System Center Endpoint Protection

설치 설명서 및 사용자 설명서

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

목차

소개

System Center Endpoint Protection을 사용해 주셔서 감사합니다 .Microsoft의 최신 검사 엔진은 최고의 검사 속도와 검출률을 제공할 뿐 아니라 시스템 공간도 매우 적게 차지하므로 모든 Linux OS 서버에 이상적인 제품입니다 .

기본 기능

수동 검사기

수동 검사기는 권한 있는 사용자(보통 시스템 관리자)가 명령줄 인터페이스 ,웹 인터페이스 또는 운영 체제의 자동 예약 도구(예: cron)를 통해 시작할 수 있습니다 .수동이라는 용어는 사용자나 시스템이 파일 시스템 오브젝트를 수동으로 검사한다는 의미입니다 .

실시간 보호

실시간 보호는 사용자 및 /또는 운영 체제가 파일 시스템 오브젝트 접근을 시도할 때마다 호출됩니다 .모든 파일 시스템 오브젝트 접근 시도에 대해 검사가 트리거되므로 *실시간*이라는 용어가 사용됩니다 .

시스템의 주요 기능

고급 엔진 알고리즘

Microsoft 안티바이러스 검사 엔진 알고리즘은 최고의 검출률 및 검색 속도를 제공합니다 .

다중 처리

System Center Endpoint Protection은 단일 프로세서 단위와 다중 프로세서 단위에서 모두 실행되도록 개발되었습니다 .

고급 인공지능

System Center Endpoint Protection에는 Win32 웜 ,백도어 감염 및 기타 형식의 맬웨어에 대한 고유 고급 인공지능 기능이 포함되어 있습니다 .

기본 제공 기능

압축 프로그램이 기본적으로 제공되므로 외부 프로그램이 없어도 압축된 오브젝트를 압축 해제할 수 있습니다 .

속도 및 효율성

시스템의 속도와 효율성을 높이기 위해 System Center Endpoint Protection의 아키텍처는 모든 검사 요청이 발송되는 실행 디먼(기존 프로그램)을 기반으로 합니다 .

향상된 보안

보안을 개선하기 위해 모든 주요 디먼(scep_dac는 제외)은 권한 없는 사용자 계정으로 실행됩니다 .

선택 가능한 구성

시스템에서 사용자 또는 클라이언트 /서버에 따라 선택 가능한 구성이 지원됩니다 .

다중 로깅 수준

다중 로깅 수준을 구성하여 시스템 활동 및 침입에 대한 정보를 확인할 수 있습니다 .

웹 인터페이스

직관적이고 사용자에게 친숙한 웹 인터페이스를 통해 구성 및 관리 기능이 제공됩니다 .

외부 라이브러리 없음

System Center Endpoint Protection 설치 시에는 LIBC를 제외한 외부 라이브러리나 프로그램이 필요하지 않습니다 .

사용자 지정 알림

침입이 검출되거나 기타 중요한 이벤트가 발생하는 경우 특정 사용자에게 알림을 표시하도록 시스템을 구성할 수 있습니다 .

낮은 시스템 요구 사항

System Center Endpoint Protection을 효율적으로 실행하는 데 필요한 하드 디스크 공간은 16MB, RAM은 32MB에 불과합니다 .System Center Endpoint Protection은 2.2.x, 2.4.x 및 2.6.x Linux OS 커널 버전에서 원활하게 실행됩니다 .

성능 및 확장성

저성능의 소형 사무실용 서버에서 사용자가 수천 명에 달하는 엔터프라이즈급 ISP 서버에 이르기까지 ,System Center Endpoint Protection은 UNIX 기반 솔루션에서 기대할 수 있는 최고의 성능과 확장성을 제공할 뿐 아니라 Microsoft 보안 제품에 포함된 최고 수준의 보안 기능도 제공합니다 .

용어 및 약어

이 섹션에서는 이 문서에서 사용되는 용어와 약어를 살펴봅니다 .굵은 글꼴로 된 항목은 제품 구성 요소 이름과 새로 정의된 용어 및 약어용으로 예약된 항목입니다 .이 장에서 정의를 제공하는 용어 및 약어에 대해서는 이 문서 뒷부분에서 보다 자세하게 설명합니다 .

SCEP

SCEP는 Microsoft에서 Linux 운영 체제용으로 개발한 보안 제품의 표준 두문자어입니다 .또한 해당 제품이 포함된 소프트웨어 패키지의 이름이기도 합니다 .

SCEP daemon

기본 SCEP 시스템 제어 및 검사 디먼은 *scep_daemon*.

SCEP 기본 디렉터리

바이러스 지문 DB가 포함된 SCEP 로드 가능 모듈이 저장되는 디렉터리입니다 .이후로는 이 디렉터리 지칭 시 약어 *@BASEDIR@*을 사용합니다 .*@BASEDIR@* 값(운영 체제에 따라 다름)은 다음과 같습니다 .

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

SCEP 구성 디렉터리

System Center Endpoint Protection 구성과 관련된 모든 파일이 저장되는 디렉터리입니다 .이후로는 이 디렉터리 지칭 시 약어 *@ETCDIR@*을 사용합니다 .*@ETCDIR@* 값(운영 체제에 따라 다름)은 다음과 같습니다 .

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

SCEP 구성 파일

기본 System Center Endpoint Protection 구성 파일입니다 .파일의 절대 경로는 다음과 같습니다 .

@ETCDIR@/scep.cfg

SCEP 바이너리 파일 디렉터리

관련 System Center Endpoint Protection 바이너리 파일이 저장되는 디렉터리입니다 .이후로는 이 디렉터리 지칭 시 약어 *@BINDIR@*을 사용합니다 .*@BINDIR@* 값(운영 체제에 따라 다름)은 다음과 같습니다 .

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

SCEP 시스템 바이너리 파일 디렉터리

관련 System Center Endpoint Protection 시스템 바이너리 파일이 저장되는 디렉터리입니다 .이후로는 이 디렉터리 지칭 시 약어 *@SBINDIR@*을 사용합니다 .*@SBINDIR@* 값(운영 체제에 따라 다름)은 다음과 같습니다 .

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

SCEP 오브젝트 파일 디렉터리

관련 System Center Endpoint Protection 오브젝트 파일이 저장되는 디렉터리입니다 .이후로는 이 디렉터리 지칭 시 약어 *@LIBDIR@*을 사용합니다 .*@LIBDIR@* 값(운영 체제에 따라 다름)은 다음과 같습니다 .

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

설치

System Center Endpoint Protection은 바이너리 파일로 배포됩니다 .

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

위에 표시된 바이너리 파일에서 *'ext'*는 Linux OS 배포별 접미사(예: Debian의 경우 'deb', RedHat 및 SuSE의 경우 'rpm', 기타 Linux OS 배포의 경우 'tgz')입니다 .

제품을 설치하거나 업그레이드하려면 다음 명령을 사용하여

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

제품의 사용자 라이센스 동의 계약을 표시합니다 .동의 계약을 확인하고 나면 설치 패키지가 현재 작업 디렉터리에 저장되며 패키지 설치 ,제거 또는 업그레이드 관련 정보가 화면에 표시됩니다 .

패키지를 설치한 후 다음 명령을 사용하여 기본 SCEP 서비스가 실행 중인지 확인할 수 있습니다 .

```
ps -C scep_daemon
```

Enter 키를 누르면 다음과 같거나 비슷한 메시지가 표시됩니다 .

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

이 메시지는 둘 이상의 SCEP 디먼 프로세스가 백그라운드에서 실행되고 있음을 나타냅니다 .첫 번째 PID는 시스템의 프로세스 및 스레드 관리자를 나타내고 ,나머지는 SCEP 검사 프로세스를 나타냅니다 .

언어 팩 설치

System Center Endpoint Protection에 해당하는 필수 언어 팩을 설치하려면 다음 명령을 사용합니다 .

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

여기서 *'lng'*는 가져올 파일의 언어 코드로 대체해야 합니다 .

Installation completed successfully 알림이 표시되면 LANG 시스템 변수를 적절하게 업데이트하고 필요한 경우 환경도 업데이트합니다 .그러면 언어 팩 설치가 완료됩니다 .

각 언어 팩에 포함된 구성 요소는 다음과 같습니다 .

- 현지화된 웹 인터페이스
- 현지화된 SCEP 에이전트 및 명령의 콘솔 출력
- 현지화된 PDF 설명서

아키텍처 개요

System Center Endpoint Protection을 정상적으로 설치한 후에는 해당 아키텍처를 파악해야 합니다 .

시스템은 다음과 같은 부분으로 구성됩니다 .

핵심 요소

System Center Endpoint Protection의 핵심 요소는 SCEP 디먼(scep_daemon)입니다 .디먼은 SCEP API 라이브러리 libscep.so 및 SCEP 로딩 모듈 em00X_xx.dat를 사용하여 검사 ,에이전트 디먼 프로세스 유지 관리 ,샘플 전송 시스템 유지 관리 ,로깅 ,알림 등의 기본 시스템 작업을 제공합니다 .자세한 내용은 *scep_daemon(8)* 설명서 페이지를 참조하십시오 .

에이전트

SCEP 에이전트 모듈은 SCEP와 Linux 서버 환경을 통합하는 데 사용됩니다 .

유틸리티

유틸리티 모듈에서는 간단하면서도 효율적인 시스템 관리 기능을 제공합니다 .또한 검역소 관리 ,시스템 설정 및 업데이트와 같은 시스템 작업을 수행합니다 .

구성

보안 시스템에서 가장 중요한 요소는 적절한 구성입니다 .이 장의 나머지 부분에서는 구성과 관련된 모든 구성 요소에 대해 설명합니다 .System Center Endpoint Protection 구성에 필수적인 정보가 포함된 *scep.cfg* 파일에 대해서도 철저하게 파악하는 것이 좋습니다 .

제품을 정상적으로 설치하고 나면 모든 구성 구성 요소가 SCEP 구성 디렉터리에 저장됩니다 .이 디렉터리에는 다음과 같은 파일이 있습니다 .

@ETCDIR@/scep.cfg

제품 기능의 주요 측면을 모두 제어하는 가장 중요한 구성 파일입니다 .scep.cfg 파일은 다양한 섹션으로 이루어져 있으며 각 섹션에는 여러 파라미터가 포함됩니다 .이 파일은 전체 섹션 하나와 여러 "에이전트 "섹션을 포함하며 ,모든 섹션 이름은 대괄호로 묶여 있습니다 .전체 섹션의 파라미터는 SCEP 디먼의 구성 옵션과 SCEP 검사 엔진 구성의 기본값을 정의하는 데 사용됩니다 .에이전트 섹션의 파라미터는 컴퓨터 및 /또는 네트워크에서 다양한 데이터 흐름 유형을 가로채는 데 사용되는 모듈의 구성 옵션을 정의하고 검사를 위해 해당 컴퓨터 및 /또는 네트워크를 준비하는 데 사용됩니다 .시스템 구성에 사용되는 여러 파라미터 외에 ,파일 구성을 제어하는 규칙도 있습니다 .이 파일을 가장 효율적으로 구성하는 방법에 대한 자세한 내용은 *scep.cfg(5)* 및 *scep_daemon(8)* 설명서 페이지와 관련 에이전트 설명서 페이지를 참조하십시오 .

@ETCDIR@/certs

이 디렉터리는 SCEP 웹 인터페이스에서 인증에 사용하는 인증서를 저장하는 데 사용됩니다 .자세한 내용은 *scep_wwwi(8)* 설명서 페이지를 참조하십시오 .

@ETCDIR@/scripts/daemon_notification_script

SCEP 구성 파일 파라미터 *'exec_script'*를 통해 활성화하는 경우 이 스크립트는 안티바이러스 시스템에서 침입을 검출할 때 실행되며 ,해당 침입 검출에 대한 이메일 알림을 시스템 관리자에게 보내는 데 사용됩니다 .

파일 시스템 서비스와의 통합

이 장에서는 바이러스 및 웜 파일 시스템 감염을 가장 효율적으로 방지하는 수동 및 실시간 보호 구성에 대해 설명합니다 . System Center Endpoint Protection의 검사 기능은 수동 검사기 명령 *'scep_scan'* 및 실시간 검사기 명령 *'scep_dac'*에서 파생됩니다 .Linux 버전 System Center Endpoint Protection에서는 미리 로드된 라이브러리 모듈 *libscep_pac.so*를 사용하는 실시간 검사기 기술을 추가로 제공합니다 .다음 섹션에서 이러한 모든 명령에 대해 설명합니다 .

수동 검사기

수동 검사기는 권한 있는 사용자(보통 시스템 관리자)가 명령줄 인터페이스 ,웹 인터페이스 또는 운영 체제의 자동 예약 도구(예: cron)를 통해 시작할 수 있습니다 .수동이라는 용어는 사용자나 시스템이 파일 시스템 오브젝트를 수동으로 검사한다는 의미입니다 .

수동 검사기는 특별한 구성 없이도 실행할 수 있습니다 .SCEP 패키지를 제대로 설치하고 나면 명령줄 인터페이스나 스케줄러 도구를 사용하여 수동 검사기를 즉시 실행할 수 있습니다 .명령줄에서 수동 검사기를 실행하려면 다음 구문을 사용합니다 .

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

여기서 FILES는 검사할 디렉터리 및 /또는 파일 목록입니다 .

SCEP 수동 검사기에서는 여러 명령줄 옵션을 사용할 수 있습니다 .전체 옵션 목록을 확인하려면 *scep_scan(8)* 설명서 페이지를 참조하십시오 .

Dazuko에서 제공하는 실시간 보호

실시간 보호는 사용자 및 /또는 운영 체제가 파일 시스템 오브젝트에 접근하면 호출됩니다 .선택한 파일 시스템 오브젝트에 대한 모든 접근 시도에 대해 검사기가 트리거되므로 *실시간*이라는 용어가 사용됩니다 .

SCEP 실시간 검사기에서 사용하는 기술은 Dazuko(da-tzu-ko) 커널 모듈에서 제공되며 ,커널 호출 가로채기를 기반으로 합니다 .Dazuko 프로젝트는 오픈 소스 방식이므로 해당 소스 코드를 자유롭게 배포할 수 있습니다 .따라서 사용자는 커널 모듈을 컴파일해 고유한 사용자 지정 커널을 생성할 수 있습니다 .Dazuko 커널 모듈은 SCEP 제품에 포함되어 있지 않으므로 실시간 명령 *scep_dac*를 사용하기 전에 컴파일하여 커널에 설치해야 합니다 .Dazuko 기술에서는 사용하는 파일 시스템 유형에 관계없이 실시간 검사를 수행할 수 있습니다 .이 기술은 Network File System (NFS), Nettalk 및 Samba를 통한 파일 시스템 오브젝트 검사에도 적합합니다 .

중요: 실시간 검사기 구성 및 사용에 대한 상세 정보를 제공하기 전에 ,검사기는 기본적으로 외부에서 연결된 파일 시스템을 보호하기 위해 개발 및 테스트되었음을 밝혀 둡니다 .외부에서 연결되지 않은 파일 시스템이 여러 개 있는 경우에는 시스템 중단을 방지하기 위해 파일 접근 제어에서 해당 시스템을 제외해야 합니다 .일반적으로 제외해야 하는 디렉터리의 예로는 */dev'* 디렉터리 및 SCEP에서 사용하는 디렉터리가 있습니다 .

작동 원칙

실시간 보호 *scep_dac*(SCEP Dazuko-powered file Access Controller)는 파일 시스템에 대한 지속적인 모니터링 및 제어 기능을 제공하는 기존 프로그램입니다 .사용자 지정 가능 파일 접근 이벤트 유형에 따라 모든 파일 시스템 오브젝트를 검사합니다 . 현재 버전에서 지원되는 이벤트 유형은 다음과 같습니다 .

open 이벤트

이 파일 접근 유형을 활성화하려면 scep.cfg 파일의 **[fac]** 섹션에서 *'event_mask'* 파라미터의 값을 open으로 설정합니다 .그러면 Dazuko 접근 마스크의 ON_OPEN 비트가 활성화됩니다 .

close 이벤트

이 파일 접근 유형을 활성화하려면 scep.cfg 파일의 **[fac]** 섹션에서 *'event_mask'* 파라미터의 값을 close로 설정합니다 .그러면 Dazuko 접근 마스크의 ON_OPEN 비트가 활성화되고 ,Dazuko 접근 마스크의 ON_CLOSE 및 ON_CLOSE_MODIFIED 비트가 활성화됩니다 .

참고: 일부 OS 커널에서는 ON_CLOSE 이벤트 가로채기가 지원되지 않습니다 .이러한 경우에는 *scep_dac*에서 닫기 이벤트를 모니터링하지 않습니다 .

exec 이벤트

이 파일 접근 유형을 활성화하려면 scep.cfg 파일의 **[fac]** 섹션에서 *'event_mask'* 파라미터의 값을 exec로 설정합니다 .그러면 Dazuko 접근 마스크의 ON_EXEC 비트가 활성화됩니다 .

실시간 보호 기능을 사용하면 *scep_daemon*에서 모든 여는 파일 ,닫는 파일 및 실행하는 파일에 바이러스가 있는지를 먼저 검사합니다 .이 검사 결과에 따라 특정 파일 접근이 거부되거나 허용됩니다 .

설치 및 구성

Dazuko 커널 모듈은 *scep_dac*를 초기화하기 전에 실행 커널 내에서 컴파일 및 설치해야 합니다 .Dazuko 컴파일 및 설치 방법에 대한 자세한 내용은 다음 웹 사이트를 참조하십시오 .

http://www.dazuko.org

Dazuko를 설치한 다음 SCEP 구성 파일(scep.cfg)의 **[global]** 및 **[fac]** 섹션을 검토하여 편집합니다 .실시간 보호 기능이 정상적으로 작동하려면 이 파일의 **[fac]** 섹션 내 *'agent_type'* 옵션을 적절하게 구성해야 합니다 .또한 실시간 보호에서 모니터링할 파일 시스템 오브젝트(예: 디렉터리 및 파일)를 정의해야 합니다 .이렇게 하려면 *'ctl_incl'* 및 *'ctl_excl'* 옵션(역시 **[fac]** 섹션 내에 있음)의 파라미터를 정의합니다 .scep.cfg 파일을 변경한 후에는 SCEP 디먼을 다시 로드하여 새로 생성된 구성을 강제로 다시 읽도록 할 수 있습니다 .

팁

scep_dac 디먼 초기화 전에 Dazuko 모듈이 로드되도록 하려면 다음 단계를 수행합니다 .

커널 모듈용으로 예약된 다음 디렉터리 중 하나에 Dazuko 모듈 복사본을 저장합니다 .

```
/lib/modules
```

검사를

```
/modules
```

커널 유틸리티 'depmod' 및 'modprobe'(BSD OS의 경우 'kldconfig' 및 'kldload' 사용)를 사용하여 새로 추가한 Dazuko 모듈의 종속성을 처리하고 해당 모듈을 정상적으로 초기화합니다 .

scep_daemon 초기화 스크립트 '/etc/init.d/scep_daemon'에서 디먼 초기화 문 앞에 다음 줄을 삽입합니다 .

```
/sbin/modprobe dazuko
```

BSD OS의 경우 아래 줄을

```
/sbin/kldconfig dazuko
```

'/usr/local/etc/rc.d/scep_daemon.sh' 스크립트에 삽입해야 합니다 .

경고! 이러한 단계는 정확히 지정된 순서대로 실행해야 합니다 .커널 모듈은 커널 모듈 디렉터리 내에 있지 않으면 정상적으로 로드되지 않아 시스템이 중단될 수 있습니다 .

미리 로드 LIBC 라이브러리를 사용한 실시간 보호

이전 섹션에서는 Dazuko에서 제공하는 실시간 보호 기능과 Linux/BSD 파일 시스템 서비스를 통합하는 방법에 대해 설명했습니다 .다음과 같은 중요한 시스템을 유지 관리하는 시스템 관리자와 같은 일부 경우에는 Dazuko를 사용하기가 어려울 수도 있습니다 .

- 실행 커널과 관련된 소스 코드 및 /또는 구성 파일을 사용할 수 없는 시스템
- 커널이 모듈 방식이 아닌 모놀리식 방식인 시스템
- 지정된 OS가 Dazuko 모듈에서 지원되지 않는 시스템

이러한 경우에는 미리 로드 LIBC 라이브러리를 기반으로 하는 실시간 검사 기술을 사용해야 합니다 .자세한 내용은 이 섹션의 다음 항목을 참조하십시오 .이 섹션은 Linux OS 사용자에게만 적용되며 미리 로드 라이브러리 *'libscep_pac.so'*를 사용한 실시간 검사기 작동 ,설치 및 구성 관련 정보를 포함합니다 .

작동 원칙

실시간 보호 *libscep_pac.so*(SCEP Preload library based file Access Controller)는 시스템 시작 시 활성화되는 공유 오브젝트 라이브러리입니다 .이 라이브러리는 FTP 서버 ,Samba 서버 등의 파일 시스템 서버에서 LIBC 호출에 사용됩니다 .사용자 지정 가능 파일 접근 이벤트 유형에 따라 모든 파일 시스템 오브젝트를 검사합니다 .현재 버전에서 지원되는 이벤트 유형은 다음과 같습니다 .

open 이벤트

esest.cfg 파일의 *'event_mask'* 파라미터(**[fac]** 섹션)에 *'open'*이라는 단어가 있으면 이 파일 접근 유형이 활성화됩니다 .

close 이벤트

scep.cfg 파일의 *'event_mask'* 파라미터(**[fac]** 섹션)에 *'close'*라는 단어가 있으면 이 파일 접근 유형이 활성화됩니다 .이 경우에는 LIBC의 모든 파일 설명자 및 FILE 스트림 close 함수를 가로챕니다 .

exec 이벤트

scep.cfg 파일의 *'event_mask'* 파라미터(**[fac]** 섹션)에 *'exec'*라는 단어가 있으면 이 파일 접근 유형이 활성화됩니다 .이 경우에는 LIBC의 모든 exec 함수를 가로챕니다 .

SCEP 디먼은 모든 여는 파일 ,닫는 파일 및 실행하는 파일에 바이러스가 있는지를 검사합니다 .이 검사 결과에 따라 지정된 파일 접근이 거부되거나 허용됩니다 .

설치 및 구성

libscep_pac.so 라이브러리 모듈은 미리 로드된 라이브러리의 표준 설치 메커니즘을 사용하여 설치됩니다 .*libscep_pac.so* 라이브러리에 대한 절대 경로를 포함하여 *'LD_PRELOAD'* 환경 변수를 정의해야 합니다 .자세한 내용은 *ld.so(8)* 설명서 페이지를 참조하십시오 .

참고: *'LD_PRELOAD'* 환경 변수는 실시간 보호를 통해 제어할 네트워크 서버 디먼 프로세스(ftp, Samba 등)에 대해서만 정의해야 합니다 .일반적으로는 모든 운영 체제 프로세스에 대해 LIBC 호출을 미리 로드하지 않는 것이 좋습니다 .이렇게 하는 경우 시스템 성능이 크게 저하되거나 시스템이 중단될 수도 있기 때문입니다 .따라서 '/etc/ld.so.preload' 파일은 사용해서는 안 되며 ,'LD_PRELOAD' 환경 변수도 전체적으로 내보내서는 안 됩니다 .이 두 작업을 수행하는 경우 관련 LIBC 호출이 모두 재정의되어 초기화 중에 시스템이 중단될 수 있습니다 .

지정된 파일 시스템 내의 관련 파일 접근 호출만 가로채려면 다음 줄을 사용하여 실행 문을 재정의할 수 있습니다 .

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

여기서 'COMMAND COMMAND-ARGUMENTS'는 원래 실행 문입니다 .

SCEP 구성 파일(scep.cfg)의 **[global]** 및 **[fac]** 섹션을 검토하여 편집합니다 .실시간 검사기가 정상적으로 작동하려면 미리 로드 라이브러리를 통해 제어해야 하는 파일 시스템 오브젝트(예: 디렉터리 및 파일)를 정의해야 합니다 .이렇게 하려면 SCEP 구성 파일의 **[fac]** 섹션에서 *'ctl_incl'* 및 *'ctl_excl'* 옵션의 파라미터를 정의합니다 .scep.cfg 파일을 변경한 후에는 SCEP 디먼을 다시 로드하여 새로 생성된 구성을 강제로 다시 읽도록 할 수 있습니다 .

팁

파일 시스템이 시작된 직후에 실시간 보호를 활성화하려면 해당하는 네트워크 파일 서버 초기화 스크립트 내에서 *'LD_PRELOAD'* 환경 변수를 정의해야 합니다 .

예: Samba 서버를 시작한 직후 실시간 검사기에서 모든 파일 시스템 접근 이벤트를 모니터링하도록 하려는 경우를 가정합니다 .이 경우 Samba 디먼 초기화 스크립트(/etc/init.d/smb) 내에서 다음 문을

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

다음 줄로 대체합니다 .

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

이렇게 하면 Samba에 의해 제어되는 선택한 파일 시스템 오브젝트를 시스템 시작 시에 검사합니다 .

중요 SCEP 메커니즘

오브젝트 처리 정책

오브젝트 처리 정책 메커니즘은 상태에 따라 검사된 오브젝트를 필터링하는 기능을 제공합니다 .이 기능은 다음 구성 옵션을 기반으로 합니다 .

- action_av
- action_av_infected
- action_av_notscanned
- action_av_deleted

이러한 옵션에 대한 자세한 내용은 *scep.cfg(5)* 설명서 페이지를 참조하십시오 .

처리된 모든 오브젝트는 먼저 *'action_av'* 옵션의 구성에 따라 처리됩니다 .이 옵션이 *'accept'*(또는 *'defer'*, *'discard'*, *'reject'*)로 설정되어 있으면 오브젝트가 수락(지연 ,무시 ,거부)됩니다 .이 옵션이 *'scan'*으로 설정되어 있으면 오브젝트에 대해 바이러스 침입 검사를 수행하고 ,*'av_clean_mode'* 옵션이 *'yes'*로 설정되어 있으면 오브젝트 치료도 수행합니다 .또한 *'action_av_infected'*, *'action_av_notscanned'* 및 *'action_av_deleted'* 구성 옵션도 고려하여 오브젝트 처리를 추가로 평가합니다 .이 세 동작 옵션에 의해 *'accept'* 동작이 수행된 경우 해당 오브젝트를 수락됩니다 .그렇지 않으면 오브젝트가 차단됩니다 .

사용자별 구성

사용자별 구성 메커니즘은 높은 수준의 사용자 지정 및 기능을 제공하는 데 사용됩니다 .시스템 관리자는 이 메커니즘을 통해 파일 시스템 오브젝트에 접근하는 사용자를 기준으로 SCEP 안티바이러스 검사기 파라미터를 정의할 수 있습니다 .

이 기능에 대한 자세한 설명은 *scep.cfg(5)* 설명서 페이지에서 확인할 수 있습니다 .이 섹션에서는 사용자별 구성의 간단한 예제만 제공합니다 .

이 예제에서는 *scep_dac* 모듈을 사용하여 /home 디렉터리 아래에 연결된 외부 디스크에 대한 ON_OPEN 및 ON_EXEC 접근 이벤트를 제어하고자 합니다 .SCEP 구성 파일의 **[fac]** 섹션에서 모듈을 구성할 수 있습니다 .아래 항목을 참조하십시오 .

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

개별 사용자에 대해 검사 설정을 지정하려면 *'user_config'* 파라미터가 개별 검사 규칙을 저장할 특수 구성 파일 이름을 지정해야 합니다 .여기에 나와 있는 예제에서 특수 구성 파일은 *'scep_dac_spec.cfg'*이며 SCEP 구성 디렉터리 내에 있습니다 .이 디렉터리는 운영 체제에 따라 다릅니다 .자세한 내용은 용어 및 약어 페이지를 참조하십시오 .

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

[fac] 섹션 내에서 *'user_config'* 파일 파라미터를 지정한 후에는 SCEP 구성 디렉터리에 *'scep_dac_spec.cfg'* 파일을 생성해야 합니다 .그리고 마지막으로 원하는 검사 규칙을 추가합니다 .

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

특수 섹션 맨 위에 개별 규칙을 적용할 사용자 이름을 입력합니다 .이 구성을 사용하는 경우 파일 시스템 접근을 시도하는 다른 모든 사용자가 정상적으로 처리됩니다 .즉 ,접근이 거부(차단)되는 *'username'* 사용자를 제외한 다른 사용자가 접근하는 모든 파일 시스템 오브젝트에 대해 침입을 검사합니다 .

스케줄러

스케줄러의 기능으로는 지정된 시간에 또는 특정 이벤트 발생 시에 예약된 작업 실행 ,미리 정의된 구성 및 속성을 사용하여 작업 관리 및 시작 등이 있습니다 .작업 구성과 속성을 사용하면 시작 날짜와 시간을 조정할 수 있을 뿐 아니라 작업 실행 중에 사용자 지정 프로필을 사용함으로써 작업 적용 범위를 확장할 수 있습니다 .

'scheduler_tasks' 옵션은 기본적으로 주석 처리되므로 기본 스케줄러 구성이 적용됩니다 .SCEP 구성 파일에서 모든 파라미터와 작업은 세미콜론으로 구분됩니다 .기타 모든 세미콜론과 백슬래시는 백슬래시로 이스케이프 처리해야 합니다 .각 작업에는 6개의 파라미터가 있으며 해당 구문은 다음과 같습니다 .

- id - 고유 번호입니다 .
- name - 작업 설명입니다 .
- flags - 지정한 스케줄러 작업을 비활성화하는 특수 플래그를 설정할 수 있습니다 .
- failstart - 작업을 예약된 날짜에 실행할 수 없는 경우 수행할 작업을 지시합니다 .
- datespec - 정기적인 날짜 지정 항목으로 ,6개(crontab 스타일의 연도 연장 형식)의 필드 ,되풀이 날짜 또는 이벤트 이름 옵션이 포함됩니다 .
- command - 명령의 절대 경로에 해당 인수나 특수 명령 이름이 붙는 형태로 ,@' 접두어가 추가됩니다(예: 안티바이러스 업데이트: *@update*).

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

datespec 옵션 대신 다음 이벤트 이름을 사용할 수 있습니다 .

- start - 디먼 시작
- startonce - 디먼 시작(최대 1일 1회)
- engine - 엔진 업데이트 성공
- login - 웹 인터페이스 로그온 시작
- threat - 위협 검출됨
- notscanned - 파일을 검사하지 않음

현재 스케줄러 구성을 표시하려면 웹 인터페이스를 사용하거나 다음 명령을 실행합니다 .

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

스케줄러 및 해당 파라미터에 대한 전체 설명은 *scep_daemon(8)* 설명서 페이지의 스케줄러 섹션을 참조하십시오 .

웹 인터페이스

웹 인터페이스에서는 사용자에게 친숙한 방식으로 SCEP 보안 시스템 구성 및 관리를 수행할 수 있습니다 .이 모듈은 별도의 에이전트이므로 명시적으로 활성화해야 합니다 .*웹 인터페이스*를 빠르게 구성하려면 SCEP 구성 파일에서 다음 옵션을 설정하고 SCEP 디먼을 다시 시작합니다 .

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

기울임꼴로 된 텍스트를 사용자 환경의 값으로 대체하고 브라우저가 *'https://주소:포트'*(https를 사용해야 함)로 이동하도록 합니다 .그런 다음 *'사용자 이름/비밀번호'*를 사용하여 로그인합니다 .기본 사용 지침은 도움말 페이지에서 확인할 수 있으며 , *scep_wwwi*에 대한 기술 세부 정보는 *scep_wwwi(1)* 설명서 페이지에서 확인할 수 있습니다 .

웹 인터페이스에서는 SCEP 디먼에 원격 접근하여 디먼을 쉽게 배포할 수 있습니다 .이 유용한 유틸리티를 통해 구성 값을 쉽게 읽고 쓸 수 있습니다 .

그림 6-1. System Center Endpoint Protection - 홈 화면

System Center Endpoint Protection의 웹 인터페이스 창은 두 개의 기본 섹션으로 구분됩니다 .기본 창에는 선택한 메뉴 옵션 및 기본 메뉴의 콘텐츠가 표시됩니다 .위쪽의 가로 막대에서는 다음의 기본 옵션 간을 이동할 수 있습니다 .

- **홈** - 기본 시스템 및 Microsoft 제품 정보가 제공됩니다 .
- **구성** - 여기에서 System Center Endpoint Protection 시스템 구성을 변경할 수 있습니다 .
- **제어** - 간단한 작업을 실행하고 scep_daemon에서 처리된 오브젝트에 대한 전체 통계를 확인할 수 있습니다 .
- **도움말** - System Center Endpoint Protection 웹 인터페이스 사용 지침을 제공합니다 .
- **로그아웃** - 현재 세션을 종료하는 데 사용됩니다 .

중요: 웹 인터페이스의 **구성** 섹션에서 변경을 수행한 후에 새 설정을 저장하려면 **변경 사항 저장** 버튼을 클릭해야 합니다 . 설정을 적용하려면 왼쪽 창에서 **변경 사항 적용**을 클릭하여 SCEP 디먼을 다시 시작해야 합니다 .

실시간 보호 구성 예제

SCEP는 두 가지 방법으로 구성할 수 있습니다 .여기서 소개하는 예제에서는 미리 로드 LIBC 라이브러리를 사용한 실시간 보호 장에서 설명하는 대로 두 방법 중 하나를 사용하여 접근 컨트롤러 모듈을 설정하는 방법을 보여 줍니다 .둘 중 적절한 옵션을 선택하면 됩니다 .

- SCEP 구성 파일 사용:

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

- 웹 인터페이스 사용:

그림 6-3. SCEP - 구성 > 실시간 검사기

웹 인터페이스에서 설정을 변경할 때는 **변경 사항 저장**을 클릭하여 구성을 저장해야 합니다 .새 변경 사항을 적용하려면 **구성** 섹션 패널에서 **변경 사항 적용** 버튼을 클릭합니다 .

수동 검사기

이 섹션에서는 바이러스 검사를 위해 수동 검사기를 실행하는 방법의 예제를 제공합니다 .

- **제어 > 수동 검사**로 이동합니다 .
- 검사할 디렉터리의 경로를 입력합니다 .
- **파일 검사** 버튼을 클릭하여 명령줄 검사기를 실행합니다 .

그림 6-4. SCEP - 제어 > 수동 검사기

Microsoft 명령줄 검사기는 백그라운드에서 자동으로 실행됩니다 .검사 진행률을 확인하려면 **보기** 링크를 클릭합니다 .그러면 새 브라우저 창이 열립니다 .

스케줄러

SCEP 구성 파일(스케줄러 장 참조) 또는 웹 인터페이스를 사용하여 스케줄러 작업을 관리할 수 있습니다 .

그림 6-5. SCEP - 전체 > 스케줄러

예약된 작업을 활성화 /비활성화하려면 확인란을 클릭합니다 .기본적으로는 다음과 같은 예약된 작업이 표시됩니다 .

- **로그 유지 관리** - 프로그램에서는 하드 디스크 공간을 절약하기 위해 오래된 로그를 자동으로 삭제합니다 .스케줄러는 로그 조각 모음을 시작합니다 .이 프로세스 중에는 비어 있는 모든 로그 항목이 제거됩니다 .그러면 로그 작업 시의 속도가 개선됩니다 .로그에 항목이 많이 포함된 경우에는 속도가 더욱 빨라집니다 .
- **시작 파일 검사** - 바이러스 지문 DB를 업데이트한 후 메모리 및 실행 중인 서비스를 검사합니다 .
- **매주 검사** - 매주 전체 파일 시스템을 검사합니다(기본적으로 월요일 오전 2:00에 검사). 이 작업은 사용자가 사용자 지정할 수 있습니다 .
- **정기적 자동 업데이트** - 컴퓨터의 보안 수준을 최대로 유지하기 위한 가장 좋은 방법은 System Center Endpoint Protection을 정기적으로 업데이트하는 것입니다 .자세한 내용은 SCEP 업데이트 유틸리티를 참조하십시오 .
- **위협 알림** - 기본적으로는 각 위협이 syslog에 기록됩니다 .또한 외부(알림) 스크립트를 실행하여 위협 검출과 관련해 이메일로 시스템 관리자에게 알리도록 SCEP를 구성할 수 있습니다 .

통계

여기서는 모든 활성 SCEP 에이전트의 통계를 확인할 수 있습니다 .**통계** 요약은 10초마다 새로 고쳐집니다 .

그림 6-6. SCEP - 제어 > 통계

로깅

SCEP에서는 syslog를 통한 시스템 디먼 로깅 기능을 제공합니다 .*Syslog*는 프로그램 메시지 로깅용 표준으로 ,네트워크 및 보안 이벤트와 같은 시스템 이벤트를 기록하는 데 사용할 수 있습니다 .

메시지는 특정 기능을 지칭합니다 .

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

메시지 보낸 사람이 메시지에 순위 /수준을 지정합니다 .

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

이 섹션에서는 syslog의 로깅 출력을 구성하고 읽는 방법을 설명합니다 .*'syslog_facility'* 옵션(기본값 *'daemon'*)이 로깅에 사용되는 syslog 기능을 정의합니다 .syslog 설정을 수정하려면 SCEP 구성 파일을 편집하거나 웹 인터페이스를 사용합니다 . 로깅 클래스를 변경하려면 *'syslog_class'* 파라미터의 값을 수정합니다 .syslog에 대해 잘 아는 경우에만 이러한 설정을 수정하는 것이 좋습니다 .예제 syslog 구성은 아래를 참조하십시오 .

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

로그 파일의 이름과 위치는 syslog 설치 및 구성(예: rsyslog, syslog-ng 등)에 따라 달라집니다 .syslog 출력 파일의 표준 파일 이름은 *'syslog'*, *'daemon.log'* 등입니다 .syslog 활동을 확인하려면 콘솔에서 다음 명령 중 하나를 실행합니다 .

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

중요: 제대로 작동하려면 먼저 SCEP 구성 파일에서 또는 SCEP 웹 인터페이스를 통해 System Center Operations Manager 를 사용한 Linux SCEP 제품 모니터링을 활성화하도록 설정해야 합니다 .앞서 언급된 구성 파일에서 *'scom_enabled'* 파라미터가 *'scom_enabled = yes'*로 설정되었는지 확인하거나 웹 인터페이스의 **구성** > **전체** > **디먼 옵션** > **SCOM이 활성화됨**에서 적합한 설정을 변경하십시오 .

SCEP 보안 시스템 업데이트

SCEP 업데이트 유틸리티

System Center Endpoint Protection의 효율성을 유지하려면 바이러스 지문 DB를 최신 상태로 유지해야 합니다 .이러한 용도로 사용할 수 있도록 *scep_update* 유틸리티가 개발되었습니다 .자세한 내용은 *scep_update(8)* 설명서 페이지를 참조하십시오 .서버에서 HTTP 프록시를 통해 인터넷에 접근하는 경우에는 추가 구성 옵션 *'proxy_addr'*, *'proxy_port'*를 정의해야 합니다 . HTTP 프록시 접근 시 사용자 이름과 비밀번호가 필요한 경우 이 섹션에서 *'proxy_username'* 및 *'proxy_password'* 옵션도 정의해야 합니다 .업데이트를 시작하려면 다음 명령을 입력합니다 .

@SBINDIR@/scep_update

최종 사용자에게 가능한 최고 수준의 보안을 제공하기 위해 Microsoft 팀에서는 전 세계에서 바이러스 정의를 끊임없이 수집하고 있습니다 .새로운 패턴은 빠른 시간 내에 바이러스 지문 DB에 추가됩니다 .그러므로 업데이트를 정기적으로 시작하는 것이 좋습니다 .업데이트 빈도를 지정하려면 SCEP 구성 파일의 **[global]** 섹션에 포함된 *'scheduler_tasks'* 옵션에서 *'@update'* 작업을 구성해야 합니다 .스케줄러를 사용하여 업데이트 빈도를 설정할 수도 있습니다 .바이러스 지문 DB를 정상적으로 업데이트하려면 SCEP 디먼이 작동 및 실행 중이어야 합니다 .

SCEP 업데이트 프로세스 설명

업데이트 프로세스는 두 단계로 구성됩니다 .먼저 미리 컴파일된 업데이트 모듈을 Microsoft 서버에서 다운로드합니다 .

업데이트 프로세스의 두 번째 단계에서는 로컬 미러에 저장된 모듈에서 System Center Endpoint Protection 검사기가 로드할 수 있는 모듈을 컴파일합니다 .일반적으로 생성되는 SCEP 로딩 모듈은 로더 모듈(em000.dat), 검사기 모듈(em001.dat), 바이러스 지문 DB 모듈(em002.dat), 압축파일 지원 모듈(em003.dat), 고급 인공지능 모듈(em004.dat) 등이며 ,모듈 생성 디렉터리는 다음과 같습니다 .

@BASEDIR@

의견 보내기

이 설명서를 통해 System Center Endpoint Protection 설치 ,구성 및 유지 관리의 요구 사항을 모두 파악하셨기를 바랍니다 . ESET의 목표는 설명서의 품질과 효율성을 지속적으로 개선하는 것입니다 .이 설명서에서 불명확하거나 불완전한 내용을 발견하신 경우 고객 지원으로 알려 주십시오 .

support.microsoft.com

앞으로도 여러분께 최고 수준의 지원을 제공하고 ,이 제품과 관련하여 문제가 발생하는 경우 문제 해결을 최대한 지원해 드릴 수 있도록 노력하겠습니다 .

부록 A. PHP 라이센스

The PHP License, version 3.01 Copyright (c) 1999 - 2006 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

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

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.