System Center Endpoint Protection

Installation Manual and User Guide

Red Hat Enterprise Linux Server 5, 6
SUSE Linux Enterprise 10, 11
CentOS 5, 6
Debian Linux 5, 6
Ubuntu Linux 10.04, 12.04
Oracle Linux 5, 6

Microsoft

Microsoft System Center Endpoint Protection

Contents

Introduction 3

Main functionality 3

Key features of the system 3

Terminology and abbreviations 5

Installation 6

Architecture Overview 7

Integration with File System services 8

On-demand scanner 8

Real-time protection using preload LIBC library 8

Operation principle 8

Installation and configuration 9

Tips 9

Real-time protection powered by Dazuko 9

Operation principle 9

Installation and configuration 10

Tips 10

Important SCEP mechanisms 11

Handle Object Policy 11

User Specific Configuration 11

Scheduler 12

Web Interface 12

Real-time protection configuration example 13

On-Demand scanner 14

Scheduler 15

Statistics 15

Logging 16

Command-line scripts 16

SCEP Security system update 17

SCEP update utility 17

SCEP update process description 17

Let us know 18

Appendix A. SCEP setup and configuration 19

Setting SCEP $PATH environment variable 19

Setting SCEP for Samba (Ubuntu upstart) 19

Appendix B. PHP License 20

Introduction

Thank you for using System Center Endpoint Protection. Microsoft's state-of-the-art scanning engine has unsurpassed scanning speed and detection rates combined with a very small footprint that makes it the ideal choice for any Linux OS server.

Main functionality

On-demand scanner

The On-demand scanner can be started by a privileged user (usually a system administrator) through the command line interface, the web interface or by the operating system's automatic scheduling tool (e.g., cron). The term *On-demand* refers to file system objects being scanned by either user or system demand.

Real-time protection

The Real-time protection is invoked whenever a user and/or operating system attempts to access file system objects. This also clarifies the use of the term *On-access*; because a scan is triggered by any attempt to access file system objects.

Key features of the system

Advanced engine algorithms

The Microsoft antivirus scanning engine algorithms provide the highest detection rate and the fastest scanning times.

Multi-processing

System Center Endpoint Protection is developed to run on single- as well as multi-processor units.

Advanced Heuristics

System Center Endpoint Protection includes unique advanced heuristics for Win32 worms, backdoor infections and other forms of malware.

Built-In features

Built-in archivers unpack archived objects without requiring any external programs.

Speed and efficiency

To increase the speed and efficiency of the system, System Center Endpoint Protection's architecture is based on the running daemon (resident program) where all scanning requests are sent.

Enhanced security

All executive daemons (except scep_dac) run under a non-privileged user account to enhance security.

Selective configuration

The system supports selective configuration based on the user or client/server.

Multiple logging levels

Multiple logging levels can be configured to get information about system activity and infiltrations.

Web interface

Configuration and administration are offered through an intuitive and user-friendly web interface.

No external libraries

The System Center Endpoint Protection installation does not require external libraries or programs except for LIBC.

User-specified notification

The system can be configured to notify specific users in the event of a detected infiltration or other important events.

Low system requirements

To run efficiently, System Center Endpoint Protection requires just 250MB of hard-disk space and 256MB of RAM. It runs smoothly under the 2.6.x Linux OS kernel versions.

Performance and scalability

From lower-powered, small office servers to enterprise-class ISP servers with thousands of users, System Center Endpoint Protection delivers the performance and scalability you expect from a UNIX based solution, in addition to the unequaled security of Microsoft security products.

Terminology and abbreviations

In this section, we will review the terms and abbreviations used in this document. Note that a boldface font is reserved for product component names and also for newly defined terms and abbreviations. Terms and abbreviations defined in this chapter are expanded upon later in this document.

SCEP

SCEP is a standard acronym for security product developed by Microsoft for Linux operating systems. It is also the name of the software package containing the products.

SCEP daemon

The main SCEP system control and scanning daemon: *scep_daemon*.

SCEP base directory

The directory where SCEP loadable modules containing the virus signature database are stored. The abbreviation *@BASEDIR@* will be used for future references to this directory. The *@BASEDIR@* value (depending on the operating system) is listed below:

```
Linux: /var/opt/microsoft/scep/lib
```

SCEP configuration directory

The directory where all files related to the System Center Endpoint Protection configuration are stored. The abbreviation *@ETCDIR@* will be used for future references to this directory. The *@ETCDIR@* value (depending on the operating system) is listed below:

```
Linux: /etc/opt/microsoft/scep
```

SCEP configuration file

Main System Center Endpoint Protection configuration file. The absolute path of the file is as follows:

@ETCDIR@/scep.cfg

SCEP binary files directory

The directory where the relevant System Center Endpoint Protection binary files are stored. The abbreviation *@BINDIR@* will be used for future references to this directory. The *@BINDIR@* value (depending on the operating system) is listed below:

```
Linux: /opt/microsoft/scep/bin
```

SCEP system binary files directory

The directory where the relevant System Center Endpoint Protection system binary files are stored. The abbreviation *@SBINDIR@* will be used for future references to this directory. The *@SBINDIR@* value (depending on the operating system) is listed below:

```
Linux: /opt/microsoft/scep/sbin
```

SCEP object files directory

The directory where the relevant System Center Endpoint Protection object files and libraries are stored. The abbreviation *@LIBDIR@* will be used for future references to this directory. The *@LIBDIR@* value (depending on the operating system) is listed below:

```
Linux: /opt/microsoft/scep/lib
```

Installation

System Center Endpoint Protection is distributed as a binary file:

```
scep.i386.ext.bin
```

In the binary file shown above, *'ext'* is a Linux OS distribution dependent suffix, i.e., 'deb' for Debian, 'rpm' for RedHat and SuSE, 'tgz' for other Linux OS distributions.

To install or upgrade the product, use the following command:

```
sh ./scep.i386.ext.bin
```

to display the product's User License Acceptance Agreement. 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 upgrade is displayed onscreen.

Once the package is installed, you can verify that the main SCEP service is running by using the following command:

```
ps -C scep_daemon
```

After pressing ENTER, you should see the following (or similar) message:

```
 PID TTY          TIME CMD
2226 ?        00:00:00 scep_daemon
2229 ?        00:00:00 scep_daemon
```

At least two SCEP daemon processes are running in the background. The first PID represents the process and threads manager of the system. The other represents the SCEP scanning process.

Installing a Language pack

In order to install the required language pack for System Center Endpoint Protection, use the following command:

```
sh ./scep-lang.lng.bin
```

where *'lng'* needs to be replaced by the language code of the file you want to import.

After the *Installation completed successfully* notification displays, update the LANG system variable accordingly and update the environment if necessary. This concludes the language pack installation.

Each language pack contains the following:

- Localized Web Interface
- Localized console outputs of SCEP agents and commands
- Localized PDF Documentation

Architecture Overview

Once System Center Endpoint Protection is successfully installed, you should become familiar with its architecture.

The system is comprised of the following parts:

CORE

The core of System Center Endpoint Protection is the SCEP daemon (scep_daemon). The daemon uses SCEP API library libscep.so and SCEP loading modules em00X_xx.dat to provide base system tasks such as scanning, maintenance of the agent daemon processes, maintenance of the samples submission system, logging, notification, etc. Please refer to the *scep_daemon(8)* man page for details.

AGENTS

The purpose of SCEP agent modules is to integrate SCEP with the Linux server environment.

UTILITIES

The utility modules provide simple and effective system management. They are responsible for system tasks such as quarantine management, system setup and update.

CONFIGURATION

Proper configuration is the most important aspect of a your security system; the remainder of this chapter is dedicated to explaining all related components. A thorough understanding of the *scep.cfg* file is also highly recommended, as this file contains information essential to the configuration of System Center Endpoint Protection.

After the product is successfully installed, all its configuration components are stored in the SCEP configuration directory. The directory consists of the following files:

@ETCDIR@/scep.cfg

This is the most important configuration file, as it controls all major aspects of the product's functionality. The scep.cfg file is made up of several sections, each of which contains various parameters. The file contains one global and several "agent" sections, with all section names enclosed in square brackets. Parameters in the global section are used to define configuration options for the SCEP daemon as well as default values for the SCEP scanning engine configuration. Parameters in agent sections are used to define configuration options of modules used to intercept various data flow types in the computer and/or its neighborhood, and prepare it for scanning. Note that in addition to the various parameters used for system configuration, there are also rules governing the organization of the file. For detailed information on the most effective way to organize this file, please refer to the *scep.cfg(5)* and *scep_daemon(8)* man pages, as well as relevant agents' man page.

@ETCDIR@/certs

This directory is used to store the certificates used by the SCEP web interface for authentication. Please see the *scep_wwwi(8)* man page for details.

@ETCDIR@/scripts/daemon_notification_script

If enabled by the SCEP configuration file parameter *'exec_script'*, this script is executed in the event of a detected infiltration by the antivirus system. It is used to send email notification about the event to the system administrator.

Integration with File System services

This chapter describes the On-demand and Real-time protection configuration which will provide the most effective protection from virus and worm file system infections. System Center Endpoint Protection's scanning power is derived from the On-demand scanner command *'scep_scan'* and the On-access scanner command *'scep_dac'*. The Linux version of System Center Endpoint Protection offers an additional On-access scanner technique which uses the preloaded library module *libscep_pac.so*. All of these commands are described in the following sections.

On-demand scanner

The On-demand scanner can be started by a privileged user (usually a system administrator) through the command line interface, web interface or by the operating system's automatic scheduling tool (e.g., cron). The term *On-demand* refers to file system objects which are scanned on user or system demand.

The On-demand scanner does not require special configuration in order to run. After the SCEP package has been properly installed, the On-demand scanner can be run immediately using the command line interface or the Scheduler tool. To run the On-demand scanner from the command line, use the following syntax:

```
@SBINDIR@/scep_scan [option(s)] FILES
```

where FILES is a list of directories and/or files to be scanned.

Multiple command line options are available using SCEP On-demand scanner. To see the full list of options, please see the *scep_scan(8)* man page.

Real-time protection using preload LIBC library

The Real-time protection is invoked by user(s) access and/or operating system access to file system objects. This also explains the term *On-access*; the scanner is triggered on any attempt to access a selected file system object.

In the following sections, we will also describe the integration of the Real-time protection powered by Dazuko with Linux/BSD file system services. Using Dazuko may not be feasible in all situations, including system administrators who maintain critical systems where:

- the source code and/or configuration files related to the running kernel are not available,
- the kernel is more monolithic than modular,
- the Dazuko module simply does not support the given OS.

In any of these cases, the On-access scanning technique based on the preload LIBC library should be used. See the following topics in this section for detailed information. Please note that this section is relevant only for Linux OS users and contains information regarding the operation, installation and configuration of the On-access scanner using the preload library *'libscep_pac.so'*.

Operation principle

The Real-time protection *libscep_pac.so* (SCEP Preload library based file Access Controller) is a shared objects library which is activated at system start up. This library is used for LIBC calls by file system servers such as FTP server, Samba server etc. Every file system object is scanned based on customizable file access event types. The following event types are supported by the current version:

Open events

This file access type is activated if the word *'open'* is present in the *'event_mask'* parameter in the esest.cfg file (**[fac]** section).

Create (close) events

This file access type is activated if the word *'close'* is present in the *'event_mask'* parameter in the scep.cfg file (**[fac]** section). In this case, all file descriptor and FILE stream close functions of the LIBC are intercepted.

Exec events

This file access type is activated if the word *'exec'* is present in the *'event_mask'* parameter in the scep.cfg (**[fac]** section). In this case, all exec functions of the LIBC are intercepted.

All opened, closed and executed files are scanned by the SCEP daemon for viruses. Based on the result of such scans, access to given files is denied or allowed.

Installation and configuration

The *libscep_pac.so* library module is installed using a standard installation mechanism of the preloaded libraries. You need to define the environment variable *'LD_PRELOAD'* with the absolute path to the *libscep_pac.so* library. For more information, please refer to the *ld.so(8)* man page.

Note: It is important that the *'LD_PRELOAD'* environment variable is defined only for the network server daemon processes (ftp, Samba, etc.) that will be under control of the Real-time protection. Generally, preloading LIBC calls for all operating system processes is not recommended, as this can dramatically slow the performance of the system or even cause the system to hang. In this sense, the '/etc/ld.so.preload' file should not be used, nor should the 'LD_PRELOAD' environment variable be exported globally. Both would override all relevant LIBC calls, which could lead to system hang ups during initialization.

To ensure that only relevant file access calls within a given file system are intercepted, executable statements can be overridden using the following line:

```
LD_PRELOAD=@LIBDIR@/libscep_pac.so COMMAND COMMAND-ARGUMENTS
```

where 'COMMAND COMMAND-ARGUMENTS' is the original executable statement.

Review and edit the **[global]** and **[fac]** sections of the SCEP configuration file (scep.cfg). In order for the On-access scanner to function correctly, you must define the file system objects (i.e. directories and files) that are required to be under control of the preload library. This can be achieved by defining the parameters of the *'ctl_incl'* and *'ctl_excl'* options in the **[fac]** section of the SCEP configuration file. After making changes to the scep.cfg file, you can force the newly created configuration to be re-read by reloading the SCEP daemon.

Tips

In order to activate the Real-time protection immediately after file system start up, the *'LD_PRELOAD'* environment variable must be defined within the appropriate network file server initialization script.

Example: Let's assume we want to have the On-access scanner to monitor all file system access events immediately after starting the Samba server. Within the Samba daemon initialization script (/etc/init.d/smb), we would replace the statement

```
daemon /usr/sbin/smbd $SMBDOPTIONS
```

with the following line:

```
LD_PRELOAD=@LIBDIR@/libscep_pac.so daemon /usr/sbin/smbd $SMBDOPTIONS
```

In this way, selected file system objects controlled by Samba will be scanned at system start-up.

Real-time protection powered by Dazuko

The technique used by SCEP On-access scanner is powered by the Dazuko (da-tzu-ko) kernel module and is based on the interception of kernel calls. The Dazuko project is open source, which means that its source code is freely distributed. This allows users to compile the kernel module for their own custom kernels. Note that the Dazuko kernel module is not a part of any SCEP product and must be compiled and installed into the kernel prior to using the On-access command *scep_dac*. The Dazuko technique makes On-access scanning independent from the file system type used. It is also suitable for scanning of file system objects via Network File System (NFS), Nettalk and Samba.

Important: Before we provide detailed information related to On-access scanner configuration and use, it should be noted that the scanner has been primarily developed and tested to protect externally mounted file systems. If there are multiple file systems that are not externally mounted, you will need to exclude them from file access control in order to prevent system hang ups. An example of a typical directory to exclude is the *'/dev'* directory and any directories used by SCEP.

Operation principle

The Real-time protection *scep_dac* (SCEP Dazuko-powered file Access Controller) is a resident program which provides continuous monitoring and control over the file system. Every file system object is scanned based on customizable file access event types. The following event types are supported by the current version:

Open events

To activate this file access type, set the value of the *'event_mask'* parameter to open in the **[fac]** section of the scep.cfg file. This will enable the ON_OPEN bit of the Dazuko access mask.

Create (close) events

To activate this file access type, set the value of the *'event_mask'* parameter to close in the **[fac]** section of the scep.cfg file. This will enable the ON_OPEN bit of the Dazuko access mask. This will enable the ON_CLOSE and ON_CLOSE_MODIFIED bits of the Dazuko access mask.

Note: Some OS kernel versions do not support the interception of ON_CLOSE events. In these cases, close events will not be monitored by *scep_dac*.

Exec events

To activate this file access type, set the value of the *'event_mask'* parameter to exec in the **[fac]** section of the scep.cfg file. This will enable the ON_EXEC bit of the Dazuko access mask.

The Real-time protection ensures that all opened, closed and executed files are first scanned by the *scep_daemon* for viruses. Depending on the scan results, access to specific files is denied or allowed.

Installation and configuration

The Dazuko kernel module must be compiled and installed within the running kernel before initializing *scep_dac*. For details on how to compile and install Dazuko, please see:

http://www.dazuko.org

Once Dazuko is installed, review and edit the **[global]** and **[fac]** sections of the SCEP configuration file (scep.cfg). Note that for the Real-time protection properly function, it is dependent upon configuration of the *'agent_type'* option within the **[fac]** section of this file. Additionally, you must define the file system objects (i.e. directories and files) that are to be monitored by the Real-time protection. This can be accomplished by defining the parameters of the *'ctl_incl'* and *'ctl_excl'* options, which are also located within the **[fac]** section. After making changes to the scep.cfg file, you can force the newly created configuration to be re-read by reloading the SCEP daemon.

Tips

To ensure that the Dazuko module loads prior to initialization of the *scep_dac* daemon, follow these steps:

Place a copy of the Dazuko module in either of the following directories reserved for kernel modules:

```
/lib/modules
```

or

```
/modules
```

Use the kernel utilities 'depmod' and 'modprobe' (For BSD OS, use 'kldconfig' and 'kldload') to handle dependencies and successfully initialize the newly added Dazuko module.

In the scep_daemon initialization script '/etc/init.d/scep_daemon', insert the following line before the daemon initialization statement:

```
/sbin/modprobe dazuko
```

For BSD OS's the line

```
/sbin/kldconfig dazuko
```

must be inserted into the '/usr/local/etc/rc.d/scep_daemon.sh' script.

Warning! It is extremely important that these steps are executed in the exact order given. If the kernel module is not located within the kernel modules directory it will not properly load, causing the system to hang.

Important SCEP mechanisms

Handle Object Policy

The Handle Object Policy mechanism provides filtering for scanned objects based on their status. This functionality is based on the following configuration options:

- action_av
- action_av_infected
- action_av_notscanned
- action_av_deleted

For detailed information on these options, please refer to the *scep.cfg(5)* man page.

Every processed object is first handled according to the configuration of the *'action_av'* option. If this option is set to *'accept'* (or *'defer'*, *'discard'*, *'reject'*) the object is accepted (or deferred, discarded, rejected). If the option is set to *'scan'* the object is scanned for virus infiltrations, and if the *'av_clean_mode'* option is set to *'yes'*, the object is also cleaned. In addition, the configuration options *'action_av_infected'*, *'action_av_notscanned'* and *'action_av_deleted'* are taken into account to further evaluate object handling. If an *'accept'* action has been taken as a result of these three action options, the object is accepted. Otherwise, the object is blocked.

User Specific Configuration

The purpose of the User Specific Configuration mechanism is to provide a higher degree of customization and functionality. It allows the system administrator to define SCEP antivirus scanner parameters based on the user who is accessing file system objects.

A detailed description of this functionality can be found in the *scep.cfg(5)* man page. In this section we will provide only a short example of a user-specific configuration.

In this example, the goal is to use the *scep_dac* module to control the ON_OPEN and ON_EXEC access events for an external disc mounted under the /home directory. The module can be configured in the **[fac]** section of the SCEP configuration file. See below:

```
[fac]
agent_type = "dazuko"
event_mask = "open"
ctl_incl = "/home"
action_av = "scan"
```

To specify scan settings for an individual user, the *'user_config'* parameter must specify the special configuration filename where the individual scanning rules will be stored. In the example shown here, the special configuration file is called *'scep_dac_spec.cfg'* and is located within the SCEP configuration directory (This directory is based on your operating system. Please see Terminology and abbreviations page).

```
[fac]
agent_type = "dazuko"
event_mask = "open"
ctl_incl = "/home"
action_av = "scan"
user_config = "scep_dac_spec.cfg"
```

Once the *'user_config'* file parameter is specified within the **[fac]** section, the *'scep_dac_spec.cfg'* file must be created in the SCEP configuration directory. Finally, add the desired scanning rules.

```
[username]
action_av = "reject"
```

At the top of the special section, enter the username to which the individual rules will be applied. This configuration will allow all other users attempting to access the file-system to be processed normally. i.e., all file system objects accessed by other users will be scanned for infiltrations, except for the user *'username'*, whose access will be rejected (blocked).

Scheduler

The Scheduler's functionality includes running scheduled tasks at a specified time or on a specific event, managing and launching tasks with predefined configuration and properties and more. Task configuration and properties can be used to influence launch dates and times, but also to expand the application of tasks by introducing the use of custom profiles during task execution.

The *'scheduler_tasks'* option is commented by default, causing the default scheduler configuration to be applied. In the SCEP configuration file all parameters and tasks are semicolon-separated. Any other semicolons (and backslashes) must be backslash escaped. Each task has 6 parameters and the syntax is as follows:

- id – Unique number.
- name – Task description.
- flags – Special flags to disable the specified scheduler task can be set here.
- failstart – Instructs what to do if task could not be run on scheduled date.
- datespec – A regular date specification with 6 (crontab like year-extended) fields, recurrent date or an event name option.
- command – Can be an absolute path to a command followed by its arguments or a special command name with the '@' prefix (e.g. anti-virus update: *@update*).

```
#scheduler_tasks = "id;name;flags;failstart;datespec;command;id2;name2;...";
```

The following event names can be used in place of the datespec option:

- start – Daemon startup.
- startonce – Daemon startup but at most once a day.
- engine – Successful engine update.
- login – Web interface logon startup.
- threat – Threat detected.
- notscanned – Not scanned file.

To display the current scheduler configuration, use the Web interface or run the following command:

```
cat @ETCDIR@/scep.cfg | grep scheduler_tasks
```

For a full description of Scheduler and its parameters refer to the Scheduler section of the *scep_daemon(8)* man page.

Web Interface

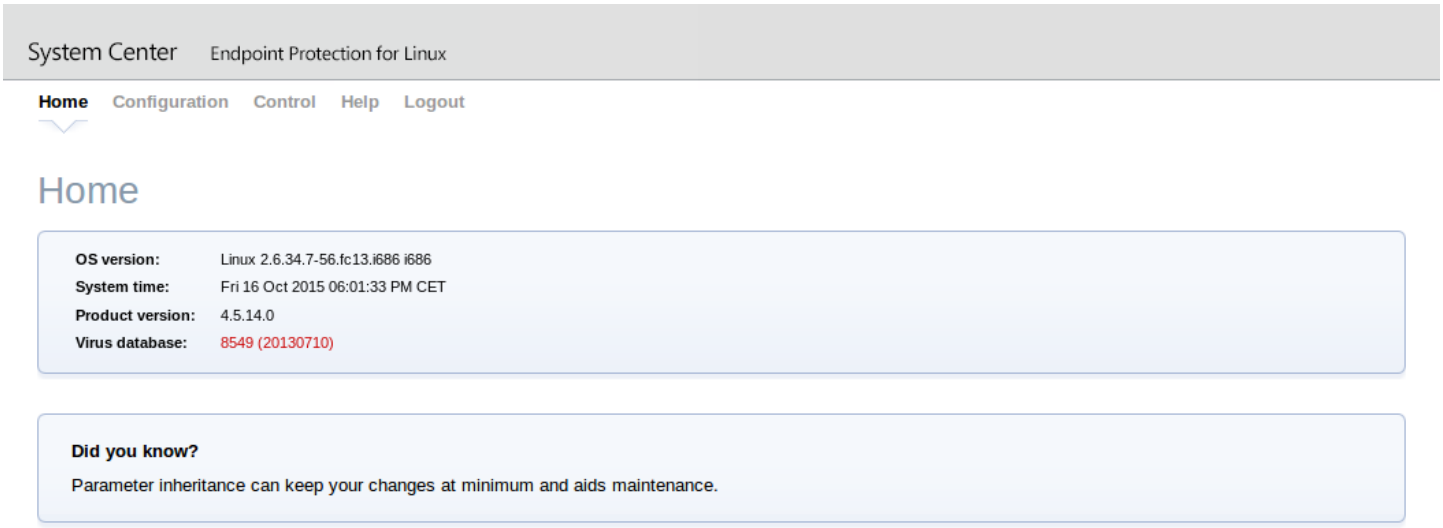
The web interface allows user-friendly configuration and administration of SCEP security systems. This module is a separate agent and must be explicitly enabled. To quickly configure the *Web Interface*, set the following options in the SCEP configuration file and restart the SCEP daemon:

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

Replace the text in italics with your own values and direct your browser to *'https://address:port'* (note the https). Login with *'username/password'*. Basic usage instructions can be found on the help page and technical details about *scep_wwwi* can be found on the *scep_wwwi(1)* man page.

The web interface allows you to remotely access the SCEP daemon and deploy it easily. This powerful utility makes it easy to read and write configuration values.

Figure 6-1. System Center Endpoint Protection - Home screen.



The web interface window of System Center Endpoint Protection is divided into two main sections. The primary window, that serves to display the contents of the selected menu option and the main menu. This horizontal bar on the top lets you navigate between the following main options:

- **Home** – provides basic system and Microsoft product information
- **Configuration** – you can change the System Center Endpoint Protection system configuration here
- **Control** – allows you to run simple tasks and view global statistics about objects processed by scep_daemon
- **Help** – provides detailed usage instructions for the System Center Endpoint Protection web interface
- **Logout** – use to end your current session

Important: Make sure you click the **Save changes** button after making any changes in the **Configuration** section of the web interface to save your new settings. To apply your settings you will need to restart the SCEP daemon by clicking **Apply changes** on the left pane.

Real-time protection configuration example

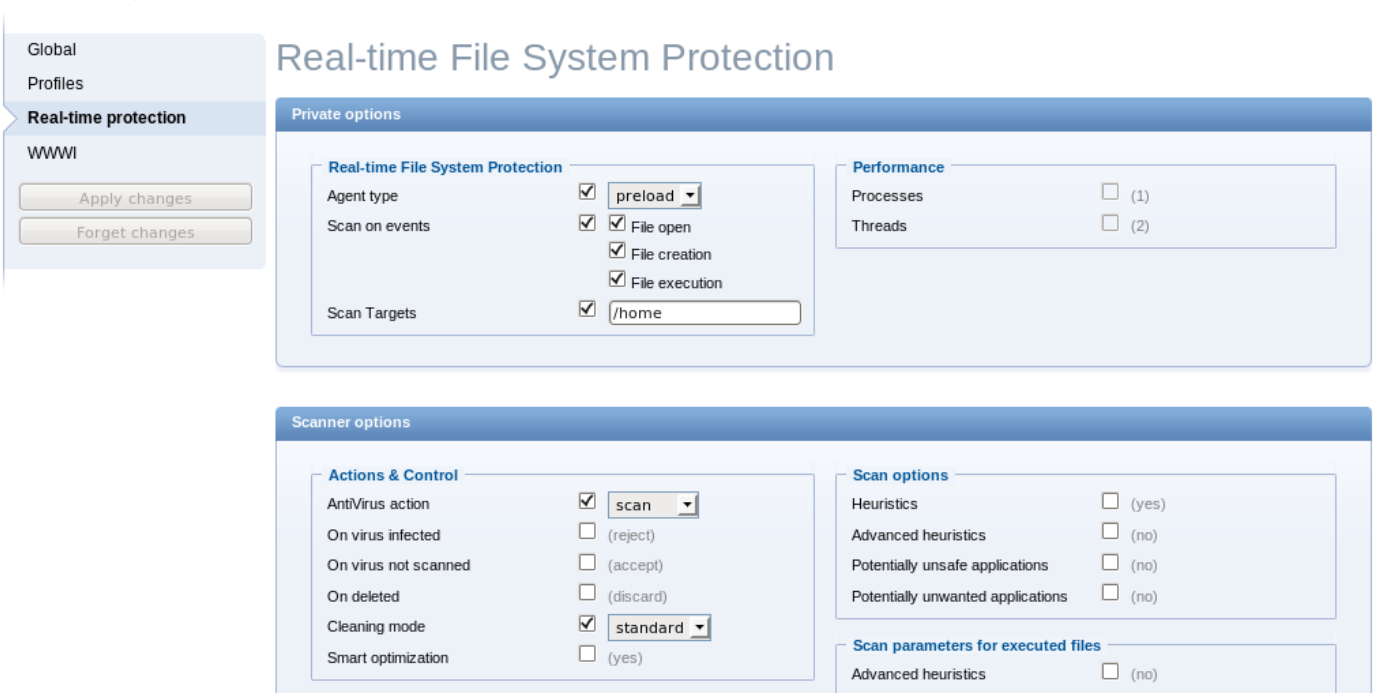
There are two ways you can to configure SCEP. In our example, we will demonstrate how to use either of them to setup the Access Controller module, described in the Real-time protection using preload LIBC library chapter. You can choose the option that best suits you.

- Using the SCEP configuration file:

```
[fac]
agent_type = "preload"
event_mask = "open"
ctl_incl = "/home"
action_av_deleted = "reject"
action_av = "scan"
action_av_infected = "reject"
```

- Using the web interface:

Figure 6-3. SCEP - Configuration > On-Access scanner.



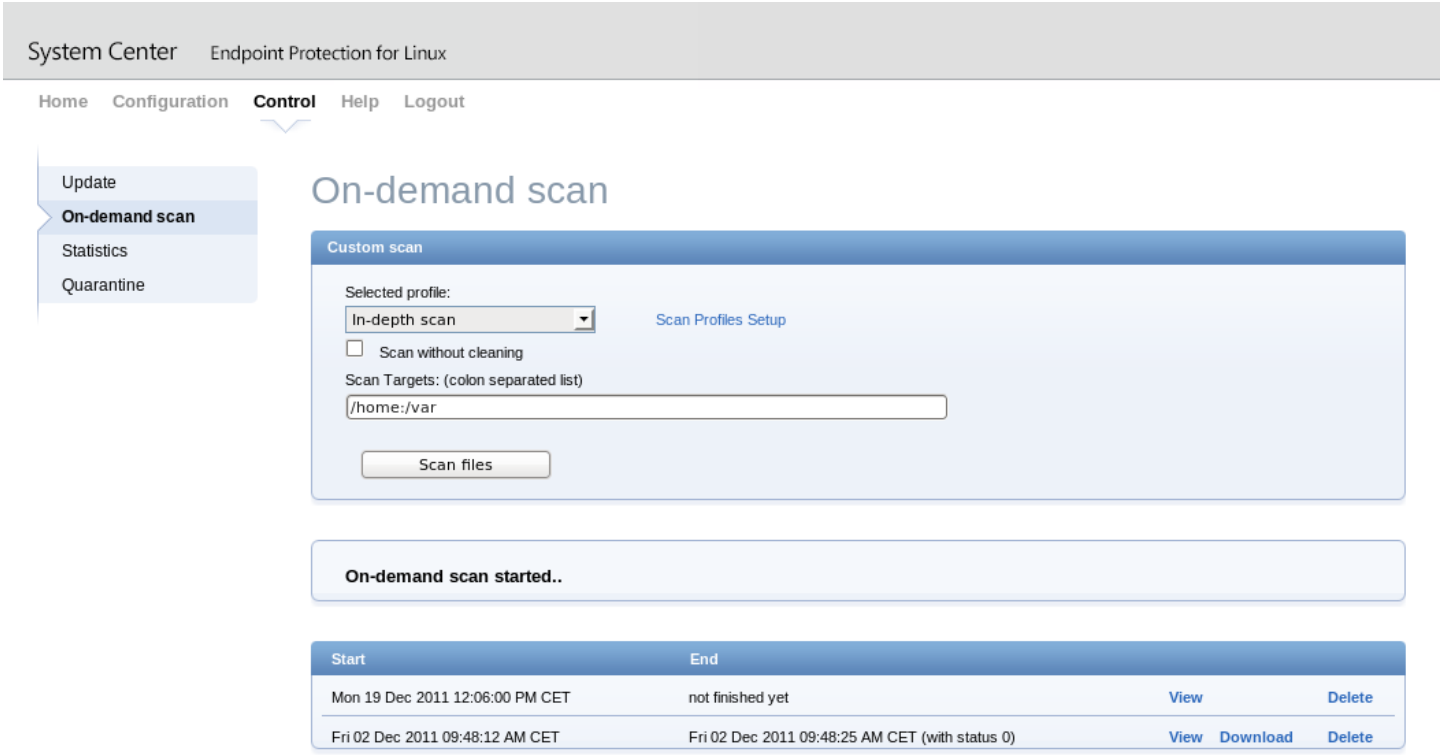
When changing settings in the web interface, always remember to save your configuration by clicking **Save changes**. To apply your new changes, click the **Apply changes** button in the **Configuration** sections panel.

On-Demand scanner

This section comprises an example on how to run the On-Demand scanner to scan for viruses:

- Navigate to **Control > On-Demand Scan**
- Enter the path to the directory you want to scan
- Execute the command-line scanner by clicking the **Scan files** button

Figure 6-4. SCEP - Control > On-Demand scanner.

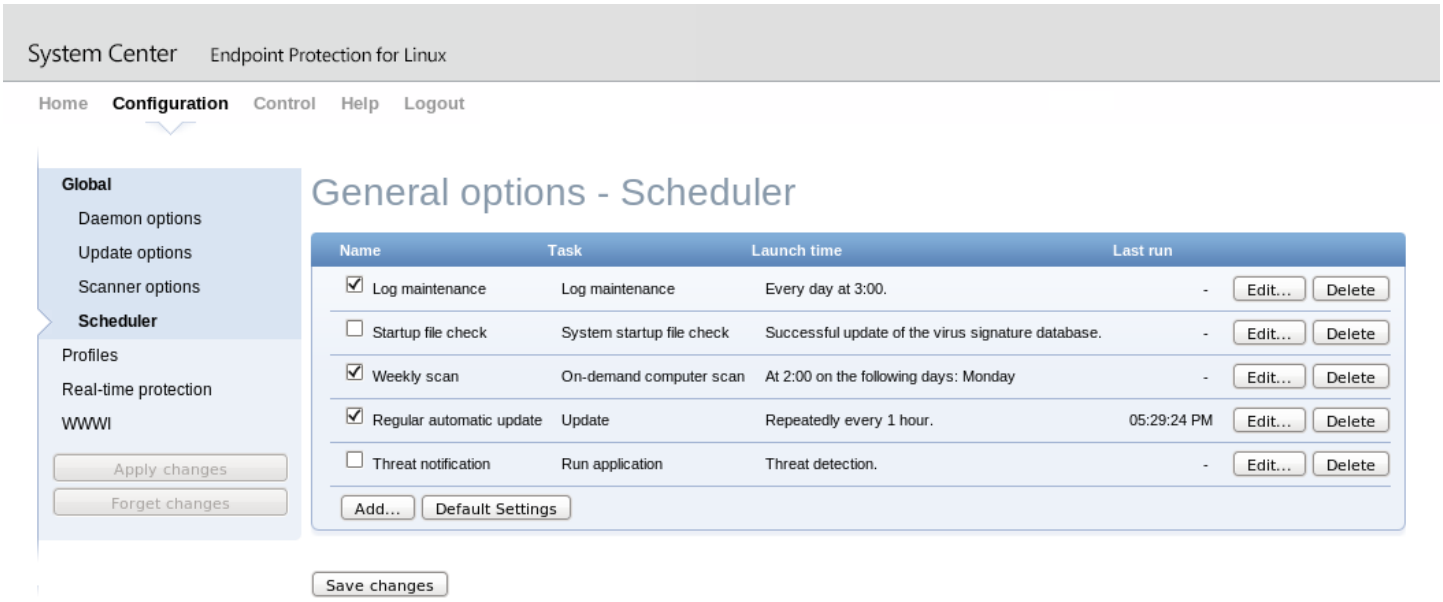


Microsoft Command-line scanner will automatically run in the background. To see the scanning progress, click the **View** link. A new browser window will open.

Scheduler

You can manage the scheduler tasks either via SCEP configuration file (see chapter Scheduler) or using the web interface.

Figure 6-5. SCEP - Global > Scheduler.



Click the checkbox to enable/disable a scheduled task. By default, the following scheduled tasks are displayed:

- **Log maintenance** – The program automatically deletes older logs in order to save hard disk space. The Scheduler will start defragmenting logs. All empty log entries will be removed during this process. This will improve the speed when working with logs. The improvement will be more noticeable if the logs contain a large number of entries.
- **Startup file check** – Scans memory and running services after a successful update of the virus signature database.
- **Weekly scan** – Scans the whole file system on weekly basis (by default on Monday at 2:00 AM). This task can be customized by user.
- **Regular automatic update** – Regularly updating System Center Endpoint Protection is the best method of keeping the maximum level of security on your computer. See SCEP update utility for more information.
- **Threat notification** – By default, each threat will be logged into syslog. In addition, SCEP can be configured to run an external (notification) script to notify a system administrator via email about threat detection.

Statistics

You can view statistics for all of active SCEP agents here. The **Statistics** summary refreshes every 10 seconds.

Figure 6-6. SCEP - Control > Statistics.

Logging

SCEP provides system daemon logging via syslog. *Syslog* is a standard for logging program messages and can be used to log system events such as network and security events.

Messages refer to a facility:

```
auth, authpriv, daemon, cron, ftp, lpr, kern, mail, ..., local0, ..., local7
```

Messages are assigned a priority/level by the sender of the message:

```
Error, Warning, Summall, Summ, Partall, Part, Info, Debug
```

This section describes how to configure and read the logging output of syslog. The *'syslog_facility'* option (default value *'daemon'*) defines the syslog facility used for logging. To modify syslog settings edit the SCEP configuration file or use the Web interface. Modify the value of the *'syslog_class'* parameter to change the logging class. We recommend you modify these settings only if you are familiar with syslog. For an example syslog configuration, see below:

```
syslog_facility = "daemon"
syslog_class = "error:warning:summall"
```

The name and location of the log file depend on your syslog installation and configuration (e.g. rsyslog, syslog-ng, etc.). Standard filenames for syslog output files are for example *'syslog'*, *'daemon.log'*, etc. To follow syslog activity, run one of the following commands from the console:

```
tail -f /var/log/syslog
tail -100 /var/log/syslog | less
cat /var/log/syslog | grep scep | less
```

Important: The monitoring of the Linux SCEP product using System Center Operations Manager must first be enabled in the SCEP configuration file or via the SCEP Web interface to function correctly. Please make sure that the *'scom_enabled'* parameter in the aforementioned configuration file is set as follows *'scom_enabled = yes'* or change the appropriate setting in the Web interface under **Configuration** > **Global** > **Daemon options** > **SCOM enabled**.

Command-line scripts

SCEP commands can be launched using the command line – manually (@SBINDIR@/scep_*) or with a batch (".sh") script. SCEP command-line usage:

`scep_daemon`	SCEP Security Daemon is the main SCEP'S system control and scanning Daemon module. It reads all the SCEP'S scanner configuration from the main SCEP'S configuration file and provides all the main tasks. Usage: `@SBINDIR@/scep_daemon [OPTIONS..]`
`scep_scan`	SCEP Command-line scanner is an on-demand anti-virus scanning module, which provides scanning of the file system objects upon user request using command line interface. Usage: `@SBINDIR@/scep_scan [OPTIONS..] FILES..`
`scep_set`	SCEPS configuration file SET-up utility allows you to modify the SCEP'S configuration file as requested by given command. Usage: `@SBINDIR@/scep_set [OPTIONS..] [COMMAND]`
`scep_setup`	SCEP'S setup utility is an interactive automated install script to help you easily integrate SCEP Security with your system. Usage: `@SBINDIR@/scep_setup [OPTIONS..] [COMMAND]`
`scep_update`	SCEP'S update utility is a system utility for the creation, update and maintenance of the SCEP'S modules storage mirrors as well as for update of SCEP'S system. Usage: `@BINDIR@/scep_update [OPTIONS..]`

SCEP Security system update

SCEP update utility

To maintain the effectiveness of System Center Endpoint Protection, the virus signature database must be kept up to date. The *scep_update* utility has been developed specifically for this purpose. See the *scep_update(8)* man page for details. In the event that your server accesses the Internet via HTTP proxy, the additional configuration options *'proxy_addr'*, *'proxy_port'* must be defined. If access to the HTTP proxy requires a username and password, the *'proxy_username'* and *'proxy_password'* options must also be defined in this section. To initiate an update, enter the following command:

@SBINDIR@/scep_update

To provide the highest possible security for the end user, the Microsoft team continuously collects virus definitions from all over the world - new patterns are added to the virus signature database in very short intervals. For this reason, we recommend that updates be initiated on a regular basis. To be able to specify the frequency of updates, you need to configure the *'@update'* task in the *'scheduler_tasks'* option in the **[global]** section of the SCEP configuration file. You can also use the Scheduler to set the update frequency. The SCEP daemon must be up and running in order to successfully update the virus signature database.

SCEP update process description

The update process consists of two stages: First, the precompiled update modules are downloaded from the Microsoft server.

The second stage of the update process is the compilation of modules loadable by the System Center Endpoint Protection scanner from those stored in the local mirror. Typically, the following SCEP loading modules are created: loader module (em000.dat), scanner module (em001.dat), virus signature database module (em002.dat), archives support module (em003.dat), advanced heuristics module (em004.dat), etc. The modules are created in the following directory:

@BASEDIR@

Let us know

We hope this guide has provided you with a thorough understanding of the requirements for System Center Endpoint Protection installation, configuration and maintenance. However, our goal is to continually improve the quality and effectiveness of our documentation. If you feel that any sections in this Guide are unclear or incomplete, please let us know by contacting Customer Care:

support.microsoft.com

We are dedicated to provide the highest level of support and look forward to helping you should you experience any problems concerning this product.

Appendix A. SCEP setup and configuration

Setting SCEP $PATH environment variable

To access SCEP command-line scripts without typing a full @BINDIR@ or @SBINDIR@ path, you can export the *$PATH* variable directly from a Unix command line using the following command:

```
export PATH=$PATH:/opt/microsoft/scep/sbin
```

After performing this command, typing a full path to SCEP command-line scripts is not be required:

Before:
```
/opt/microsoft/scep/sbin/scep_update
```

After:
```
scep_update
```

Note that this command will be active only for a current shell session. You have to save this command to the *~/.bashrc* file, or somewhere to */etc*, depending on a type of a Unix operating system you use.

Setting SCEP for Samba (Ubuntu upstart)

Upstart is an event-based init daemon used in Ubuntu Linux which starts tasks and services during boot, stops them during shutdown and supervises them while the system is running.

To scan files using the Samba daemon, follow the steps below to use the LIBC preload method:

1. Replace the original command in the Samba service configuration file */etc/init/smbd.conf* with the following:
   ```
   exec env LD_PRELOAD=@LIBDIR@/libscep_pac.so smbd -F
   ```
2. Save the Samba service configuration file and restart the service:
   ```
   stop smbd && start smbd
   ```

Appendix B. PHP License

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/>”.