of parameters 'av_scan_archive_max_size' and 'av_scan_archive_max_level' described in **ESETS SCANNER COMMON OPTIONS** in case the scanned file was catched by on-access scanner within 'ON_CREATE' event.

av_create_scan_archive_max_level = *lvl*

type: integer

default: *lvl* = 10

Specifies the maximum level *lvl* an archive is descended, unpacked and scanned in case the scanned file was catched by on-access scanner within 'ON_CREATE' event. When a scan is terminated prematurely because this limit was reached, the scanned object is considered as not scanned.

Note that option 'av_create_scan_def_arch' must be enabled in order to use this option.

av_create_scan_archive_max_size = *size*

type: integer

default: *size* = 0

Specifies the maximum unpacked size*size* (measured in bytes) of a file from archive, which will be scanned in case the scanned file was catched by on-access scanner within 'ON_CREATE' event. Zero (0) means no limit. When a scan is terminated prematurely because this limit was reached, the scanned object is considered as not scanned.

Note that option 'av_create_scan_def_arch' must be enabled in order to use this option.

USER SPECIFIC CONFIGURATION

The ESETS system implements possibility to define so called user specific configuration, i.e. relevant configuration parameters specific for users accessing file system objects can be defined.

As described in section **USER SPECIFIC CONFIGURATION** of esets.cfg(5) manual page the user specific configuration is created when an appropriate special configuration section created within a special configuration file *path* referenced from this agent section (see main ESETS configuration file) by using option **user_config =** *path*.

The header name of user specific section must be the 'username' of a user for which we want to define special configuration.

Thus the user specific section can be for instance as follows.

```
[username]
        av_scan_obj_archives = no
```

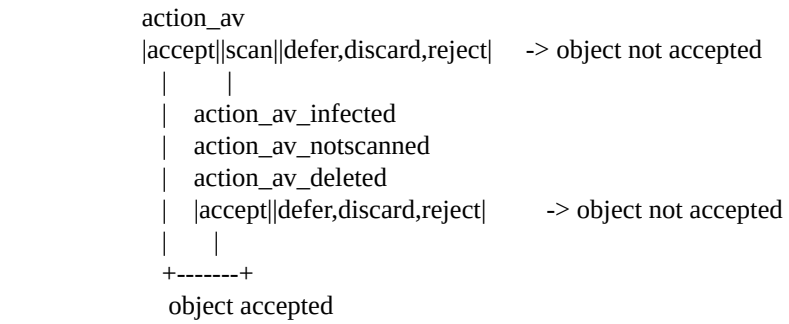
Once user specific configuration defined, it will be used automatically, as this agent automatically provides main ESETS scanning daemon **esets_daemon** with the user ID information during the session.

When information, i.e. user name has been passed to the daemon, an appropriate configuration is selected with respect to the following algorithm:

1. If user name matches header name of the section, the appropriate configuration is chosen for scanning and selection process is finished.
2. If configuration is not chosen yet the configuration appropriate to the agent section from main ESETS configuration file is chosen and selection is finished.

HANDLE OBJECT POLICY

The Handle Object Policy (see figure below) is a mechanism that provides handling of the scanned objects depending on their scanning status. The mechanism implemented in this module is based on so called action configuration options **action_av**, **action_av_infected**, **action_av_notscanned** and **action_av_deleted**. To get description of these configuration options, see esets.cfg(5) manual page.



Every object processed by this module is first handled with respect to the setting of the configuration option **action_av**. Once the option is set to 'accept' (resp. 'defer', 'discard', 'reject') the object is accepted (resp. deferred, discarded, rejected). If the option is set to 'scan' the object is scanned (resp. also cleaned if requested by configuration option **av_clean_mode**) for virus infiltrations and set of action configuration options **action_av_infected**, **action_av_notscanned** and **action_av_deleted** is taken into account to evaluate further handling of the object. If action 'accept' has been taken as a result of the above action options the object is accepted, i.e. the access to the object is allowed. On the other hand if any of action configuration options caused other than 'accept' value, the object is blocked, i.e. access to the object is denied.

You have probably noticed that each of the action configuration options discussed above accepts a variety of the values whose list can be found in esets.cfg(5) manual page. As also stated there the values listed are handled individually by every ESETS agent module. Thus to be consistent in the following we review the meaning of the values for this ESETS agent module.

accept Accept object on this level of Handle Object Policy, i.e. access to the object is allowed by the particular action configuration option.

scan Scan object for virus infiltrations and clean infected objects if requested by configuration option **av_clean_mode**.

defer, discard, reject
Block access to the object, i.e. the access to the object is denied by this particular action configuration option.

LOGGING

Logging functionality of the ESETS agent modules has been developed to fine tune or to troubleshoot the agent module performance. Thus all the ESETS agent modules support only logging using syslogd daemon which logs system messages on *nix systems. To get more information on this topic please, refer to manual pages syslog(2), syslog.conf(5) and syslogd(8).

Regarding ESETS agent modules, this functionality can be invoked by setting ESETS module common configuration option *syslog_facility* to value other than **none**. To get description of the introduced ESETS module common configuration options please, refer to esets.cfg(5) manual page.

Once the syslog logging enabled, the ESETS agent module messages are logged with one of the following syslog priorities:

LOG_ERR
Error messages concerned with the ESETS agent module performance are logged with this priority. Message logged with this priority usually means that error occurred during the ESETS agent module running and thus the module could not accomplish its operation or even the module process exited.

LOG_WARNING
Warning messages concerned with the ESETS agent module performance are logged or 'summary' messages concerned with action other than 'accept' taken as a consequence of object scanning status are logged with this priority.

LOG_NOTICE
The 'summary' messages concerned with action taken as a consequence of object scanning status are logged with this priority.

LOG_INFO
The common tasks are logged with this priority.

LOG_DEBUG
Debug information concerned with the ESETS agent module performance is logged with this priority.

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.

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