NAME

libesets_pac.so − ESETS Preload library based file Access Control module.

SYNOPSIS

LD_PRELOAD=/path/to/libesets_pac.so command arguments...

where **/path/to/libesets_pac.so** is absolute path to the ESETS Preload library **libesets_pac.so** and **command** with arguments means executable command of an appropriate file system server with its command line arguments.

DESCRIPTION

The **libesets_pac.so** is a shared objects library used as a preload library of LIBC. It is thus applicable for file system servers implemented using LIBC calls, for instance ftp server, Samba server etc.

Scanning of file system objects is performed upon customizable file access event. The following events are supported:

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 regular files that **command** opens, creates and executes are scanned by **esets_daemon** for viruses. Based on the scanning result the access to the files is allowed or denied.

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

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. Note that this option is required and there is no default.

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:

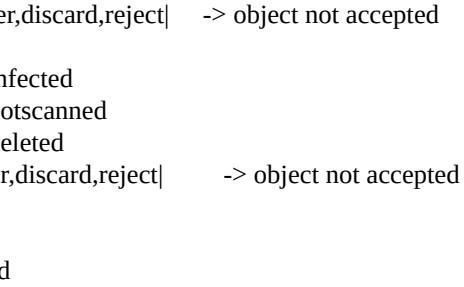
1. If user name matches header name of the section, the appropriate configuration is chosen for scanning and selection process is finished.
2. If configuration is not chosen yet the configuration appropriate to the agent section from main ESETS configuration file is chosen and selection is finished.

COMMAND LINE OPTIONS

This module does not implement command line interface.

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.

NOTES

This module preloads all libc's file open, create and exec functions. Although it does not preload e.g. system(3) or popen(3), the shell which is executed by them inherits the LD_PRELOAD environment variable, thus all executions performed by shell are scanned, too.

By no way it intercepts any real system call. Open/create/exec system calls performed by other means (e.g. direct assembly) are thus not checked. The purpose of this module is to protect trustworthy applications.

On 64-bit systems, both 64-bit and 32-bit preload libraries are provided in corresponding library directories.

TIPS

You can export the LD_PRELOAD environment variable also in shell or scripts. Just note, that libesets_pac will be loaded only by (and check file accesses of) commands started inside.

You can also install it systemwide in */etc/ld.so.preload*. For more information see ld.so(8).

In upstart initscripts you have to change the exec line to: exec env LD_PRELOAD=/path/to/libesets_pac.so command arguments...

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

Developed by ESET, spol. s r.o. 2011 (C). www.eset.com <URL:www.eset.com>

Developed with ProWeb Consulting. www.pwc.sk <URL:www.pwc.sk>

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