CA Access Control

Windows 용 끝점 관리 안내서

12.7

타사 고지 사항

샘플 스크립트와 샘플 SDK 코드

CA Access Control 제품에 포함된 샘플 스크립트와 샘플 SDK 코드는 정보 제공 목적으로만 "있는 그대로" 제공됩니다. 이 항목은 특정 환경에 맞게 수정이 필요할 수 있으며, 프로덕션 환경에 사용하려면 프로덕션 시스템에 배포하기 전에 반드시 테스트 및 검사를 수행해야 합니다.

CA Technologies 는 이러한 샘플에 대한 지원을 제공하지 않으며 이 스크립트로 인한 어떠한 오류에도 책임을 지지 않습니다.

CA Technologies 제품 참조

이 문서는 다음 CA Technologies 제품을 참조합니다 :

- CA Access Control
- CA Access Control
- CA Single Sign-On(CA SSO)
- CA Top Secret®
- CA ACF2™
- CA Audit
- CA NSM(CA Network and Systems Management, 이전의 Unicenter NSM 및 Unicenter TNG)
- CA Software Delivery(이전의 Unicenter Software Delivery)
- CA Service Desk(이전 이름: Unicenter Service Desk)
- 사용자 활동 보고 (이전 명칭: CA Enterprise Log Manager)
- CA Identity Manager

설명서 규칙

CA Access Control 설명서는 다음과 같은 규칙을 따릅니다.

형식	의미
고정 폭 글꼴	코드 또는 프로그램 출력
기울임꼴	강조 또는 새 용어
굵게	표시된 대로 동일하게 입력해야 하는 텍스트
슬래시(/)	UNIX 및 Windows 경로를 기술하는 데 사용되는 플랫폼 독립적인 디렉터리 구분 기호

이 설명서는 또한 명령 구문과 사용자 입력(고정 폭 글꼴로 표시됨)을 설명할 때 다음과 같은 특별한 규칙을 사용합니다.

형식	의미
기울임꼴	반드시 입력해야 하는 정보
대괄호([ ]) 사이	선택적 피연산자
중괄호({ }) 사이	필수 피연산자 집합
파이프(|)로 구분된 선택 사항	대체 피연산자(하나 선택)를 구분합니다. 예를 들어, 다음은 사용자 이름 *또는* 그룹 이름 중 *하나*라는 의미입니다. {*username*|*groupname*}
…	앞의 항목 또는 항목 그룹이 반복될 수 있음을 나타냅니다.
밑줄	기본값
줄 마지막에 공백 다음의 백슬래시(\)	때때로 이 안내서에서 명령이 한 줄에 모두 표시되지 않는 경우가 있습니다. 이런 경우에는 줄 끝에 공백과 백슬래시(\)를 표시하여 명령이 다음 줄에서 계속됨을 나타냅니다. **참고:** 실제 명령을 입력할 때는 이러한 백슬래시를 포함하지 말고 줄바꿈 없이 명령을 한 줄에 입력하십시오. 백슬래시 및 줄바꿈은 실제 명령 구문에 포함되지 않습니다.

예제: 명령 표기 규칙

다음 코드는 이 안내서에서 명령 규칙이 사용되는 방식을 보여 줍니다.

```
ruler className [props({all|{propertyName1[,propertyName2]...})]
```

설명:

- 표시되는 그대로 입력해야 하는 명령 이름(ruler)은 일반 고정 폭 글꼴로 표시됩니다.
- *className* 옵션은 클래스 이름(예: USER)이 들어갈 자리이므로 기울임꼴로 표시됩니다.

- 대괄호로 묶인 두 번째 부분은 선택적 피연산자를 의미하므로 이 부분 없이 명령을 실행할 수도 있습니다.
- 옵션 매개 변수(props)를 사용할 때 키워드 *all* 을 선택하거나 하나 이상의 속성 이름을 쉼표로 구분하여 지정할 수 있습니다.

파일 위치 규칙

CA Access Control 설명서는 다음과 같은 파일 위치 규칙을 따릅니다.

- *ACInstallDir* - 기본 CA Access Control 설치 디렉터리입니다.
  - Windows - C:\Program Files\CA\AccessControl\
  - UNIX - /opt/CA/AccessControl/
- *ACSharedDir* - UNIX 에서 CA Access Control 에 의해 사용되는 기본 디렉터리입니다.
  - UNIX - /opt/CA/AccessControlShared
- *ACServerInstallDir* - 기본 CA Access Control 엔터프라이즈 관리 설치 디렉터리입니다.
  - /opt/CA/AccessControlServer
- *DistServerInstallDir* - 기본 배포 서버 설치 디렉터리입니다.
  - /opt/CA/DistributionServer
- *JBoss_HOME* - 기본 JBoss 설치 디렉터리입니다.
  - /opt/jboss-4.2.3.GA

CA 에 문의

기술 지원팀에 문의

온라인 기술 지원 및 지사 목록, 기본 서비스 시간, 전화 번호에 대해서는 http://www.ca.com/worldwide 에서 기술 지원팀에 문의하십시오.

설명서 변경 사항

이 설명서가 마지막으로 릴리스된 이후에 다음과 같이 업데이트되었습니다.

- 모니터링 및 감사- 업데이트되어 다음 변경 내용이 포함되었습니다.
  - 감사 규칙 설정(Windows)

목차

제 1 장: 소개

이 섹션은 다음 항목을 포함하고 있습니다.

안내서 정보

이 안내서에서는 개방형 시스템을 위한 전체적인 보안 솔루션을 제공하는 제품인 Windows 용 CA Access Control 에서 사용되는 개념에 대해 설명하며, Windows 끝점 관리 작업 및 개념에 대해서도 설명합니다.

이 안내서는 또한 엔터프라이즈 관리 및 보고 기능과 고급 정책 관리 기능을 제공하는 CA Access Control 과 함께 제공됩니다.

용어를 간단히 나타내기 위해 이 안내서에서는 제품을 CA Access Control 이라고 합니다.

본 안내서의 사용자

이 안내서는 CA Access Control 보호 환경을 구현 및 유지 보수하는 보안 관리자와 시스템 관리자를 대상으로 합니다.

제 2 장: 끝점 관리

CA Access Control 은 개방형 시스템을 위한 포괄적인 액티브 보안 소프트웨어 솔루션으로서 운영 체제와 동적으로 연결되어 있습니다. 사용자가 파일 열기, 사용자 ID 대체, 네트워크 서비스 획득 등과 같이 보안상 중요한 작업을 요청할 때마다 CA Access Control 은 실시간으로 이벤트를 차단하고 표준 운영 체제(OS) 기능으로 제어권을 넘겨주기 전에 그 유효성을 평가합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

CA Access Control 란 무엇입니까?

CA Access Control 은 사용자 기본 플랫폼의 보안을 관리하는 강력한 도구를 제공하여 기업의 보안 요구 사항에 완벽하게 맞춰 사용자 지정할 수 있는 보안 정책을 구현할 수 있습니다. CA Access Control 을 사용하면 사용자, 그룹 및 리소스에 대해 기본 운영 체제에서 제공되는 보안 이상의 보안을 제공할 수 있으며, 한 곳에서 조직 전체의 보안을 관리하고 서로 다른 환경의 Windows 및 UNIX 보안 정책을 통합할 수 있습니다.

Access Control 이 보호하는 엔티티

CA Access Control 은 다음과 같은 항목을 보호합니다.

- **파일**

  사용자가 특정 파일에 대한 액세스 권한을 부여받았습니까?

  CA Access Control 은 파일에 액세스할 수 있는 사용자의 능력을 제한합니다. 사용자에게 READ, WRITE, EXECUTE, DELETE 및 RENAME 과 같은 하나 이상의 액세스 유형을 지정할 수 있습니다. 개별 파일에 대한 액세스를 지정하거나, 이름이 유사한 파일의 집합에 대한 액세스를 지정할 수 있습니다.

- **터미널**

  사용자에게 특정 터미널을 사용할 권한이 있습니까?

  이 검사는 로그인 프로세스 중에 수행됩니다. 개별 터미널 및 터미널 그룹은 CA Access Control 데이터베이스에서 정의할 수 있으며 터미널이나 터미널 그룹을 사용할 수 있는 사용자 또는 사용자 그룹을 설명하는 액세스 규칙도 함께 정의할 수 있습니다. 터미널 보호를 사용하면 권한 없는 터미널이나 스테이션을 통해 강력한 권한을 가진 사용자 계정에 로그인할 수 없습니다.

- **로그온 시간**

  사용자가 특정 요일의 특정 시간에 로그온할 수 있는 권한을 부여받았습니까?

  대부분의 사용자는 스테이션을 주중과 근무 시간에만 사용합니다. 휴일 제한 뿐만 아니라 시간 및 요일 로그인 제한은 해커와 권한 없는 다른 접근자로부터 보호를 제공합니다.

- **TCP/IP**

  다른 스테이션이 로컬 컴퓨터에서 TCP/IP 서비스를 받을 수 있는 권한을 부여받았습니까? 다른 스테이션이 로컬 컴퓨터에 TCP/IP 서비스를 제공할 권한이 있습니까? 다른 스테이션이 로컬 스테이션의 모든 사용자로부터 서비스를 수신할 수 있습니까?

  컴퓨터와 네트워크가 모두 개방되어 있는 개방형 시스템의 장점은 단점이 되기도 합니다. 컴퓨터를 외부에 연결하면 누가 시스템에 들어오고 외부 사용자가 고의적으로 또는 실수로 어떠한 피해를 입힐 수 있는지 확신할 수 없습니다. CA Access Control 에는 로컬 스테이션과 서버가 알 수 없는 스테이션에 서비스를 제공하는 것을 방지하는 "방화벽"이 있습니다.

- **다중 로그인 권한**

  사용자가 두 번째 터미널에서 로그인할 수 있습니까?

  *동시 로그인*이라는 용어는 두 개 이상의 터미널에서 시스템에 로그인할 수 있는 사용자의 능력을 의미합니다. CA Access Control 에서는 사용자가 두 번 이상 로그인하지 못하게 할 수 있습니다. 이러한 기능을 통해 침입자는 이미 로그인되어 있는 사용자 계정에 로그인할 수 없습니다.

- **사용자-정의된 엔터티**

  일반 항목(예: TCP/IP 서비스 및 터미널)과 *추상* 개체라고도 하는 기능 항목(예: 트랜잭션 수행 및 데이터베이스의 레코드 액세스)을 모두 정의하고 보호할 수 있습니다.

- **관리자 권한 측면**

  CA Access Control 에서는 슈퍼 사용자 권한을 운영자에게 위임하고 슈퍼 사용자 계정의 권한을 제한할 수 있습니다.

- **레지스트리 키**

  사용자가 특정 레지스트리 키에 액세스할 수 있는 권한이 있습니까?

  CA Access Control 은 레지스트리 키를 액세스할 수 있는 사용자의 능력을 제한합니다. 사용자에게 READ, WRITE 및 DELETE 와 같은 하나 이상의 액세스 유형을 지정할 수 있습니다. 개별 레지스트리 키에 대한 액세스를 지정하거나 이름이 유사한 레지스트리 키의 집합에 대한 액세스를 지정할 수 있습니다.

- **프로그램**

  특정 프로그램을 트러스트할 수 있습니까? 사용자가 이 프로그램을 호출할 권한이 있습니까? 사용자가 프로그램을 사용하여 특정 리소스를 액세스할 수 있습니까?

  보안 관리자는 프로그램을 테스트하여 프로그램에 대한 무단 액세스를 얻는 데 사용될 수 있는 보안상 허점이 없는지 확인할 수 있습니다. 테스트를 통과하여 안전한 것으로 간주되는 프로그램은 트러스트된 프로그램으로 정의됩니다. **watchdog** 이라고도 하는 CA Access Control 자체 보호 모듈은 특정 시간에 어떤 프로그램이 제어되고 있는지를 파악하고, 이 프로그램이 트러스트된 것으로 분류된 이후 수정 또는 이동되었는지 여부를 확인합니다. 트러스트된 프로그램이 수정 또는 이동된 경우에는 더 이상 트러스트된 것으로 간주되지 않으며 CA Access Control 은 프로그램의 실행을 허용하지 않습니다.

또한 CA Access Control 은 의도하거나 의도하지 않은 다음과 같은 위협으로부터 보호합니다.

- **중지 시도**

  CA Access Control 을 사용하면 중지 시도로부터 중요한 서버와 서비스 또는 데몬을 보호할 수 있습니다.

- **암호 공격**

  CA Access Control 은 여러 유형의 암호 공격으로부터 보호하며 사이트의 암호-정의 정책을 시행하고 침입-시도를 감지합니다.

- **암호를 이용한 범죄**

  CA Access Control 정책은 사용자가 최적의 암호를 만들어 사용하도록 하는 규칙을 설정합니다. 사용자가 적합한 암호를 만들어 사용할 수 있도록 CA Access Control 은 암호의 최대 및 최소 수명을 설정하고, 특정 단어를 사용하지 못하게 하며, 문자를 반복해서 사용할 수 없도록 하고, 기타 제한 사항을 적용할 수 있습니다. 암호는 너무 오랫동안 사용할 수 없습니다.

- **계정 관리**

  CA Access Control 정책을 통해 유휴 계정이 적절히 처리되도록 합니다.

보호 방법

CA Access Control 은 운영 체제에서 초기화가 완료된 후 바로 시작됩니다. CA Access Control 은 보호가 필요한 시스템 서비스에 후크를 배치합니다. 이런 방식으로 서비스가 수행되기 전에 CA Access Control 로 제어가 전달됩니다. CA Access Control 은 해당 사용자에게 서비스를 허용할지 여부를 결정합니다.

예를 들어 사용자는 CA Access Control 에 의해 보호되는 리소스에 액세스하려고 시도할 수 있습니다. 이러한 액세스를 요청하면 리소스를 열기 위해 커널에 대한 시스템 호출이 생성됩니다. CA Access Control 은 해당 시스템 호출을 차단하고 액세스 권한을 부여할지 여부를 결정합니다. 사용 권한이 부여되면, CA Access Control 은 일반 시스템 서비스로 제어를 넘겨줍니다. CA Access Control 이 사용 권한을 거부하면, 시스템 호출을 활성화한 프로그램에 표준 사용 권한-거부 오류 코드를 반환하고 시스템 호출이 종료됩니다.

데이터베이스에 정의된 액세스 규칙 및 정책을 기준으로 결정이 이루어집니다. 데이터베이스는 접근자와 리소스라는 두 가지 개체 유형을 설명합니다. *접근자*는 사용자 및 그룹이며, *리소스*는 파일과 서비스와 같은 보호할 개체입니다. 데이터베이스의 각 레코드는 접근자 또는 리소스에 대해 설명합니다.

각 개체는 동일한 유형의 개체 모음인 클래스에 속합니다. 예를 들어, TERMINAL 은 CA Access Control 로 보호되는 터미널(워크스테이션) 개체를 포함하는 클래스입니다.

클래스 활성화

클래스 상태에 대한 정보(클래스가 활성 또는 비활성인지 여부)는 데이터베이스에 저장됩니다. 리소스에 액세스하려는 모든 시도는 CA Access Control 에 의해 차단되고 데이터베이스에서 상태를 확인합니다. 클래스가 비활성일 경우, 더 이상 권한 부여를 확인하지 않고 액세스가 허용됩니다.

CA Access Control 은 엔진이 시작되고 사용자가 클래스 활성 상태를 변경할 때 활성 클래스 목록을 작성합니다. 클래스가 비활성 상태이면 리소스에 대한 액세스가 차단되지 않아 오버헤드가 감소됩니다.

접근자 요소

각 사용자는 데이터베이스에 있는 사용자 레코드가 내부 메모리에 반영된 형태인 *접근자 요소*(ACEE)로 나타납니다. CA Access Control 은 로그인 프로세스 중에 접근자 요소를 작성합니다. 접근자 요소는 사용자 프로세스와 연결되어 있습니다. 프로세스가 CA Access Control 로 보호되는 시스템 서비스를 요청하거나 암시적으로 리소스 액세스를 요청할 때마다 CA Access Control 이 해당 리소스 레코드에 액세스합니다. 그런 다음 사용자의 보안 수준, 모드 및 그룹과 같이 이미 작성된 접근자 요소의 정보로 사용자가 리소스에 액세스할 수 있는지 여부를 결정합니다.

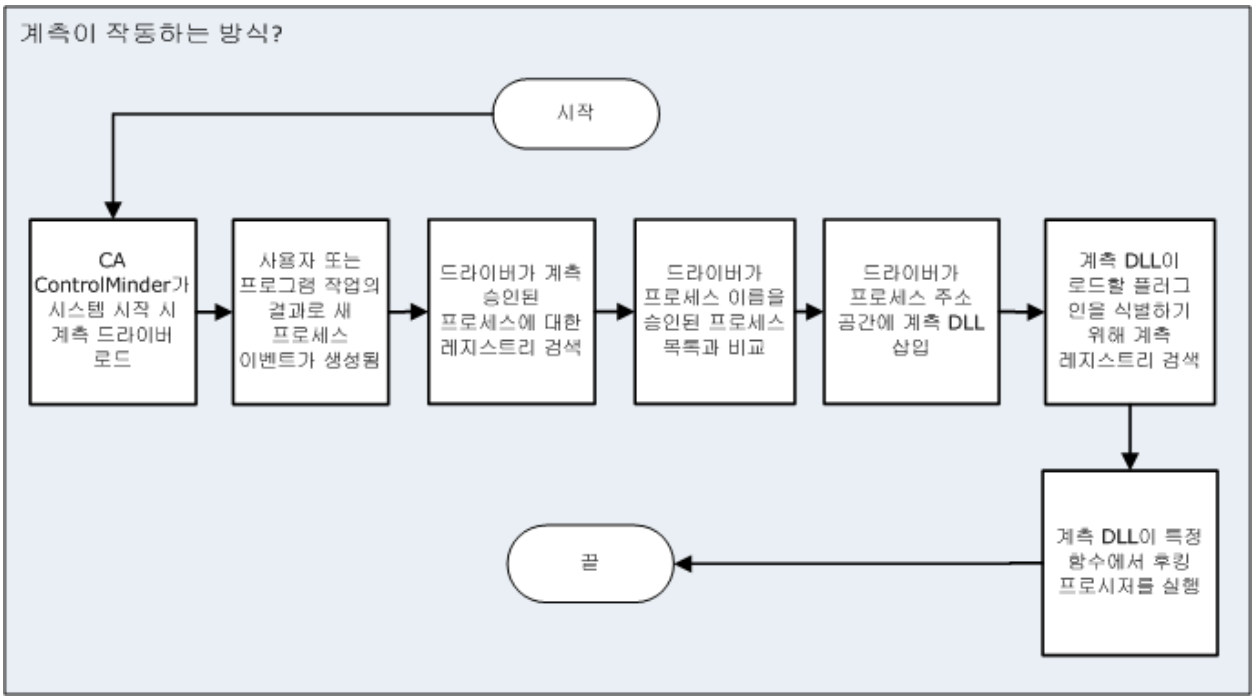
계측 작동 방식?

CA Access Control 은 *계측*을 사용하여 응용 프로그램의 실행 흐름을 모니터링하고, 추적하고, 변경할 수 있습니다. CA Access Control 은 계측을 사용하여 시스템 프로세스를 모니터링하고, 응용 프로그램 주소 공간에서 적절한 모듈을 차단 및 구현합니다.

계측 프로세스는 *커널 계측* 단계와 *사용자 모드 계측* 단계의 두 개 단계로 구성됩니다.

참고: 커널 및 사용자 모드 차단에 대한 자세한 내용은 *계정 보호* 장을 참조하십시오. 계측에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

다음 다이어그램은 계측 프로세스를 설명합니다.



커널 계측 단계에서 CA Access Control 은 다음을 수행합니다.

1. CA Access Control 은 시스템이 시작할 때 계측 드라이버(cainstrm.sys)를 로드합니다.
2. 사용자 또는 프로그램 작업의 결과로 새 프로세스 이벤트가 생성됩니다.
3. 지정된 간격으로 계측 드라이버는 레지스트리 하이브에서 계측 승인된 프로세스를 검색합니다.

   계측 ApplyonProcesses 레지스트리 키를 사용하여 계측 승인된 프로세스의 목록을 지정합니다. 계측 레지스트리 키에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.
4. CA Access Control 이 새 프로세스 이벤트를 식별하면 승인된 프로세스의 목록에서 프로세스 이름을 검색합니다. 발견한 경우 드라이버는 계측 dll 을 프로세스 주소 공간에 넣습니다.

사용자 모드 계측 단계에서 CA Access Control 은 다음을 수행합니다.

1. 계측 dll 은 계측 레지스트리 하이브를 검색하여 프로세스 주소 공간에 로드할 플러그 인을 식별하고 다음 중 하나를 수행합니다.
   - 나열된 모든 플러그 인을 프로세스 메모리 공간에 로드합니다. 2 단계에서 계속합니다.
   - 나열된 플러그 인이 없는 경우 자신을 언로드합니다.
2. CA Access Control 은 Microsoft Detours 라이브러리를 사용하여 각 플러그 인이 포함하고 있는 특정 기능에 기초하여 후킹 프로시저를 실행합니다.

   Microsoft Detours 는 Win32 함수를 계측하는 라이브러리입니다. Microsoft Detours 에 대한 자세한 내용은 *Microsoft Detours 웹 사이트* (http://www.microsoft.com/about/legal/en/us/intellectualproperty/iplicensing/programs/detours.aspx)를 참조하십시오.

기본 Windows 보안 확장

다음과 같은 CA Access Control 기능은 기본 보안을 확장합니다.

슈퍼 사용자 계정 제한 사항

운영 체제를 관리하는 사용자는 일반적으로 UNIX 시스템의 root 계정 및 Windows 시스템의 Administrator 계정과 같이 시스템 설치 중에 자동으로 생성되는 미리 정의된 계정의 구성원입니다. 미리 정의된 각 계정은 일련의 특정 시스템 기능을 실행하기 위해 존재합니다.

root 또는 Administrator 역할을 수행하는 사용자는 사용자 작성, 삭제 및 수정에서 서버 잠금, 재구성 및 종료에 이르는 광범위한 작업을 수행할 수 있습니다.

이러한 운영 체제에서 가장 큰 보안 위험 중 하나는 권한 없는 사용자가 이러한 계정을 제어할 수 있다는 점입니다. 이런 경우 권한 없는 사용자로 인해 시스템에 엄청난 손상이 발생할 수 있습니다.

CA Access Control 을 통해 이러한 계정에 부여되는 권한을 제한하고 이러한 계정이 구성원으로 포함된 사용자 그룹의 구성원인 사용자 권한을 제한할 수 있습니다. 이 기능을 통해 운영 시스템의 취약성을 보완할 수 있습니다.

CA Access Control 관리자

CA Access Control 을 설치할 때 하나 이상의 CA Access Control 관리자 이름을 지정하라는 요청을 받았습니다. CA Access Control 관리자는 규칙 데이터베이스를 일부 수정하거나 전부 수정할 수 있는 권한을 가지고 있습니다. 모든 권한을 가진 관리자는 한 명 이상이어야 합니다. 관리자는 원하는 대로 액세스 규칙을 수정하거나 작성할 수 있으며, 다른 수준의 관리자를 지정할 수 있습니다.

시스템 사용자를 정의한 후에 다른 사용자에게 ADMIN 특성을 할당하여 관리 권한을 할당할 수 있습니다.

참고: ADMIN 특성을 가진 사용자는 강력한 권한을 보유합니다. 따라서 ADMIN 사용자의 수는 엄격히 제한되어야 합니다. CA Access Control 보안 관리자를 한 명 이상 설정한 후 슈퍼 사용자에서 ADMIN 특성을 제거하여 기본 슈퍼 사용자와 ADMIN 의 역할을 구분하는 것도 좋습니다.

데이터베이스를 관리할 수 있는 권한을 가진 사용자가 항상 한 명 이상 필요하므로, CA Access Control 에서 ADMIN 특성을 가진 마지막 사용자는 삭제할 수 없습니다.

CA Access Control 관리자 중 한 명 이상이 워크스테이션에서 다른 호스트를 관리할 것으로 예상할 경우, 해당 호스트의 데이터베이스에 있는 규칙이 워크스테이션에서 READ 및 WRITE 액세스를 해당 관리자에게 부여하는지 확인하십시오.

하위 관리

CA Access Control 에는 *하위 관리* 기능이 있습니다. 이 기능을 통해 관리자는 일반 사용자가 특정 클래스를 관리하게 하는 특정 권한을 부여할 수 있습니다. 이러한 사용자를 하위 관리자라고 합니다.

예를 들어, 특정 사용자에게 사용자와 그룹만 관리하도록 허용할 수 있습니다.

또한 특정 클래스에 대한 액세스를 허용하면서 해당 클래스의 지정된 레코드에 대한 액세스도 허용함으로써 더 높은 수준의 하위 관리를 지정할 수 있습니다.

일반 사용자를 위한 관리자 권한

CA Access Control 은 일반 사용자(관리자가 아닌 사용자)에게 필요한 권한을 부여하여 해당 사용자가 관리자 그룹의 구성원이 아니더라도 관리 작업을 수행할 수 있습니다. 이런 세분화된 방법으로 관리 권한을 부여하여 작업을 위임하는 기능은 CA Access Control 의 중요한 장점입니다.

- SUDO 클래스의 레코드에는 사용자가 빌린 권한으로 스크립트를 실행할 수 있는 명령 스크립트가 저장되어 있습니다.
- 데이터 속성 값이 명령 스크립트입니다. 값에 선택적인 스크립트 매개 변수 값을 추가함으로써 값을 수정할 수 있습니다.
- SUDO 클래스의 각 레코드는 다른 사용자로부터 사용 권한을 빌려 올 수 있는 명령을 식별합니다.
- SUDO 클래스 레코드의 키는 SUDO 레코드 이름입니다. 이 이름은 사용자가 SUDO 레코드에서 명령을 실행할 때 명령 이름 대신 사용됩니다.

향상된 파일 보호

CA Access Control 은 논리적 파일 이름 형식과 절대 파일 이름 형식을 모두 지원합니다. 예를 들어 foo.txt 파일이 D 드라이브의 \tmp 디렉터리에 있고 논리적 이름 "D:"가 실제 디스크 1, 파티션 0 에 할당된 경우, CA Access Control 데이터베이스에 파일을 정의하기 위해 논리적 파일 이름이나 절대 파일 이름을 사용할 수 있습니다.

```
nr file D:\tmp\foo.txt
```

또는

```
nr file \Device\HardDisk1\Partition1\tmp\foo.txt
```

참고: 두 번째 형식을 사용할 경우 디스크의 논리적 이름이 변경되더라도 파일은 계속 보호됩니다. 절대 파일 이름 형식은 CA Access Control 일반 파일 보호에서도 지원됩니다.

CA Access Control 은 지원되는 Windows 운영 체제에서 현재 사용되는 모든 파일 시스템을 보호합니다. 가장 많이 사용되는 두 가지 파일 시스템은 Windows 파일 시스템(NTFS)과 파일 할당 테이블(FAT)입니다. CA Access Control 은 CDFS(특히 CD 용 파일 시스템)도 지원합니다.

CA Access Control 은 FAT(파일 할당 표)에 대한 완전한 보안 솔루션을 제공하고, NTFS 및 CDFS 를 포함한 다른 파일 시스템에 대한 보안 층을 추가로 제공합니다.

일반 파일 보호

CA Access Control 은 논리적 파일 이름 형식과 절대 파일 이름 형식을 모두 지원합니다. 절대 파일 이름 형식은 CA Access Control 일반 파일 보호에서도 지원됩니다.

일반 파일 보호 기능을 통해 지정된 와일드카드 패턴(정규 표현식)과 일치하는 모든 파일을 보호할 수 있습니다. 지정 와일드카드 패턴과 일치하는 이름을 가진 모든 리소스는 지정한 일반 액세스 규칙에 의해 보호됩니다. CA Access Control 에서는 파일을 전체적으로 보호할 수 있습니다.

리소스가 둘 이상의 일반 액세스 규칙과 일치할 경우 CA Access Control 은 파일과 가장 근접하게 일치하는 규칙을 선택합니다.

일반 파일 보호의 경우, 보호가 필요한 여러 개의 파일을 보호하기 위해 5 개 이하의 보안 규칙을 정의해야 합니다.

암호 보호

기본 Windows 보안은 여러 가지 방법으로 암호를 보호하고 암호 품질을 지정할 수 있습니다. Windows 는 다음과 같은 기능을 제공합니다.

- 암호의 최대 사용 기간 제한
- 암호의 최소 길이 제한
- 최대 24 개의 사용자 암호 생성
- 반복되는 로그인 실패 시 계정 잠금
- 암호 변경을 위해 사용자가 Windows 에 로그온하도록 요구

CA Access Control 은 동일한 규칙을 적용하지만 자체의 고유한 메커니즘을 사용합니다. 또한 CA Access Control 은 메인프레임 컴퓨터와의 양방향 암호 동기화를 구현합니다.

향상된 암호 보호

기본 Windows 보안은 상당한 수준의 사용자 암호 보호 (페이지 28)를 제공합니다. 그러나 CA Access Control 은 암호 보호 기능을 현저히 확장하여 해커가 암호를 알아내는 데 성공할 확률을 크게 줄여줍니다.

CA Access Control 을 사용할 때 사용자가 더욱 안전한 암호를 선택하도록 하는 추가 규칙을 작성할 수 있습니다. 예를 들어, 사용자에게 일정한 수 이상의 알파벳, 숫자, 특수 문자, 소문자 또는 대문자를 선택하도록 요구할 수 있습니다. 또한 사용자가 선택한 새 암호의 일부에 기존 암호가 포함되거나 기존 암호의 일부가 새 암호로 사용되지 않도록 지정할 수 있습니다.

프로그램 경로 지정

*프로그램 경로 지정*은 특정 프로그램을 통해서만 파일에 액세스하도록 하는 파일 관련 액세스 규칙입니다. 프로그램 경로 지정을 통해 중요한 파일의 보안이 상당히 향상됩니다. CA Access Control 에서는 프로그램 경로 지정을 사용하여 시스템의 파일에 대한 보호를 추가로 제공할 수 있습니다.

B1 보안 수준 인증

CA Access Control 에는 보안 수준, 보안 범주, 보안 레이블 등의 B1 "Orange Book" 기능이 포함되어 있습니다.

- 데이터베이스의 접근자와 리소스에 *보안 수준*을 할당할 수 있습니다. 보안 수준은 1 에서 255 사이의 정수입니다. 접근자는 보안 수준이 리소스에 할당된 보안 수준과 같거나 높은 경우에만 리소스에 액세스할 수 있습니다.
- 데이터베이스의 접근자와 리소스는 하나 이상의 *보안 범주*에 속할 수 있습니다. 접근자는 리소스에 할당된 모든 보안 범주에 속해 있을 경우에만 리소스를 액세스할 수 있습니다.
- *보안 레이블*은 특정 보안 수준을 0 개 이상의 보안 범주 집합에 연결하는 이름입니다. 사용자에게 보안 레이블을 할당하면 보안 수준과 보안 레이블에 관련된 보안 범주가 사용자에게 모두 부여됩니다.

참고: B1 Orange Book 기능에 대한 자세한 내용은 *구현 안내서*를 참조하십시오.

감사 절차 설정

CA Access Control 은 데이터베이스에 정의되어 있는 감사 규칙에 따라 액세스 거부 및 허용 이벤트에 대한 감사 레코드를 유지합니다. 특정 이벤트의 기록 여부는 다음 규칙에 따라 결정됩니다.

- 모든 접근자와 리소스에는 액세스 성공, 실패 또는 이 모두를 로그 파일에 기록할지 여부를 지정할 수 있는 AUDIT 속성이 있습니다. 또한 접근자의 AUDIT 속성은 로그인 성공, 실패 또는 이 모두를 로그 파일에 기록할지 여부를 지정할 수 있습니다.
- 리소스 또는 접근자에 AUDIT(ALL) 특성이 있는 경우, CA Access Control 에 의해 보호되는 해당 리소스에 대한 모든 이벤트는 액세스 성공 여부와 상관없이 로그 파일에 기록됩니다.
- CA Access Control 에 의해 보호되는 리소스에 성공적으로 액세스하고 사용자 또는 리소스에 AUDIT(SUCCESS)가 있는 경우 이벤트가 로그 파일에 기록됩니다.
- CA Access Control 에 의해 보호되는 리소스에 액세스하지 못하고 사용자 또는 리소스에 AUDIT(FAIL)가 있는 경우 이벤트가 로그 파일에 기록됩니다.

AUDITOR 특성이 할당된 사용자인 시스템 감사자만 사용자 및 리소스에 할당된 감사 특성의 변경과 같은 감사 작업을 수행할 수 있습니다.

리소스가 경고 모드에 있으면 이 리소스에 대해 액세스 규칙을 위반하는 모든 액세스는 경고 모드 감사 레코드에 기록됩니다. 이 레코드에는 CA Access Control 이 리소스에 대한 액세스를 허용했다는 내용이 포함됩니다.

감사 레코드는 *감사 로그*(seos.audit)라는 파일을 구성합니다. 감사 로그의 위치는 오류 로그의 위치와 마찬가지로 레지스트리에서 지정됩니다.

감사 로그(및 오류 로그)는 다음 레지스트리 키에서 지정됩니다.

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\logmgr
```

감사 로그는 바이너리 파일로, 편집하거나 변경할 수 없습니다. 그러나 CA Access Control 끝점 관리를 사용하여 기록된 이벤트를 보고, 시간 제한 또는 이벤트 유형 등을 기준으로 이벤트를 필터링할 수는 있습니다. 또한 seaudit 유틸리티를 사용하여 동일한 작업을 수행할 수도 있습니다.

이벤트를 나중에 검사할 수 있도록 오래된 감사 로그와 오류 로그를 보관(백업)하는 방법을 고려하십시오.

Unicenter TNG 로 감사 이벤트 전송

Unicenter TNG 와의 통합은 설치할 때 설정됩니다.

감사 데이터를 Unicenter TNG 로 전송하도록 선택하거나 Unicenter TNG 에서 CA Access Control 의 시작을 허용하도록 선택하거나, 두 가지 작업을 모두 선택할 수 있습니다. 두 가지 옵션은 상호 연관성이 없습니다.

첫 번째 옵션을 선택하면 다음 하위 키에 있는 레지스트리 값이 설정됩니다.

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\UCTNG
```

Integration 값은 1(yes)로 설정되고 EvtManagerServer 값은 Unicenter TNG 호스트 이름을 문자열 값으로 받습니다.

Unicenter TNG 에 전달된 감사 이벤트는 Unicenter Enterprise Management\Enterprise Managers\Windows NT\Event 창에 있는 콘솔 로그에 표시됩니다.

감사 이벤트	표시 색상	심각도
성공	파란색	-S
거부됨	주황색	F

감사 이벤트	표시 색상	심각도
실패	주황색	F
경고	파란색	W
CA Access Control 이 중지됨(감사 중단)	파란색	-I
CA Access Control 이 시작됨(감사 시작)	파란색	-I

두 번째 옵션을 사용하면 "관리 개체" 창에 있는 TCP/IP 네트워크를 나타내는 아이콘를 가리키고 마우스 오른쪽 버튼을 클릭하면 나타나는 메뉴에서 CA Access Control 을 선택하여 Unicenter WorldView 메뉴에서 CA Access Control 을 시작할 수 있습니다.

CA Access Control 에서는 이벤트에 대한 다음 정보도 보냅니다.

- 제품 이름(CA Access Control + 버전 번호)
- 사용자 이름
- 터미널 이름
- 클래스 이름
- 리소스 이름
- 프로세스 이름
- 이벤트 시간
- CA Access Control 감사 형식의 전체 감사 메시지

이벤트 유형에 따라서는 사용자 이름, 터미널 이름, 클래스 이름, 리소스 이름 및 프로세스 이름 필드를 보내지 않을 수도 있습니다.

구성 요소

CA Access Control 에는 데이터베이스(seosdb), 두 개의 드라이버(seosdrv 및 drveng), 여러 서비스(Watchdog, 에이전트, 엔진(seosd), 정책 모델 및 작업 위임 등) 및 그래픽 사용자 인터페이스가 포함되어 있습니다.

데이터베이스

데이터베이스에는 다음과 같은 요소에 대한 정의가 있습니다.

- 조직의 사용자 및 그룹
- 보호가 필요한 시스템 리소스
- 시스템 리소스에 대한 사용자 및 그룹 액세스를 제어하는 규칙

드라이버

드라이버는 다음 작업을 수행하여 모든 CA Access Control 파일과 레지스트리 키를 보호합니다.

- 파일 또는 레지스트리 키 열기, 프로세스 종료 및 네트워크 활동 수행의 모든 요청 차단
- 이러한 요청을 CA Access Control 엔진에 전달하고 요청의 허용 또는 거부에 대한 엔진의 결정 수신
- 결정을 운영 체제의 기존 시스템 호출로 전달하고, 드라이버로부터 수신한 답변에 따라 처리 작업을 계속 수행

서비스

Watchdog

Watchdog 은 다른 CA Access Control 서비스가 실행 중인지를 계속해서 검사합니다. 드물지만 Watchdog 이 중지된 다른 서비스를 발견한 경우 서비스가 즉시 다시 시작됩니다.

에이전트

또한 에이전트는 다음과 같은 작업을 수행합니다.

- TCP/IP 에서 독점 응용 프로그램 프로토콜을 통해 CA Access Control 클라이언트와 통신
- CA Access Control 사용자를 위해 보안 관리

엔진

엔진은 다음 작업을 담당합니다.

- 모든 데이터베이스 업데이트 제어를 포함한 데이터베이스 관리
- 드라이버와 에이전트로부터 받은 액세스 요청의 허가 여부 결정
- Watchdog 서비스가 실행 중인지 확인하고, Watchdog 의 실행이 중단된 것으로 인식되면 Watchdog 재시작

엔진은 데이터베이스 액세스 요청을 처리*하고* 액세스를 결정하여 효율적인 서비스를 수행합니다.

정책 모델

수십 또는 수백 개의 데이터베이스를 개별적으로 관리하는 것은 실용적이지 않습니다. 따라서 CA Access Control 은 한 대의 컴퓨터에서 여러 대의 컴퓨터를 관리하도록 허용하는 구성 요소인 정책 모델 서비스를 제공합니다. 정책 모델 서비스를 사용하는 것은 선택사항이지만 큰 사이트에서 이 서비스를 사용하면 관리 작업이 상당히 간단해집니다.

정책 모델 서비스와 함께 PMDB(정책 모델 데이터베이스)를 사용합니다. 다른 CA Access Control 데이터베이스와는 달리 PMDB 에는 사용자, 그룹, 보호된 리소스, 리소스에 대한 액세스를 제어하는 규칙 등이 포함됩니다. 또한 PMDB 에는 구독자 스테이션 목록도 포함됩니다. 구독자 스테이션은 PMDB 에 대한 모든 변경 내용이 구독자 데이터베이스로 자동으로 보내질 수 있도록 PMDB 에 연결된 스테이션입니다.

조직에 대한 기본 보안 정책을 만들고 필요한 모든 규칙을 하나의 정책 모델 데이터베이스에 구현할 수 있습니다. 구독자는 Windows 및 UNIX 스테이션 등을 포함하여 최소한의 관리 노력으로 통일된 규칙을 유지할 수 있습니다.

시스템 또는 보안 관리자가 PMDB 를 업데이트합니다. 그러면 PMDB 는 PMDB 의 모든 업데이트를 해당구독자에게 배치 모드로 전파하여 관리자가 다른 작업을 수행할 수 있도록 합니다.

PMDB 에는 또 다른 PMDB 와 로컬 데이터베이스 등 두 가지 유형의 데이터베이스가 있을 수 있습니다. 이 PMDB 에는 데이터베이스 업데이트를 전파할 구독자 목록도 포함됩니다. 이 기능을 사용하면 PMDB 계층을 작성할 수 있습니다. 로컬 데이터베이스는 스테이션에 정의된 사용자, 그룹 및 리소스를 보호하는 데 사용할 수 있습니다.

selang

명령행 언어인 selang 은 CA Access Control 의 모든 기능을 실행합니다. selang 명령을 사용하려면, 명령 프롬프트 창을 열고 selang 을 시작하십시오. selang 은 스크립트에서도 사용할 수 있습니다.

selang 및 해당 명령에 대한 자세한 내용은 *참조 안내서*의 "selang 명령 언어" 장을 참조하십시오.

끝점 관리

CA Access Control 에서는 기업의 리소스를 관리하고 리소스에 액세스할 수 있는 사용자를 제어하게 하는 두 가지 방법을 제공합니다.

- **selang** - CA Access Control 명령 언어입니다.

  selang 명령 언어를 사용하면 CA Access Control 데이터베이스에서 정의를 만들 수 있습니다. selang 명령 언어는 명령 정의 언어입니다.

  참고: selang 사용에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

- **CA Access Control 끝점 관리** - 끝점 관리 인터페이스입니다.

  이 웹 기반 인터페이스에서는 중앙 관리 서버를 통해 원격 끝점을 관리할 수 있습니다.

  참고: CA Access Control 끝점 관리 설치에 대한 자세한 내용은 *구현 안내서*를 참조하십시오.

제 3 장: 사용자 및 그룹 관리

이 섹션은 다음 항목을 포함하고 있습니다.

사용자 및 그룹

CA Access Control 에서 모든 작업이나 액세스 시도는 요청을 제출할 책임이 있는 사용자를 위해 수행됩니다. 따라서 시스템의 모든 프로세스는 특정 사용자 이름과 연결되어 있습니다. CA Access Control 은 사용자 이름으로 사용자를 식별합니다.

*사용자*는 배치 또는 데몬 프로그램의 소유자가 될 수 있거나 로그온할 수 있는 개인입니다. CA Access Control 에서 모든 액세스 시도는 사용자가 수행합니다. CA Access Control 은 CA Access Control 데이터베이스 및 엔터프라이즈 사용자 저장소에 있는 사용자 정보를 사용할 수 있습니다. 사용자 정보는 데이터베이스의 USER 레코드나 XUSER 레코드에 저장됩니다.

참고: *엔터프라이즈 사용자 저장소*는 사용자나 그룹을 저장하는 운영 체제의 저장소입니다(예: UNIX 의 /etc/passwd 및 /etc/groups 또는 Windows 의 Active Directory)

*그룹*은 사용자 집합입니다. 그룹은 해당 그룹의 사용자에 대한 공통 액세스 규칙을 정의합니다. 그룹은 중첩될 수 있습니다(다른 그룹에 속함). CA Access Control 은 CA Access Control 데이터베이스와 엔터프라이즈 사용자 저장소에서 그룹 정보를 사용할 수 있습니다. 일반적으로 그룹을 작성한 다음 database_administrators 와 같은 역할을 기준으로 사용자를 이 그룹에 할당합니다.

사용자 레코드는 주요 접근자 레코드입니다. CA Access Control 에서 그룹을 사용하는 주요 목적은 그룹의 모든 사용자에게 액세스 권한을 한 번에 할당하기 위해서입니다. 액세스 권한을 한 번에 할당하는 것이 각 사용자에게 개별적으로 할당하는 것보다 더 간편하고 오류가 덜 발생합니다.

접근자 정보가 저장된 위치

CA Access Control 에서 사용하는 사용자 및 그룹 정보는 CA Access Control 데이터베이스와 호스트 운영 체제에 저장됩니다. 호스트 운영 체제 정보 저장소를 *엔터프라이즈 사용자 저장소* 또는 *엔터프라이즈 저장소*라고 합니다. 기본적으로 CA Access Control 은 엔터프라이즈 저장소를 사용하지 않도록 구성됩니다. 그러나 CA Access Control 이 데이터베이스에 정의된 사용자나 그룹을 찾을 수 없는 경우 엔터프라이즈 저장소에 정의된 사용자 및 그룹 구성원을 검색하고 해당 정보를 사용하도록 구성할 수 있습니다.

참고: CA Access Control 에서는 엔터프라이즈 저장소에서 정보를 사용할 뿐만 아니라 네이티브 환경에서 selang 명령을 사용할 경우 엔터프라이즈 저장소에 쓰기도 합니다.

권한 부여를 확인할 때 CA Access Control 에서는 항상 자체 데이터베이스에 정의된 접근자를 확인한 후 엔터프라이즈 저장소를 확인합니다. CA Access Control 데이터베이스에 정의된 사용자와 동일한 이름을 가진 엔터프라이즈 사용자가 있는 경우에는 엔터프라이즈 사용자는 CA Access Control 에서 무시됩니다.

CA Access Control 이 사용자 레코드를 찾는 방법

사용자가 로그인할 때 CA Access Control 은 사용자와 관련된 레코드를 찾을 때까지 다음 순서대로 검색을 수행합니다.

1. CA Access Control 은 데이터베이스에 정의된 사용자를 검색합니다.
2. CA Access Control 은 캐시에서 해당 이름을 가진 엔터프라이즈 사용자를 검색합니다.

   네트워크의 연결이 끊긴 경우 운영 체제(OS)는 OS 에 캐시된 자격 증명을 사용하여 사용자가 로그인할 수 있게 합니다. CA Access Control 캐시의 목적은 이러한 상황에서 CA Access Control 이 엔터프라이즈 사용자의 레코드를 사용할 수 있게 하는 것입니다.
3. CA Access Control 은 운영 체제를 사용하여 엔터프라이즈 사용자 저장소에서 해당 이름을 가진 사용자를 검색합니다.
4. CA Access Control 이 데이터베이스나 엔터프라이즈 저장소에서 사용자와 관련된 레코드를 찾지 못하면 CA Access Control 은 사용자에게 _undefined USER 레코드에 있는 속성을 할당합니다.

엔터프라이즈 사용자 저장소와 통합

일반적으로 CA Access Control 에서 엔터프라이즈 사용자 저장소에 정의된 그룹과 사용자를 사용하도록 구성합니다.

CA Access Control 을 이와 같이 구성하면 기본적으로 엔터프라이즈 사용자 저장소를 참조하는 액세스 규칙이 작성되거나 사용자가 운영 체제에 로그인할 때 CA Access Control 은 데이터베이스에서 기존 레코드가 없는 경우 해당 사용자나 그룹에 대한 레코드를 생성합니다. 이러한 레코드에는 XUSER(엔터프라이즈 사용자용) 또는 XGROUP(엔터프라이즈 그룹용) 클래스가 있습니다. 레코드에는 CA Access Control 에서 액세스 규칙을 적용하기 위해 필요한 속성이 저장되어 있습니다. CA Access Control 에서 필요에 따라 레코드를 생성하므로 레코드를 관리할 필요가 없습니다.

CA Access Control 이 엔터프라이즈 사용자 저장소에서 가져오는 엔터프라이즈 사용자 또는 그룹의 속성은 이름과 그룹 구성원 속성뿐입니다.

엔터프라이즈 저장소에서 접근자 관리에 대한 지침

엔터프라이즈 사용자 저장소에서 접근자를 관리하려면 다음 절의 지침을 고려해야 합니다.

데이터베이스에 정의해야 하는 사용자 및 그룹

CA Access Control 은 일부 사용자와 그룹을 엔터프라이즈 사용자 저장소가 아니라 데이터베이스에 저장하도록 합니다. 해당되는 정보는 다음과 같습니다.

- 미리 정의된 사용자 (페이지 44)
- 미리 정의된 그룹 (페이지 45)
- CA Access Control 관리자
- 프로필 그룹
- 논리적 사용자

엔터프라이즈 사용자 사용 제한 사항

CA Access Control 에서는 엔터프라이즈 사용자 사용에 대해 다음 제한 사항을 적용합니다.

- 데이터베이스에 동일한 이름을 가진 사용자가 정의되어 있는 경우에는 CA Access Control 에서 엔터프라이즈 사용자를 생성하거나 참조할 수 없습니다.
- selang AC 환경을 사용하여 엔터프라이즈 사용자를 작성, 삭제 또는 수정할 수 없습니다.
- 엔터프라이즈 사용자를 논리적 사용자로 사용할 수 없습니다.
- 기본적으로 엔터프라이즈 사용자 저장소에 사용자가 이미 정의되어 있지 않으면 CA Access Control 에 엔터프라이즈 사용자를 만들 수 없습니다. 그러나 UNIX 시스템에서는 이 동작을 활성화 또는 비활성화할 수 있습니다.

추가 정보:

UNIX 에서 XUSER 레코드를 생성하기 전에 엔터프라이즈 저장소 검사 활성화 또는 비활성화 (페이지 40)

엔터프라이즈 그룹 사용 제한 사항

CA Access Control 에서는 엔터프라이즈 그룹 사용에 대해 다음 제한 사항을 적용합니다.

- selang AC 환경에서는 엔터프라이즈 그룹을 작성하거나 삭제할 수 없습니다.
- selang AC 환경에서는 엔터프라이즈 그룹 구성원을 변경할 수 없습니다.
- 엔터프라이즈 그룹을 프로필 그룹 (페이지 46)으로 사용할 수 없습니다.

엔터프라이즈 사용자 및 그룹 사용 활성화 또는 비활성화

기본적으로 CA Access Control 에서는 엔터프라이즈 사용자 저장소에 정의된 그룹과 사용자를 사용할 수 없지만 사용하도록 CA Access Control 을 활성화할 수 있습니다. 이전 CA Access Control 버전과의 호환성이 필요한 경우가 아니면 이 기능을 활성화하는 것이 좋습니다.

CA Access Control 에서 엔터프라이즈 사용자와 그룹을 사용하게 하려면 구성 설정 osuser_enabled 를 'yes'로 설정하십시오. 이 동작을 비활성화하려면 osuser_enabled 값을 'no'로 설정하십시오.

예: Windows 에서 엔터프라이즈 사용자 및 그룹 사용 활성화

다음 레지스트리 설정은 Windows 에서 엔터프라이즈 사용자 및 그룹의 사용을 활성화합니다.

- 키: HKLM\SOFTWARE\ComputerAssociates\AccessControl\OS_user
- 이름: osuser_enabled
- 유형: REG_DWORD
- 값: yes

예: UNIX 에서 엔터프라이즈 사용자 및 그룹 사용 활성화

다음 명령은 CA Access Control 을 중지하고 UNIX 에서 엔터프라이즈 사용자 및 그룹 사용을 활성화한 다음 CA Access Control 을 다시 시작합니다.

```
secons -s
seini -s OS_User.osuser_enabled yes
seload
```

엔터프라이즈 사용자 로그인 시 XUSER 레코드 생성 활성화 또는 비활성화

CA Access Control 은 엔터프라이즈 사용자를 사용하도록 활성화된 경우 기본적으로 해당 사용자가 로그인할 때 사용자에 대한 레코드(XUSER 클래스에서)를 생성합니다. 매일 같은 시간에 수천 명의 사용자가 로그인하는 경우 이 작업을 수행하지 않을 수 있습니다.

사용자가 로그인할 때 CA Access Control 에서 XUSER 레코드를 생성하지 않게 하려면 구성 설정 create_user_in_db 값을 0(영)으로 변경합니다. 이 동작을 다시 활성화하려면 값을 1(일)로 설정합니다.

예: Windows 에서 엔터프라이즈 사용자 로그인 시 XUSER 레코드의 자동 생성 비활성화

다음 레지스트리 설정은 Windows 에서 CA Access Control 의 엔터프라이즈 사용자 레코드 자동 생성을 비활성화합니다.

- 키: HKLM\Software\ComputerAssociates\AccessControl\OS_user
- 이름: create_user_in_db
- 유형: REG_DWORD
- 값: 0

예: UNIX 에서 엔터프라이즈 사용자 로그인 시 XUSER 레코드의 자동 생성 비활성화

다음 명령은 CA Access Control 을 중지하고 UNIX 에서 XUSER 의 자동 생성을 비활성화한 다음 CA Access Control 을 다시 시작합니다.

```
secons -s
seini -s OS_User.create_user_in_db 0
seload
```

UNIX 에서 XUSER 레코드를 생성하기 전에 엔터프라이즈 저장소 검사 활성화 또는 비활성화

사용자가 엔터프라이즈 사용자 저장소에 정의되어 있지 않으면 CA Access Control 에서 엔터프라이즈 사용자를 생성할 수 없습니다. Windows 의 경우 사용자가 Windows 사용자 저장소에 정의되어 있지 않으면 CA Access Control 에서 엔터프라이즈 사용자를 생성할 수 없습니다. UNIX 의 경우 기본 동작이 Windows 와 반대입니다. 그러나 UNIX 의 경우 이 기본 동작을 활성화하거나 비활성화할 수 있습니다.

검사를 비활성화하여 일치하는 엔터프라이즈 사용자가 없을 경우 CA Access Control 에서 XUSER 레코드를 생성하게 하려면 구성 설정 verify_osuser 값을 0 으로 변경하십시오. 검사를 적용하려면 값을 1 로 설정합니다.

예: 엔터프라이즈 사용자 저장소 검사 없이 XUSER 레코드 생성 활성화

다음 명령 집합은 CA Access Control 을 중지하고 일치하는 엔터프라이즈 저장소 없이 XUSER 레코드 생성을 활성화한 다음 CA Access Control 을 다시 시작합니다.

```
secons -s
seini -s OS_User.verify_osuser 0
seload
```

Windows 에서 재사용된 엔터프라이즈 저장소 계정

*재사용 계정*은 삭제되었으나 같은 이름으로 다시 생성된 엔터프라이즈 저장소 사용자 또는 그룹입니다. 이는 예를 들어 사용자가 사임하는 경우 사용자 저장소에서 사용자를 제거한 다음 이전에 제거된 사용자와 같은 이름을 가진 새 사용자의 계정을 새로 만드는 것과 같습니다.

같은 이름을 가진 이전 계정에 부여했던 것과 동일한 액세스 권한을 새 접근자에게 부여할 필요가 없으므로 재사용 계정은 보안 문제를 일으킬 수 있습니다. 이 문제를 해결하기 위해 CA Access Control 권한 부여는 SID 를 기반으로 합니다. 따라서 기존 액세스 권한을 가진 삭제된 접근자와 동일한 이름을 가진 새 접근자를 만들 때 새 접근자는 이전 접근자의 이전 사용 권한을 자동으로 부여받지 않습니다.

중요! 재사용 계정 접근자는 이전 액세스 권한을 상속하지 *않습니다*. 그러나 데이터베이스 액세스 규칙에는 SID 가 아니라 접근자 이름이 표시되므로 이러한 규칙이 계속 적용되는 것처럼 보일 수 있습니다. 이를 확인하려면 secons -checkSID 명령을 사용합니다.

Windows 에서 재사용된 엔터프라이즈 계정 확인

데이터베이스 규칙이 관련되어 있는 엔터프라이즈 계정(사용자 또는 그룹)이 재사용(삭제된 후 같은 이름으로 생성)될 경우 이전 데이터베이스 규칙이 계속 새 계정에 적용되는 것처럼 보일 수 있습니다. 그러나 CA Access Control 권한 부여는 SID 를 기반으로 하므로 이러한 규칙은 더 이상 적용되지 않으며 새 그룹에 대한 새 규칙을 만들어야 합니다. 새 규칙을 만들려면 먼저 재사용 계정을 확인해야 합니다.

재사용 엔터프라이즈 계정을 확인하려면 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
secons -checkSID -users
secons -checkSID -groups
```

CA Access Control 은 포함되어 있는 모든 엔터프라이즈 사용자 계정(XUSER 레코드)을 통과한 다음 모든 그룹 계정(XGROUP 레코드)을 통과하여 엔터프라이즈 계정 SID 와 다른 SID 를 가진 계정을 식별합니다. 그런 다음 명명 규칙 *SID*(*accountName*)를 사용하여 이러한 계정 이름을 변경합니다.

이제 재사용 계정에 대한 새 규칙을 만들 수 있습니다.

참고: 재사용 사용자 계정은 사용자가 로그인하거나 리소스에 액세스를 시도할 때 이런 방식으로 확인됩니다. 엔터프라이즈 계정을 만들 때 secons -checkSID 명령을 예약 작업으로 실행하는 것이 좋습니다.

예: 재사용 그룹 계정

회사 ABCD 에는 엔터프라이즈 저장소에 *interns* 라는 그룹이 있습니다. 이 그룹에는 productA 작업을 하는 아홉 명의 구성원이 있습니다. 관리자는 그룹을 CA Access Control 에 인식시키고 다음과 같이 그룹 구성원이 액세스하는 데 필요한 파일 액세스 권한을 그룹에 할당합니다.

```
nxg interns owner(msmith)
auth file c:\products\productA\materials\* xgid(interns) access(all)
auth file c:\HR\interns\* xgid(interns) access(read)
```

interns 가 ABCD 보유를 완료할 때 엔터프라이즈 저장소 관리자는 그룹을 삭제합니다. 3 개월 이후 구성원이 여섯 명인 새로운 interns 그룹이 같은 이름으로 엔터프라이즈 저장소에 생성됩니다. CA Access Control 데이터베이스에 이전 규칙이 계속 존재하므로 새 *interns* 그룹이 이전 그룹의 사용 권한을 상속한 것처럼 보입니다. 그러나 이러한 규칙은 이전 interns 그룹에만 적용되므로 CA Access Control 관리자는 새 그룹에 대한 새 규칙을 만들어야 합니다.

이렇게 하려면 관리자가 다음과 같이 재사용 interns 계정을 식별 및 확인해야 합니다.

```
secons -checkSID -groups interns
```

이 명령은 XGROUP 리소스 및 리소스에 대한 액세스 규칙 참조의 이름을 "*SID*(*domain*\interns)"로 변경합니다. 이제 관리자는 productB 작업을 수행하는 새 interns 그룹에 대한 새 규칙을 만들 수 있습니다.

```
nxg interns owner(msmith)
auth file c:\products\productB\materials\* xgid(interns) access(all)
auth file c:\HR\interns\* xgid(interns) access(read)
```

참고: secons 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

데이터베이스 접근자

다음 절에 설명된 대로 사용자 관리 방법에 관계없이 일부 접근자는 CA Access Control 데이터베이스에 정의되어야 합니다.

미리 정의된 사용자

CA Access Control 에서는 삭제할 수 없는 다음 사용자를 미리 정의합니다.

+devcalc

(Windows) CA Access Control 이 위반 계산 프로세스인 devcalc 를 실행하는 데 사용하는 사용자 이름입니다.

_dms

고급 정책 관리 서버 구성 요소 데이터베이스(DMS, DH 구독기 및 DH 작성기)에 설치되는 _dms 사용자는 policyfetcher 및 devcalc 에서 DH 및 DMS 와 통신하는 데 사용됩니다.

nobody

nobody 사용자는 실제 사용자와 일치할 수 없는 사용자 레코드입니다. 이 레코드를 사용하여 사용자에게 관련 사용 권한을 제공하지 않는 규칙을 만듭니다. 예를 들어 *nobody* 를 리소스 소유자로 설정하면 모든 사용자가 레코드 소유와 관련된 사용 권한을 부여받을 수 없습니다.

+reportagent

CA Access Control 이 보고서 에이전트를 실행하는 데 사용하는 사용자 이름입니다.

_seagent

_seagent 는 CA Access Control 이 다음과 같은 내부 프로세스를 실행하는 데 사용하는 사용자 이름입니다.

- PMDB 프로세스, sepmdd
- (UNIX) 위반 계산 프로세스, devcalc
- 사용자 및 그룹 레코드 업데이트 종료 프로세스

_seagent 사용자는 SERVER 특성을 가지고 있습니다.

_sebuildla

(UNIX) _sebuildla 사용자는 CA Access Control 데몬인 seosd 에 대한 참조(lookaside) 데이터베이스를 만들기 위해 CA Access Control 이 sebuildla 유틸리티를 실행하는 데 사용하는 사용자 이름입니다.

_seoswd

(UNIX)_seoswd 는 파일 정보와 데이터베이스에서 신뢰하는 프로그램으로 정의된 프로그램의 디지털 서명을 모니터링하기 위해 seoswd watchdog 데몬을 실행하는 데 사용되는 사용자 이름입니다.

_undefined

_undefined 는 CA Access Control 에 정의되지 않은 모든 사용자를 나타냅니다. _undefined 를 사용하여 정의되지 않은 사용자를 ACL 에 포함할 수 있습니다.

미리 정의된 그룹

CA Access Control 에는 미리 정의된 그룹이 제공됩니다. _interactive 및 _network 그룹을 제외하고, 다른 그룹에 수행하는 것과 동일한 방법으로 이러한 그룹에 사용자를 추가할 수 있습니다.

_abspath

로그인 시 사용자가 _abspath 그룹에 있는 경우 프로그램을 호출하려면 절대 경로 이름을 사용해야 합니다.

_interactive

사용자는 액세스 시도의 목적으로만 _interactive 그룹의 구성원입니다. 사용자는 액세스를 시도하고 있는 리소스와 동일한 호스트에 로그인되어 있는 경우 _interactive 그룹의 구성원입니다. CA Access Control 에서는 _interactive 그룹의 구성원을 동적으로 자동 관리하므로 사용자가 구성원을 변경할 수 없습니다.

_network

이 그룹은 _interactive 의 보조 그룹입니다. 사용자는 액세스 목적으로만 _network 그룹의 구성원입니다. 사용자는 리소스가 속한 호스트와 다른 호스트에서 리소스에 액세스하고 있는 경우 _network 그룹의 구성원입니다. CA Access Control 에서는 _network 그룹의 구성원을 동적으로 자동 관리하므로 사용자가 구성원을 변경할 수 없습니다.

_restricted

_restricted 그룹의 사용자에 대한 모든 파일과 Windows 의 레지스트리 키는 CA Access Control 로 보호됩니다. 파일이나 Windows 레지스트리 키에 명시적으로 정의된 액세스 규칙이 없는 경우에는 액세스 권한이 해당 클래스(FILE 또는 REGKEY)의 _default 레코드로 처리됩니다.

참고: _restricted 그룹의 사용자는 작업을 수행할 충분한 권한을 가질 수 없습니다. _restricted 그룹에 사용자를 추가하려면 초기에 경고 모드를 사용해 보십시오.

_surrogate

사용자가 _surrogate 그룹의 구성원을 대리 사용자로 사용할 경우 CA Access Control 에서는 대리 사용자 작업의 감사 기록에 전체 추적을 기록하고 원래 사용자 이름으로 태그를 지정합니다.

예: selang 을 사용하여 _restricted 그룹에 사용자 추가

다음 selang 명령은 _restricted 그룹에 엔터프라이즈 사용자 john_smith 를 추가합니다.

```
joinx john_smith group(_restricted)
```

프로필 그룹

*프로필 그룹*은 CA Access Control 데이터베이스에 정의되는 사용자 속성 기본값이 포함된 그룹입니다. 사용자에게 프로필 그룹을 할당하면 해당 값이 이미 사용자에 대해 설정된 경우가 아니면 프로필 그룹이 해당 값을 사용자에게 제공합니다.

사용자를 만들 때 사용자의 프로필 그룹을 지정하거나 나중에 프로필 그룹에 사용자를 할당할 수 있습니다.

프로필 그룹을 사용하여 관리자는 해당 그룹에 할당된 새 사용자를 위한 특정 권한을 포함한 표준 설정을 효율적으로 작성할 수 있습니다. 이 설정은 사용자의 홈 디렉터리, 감사 속성, 액세스 권한을 정의하는 PMDB, 프로필 그룹과 연결된 사용자에게 영향을 주는 다양한 암호 규칙과 같은 사항을 정의할 수 있습니다.

CA Access Control 이 프로필 그룹을 사용하여 사용자 속성을 파악하는 방법

다음 프로세스는 CA Access Control 이 프로필 그룹을 사용하여 사용자 속성을 파악하는 방법에 대해 설명합니다.

1. CA Access Control 은 USER 또는 XUSER 클래스에 있는 사용자의 레코드에 속성 값이 있는지 확인합니다.

   사용자의 레코드에 속성 값이 있으면 CA Access Control 은 이 값을 사용합니다.

2. CA Access Control 은 사용자가 프로필 그룹에 할당되었는지 여부를 확인합니다.

   사용자가 프로필 그룹에 할당된 경우 프로세스가 계속 진행됩니다. 사용자가 프로필 그룹에 할당되지 않은 경우 CA Access Control 은 기본 속성 값을 이 사용자에게 할당합니다.

3. CA Access Control 은 프로필 그룹에 이 속성 값이 있는지 확인합니다.

   프로필 그룹에 이 속성 값이 있는 경우 CA Access Control 은 이 값을 해당 사용자에게 할당합니다. 프로필 그룹에 속성 값이 없는 경우 CA Access Control 은 기본 속성 값을 이 사용자에게 할당합니다.

   참고: 사용자 또는 프로필 그룹의 감사 속성이 설정되어 있지 않으면 그룹의 감사 속성이 사용자의 감사 속성에 영향을 줄 수 있습니다.

추가 정보:

CA Access Control 이 사용자의 감사 모드를 결정하는 방법 (페이지 120)

접근자 관리

CA Access Control 끝점 관리 또는 selang 을 사용하여 데이터베이스나 엔터프라이즈 사용자 또는 그룹을 생성, 수정 및 삭제할 수 있습니다.

사용자 또는 그룹 관리

특정 접근자의 속성을 표시 또는 수정하거나 접근자를 삭제하려면 먼저 해당 접근자를 찾아야 합니다.

사용자 또는 그룹을 관리하려면

1. CA Access Control 끝점 관리에서 다음을 수행하십시오.
   a. 사용자를 클릭합니다.
   b. "사용자" *또는* "그룹" 하위 탭을 클릭합니다.

   선택에 따라 "사용자" 또는 "그룹" 페이지가 나타납니다.
2. "검색" 섹션에서 다음 필드를 완료합니다.

   사용자/그룹 이름

   찾을 접근자의 마스크를 정의합니다. 나중에 접근자의 전체 이름을 입력하거나 마스크를 사용할 수 있습니다. 예를 들어 이름에 "admin"이 포함된 접근자를 나열하려면 *admin*을 사용합니다.

   모든 접근자를 나열하려면 *(별표)를 사용하고 하나의 문자만 대체하려면 ?(물음표)를 사용하십시오.

   사용자/그룹 리포지토리

   접근자 목록을 가져오려는 소스를 지정합니다. 소스는 다음 중 하나가 될 수 있습니다.

   - **내부 계정** - CA Access Control 데이터베이스에 정의된 접근자입니다.
   - **엔터프라이즈 계정** - 특정 엔터프라이즈 사용자 저장소에 정의된 접근자입니다.

AC 계정/프로필만 표시합니다.

다음과 같이 CA Access Control 데이터베이스에 레코드가 있는 계정만 나열할지 여부를 지정합니다.

- "내부 계정"을 선택한 경우에는 CA Access Control 데이터베이스에 있는 계정(네이티브가 아닌 계정)만 나열합니다.
- 엔터프라이즈 계정을 선택한 경우에는 CA Access Control 엔터프라이즈 프로필이 있는 계정(XUSER 또는 XGROUP 레코드)만 나열합니다.

"실행"을 클릭합니다.

선택한 리포지토리에 있는 접근자 목록이 나타납니다.

3. 다음 작업 중 *하나*를 수행합니다.
   - "보기" 열에서  을 클릭하여 접근자의 속성을 표시합니다.
   - 접근자를 삭제하려면 "삭제" 열에서  을 클릭합니다.
   - 접근자의 속성을 수정하려면 접근자의 이름을 클릭합니다.
   - 삭제할 접근자를 선택하고 "삭제"를 클릭합니다.
   - CA Access Control 데이터베이스에서 사용자 또는 그룹 레코드를 만들려면 "사용자 만들기" 또는 "그룹 만들기"를 클릭합니다.

예: 리포지토리에서 엔터프라이즈 사용자 검색

다음 그림은 ABC-DM1 엔터프라이즈 사용자 저장소에 있는 모든 사용자의 검색 결과를 보여 줍니다.

selang 을 사용한 사용자 관리

엔터프라이즈 사용자 레코드에 다음 selang 명령을 사용합니다.

- **newxusr** 및 **editxusr** - 새 엔터프라이즈 사용자 레코드 정의
- **chxusr** 및 **editxusr** - 엔터프라이즈 사용자의 CA Access Control 속성 변경
- **find xuser** - CA Access Control 레코드가 있는 엔터프라이즈 사용자 나열
- **rmxusr** - 사용자 삭제
- **show xuser** - 엔터프라이즈 사용자의 CA Access Control 속성 표시

CA Access Control 데이터베이스 사용자 레코드에 다음 selang 명령을 사용합니다.

- **newusr** 및 **editusr** - 새 사용자 레코드 정의
- **chusr** 및 **editusr** - 사용자 속성 변경
- **rmusr** - 사용자 삭제
- **find user** - 데이터베이스 사용자 나열
- **show user** - 사용자 속성 표시

예: selang 을 사용하여 데이터베이스에서 사용자 정의

다음 selang 명령은 CA Access Control 데이터베이스에서 보안 수준 100 을 사용하여 새 사용자를 정의합니다.

```
newusr internalUser level(100)
```

예: selang 을 사용하여 엔터프라이즈 사용자 속성 변경

다음 selang 명령은 엔터프라이즈 사용자 Terry 에게 AUDITOR 속성을 제공합니다.

```
chxusr Terry auditor
```

selang 을 사용한 그룹 관리

엔터프라이즈 그룹의 이름과 구성원을 변경할 수 없다는 점을 제외하고 모든 그룹의 모든 속성을 변경할 수 있습니다(CA Access Control 내에서).

그룹 속성을 변경하거나 그룹과 관련된 액세스 권한을 할당하려면 CA Access Control 끝점 관리를 사용하거나 다음 selang 명령을 사용합니다.

- **join[-]** 및 **joinx[-]**

  내부 그룹 구성원 변경

  그룹에 내부 접근자를 추가하려면 join 을 사용합니다. 내부 그룹에 엔터프라이즈 그룹 및 사용자를 추가하려면 joinx 를 사용합니다. 접근자를 제거하려면 명령의 -(빼기) 양식을 사용합니다.

- **editgrp, newgrp, chgrp**

  내부 그룹의 비구성원 속성 변경

- **editxgrp, newxgrp, chxgrp**

  엔터프라이즈 그룹의 비구성원 속성 변경

- **rmgrp, rmxgrp**

  사용자 그룹 삭제

예: selang 을 사용하여 데이터베이스에서 그룹 정의

다음 selang 명령은 데이터베이스에서 새 그룹 "sales"를 정의합니다. 그룹의 전체 이름은 "Sales Department"입니다.

```
newgrp sales name('Sales Department')
```

예: selang 을 사용하여 데이터베이스에 정의된 그룹 속성 변경

다음 selang 명령은 CA Access Control 이 그룹 AC_admins 의 구성원에 대한 모든 이벤트를 감사하게 합니다.

```
chgrp AC_admins audit(all)
```

예: selang 을 사용하여 ACL 에 엔터프라이즈 그룹 추가

다음 selang 명령은 myfile 의 ACL 에 엔터프라이즈 그룹 mygroup 을 추가합니다.

```
Authorize FILE (myfile) xgid(mygroup)
```

예: selang 을 사용하여 데이터베이스에 정의된 그룹에 엔터프라이즈 사용자 추가

다음 selang 명령은 데이터베이스에 정의된 그룹 AC_admins 에 엔터프라이즈 사용자 mydomain\administrator 를 추가합니다.

```
joinx mydomain\administrator group(AC_admins)
```

예: selang 을 사용하여 데이터베이스에 정의된 그룹에 엔터프라이즈 그룹 추가

다음 selang 명령은 _restricted 그룹에 엔터프라이즈 그룹 Guests 를 추가합니다.

```
joinx Guests group(_restricted)
```

제 4 장: 리소스 관리

이 섹션은 다음 항목을 포함하고 있습니다.

리소스

*리소스*는 접근자가 액세스하고 액세스 규칙으로 보호할 수 있는 엔터티이거나 이 엔터티에 해당하는 CA Access Control 데이터베이스 레코드입니다. 리소스의 예로는 파일, 프로그램, 호스트 및 터미널이 있습니다.

CA Access Control 에서 리소스 레코드를 생성하는 주요 목적은 리소스 레코드와 일치하는 리소스의 액세스 사용 권한을 정의하기 위해서입니다. 리소스 액세스에 필요한 액세스 사용 권한은 리소스 레코드의 액세스 제어 목록에 지정됩니다.

리소스 그룹

*리소스 그룹*은 다른 리소스 목록을 포함하는 리소스입니다. 리소스 그룹은 CONTAINER, GFILE, GSUDO, GTERMINAL 또는 GHOST 클래스 중 하나의 구성원입니다.

리소스 그룹 자체가 리소스이기 때문에 동일한 속성을 구성원 리소스로 포함합니다. 따라서 리소스 그룹을 사용하면 관리를 간소화할 수 있습니다. 리소스 그룹의 속성을 변경하면 모든 구성원 리소스의 속성을 변경할 수 있습니다.

참고: Windows 에서 CA Access Control 은 리소스에 대한 사용자 권한 부여를 확인할 때 리소스 그룹 소유권을 고려합니다. 이 동작은 r12.0 에서 처음 도입되었습니다. 이전 릴리스에서는 권한 부여 프로세스에서 리소스의 소유자만 고려했습니다.

예를 들어, 기본 액세스와 소유자 없이 FILE 리소스를 정의합니다. FILE 리소스는 명명된 소유자가 있는 GFILE 리소스의 구성원입니다. CA Access Control r12.0 이상에서는 명명된 그룹 소유자가 파일에 대한 모든 액세스 권한을 갖습니다. 이전 릴리스에서는 아무도 파일에 대한 액세스 권한을 갖지 않습니다.

클래스

CA Access Control 에서 레코드 *클래스*는 레코드가 가질 수 있는 속성을 정의합니다. 클래스의 모든 레코드는 속성 값은 다르지만 동일한 속성을 가집니다.

다음은 클래스의 예입니다.

- TERMINAL 클래스. tty1, tty 와 같은 터미널 레코드를 포함합니다.
- FILE 클래스. 파일 레코드를 포함합니다.
- PROGRAM 클래스. 프로그램 레코드를 포함합니다.

각 레코드에는 레코드 클래스에 해당하는 속성 값이 포함됩니다. 예를 들어 XUSER 클래스의 레코드에는 엔터프라이즈 사용자의 위치 및 근무 시간과 같은 속성이 포함되고, HOSTNET 클래스의 레코드에는 네트워크 서비스 및 IP 주소 데이터와 같은 속성이 포함됩니다.

CA Access Control 에는 미리 정의된 클래스가 있습니다. 사용자 정의 클래스라는 새 클래스를 정의할 수도 있습니다.

클래스의 기본 레코드

대부분의 클래스에는 자체 데이터베이스에 정의되지 않은 해당 클래스의 리소스에 대한 액세스 유형을 지정하는 기본 레코드(_default)가 포함될 수 있습니다.

다른 리소스 레코드와 마찬가지로 _default 레코드에는 ACL 및 defaces 필드가 포함될 수 있습니다. USER, GROUP, CATEGORY, SECLABEL 및 SEOS 를 제외한 모든 클래스에 대해 _default 레코드를 만들 수 있습니다.

UACC 클래스(폐기됨)

UACC 클래스는 더 이상 권장되지 않습니다. 클래스에서 레코드 기본값을 지정하려면 _default 레코드를 사용합니다.

몇몇 CA Access Control 의 초기 버전에서는 다른 클래스의 _default 레코드와 유사한 레코드에 대해 UACC 라는 별도의 클래스를 사용했습니다. UACC 클래스는 더 이상 권장되지 않으며 _default 레코드를 사용하는 경우 UACC 클래스에서 동일한 레코드를 확인하지 않습니다. 향후 버전에서는 UACC 클래스를 더 이상 지원하지 않을 수도 있습니다.

예를 들어 Henderson 이라는 사용자가 store_log 프로세스 중지를 시도한다고 가정하면 CA Access Control 은 다음 순서에 따라 권한 부여를 검사합니다. 첫 번째 질문은 다음과 같습니다. store_log 프로세스가 데이터베이스에 정의되어 있습니까? CA Access Control 은 데이터베이스를 검색하여 PROCESS 클래스에서 이름이 store_log 인 레코드를 찾습니다.

- 그런 레코드가 없으면 해당 프로세스는 CA Access Control 에 정의되어 있지 않습니다. 이 경우 CA Access Control 은 PROCESS 클래스의 _default 레코드나 UACC 클래스의 PROCESS 레코드를 사용하여 Henderson 이 store_log 를 종료할 수 있는지 여부를 확인합니다.
  - Henderson 이라는 사용자가 _default 레코드의 ACL 에 나타날 경우 ACL 에 지정된 권한이 적용됩니다.
  - Henderson 이 _default 레코드의 ACL 에 나타나지 *않을* 경우 _default 레코드의 defaccess 속성에 지정된 권한이 적용됩니다. 이 권한은 _default ACL 에 나타나지 않는 모든 사용자에게 적용됩니다.
- store_log 프로세스가 데이터베이스에 정의되어 있을 경우 문제는 Henderson 이라는 사용자가 데이터베이스의 store_log 프로세스 ACL 에 나타나는지 여부입니다.
  - 사용자 Henderson 이 store_log 프로세스에 대한 ACL 에 있는 경우 이 목록에 지정되어 있는 권한이 적용됩니다.
  - Henderson 이 ACL 에 나타나지 *않을* 경우 CA Access Control 은 store_log 리소스의 기본 액세스 속성에 지정된 권한을 적용합니다. 이러한 권한을 리소스의 기본 액세스 권한이라고 합니다.

참고: _default 의 기본 액세스 권한(defaccess)을 NONE 으로 설정하거나 _default 를 지정하지 않고 UACC 클래스의 해당 리소스 기본값이 NONE 일 경우, 클래스에 정의되지 않은 리소스에 액세스하려는 모든 접근자는 리소스에 대한 액세스가 거부됩니다.

_default(또는 UACC)의 기본 액세스 권한을 최상위 권한(ALL 또는 경우에 따라 READ 또는 EXECUTE)으로 설정할 경우 모든 사용자는 명시적으로 보호되지 않은 모든 리소스에 액세스할 수 있습니다.

미리 정의된 클래스

미리 정의된 클래스는 다음 유형으로 분류됩니다.

클래스 유형	목적
접근자	사용자 및 그룹과 같은 리소스에 액세스하는 개체를 정의합니다.
정의	보안 레이블 및 범주와 같은 보안 항목을 정의하는 개체를 정의합니다.
설치	CA Access Control 의 동작을 제어하는 객체를 정의합니다.
리소스	액세스 규칙에 의해 보호되는 개체를 정의합니다.

다음 표에서는 미리 정의된 클래스의 전체 목록을 보여 줍니다.

클래스	클래스 유형	설명
ADMIN	정의	ADMIN 특성이 없는 사용자에게 관리 업무를 위임하게 합니다. 이러한 사용자에게 전역 권한 부여 특성을 제공하고 관리 권한 범위를 제한합니다.
AGENT	리소스	CA Access Control 에 적용되지 않습니다.
AGENT_TYPE	리소스	CA Access Control 에 적용되지 않습니다.
APPL	리소스	CA Access Control 에 적용되지 않습니다.
AUTHHOST	접근자	CA Access Control 에 적용되지 않습니다.
CALENDAR	리소스	사용자, 그룹 및 리소스에 적용된 시간 제한에 대한 Unicenter TNG 달력 개체를 정의하게 합니다.
CATEGORY	정의	보안 범주를 정의하게 합니다.

클래스	클래스 유형	설명
CONNECT	리소스	나가는 연결을 보호하게 합니다. 이 클래스의 레코드는 어떤 사용자가 어떤 인터넷 호스트에 액세스할 수 있는지 정의합니다. CONNECT 클래스를 활성화하려면 먼저 스트림 모듈이 활성화되어 있는지 확인하십시오.
CONTAINER	리소스	다른 리소스 클래스의 개체 그룹을 정의하게 하므로 개체의 여러 다른 클래스에 규칙을 적용할 때 액세스 규칙을 간단하게 정의할 수 있습니다.
FILE	리소스	파일, 디렉터리 또는 파일 이름 마스크를 보호하게 합니다.
GAPPL	리소스	CA Access Control 에 적용되지 않습니다.
GAUTHHOST	정의	CA Access Control 에 적용되지 않습니다.
GFILE	리소스	GFILE 클래스의 각 레코드는 파일 또는 디렉터리 그룹을 정의합니다. 그룹화는 사용자를 그룹에 연결하는 것과 같은 방식으로 파일 또는 디렉터리(FILE 클래스의 리소스)를 GFILE 리소스에 명시적으로 연결하여 수행됩니다.
GHOST	리소스	GHOST 클래스의 각 레코드는 호스트 그룹을 정의합니다. 그룹화는 사용자를 그룹에 연결하는 것과 같은 방식으로 호스트(HOST 클래스의 리소스)를 GHOST 리소스에 명시적으로 연결하여 수행됩니다.
GROUP	접근자	이 클래스의 각 레코드는 내부 그룹을 정의합니다.
GSUDO	리소스	이 클래스에 있는 각 레코드는 명령 그룹을 정의합니다. 이 명령은 마치 다른 사용자가 명령을 실행하고 있는 것처럼 실행할 수 있습니다. sesudo 명령은 이 클래스를 사용합니다.
GTERMINAL	리소스	GTERMINAL 클래스의 각 레코드는 터미널 그룹을 정의합니다.
HNODE	정의	HNODE 클래스에는 조직의 CA Access Control 호스트에 대한 정보가 포함되어 있습니다. 클래스의 각 레코드는 엔터프라이즈의 노드를 나타냅니다.
HOLIDAY	정의	HOLIDAY 클래스의 각 레코드는 사용자가 로그인할 때 추가 권한이 필요한 하나 이상의 기간을 정의합니다.

클래스	클래스 유형	설명
HOST	리소스	HOST 클래스의 각 레코드는 호스트를 정의합니다. 호스트는 이름이나 IP 주소로 식별합니다. 개체에는 로컬 호스트가 이 호스트로부터 서비스를 받을 수 있는지 여부를 결정하는 액세스 규칙이 포함됩니다. HOST 클래스를 활성화하려면 먼저 스트림 모듈이 활성화되어 있는지 확인하십시오.
HOSTNET	리소스	HOSTNET 클래스의 각 레코드는 IP 주소 마크스로 식별하고 액세스 규칙이 포함되어 있습니다.
HOSTNP	리소스	이 클래스에 있는 각 레코드는 호스트 그룹을 정의합니다. 이 그룹에 속하는 호스트는 모두 동일한 이름 패턴을 가집니다. 각 HOSTNP 개체의 이름에는 와일드카드가 포함됩니다.
LOGINAPPL	정의	LOGINAPPL 클래스의 각 레코드는 로그인 응용 프로그램을 정의하고, 프로그램을 사용하여 로그인할 수 있는 사용자를 식별하며, 로그인 프로그램이 사용되는 방식을 제어합니다.
MFTERMINAL	정의	MFTERMINAL 클래스의 각 레코드는 메인프레임 CA Access Control 관리 컴퓨터를 정의합니다.
POLICY	리소스	POLICY 클래스의 각 레코드는 정책을 배포하고 제거하는 데 필요한 정보를 정의합니다. 여기에는 정책을 배포하고 제거하는 데 사용되는 selang 명령이 있는 RULESET 개체에 연결하는 링크가 포함되어 있습니다.
PROCESS	리소스	PROCESS 클래스의 각 레코드는 실행 파일을 정의합니다.
PROGRAM	리소스	PROGRAM 클래스의 각 레코드는 조건부 액세스 규칙과 함께 사용할 수 있는 트러스트된 프로그램을 정의합니다. 트러스트된 프로그램은 프로그램이 손상되지 않도록 Watchdog 에서 모니터링하는 setuid/setgid 프로그램입니다.
PWPOLICY	정의	PWPOLICY 클래스의 각 레코드는 암호 정책을 정의합니다.
RESOURCE_DESC	정의	CA Access Control 에 적용되지 않습니다.
RESPONSE_TAB	정의	CA Access Control 에 적용되지 않습니다.
RULESET	리소스	RULESET 클래스의 각 레코드는 정책을 정의하는 규칙 집합을 나타냅니다.
SECFILE	정의	SECFILE 클래스의 각 레코드는 변경해서는 안 되는 파일을 정의합니다.

클래스	클래스 유형	설명
SECLABEL	정의	SECLABEL 클래스의 각 레코드는 보안 레이블을 정의합니다.
SEOS	설치	SEOS 클래스의 한 레코드는 활성 클래스 및 암호 규칙을 지정합니다.
SPECIALPGM	설치	SPECIALPGM 클래스의 각 레코드는 Windows 의 백업, DCM, PBF 및 PBN 기능이나 UNIX 의 xdm, 백업, 메일, DCM, PBF 및 PBN 프로그램을 등록하거나 특수 CA Access Control 권한 보호가 필요한 응용 프로그램을 논리적 사용자 ID 에 연결합니다. 이렇게 하면 수행하는 사람이 아니라 작업의 내용에 따라 액세스 권한을 설정할 수 있습니다.
SUDO	리소스	sesudo 명령에서 사용하는 이 클래스는 root 같은 다른 사용자가 명령을 실행하는 것처럼 일반 사용자 같은 한 사용자가 실행할 수 있는 명령을 정의합니다.
SURROGATE	리소스	이 클래스의 각 레코드에는 접근자를 대리 사용자로 사용할 수 있는 사용자를 정의하는 해당 접근자에 대한 액세스 규칙이 포함되어 있습니다.
TCP	리소스	이 클래스의 각 레코드는 메일, http 또는 ftp 와 같은 TCP/IP 서비스를 정의합니다.
TERMINAL	리소스	TERMINAL 클래스의 각 레코드는 사용자가 로그인할 수 있는 장치인 터미널을 정의합니다.
UACC	리소스	각 리소스 클래스에 대한 기본 액세스 규칙을 정의합니다.
USER	접근자	이 클래스의 각 레코드는 내부 사용자를 정의합니다.
USER_ATTR	정의	CA Access Control 에 적용되지 않습니다.
USER_DIR	리소스	CA Access Control 에 적용되지 않습니다.
XGROUP	리소스	이 클래스의 각 레코드는 CA Access Control 에 대해 엔터프라이즈 그룹을 정의합니다.
XUSER	리소스	이 클래스의 각 레코드는 CA Access Control 에 대해 엔터프라이즈 사용자를 정의합니다.

참고: CA Access Control 데이터베이스 클래스 TCP 및 SURROGATE 는 기본적으로 활성화되지 않습니다.

TCP 클래스가 활성화되어 있지만 어떠한 TCP 레코드도 없고 _default TCP 리소스가 변경되지 않은 이전 릴리스에서 업그레이드하는 경우 CA Access Control 은 업그레이드 중에 이 클래스를 비활성화합니다. 이 사항은 SURROGATE 클래스에 대해서도 동일합니다.

SURROGATE 클래스가 활성화되어 있는 이전 릴리스에서 업그레이드하고, SURROGATE 레코드를 정의했거나 임의의 SURROGATE 레코드의 값을 기본값에서 변경한 경우 CA Access Control 은 업그레이드 후 SURROGATE 클래스 구성을 유지합니다. 클래스와 커널 모드 차단이 활성화된 상태로 유지됩니다.

참고: CA Access Control 클래스에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

사용자 정의 클래스

CA Access Control 을 통해 새 클래스를 정의하면 추상 개체에 적합한 레코드를 작성하여 추상 개체를 보호할 수 있습니다.

예: 데이터베이스 보기에 대한 사용자 정의 클래스

사이트는 데이터베이스를 사용하여 독점 데이터를 저장 및 표시할 수 있습니다.

사용자 정의 클래스 DATABASE_VIEWS 를 정의하고 각 데이터베이스 보기가 해당 클래스의 리소스 구성원이 되도록 정의할 수 있습니다. 해당 데이터베이스 보기를 작성하는 데 필요한 액세스 권한을 정의하는 ACL 을 리소스에 제공합니다. 사용자가 데이터베이스 보기를 만들려고 하면 CA Access Control 이 사용자의 액세스 권한을 확인하고 ACL 에 따라 작성을 허용하거나 거부합니다.

사용자 정의 클래스 리소스의 와일드카드

사용자 정의 클래스의 리소스 이름에 와일드카드를 사용하면 여러 실제 리소스와 일치하는 리소스 레코드를 작성할 수 있습니다. 와일드카드 패턴과 일치하는 이름을 가진 실제 리소스는 리소스 레코드와 관련된 액세스 권한으로 보호됩니다.

사용할 수 있는 와일드카드는 다음과 같습니다.

- * - 수 제한 없는 모든 문자
- ? - 문자 한 개

실제 리소스 이름이 리소스 레코드 이름 여러 개와 일치할 경우에는 가장 긴 와일드카드가 아닌 일치 항목이 해당 리소스에 사용됩니다.

CA Access Control 은 다음 와일드카드 패턴을 리소스 이름으로 허용하지 *않습니다*.

- *
- /*
- /tmp/*
- /etc/*

사용자 정의 클래스 - 예

시스템에서 은행 서비스를 제공하고 계좌 간 거액 이체를 보호하려고 합니다. 다음 개요를 사용하여 이 보안을 설정할 수 있습니다.

1. 예를 들어 TRANSFERS 라는 이체를 설명하는 레코드가 포함된 클래스를 정의합니다.
2. 보호할 각 이체 금액 수준마다 TRANSFERS 클래스에 레코드를 정의합니다.

   예를 들어 Upto.$1K, Upto.$1M, Upto.$10M 및 Over.$10M 이라는 레코드를 정의할 수 있습니다.

   이체를 제어하는 데 필요한 다른 리소스를 TRANSFERS 클래스의 구성원으로 정의합니다.
3. 서로 다른 사용자에게 서로 다른 최대 이체 권한을 부여하려면 TRANSFERS 클래스의 다양한 레코드에 대한 액세스 권한을 부여하거나 거부합니다.
4. 또한 프로그래밍 방식의 전송을 처리하려면 은행의 송금 프로그램에 CA Access Control API 에 대한 호출을 삽입하여 송금을 허용하기 전에 사용자의 권한을 확인하도록 합니다.

Windows 서비스 보호

CA Access Control 에서는 Windows 서비스를 보호할 수 있습니다. *Windows 서비스*는 Windows 의 배경에서 실행되는 프로그램으로서 UNIX 의 데몬과 같습니다.

CA Access Control Windows 서비스 보호는 다음 중 하나에서 시작되는 서비스 액세스 이벤트를 차단합니다.

- 서비스 관리 및 정보 이벤트.

  CA Access Control 은 각 서비스 액세스에 대해 services.exe 프로세스를 차단합니다. 여기에는 서비스 시작 또는 중지가 포함됩니다. 예를 들어 net start *service*, net stop *service* 등이 보호됩니다.

  이 경우 차단된 이벤트는 보호된 서비스 이름을 사용하여 감사됩니다.

- 서비스 데이터베이스 관리 이벤트.

  CA Access Control 은 서비스 상태 쿼리 또는 변경으로부터 보호하기 위해 서비스 제어 관리 데이터베이스에 대한 레지스트리 호출을 차단합니다. 즉, CA Access Control 은 보호된 서비스와 관련된 레지스트리 영역을 자동으로 보호합니다. 결과적으로 서비스 보호를 정의할 경우 CA Access Control 은 다음 레지스트리 키를 보호합니다.

  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*service_name*
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*service_name**

  이 경우 차단된 이벤트는 전체 레지스트리 경로를 사용하여 감사됩니다.

다른 리소스를 보호하는 것과 동일한 방법으로, 즉 서비스에 리소스를 할당하고 리소스의 액세스 제어 목록에 접근자를 추가하여 Windows 서비스를 보호할 수 있습니다. Windows 서비스의 리소스 클래스는 WINSERVICE 입니다. WINSERVICE 리소스는 두 개의 액세스 제어 목록, 즉 ACL 과 NACL 을 가지고 있습니다. WINSERVICE 액세스 제어 목록의 항목에 대한 유효한 액세스 유형은 다음과 같습니다.

- 읽기
- 수정
- Start
- 중지
- 일시 중지
- 다시 시작

Windows 서비스 보호 활성화 및 비활성화

CA Access Control 의 Windows 서비스 보호를 활성화 또는 비활성화할 수 있습니다.

Windows 서비스 보호를 활성화하려면 CA Access Control 레지스트리의 Instrumentation\PlugIns\WinServiceplg 섹션에 있는 구성 설정 OperationMode 를 1 로 설정하고, 비활성화하려면 OperationMode 를 0 으로 설정하십시오.

기본적으로 CA Access Control 은 Windows 서비스 보호를 활성화합니다.

CA Access Control 에서 Windows 서비스를 활성화하려면, 보호가 활성화되고 WINSERVICE 클래스가 활성 상태여야 합니다.

Windows 서비스 보호

Windows 서비스를 보호하고 이에 따라 Windows 작업에 추가 보호를 제공할 수 있습니다.

Windows 서비스를 보호하려면

1. Windows 서비스 보호를 활성화 (페이지 66)했는지 확인합니다.
2. WINSERVICE 클래스가 활성 상태인지 확인합니다. 이 클래스는 기본적으로 활성 상태입니다.
3. 보호하려는 Windows 서비스와 같은 이름으로 CA Access Control 에 WINSERVICE 레코드를 만듭니다.

   참고: Windows 서비스 이름은 "Windows 서비스 속성" 대화 상자의 "일반" 탭에 표시되지만 이 탭의 "표시 이름"과 같지는 않습니다.
4. 접근자와 접근자의 액세스 권한을 서비스에 할당합니다.

   이제 서비스가 보호됩니다.

예: 인쇄 스풀러에 대한 액세스 제한

Windows 에서 인쇄 스풀러의 서비스 이름은 spooler 입니다. 다음 selang 명령은 WINSERVICE 클래스가 활성화되도록 하고 스풀러에 대한 기본 액세스를 *읽기*로 설정합니다.

```
setoptions class+(WINSERVICE)
editres WINSERVICE(spooler) defacc(R)
```

비 IPv4 텔넷 연결이 Windows Server 2008 에서 보안되지 않음

Windows Server 2008 에서 IPv4 를 사용하지 않는 경우 CA Access Control 은 텔넷 통신의 보안을 유지할 수 없습니다.

Windows Server 2008 에서 로컬 호스트 텔넷 연결(로컬 호스트에서 로컬 호스트로의 텔넷)을 보호하려면 다음과 같이 /etc/HOSTS 파일을 수정해야 합니다.

```
127.0.0.1       localhost
#       ::1                 localhost
127.0.0.1       <도메인 접미사를 생략한 서버 이름>
```

IPv6 도메인에 컴퓨터가 있는 경우 다음 줄을 추가해야 합니다.

```
127.0.0.1       <도메인 접미사를 포함한 서버 이름>
```

보호된 Windows 서비스에 대한 액세스 시도 보기

CA Access Control 은 Windows 서비스를 보호할 때 서비스와 관련된 액세스 시도를 차단하고 감사 로그에 기록합니다. 이러한 액세스 시도는 시작, 중지 등 서비스를 관리하기 위해 services.exe 프로세스를 사용한 결과이거나, 보호된 서비스의 서비스 데이터베이스 관리 영역에 대한 레지스트리 액세스의 결과일 수 있습니다. 전자의 경우 서비스 이름만 포함되지만 후자(레지스트리 액세스)의 경우 전체 레지스트리 경로가 포함됩니다. Windows 서비스와 관련된 모든 액세스 시도를 보려면 와일드카드를 사용해야 합니다.

보호된 Windows 서비스에 대한 액세스 시도를 보려면 WINSERVICE 클래스 및 리소스 이름 **myService*의 감사 레코드를 필터링하는 감사 필터를 작성하십시오.

레지스트리와 서비스 관리 인터페이스 중 어디를 통해 액세스가 시도되었든, CA Access Control 은 사용자가 정의한 WINSERVICE 리소스에 대한 모든 감사 레코드를 표시합니다.

예: 인쇄 스풀러 서비스에 대한 모든 액세스 시도 보기

이 예는 사용자가 다음과 같이 액세스 없이 CA Access Control 에 대해 인쇄 스풀러 서비스를 정의했다고 가정합니다.

```
er winservice spooler defaccess(none) owner(nobody)
```

그러면 다음과 같이 seaudit 유틸리티를 사용하여 인쇄 스풀러 서비스에 대한 모든 액세스 시도를 나열할 수 있습니다.

```
seaudit -resource WINSERVICE *spooler* *
```

이 명령은 인쇄 스풀러 서비스에 대한 액세스 시도가 기록된, WINSERVICE 클래스에 대한 모든 감사 레코드를 나열합니다. 출력 결과는 다음과 유사할 수 있습니다.

```
seaudit - Audit log lister
3 Apr 2008 16:48:53 D WINSERVICE   bigHost1\Administrator Read      69  2 Spooler
   c:\WINDOWS\system32\services.exe bigHost1.comp.com
3 Apr 2008 16:48:53 D WINSERVICE   bigHost1\Administrator Read      69  2 Spooler
   c:\WINDOWS\system32\services.exe bigHost1.comp.com
3 Apr 2008 16:50:53 D WINSERVICE   bigHost1\Administrator Read      69  2 Spooler
   c:\WINDOWS\system32\services.exe bigHost1.comp.com
3 Apr 2008 16:50:53 D WINSERVICE   bigHost1\Administrator Read      69  2 Spooler
   c:\WINDOWS\system32\services.exe bigHost1.comp.com
3 Apr 2008 16:53:53 D WINSERVICE   bigHost1\Administrator Read      69  2 Spooler
```

```
   c:\WINDOWS\system32\services.exe bigHost1.comp.com
3 Apr 2008 16:53:53 D WINSERVICE   bigHost1\Administrator Read       69  2 Spooler
   c:\WINDOWS\system32\services.exe bigHost1.comp.com
03 Apr 2008 16:54:10 D WINSERVICE   bigHost1\Administrator Read       69  2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler
   C:\WINDOWS\regedit.exe bigHost1.comp.com
03 Apr 2008 16:54:10 D WINSERVICE   bigHost1\Administrator Read       69  2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler
   C:\WINDOWS\regedit.exe bigHost1.comp.com
03 Apr 2008 16:54:19 D WINSERVICE   bigHost1\Administrator Read       69  2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler
   C:\WINDOWS\regedit.exe bigHost1.comp.com
03 Apr 2008 16:54:26 D WINSERVICE   bigHost1\Administrator Read       69  2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler
   C:\WINDOWS\regedit.exe bigHost1.comp.com
03 Apr 2008 16:54:26 D WINSERVICE   bigHost1\Administrator Modify     69  2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler
   C:\WINDOWS\regedit.exe bigHost1.comp.com

Total records displayed 11
```

Windows 레지스트리 보호

CA Access Control 에서는 Windows 레지스트리의 항목을 보호할 수 있습니다.

REGKEY 클래스의 리소스를 키에 할당하여 레지스트리 키를 보호합니다. 그런 다음 다른 리소스와 마찬가지로 키에 대한 액세스 권한을 지정할 수 있습니다.

키에 액세스 권한을 지정하더라도 키의 하위 키에 대한 액세스에는 영향을 주지 않습니다. 단, 키에 대해 읽기 액세스를 필요로 하는, 하위 키의 열거(나열)는 예외입니다.

CA Access Control 은 Windows Server 2003 및 그 이후 Windows 시스템의 AC 환경에서 REGVAL 리소스만 지원합니다. 이러한 시스템에서 CA Access Control 은 REGVAL 클래스와 함께 레지스트리 값을 보호하며, REGKEY 액세스 권한은 키의 값에 대한 액세스에 영향을 주지 않습니다.

이전 시스템에서 CA Access Control 은 AC 환경에서 REGVAL 리소스를 지원하지 않으며, REGKEY 레코드에 적용된 액세스 권한은 키의 값에 대한 액세스에 영향을 줍니다.

REGKEY 및 REGVAL 레코드의 구조는 동일합니다. 각 레코드에는 다음과 같은 액세스 제어 목록이 포함되어 있습니다.

- ACL
- CALACL
- NACL
- PACL

REGVAL 및 REGKEY 레코드는 모두 다음과 같은 동일한 액세스 유형을 허용합니다.

- READ
- WRITE
- DELETE
- NONE

참고: CA Access Control 레지스트리 보호는 하이브를 로드 및 언로드하는 레지스트리 작업을 보호하지 않습니다. Windows Server 2008 이상 시스템에서 접근자가 NONE 액세스 권한을 사용하여 보호된 레지스트리 값에 액세스를 시도하면 CA Access Control 에서 REG_NONE 값이 반환됩니다. REG_NONE 값은 값이 있음을 확인하지만 값이 무엇인지 지정하지 않습니다.

Windows 레지스트리 항목 보호

Windows 레지스트리 항목을 보호하고 이에 따라 Windows 작업에 추가 보호를 제공할 수 있습니다.

Windows 레지스트리 항목을 보호하려면

1. REGKEY 및 REGVAL 클래스 레코드를 사용하려면 이러한 클래스가 활성 상태인지 확인합니다. 이 둘은 기본적으로 활성 상태입니다.
2. 보호하려는 레지스트리 키 또는 값의 이름으로 REGKEY 또는 REGVAL 레코드를 만듭니다.

   참고: 키 또는 값을 지정하려면 전체 레지스트리 경로 이름을 사용하십시오. 와일드카드를 사용하여 키에 중첩된 모든 하위 키 또는 하위 키 값을 지정할 수 있습니다.

   이제 레지스트리 항목은 CA Access Control 이 레코드에 제공하는 기본 액세스와 함께 보호됩니다.
3. (선택 사항) 사용자와 그룹을 해당 액세스 권한과 함께 REGKEY 또는 REGVAL 레코드에 있는 해당 액세스 제어 목록에 할당합니다.

예: NONE 의 기본 액세스를 레지스트리 키에 제공

다음 selang 명령은 NONE 의 기본 액세스를 레지스트리 키에 제공합니다.

```
er REGKEY HKEY_LOCAL_MACHINE\SOFTWARE\Test\Key1 defacc(NONE) owner(nobody)
```

그 결과 key1 에 대한 기본 액세스는 다음과 같습니다.

동작	Windows Server 2003 이전 시스템	Windows Server 2003 시스템 이상	Windows Server 2008 시스템 이상
하위 키 열거	거부	거부	거부
키 쿼리, 수정, 이름 바꾸기 또는 삭제	거부	거부	거부
키에 대한 하이브 로드 또는 언로드	거부	거부	거부
값 열거	거부	거부	허용

동작	Windows Server 2003 이전 시스템	Windows Server 2003 시스템 이상	Windows Server 2008 시스템 이상
값 읽기, 만들기, 이름 바꾸기 또는 삭제	거부	허용	허용
하위 키의 하위 키 열거	거부	허용	허용
하위 키 만들기	허용	허용	허용
하위 키 쿼리, 수정, 이름 바꾸기 또는 삭제	허용	허용	허용
하위 키에 대한 하이브 로드 또는 언로드	허용	허용	허용

예: READ 의 기본 액세스를 레지스트리 키에 제공

다음 selang 명령은 READ 의 기본 액세스를 레지스트리 키에 제공합니다.

```
er REGKEY HKEY_LOCAL_MACHINE\SOFTWARE\Test\Key1 defacc(READ) owner(nobody)
```

그 결과 Key1 에 대한 기본 액세스는 다음과 같습니다.

동작	Windows Server 2003 이전 시스템	Windows Server 2003 이상	Windows Server 2008 이상
하위 키 열거	허용	허용	허용
키 읽기	허용	허용	허용
키 수정, 이름 바꾸기 또는 삭제	거부	거부	거부
키에 대한 하이브 로드 또는 언로드	거부	거부	거부
값 열거	허용	허용	허용

동작	Windows Server 2003 이전 시스템	Windows Server 2003 이상	Windows Server 2008 이상
값 읽기	허용	허용	허용
값 만들기, 이름 바꾸기 또는 삭제	거부	허용	허용
하위 키의 하위 키 열거	허용	허용	허용
하위 키 만들기	허용	허용	허용
하위 키 쿼리, 수정, 이름 바꾸기 또는 삭제	허용	허용	허용
하위 키에 대한 하이브 로드 또는 언로드	허용	허용	허용
하위 키 값 열거	허용	허용	허용
하위 키 값 만들기	허용	허용	허용

예: NONE 의 기본 액세스를 레지스트리 키 와일드카드에 제공

다음 selang 명령은 NONE 의 기본 액세스를 레지스트리 키의 모든 하위 키에 제공합니다.

```
er REGKEY HKEY_LOCAL_MACHINE\SOFTWARE\Test\Key1\* defacc(NONE) owner(nobody)
```

와일드카드(*)는 Key1 에 적용되지 않고 Key1 의 모든 하위 키에 적용되므로 Key1 의 모든 하위 키에 대한 모든 형태의 액세스가 거부됩니다. 부모 보호 규칙으로 인해 Key1 의 이름을 바꾸고 삭제하기 위한 액세스도 거부됩니다.

이 명령은 Key1 의 값에 대한 액세스를 허용합니다. Key1 하위 키의 값(예: Key1\subkey1\의 값)에 대한 액세스는 Windows 시스템 간에 다릅니다.

- Windows Server 2003 및 그 이후 시스템에서는 key1 하위 키의 값을 열거하기 위한 액세스는 거부되지만 값을 만들고 이름을 변경하고 삭제하고 읽기 위한 액세스는 허용됩니다.
- Windows Server 2003 이전 시스템에서는 Key1 하위 키의 값에 대한 모든 액세스가 거부됩니다.

예: NONE 의 기본 액세스를 레지스트리 값에 제공

다음의 selang 명령은 Windows Server 2003 및 그 이후 시스템에서 NONE 액세스와 함께 특정 레지스트리 값을 보호합니다.

```
er REGVAL HKEY_LOCAL_MACHINE\SOFTWARE\TestKey\value1 defacc(NONE) owner(nobody)
```

참고: Windows Server 2008 이상 시스템에서 접근자가 NONE 액세스 권한을 사용하여 보호된 레지스트리 값에 액세스를 시도하면 CA Access Control 에서 REG_NONE 값이 반환됩니다. REG_NONE 값은 값이 있음을 확인하지만 값이 무엇인지 지정하지 않습니다.

파일 스트림 보호

스트림은 일련의 바이트입니다. 파일 스트림은 파일 데이터를 포함하며 파일에 대한 추가 정보를 제공합니다. 예를 들어, 키워드나 메타데이터를 포함하는 스트림을 만들 수 있습니다.

참고: 파일 스트림은 NTFS 파일 시스템에서만 이용할 수 있습니다. 파일 스트림에 대한 자세한 내용은 MSDN(Microsoft Developer Network) 라이브러리 웹 사이트를 참조하십시오.

FILE 규칙을 작성하면 CA Access Control 은 파일에 대한 기본 데이터 스트림을 자동으로 보호합니다. 예를 들어 c:\foo.txt 파일을 보호하는 규칙은 c:\foo.txt::$DATA 에 대한 사용 권한도 관리합니다. 그러나 CA Access Control 은 기본 이외의 데이터 스트림을 자동으로 보호하지 않습니다. 이들에 대해서는 추가 파일 보호 규칙을 작성해야 합니다.

파일 스트림을 보호하려면 다음 중 *하나*를 수행하십시오.

- 특정 스트림을 보호하려면 다음 형식으로 파일 규칙을 작성하십시오.

  drive:*path**filename.ext*:*stream*

- 특별한 스트림의 특정 스트림 유형을 보호하려면 다음 형식으로 파일 규칙을 작성하십시오.

  drive:*path**filename.ext*:*stream*:*type*

- 모든 스트림을 보호하려면 다음 형식으로 일반 파일 규칙을 작성하십시오.

  drive:*path**filename.ext*:*

예: 모든 파일 스트림 보호

다음 selang 명령은 c:\foo.txt 파일에 있는 모든 스트림을 보호하는 일반 파일 규칙을 생성합니다.

```
er file c:\foo.txt:* owner(nobody) defaccess(none)
```

예: 특정 스트림 보호

다음 selang 명령은 c:\foo.txt 파일에 있는 *mystream* 스트림을 보호하는 파일 규칙을 생성합니다.

```
er file c:\foo.txt:mystream owner(nobody) defaccess(none)
```

내부 파일 보호

설치 중 CA Access Control 은 다음과 같은 두 가지 유형의 내부 파일을 보호하기 위한 규칙을 작성합니다:

- 내부 규칙 - 구성 파일, 로그 파일, 데이터베이스 파일을 보호합니다.

  내부 규칙은 삭제할 수 없습니다.

- 기본 규칙 - 통신을 암호화하고 인증하는 데 사용하는 루트 및 서버 인증서와 같은 민감한 파일을 보호합니다.

  설치 후에 기본 규칙을 삭제할 수 있습니다.

내부 파일 규칙

내부 파일 규칙은 구성 파일, 로그 파일, 데이터베이스 파일을 보호합니다. 내부 파일 규칙은 selang 에서 보이지 않으며 감지되지 않습니다. 하지만 FILE 규칙을 작성하여 내부 파일 규칙을 무시할 수 있습니다. 이러한 FILE 규칙을 삭제하면 CA Access Control 이 내부 파일 규칙으로 되돌립니다.

데이터베이스 파일을 제외하고, CA Access Control 이 내부 파일 규칙을 사용하여 보호하는 파일은 다음 액세스 권한을 갖습니다.

- CA Access Control 내부 프로세스에 대한 완전한 액세스
- 기타 모든 접근자에 대한 읽기 및 실행(해당하는 경우) 액세스

CA Access Control 이 내부 파일 규칙을 사용하여 보호하는 파일은 다음 액세스 권한을 갖습니다.

- CA Access Control 내부 프로세스는 데이터베이스에 대한 모든 액세스 권한을 갖습니다.
- NT AUTHORITY\System 사용자는 데이터베이스에 대해 읽기 액세스 권한을 갖습니다.
- 모든 다른 접근자는 데이터베이스에 대한 액세스 권한이 없습니다.

  참고: r12.5 SP3 에서 모든 다른 접근자에 대한 기본 액세스 권한이 변경되었습니다. 이전 릴리스에서 모든 다른 접근자는 기본적으로 데이터베이스 파일에 대해 읽기 액세스 권한이 있었습니다.

CA Access Control 은 내부 파일 규칙을 사용하여 다음 파일을 보호합니다. 표의 두 번째 열에는 레지스트리 하위 키와 파일 위치를 지정하는 항목(해당하는 경우)이 나열되어 있습니다. CA Access Control 은 다음 레지스트리 키에 레지스트리 항목을 작성합니다.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl
```

참고: 일부 파일 위치는 내부적으로 정의되며 레지스트리 항목이 없습니다. 이러한 파일의 위치는 구성할 수 없습니다.

파일	레지스트리 하위 키 및 항목	기본 파일 위치
seosdrv.sys	-	%SystemRoot%\system32\drivers\seosdrv.sys
cainstrm.sys	-	%SystemRoot%\system32\drivers\cainstrm.sys
drveng.sys	-	%SystemRoot%\system32\drivers\drveng.sys
pwdchange.dll	-	%SystemRoot%\system32\pwdchange.dll
SUSRAUTH.dll	-	%SystemRoot%\system32\SUSRAUTH.dll
eACSubAuth.dll	-	%SystemRoot%\system32\eACSubAuth.dll
eACPasswordFltr.dll	-	%SystemRoot%\system32\eACPasswordFltr.dll
모든 데이터베이스 파일	SeOSD\dbdir	*ACInstallDir*\Data\seosdb
모든 도움말 파일	lang\help_path	*ACInstallDir*\Data\help
모든 바이너리	-	*ACInstallDir*\bin
seosd.trace	SeOSD\trace_file	*ACInstallDir*\log
seos.audit	logmgr\audit_log	*ACInstallDir*\log
seos.audit.bak	logmgr\audit_back	*ACInstallDir*\log
seos.error	logmgr\error_log	*ACInstallDir*\log
seos.error.bak	logmgr\error_back	*ACInstallDir*\log
seos.msg	message\filename	*ACInstallDir*\Data
stop.ini	STOP\STOPIniFileName	*ACInstallDir*\Data
stopsignature.dat	STOP\STOPSignatureFileName	*ACInstallDir*\Data
response.ini	SeOSD\ResponseFile	*ACInstallDir*\Data

파일	레지스트리 하위 키 및 항목	기본 파일 위치
audit.cfg	logmgr\AuditFiltersFile	*ACInstallDir*\Data

참고: 구성 설정에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

기본 파일 규칙

CA Access Control 은 민감한 파일을 보호하기 위해 설치 중 기본 파일 규칙을 만듭니다. 기본 파일 규칙은 selang 에서 보이며 삭제될 수 있습니다.

다음 표는 CA Access Control 이 기본 파일 규칙을 사용하여 보호하는 민감한 파일과 이 파일에 대한 액세스 권한 및 허용된 접근자를 나열합니다.

표에서 *PMDBDir* 는 정책 모델 데이터베이스(PMDB)가 있는 디렉터리이고, *pmd_name* 은 각 정책 모델의 이름입니다. 기본적으로 *PMDBDir* 는 *ACInstallDir*\Data 에 있습니다. *PMDBDir* 의 위치는 다음 레지스트리 항목에 정의되어 있습니다.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Pmd\_Pmd_directory_
```

파일	기본 액세스	허용된 접근자
ACInstallDir\data\crypto\crypto.dat	없음	sechkey
ACInstallDir\data\crypto\def_root.pem*	없음	sechkey
ACInstallDir\data\crypto\sub.key	없음	sechkey
ACInstallDir\data\crypto\sub.pem	없음	sechkey
ACInstallDir\log\policyfetcher.log	읽기	+policyfetcher
*PMDBDir**pmd_name*	Read, Chdir	-
*PMDBDir**pmd_name**	Read, Execute	-

제 5 장: 권한 부여 관리

이 섹션은 다음 항목을 포함하고 있습니다.

액세스 권한

CA Access Control 의 주요 목적은 액세스 권한을 할당 및 적용하는 것입니다.

액세스 권한은 항상 다음 구성 요소로 구성됩니다.

- 액세스 권한이 적용되는 리소스(예: 파일, 호스트 또는 터미널)
- 액세스 유형(예: 읽기, 쓰기, 삭제, 로그인, 실행)
- 접근자(사용자 또는 그룹)

사용자는 다음 중 하나 이상에 해당되기 때문에 특정 방법으로 리소스에 액세스하는 권한을 가집니다.

- 사용자가 리소스 ACL 에서 부여하는 액세스 권한을 가집니다.
- 사용자가 액세스 권한을 가진 그룹의 구성원입니다.
- 사용자가 액세스 권한을 가진 프로그램을 실행하고 있습니다. 예를 들어 사용자가 SPECIALPGM 클래스의 프로그램을 실행하거나 SUDO 클래스의 명령을 실행하는 권한을 가집니다.

참고: 클래스별 액세스 권한에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

액세스 권한 설정 - 예

예: 내부 사용자에게 읽기 액세스 권한 제공

다음 selang 명령은 터미널 tty30 의 ACL 에 내부 사용자 internal_user 를 추가하여 터미널에 대한 읽기 액세스 권한을 제공합니다.

```
authorize TERMINAL tty30 access(READ) uid(internal_user)
```

예: 엔터프라이즈 사용자에게 읽기 액세스 권한 제공

다음 selang 명령은 터미널 tty30 의 ACL 에 엔터프라이즈 사용자 Terry 를 추가하여 터미널에 대한 읽기 액세스 권한을 제공합니다.

```
authorize TERMINAL tty30 access(READ) xuid(Terry)
```

예: 엔터프라이즈 사용자의 액세스 권한을 리소스로 변경

다음 selang 명령은 Terry 의 터미널 tty30 액세스 권한을 없음으로 설정하므로 Terry 의 액세스를 거부합니다.

```
authorize TERMINAL tty30 access(NONE) xuid(Terry)
```

예: 리소스에서 엔터프라이즈 사용자의 액세스 권한 제거

다음 selang 명령은 터미널 tty30 의 ACL 에서 Terry 를 제거합니다.

```
authorize- TERMINAL tty30 xuid(Terry) access-
```

이제 Terry 는 터미널에 대한 기본 액세스 권한을 가집니다.

예: 엔터프라이즈 사용자에게 하위 관리자 액세스 권한 제공

다음 selang 명령은 엔터프라이즈 사용자 Terry 를 사용자와 파일을 관리하는 권한을 가진 하위 관리자로 설정합니다.

```
authorize ADMIN USER xuid(Terry)
authorize ADMIN FILE xuid(Terry)
```

액세스 제어 목록

리소스 액세스 권한은 액세스 제어 목록에 지정됩니다. 모든 리소스 레코드에는 액세스 제어 목록이 여러 개 있습니다.

ACL

리소스 액세스 권한을 부여받은 접근자와 부여받은 액세스 유형을 함께 지정합니다.

NACL

리소스 권한 부여가 거부된 접근자와 거부된 액세스 유형을 함께 지정합니다.

액세스 권한은 사용자가 로컬에서 로그인했는지 여부와 같은 액세스 관련 상황에 따라 달라질 수도 있습니다.

조건부 액세스 제어 목록

CACL(조건부 액세스 제어 목록)은 ACL 에 대한 확장을 제공합니다. 접근자가 리소스에 액세스하려고 할 때 리소스의 ACL 및 NACL 이 사용자의 액세스 권한을 정의하지 않을 경우 CA Access Control 에서는 조건부 액세스 제어 목록을 검사합니다.

조건부 제어 목록은 지정된 프로그램 사용과 같은 특정 방법으로 액세스하는 리소스에 대한 액세스 권한을 지정합니다.

예를 들어 조건부 액세스 제어 목록을 사용하여 프로그램 경로 지정 규칙을 정의할 수 있습니다.

CA Access Control 에서는 다음과 같은 조건부 액세스 제어 목록을 허용합니다.

- 프로그램 액세스 제어 목록(PACL)
- TCP 클래스 액세스 제어 목록
- CALENDAR 클래스 액세스 제어 목록

조건부 액세스 제어 목록 항목에서 항목을 정의하려면 selang authorize 명령의 via 옵션을 사용할 수 있습니다.

다른 액세스 제어 목록과 같이 조건부 액세스 제어 목록의 각 항목은 리소스 액세스 권한을 부여받은 접근자와 부여받은 액세스 유형을 함께 지정합니다. 또한 조건부 액세스 제어 목록의 항목은 권한 할당에 사용되는 조건을 지정합니다. PACL 의 경우 조건은 접근자가 액세스 권한을 갖기 위해 실행해야 하는 프로그램 이름입니다.

예: PACL 사용

엔터프라이즈 사용자 sysadm1 이 프로그램 secured_su 실행을 통해서만 슈퍼 사용자가 되게 하려면 다음 selang 명령을 사용하여 해당하는 조건부 액세스 규칙을 지정합니다.

```
authorize SURROGATE user.root xuid(sysadm1) via(pgm(secured_su))
```

defaccess - 기본 액세스 필드

리소스 레코드에는 기본 액세스 필드인 defaccess 가 포함될 수 있습니다. defaccess 필드 값은 리소스 액세스 제어 목록이 적용되지 않는 접근자에게 허용되는 액세스 권한을 지정합니다.

리소스 액세스 권한을 확인하는 방법

접근자가 리소스에 액세스하려고 하면 CA Access Control 에서는 결과를 얻을 때까지 미리 결정된 순서에 따라 하나 이상의 검사를 실행하여 액세스 권한을 확인합니다. 검사에서 액세스 결과(액세스 거부 또는 허용)가 생성되면 CA Access Control 에서는 추가로 검사하지 않고 결과를 반환합니다.

이러한 검사를 실행하는 순서가 중요합니다. 각 리소스에 대해 CA Access Control 에서는 기본적으로 다음 순서에 따라 액세스 레코드를 검사합니다.

1. 리소스의 시간 기반 제한 사항
2. 리소스 소유권(소유자에게 액세스가 허용됨)
3. B1 검사
4. 리소스 NACL
5. 리소스 ACL

6. 리소스 PACL
7. 리소스 defaccess 필드

마지막 검사 두 개의 순서는 accpacl 옵션에 의해 결정됩니다. selang 명령 setoptions setpacl-를 사용하여 리소스 PACL 사용을 비활성화할 수 있습니다.

하나의 액세스 제어 목록에는 사용자에게 영향을 미치는 항목이 여러 개 포함될 수 있습니다. 예를 들어 사용자를 명시적으로 언급하는 한 항목과 사용자가 속한 각 그룹에 대한 여러 항목을 포함할 수 있습니다. CA Access Control 에서는 다음 수준으로 이동하기 전에 각 수준에서 가능한 항목을 모두 확인합니다. 각 수준에서 충돌하는 규칙을 확인하는 방법에 대한 자세한 내용은 사용자 및 그룹 액세스 권한 간 상호 작용 (페이지 84)을 참조하십시오.

예: 파일에 대한 결과 사용 권한

다음 표에서는 user1 이라는 접근자가 리소스 file1 을 읽으려고 한다고 가정합니다.

다음 표에서 CA Access Control 은 PACL 을 사용하는 accpacl 옵션의 기본 설정을 따르고 있습니다.

user1 에 대한 NACL 의 항목	user1 에 대한 ACL 의 항목	user1 에 대한 PACL 의 항목	defaccess 의 항목	결과 사용 권한
읽기	*(모두)*	*(모두)*	*(모두)*	읽기 거부됨
(정의되지 않음)	없음	*(모두)*	*(모두)*	읽기 거부됨
(정의되지 않음)	읽기	*(모두)*	*(모두)*	읽기 허용됨
(정의되지 않음)	*(정의되지 않음)*	via pgm securereader	*(모두)*	securereader 프로그램을 통해 읽기 허용됨
(정의되지 않음)	*(정의되지 않음)*	*(정의되지 않음)*	읽기	읽기 허용됨

항목이 *(정의되지 않음)*으로 표시되면 해당 액세스 제어 목록에 user1 항목이 없음을 의미합니다.

항목이 *(모두)*로 표시되면 CA Access Control 에서 액세스 제어 목록을 확인하지 않으므로 해당 액세스 제어 목록의 항목이 문제가 없음을 의미합니다.

CA Access Control 에서 확인하는 순서는 왼쪽에서 오른쪽입니다. 모든 행의 경우 거부된 액세스 권한이 포함된 셀의 오른쪽에 있는 셀 값이 *(모두)*임을 알 수 있습니다. 반대로 거부된 액세스 권한이 포함된 셀의 왼쪽에 있는 모든 셀 값은 *(정의되지 않음)*입니다.

사용자 및 그룹 액세스 권한 간 상호 작용

액세스 권한을 사용자 및 사용자가 속한 그룹에 명시적으로 부여하거나 거부할 수 있습니다. 경우에 따라 이러한 권한이 충돌할 수 있습니다. 다음 예에서는 사용자가 그룹 두 개(Group 1 및 Group 2)의 구성원일 때 충돌하는 액세스 권한이 동일한 리소스에 할당될 경우 발생하는 결과를 보여 줍니다.

이 예에서는 누적된 그룹 권한 (페이지 85) 옵션이 설정되었다고 가정합니다(기본 설정).

사용자에 대한 액세스 권한	Group 1 에 대한 액세스 권한	Group 2 에 대한 액세스 권한	결과 액세스 권한
거부된 액세스	*(모두)*	*(모두)*	거부된 액세스
액세스 허용됨	*(모두)*	*(모두)*	액세스 허용됨
(정의되지 않음)	액세스 허용됨	*(정의되지 않음)*	액세스 허용됨
(정의되지 않음)	*(정의되지 않음)*	액세스 허용됨	액세스 허용됨
(정의되지 않음)	액세스 허용됨	액세스 허용됨	액세스 허용됨
(정의되지 않음)	거부된 액세스	*(모두)*	거부된 액세스
(정의되지 않음)	*(모두)*	거부된 액세스	거부된 액세스

항목이 *(정의되지 않음)*으로 표시되면 거부된 사용자 또는 그룹 항목이 없음을 의미합니다.

항목이 *(모두)*로 표시되면 CA Access Control 에서 액세스 권한을 확인하지 않으므로 해당 액세스 권한이 문제가 없음을 의미합니다.

ACCGRR(누적된 그룹 권한)

ACCGRR(*누적된 그룹 권한*) 옵션은 CA Access Control 이 리소스 ACL 을 검사하는 방법에 영향을 미칩니다. ACCGRR 이 활성화되면 CA Access Control 은 사용자가 속한 모든 그룹에서 부여된 권한의 ACL 을 확인합니다. ACCGRR 이 비활성화되면 CA Access Control 은 ACL 에서 적용 가능한 항목에 값 none 이 포함되어 있는지 확인합니다. 값 none 이 포함되어 있으면 액세스가 거부됩니다. 그렇지 않으면 CA Access Control 은 액세스 제어 목록의 첫 번째 적용 가능한 항목을 제외하고 모든 그룹 항목을 무시합니다. 기본적으로 이 옵션은 활성화됩니다.

ACCGRR 옵션을 활성화하려면 다음 selang 명령을 사용합니다.

```
setoptions accgrr
```

ACCGRR 옵션을 비활성화하려면 다음 selang 명령을 사용합니다.

```
setoptions accgrr-
```

보안 수준, 범주 및 레이블

보안 수준과 보안 범주는 리소스 액세스 권한을 제한하여 액세스 제어 목록 사용을 보완하는 방법을 추가로 제공합니다.

보안 레이블은 보안 수준과 보안 범주를 더욱 쉽게 관리할 수 있도록 함께 번들로 제공하는 수단입니다.

보안 수준

*보안 수준*은 접근자와 리소스에 할당할 수 있는 0 에서 255 사이의 정수입니다. 리소스의 액세스 제어 목록에서 사용자에게 액세스 권한이 부여된 경우에도 접근자의 보안 수준이 리소스에 할당된 보안 수준보다 낮으면 접근자는 리소스에 액세스할 수 없습니다. 리소스에 0 보안 수준이 있으면 보안 수준 검사가 해당 리소스를 확인하지 않습니다.

보안 수준이 0 인 접근자는 0 이 아닌 보안 수준을 가진 리소스에 액세스할 수 없습니다.

보안 범주

*보안 범주*는 CATEGORY 클래스의 레코드 이름입니다. 접근자와 리소스에 보안 범주를 할당할 수 있습니다. 접근자는 리소스에 할당된 모든 보안 범주에 할당된 경우에만 리소스에 액세스할 수 있습니다.

보안 레이블

*보안 레이블*은 SECLABEL 클래스의 레코드 이름입니다. 보안 레이블은 보안 수준과 일련의 보안 범주를 함께 번들로 제공합니다. 접근자나 리소스에 보안 레이블을 할당하면 보안 레이블과 관련된 결합 보안 수준 및 보안 범주가 접근자나 리소스에 제공됩니다. 보안 레이블은 접근자나 리소스의 특정 보안 수준 및 범주 할당보다 우선합니다.

예: 보안 레이블 High_Security 사용

보안 수준 255 와 보안 범주 MANAGEMENT 및 CONFIDENTIAL 이 포함된 보안 레이블이 High_Security 라고 가정합니다.

사용자 user1 에 보안 레이블 High_Security 를 할당하면 user1 의 보안 수준은 255 가 되고 보안 범주는 MANAGEMENT 및 CONFIDENTIAL 이 됩니다.

제 6 장: 계정 보호

이 섹션은 다음 항목을 포함하고 있습니다.

사용자 가장 보호

CA Access Control 에서 SURROGATE 클래스를 활성화 때 사용자 가장 보호가 활성화됩니다. 사용자 가장 보호를 사용하면 특정 규칙에서 허용하는 경우에만 사용자 또는 그룹이 자신의 SID(보안 식별자)를 다른 SID 로 변경할 수 있도록 지정할 수 있습니다. 이렇게 하면 허가 받지 않은 사용자가 다른 사용자의 ID 로 가장하는 것을 방지합니다.

참고: 보안 식별자는 운영 체제에 대해 사용자 또는 그룹을 식별하는 숫자 값입니다.

예를 들어, 어떠한 사용자도 Administrator 로 가장할 수 없도록 하는 CA Access Control 규칙을 정의합니다. 이때 사용자가 Tom 이 Administrator 로 일부 작업을 수행하는 프로그램을 실행하려고 시도합니다. CA Access Control 은 Tom 에게 Administrator 로 가장하기 위한 권한이 없으므로 이 프로그램의 실행을 허용하지 않습니다.

다음 두 가지 모드로 사용자 가장 보호를 실행할 수 있습니다.

- 사용자 모드 차단
- 커널 모드 차단

사용자 모드 차단

사용자 모드 차단을 활성화하면 CA Access Control 은 Windows RunAs 유틸리티에서 받는 가장 요청만 차단합니다. 사용자 모드 차단은 지원되는 모든 Windows 버전에서 사용할 수 있습니다.

참고: 사용자 모드 차단은 사용자 가장 보호를 활성화할 때(즉, SURROGATE 클래스를 활성화할 때) 기본적으로 활성화됩니다.

사용자 모드 차단의 이점은 다음과 같습니다.

- CA Access Control 이 원래 가장 요청을 한 사용자를 식별할 수 있습니다.

  RunAs 유틸리티를 포함한 많은 Windows 응용 프로그램에서 NT AUTHORITY\SYSTEM 사용자가 요청하는 사용자를 가장하고 가장 요청을 합니다. 사용자 모드 차단은 요청을 하는 NT AUTHORITY\SYSTEM 사용자가 아닌 유틸리티를 실행하는 사용자를 식별합니다. 예를 들어, Tom 이 Administrator 를 가장하기 위해 RunAs 를 실행하면 NT AUTHORITY\SYSTEM 사용자가 가장 요청을 하고 CA Access Control 이 Tom 을 요청하는 사용자로 식별합니다.

- CA Access Control 은 사용자가 RunAs 유틸리티를 실행할 때만 가장 요청을 차단합니다.

  이렇게 하면 성능에 대한 영향이 최소화됩니다.

사용자 모드 차단의 단점은 CA Access Control 이 모든 Windows 프로세스에서 모든 가장 요청을 차단하지 않는다는 것입니다.

커널 모드 차단

커널 모드 차단을 활성화하는 경우 CA Access Control 은 모든 Windows 프로세스로부터 받는 모든 가장 요청을 차단합니다. 커널 모드 차단은 지원되는 모든 Windows 버전에서 사용할 수 없습니다.

참고: 커널 모드 차단을 사용할 수 없는 Windows 버전에 대한 자세한 내용은 *릴리스 정보*를 참조하십시오.

커널 모드 차단의 장점은 Windows 컴퓨터에서 수행한 모든 가장 요청을 보호할 수 있다는 것입니다.

커널 모드 차단의 단점은 다음과 같습니다.

- NT AUTHORITY\SYSTEM 사용자가 요청하는 사용자를 가장하고 가장 요청을 하면 CA Access Control 은 원래 가장 요청을 한 사용자를 식별하지 않습니다.

  예를 들어, RunAs, ftp, 텔넷 요청은 모두 NT AUTHORITY\SYSTEM 사용자가 수행합니다. Tom 이 Administrator 를 가장하기 위해 RunAs 를 실행하면 NT AUTHORITY\SYSTEM 사용자가 가장 요청을 하고 CA Access Control 이 NT AUTHORITY\SYSTEM 을 요청하는 사용자로 식별합니다.

- CA Access Control 은 OS 가 정상적인 작업의 일부로 수행하는 모든 가장 요청을 차단하므로 성능에 영향을 줄 수 있습니다.

  CA Access Control 이 가장 요청을 캐시에 저장함에도 불구하고 권한 부여 엔진은 여전히 많은 가장 이벤트를 인증해야 합니다.

CA Access Control 이 사용자 가장 요청에 응답하는 방식

SURROGATE 클래스의 각 레코드는 가장 시도로부터 사용자를 보호하는 제한을 정의합니다. CA Access Control 은 가장 요청을 권한 있는 사용자만 액세스할 수 있는 추상 개체로 간주합니다. SURROGATE 클래스의 레코드는 대리(가장)로부터 보호되는 각 사용자 또는 그룹을 나타냅니다.

사용자 또는 그룹이 다른 사용자 또는 그룹으로 가장하기 위한 요청을 하면 CA Access Control 은 다음을 수행합니다.

1. 사용자 또는 그룹에 대한 SURROGATE 레코드의 액세스 권한을 검사합니다. SURROGATE 레코드에 따라 다음 중 *하나*가 발생합니다.
   - 사용자 또는 그룹의 SURROGATE 레코드가 가장을 명시적으로 허용 또는 거부합니다.

     CA Access Control 은 SURROGATE 레코드의 액세스 권한을 사용하여 가장 요청을 허용 또는 거부합니다.
   - 사용자 또는 그룹에 SURROGATE 레코드가 없습니다.

     프로세스가 2 단계로 이동합니다.
2. 다음과 같이 사용자 또는 그룹에 대한 기본 SURROGATE 레코드의 액세스 권한을 검사합니다.
   - 요청자가 사용자인 경우 CA Access Control 은 사용자에게 USER._default SURROGATE 레코드에 정의된 액세스 유형을 부여합니다.
   - 요청자가 그룹인 경우 CA Access Control 은 사용자에게 GROUP._default SURROGATE 레코드에 정의된 액세스 유형을 부여합니다.

   참고: USER._default, GROUP._default, _default SURROGATE 레코드의 기본 액세스 권한은 읽기입니다. 즉, 사용자 또는 그룹에 대한 SURROGATE 레코드가 가장 요청을 금지하지 않는 한, CA Access Control 이 사용자 또는 그룹으로 가장하기 위한 모든 요청을 허용합니다. 이 동작을 변경하려면 USER._default 및 GROUP._default 레코드의 액세스 권한을 변경하십시오. 또한 _default SURROGATE 레코드의 액세스 권한을 변경하여 사용자 및 그룹에 동일한 기본값을 설정할 수도 있습니다.

사용자 가장 보호 활성화

사용자 가장 보호를 사용하면 특정 사용자 및 그룹을 가장하기 위한 요청을 승인 또는 거부하는 규칙을 설정할 수 있습니다.

사용자 가장 보호를 활성화하려면

1. (선택 사항) 다음과 같이 커널 모드 차단을 활성화합니다.
   a. CA Access Control 을 중지합니다.
   b. 다음 레지스트리의 값을 1 로 변경합니다.

      ```
      HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\
      SeOSD\SurrogateInterceptionMode
      ```

   c. CA Access Control 을 다시 시작합니다.

   참고: 사용자 모드 차단은 기본적으로 활성화됩니다.
2. selang 명령 프롬프트 창을 엽니다.
3. SURROGATE 클래스를 활성화합니다.

   ```
   setoptions class+(SURROGATE)
   ```
4. CA Access Control 환경의 SURROGATE 레코드에 대한 selang 규칙을 정의합니다.
5. (커널 모드 차단에만 해당) SYSTEM 사용자가 가장 요청을 하는 사용자를 가장할 수 있게 하는 규칙을 정의합니다.

   ```
   auth SURROGATE USER.Administrator uid("NT AUTHORITY\SYSTEM") acc(R)
   ```

   Windows 는 많은 유틸리티와 서비스(예: Run As)를 유틸리티를 실행하는 사용자가 아닌, 사용자 "NT AUTHORITY\SYSTEM"으로 식별합니다.
   이러한 유틸리티를 실행하는 사용자가 다른 사용자를 가장할 수 있도록 하려면 SYSTEM 사용자에 대한 규칙을 정의해야 합니다.

예: 모든 가장 요청 승인

다음 selang 규칙은 데이터베이스의 규칙이 명시적으로 가장을 금지하지 않는 한, 임의의 사용자가 다른 사용자로 가장할 수 있게 해줍니다.

```
editres SURROGATE _default defaccess(READ)
```

예: 특정 사용자의 가장 금지

다음 selang 규칙은 데이터베이스의 규칙이 사용자 가장을 명시적으로 허용하지 않는 한, 모든 사용자가 Administrator 를 가장하는 것을 금지합니다.

```
newres SURROGATE USER.Administrator defaccess(NONE)
```

예: 그룹이 사용자를 가장하도록 허용

다음 규칙은 Administrators 그룹의 구성원이 Administrator 를 가장하도록 허용합니다.

```
authorize SURROGATE USER.Administrator gid("Administrators")
```

Surrogate DO 기능 설정

작업자, 생산 직원 및 최종 사용자가 슈퍼 사용자만 수행할 수 있는 작업을 수행하는 경우가 있습니다.

기존의 솔루션은 이러한 모든 사용자에게 슈퍼 사용자 암호를 제공하는 것이었지만 이 경우 사이트의 보안이 위협을 받게 됩니다. 암호의 보안을 유지하는 대체 보안 방법을 사용하면 사용자의 일상적인 작업 수행 관련 요청으로 인해 시스템 관리자의 업무량이 늘어납니다.

Surrogate DO(sesudo) 유틸리티로 이 문제를 해결할 수 있습니다. 이 유틸리티로 사용자는 SUDO 클래스에 정의된 동작을 수행할 수 있습니다. SUDO 클래스의 각 레코드에는 스크립트가 포함되고 스크립트를 실행할 수 있는 사용자와 그룹이 지정되며 목적에 따라 필요한 권한이 부여됩니다.

예를 들어, 사용자가 System 역할을 수행하여 "인쇄 스풀러" 서비스를 시작하는 SUDO 리소스를 정의하려면 다음 selang 명령을 입력하십시오.

```
newres SUDO StartSpooler data("net start spooler")
```

newres 명령은 StartSpooler 를 일부 사용자가 해당 System 권한을 가질 수 있는 보호된 작업으로 정의합니다.

중요! 데이터 속성에서는 전체 절대 경로 이름을 사용하십시오. 상대 경로 이름을 사용하면 보호되지 않은 디렉터리에 있는 트로이 목마 프로그램이 실수로 실행될 수 있습니다.

또한 사용자는 authorize 명령을 사용하여 StartSpooler 작업을 수행할 수 있습니다. 예를 들어, *operator1* 사용자가 "인쇄 스풀러" 서비스를 시작할 수 있게 하려면 다음 selang 명령을 입력하십시오.

```
authorize SUDO StartSpooler uid(operator1)
```

또한 authorize 명령을 사용하여 사용자가 보호된 작업을 수행할 수 없도록 할 수 있습니다. 예를 들어, *operator2* 사용자가 "인쇄 스풀러" 서비스를 시작하지 못하게 하려면 다음 명령을 입력합니다.

```
authorize SUDO StartSpooler uid(operator2) access(None)
```

sesudo 유틸리티를 실행하면 보호된 작업이 수행됩니다. *예를 들어,* operator1 사용자는 다음 명령을 사용하여 "인쇄 스풀러" 서비스를 시작합니다.

```
sesudo -do StartSpooler
```

sesudo 유틸리티는 처음에는 사용자가 SUDO 작업을 수행할 수 있는 권한이 있는지 여부를 확인한 다음 사용자가 리소스에 대한 권한이 있는 경우 리소스에 정의되어 있는 명령 스크립트를 실행합니다. 위의 예제에서 sesudo 는 operator1 이 StartSpooler 작업을 수행할 수 있는 권한이 있는지 확인한 후 System 자격 증명으로 "net start spooler" 명령을 호출합니다.

참고: sesudo 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

SUDO 레코드 정의(작업 위임)

SUDO 클래스의 레코드에는 사용자가 빌린 권한으로 스크립트를 실행할 수 있는 명령 스크립트가 저장되어 있습니다. 권한을 빌려올 수 있는 자격은 스크립트를 실행하는 sesudo 명령과 SUDO 레코드가 엄격하게 제어합니다.

참고: 대화형 Windows 응용 프로그램에 대해 SUDO 레코드를 만드는 경우 SUDO 레코드에 대한 대화형 플래그를 설정해야 합니다. 대화형 플래그를 설정하지 않는 경우 응용 프로그램이 백그라운드로 실행되고 사용자가 응용 프로그램과 상호 작용할 수 없습니다. 자세한 내용은 *문제 해결 안내서*를 참조하십시오.

SUDO 레코드에서 comment 속성은 특수한 용도로 사용되며 흔히 data 속성이라는 다른 이름으로 알려져 있습니다.

comment 속성 값은 명령 스크립트이며 금지하거나 허용할 스크립트 매개 변수 값을 하나 이상 이 속성 값에 선택적으로 추가할 수 있습니다. 트로이 목마 바이러스가 침입하는 것을 방지하려면 전체 comment 속성 값을 작은따옴표로 묶어야 하고 실행 파일을 전체 경로 이름으로 참조해야 합니다.

comment 속성의 형식은 다음과 같습니다.

```
comment('cmd[;[prohibited-values][;permitted-values]]')
```

금지된 값과 허용된 값의 목록은 선택 사항이므로 간단한 comment 속성 값은 다음과 같을 수 있습니다.

```
newres SUDO NET comment('net use')
```

이 명령에서 간단한 값은 sesudo NET 명령이 'net use' 명령을 실행한다는 의미입니다. 금지되는 특정 스크립트 매개 변수 값은 없으며, 모두 허용됩니다.

와일드카드와 적합한 변수를 사용하면 금지된 매개 변수와 허용된 매개 변수를 다양하게 지정할 수 있습니다. 사용할 수 있는 와일드카드는 표준 Windows 와일드카드입니다. 금지된 매개 변수와 허용된 매개 변수는 다음 변수를 포함할 수 있습니다.

변수	설명
$A	영문자 값
$G	기존의 CA Access Control 그룹 이름
$H	(UNIX 전용) 사용자의 홈 디렉터리로 시작되는 매개 변수
$N	숫자 값
$O	sesudo 를 실행하는 사용자의 CA Access Control 이름
$U	기존 CA Access Control 사용자 이름
$e	비어 있는 항목. 규칙에 대한 매개 변수가 없는 SUDO 명령을 지정할 때 사용합니다
$f	기존 파일 이름
$g	기존 Windows 그룹 이름
$h	기존 호스트 이름

변수	설명
$r	Windows 읽기 권한이 있는 기존 파일
$u	기존 Windows 사용자 이름
$w	Windows 쓰기 권한이 있는 기존 파일
$x	Windows 실행 권한이 있는 기존 파일

금지된 매개 변수 값의 목록을 스크립트에 추가하는 경우:

- 금지된 매개 변수 값과 스크립트를 세미콜론으로 구분하지만, 앞뒤에 작은 따옴표를 입력해야 합니다. 예를 들어 사용자가 -start 매개 변수를 제외한 다른 모든 매개 변수를 사용할 수 있도록 지정하려면 다음 명령을 입력하십시오.

  ```
  newres SUDO scriptname comment('cmd;-start')
  ```

  여기서 *cmd* 는 스크립트를 나타냅니다.

  또한 매개 변수 값을 허용하지 않으면서 모든 매개 변수를 기본값으로 사용하려면 다음과 같이 SUDO 레코드를 정의하십시오.

  ```
  newres SUDO scriptname comment('cmd;*')
  ```

- script 매개 변수에 둘 이상의 금지된 값이 있는 경우, 공백을 구분자로 사용하십시오. 예를 들어 사용자가 -start 및 -stop 매개 변수를 제외한 다른 모든 매개 변수를 사용할 수 있도록 지정하려면 다음 명령을 입력하십시오.

  ```
  newres SUDO scriptname comment('cmd;-start -stop')
  ```

- 둘 이상의 script 매개 변수에 금지된 값이 있는 경우, 파이프 문자(|)를 금지된 값 집합 사이의 구분자로 사용하십시오. 예를 들어 사용자가 스크립트의 첫 번째 매개 변수로 -start 및 -stop 을 사용하지 못하도록 하고 두 번째 매개 변수로 기존 Windows 사용자 이름을 사용하지 못하도록 하려면(앞의 변수 목록 참조) 다음 명령을 입력하십시오.

  ```
  newres SUDO scriptname comment('cmd;-start -stop | $u')
  ```

  스크립트에 나열한 매개 변수보다 많은 수의 매개 변수가 있는 경우 금지된 매개 변수의 마지막 집합은 나머지 모든 매개 변수에 적용됩니다.

허용된 매개 변수 값의 목록을 스크립트에 추가하는 경우:

- sesudo 유틸리티에서는 두 가지 검사를 실행합니다.
  - 매개 변수 값이 *금지된* 해당 값 중 어느 것과도 일치하지 *않는지* 검사합니다.
  - 매개 변수 값이 *허용된* 해당 값과 최소한 하나라도 일치하는지 검사합니다.

  따라서 매개 변수 값이 금지된 목록에 있으면 허용된 목록에 지정되어 있더라도 허용되지 않습니다.
- *허용된* 값의 목록과 *금지된* 값의 목록을 세미콜론으로 구분하지만, 두 개의 목록 앞뒤에는 각각 작은 따옴표를 입력해야 합니다. 금지된 값의 목록이 없는 경우에도 세미콜론은 필요합니다. 세미콜론이 없으면 허용하려고 했던 값이 금지될 수 있습니다. 예를 들어 스크립트에 대한 매개 변수 값으로 NAME 값만 허용하려면 다음 명령을 입력하십시오.

  ```
  newres SUDO scriptname comment('cmd;;NAME')
  ```
- 다른 목록에서도 다음 작업을 수행하십시오.
  - script 매개 변수에 둘 이상의 허용된 값이 있는 경우, 공백을 구분자로 사용하십시오.
  - 둘 이상의 script 매개 변수에 허용된 값이 있는 경우, 파이프 문자(|)를 허용된 값 집합 사이의 구분자로 사용하십시오.

  예를 들어 두 개의 매개 변수가 있을 때 첫 번째 매개 변수는 숫자여야 하지만 Windows 사용자 이름을 사용할 수 없으며, 두 번째 매개 변수는 영문자여야 하지만 Windows 그룹 이름을 사용할 수 없는 경우에는 다음 명령을 입력하십시오.

  ```
  newres SUDO scriptname comment('cmd;$u | $g ;$N | $A')
  ```

  스크립트에 나열한 매개 변수보다 많은 수의 매개 변수가 있는 경우 허용된 매개 변수의 마지막 집합은 나머지 모든 매개 변수에 적용됩니다.

따라서 comment 속성의 전체 형식은 먼저 스크립트가 온 다음 금지된 값이 매개 변수별로 오고 마지막으로 허용된 값이 매개 변수별로 옵니다.

```
comment('cmd; \
param1_prohib1 param1_prohib2 ... param1_ prohibN | \
param2_prohib1 param2_prohib2 ... param2_ prohibN | \
 ...
paramN_prohib1 paramN_prohib2 ... paramN_prohibN ; \
param1_permit1 param1_permit2 ... param1_ permitN | \
param2_permit1 param2_permit2 ... param2_ permitN |
 ...
```

```
paramN_permit1 paramN_permit2 ... paramN_permitN')
```

Sesudo 유틸리티는 사용자가 입력한 각 매개 변수를 다음과 같은 방식으로 검사합니다.

1. N 매개 변수가 허용된 매개 변수 N 과 일치하는지 테스트합니다. 허용된 매개 변수 N 이 없을 경우 허용된 마지막 매개 변수가 사용됩니다.
2. N 매개 변수가 금지된 매개 변수 N 과 일치하는지 테스트합니다. 금지된 매개 변수 N 이 없을 경우 금지된 마지막 매개 변수가 사용됩니다.

모든 매개 변수가 허용된 매개 변수와 일치하며 금지된 매개 변수와 일치하는 매개 변수가 없을 경우에만 sesudo 는 명령을 실행합니다.

예: 사용자가 net send 를 실행하도록 허용하는 작업 위임 설정

다음 절차는 사용자 Takashi 가 net send 명령을 실행하는 것은 허용하고 net start 명령을 실행하는 것은 금지하는 방법을 보여줍니다.

1. CA Access Control 끝점 관리에서 "사용자" 탭을 클릭한 다음 "권한 부여 및 위임" 하위 탭을 클릭합니다.

   "권한 부여 및 위임" 메뉴 옵션이 왼쪽에 나타납니다.
2. "작업 위임"을 클릭합니다.

   "작업 위임" 페이지가 나타납니다.
3. "작업 만들기"를 클릭합니다.

   "작업 만들기" 페이지가 나타납니다.
4. 다음과 같이 대화 상자 필드를 완료합니다.

필드	값
이름	NET
데이터	net;start;send *
소유자	nobody
기본 액세스	없음(옵션 지움)
권한이 있는 접근자	USER: Takashi Allow: Execute

"저장"을 클릭합니다.

새 작업 위임(SUDO) 레코드가 작성됩니다.

5. 작업 위임 규칙을 테스트합니다.
   a. Takashi 로 로그인합니다.
   b. 명령 프롬프트를 열고 다음 명령을 실행합니다.

      ```
      sesudo -do NET start
      ```

      다음과 같은 메시지가 나타납니다.

      ```
      sesudo: 'start'을(를) 매개 변수 번호 1(으)로 사용할 수 없습니다.
      ```

      참고: *net start* 는 금지된 값으로 정의되었기 때문에 실행되지 않습니다.
   c. 다음 값을 실행합니다.

      ```
      sesudo -do NET send comp message
      ```

      명령이 실행됩니다.

예: 대화형 응용 프로그램을 사용하여 권한 있는 작업을 수행하도록 사용자에게 권한 부여

사용자는 다음 예제와 같이 스냅인 MSC 모듈을 사용하여 높은 수준의 권한이 필요한 작업을 수행할 수 있습니다.

1. CA Access Control 끝점 관리에서 "사용자" 탭을 클릭한 다음 "권한 부여 및 위임" 하위 탭을 클릭합니다.

   "권한 부여 및 위임" 메뉴 옵션이 왼쪽에 나타납니다.
2. "작업 위임"을 클릭합니다.

   "작업 위임" 페이지가 나타납니다.
3. "작업 만들기"를 클릭합니다.

   "작업 만들기" 페이지가 나타납니다.
4. 다음과 같이 대화 상자 필드를 완료합니다.

필드	값
이름	services
데이터	c:\winnt\system32\mmc.exe
소유자	nobody
옵션	대화형(옵션 선택함)
기본 액세스	없음(옵션 지움)

필드	값
권한이 있는 접근자	USER: Tori Allow: Execute

"저장"을 클릭합니다.

새 작업 위임(SUDO) 레코드가 작성됩니다. 대화형 옵션은 서비스가 시작되었을 때 로그인한 사용자가 사용할 수 있는 데스크톱 사용자 인터페이스를 제공합니다. 이 기능은 서비스가 LocalSystem 계정으로 실행 중인 경우에만 사용할 수 있습니다.

5. 작업 위임 규칙을 테스트합니다.
   a. Tori 로 로그인합니다.
   b. 명령 프롬프트를 열고 다음 명령을 실행합니다.

      ```
      sesudo -do services
      ```

   c. mmc.exe 가 시작됩니다.

사용자 비활성 상태 검사

비활성 기능을 사용하면 조직에 더 이상 소속되지 않은 소유자의 계정으로 시스템에 무단 액세스하는 것을 금지할 수 있습니다. 비활성 일은 사용자가 로그인하지 않는 날입니다. 사용자 계정이 일시 중지되거나 로그인할 수 없는 제한 비활성 기간 일수를 지정할 수 있습니다. 계정이 일시 중지되면 해당 계정을 수동으로 다시 활성화해야 합니다.

참고: 비활성 검사에서 암호 변경은 활동으로 간주됩니다. 사용자의 암호가 변경되면 해당 사용자는 비활성이 되기 때문에 일시 중단할 수 없습니다.

USER 클래스 레코드 또는 GROUP 클래스 레코드의 비활성 속성으로 비활성 기간의 일 수를 설정할 수 있습니다. GROUP 클래스 레코드는 그룹을 프로필 그룹으로 가지는 사용자에게만 영향을 미칩니다. SEOS 클래스의 INACT 속성으로 전체 시스템에서 모든 사용자에 대한 비활성을 설정할 수도 있습니다.

selang 에서 다음 명령을 사용하여 비활성을 전체적으로 지정하십시오.

```
setoptions inactive (numdays)
```

그룹에 대한 전체 시스템의 비활성 설정보다 우선적되는 그룹에 대한일수를 설정하려면 다음 명령을 사용합니다.

```
editgrp groupName inactive (numdays)
```

사용자에 대한 그룹 및 전체 시스템 설정보다 우선되는 사용자에 대한 일수를 설정하려면 다음 명령을 사용합니다.

```
editusr userName inactive (numdays)
```

일시 중지된 사용자 계정을 다시 활성화하려면 다음 명령을 사용합니다.

```
editusr userName resume
```

일시 중지된 프로필 그룹을 다시 활성화하려면 다음 명령을 사용합니다.

```
editgrp userName resume
```

전체 시스템 수준의 비활성 로그인 확인을 비활성화하려면 다음 명령을 사용합니다.

```
setoptions inactive-
```

그룹에 대한 비활성 로그인 검사를 비활성화하려면 다음 명령을 사용합니다.

```
editgrp groupName inactive-
```

사용자에 대한 비활성 로그인 검사를 비활성화하려면 다음 명령을 사용합니다.

```
editusr userName inactive-
```

제 7 장: 사용자 암호 관리

이 섹션은 다음 항목을 포함하고 있습니다.

암호 관리 및 잠금 정책

암호는 가장 많이 사용되는 인증 방식이지만 암호 보호 방법은 여러 잘 알려진 문제점들이 있습니다. 쉬운 암호는 추측하기 쉽고, 수년 동안 바꾸지 않고 번갈아 사용하는 암호는 결과적으로 노출됩니다. 네트워크에서 일반 텍스트로 암호를 보내면 수신자가 암호를 가로챌 수도 있습니다.

Windows 에는 이와 같은 일반적인 위험을 방지하기 위해 사용자가 암호를 사용하는 경우 지켜야 하는 일련의 암호 규칙과 정책이 있습니다. CA Access Control 에는 사용자가 더욱 안전한 암호를 선택할 수 있도록 돕는 추가 규칙이 있습니다.

CA Access Control 에서 다음 규칙을 지정할 수 있습니다.

- 새 암호는 이전 암호와 동일할 수 없습니다. CA Access Control 이 저장하는 기존 암호의 개수는 암호 정책에서 지정됩니다.
- 새 암호에는 사용자 이름이 들어갈 수 없습니다.
- 새 암호는 변경 중인 암호를 포함할 수 없습니다.
- 새 암호는 바꾸려는 현재 암호와 같을 수 없습니다. CA Access Control 은 대/소문자를 무시합니다.
- 새 암호에는 암호 정책에서 지정된 최소 영숫자, 특수 문자, 숫자, 소문자 및 대문자를 사용해야 합니다.

- 새 암호에는 암호 정책에서 지정된 수보다 많은 반복 문자가 나올 수 없습니다.
- 새 암호는 CA Access Control 에 포함된 사전의 제한된 단어 중 하나가 될 수 없습니다. 사전은 다음 레지스트리 하위 키의 Dictionary 값에 지정됩니다.

  HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\passwd

각 암호에는 최대 수명이 있어야 합니다. 즉, 사용자가 특정 간격 이후 새 암호를 선택하도록 암호가 만료되어야 합니다.

- 각 암호에는 최소 수명이 있어야 합니다. 최소 수명을 지정하여 사용자가 자주 반복적으로 암호를 변경하는 것을 금지할 수 있습니다. 암호가 자주 변경되면, 암호 내역 스택에 오버플로를 유발하여 기존 암호를 재사용할 수 있습니다.

암호 품질 검사 구성

암호 품질 검사를 구성하려면

1. CA Access Control 끝점 관리에서 "구성" 탭을 클릭합니다.

   구성 메뉴 옵션이 왼쪽에 나타납니다.

2. "기타" 섹션 옵션에서 "클래스 활성화"를 클릭합니다.

   "클래스 활성화" 페이지가 나타납니다.

3. "사용자 ID 제어" 섹션에서 PASSWORD 를 선택하고 "저장"을 클릭합니다.

   암호 품질 검사가 활성화됩니다.

4. "정책" 섹션 옵션에서 "사용자 암호 정책"을 클릭합니다.

   "사용자 암호 정책" 페이지가 나타납니다.

5. 암호 검사에 사용할 규칙을 정의하고 "저장"을 클릭합니다.

   이에 암호 검사에 대해 정의한 규칙이 암호가 변경될 때 적용됩니다.

6. (UNIX 에만 해당) sepass 유틸리티를 사용하여 새 암호를 업데이트합니다.

   참고: sepass 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

예: 암호 검사 규칙 정의

다음 selang 명령은 암호 품질 검사를 활성화하고 다음과 같은 최소값을 적용하는 암호 규칙을 정의합니다.

- 영숫자 문자 6 자
- 소문자 3 자
- 숫자 2 자

```
setoptions class+ (PASSWORD)
setoptions password(rules(alpha("6") lowercase("3") numeric("2")))
```

참고: setoptions 명령 형식에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

오류 메시지 확인

Windows NT 시스템에서 사용자 암호를 설정하는 경우, 다음 메시지가 나타날 수 있습니다.

```
암호가 요구된 것보다 짧습니다.
```

이 오류는 암호가 정책 요구 사항에 맞지 않는다는 의미입니다. 오류 발생 원인은 다음 중 하나입니다.

- 암호가 필요한 길이보다 짧거나 깁니다.
- 최근에 사용된 적이 있으며 Windows NT 변경 내역 필드에 존재하는 암호입니다.
- 암호의 고유 문자가 부족합니다.
- 암호가 다른 암호 정책 요구 사항(예: CA Access Control 암호 정책에 따라 설정된 요구 사항)과 맞지 않습니다.

이 오류를 방지하려면, 적용되는 모든 요구 사항에 적합한 암호를 설정해야 합니다.

제 8 장: 모니터 및 감사

이 섹션은 다음 항목을 포함하고 있습니다.

보안 감사자

보안 감사자 및 시스템 관리자의 가장 중요한 작업 중 하나는 시스템 활동을 감사하거나 모니터링하여 의심스럽거나 올바르지 않은 활동을 감지하는 것입니다. 보안 감사는 보안 환경에서 중요한 역할을 수행하며, CA Access Control 의 보안 감사 특징은 다음과 같습니다.

- 누가 시스템에 액세스했는지, 어떤 리소스에 액세스했는지, 어떤 방법으로 리소스에 액세스했는지(예: 파일 읽기), 언제 리소스에 액세스했는지를 신뢰할 수 있게 표시
- 시도가 실패한 경우에도 보안 위반이 시도되었으면 해당 사용자에게 통지 및 경고
- 보안 규칙의 변경 사항 및 변경한 사용자 표시
- 적용 전에 액세스 규칙 효과 테스트 방법 제공

CA Access Control 감사는 실제 수행되는 감사를 모델로 합니다. 사용자가 구현 내용을 변경할 수 있더라도 보안 감사자는 시스템 및 보안 관리자와 독립적으로 활동하며, 일부 다른 모델이 사용자 환경에 대해 더 적합한 경우에도 보안 감사자의 활동을 제약할 수 없습니다.

보안 감사자는 AUDITOR 특성이 할당되는 사용자입니다. 보안 감사자로 정의된 사용자는 사용자 및 리소스에 할당된 감사 규칙 변경과 같은 감사 작업을 수행할 수 있습니다. 또한 ADMIN 특성을 갖지 않고도 CA Access Control 감사 유틸리티를 사용할 수 있는 권한을 가집니다.

이벤트 차단

다음 두 조건이 충족되는 경우 CA Access Control 은 이벤트를 차단합니다.

- 적절한 클래스가 활성화된 경우
- 데이터베이스에 이 이벤트를 기대하는 규칙이 있는 경우

예를 들어, 다음과 같은 일반 규칙을 사용하여 c:\data\payroll 디렉터리에 있는 파일에 대한 모든 파일 액세스를 감사할 수 있습니다.

```
newres FILE c:\data\payroll\*
```

FILE 클래스가 활성화(기본 설정)되어 있는지 확인해야 합니다.

차단된 이벤트 유형

CA Access Control 은 다음 두 종류의 이벤트를 차단합니다.

- 차단 이벤트

  차단 이벤트에 포함된 정보는 감사 이벤트가 나중에 사용할 수 있도록 프로세스의 일부로서 캐시에 저장됩니다.
- 감사 이벤트

차단 모드

차단 모드를 기준으로 CA Access Control 은 차단을 실행하고, 권한 부여를 검사하고, 액세스 요청 이벤트의 감사 레코드를 로그에 기록합니다. CA Access Control 에는 다음과 같은 차단 모드가 있습니다.

- 전체 적용 모드
- 감사 전용 모드
- 차단 사용 안 함 모드

참고: 경고 모드 (페이지의 정의 참조 108)는 차단 모드가 아닙니다. 경고 모드는 구현 중에 임시로 사용하기 위한 모드로서 전체 적용 모드에서만 작동합니다.

감사 전용 모드

*감사 전용 모드*는 액세스 규칙을 검사 또는 적용하지 않은 채 차단된 모든 이벤트를 기록합니다. 이 모드를 사용하여 규정 또는 법규에 대한 데이터를 수집할 수 있습니다. 감사 전용 모드에서 CA Access Control 은 이벤트를 차단하고 감사 이벤트를 기록하지만, 권한 부여에 대한 요청을 처리하거나 규칙을 적용하지는 않습니다. 그 결과로 CA Access Control 은 차단하는 모든 액세스 요청을 허용합니다. 즉, 모든 이벤트에 대해 감사 로그에 기록된 권한 부여 결과는 *P*('P'ermitted - 허용됨)가 됩니다.

감사 전용 모드에는 다음과 같은 제한이 있습니다.

- Unicenter 에는 어떠한 감사 레코드도 전달되지 않습니다.

  감사 전용 모드에서 모든 이벤트는 허용(*P*)됩니다. 허용되는 이벤트는 Unicenter 에 전달되지 않습니다.

- 리소스와 사용자의 감사 속성은 *고려되지 않습니다*.

  감사 전용 모드는 리소스 또는 사용자 고유 설정에 관계없이 *차단된 모든* 이벤트를 기록합니다.

감사 전용 모드 설정

*감사 전용 모드*는 액세스 규칙을 검사 또는 적용하지 않은 채 차단된 모든 이벤트를 기록합니다. 이 모드를 사용하여 규정 또는 법규에 대한 데이터를 수집할 수 있습니다.

감사 전용 모드를 설정하려면 SeOSD\GeneralInterceptionMode CA Access Control 레지스트리 항목을 1 로 설정하십시오.

중요! 감사 전용 모드를 사용하는 경우 감사 로그를 위한 충분한 디스크 공간이 있는지, 그리고 감사 로그의 크기 제한이 충분히 큰지를 확인하십시오. 또한 감사 로그 백업 (페이지 141)을 위한 옵션도 고려해야 합니다.

경고 모드

*경고 모드*는 리소스에 적용할 수 있는 속성이고 클래스에 적용할 수 있는 옵션입니다. 경고 모드가 리소스나 클래스에 적용되고 액세스가 액세스 규칙을 위반하면 CA Access Control 은 감사 로그 항목과 반환 코드 W 를 기록하지만 리소스 액세스를 허용합니다. 클래스가 경고 모드에 있는 경우에는 해당 클래스의 모든 리소스가 경고 모드에 있습니다.

CA Access Control 이 전체 적용 모드에 있는 경우에만 경고 모드가 적용됩니다.

참고: "전체 적용 모드"는 UNIX 용 CA Access Control 에서 지원하는 유일한 모드입니다. Windows 용 CA Access Control 에서는 감사 전용 모드도 지원합니다.

액세스 정책을 도입하거나 수정할 때 경고 모드를 사용할 수 있습니다. 이 작업을 수행할 경우 감사 로그를 검토하여 원하는 정책을 적용하기 전에 해당 정책 결과를 미리 볼 수 있습니다. seaudit 명령을 사용하여 감사 로그를 표시할 수 있습니다.

클래스에 *warning* 속성이 있으면 해당 클래스에 경고 모드를 적용할 수 있습니다. 리소스 그룹이나 클래스가 경고 모드에 있으면 액세스 규칙이 위반될 때 CA Access Control 은 액세스를 허용하고 리소스 그룹이나 클래스가 아니라 리소스를 참조하는 항목을 감사 로그에 기록합니다.

리소스와 클래스에 대한 경고 모드 설정은 독립적입니다. 리소스에 경고 모드를 적용하면 리소스가 클래스에 속하고 해당 클래스에서 경고 모드를 제거하는 경우에도 리소스가 경고 모드로 유지됩니다.

참고: 리소스나 클래스에 *warning* 속성이 있는 경우에만 해당 리소스나 클래스에 경고 모드를 적용할 수 있습니다. 일부 리소스나 클래스에는 이 속성이 포함되어 있지 않습니다.

추가 정보:

감사 전용 모드 (페이지 107)

리소스에 경고 모드 적용

리소스에 경고 모드를 적용하여 액세스 규칙을 적용할 필요 없이 해당 규칙의 영향을 모니터링할 수 있습니다.

참고: 개별 리소스에 경고 모드를 적용할 뿐만 아니라 클래스에 경고 모드를 적용 (페이지 110)할 수 있습니다.

리소스에 경고 모드를 적용하려면

1. CA Access Control 끝점 관리에서 경고 모드를 적용하려는 리소스를 편집합니다.

   해당 "수정" 페이지가 나타납니다.

2. "감사" 탭을 클릭합니다.

   리소스의 "감사 모드" 페이지가 나타납니다.

3. "경고 모드"를 선택하고 "저장"을 클릭합니다.

   이제 수정한 리소스가 경고 모드에 있습니다.

참고: 경고 모드에서 CA Access Control 은 액세스가 허용되지만 액세스 규칙이 위반될 때 항상 경고 레코드를 감사 로그에 기록합니다. 이 작업을 수행하기 위해 리소스에 대한 audit 속성을 설정할 필요가 없습니다.

sereport 유틸리티(보고서 번호 6)를 사용하여 경고 모드에서 모든 리소스를 확인할 수 있습니다.

예: 파일에 경고 모드 적용

다음 selang 예에서는 c:\myfile 파일에 경고 모드를 적용합니다.

```
chres FIlE c:\myfile warning
```

예: 파일에서 경고 모드 지우기

다음 selang 예에서는 c:\myfile 파일을 경고 모드에서 해제합니다.

```
chres FIlE c:\myfile warning-
```

이제 myfile 에 대한 경고 모드가 활성 상태가 아니므로 CA Access Control 은 myfile 의 액세스 규칙을 적용합니다.

예: 터미널에 경고 모드 적용

다음 selang 예에서는 myterminal 터미널에 경고 모드를 적용합니다.

```
chres terminal myterminal warning
```

CA Access Control 은 myterminal 터미널을 통한 권한 있는 사용자의 액세스를 허용하지만 일반적으로 해당 터미널을 통한 액세스가 거부된 사용자에 대한 감사 레코드를 기록합니다.

클래스에 경고 모드 적용

개별 레코드에 경고 모드를 적용하지 않고 클래스의 모든 레코드에 경고 모드를 적용할 수 있습니다. 경고 모드를 사용하여 액세스 규칙을 적용하지 않고 해당 규칙을 모니터링할 수 있습니다.

경고 모드로 클래스 사용

1. CA Access Control 끝점 관리에서 다음을 수행하십시오.
   a. "구성"을 클릭합니다.
   b. "클래스 활성화"를 클릭합니다.

   "클래스 활성화" 페이지가 나타납니다.
2. "경고" 열에서 경고 모드로 사용할 클래스에 대한 확인란을 선택합니다.
3. "저장"을 클릭합니다.

   CA Access Control 옵션이 성공적으로 업데이트되었음을 알리는 메시지가 표시됩니다.

경고 모드에 어떤 리소스가 있는지 알아보기

CA Access Control 을 구현할 때 경고 모드를 임시 방법으로 사용해야 합니다. 사용자들이 필요한 리소스에 대해 필요한 액세스를 가지고 있는 것으로 확인되면 경고 모드를 종료해야 합니다. 그러면 CA Access Control 이 관련된 규칙의 적용을 시작합니다.

경고 모드에 어떤 리소스가 있는지 알아보려면 경고 모드와 함께 모든 리소스를 보여주는 보고서를 작성할 수 있습니다.

보고서를 작성하려면 다음 명령을 입력합니다.

```
sereport -f pathname.html -r 6
```

CA Access Control 이 보고서를 작성합니다.

참고: sereport 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

경고 모드에 있는 클래스 찾기

CA Access Control 을 구현할 때 경고 모드를 임시 방법으로 사용해야 합니다. 사용자들이 필요한 리소스에 대해 필요한 액세스를 가지고 있는 것으로 확인되면 경고 모드를 종료해야 합니다. 그러면 CA Access Control 이 관련된 규칙의 적용을 시작합니다.

경고 모드에 있는 클래스를 찾으려면 CA Access Control 이 이 데이터를 표시하게 합니다.

이 데이터를 표시하려면 다음 selang 명령을 입력합니다.

```
setoptions cwarnlist
```

CA Access Control 이 경고 모드에 있는 클래스를 나타내는 테이블을 표시합니다.

참고: setoptions 에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

시스템 유지 관리를 수행하는 방법

언젠가는 시스템을 업그레이드하고 새 응용 프로그램을 설치하는 등의 시스템 유지 관리를 수행해야 합니다. 시스템 유지 관리 중에 경고 모드에서 CA Access Control 규칙을 설정해야 합니다. 유지 관리로 인해 사용자들이 필요한 리소스에 대해 필요한 액세스가 영향을 받지 않는 것으로 판단되면 경고 모드를 종료해야 합니다. 그러면 CA Access Control 이 관련된 규칙의 적용을 시작합니다.

시스템 유지 관리를 수행할 때 경고 모드를 사용하려면 다음을 수행하십시오.

1. 유지 관리를 시작하기 전에 다음 selang 규칙을 사용하여 경고 모드에 적절한 클래스를 설정합니다.

   ```
   setoptions class(NAME) flags(W)
   ```

2. 유지 관리를 수행합니다.
3. 유지 관리를 수행한 이후에 seretrust 를 실행합니다.

   seretrust 유틸리티는 프로그램을 다시 트러스트하고 데이터베이스에 정의된 파일을 보호하기 위해 필요한 selang 명령을 생성합니다.
4. selang 명령을 실행하여 데이터베이스 정의된 프로그램을 다시 트러스트합니다.
5. 정책을 시행하기 위해 다음 selang 규칙을 사용하여 클래스에서 경고 모드를 제거합니다.

   ```
   setoptions class(NAME) flags-(W)
   ```

6. CA Access Control 감사 로그 파일을 검토합니다.

   감사 로그는 유지 관리에 따른 영향을 받은 리소스에 대한 경고를 수록하고 있습니다.

참고: seretrust 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

Access Control 활동 모니터

CA Access Control 추적은 CA Access Control 에서 수행한 모든 작업을 보여줄 수 있는 실시간 로그입니다. 추적 레코드는 *ACInstallDir*\log\seosd.trace(*ACInstallDir* 은 <eAC 을> 설치한 디렉터리임)에 누적됩니다.

또는 다음과 같이 레지스트리 하위 키에 *trace_file* 값으로 지정한 파일마다 누적됩니다.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\SeOSD\
```

추적 파일에서 레코드를 필터링할 수 있지만 추적 메커니즘은 보안 감사가 아닌 시스템 모니터를 위해 설계되었습니다.

기본적으로 CA Access Control 은 CA Access Control 초기화 중에만 추적 메시지를 생성합니다. CA Access Control 이 초기화되면 추적 메커니즘이 중지되므로 추적 메시지가 생성되지 않습니다.

추적 레코드 필터

CA Access Control 은 두 가지 유형의 추적 레코드를 만듭니다.

- 사용자 추적 레코드 - 사용자에 의해 완료된 레코드 작업(예:user1 이 c:\tmp\tmp.exe 파일을 액세스함).
- 일반 추적 레코드 - 시스템에 의해 완료된 레코드 작업(예: Watchdog 이 프로그램을 트러스트되지 않음으로 설정).

추적 레코드는 seos.trace 파일에 기록되며 trcfilter.ini 파일을 사용하여 필터링할 수 있습니다.

사용자를 추적 가능하게 설정하면, 해당 사용자에 대해 추적 레코드가 기록될 때마다 seos.audit 파일에 일치하는 감사 레코드가 기록됩니다. 감사 레코드는 audit.cfg 파일에 의해 필터링됩니다.

참고: 추적 이벤트에 의해 생성된 감사 레코드는 캐시되지 않으며 항상 전체 실행 흐름을 따라 처리됩니다.

다음 selang 명령은 사용자를 추적 가능하게 만듭니다.

```
editusr userName audit(trace)
```

추적 또는 감사 레코드를 보려면 seaudit 유틸리티를 사용하십시오.

추적 레코드 필터링

추적 필터 파일을 사용하여 특정 유형의 활동이 추적 파일에 표시되지 않도록 지정할 수 있습니다. 추적 필터 파일은 다음 레지스트리 키의 *trace_filter* 값을 사용하여 지정합니다.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\SeOSD
```

기본값은 *ACInstallDir*\log\trcfilter.ini(*ACInstallDir* 은 CA Access Control 을 설치한 디렉터리임)입니다.

중요! *seosd.trace* 단일 행을 사용하여 설치할 경우 CA Access Control 은 추적 필터 파일을 작성합니다. 이 레코드를 절대로 삭제하지 마십시오.

추적 필터 파일의 각 행은 추적하지 *않아야* 하는 액세스 또는 활동을 나타냅니다. 예를 들어, Microsoft Word 에 대한 사용자 액세스 추적을 제거하려면, 추적 필터 파일에 다음 줄을 추가하십시오.

```
*winword.exe*
```

CA Access Control 감사 대상

보안 감사를 위해 CA Access Control 은 차단된 이벤트의 감사 레코드를 데이터베이스에 정의된 감사 규칙 및 실행 중인 적용 모드에 따라 유지합니다. 감사 로그의 레코드는 이러한 감사 규칙에 따라 누적됩니다.

전체 감사는 다음과 같은 차단된 모든 이벤트에 대한 감사 레코드를 제공합니다.

- 파일 액세스(FILE 클래스)
- 프로그램 실행(PROGRAM 클래스)
- 레지스트리 액세스(REGKEY 및 REGVAL 클래스).
- 가장(impersonation) 제어(SURROGATE 클래스)
- 네트워크 제어(CONNECT, TCP, HOST, GHOST, HOSTNET, HOSTNP 클래스)
- 로그인(TERMINAL 클래스)

  참고: 차단된 로그인 이벤트는 캐시에 저장되지 않고 항상 차단 이벤트에 대한 감사 프로세스를 따릅니다.
- 서비스 보호(WINSERVICE 클래스)

- 암호 확인 실패(PASSWORD 클래스)
- 프로세스 종료(PROCESS 클래스)

이벤트를 로그할지 여부는 CA Access Control 차단 모드에 의해 결정됩니다.

로그인 차단 제한 사항

Windows 의 로그인 차단은 CA Access Control 하위 인증 방식에서만 지원됩니다.

커널을 통해 로그인 차단을 설정할 수 없습니다. 그 결과 다음을 고려해야 합니다.

- Windows 도메인 환경에서 하위 인증 구성 요소는 DC(도메인 컨트롤러) 수준에서 작동하며, 어떤 DC 가 사용자의 로그인 이벤트를 인증하고 CA Access Control 하위 인증 모듈을 트리거할 것인지를 OS 가 결정하므로, CA Access Control 을 모든 DC 에 설치해야 합니다.
- Windows 도메인 환경에서 작업할 때 CA Access Control 로그인 정책(TERMINAL 규칙)은 DC 에 있어야 하며 대상 서버에는 있을 필요가 없습니다.

  Windows 도메인의 일부이며 DC 가 아닌 파일 서버에서 도메인 사용자가 만든 로그인 이벤트를 보호 또는 감사하려는 경우, 대상 파일 서버가 아니라 DC 에서 CA Access Control 로그인 정책을 정의해야 합니다. 그 이유는 도메인 사용자가 공유 파일 디렉터리에 액세스할 때 파일 서버가 아니라 DC 에서 로그인 권한 부여가 발생하기 때문입니다.
- DC 가 여러 개인 경우 CA Access Control 로그인 권한 부여는 이 DC 중 하나에서 처리될 수 있습니다. 따라서 모든 DC 간에 CA Access Control 로그인 정책을 동기화하는 것이 좋습니다.

  이 작업은 모든 DC 가 PMDB 의 구독자인 정책 모델 메커니즘을 통해 구현할 수도 있고, 모든 DC 를 호스트 그룹에 추가한 후 고급 정책 관리를 사용하여 공통된 정책을 배포함으로써 구현할 수도 있습니다.

- 로그인 이벤트와 상응하는 일부 사용자 속성은 런타임에(이벤트 권한 부여 중에) 업데이트됩니다. 로그인 권한 부여는 DC 중 하나에서만 발생하므로 이러한 속성이 동기화되지 않을 수 있습니다. 이러한 속성에는 *유예 로그인*, *마지막으로 액세스한 날짜* 및 *마지막 액세스 시간*이 있습니다.

  예를 들어, CA Access Control 하위 인증은 모든 DC 가 아니라 DC 중 하나에서 트리거되므로 사용자 속성인 *마지막 액세스 시간* 값이 DC 간에 다를 수 있습니다.

- 로컬 사용자(도메인 사용자가 아님) 로그인 이벤트를 적용하려면 로컬 사용자가 액세스해야 할 로컬 컴퓨터에 CA Access Control 을 설치해야 합니다. 이는 로컬 컴퓨터가 도메인 컴퓨터로 사용되기 때문입니다(도메인이 로컬 컴퓨터임).

- RDP(원격 데스크톱 프로토콜)/터미널 서비스 로그인 이벤트는 대상 서버에 적용되는데, 이는 대상 서버에 이전 CA Access Control 버전이 있었기 때문입니다. 그러나 대상 서버에서 RDP 로그인 이벤트에 대해 CA Access Control 로그인 정책을 정의해야 합니다.

전체 적용 모드에서 CA Access Control 감사 대상

전체 적용 모드(정상 동작)에서 CA Access Control 은 다음과 같이 이벤트를 로깅합니다.

- 경고 모드가 차단된 리소스에 대해 *꺼져* 있으면 CA Access Control 은 규칙을 적용하고 리소스 또는 사용자의 *감사* 속성을 기준으로 이벤트를 로그합니다.

감사 속성	로그에 기록되는 이벤트
ALL	*모두*
SUCCESS	허용된 액세스
FAIL	거부된 액세스

- 경고 모드가 차단된 리소스에 대해 *켜져* 있으면 액세스 요청이 액세스 규칙을 위반하는 경우 감사 로그에 기록이 작성됩니다(이때 만약 규칙이 적용되면 요청은 실패함). 감사 레코드는 경고 모드가 사용되었으므로 해당 규칙 위반이 허용되었음을 나타냅니다.

  이 모드에서는 규칙이 적용되지 않습니다.

감사 전용 모드에서 CA Access Control 감사 대상

감사 전용 모드에서 CA Access Control 은 권한 부여 요청을 처리하거나 규칙을 적용하지 않습니다. 액세스의 성공 또는 실패 여부에 상관없이 액세스를 시도한 사람의 차단된 모든 로그인 이벤트와 CA Access Control 에서 보호되는 리소스에 대한 차단된 모든 이벤트가 로그에 기록됩니다.

CA Access Control 이 감사 로그에 기록하는 내용을 변경하는 방법

다음과 같은 두 가지 방법으로 CA Access Control 이 감사 로그에 기록하는 내용을 변경할 수 있습니다.

- 리소스 또는 접근자의 AUDIT 속성을 사용하여 CA Access Control 이 감사 로그에 기록하는 감사 이벤트를 정의합니다.

  참고: GROUP 또는 XGROUP 의 AUDIT 속성을 사용하여 그룹의 모든 구성원에 대한 감사 속성을 설정할 수 있습니다. 하지만 사용자의 감사 모드가 USER 레코드, XUSER 레코드 또는 프로필 그룹에 정의되어 있는 경우 AUDIT 속성을 사용하여 그룹 구성원에 대한 감사 모드를 설정할 수 없습니다.

- 감사 구성 파일 audit.cfg 를 사용하여 CA Access Control 이 audit log 로 보내는 이벤트를 필터링합니다. audit.cfg 파일을 사용하여 감사 로그에 이벤트를 추가할 수 없습니다.

감사 레코드의 수를 줄이기 위해 로그 파일에 기록된 연속적인 감사 이벤트를 조정할 수도 있습니다. 이러한 사용자 지정은 일치하는 연속적인 감사 이벤트(즉, 동일한 프로세스 ID, 스레드 ID, 규칙 ID, 사용자 ID, 액세스 마스크를 갖는 하나의 리소스에 대한 액세스) 사이의 시간 간격을 기준으로 조정합니다. 시간 간격(초)은 AuditRefreshPeriod 레지스트리 항목의 값을 지정하여 설정할 수 있습니다. 기본적으로 AuditRefreshPeriod 는 0 으로 설정되며, 이 경우 모든 이벤트가 로그 파일에 기록됩니다.

감사 규칙 설정

보안 감사를 위해 CA Access Control 는 데이터베이스에 정의된 감사 규칙에 따라 액세스 거부 또는 허용 이벤트에 대한 감사 레코드를 유지합니다.

모든 접근자와 리소스는 다음 값 중 하나 이상으로 설정될 수 있는 AUDIT 속성을 가집니다.

FAIL

접근자의 리소스 액세스 실패를 로그 파일에 기록합니다.

SUCCESS

접근자의 리소스 액세스 성공을 로그 파일에 기록합니다.

LOGINFAIL

접근자의 모든 로그온 실패를 로그 파일에 기록합니다. 이 값은 리소스에 적용되지 않습니다.

참고: 로그인 이벤트에는 암호 시도 이벤트(A LOGIN)와 로그인 이벤트(P/D/W LOGIN)의 두 가지 유형이 있습니다. 자세한 내용은 *참조 안내서*를 참조하십시오.

중요: 암호 시도 이벤트는 Unix 에서만 유효합니다.

LOGINSUCCESS

접근자의 모든 성공적인 로그온을 로그 파일에 기록합니다. 이 값은 리소스에 적용되지 않습니다.

ALL

접근자의 경우 FAIL, SUCCESS, LOGINFAIL 및 LOGINSUCCESS 와 같은 정보를 로그 파일에 기록하고, 리소스의 경우 FAIL 및 SUCCESS 와 같은 정보를 로그 파일에 기록합니다.

NONE

접근자 또는 리소스와 관련된 어떤 정보도 로그 파일에 기록하지 않습니다.

데이터베이스에서 접근자 또는 리소스 레코드를 작성하거나 업데이트할 때마다 AUDIT 속성을 지정할 수 있습니다. 또한 기록된 이벤트의 전자 메일 통지를 전송해야 하는지 여부와 전자 메일 통지의 수신인을 지정할 수 있습니다.

감사 로그의 레코드는 이러한 감사 규칙에 따라 누적됩니다. 이벤트를 어떤 기준에 따라 로그 파일에 기록할지 여부는 다음 사항에 따라 결정됩니다.

- 리소스 또는 접근자에 AUDIT(ALL)이 있는 경우, CA Access Control 에서 보호되는 해당 리소스에 대한 접근자와 모든 이벤트에 대한 모든 로그인 이벤트가 액세스 성공 여부에 상관 없이 기록됩니다.
- CA Access Control 에서 보호되는 리소스에 대한 액세스가 성공하고 사용자 또는 리소스에 AUDIT(SUCCESS)가 있는 경우 이벤트가 기록됩니다.
- CA Access Control 에서 보호되는 리소스에 대한 액세스가 실패하고 접근자 또는 리소스에 AUDIT(FAIL)이 있는 경우 이벤트가 기록됩니다.

또한 사용자를 추적 가능으로 설정하면 해당 사용자에 대한 추적 레코드가 기록될 때마다 해당 감사 레코드가 감사 로그에 기록됩니다.

CA Access Control 이 감사 로그에 기록하는 감사 이벤트 정의

CA Access Control 은 감사 로그에 성공 및 실패한 액세스를 기록합니다. 감사 대상 접근자 또는 리소스의 AUDIT 속성 값을 변경하여 CA Access Control 이 감사 로그에 기록하는 액세스 이벤트를 정의합니다. 이 방법을 사용하여 CA Access Control 이 감사 로그에 모든 추적 이벤트를 기록하도록 지정할 수도 있습니다.

AUDIT 속성을 사용하여 CA Access Control 이 감사 로그에 기록하는 감사 이벤트를 지정합니다. selang 또는 CA Access Control 끝점 관리를 사용하여 다음과 같이 리소스 또는 접근자에 대한 AUDIT 속성을 설정하십시오.

AUDIT 값	CA Access Control 이 기록하는 대상	해당되는 대상
FAIL	액세스 실패	사용자 및 리소스
SUCCESS	액세스 성공	사용자 및 리소스
LOGINFAIL	로그인 실패	사용자
LOGINSUCCESS	로그인 성공	사용자
ALL	FAIL, SUCCESS, LOGINFAIL, LOGINSUCCESS, INTERACTIVE 에 해당	사용자 및 리소스

AUDIT 값	CA Access Control 이 기록하는 대상	해당되는 대상
TRACE	ALL 및 모든 시스템 이벤트에 해당	사용자
INTERACTIVE	UNIX 컴퓨터의 사용자 세션	사용자
NONE	로깅 없음	사용자 및 리소스

참고: 사용자의 감사 속성이 설정되어 있지 않으면 그룹 또는 프로필 그룹의 AUDIT 값이 CA Access Control 이 해당 사용자에 대해 사용하는 감사 모드에 영향을 줄 수 있습니다.

CA Access Control 이 사용자의 감사 모드를 결정하는 방법

사용자에 대한 감사 모드는 CA Access Control 이 해당 사용자에 대한 감사 로그로 전달하는 감사 이벤트를 지정합니다. 다음 프로세스는 CA Access Control 이 사용자에 대한 감사 모드를 결정하는 방식을 설명합니다.

1. CA Access Control 은 USER 또는 XUSER 클래스에 있는 사용자의 레코드에 AUDIT 속성 값이 있는지 확인합니다.

   사용자의 레코드에 AUDIT 속성 값이 있으면 CA Access Control 은 이 값을 해당 사용자의 감사 모드로 사용합니다.

2. CA Access Control 은 사용자가 프로필 그룹에 할당되었는지 여부를 확인합니다. 사용자가 프로필 그룹에 할당된 경우 CA Access Control 은 GROUP 클래스의 프로필 그룹 레코드에 AUDIT 속성 값이 있는지 확인합니다.

   사용자가 프로필 그룹에 할당되어 있고 프로필 그룹의 레코드에 AUDIT 속성 값이 있는 경우 CA Access Control 은 이 값을 해당 사용자의 감사 모드로 사용합니다.

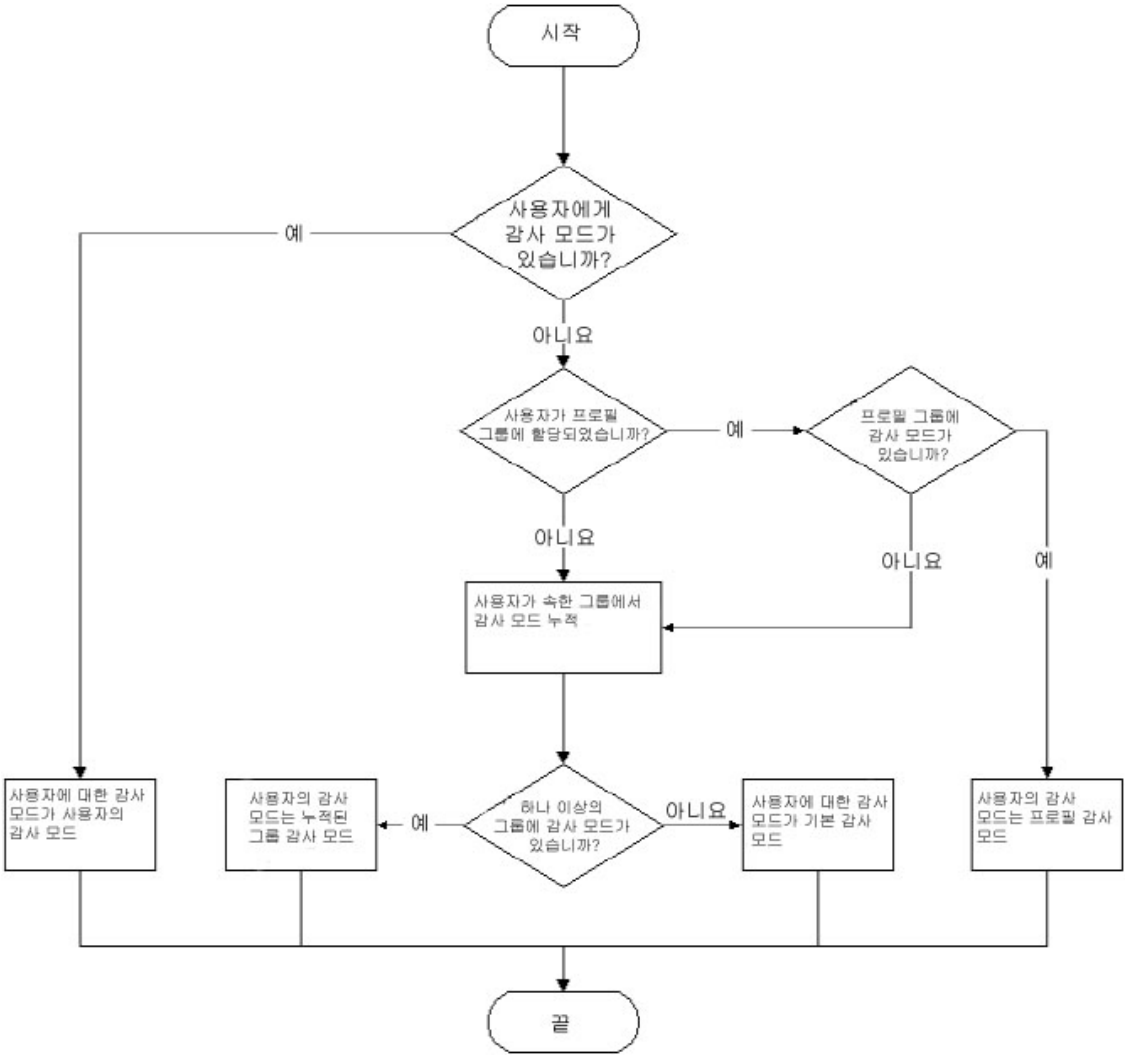
3. CA Access Control 은 사용자가 그룹의 구성원이지 확인합니다. 사용자가 그룹 구성원인 경우 CA Access Control 은 GROUP 또는 XGROUP 클래스의 그룹 레코드에 AUDIT 속성 값이 있는지 확인합니다.

   사용자가 그룹 구성원이고 그룹의 레코드에 AUDIT 속성 값이 있는 경우 CA Access Control 은 이 값을 해당 사용자의 감사 모드로 사용합니다. 사용자가 그룹 구성원이 아니거나 그룹의 레코드에 AUDIT 속성 값이 없는 경우 CA Access Control 은 해당 사용자에게 시스템 전체 감사 모드를 할당합니다.

   참고: 사용자가 서로 다른 감사 모드를 갖는 여러 그룹에 속한 구성원인 경우 사용자의 감사 모드는 누적됩니다. 이러한 사용자의 감사 모드는 속한 그룹의 모든 감사 모드의 합계입니다.

참고: CA Access Control 이 그룹의 AUDIT 속성 값을 사용하여 사용자의 감사 모드를 결정하는 경우, 사용자가 로그인된 상태에서 그룹의 감사 모드를 변경하면 로그인된 사용자의 감사 모드 또한 변경됩니다. 그룹 감사 모드의 변경 사항이 반영되도록 사용자가 로그오프할 필요는 없습니다.

다음 그림은 CA Access Control 이 사용자에 대한 감사 모드를 결정하는 방식을 설명합니다.



예: 그룹별 감사

사용자 Jan 은 그룹 A 및 그룹 B 의 구성원입니다. 그룹 A 의 감사 모드는 'FAIL'이고 그룹 B 의 감사 모드는 'SUCCESS'입니다. Jan 이 두 그룹 모두의 구성원이므로 Jan 은 누적된 감사 모드인 'FAIL' 및 'SUCCESS'를 갖습니다.

추가 정보:

CA Access Control 이 프로필 그룹을 사용하여 사용자 속성을 파악하는 방법 (페이지 47)

사용자 및 엔터프라이즈 사용자의 기본 감사 모드

사용자(USER 개체)를 만들 때 CA Access Control 은 개체에 기본 AUDIT_MODE 를 할당합니다. AUDIT_MODE 속성의 기본 값은 Failure, SuccessLogin, SuccessFailure 입니다.

엔터프라이즈 사용자(XUSER 개체)를 만들 때 기본적으로 CA Access Control 은 개체에 기본 AUDIT_MODE 값을 할당하지 않습니다.

참고: (UNIX) USER 개체에 대한 AUDIT_MODE 속성의 기본값을 변경하려면 lang.ini 파일의 [newusr] 섹션에 있는 DefaultAudit 값을 편집하십시오.

일부 사용자에 대해 기본 감사 값으로 변경

r12.0 SP1 CR1 이전 버전에서는 다음 접근자에 대한 기본 감사 모드가 'None'이었습니다.

- 해당 USER 클래스 레코드에 정의된 AUDIT 값이 없는 사용자 및 정의된 AUDIT 값이 있는 프로필 그룹에 연결되지 않은 사용자
- 데이터베이스에 정의되지 않은 모든 사용자(_undefined 사용자 레코드로 표시됨)

  참고: 엔터프라이즈 사용자의 경우 CA Access Control 은 어떠한 사용자도 정의되지 않은 사용자로 간주하지 않습니다. _undefined 사용자의 속성은 이 경우 관련이 없습니다.

r12.0 SP1 CR1 부터 이러한 접근자에 대한 기본 감사 모드는 Failure, LoginSuccess, LoginFailure 입니다. 이전 방식을 사용하려면 이러한 사용자에 대해 AUDIT 속성의 값을 'None'으로 지정하십시오.

GROUP 레코드에 대한 AUDIT 속성 값 변경

다음의 두 가지 기능이 있는 GROUP 레코드가 있는 경우:

- 한 세트의 사용자에 대한 감사 정책을 정의하는 프로필
- 두 번째 사용자 세트에 대한 컨테이너

r12.0 SP1 CR1 부터 GROUP 레코드도 두 번째 사용자 세트에 대한 감사 정책을 정의합니다. 이 방식 변경에 따른 잠재적인 문제를 방지하려면 두 번째 사용자 세트에 대한 별도의 GROUP 을 만드십시오.

Windows 에서 감사 정책 설정

접근자 및 리소스에 대한 액세스 규칙을 설정하는 것 이외에도 감사 로그에 기록할 Windows 이벤트를 지정할 수 있습니다. 그룹, 프로필 그룹 또는 개별 사용자를 기준으로 전체 조직에 대해 이러한 감사 정책을 지정할 수 있습니다.

예: 프로필 그룹의 모든 구성원에 대한 감사 정책 설정

다음 예는 프로필 그룹에 속한 모든 사용자에 대해 감사 정책을 설정하는 방법을 보여줍니다.

1. 필요한 감사 모드로 새 프로필 그룹을 만듭니다. 예:

   ```
   newgrp profileGroup audit(failure) owner(nobody)
   ```

2. 새 사용자를 만들어 위에서 만든 프로필 그룹에 추가합니다. 예:

   ```
   newusr user1 profile(profileGroup) owner(nobody)
   ```

3. 사용자의 감사 설정을 제거합니다. 예:

   ```
   chusr user1 audit-
   ```

이제 이 설정이 유효한지 검사할 수 있습니다.

1. 다음과 같이 새 사용자로 로그온합니다.

   ```
   runas /user:user1 cmd.exe
   ```

2. user1 의 명령 프롬프트 창에서 다음을 입력합니다.

   ```
   secons -whoami
   ```

   이 명령을 입력하면 권한 부여에 사용되며 user1 의 ACEE 에 보관되는 정보가 표시됩니다.

   ```
   ACEE audit mode is: Failure; Originated from Profile group definition
   ```

   이 메시지를 통해 감사 정책이 사용자가 추가된 프로필 그룹에서 파생되었음을 알 수 있습니다.

예: 그룹 구성원에 대한 감사 정책 설정

이 예제에서는 가상의 회사인 "Forward Inc"가 CA Access Control 을 사용하여 /production 디렉터리에 있는 모든 파일을 보호하려고 합니다. /production 디렉터리는 네이티브 환경에서 전체 액세스 권한을 가지고 있습니다.

Forward Inc 는 /production 디렉터리에 대한 액세스를 거부하고 모든 액세스 시도를 감사 기록하려고 합니다. 하지만 Forward Inc 는 개발자들에게는 /production 디렉터리에 대한 읽기 액세스를 부여하려고 합니다. 이 액세스는 감사 기록하지 않습니다. 개발자가 /production 디렉터리에 쓰기를 시도하면 이 시도가 거부되고 감사 기록됩니다.

개발자는 /production 디렉터리에 대한 전체 액세스를 요구할 수 있습니다. Forward Inc 는 전체 액세스 권한을 가진 사용자가 /production 디렉터리에서 수행하는 모든 작업을 감사 기록합니다.

다음 프로세스는 앞의 시나리오를 구현하기 위해 Forward Inc 가 수행하는 절차를 설명합니다.

1. 네이티브 환경에서 "Developers"란 이름의 그룹을 만듭니다. 모든 개발자를 이 그룹에 추가하십시오.
2. 네이티브 환경에서 "Dev_Access_All"이란 이름의 그룹을 만듭니다. 이 그룹에 어떤 사용자도 추가하지 마십시오.
3. 다음과 같이 /production 디렉터리에 대한 일반 액세스 규칙을 정의합니다.

   ```
   authorize FILE /production/* access(none) uid(*)
   ```

   이 규칙은 기본 액세스를 지정하지 않습니다.
4. 다음과 같이 /production 디렉터리에 대한 일반 감사 규칙을 정의합니다.

   ```
   editres FILE /production/* audit(failure)
   ```

   이 규칙은 /production 디렉터리에 대한 모든 실패한 액세스 시도를 감사 기록합니다.
5. 다음과 같이 "Developers" 그룹에 대한 액세스 규칙을 정의합니다.

   ```
   authorize FILE /production/* access(read) xgid(Developers)
   ```

   이 규칙은 "Developers" 그룹의 구성원에게 /production 디렉터리에 대한 읽기 액세스를 부여합니다.

   참고: 4 단계에서 설정한 규칙은 개발 그룹의 구성원을 포함한 모든 사용자에 의한 모든 실패한 액세스 시도가 CA Access Control 에서 감사 기록될 수 있도록 해 줍니다.

6. 다음과 같이 "Dev_Access_All" 그룹에 대한 액세스 규칙을 정의합니다.

   ```
   authorize FILE /production/* access(all) xgid(Dev_Access_All)
   ```

   이 규칙은 "Dev_Access_All" 그룹의 구성원에게 /production 디렉터리에 대한 전체 액세스 권한을 부여합니다.

7. 다음과 같이 "Dev_Access_All" 그룹에 대한 감사 규칙을 정의합니다.

   ```
   chxgrp Dev_Access_All audit(all)
   ```

   이 규칙은 Dev_Access_All 그룹의 구성원이 수행하는 모든 작업을 감사 기록합니다.

8. "Developers" 그룹의 한 구성원에게 /production 디렉터리에 대한 전체 액세스 권한이 필요한 경우 이 사용자를 네이티브 환경에서 Dev_Access_All 그룹에 추가합니다.

   이렇게 하면 해당 사용자는 /production 디렉터리에 대한 전체 액세스 권한을 갖게 되며 CA Access Control 은 이 사용자가 수행하는 모든 작업을 감사 기록합니다.

   참고: 그룹 구성원의 변경 사항이 반영되려면 사용자가 새 로그온 세션을 시작해야 합니다.

9. 이 사용자가 /production 디렉터리에서 작업을 완료한 다음에는 이 사용자를 네이티브 환경에서 Dev_Access_All 그룹으로부터 제거합니다.

   이제 이 사용자는 /production 디렉터리에 대한 읽기 액세스 권한을 갖습니다. 이 사용자가 /production 디렉터리에 대한 다른 액세스를 시도하면 CA Access Control 은 액세스를 거부하고 이 시도를 감사 기록합니다.

   참고: 그룹 구성원의 변경 사항이 반영되려면 사용자가 새 로그온 세션을 시작해야 합니다.

감사 프로세스

감사 필요성에 맞게 CA Access Control 을 구성하려면 우선 감사가 동작하는 방식에 대해 이해해야 합니다. 감사 기능을 사용하면 CA Access Control 이 차단한 액세스 요청(이벤트)을 추적할 수 있습니다. 이 데이터를 사용하여 시행되는 규정을 준수하고, 보안 조건에 맞춰 액세스 규칙을 분석 및 세부 조정하고, 액세스 요청을 모니터링할 수 있습니다.

감사 이벤트를 로그에 기록하기 위해 CA Access Control 이 따르는 프로세스는 다음과 같이 차단하는 이벤트의 유형에 따라 다릅니다.

- 차단 이벤트 (페이지 128)

  참고: 통합된 로그인 이벤트(TERMINAL 클래스) 및 사용자 추적에 의해 생성된 감사 레코드는 캐시에 저장되지 않으며 항상 차단 이벤트를 위한 감사 프로세스를 따릅니다.

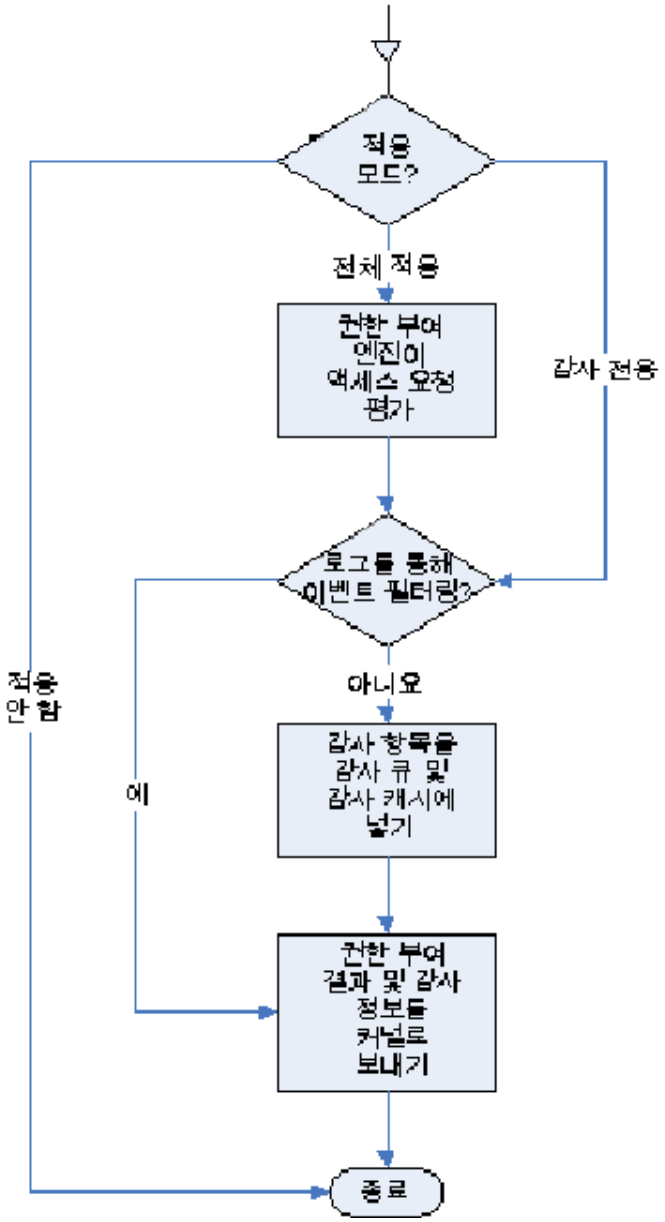
- 감사 이벤트 (페이지 130)

참고: CA Access Control 은 적절한 클래스가 활성화되어 있고 데이터베이스에 이 이벤트를 기대하는 규칙이 있는 경우에만 이벤트를 차단합니다.

차단 이벤트에 대해 감사 기능이 동작하는 방식

*차단 이벤트*는 CA Access Control 이 처음 접하는 이벤트이며, 차단 이벤트에 대해 어떠한 권한 부여 정보 또는 감사 정보도 커널 캐시에 존재하지 않습니다.

감사 레코드를 로그에 기록하기 위해 CA Access Control 은 다음 작업을 수행하고 차단 이벤트에 대해 이러한 효과를 발생시킵니다.



- 적용 안 함 모드에서는 이벤트가 차단되거나 감사가 실행되지 않습니다.
- 전체 적용 모드에서 CA Access Control 은 다음을 수행합니다.

  1. 권한 부여 결과를 기준으로 권한 부여 엔진이 감사 항목을 감사 큐 및 감사 캐시에 저장합니다.

     CA Access Control 은 리소스 또는 액세스 시도자에 대한 감사 속성이 결과 이벤트를 감사하도록 설정되어 있고 감사 필터 파일이 이 이벤트를 필터링하도록 설정되지 않은 경우에만 감사 항목을 기록합니다.

  2. 권한 부여 엔진은 감사 관련 정보 및 권한 부여 결과에 대한 상세 내용을 커널에 반환합니다.

- 감사 전용 모드에서 CA Access Control 은 권한 부여 요청을 처리하지 않습니다. 감사 정보는 리소스 및 사용자의 감사 속성에 관계없이 항상 기록됩니다.

  CA Access Control 은 감사 필터 파일이 이 이벤트를 필터링하도록 설정되지 않은 경우에만 감사 항목을 기록합니다. 이 모드의 권한 부여 결과는 항상 *P*('P'ermitted - 허용됨)입니다.

참고: 차단된 로그인 이벤트(TERMINAL 클래스)와 사용자 추적에 의해 생성된 감사 레코드는 캐시에 저장되지 않으며 권한 부여 엔진은 항상 이러한 이벤트에 대한 감사 레코드를 기록합니다.

감사 이벤트에 대해 감사 기능이 동작하는 방식

다음 도표와 단계는 감사가 감사 이벤트에 대해 작동하는 방식을 설명합니다.

캐시에 저장된 차단 이벤트를 커널이 CA Access Control 에 통보하면 CA Access Control 은 다음 동작을 수행하여 감사 이벤트를 로그합니다.

1. 커널이 보낸 정보에 들어 있는 감사 캐시를 사용하여 감사 데이터를 재구성합니다.
2. 감사 큐에 감사 항목을 넣습니다.

커널 및 감사 캐시

*커널 캐시*에는 이전에 차단된 이벤트에 대한 데이터가 포함되어 있습니다. 커널은 그러한 캐시된 차단된 이벤트(감사 이벤트)를 식별하고, 처리를 위해 이러한 이벤트를 CA Access Control 로 보냅니다. 결과적으로 CA Access Control 은 커널 캐시를 사용하여 이전에 차단된 이벤트와 동일한 패턴을 따르는 이벤트를 차단합니다.

*감사 캐시*에는 CA Access Control 이 권한 부여 프로세스를 따를 필요 없이 반복되는 감사 레코드를 재구성하여 감사 큐로 보내기 위해 사용할 수 있는 데이터가 포함되어 있습니다. 따라서 캐시에 충분한 정보가 이미 들어 있는 차단된 이벤트(감사 이벤트)는 신속히 처리되어 감사 큐에 추가됩니다. 권한 부여 엔진은 차단한 최초 이벤트(차단 이벤트)의 결과로부터 감사 캐시와 커널에 저장된 데이터를 제공합니다.

캐시 재설정

CA Access Control 은 다음과 같은 경우에 커널과 감사 캐시를 모두 삭제합니다.

- 데이터베이스가 변경되는 경우

  CA Access Control 은 데이터베이스 정보가 변경될 때 모든 캐시를 삭제합니다. 새 액세스 규칙 또는 수정된 액세스 규칙은 잠재적으로 기존 캐시를 부정확하게 만듭니다.

- 시간 검사점에 도달하는 경우

  시간 검사점이 임의의 이벤트에 대한 권한 부여 결과에 영향을 줄 경우 CA Access Control 은 전체 캐시를 삭제합니다. DAYTIME 제한 속성 또는 HOLIDAY 클래스 레코드가 변경될 때 권한 부여 결과도 변경될 수 있으며 캐시가 잠재적으로 부정확하게 될 수 있습니다.

- PROGRAM 리소스가 변경되는 경우

  watchdog 이 PROGRAM 리소스가 변경되어 언트러스트된 것으로 인식하게 되면 CA Access Control 은 전체 캐시를 삭제합니다. 언트러스트된 프로그램은 해당 프로그램과 관련된 권한 부여 요청의 결과에 영향을 줍니다. 그 결과로 캐시가 잠재적으로 부정확하게 됩니다.

- 감사 캐시가 꽉 찬 경우

  감사 캐시가 가득 차면 CA Access Control 은 사용 빈도가 낮은 순서대로 캐시 항목의 10%를 삭제합니다.

캐시가 삭제된 이후에는 캐시를 다시 채우고 CA Access Control 이 감사 이벤트를 차단할 수 있도록 새 차단 이벤트의 정보가 필요합니다.

감사 이벤트 보기

CA Access Control 은 감사 이벤트를 감사 로그로 보냅니다. 감사 로그는 다음 CA Access Control 도구를 사용하여 볼 수 있습니다.

- CA Access Control 끝점 관리
- seaudit 유틸리티

Windows 이벤트 로그로 감사 이벤트도 전달하도록 CA Access Control 을 구성할 수도 있습니다. 이벤트 로그는 여러 응용 프로그램에서 가져온 감사 이벤트를 하나의 컬렉션에 저장합니다. Windows 이벤트 뷰어를 사용하여 이벤트 로그에 있는 감사 이벤트를 볼 수 있습니다.

Windows 이벤트 로그에 있는 감사 이벤트

Windows 이벤트 로그는 여러 출처에서 가져온 감사 이벤트를 하나의 컬렉션에 저장합니다. CA Access Control 이 감사 이벤트를 이벤트 로그로 전달하도록 구성하면 seosd 가 CA Access Control 감사 로그에 감사 이벤트를 기록할 때마다 해당 이벤트가 이벤트 로그로 전달됩니다.

audit.cfg 파일은 감사 로그 및 이벤트 로그 모두에서 감사 이벤트를 필터링합니다. 감사 이벤트가 감사 로그에 기록되지 않으면 이벤트 로그로 전달되지 않습니다.

감사 이벤트의 원래 응용 프로그램, 크기, 대상에 따라 Windows 2008 이벤트 로그는 또한 감사 이벤트를 채널이라고 불리는 컨테이너로 보냅니다. CA Access Control 채널은 CA-AccessControl-AuthorizationEngine/Audit 으로 명명됩니다.

Windows 2008 Server 에 CA Access Control 을 배포한 경우 다음 대상으로 감사 이벤트를 보내도록 선택할 수 있습니다.

- 이벤트 로그
- 채널
- 이벤트 로그 및 채널
- 이벤트 로그 또는 채널 제외

Windows 이벤트 로그로 감사 이벤트 보내기

CA Access Control 이 감사 이벤트를 Windows 이벤트 로그로 전달하도록 구성하면 seosd 가 CA Access Control 감사 로그에 감사 이벤트를 기록할 때마다 해당 이벤트가 이벤트 로그로 전달됩니다. 또한 CA Access Control 이 이벤트 로그로 정책 모델 감사 이벤트를 보내도록 구성할 수도 있습니다.

이벤트 로그로 이벤트를 보내려면

1. 다음 명령을 사용하여 CA Access Control 을 중지합니다.

   ```
   secons -s
   ```

   CA Access Control 이 중지됩니다.

2. logmgr 섹션의 SendAuditToNativeLog 구성 설정의 값을 1 로 설정합니다.

   감사 이벤트가 Windows 이벤트 로그로 전달됩니다.

3. (선택 사항) Pmd 섹션의 SendAuditToNativeLog 구성 설정의 값을 1 로 설정합니다.

   정책 모델의 감사 이벤트가 Windows 이벤트 로그로 전달됩니다.

4. 다음 명령을 사용하여 CA Access Control 을 다시 시작합니다.

   ```
   seosd -start
   ```

   CA Access Control 이 다시 시작됩니다.

예: 이벤트 로그로 감사 이벤트 보내기

다음 예는 이벤트 로그로 감사 이벤트를 보냅니다. 이 명령을 사용하려면 원격 구성 환경(env config)을 사용해야 합니다.

```
er config ACROOT section(logmgr) token(SendAuditToNativeLog) value(1)
```

예: 이벤트 로그로 정책 모델 감사 이벤트 보내기

다음 예는 이벤트 로그로 정책 모델 감사 이벤트를 보냅니다. 이 명령을 사용하려면 원격 구성 환경(env config)을 사용해야 합니다.

```
er config ACROOT section(Pmd) token(SendAuditToNativeLog) value(1)
```

추가 정보:

구성 설정 변경 (페이지 200)

Windows 이벤트 로그 채널로 감사 이벤트 보내기

Windows Server 2008 에만 해당

CA Access Control 이 감사 이벤트를 Windows 이벤트 로그 채널로 전달하도록 구성하면 seosd 가 CA Access Control 감사 로그에 감사 이벤트를 기록할 때마다 해당 이벤트가 이벤트 로그 채널로 전달됩니다. CA Access Control 이벤트 로그 채널은 CA-AccessControl-AuthorizationEngine/Audit 으로 명명됩니다.

또한 CA Access Control 이 이벤트 로그 채널로 정책 모델 감사 이벤트를 보내도록 구성할 수도 있습니다. 정책 모델 이벤트 로그 채널은 CA-AccessControl-Policy Models/Audit 으로 명명됩니다.

이벤트 로그 채널로 이벤트를 보내려면

1. 다음 명령을 사용하여 CA Access Control 을 중지합니다.

   ```
   secons -s
   ```

   CA Access Control 이 중지됩니다.

2. logmgr 레지스트리 하위 키의 SendAuditToNativeChannel 토큰 값을 1 로 설정합니다.

   감사 이벤트가 Windows 이벤트 로그 채널로 전달됩니다.

3. (선택 사항) Pmd 레지스트리 하위 키의 SendAuditToNativeChannel 토큰 값을 1 로 설정합니다.

   정책 모델 감사 이벤트가 Windows 이벤트 로그 채널로 전달됩니다.

4. 다음 명령을 사용하여 CA Access Control 을 다시 시작합니다.

   ```
   seosd -start
   ```

   CA Access Control 이 다시 시작됩니다.

예: 이벤트 로그 채널로 감사 이벤트 보내기

다음 예는 이벤트 로그 채널로 감사 이벤트를 보냅니다. 이 명령을 사용하려면 원격 구성 환경(env config)을 사용해야 합니다.

```
er config ACROOT section(logmgr) token(SendAuditToNativeChannel) value(1)
```

예: 이벤트 로그 채널로 정책 모델 감사 이벤트 보내기

다음 예는 이벤트 로그 채널로 정책 모델 감사 이벤트를 보냅니다. 이 명령을 사용하려면 원격 구성 환경(env config)을 사용해야 합니다.

```
er config ACROOT section(Pmd) token(SendAuditToNativeChannel) value(1)
```

감사 로그

감사 로그는 파일에 저장됩니다. 다음 Windows 레지스트리 하위 키의 *audit_log* 값은 감사 로그 파일의 위치를 지정합니다.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\logmgr
```

키의 기본값은 다음과 같습니다.

```
C:\Program Files\CA\AccessControl\\log\seos.audit
```

기본적으로 CA Access Control 은 감사 로그의 크기가 1024KB 에 이르면 감사 로그를 자동으로 백업합니다. 하위 키에서 *audit_size* 값을 변경하여 이 크기를 변경할 수 있습니다.

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\logmgr
```

또한 Windows 레지스트리 하위 키에서 *BackUp_Date* 값을 변경하여 감사 로그를 일별, 주별 또는 월별과 같이 주기적으로 백업하도록 선택할 수 있습니다.

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\logmgr
```

참고: 이러한 레지스트리 하위 키에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

감사 로그 사용

CA Access Control 은 감사 로그를 보고, 필터링하고, 검색하기 위한 다음과 같은 두 개의 내장 도구를 제공합니다.

- CA Access Control 끝점 관리
- seaudit 유틸리티

감사 로그의 모든 레코드를 표시하거나 또는 필터를 사용하여 감사 로그에서 특정 레코드를 선택할 수 있습니다.

이 장의 나머지 부분은 CA Access Control 끝점 관리에서 감사 필터를 사용할 때 감사 로그의 레코드를 보는 방법에 대해 설명합니다.

감사 레코드 필터

audit.cfg 파일은 감사 파일로 전달되면 안 되는 레코드를 정의하여 호스트의 감사 레코드를 필터링합니다. 파일의 각 줄은 감사 정보를 필터링하는 규칙을 나타냅니다. 즉, 줄에 있는 기준과 일치하는 레코드는 감사 파일에 나타나지 않습니다. 이렇게 하면 필요한 레코드만 저장되므로 seos.audit 파일 크기를 제한하는 데 유용합니다. 엔터프라이즈 요구 사항에 맞게 audit.cfg 파일을 편집할 수 있습니다.

기본적으로 audit.cfg 파일은 *ACInstallDir*/etc 디렉터리(UNIX) 또는 *ACInstallDir*\data 디렉터리(Windows)에 있습니다. seos.ini 파일의 [logmgr] AuditFiltersFile 토큰(UNIX) 또는 logmgr 레지스트리 키의 AuditFiltersFile 항목(Windows)을 편집하여 audit.cfg 파일의 위치를 변경할 수 있습니다.

CA Access Control 엔진인 seosd 는 시작할 때 audit.cfg 파일을 읽습니다. 메시지를 감사 파일에 보낼 때 seosd 는 메시지가 audit.cfg 파일에 있는 규칙 중 하나와 일치하는지 여부를 검사합니다. 메시지가 규칙과 일치하면 메시지는 감사 파일에 기록되지 않습니다.

참고: audit.cfg 파일에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

감사 표시 필터

감사 로그의 레코드 수가 엄청나게 많아질 수 있으므로, *표시되는* 레코드 수를 줄이려면 필터를 사용하여 표시할 레코드 유형을 지정하십시오. 시간 또는 이벤트 유형을 포함하는 다양한 기준에 따라 이벤트를 필터링할 수 있습니다.

참고: 또한 CA Access Control 감사 구성 설정(audit.cfg 파일)을 사용하여 감사 파일에 *기록하는* 감사 레코드를 필터링할 수도 있습니다.

이름을 부여하고 하나 이상의 스위치를 선택하여 CA Access Control 끝점 관리에서 간단히 필터를 만들 수 있습니다. 그런 다음 추가 스위치를 선택하고 하나 이상의 옵션을 할당할 수 있습니다. 또한 seaudit 유틸리티를 사용하여 레코드를 필터링할 수 있습니다.

CA Access Control 끝점 관리는 여러 개의 미리 정의된 필터를 제공하며 자체 필터를 작성할 수 있습니다.

필터 마법사, 이름 및 스위치 선택 페이지

필터 마법사의 "이름 및 스위치 선택" 페이지에서는 만들려는 감사 표시 필터의 이름을 정의하고 이 필터에 적용할 스위치를 정의할 수 있습니다.

이 창에는 다음 필드가 있습니다.

필터 이름

만들려는 감사 표시 필터의 이름을 정의합니다.

감사 이벤트 레코드

필터에서 모든 감사 레코드를 표시할지 또는 선택한 스위치만 표시할지 여부를 지정합니다.

모든 레코드를 나열하도록 선택하면 이 페이지의 스위치가 적용되지 않습니다.

호스트 및 서비스의 INET 감사 레코드 목록 나열

지정된 서비스의 지정된 호스트에서 받은 TCP 요청의 INET 감사 레코드를 나열할지 여부를 지정합니다. Host 와 service 는 모두 검색된 호스트 및 서비스 집합을 나타내는 마스크입니다.

터미널의 사용자에 대한 LOGIN 표시

다음을 나열하도록 지정합니다.

- 지정된 터미널의 지정된 사용자에 대한 LOGIN 레코드. *user* 와 *terminal* 은 모두 정의하는 마스크입니다.
- 잘못된 암호를 여러 번 입력했을 때 권한 부여 엔진에서 작성된 레코드

사용자에 대한 리소스에서 클래스의 RESOURCE 감사 나열

리소스 레코드를 나열할지 여부를 지정합니다. 다음은 이후에 정의할 수 있습니다.

- *Class*-액세스된 리소스가 속해 있는 클래스를 나타내는 마스크입니다.
- *Resource*-액세스된 리소스의 이름을 나타내는 마스크입니다.
- *User*-리소스에 액세스한 사용자의 이름을 나타내는 마스크입니다.

데이터베이스 업데이트 목록 나열

데이터베이스 업데이트 감사 레코드를 나열합니다. 다음을 정의할 수 있습니다.

- *Cmd*-검색할 selang 명령을 나타내는 마스크입니다.
- *Class*-검색할 클래스를 나타내는 마스크입니다.
- *Object*-검색할 레코드를 나타내는 마스크입니다.
- *User*-명령을 실행한 사용자를 나타내는 마스크입니다.

시작/종료 메시지 목록 나열

CA Access Control 서비스의 시작 및 종료 메시지를 나열할지 여부를 지정합니다.

WATCHDOG 감사 레코드 나열

Watchdog 감사 레코드를 나열할지 지정합니다.

추적 레코드만 표시

추적 기능을 사용하여 감사 로그로 보낸 레코드만 나열할지 지정합니다.

필터 마법사, 옵션 편집 페이지

필터 마법사의 "옵션 편집" 페이지에서는 감사 표시 필터에 적용할 옵션을 정의할 수 있습니다.

이 창에는 다음 필드가 있습니다.

목록 오늘 시작

오늘을 시작 날짜로 지정합니다. 오늘 이전에 로깅된 레코드는 나열되지 않습니다.

목록 시작 날짜

시작 날짜를 지정합니다. 지정된 날짜 전에 기록된 레코드는 나열되지 않습니다.

목록 시작 시간

시작 시간을 지정합니다. 지정된 시간 전에 기록된 레코드는 나열되지 않습니다.

목록 종료 날짜

종료 날짜를 지정합니다. 지정된 날짜 이후에 기록된 레코드는 나열되지 않습니다.

목록 종료 시간

종료 시간을 지정합니다. 지정된 시간 이후에 기록된 레코드는 나열되지 않습니다.

호스트 이름 대신 인터넷 주소 표시

TCP/IP 레코드에 호스트 이름 대신 인터넷 주소가 나열되도록 지정합니다.

실패 숨기기

실패가 나열되지 않도록 지정합니다.

모든 액세스 허가 숨기기

성공한(허용된) 액세스가 나열되지 않도록 지정합니다.

로그아웃 레코드 숨기기

로그아웃 레코드가 나열되지 않도록 지정합니다.

NOTIFY 감사 레코드 숨기기

NOTIFY 감사 레코드가 나열되지 않도록 지정합니다.

암호 시도 및 작업 숨기기

암호 시도 레코드가 나열되지 않도록 지정합니다.

경고 레코드 숨기기

경고 레코드가 나열되지 않도록 지정합니다.

이름 대신 포트 번호 표시

서비스 이름 대신 포트 번호가 나열되도록 지정합니다.

호스트에서 가져온 레코드만 표시

지정된 호스트에서 발생하는 레코드만 나열되도록 지정합니다. 이 옵션은 UNIX 워크스테이션에 연결된 경우에만 적용됩니다.

미리 정의된 필터

CA Access Control 에는 다음과 같이 미리 정의된 필터가 함께 제공됩니다.

모든 레코드

감사 로그의 모든 레코드를 표시합니다. 필터링이 발생하지 않습니다.

오늘의 레코드

오늘 작성된 모든 레코드를 표시합니다.

마지막 2 일의 레코드

어제와 오늘 작성된 모든 레코드를 표시합니다.

마지막 7 일의 레코드

마지막 7 일 동안 작성된 모든 레코드를 표시합니다.

CA Access Control 서비스 연결

사용자가 CA Access Control 끝점 관리 또는 selang 과 같은 CA Access Control 서비스에 연결할 때 나타나는 레코드를 표시합니다.

참고: UNIX 워크스테이션에 연결할 경우 해당 필터의 이름이 로그인 레코드가 됩니다. 레코드는 사용자 로그인을 나타냅니다.

관리 작업

CA Access Control 또는 운영 체제 데이터베이스를 업데이트하는 모든 레코드를 표시합니다. 데이터베이스 업데이트에는 모든 유형의 레코드에 대한 추가, 삭제 및 변경이 포함됩니다.

사용자 정의 필터 작성

필요한 만큼의 필터를 작성할 수 있습니다. 감사 레코드의 특정 집합만 보고자 할 경우 사용자 정의 필터를 작성하십시오.

사용자 정의 필터를 작성하려면

1. CA Access Control 끝점 관리에서 감사 이벤트 탭을 클릭합니다.

   감사 레코드 뷰어 - 필터 설정 섹션에는 저장된 필터 목록이 표시됩니다.

2. 저장된 필터 섹션에서 "필터 만들기"를 클릭합니다.

   감사 필터 마법사가 나타납니다.

3. 마법사 페이지를 완료합니다.

   이름 및 스위치 선택

   필터에서 사용할 스위치 (페이지 137)를 지정합니다.

   스위치 편집

   선택한 스위치의 설정을 지정합니다. 기본적으로 이러한 설정은 필터링할 감사 이벤트에 대해 정의할 수 있는 마스크입니다.

   옵션 편집

   감사 필터링에 대해 설정할 옵션 (페이지 139)을 지정합니다.

   "마침"을 클릭합니다.

   정의한 새 감사 필터가 저장 및 로드됩니다.

감사 로그 백업

CA Access Control 은 자동으로 감사 로그 파일을 백업하여 보관할 수 있습니다.

감사 로그 백업 파일의 이름은 logmgr\audit_back CA Access Control 레지스트리 항목에 설정됩니다.

다음 방법을 사용하여 감사 로그 파일을 백업할 수 있습니다.

- 크기를 기준으로 트리거되는 백업
- 날짜를 기준으로 트리거되는 백업

감사 로그 파일을 백업하기 위해 선택하는 방법 및 설정의 기준은 다음과 같습니다.

- 로그 파일의 백업 사본이 필요한지 여부
- 현재 환경에서 생성될 감사 데이터의 추정 크기
- 시스템 성능 고려(예를 들어, 감사 로그 파일의 크기가 클수록 처리 시간이 늘어남)

참고: 기본적으로 타임스탬프가 지정된 백업을 유지하도록 설정을 구성하면 CA Access Control 에서 감사 로그 백업 파일을 보호합니다. 이 기능은 크기로 트리거되는 감사 백업 파일에 적용되는 것과 동일한 기본 보호입니다. 이러한 파일을 제거하려면 데이터베이스에 허용 규칙을 설정하면 됩니다.

자동 백업되는 감사 로그의 크기 설정

감사 로그 파일의 최대 크기를 설정할 수 있습니다. 파일이 정의된 크기에 도달하면 CA Access Control 은 자동으로 파일의 백업 사본을 작성하고 로그를 지웁니다. 결과적으로 파일이 주기적으로 자동 백업됩니다.

감사 로그가 자동으로 백업되는 크기를 설정하려면 logmgr\audit_size CA Access Control 레지스트리 항목에 원하는 최대 크기를 KB 단위로 설정하십시오.

참고: logmgr\audit_back CA Access Control 레지스트리 항목을 설정하여 백업 파일 이름을 정의할 수 있습니다.

중요! logmgr/BackUp_Date CA Access Control 레지스트리 항목이 yes(기본값은 'no')로 설정되면 크기를 기준으로 트리거된 감사 로그의 각 백업 사본에 타임스탬프가 접미사로 추가됩니다. 날짜로 트리거되는 백업이 구성된 경우를 포함한 다른 모든 경우에는 각 백업 사본이 이전에 기록된 백업 사본을 *덮어씁니다.*

예: 감사 로그 파일의 크기가 5 MB 에 도달하면 자동 백업하도록 설정

이 예제는 감사 로그 파일의 크기가 5 MB(5120 KB)에 도달하면 이 파일이 백업되도록 설정하는 방법을 보여줍니다. 이렇게 하려면 logmgr\audit_size CA Access Control 레지스트리 항목을 **5120** 으로 설정하십시오.

감사 로그 파일의 크기가 5 MB 에 도달하면 CA Access Control 은 기본적으로 seos.audit.bak 이란 이름으로 이 파일의 백업 사본을 만든 다음 로그를 삭제합니다.

예: 감사 로그 파일의 크기가 1 MB 에 도달하면 사용자 지정된 이름과 타임스탬프를 사용하여 자동 백업하도록 설정

이 예제는 감사 로그 파일의 크기가 1 MB(1024 KB)에 도달하면 사용자 지정된 이름을 사용하고 타임스탬프를 이 이름에 추가하여 감사 로그 파일이 백업되도록 설정하는 방법을 보여줍니다.

이렇게 하려면 다음과 같이 CA Access Control 레지스트리 항목을 설정하십시오.

- logmgr\audit_size=1024
- logmgr\audit_back=log\ac_audit.old
- logmgr\BackUp_Date=yes

감사 로그 파일의 크기가 1 MB 에 도달하면 CA Access Control 이 파일의 백업 사본을 만든 다음 로그를 삭제합니다. 백업 로그 파일의 이름은 ac_audit.old.*timestamp* 와 같습니다. 여기서 *timestamp* 는 DD-Mon-YYYY.hhmmss 형식의 날짜 및 시간입니다. 예:

```
ac_audit.old.06-Feb-2007.144330
```

감사 로그가 자동으로 백업되는 시간 주기 설정

CA Access Control 이 감사 로그 파일의 백업 사본을 자동으로 생성하고 로그를 지우는 시간 주기(매일, 매주, 매월)를 정의할 수 있습니다.

감사 로그가 자동으로 백업되는 시간 주기를 설정하려면 logmgr\BackUp_Date CA Access Control 레지스트리 항목에 이 주기를 설정하십시오. 주기는 다음 중 하나가 될 수 있습니다.

매일

감사 로그 파일을 하루 한 번 백업합니다.

매주

감사 로그 파일을 일주일에 한 번 백업합니다.

매월

감사 로그 파일을 한 달에 한 번 백업합니다.

참고: logmgr\audit_back CA Access Control 레지스트리 항목을 설정하여 백업 파일 이름을 정의할 수 있습니다.

중요! 백업 주기에 도달하기 전에 감사 로그의 크기가 logmgr\audit_size CA Access Control 레지스트리 항목에 정의된 제한 크기에 도달하면 CA Access Control 은 타임스탬프 없이 백업 사본을 작성합니다. 이러한 각 백업 사본은 잠재적으로 이전의 사본을 덮어쓸 수 있습니다.

예: 감사 로그 파일을 매일 백업하도록 설정

이 예제는 감사 로그 파일이 매일 백업되도록 설정하는 방법을 보여줍니다. 이렇게 하려면 logmgr\BackUp_Date CA Access Control 레지스트리를 **daily** 로 설정하십시오.

하루에 한 번, CA Access Control 은 파일의 백업 사본을 만든 다음 로그를 지웁니다. 백업 로그 파일의 이름은 *.timestamp* 접미사를 갖습니다. 여기서 *timestamp* 는 DD-Mon-YYYY.hhmmss 형식의 날짜 및 시간입니다. 예:

```
seos.audit.bak.06-Feb-2007.144330
```

제 9 장: 관리 인증 범위

이 섹션은 다음 항목을 포함하고 있습니다.

전역 권한 부여 특성

전역 권한 부여 특성은 사용자 레코드에 설정됩니다. 각 전역 권한 부여 특성을 사용하여 사용자가 특정 유형의 기능을 수행할 수 있습니다. 이 단원에서는 각 전역 권한 부여 특성의 기능과 제한에 대해 설명합니다.

ADMIN 특성

ADMIN 특성을 사용하면 CA Access Control 에서 대부분의 명령을 실행할 수 있습니다. ADMIN 특성을 가진 데이터베이스에 정의된 사용자는 데이터베이스에서 사용자, 그룹 및 리소스를 정의하고 업데이트할 수 있습니다. 이 특성은 CA Access Control 에서 가장 강력한 특성이지만 다음과 같은 제한이 있습니다.

- 데이터베이스에서 한 명의 사용자만 ADMIN 특성을 가지는 경우 해당 사용자를 삭제할 수 없고 ADMIN 특성을 레코드에서 제거할 수 없습니다.
- ADMIN 특성은 있지만 AUDITOR 특성이 없는 사용자는 사용자, 그룹 또는 리소스에서 실행된 감사 유형(감사 모드)을 변경할 수 없습니다. ADMIN 특성이 있으며 사용자, 그룹 또는 리소스의 감사 특성을 변경해야 하는 경우 사용자 자신에게 AUDITOR 특성을 할당합니다.
- ADMIN 특성이 있는 사용자는 슈퍼 사용자(UNIX 의 root 계정 또는 Windows 의 Administrator 계정)를 삭제할 수 없지만 root 를 ADMIN 이 아닌 사용자로 설정할 수 있습니다.

AUDITOR 특성

AUDITOR 특성을 가진 사용자는 시스템 사용을 모니터할 수 있습니다. AUDITOR 특성을 가진 사용자의 명시된 권한은 다음과 같습니다.

- 사용자는 데이터베이스에 정보를 표시할 수 있습니다.

  감사자는 selang 명령 *showusr*, *showgrp*, *showres* 및 *showfile* 을 실행할 수 있습니다.

- 사용자는 기존 레코드에 대한 감사 모드를 설정할 수 있습니다.

  감사자는 selang 명령 *chusr*, *chgrp*, *chres* 및 *chfile* 을 실행할 수 있습니다.

OPERATOR 특성

OPERATOR 특성을 가진 사용자는 모든 파일에 대한 READ 권한이 있습니다. 이 권한을 사용하면 데이터베이스의 모든 내용을 나열할 수 있을 뿐만 아니라 백업 작업을 실행할 수 있습니다. 운영자는 showusr, showgrp, showres, showfile, find 명령을 사용하여 데이터베이스 레코드를 나열할 수 있습니다. OPERATOR 특성은 사용자가 secons 유틸리티를 사용하게 합니다.

참고: secons 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

PWMANAGER 특성

PWMANAGER 특성은 일반 사용자에게 chusr 또는 sepass 명령을 사용하여 다른 사용자의 암호를 변경할 수 있는 권한을 제공합니다.

참고: PWMANAGER 가 ADMIN 사용자의 암호를 변경하게 하려면 setoptions 명령의 *cng_adminpwd* 옵션을 설정합니다. 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

PWMANAGER 권한에는 showusr 및 find 명령의 사용도 포함됩니다.

PWMANAGER 의 권한에는 showusr 및 find 명령의 사용이 포함됩니다.

참고: 사용자의 nochngpass 속성이 yes 로 설정된 경우 PWMANAGER 는 해당 사용자의 암호를 변경할 수 없습니다.

SERVER 특성

많은 다른 보안 모델과 같이 CA Access Control 은 일반 사용자에게 "사용자 A 가 리소스 X 에 액세스할 수 있습니까?"와 같은 질문을 허용하지 않으며 유일하게 허용되는 질문은 "리소스 X 에 내가 액세스할 수 있습니까?"입니다. 그러나 데이터베이스 서버 서비스나 내부 응용 프로그램과 같이 많은 사용자에게 서비스를 제공하는 프로세스는 다른 사용자를 위한 권한 부여를 요청할 수 있게 허용되어야 합니다.

SERVER 특성을 사용하면 프로세스에서 사용자에 대한 권한 부여를 요청할 수 있습니다. SERVER 특성을 가진 사용자는 SEOSROUTE_VerifyCreate API 를 실행할 수 있습니다.

참고: 서버 특성 및 CA Access Control API 에 대한 자세한 내용은 *SDK 안내서*를 참조하십시오.

IGN_HOL 특성

IGN_HOL 특성을 통해 사용자는 holiday 레코드에 정의된 기간에는 언제든지 로그인할 수 있습니다. HOLIDAY 클래스의 각 레코드는 사용자가 로그인하기 위해 추가 권한이 필요할 때 하나 이상의 기간을 정의합니다. IGN_HOL 특성을 사용하면 holiday 레코드에 정의된 기간과 관계없이 사용자가 언제든지 로그인할 수 있습니다.

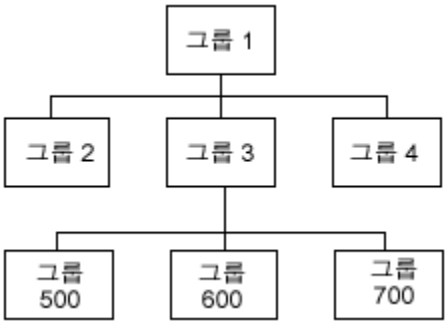
참고: HOLIDAY 클래스에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

그룹 권한 부여

그룹 권한 부여 특성을 설명하기 전에 부모-자식 관계의 개념을 이해할 필요가 있습니다.

부모-자식 관계

부모-자식 관계라고도 하는 종속 및 부모 그룹의 개념은 그룹 관리 권한을 설명할 때 중요합니다. 하나의 그룹은 하나 이상의 그룹의 부모(상위 그룹)가 될 수 있습니다. *child* 또는 종속 그룹은 하나의 parent 만을 포함할 수 있습니다. 그룹에 부모를 할당하는 것은 선택 사항입니다. 다음 그림을 참조하십시오.



그룹 1 은 세 그룹(20, 30, 40)의 부모 그룹입니다. 그룹 30 은 세 그룹(500, 600, 700)의 부모 그룹입니다. 그룹 600 은 그룹 30 의 유일한 부모 그룹입니다. 그룹 1 에는 부모 그룹이 없습니다.

그룹 권한 부여 특성

리소스 레코드와 접근자 레코드 등을 포함한 모든 레코드에는 소유자가 있습니다. 레코드를 소유하는 것은 레코드를 표시, 편집 및 제거할 권한을 갖는 것입니다.

그룹은 고유의 레코드를 소유할 수 있습니다. 하지만 레코드를 소유하는 그룹에서 특정 권한을 부여 받은 사용자만 레코드를 관리할 수 있습니다. 이러한 특별한 사용자들은 자신의 사용자 레코드에 그룹 권한 부여 특성 세트를 갖습니다. 그룹 권한 부여 특성은 다음과 같습니다.

- GROUP-ADMIN
- GROUP-AUDITOR
- GROUP-OPERATOR
- GROUP-PWMANAGER

권한을 부여받은 사용자만이 실행할 수 있는 join 명령으로 이 특성을 설정합니다. join 명령은 사용자를 그룹에 할당하고 사용자의 그룹 권한 부여 특성(있는 경우)을 지정합니다.

GROUP-ADMIN 특성

추가 정보:

소유권 (페이지 151)

GROUP-ADMIN 특성

그룹 관리 권한 부여 특성을 가진 사용자는 레코드 세트를 생성할 수 있습니다. 레코드를 생성하려면 그룹 관리자가 레코드의 소유자를 지정해야 합니다.

사용자가 그룹 권한 부여 특성을 갖고 있는 그룹이 레코드의 소유자가 되어야 합니다. 이 그룹이 다른 그룹의 부모이면 소유자도 하위 그룹 중 하나에 속할 수 있습니다. 전체 레코드 세트를 그룹 범위라고 합니다. 제공된 권한 부여 예에서는 그룹 범위 개념을 설명합니다.

GROUP-ADMIN 특성을 가진 사용자는 자신의 그룹 범위 내의 레코드에 대해 다음과 같은 액세스 권한을 가집니다.

액세스	설명	명령
읽기	레코드 속성을 표시합니다.	showusr, showgrp, showres, showfile
작성	데이터베이스에 새 레코드를 생성합니다. 소유자를 지정해야 합니다.	newusr, newgrp, newres, newfile
수정	레코드 속성을 변경합니다.	chusr, chgrp, chres, chfile
삭제	데이터베이스에서 레코드를 제거합니다.	rmusr, rmgrp, rmres, rmfile
연결	사용자를 그룹에 조인하거나 그룹에서 사용자를 분리합니다.	join, join-

GROUP-ADMIN 특성은 또한 다음과 같은 제한이 있습니다.

- GROUP-ADMIN 사용자는 리소스가 자신에게 액세스할 수 없도록 만들 수 없습니다. 따라서,
  - GROUP-ADMIN 사용자는 자신의 보안 수준보다 높은 보안 수준을 할당할 수 없습니다.
  - GROUP-ADMIN 사용자는 자신이 소유하지 않은 보안 범주나 보안 레이블을 할당할 수 없습니다.
- GROUP-ADMIN 사용자는 데이터베이스에서 슈퍼 사용자 사용자(UNIX 의 root 계정 또는 Windows 의 Administrator 계정)를 삭제할 수 없습니다.
- 일부 제한 사항이 이 장의 전역 권한 부여 특성에서 설명된 전역 권한 부여 특성에 적용됩니다.
  - GROUP-ADMIN 사용자는 데이터베이스에서 ADMIN 사용자 레코드만 삭제할 수 없습니다.
  - GROUP-ADMIN 사용자는 데이터베이스에 있는 마지막 ADMIN 사용자의 레코드에서 ADMIN 속성을 제거할 수 없습니다.
  - AUDITOR 속성이 없는 GROUP-ADMIN 사용자는 감사 모드를 업데이트할 수 없습니다. AUDITOR 속성이 있는 GROUP-ADMIN 사용자만 감사 모드를 업데이트할 수 있습니다.
  - GROUP-ADMIN 사용자는 모든 사용자에 대해 ADMIN, AUDITOR, OPERATOR, PWMANAGER 및 SERVER 의 전역 권한 부여 특성을 설정할 수 없습니다.

GROUP-AUDITOR 특성

GROUP-AUDITOR 속성을 가진 사용자는 그룹 범위 내의 모든 레코드에 대한 속성을 나열할 수 있습니다. 또한 그룹 감사자는 그룹 범위에 속해 있는 모든 레코드에 대해 감사 모드를 설정할 수 있습니다.

GROUP-OPERATOR 특성

GROUP-OPERATOR 속성을 가진 사용자는 그룹 범위 내의 모든 레코드에 대한 속성을 나열할 수 있습니다.

GROUP-PWMANAGER 특성

GROUP-PWMANAGER 특성을 가진 사용자는 그룹 범위에 속해 있는 레코드를 소유한 사용자의 암호를 변경할 수 있습니다.

소유권

데이터베이스의 모든 레코드(접근자 레코드 및 리소스 레코드 포함)에는 소유자가 있습니다. 데이터베이스에 레코드를 추가할 때 소유자 매개 변수를 사용하여 명시적으로 소유자를 할당할 수도 있고 CA Access Control 가 레코드 소유자로서 레코드를 정의하는 사용자를 할당하도록 할 수도 있습니다.

다음 중 해당 사항이 *하나라도* 있는 경우 접근자가 레코드를 소유합니다.

- 접근자가 레코드의 소유자로 정의되었습니다.
- 접근자가 레코드의 소유자로 정의된 그룹의 구성원이고 GROUP-ADMIN 속성이 있는 그룹에 속합니다.
- 접근자가 해당 리소스가 구성원인 리소스 그룹 레코드의 소유자입니다.

데이터베이스에서 레코드를 소유하는 사용자나 그룹을 제거하면 레코드는 더 이상 소유자를 갖지 않습니다.

레코드를 소유하는 사용자는 자신들이 소유한 레코드에 대해 다음과 같은 액세스 권한을 가집니다.

액세스	설명	명령
읽기	레코드 속성을 표시합니다.	showusr, showgrp, showres, showfile
수정	레코드 속성을 변경합니다.	chusr, chgrp, chres, chfile
삭제	데이터베이스에서 레코드를 제거합니다.	rmusr, rmgrp, rmres, rmfile
연결	사용자를 그룹에 조인하거나 그룹에서 사용자를 분리합니다.	join, join-

사용자 또는 그룹이 특정 레코드에 대한 소유 권한을 갖도록 하고 싶지 않으면 해당 레코드 및 해당 레코드가 구성원으로 속한 모든 리소스 그룹 레코드에 소유자 *nobody* 를 할당하십시오.

소유권의 제한 사항은 다음과 같습니다.

- 데이터베이스에서 마지막 ADMIN 사용자의 소유자는 해당 사용자 레코드를 삭제할 수 없습니다.
- AUDITOR 특성이 없는 소유자는 감사(Audit) 모드를 업데이트할 수 없습니다. AUDITOR 특성이 있는 소유자만 감사(Audit) 모드를 업데이트할 수 있습니다.
- 슈퍼 사용자(UNIX 의 root 계정 또는 Windows 의 Administrator 계정)의 소유자는 데이터베이스에서 root 를 삭제할 수 없습니다.
- 소유자는 액세스할 수 없는 리소스를 만들 수 없습니다. 따라서,
- 소유자는 액세스할 수 없는 리소스를 만들 수 없습니다. 따라서,
  - 소유자는 자신의 보안 수준보다 높은 보안 수준을 할당할 수 없습니다.
  - 소유자는 자신이 갖고 있지 않은 보안 범주나 보안 레이블을 할당할 수 없습니다.

파일 소유권

CA Access Control 은 파일 소유자가 FILE 클래스에서 레코드를 정의하여 파일을 보호하게 합니다. 파일의 소유자는 해당 파일의 레코드에 대해 모든 권한을 가지기 때문에 newfile, chfile, showfile, authorize 및 authorize- 명령을 해당 레코드에 대한 모든 매개 변수와 함께 사용하여 파일을 보호할 수 있습니다.

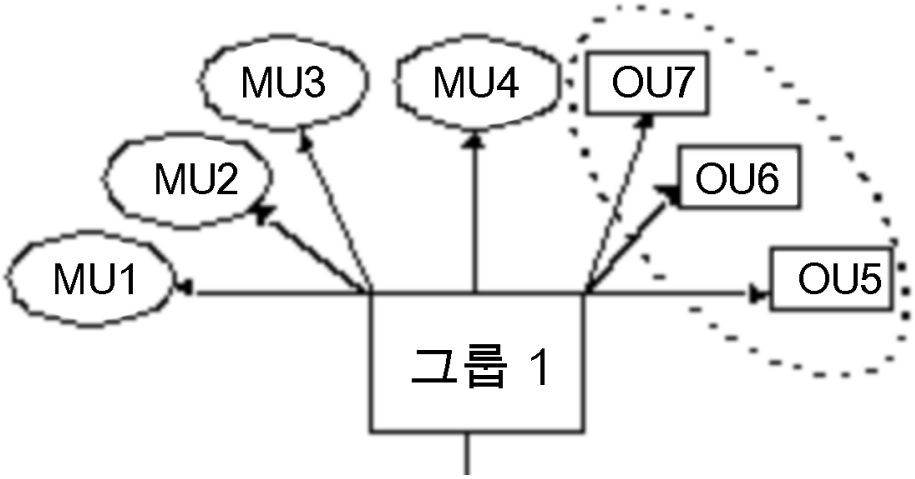
UNIX 의 경우 사용자가 파일을 생성하면 UNIX 는 그 사용자를 파일 소유자로 할당합니다. CA Access Control 에서는 이 기능이 명시적으로 비활성화되지 않는 한 UNIX 파일 소유자가 FILE 레코드를 정의할 수 있습니다. 파일 소유자가 파일(FILE) 레코드를 정의하지 못하도록 하려면 seos.ini 파일에서 [seos] 섹션의 use_unix_file_owner 토큰을 no 로 설정해야 합니다(기본 설정).

권한 부여 예제

다음은 그룹 권한 부여 특성, 상위 관계, 소유권, 구성원, 그룹 범위 등의 개념을 설명하는 다이어그램입니다. 이러한 다이어그램에서 특정 사용자와 그룹만 설명하지만 소유권의 개념은 리소스 및 파일 레코드에도 적용됩니다.

단일 그룹 권한 부여

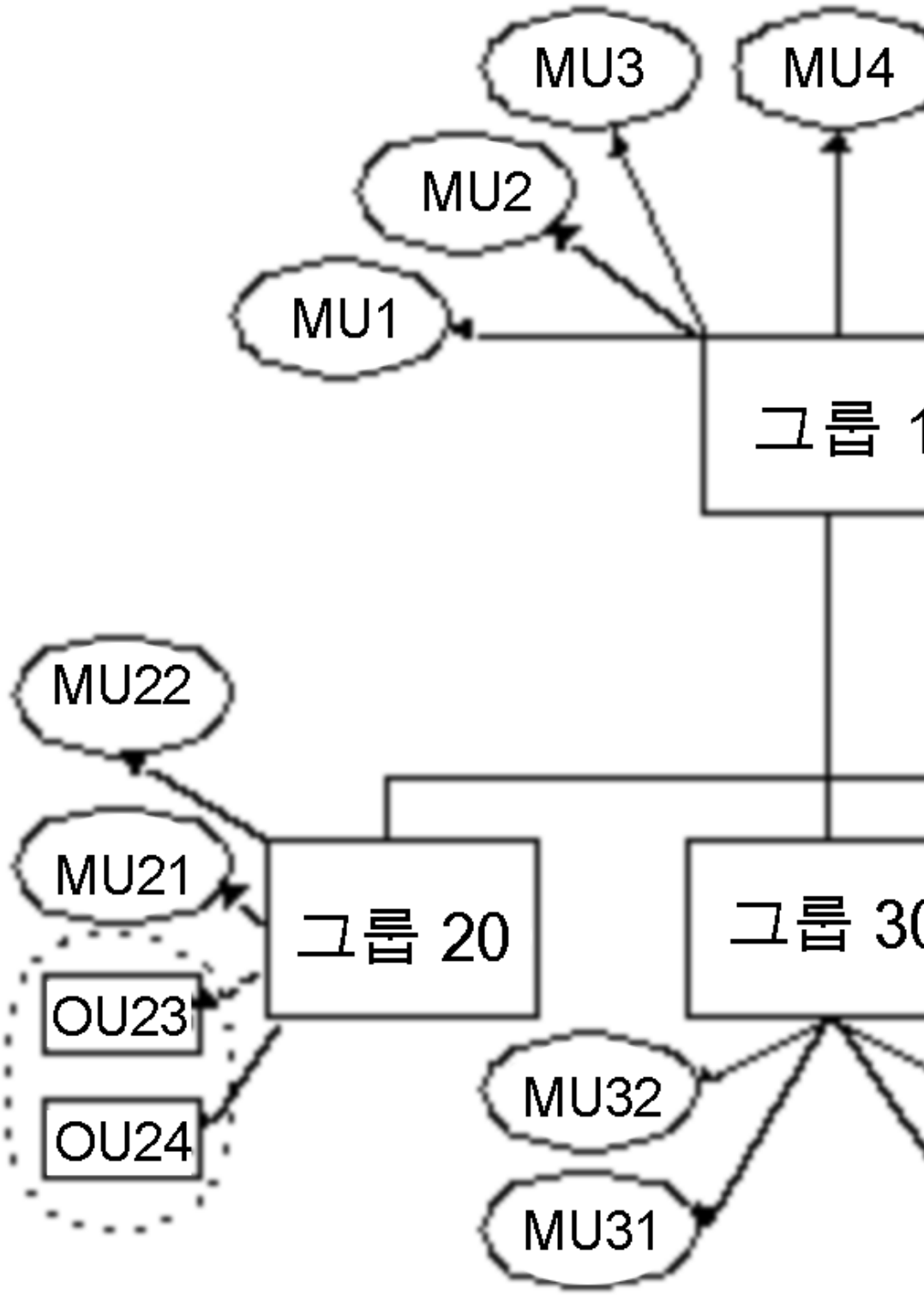
다음 다이어그램에서 네 사용자(MU1, MU2, MU3, MU4)는 그룹 1 의 구성원입니다. 또한 그룹 1 은 세 사용자(OU5, OU6, OU7)를 소유합니다. 구성원 MU4 는 GROUP-ADMIN 특성을 가집니다.



타원은 사용자 MU4 가 실행한 명령의 그룹 범위를 나타냅니다. 또한 그룹 1 이 소유한 모든 사용자(OU5, OU6, OU7)를 포함합니다.

부모 및 자식 그룹

다음 다이어그램에서 네 사용자(MU1, MU2, MU3, MU4)는 그룹 1 의 구성원입니다. 또한 그룹 1 은 세 사용자(OU5, OU6, OU7)를 소유합니다. 구성원 MU4 는 레코드에 설정된 GROUP-ADMIN 특성을 가집니다.


MU3
MU4
MU2
MU1
그룹 1
MU22
MU21
그룹 20
그룹 30
OU23
OU24
MU32
MU31

그룹 1 은 세 그룹(20, 30, 40)의 부모 그룹입니다. 각 하위 그룹에는 그룹의 구성원인 두 명의 사용자와 그룹에서 소유한 두 명의 사용자가 있습니다.

네 개의 타원은 사용자 MU4 가 실행하는 명령의 그룹 범위를 나타냅니다. 또한 그룹 1 이 소유한 모든 사용자와 그룹 1 의 하위 그룹에서 소유한 사용자를 포함합니다. MU4 의 그룹 범위에 속하는 사용자는 OU5, OU6, OU7, OU23, OU24, OU33, OU34, OU43 및 OU44 입니다.

사용자, 그룹 또는 리소스를 소유한 그룹 20, 30 또는 40 에 하위 그룹이 있는 경우 이러한 그룹이 소유한 레코드도 사용자 MU4 가 실행하는 명령 그룹 범위에 속합니다.

하위 관리

보안 관리자(ADMIN 특성을 가진 사용자)는 일반 사용자에게 특정 관리 권한을 부여할 수 있습니다. 이러한 일반 사용자를 하위 관리자라고 부릅니다. 하위 관리자는 지정된 CA Access Control 클래스 또는 개체만 관리할 수 있는 권한을 가지고 있습니다. 예를 들어, 하위 관리자는 사용자와 그룹 개체만 관리할 수 있는 권한을 부여받을 수 있습니다. 하위 관리자에게 클래스의 특정 개체에 대한 관리 권한을 부여하여 보다 높은 수준의 하위 관리를 설정할 수 있습니다.

사용자, 그룹 및 리소스의 하위 관리자는 selang 을 사용하여 이러한 리소스와 관련된 관리 작업을 수행할 수 있습니다.

일반 사용자에게 특정 관리 권한을 부여하는 방법

ADMIN 특성을 가진 관리자 사용자는 CA Access Control 에서 거의 모든 작업을 실행할 수 있으므로 특정 관리 작업을 하위 관리자에게 위임할 수 있습니다. 이렇게 하려면 다음과 같이 CA Access Control 데이터베이스에서 사용자가 수행해야 하는 특정 관리 작업을 제어하는 클래스에 대한 권한을 사용자에게 부여해야 합니다.

1. 위임할 작업을 제어하는 하나 이상의 클래스를 식별합니다.

   예를 들어 CA Access Control 에서는 USER 및 GROUP 클래스를 사용하여 접근자 리소스를 생성합니다. 접근자 관리를 위임하려면 ADMIN 클래스의 USER 및 GROUP 레코드를 사용해야 합니다.

2. 한 명 이상의 하위 관리자에게 ADMIN 클래스의 적용 가능한 리소스에 대한 권한을 부여합니다.

   예를 들어 하위 관리자가 사용자 레코드를 표시 및 수정하게 하려면 사용자에게 ADMIN 클래스의 USER 레코드에 대한 *읽기* 및 *수정* 액세스 권한을 부여합니다.

ADMIN 클래스

ADMIN 클래스에서 레코드의 ACL(접근자 제어 목록)에 나열된 사용자인 하위 관리자는 ADMIN 특성을 가진 사용자와 유사한 권한을 갖습니다. 그러나 클래스에 의한 액세스(ADMIN) 클래스의 레코드에 대한 액세스 제어 목록(ACL)에 있는 사용자 권한은 레코드가 표시하는 특정 클래스로 제한됩니다. 예를 들어 클래스에 의한 액세스 클래스의 사용자 ID 교체(SURROGATE) 레코드는 사용자 ID 교체(SURROGATE) 클래스의 레코드를 관리할 수 있는 사용자를 결정합니다.

참고: CA Access Control 클래스에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

ACL 에 있는 사용자는 ADMIN 클래스의 특정 레코드에 대해 다음 명령을 실행할 수 있습니다.

액세스	설명	명령
읽기	클래스에서 레코드 속성을 표시합니다.	showusr, showgrp, showres, showfile, find

액세스	설명	명령
작성	클래스에서 새 데이터베이스 레코드를 작성합니다.	newusr, newgrp, newres, newfile
수정	클래스에서 속성을 변경합니다.	chusr, chgrp, chres, chfile
삭제	데이터베이스에서 기존 클래스 레코드를 제거합니다.	rmusr, rmgrp, rmres, rmfile
연결	그룹에서 사용자를 추가 및 제거합니다. 이 액세스는 GROUP 레코드의 ACL 에서만 유효합니다.	join, join-
암호	데이터베이스 내부의 모든 사용자 암호와 암호 특성을 제어합니다. 이 액세스는 PWMANAGER 특성이 있는 사용자에게 허용되는 액세스와 동일한 권한을 부여합니다. 이 액세스는 USER 레코드의 ACL 에서만 유효합니다.	chusr

ADMIN 클래스 권한이 있는 사용자에게는 다음과 같은 제한 사항이 있습니다.

- ADMIN 클래스에 있는 USER 레코드의 ACL 에 정의된 사용자는 데이터베이스의 마지막 ADMIN 사용자를 삭제할 수 없습니다.
- ADMIN 클래스 사용자는 자신이 소유한 사용자에 대한 전역 권한 특성인 ADMIN, AUDITOR, OPERATOR 및 PWMANAGER 을 설정할 수 없습니다.
- 모든 ADMIN 클래스 사용자가 감사 모드를 업데이트할 수 있는 것은 아닙니다. AUDITOR 특성을 가진 ADMIN 클래스 사용자만 감사 모드를 업데이트할 수 있습니다.

- ADMIN 클래스 사용자는 슈퍼 사용자(UNIX 의 root 계정 또는 Windows 의 Administrator 계정)를 삭제할 수 없지만 root 를 NOADMIN 으로 설정할 수 있습니다.
- ADMIN 클래스 사용자는 액세스할 수 없는 리소스를 만들 수 없습니다. 따라서,
  - ADMIN 클래스 사용자는 자신의 보안 수준보다 높은 보안 수준을 할당할 수 없습니다.
  - ADMIN 클래스 사용자는 자신이 갖고 있지 않은 보안 범주나 보안 레이블을 할당할 수 없습니다.

이러한 제한 사항은 B1 보안 수준 인증의 일부입니다.

환경 고려 사항

데이터베이스에서 정보를 업데이트할 수 있는지 여부를 결정하는 요인들 중 하나는 사용자의 직위입니다.

원격 관리 제한 사항

네트워크를 통해 원격 스테이션에 액세스하고 원격 스테이션에서 데이터베이스를 업데이트할 수 있습니다. 원격 스테이션 상의 데이터베이스를 업데이트하려면 사용자와 사용자의 터미널 모두에게 권한이 있어야 합니다.

- 사용자는 원격 스테이션의 데이터베이스에서 명시적으로 사용자로 정의되어야 합니다. 실행하려는 명령에 관계없이 원격 스테이션의 데이터베이스에 있는 사용자 레코드에 적절한 특성을 설정해야 합니다.
- 원격 스테이션에 액세스할 수 있는 쓰기 권한을 부여하는 규칙에 로컬 터미널의 요구 내용을 명시적으로 언급해야 합니다. 그렇지 않으면 여기에서 CA Access Control 관리 작업을 수행할 수 없습니다.

  기본 액세스 필드인 _default 또는 UACC 클래스를 통해 WRITE 권한을 사용하면 원격 스테이션에서 selang 명령 셸을 입력할 수 있습니다. 그러나 selang 명령을 실행하거나 원격 데이터베이스에 액세스할 수는 *없습니다*. 읽기 권한이 있으면 원격 스테이션에 로그인할 수는 있어도 CA Access Control 관리 작업을 수행할 수는 없습니다.

  다음은 WRITE 권한과 READ 권한의 차이를 설명하는 예입니다.

  1. 새 터미널의 기본 액세스 권한을 READ 로 지정하여 관리자가 터미널에서 로그인할 수는 있지만, 데이터베이스를 조작하지는 못하도록 하려면 다음 명령을 실행합니다.

     ```
     newres TERMINAL tty13 defacc(read)
     ```

  2. 사용자 ADMIN1 에게 새 터미널에서 데이터베이스를 조작할 수 있는 권한을 부여하려면 다음 명령을 실행합니다.

     ```
     authorize TERMINAL tty13 uid(ADMIN1) access(r,w)
     ```

UNIX 환경

UNIX 에서 사용자 및 그룹을 관리하는 경우 전역 또는 그룹 권한 부여 특성을 가진 CA Access Control 사용자는 CA Access Control 에서와 동일한 권한과 제한을 UNIX 에 대해 가집니다.

seosd 데몬이 실행되지 *않을* 경우(예: 설치 시) selang 을 사용하려면 다음 규칙을 따라야 합니다.

- selang 명령에 -l 옵션을 포함시켜야 합니다.
- selang 사용자는 root 여야 합니다. 이 고유 root 권한에는 일반적인 UNIX 제한 사항이 적용됩니다.

Windows 환경

기본 Windows 환경에 해당

CA Access Control 이 실행될 때 selang 을 사용하여 네이티브 Windows 환경에서 리소스를 변경하면 CA Access Control 에이전트는 적절한 Windows 리포지토리에서 이 리소스를 변경합니다. 리소스를 변경하기 위한 추가적인 Windows 권한은 필요 없습니다. 즉, 전역 또는 그룹 권한 부여 특성이 있는 CA Access Control 의 사용자가 네이티브 Windows 환경에서 selang 명령을 수행하면 이 사용자는 Windows 에 대해 CA Access Control 과 동일한 권한 및 제한이 부여됩니다.

CA Access Control 이 실행 중이지 않을 때 네이티브 Windows 환경에서 리소스를 변경하기 위해 selang 을 사용하는 경우에는 다음 규칙을 반드시 따라야 합니다.

- selang 명령에 -l 옵션을 포함시켜야 합니다.
- ADMIN 특성 또는 관리자에 준하는 권한이 있어야 합니다.
- 리소스를 변경하려면 충분한 Windows 권한이 있어야 합니다.

  이 제한 사항은 CA Access Control 에이전트가 아닌 selang 프로세스가 Windows 리포지토리에서 리소스를 변경하기 때문입니다.

예를 들어, 사용자 Emma 는 C:\tmp.txt 파일의 소유자를 변경하기 위해 네이티브 Windows 환경에서 selang 명령을 사용하려고 합니다. CA Access Control 이 실행 중인 경우, Emma 는 파일 소유자를 변경하기 위해 충분한 CA Access Control 권한이 필요하지만 추가 Windows 권한은 필요 없습니다. CA Access Control 이 실행 중이지 않으면 Emma 는 파일 소유자를 변경하기 위해 CA Access Control 과 Windows 의 권한이 모두 필요합니다.

데이터베이스에 액세스하기 위한 기본 권한

CA Access Control 은 실행될 때 내부 파일 규칙을 사용하여 내부 데이터베이스인 seosdb 를 보호합니다. 내부 파일 규칙은 selang 에서 보이지 않으며 감지되지 않습니다. FILE 규칙을 작성하여 내부 파일 규칙을 무시할 수 있습니다. 이러한 FILE 규칙을 삭제하면 CA Access Control 이 내부 파일 규칙으로 되돌립니다.

다음 내부 파일은 규칙은 CA Access Control 이 실행될 때 데이터베이스를 보호합니다.

- CA Access Control 내부 프로세스는 데이터베이스에 대한 모든 액세스 권한을 갖습니다.
- NT AUTHORITY\System 사용자는 데이터베이스에 대해 읽기 액세스 권한을 갖습니다.
- 모든 다른 접근자는 데이터베이스에 대한 액세스 권한이 없습니다.

참고: r12.5 SP3 에서 모든 다른 접근자에 대한 기본 액세스 권한이 변경되었습니다. 이전 릴리스에서 모든 다른 접근자는 기본적으로 데이터베이스 파일에 대해 읽기 액세스 권한이 있었습니다.

기본적으로 CA Access Control 서비스는 CA Access Control 을 설치하거나 끝점을 다시 시작한 이후에 자동으로 실행됩니다. 결과적으로, 기본적으로 데이터베이스에 액세스할 수 있는 유일한 사용자는 NT AUTHORITY\System 이 됩니다. 설치 중 정의하는 CA Access Control 관리자 또한 selang 과 같은 유틸리티를 사용하여 데이터베이스를 업데이트할 수 있습니다.

데이터베이스에 액세스하기 위한 네이티브 권한

CA Access Control 이 중지되면 데이터베이스 파일에 대한 액세스 권한은 네이티브 Windows 권한에 의해 결정됩니다. 권한은 CA Access Control 이 설치된 부모 디렉터리에서 상속됩니다. 이 상속으로 인해 CA Access Control 이 중지되면 데이터베이스에 대한 기본 액세스 권한이 '읽기' 권한이 됩니다.

중지되었을 때 CA Access Control 을 보호하기 위해 데이터베이스에 대한 Windows 권한을 회사의 요구 사항에 맞게 변경할 수 있습니다. 권한을 변경하기 전에 다음을 고려하십시오.

- NT AUTHORITY\System 사용자는 *반드시* 데이터베이스 파일을 읽고 쓸 수 있는 Windows 권한이 있어야 합니다.

  CA Access Control 인증 엔진은 NT AUTHORITY\System 사용자로부터 사용 권한을 상속합니다. 이 사용자가 데이터베이스에 액세스할 수 없으면 엔진은 데이터베이스를 업데이트하기 위한 충분한 네이티브 권한이 없습니다.

- 중지되었을 때 CA Access Control 에 대한 읽기 및 쓰기 액세스 권한이 필요한 사용자는 반드시 이 데이터베이스 파일을 읽고 쓸 수 있는 Windows 권한이 필요합니다.

  읽고 쓸 수 있는 액세스 권한이 필요한 사용자는 CA Access Control 을 백업, 복원, 업그레이드하는 사용자가 포함됩니다.

- CA Access Control 이 중지되었을 때 selang(selang -l 옵션)을 사용할 수 있는 사용자는 다음 권한이 있어야 합니다.
  - ADMIN 특성 또는 관리자에 준하는 권한
  - 데이터베이스 파일을 읽고 쓸 수 있는 Windows 권한
  - 필요한 경우 네이티브 리포지토리를 변경할 수 있는 Windows 권한

    예를 들어, CA Access Control 이 중지되었을 때 CA Access Control 레지스트리 항목을 변경하기 위해 config 환경을 사용하려면 레지스트리를 변경하기 위한 충분한 Windows 권한이 있어야 합니다.

  CA Access Control 관리자(ADMIN 특성 또는 관리자에 준하는 권한이 있는 사용자)만 CA Access Control 이 중지되었을 때 selang 을 사용하여 데이터베이스를 관리할 수 있습니다. CA Access Control 이 중지되었을 때 CA Access Control 관리자가 데이터베이스에 액세스할 수 없으면 어떤 사용자도 오프라인 데이터베이스 관리를 수행할 수 없으며 교착 상태가 발생할 수 있습니다.

제 10 장: 정책 모델 관리

이 섹션은 다음 항목을 포함하고 있습니다.

정책 모델 데이터베이스

수십 또는 수백 개의 데이터베이스를 개별적으로 관리하는 것은 실용적이지 않습니다. CA Access Control 은 하나의 중앙 데이터베이스에서 여러 데이터베이스를 관리할 수 있는 구성 요소인 정책 모델 서비스를 제공합니다. 정책 모델(PMD) 서비스를 사용하는 것은 선택 사항이지만 큰 사이트에서 이 서비스를 사용하면 관리 작업이 상당히 간단합니다.

참고: Windows 작업 관리자에서 정책 모델 서비스는 sepmdd.exe 로 나타납니다.

정책 모델 서비스에서는 PMDB(정책 모델 데이터베이스)를 사용합니다. 다른 CA Access Control 데이터베이스와는 달리 PMDB 에는 사용자, 그룹, 보호된 리소스, 리소스에 대한 액세스를 제어하는 규칙 등이 포함됩니다. 또한 PMDB 에는 *구독자* 데이터베이스 목록도 포함됩니다. 각 구독자는 별도의 컴퓨터에 있는 CA Access Control 데이터베이스이거나, 동일한 컴퓨터 또는 다른 컴퓨터에 있는 또 다른 PMDB 입니다. 구독자를 업데이트하는 PMDB 를 구독자의 *부모*라고 합니다.

PMDB 는 권한 제한 및 액세스 규칙이 유사한 여러 데이터베이스를 관리하는 데 유용한 도구입니다.

Windows 에서 정책 모델 이름은 UNIX 와의 호환성을 위해 대소문자를 구분합니다. 명령에 PMDB 이름을 지정할 때는 대소문자 구분에 주의하십시오. PMDB 이름의 첫 번째 문자는 영숫자 문자 '-' 및 '_'로 구성되어야 합니다.

참고: PMDB 및 호스트 이름에 영어 이외의 문자를 사용할 수 없습니다.

PMDB 이름이 대소문자를 구분하지만 동일한 컴퓨터에서 대소문자만 다른 동일한 이름의 PMDB 를 사용할 수 없습니다. CA Access Control 은 PMDB 이름을 파일 경로의 일부로 사용하지만 Windows 는 대소문자를 구분하지 않으므로 이렇게 사용할 수 없습니다. 예를 들어, myPMDB 와 MYpmdb 는 서로 다른 정책 모델 데이터베이스이지만 동일한 시스템에 함께 존재할 수 없습니다.

참고: PMDB 를 관리하는 방법(sepmd 유틸리티)에 대한 자세한 내용은 *참조 안내서*를 참조하십시오. selang 을 사용하여 원격으로 PMDB 를 관리하는 방법에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

디스크에서 PMDB 의 위치

컴퓨터에 있는 모든 PMDB 는 공용 디렉터리에 위치합니다. Windows 레지스트리의 다음 하위 키에서 _pmd_directory_ 값은 디렉터리의 이름을 지정합니다.

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\Pmd
```

NTFS 루트 디렉터리에서 _pmd_directory_의 기본값은 *ACInstallDir*\data 입니다. 여기서 *ACInstallDir* 은 CA Access Control 이 설치된 디렉터리입니다(기본적으로 C:\Program Files\CA\AccessControl\).

각 PMDB 는 공용 디렉터리의 하위 디렉터리를 사용합니다. 하위 디렉터리의 파일에는 정책 모델을 정의하는 데 필요한 데이터가 모두 들어 있습니다. 정책 모델 구성 설정은 CA Access Control 레지스트리 설정의 Pmd 하위 키에 저장됩니다. 하위 키의 이름은 정책 모델 이름입니다.

로컬 PMDB 관리

CA Access Control 에서는 PMDB 를 관리하기 위한 다음과 같은 유틸리티를 제공합니다.

sepmd

다음을 수행할 수 있는 PMDB 관리 유틸리티입니다.

- 구독자 관리
- 업데이트 파일 잘라내기
- 관리자 이중 제어
- 정책 모델 로그 파일 관리
- 기타 관리 작업 수행

참고: sepmd 에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

원격 PMDB 관리

또한 CA Access Control 은 pmd 환경에서 사용할 수 있는 여러 selang 명령을 제공합니다. 다음 명령을 사용하여 PMDB 를 원격으로 관리할 수 있습니다.

backuppmd

PMDB 백업

createpmd

PMDB 를 작성합니다.

deletepmd

PMDB 를 삭제합니다.

findpmd

컴퓨터에 있는 모든 PMDB 이름을 표시합니다.

listpmd

PMDB 에 대한 다음 정보를 나열합니다.

- 구독자 및 구독자 상태
- PMDB 에 대한 설명 및 PMDB 상태
- 업데이트 파일의 명령과 해당 오프셋
- 오류 로그의 내용

pmd

다음을 수행할 수 있는 PMDB 관리 명령입니다.

- 사용할 수 없는 구독자 목록에서 구독자 제거
- 정책 모델 오류 로그 지우기
- 정책 모델 서비스 시작 및 중지
- 정책 모델 잠금 및 잠금 해제
- 업데이트 파일 잘라내기

restorepmd

백업 파일로부터 PMDB 를 복원합니다.

subs

다음을 수행할 수 있는 PMDB 구독 명령입니다.

- 부모 PMDB 에 기존 구독자 추가
- 부모 PMDB 에 새 구독자 추가
- 데이터베이스(CA Access Control 또는 다른 PMDB)에 부모 PMDB 할당

subspmd

로컬 데이터베이스에 부모 PMDB 를 할당합니다.

unsubs

PMDB 에서 구독자를 제거합니다.

참고: pmd 환경에서 사용할 수 있는 selang 명령에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

아키텍처 종속성

CA Access Control 을 배포할 때는 사용자 환경의 계층 구조를 고려해야 합니다. 많은 사이트에서 네트워크는 다양한 아키텍처를 포함합니다. 트러스트된 프로그램 목록과 같은 일부 정책 규칙은 아키텍처에 따라 다릅니다. 반면 대부분의 규칙은 시스템 아키텍처와 관련이 없습니다.

계층을 사용하여 두 종류 규칙 모두를 포함할 수 있습니다. 아키텍처와 관련되지 않은 규칙에 대해 전역 데이터베이스를 정의하고 이 데이터베이스에 아키텍처에 따라 달라지는 규칙을 정의하는 구독자 PMDB 를 지정할 수 있습니다.

참고: 루트 PMBD 와 모든 구독자는 사용자 환경의 실제 필요에 따라 같은 컴퓨터나 개별 컴퓨터에 있을 수 있습니다.

예: 두 개 계층으로 이루어진 배포 계층 구조

다음 UNIX 예제는 약간만 수정하면 Windows 아키텍처에도 적용됩니다.

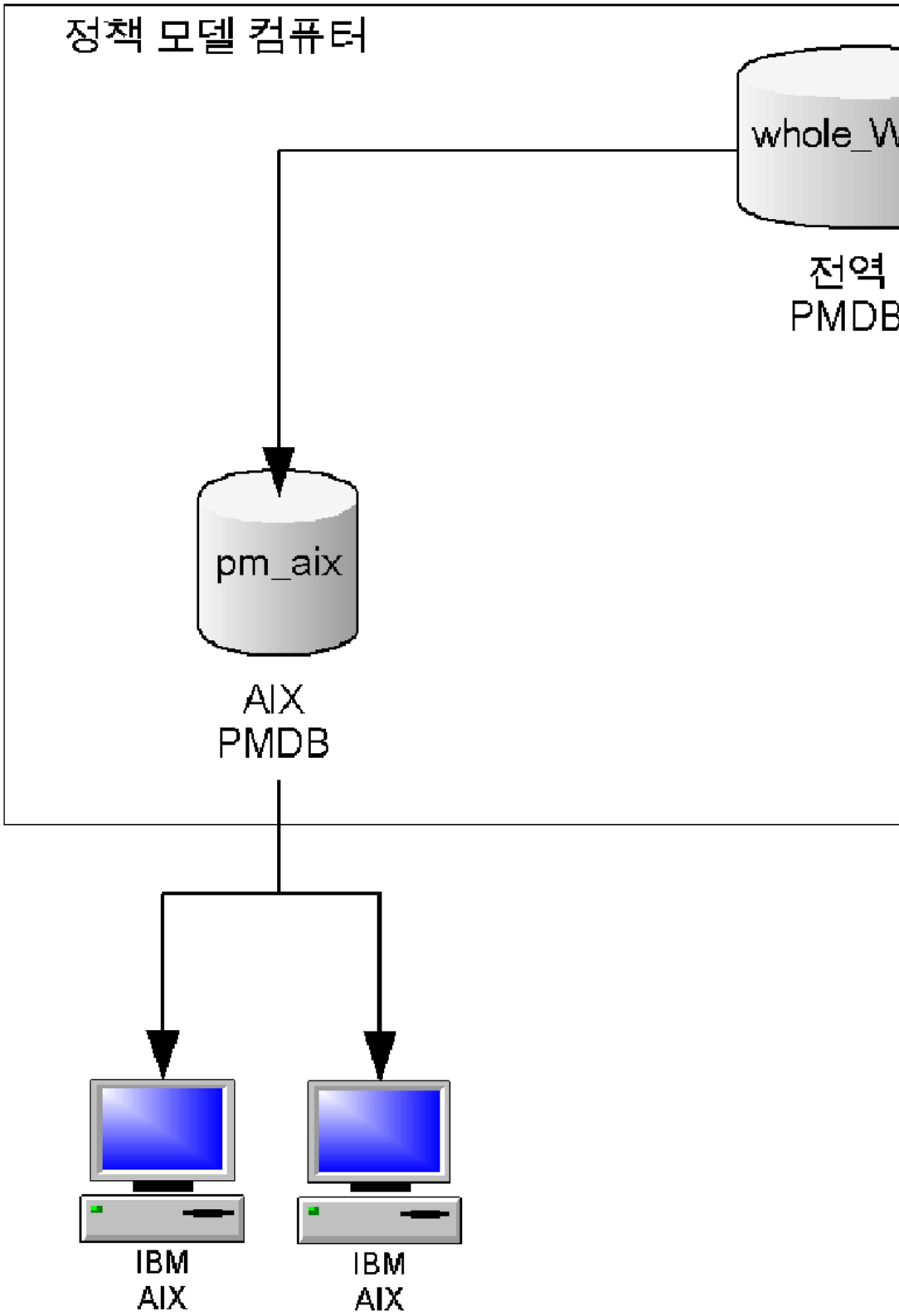
이 예에서 사이트는 IBM AIX 및 Sun Solaris 시스템으로 구성됩니다. IBM AIX 의 트러스트된 프로그램 목록이 Sun Solaris 의 트러스트된 프로그램 목록과 다르기 때문에 PMDB 는 아키텍처 종속성을 고려해야 합니다.

다중 아키텍처 PMDB 를 설정하려면 PMDB 를 다음과 같이 설정하십시오.

1. 이름이 whole_world 인 PMDB 를 정의하여 사용자, 그룹 및 기타 아키텍처 독립적인 모든 정책을 포함합니다.
2. 이름이 pm_aix 인 PMDB 를 정의하여 IBM AIX 고유의 모든 규칙을 포함합니다.

3. 이름이 pm_sol 인 PMDB 를 Sun Solaris 고유의 모든 규칙을 포함하도록 정의합니다.

   pm_aix 및 pm_solaris PMDB 는 whole_world PMDB 의 구독자입니다. 사이트의 모든 IBM AIX 컴퓨터는 pm_aix 의 구독자입니다. 사이트의 모든 Sun Solaris 컴퓨터는 pm_sol 의 구독자입니다. 이 개념은 다음 차트에 설명되어 있습니다.


정책 모델 컴퓨터
whole_W
전역
PMDB
pm_aix
AIX
PMDB
IBM
AIX
IBM
AIX

4. 사용자 추가 또는 SURROGATE 규칙 설정과 같은 플랫폼에 독립적인 명령을 whole_world 에 입력하면 사이트의 모든 데이터베이스가 자동으로 업데이트됩니다.
5. pm_aix 에 트러스트된 프로그램을 추가하면 IBM AIX 컴퓨터만 업데이트되고 Sun Solaris 시스템에는 영향을 주지 않습니다.

중앙에서 정책을 관리하기 위한 방법

CA Access Control 을 사용하면 다음 방법으로 단일 컴퓨터에서 여러 데이터베이스를 관리할 수 있습니다.

- **자동 규칙 기반 정책 업데이트** - 중앙 데이터베이스(PMDB)에서 정의한 일반 규칙은 구성된 계층의 데이터베이스에 자동으로 전파됩니다.

  참고: 이중 제어는 이 방법으로만, UNIX 에서만 사용할 수 있습니다. 자동 규칙 기반 정책 업데이트의 이중 제어에 대한 자세한 내용은 *UNIX 용 끝점 관리 안내서*를 참조하십시오. 자동 규칙 기반 정책 업데이트에 대한 자세한 내용은 *Windows 용 끝점 관리 안내서*를 참조하십시오.
- **고급 정책 관리** - 사용자가 배포하는 정책(규칙 그룹)은 호스트 또는 호스트 그룹 할당에 따라 모든 데이터베이스에 전파됩니다. 정책을 배포 취소(제거)하고 배포 상태와 배포 위반을 확인할 수도 있습니다. 이 기능을 사용하려면 추가 구성 요소를 설치하고 구성해야 합니다.

  참고: 고급 정책 관리에 대한 자세한 내용은 *엔터프라이즈 관리 안내서*를 참조하십시오.

자동 규칙 기반 정책 업데이트

중앙 데이터베이스에서 만든 단일 규칙 정책 업데이트(일반 selang 규칙)은 자동으로 구독자 데이터베이스에 전파됩니다. 동일한 데이터베이스에 여러 컴퓨터를 구독하고 데이터베이스를 서로 구독하여 계층을 작성할 수 있습니다. 설치 후 자동 규칙 기반 정책 업데이트에 맞게 환경을 구성해야 합니다.

참고: 이 정책 관리 방법에서는 계층 간에 단일 규칙 정책 업데이트만 만들 수 있습니다. 다른 기능은 고급 정책 관리 및 보고를 구현해야 사용할 수 있습니다.

자동 규칙 기반 정책 업데이트의 작동 방법

자동 규칙 정책 업데이트에 맞게 환경을 구성하면 중앙 데이터베이스에서 정의하는 각 규칙이 다음과 같은 방법으로 모든 구독자에게 자동으로 전파됩니다.

1. 하나 이상의 구독자가 있는 모든 PMDB 에 대해 규칙이 정의됩니다.
2. PMDB 에서 모든 구독자 데이터베이스에 명령을 보냅니다.
3. 구독자 데이터베이스에서 전파된 명령을 적용합니다.
   a. 구독자 데이터베이스가 응답하지 않는 경우 PMDB 는 구독자 데이터베이스가 업데이트될 때까지 일정한 간격(기본값: 30 분마다)으로 명령을 보냅니다.
   b. 구독자 데이터베이스가 응답하지만 명령 적용을 거부하는 경우 PMDB 는 해당 명령을 정책 모델 오류 로그 (페이지 181)에 기록합니다.
4. 구독자 데이터베이스가 다른 구독자의 부모인 경우 해당 구독자에 명령을 보냅니다.

예: 계층에 있는 모든 컴퓨터에서 사용자 제거

rmusr 명령을 사용하여 PMDB 에서 사용자를 삭제하면 동일한 rmusr 명령이 모든 구독자 데이터베이스에 보내집니다. 이런 방법으로 하나의 rmusr 명령을 통해 여러 컴퓨터의 많은 데이터베이스에서 사용자를 제거할 수 있습니다.

PMDB 를 사용하여 구성 설정을 전파하는 방법

정책 모델의 구성을 편집하면 새 구성 값이 정책 모델의 구독자로 전파됩니다.

다음 프로세스는 구성 업데이트가 정책 모델의 구독자에게 전파되는 방법을 설명합니다.

1. 정책 모델의 구성 값을 하나 이상 편집합니다.
2. 정책 모델이 새 구성 값을 가상 구성 파일에 기록합니다.

   참고: 가상 구성 파일은 audit.cfg 파일에 대한 값을 포함하지 않습니다. 정책 모델은 이 파일에 대한 변경 내용을 가상 구성 파일에 기록하지 않습니다.
3. 정책 모델은 새 구성 값을 해당 구독자에게 보냅니다.
4. selang 명령은 각 구독자를 새 구성 값으로 업데이트합니다.

가상 구성 파일

각 정책 모델에는 해당 구독자에 대한 구성 값을 포함하는 가상 구성 파일이 있습니다. 가상 구성 파일은 PMD 디렉터리에 cfg_*configname* 이란 이름으로 위치합니다. 여기서 *configname* 은 정책 모델 구성의 이름입니다.

가상 구성 파일은 audit.cfg 파일에 있는 구성 값을 포함하지 않습니다.

새 구독자가 구성되는 방법

정책 모델은 기존 구성 값을 사용하여 각각의 새 구독자를 구성합니다. 기존 구성 값은 가상 구성 파일에 저장되어 있습니다.

참고: 가상 구성 파일은 audit.cfg 파일에 있는 구성 값을 저장하지 않습니다. 새 구독자를 만들기 전에 audit.cfg 파일에 대해 수행한 모든 변경 내용은 새 구독자에게 전파되지 않습니다.

다음 프로세스는 정책 모델이 새 구독자를 구성하는 방법을 설명합니다.

1. 정책 모델에 새 구독자를 만듭니다.
2. 정책 모델이 해당 가상 구성 파일에 있는 값을 읽습니다.
3. 정책 모델은 가상 구성 파일에 있는 구성 값을 updates.dat 파일에 추가합니다. updates.dat 파일은 또한 정책에 대한 액세스 규칙을 포함합니다.
4. 정책 모델은 updates.dat 파일을 새 구독자에게 보냅니다.
5. selang 명령은 updates.dat 파일에 있는 값을 사용하여 새 구독자를 구성합니다.

계층을 설정하는 방법

CA Access Control 은 정책 모델 서비스를 사용하여 규칙 기반 정책 업데이트를 구성된 계층에 전파합니다. 여러 CA Access Control 컴퓨터가 동일한 PMDB 를 구독하게 하고 PMDB 를 서로 구독하게 하여 계층을 작성할 수 있습니다.

PMDB 계층을 설정하는 가장 간단한 방법은 CA Access Control 을 설치하면서 설정하는 것이므로, 설치를 시작하기 전에 계층을 어떻게 구성할지 미리 생각해 두는 것이 좋습니다. 부모 PMDB 와 해당 구독자는 서로 통신할 수 있어야 하므로 PMDB 계층의 모든 호스트가 동일한 네트워크에 속해 있는지 확인하십시오. 즉, 부모 PMDB 는 이름을 통해 구독자와 연결할 수 있어야 하며, 모든 구독자는 이름을 통해 부모 PMDB 와 연결할 수 있어야 합니다.

참고: CA Access Control 설치에 대한 자세한 내용은 *구현 안내서*를 참조하십시오.

설치 중에 작성한 구성을 변경하거나 설치 중에 PMDB 구조를 작성하지 않은 경우, 언제든지 PMDB 구성을 변경하거나 작성할 수 있습니다. 다음 방법 중 하나를 사용하여 이 작업을 수행할 수 있습니다.

- CA Access Control 끝점 관리 사용
- sepmd 유틸리티 사용

설치 후 PMDB 계층을 작성하고 자동 규칙 기반 정책 업데이트를 활성화하려면 다음 작업을 수행합니다.

1. 마스터 PMDB 를 작성하고 구성합니다.
2. (선택 사항) 구독자 PMDB 를 작성하고 구성합니다.
3. *끝점*이라고 하는 구독 컴퓨터에 대해 부모 PMDB 를 정의합니다.

구독자 업데이트

구독자를 업데이트할 때 정책 모델에서는 다음 작업을 수행합니다.

1. 정책 모델은 구독자 이름이 정책 모델에서 추가되거나 삭제될 때 구독자 이름을 정규화하려고 시도합니다.
2. PMDB 서비스 sepmdd 에서 구독자 데이터베이스를 업데이트하려고 시도합니다.
3. 최대 시간이 경과했는데도 서비스에서 구독자를 업데이트하지 못한 경우에는 해당 구독자를 건너뛰고 목록의 나머지 구독자를 업데이트합니다.
4. 구독자 목록의 최초 검사를 완료하면 sepmdd 는 두번째 검사를 수행합니다. 여기서는 첫번째 검사에서 업데이트에 실패한 구독자를 업데이트하려고 시도합니다.

참고: PMDB 에서 구독자에게 업데이트를 전파하는 동안 오류가 발생하면 sepmdd 서비스는 정책 모델 오류 로그 파일 (페이지 181)에 항목 하나를 작성합니다. 이 ERROR_LOG 파일은 PMDB 디렉터리 (페이지 166)에 있습니다.

정책 모델 데이터베이스 업데이트

PMDB 가 있는 컴퓨터에서 작업할 때 PMDB 가 자동으로 업데이트되지는 않습니다. PMDB 를 업데이트하려면 해당 PMDB 를 대상 데이터베이스로 지정해야 합니다.

selang 또는 CA Access Control 끝점 관리를 사용하여 PMDB 를 지정할 수 있습니다. selang 을 사용하여 대상 데이터베이스를 지정하려면 selang 명령 셸에서 hosts 명령을 사용합니다.

```
hosts pmd_name@pmd_host
```

이제 모든 selang 명령으로 지정한 정책 모델 데이터베이스가 업데이트됩니다. 그런 다음 명령이 이 컴퓨터와 모든 구독자 컴퓨터의 활성 데이터베이스에 자동으로 전파됩니다.

예: 대상 PMDB 지정

myPMD_host 에서 대상 데이터베이스를 policy1 로 설정하려면 다음 명령을 사용합니다.

```
hosts policy1@myPMD_host
```

이제 newusr 명령을 입력하면 새 사용자가 이 컴퓨터와 모든 구독자 컴퓨터의 활성 데이터베이스뿐 아니라 policy1 데이터베이스에도 추가됩니다.

업데이트 파일 정리

sepmd 유틸리티는 pmd.ini 파일에 받은 각 업데이트를 자동으로 씁니다. 이 파일이 지나치게 커지지 않게 하려면 파일에서 처리된 업데이트를 정기적으로 삭제하는 것이 좋습니다.

업데이트 파일을 정리하려면 다음 명령을 사용합니다.

```
sepmd -t pmdbName auto
```

sepmd 는 전파되지 않은 첫 번째 업데이트 항목의 오프셋을 계산하여 그 이전의 모든 업데이트 항목을 삭제합니다.

참고: sepmd 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

암호 전파 및 동기화

PMDB 계층을 설정한 후에는, Windows 사용자 관리자 또는 CA Access Control 이외의 소프트웨어를 사용하여 사용자 암호를 변경할 때 이 계층을 사용하여 사용자 암호가 시스템 전체에서 동기화되도록 할 수 있습니다.

참고: CA Access Control 은 메인프레임 암호 동기화도 지원합니다.

다음 단계를 수행하십시오.

1. PMDB 계층을 작성합니다.
2. 사용자 또는 관리자가 암호를 변경할 수 있는 모든 스테이션의 레지스트리에서 passwd_pmd 항목 값으로 해당되는 부모 PMDB 의 이름을 입력합니다.

   ```
   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\AccessControl\passwd_pmd
   ```

   그러면 PMDB 가 모든 PMDB 의 구독자에게 암호 변경 사항을 전파합니다.

참고: 사용자가 정의되어 있지 않은 구독자에게 PMDB 가 사용자 암호를 전송할 경우, 설정은 변경되지 않으며 사용자는 구독자에 대해 정의되지 않은 상태로 유지됩니다.

구독자 제거

특정 구독자에 더 이상 업데이트를 전파하지 않으려면 해당 구독자를 제거해야 합니다.

구독자를 제거하려면

1. 구독자 목록에서 컴퓨터를 제거합니다.

   ```
   sepmd -u PMDB_name computer_name
   ```

   컴퓨터가 정책 모델 구독 목록에서 제거됩니다.
2. 구독 목록에서 제거한 컴퓨터에서 seosd 를 종료합니다.

   ```
   secons -s
   ```

   seosd 서비스가 종료됩니다.

3. 구독 목록에서 제거한 컴퓨터의 다음 레지스트리 키에 있는 parent_pmd 레지스트리 값을 삭제합니다.

   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\AccessControl

   컴퓨터에서 더 이상 부모 PMDB 의 업데이트를 받지 않습니다.

4. seosd 를 다시 시작합니다.

   구독 목록에서 제거한 컴퓨터의 활성 데이터베이스는 더 이상 지정한 PMDB 의 구독자가 아닙니다.

참고: 데이터베이스가 PMDB 에서 구독 취소된 경우 PMDB 는 더 이상 명령을 보내지 않습니다.

업데이트 필터링

PMDB 를 통해 다른 구독자 데이터베이스에 있는 다른 데이터 하위 집합을 업데이트하려면 구독자 데이터베이스로 보낼 레코드를 정의해야 합니다.

업데이트를 필터링하려면

1. PMDB 가 구독자의 하위 집합에 대한 부모 역할을 하도록 구성합니다.
2. 부모 PMDB 의 레지스트리 키에 있는 *Filter* 레지스트리 항목을 수정하여 같은 컴퓨터에서 설정한 필터 파일을 가리키도록 만듭니다.

   그러면 구독자 데이터베이스에 대한 업데이트가 해당 필터를 통과하는 레코드로 제한됩니다.

정책 모델 필터 파일

필터 파일은 각각 6 개의 필드가 있는 줄로 구성됩니다. 이 필드에는 다음에 대한 정보가 포함되어 있습니다.

- 허용되거나 거부된 액세스 형식.

  예: EDIT 또는 MODIFY

- 영향 받는 환경. 예:

  예: AC 또는 기본

- 레코드 클래스.

  예: USER 또는 TERMINAL

- 규칙이 적용되는 클래스 내 개체.

  예를 들어 User1, AuditGroup, 또는 COM2 입니다.

- 레코드가 허용하거나 취소한 속성.

  예를 들어 필터 줄의 OWNER 와 FULL_NAME 은 이러한 속성을 갖는 명령이 필터링된다는 것을 나타냅니다. *참조 안내서*에 나타난 대로 각 속성을 정확히 입력해야 합니다.
- 레코드를 구독자 데이터베이스로 전달해야 하는지 여부.

  PASS 또는 NOPASS

필터 파일의 각 줄에 다음 규칙이 적용됩니다.

- 필드에 별표(*)를 사용하여 가능한 모든 값을 표시할 수 있습니다.
- 두 개 이상의 행이 동일한 레코드를 포함할 경우, *첫 번째* 적용 가능한 행이 사용됩니다.
- 필드는 공백으로 구분합니다.
- 필드에 여러 개의 값이 있는 경우 세미콜론으로 값을 구분합니다.
- #으로 시작하는 줄은 설명 줄로 간주됩니다.
- 빈 줄은 허용되지 않습니다.

예: 필터 파일

다음 예에서는 필터 파일의 행을 설명합니다.

CREATE	AC	USER	*	FULL_NAME;OBJ_TYPE	NOPASS
액세스 형식	환경	class	레코드 이름 (* =모두)	properties	처리

예를 들어, 이 줄이 있는 파일의 이름이 Printer1_Filter.flt 이고 filter=C:\Program Files\CA\AccessControl\\Printer1_Filter.flt 가 되도록 PMDB PM-1 의 레지스트리를 편집하는 경우, PMDB PM-1 은 FULL_NAME 및 OBJ_TYPE 속성으로 새 사용자를 작성하는 레코드를 구독자에게 전파하지 않습니다.

정책 모델 오류 로그 파일

정책 모델 오류 로그는 시간순으로 구성되며 다음과 비슷합니다.

오류 텍스트	오류 범주
20 Nov 03 11:56:07 (pmdb1): fargo nu u5 0 Retry 오류: 로그인 절차가 실패했습니다.(10068) 오류: 상위가 아닌 PMDB에서 업데이트할 수 없습니다. (pmdb1@name.company.com) (10104)	구성 오류
20 Nov 03 19:53:17 (pmdb1): fargo nu u5 0 Retry 오류: 연결하지 못했습니다.(10071) 호스트에 연결할 수 없습니다.(12296)	연결 오류
20 Nov 03 11:57:06 (pmdb1): fargo nu u5 560 Cont 오류: 사용자 u5를 작성하지 못했습니다.(10028) 이미 있음(-9) 20 Nov 03 11:57:06 (pmdb1): fargo nu u5 1120 Cont 오류: 사용자 u5를 작성하지 못했습니다.(10028) 이미 있음(-9)	데이터베이스 업데이트 오류

정책 모델 오류 로그는 이진 형식이기 때문에 다음 명령을 입력해야만 볼 수 있습니다.

ACInstallDir/bin sepmd -e *pmdname*

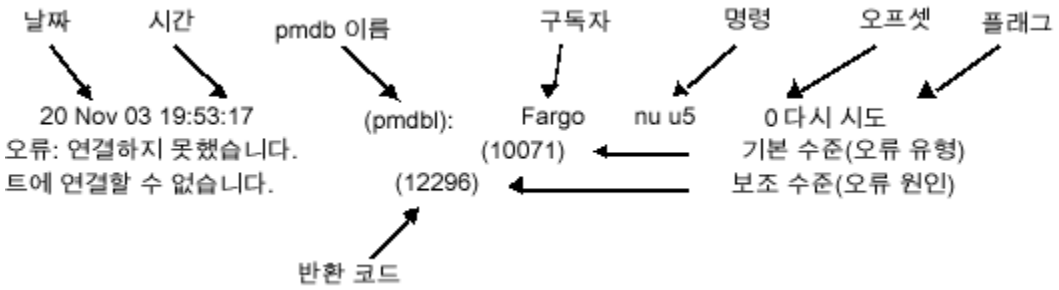
참고: rm 등과 같은 UNIX 명령을 사용하여 오류 로그를 수동으로 삭제하지 마십시오. 로그를 삭제하려면 다음 명령만 사용해야 합니다.

ACInstallDir/bin sepmd -c *pmdname*

중요! CA Access Control r5.1 이상 버전의 오류 로그 형식은 이전 버전 형식과 호환되지 않습니다. sepmd 는 이러한 이전 버전의 오류 로그를 처리할 수 없습니다. 이 형식의 버전으로 업그레이드하는 경우 이전의 오류 로그는 ERROR_LOG.bak 으로 복사되고 sepmdd 를 시작할 때 새 로그 파일이 생성됩니다.

예: PMDB 업데이트 오류 메시지

다음은 일반적인 오류 메시지의 예입니다.



- 맨 위 행은 항상 날짜, 시간 및 구독자로 구성됩니다. 그런 다음 오류를 생성한 명령과 오프셋(10 진수 형식)을 차례로 표시하여 업데이트 파일 내에서 실패한 업데이트의 위치를 나타냅니다. 마지막으로 PMDB 에서 업데이트를 자동으로 다시 시도할지 여부를 나타내는 플래그가 표시됩니다.
- 두 번째 행에는 기본 수준 메시지(발생 오류 유형) 및 해당 반환 코드의 예가 표시됩니다.
- 세 번째 행에는 보조 수준 메시지(오류가 발생한 이유)와 반환 코드의 예가 표시됩니다.

예: 오류 메시지

명령은 두 개 이상의 오류를 생성하고 표시할 수도 있습니다. 또한 하나의 오류가 기본 수준 메시지 및/또는 보조 수준 메시지로 구성될 수도 있습니다.

다음 오류는 하나의 메시지 수준만 가집니다.

```
Fri Dec 29 10:30:43 2003 CIMV_PROD:해제하지 못했습니다. 반환 코드 = 9241
```

이 메시지는 sepmd pull 이 이미 사용 가능한 구독자를 해제하려고 할 때 표시됩니다.

기본 정책 모델 저장소

모든 네이티브 환경 사용자 및 그룹 개체 유형을 PMDB 에 저장할 수 있습니다. 이 정보를 PMDB 에 저장하면, show user 또는 show group 과 같은 show 명령을 사용하여 개체에 대한 정보를 수신할 수 있습니다. 반환된 개체는 Windows 또는 UNIX 구독자에서 정의한 실제 개체의 이미지입니다.

정책 모델에 연결한 후 사용자는 다음 환경을 선택할 수 있습니다.

- AC
- 기본
- NT
- UNIX
- 구성

참고: 기본은 Windows 운영 체제에서 작업할 때는 Windows 와 동일하게 작동하고, UNIX 운영 체제에서 작업할 때는 UNIX 와 동일하게 작동합니다.

네이티브 환경 저장소를 사용하려면 다음 명령을 사용합니다.

- selang 프롬프트에 다음 명령을 입력합니다.

  ```
  env NT; find
  ```

  결과 목록에는 모든 네이티브 환경 개체 유형이 나열됩니다.

  참고: 이러한 개체 유형에 대한 설명은 *참조 안내서*의 Windows 환경 클래스 및 속성을 참조하십시오.

- NT 및 Active Directory USER 속성 목록을 수신하려면 다음 명령을 입력합니다.

  ```
  env NT; ruler user
  ```

- NT 및 Active Directory GROUP 속성 목록을 수신하려면 다음 명령을 입력합니다.

  ```
  env NT; ruler group
  ```

정책 모델이 다른 (부모) 정책 모델의 구독자일 경우, 정책 모델은 전파 과정을 통해 부모 정책 모델로부터 데이터를 수신하고 데이터베이스에 모든 사용자 및 그룹 속성을 저장하여 사용자는 이러한 정보를 확인하고 변경할 수 있습니다.

참고: 자세한 내용은 *참조 안내서*의 sepmd 유틸리티를 참조하십시오.

정책 모델 백업

PMDB 를 백업할 때는 정책 모델 데이터베이스에 있는 데이터를 다른 디렉터리로 복사합니다. 여기에는 다음이 포함됩니다.

- 정책 정보
- 정책 모델 구독자의 목록
- 구성 설정
- 레지스트리 항목
- updates.dat 파일

다른 플랫폼, 운영 체제, CA Access Control 버전을 사용하는 백업 파일로부터 PMDB 를 복원할 수 없습니다. 정책 모델은 반드시 동일한 플랫폼, 운영 체제, CA Access Control 버전을 실행하는 호스트에 백업해야 합니다.

sepmd 를 사용하여 PMDB 백업

PMDB 를 백업할 때 정책 모델 데이터베이스의 데이터를 지정된 디렉터리로 복사합니다. 백업된 PMDB 파일은 가급적 CA Access Control 액세스 규칙에 의해 보호되는 안전한 곳에 보관해야 합니다.

sepmd 유틸리티를 사용하여 로컬 호스트의 PMDB 를 백업할 수 있습니다. selang 명령을 사용하여 원격 호스트의 PMDB 를 백업할 수도 있습니다.

참고: PMDB 를 재귀적으로 백업할 수 있습니다. 재귀적 백업은 계층에 있는 모든 PMDB 를 지정하는 호스트에 백업하고, 백업이 호스트로 이동되었을 때 구독이 계속 유효하도록 PMDB 구독자를 수정합니다. 동일한 호스트에 마스터 및 자식 PMDB 가 배포된 경우에는 재귀적 백업만 사용할 수 있습니다.

sepmd 를 사용하여 PMDB 를 백업하려면

1. 다음 명령을 사용하여 PMDB 를 잠급니다.

   ```
   sepmd -bl pmdb_name
   ```

   PMDB 가 잠기며, 이제 구독자에게 어떤 명령도 보낼 수 없습니다.

2. 다음 작업 중 하나를 수행합니다.

   - 다음 명령을 사용하여 PMDB 를 백업합니다.

     ```
     sepmd -bh pmdb_name [destination_directory]
     ```

   - 다음 명령을 사용하여 PMDB 를 재귀적으로 백업합니다.

     ```
     sepmd -bh pmdb_name [destination_directory] [backup_host_name]
     ```

   참고: 대상 디렉터리를 지정하지 않으면 백업이 다음 디렉터리에 지정됩니다.

   ACInstallDir\data\policies_backup\pmdb_name

3. 다음 명령을 사용하여 PMDB 를 잠금 해제합니다.

   ```
   sepmd -ul pmdb_name
   ```

   PMDB 가 잠금 해제되며, 이제 구독자에게 명령을 보낼 수 있습니다.

selang 을 사용하여 PMDB 백업

PMDB 를 백업할 때 정책 모델 데이터베이스의 데이터를 지정된 디렉터리로 복사합니다. 백업된 PMDB 파일은 가급적 CA Access Control 액세스 규칙에 의해 보호되는 안전한 곳에 보관해야 합니다.

selang 명령을 사용하여 로컬 또는 원격 호스트의 PMDB 를 백업할 수 있습니다. sepmd 유틸리티를 사용하여 로컬 호스트에서 PMDB 를 백업할 수도 있습니다.

참고: PMDB 를 재귀적으로 백업할 수 있습니다. 재귀적 백업은 계층에 있는 모든 PMDB 를 지정하는 호스트에 백업하고, 백업이 호스트로 이동되었을 때 구독이 계속 유효하도록 PMDB 구독자를 수정합니다. 동일한 호스트에 마스터 및 자식 PMDB 가 배포된 경우에는 재귀적 백업만 사용할 수 있습니다.

selang 을 사용하여 PMDB 를 백업하려면

1. (선택 사항) selang 을 사용하여 원격 호스트로부터 PMDB 에 연결하려는 경우 다음 명령으로 PMDB 호스트에 연결합니다.

   ```
   host pmdb_host_name
   ```

2. 다음 명령을 사용하여 PMD 환경으로 이동합니다.

   ```
   env pmd
   ```

3. 다음 명령을 사용하여 DMS 를 잠급니다.

   ```
   pmd pmdb_name lock
   ```

   PMDB 가 잠기며, 이제 구독자에게 어떤 명령도 보낼 수 없습니다.

4. 다음 명령을 사용하여 DMS 데이터베이스를 백업합니다.

   ```
   backuppmd pmdb_name [destination(destination_directory)] [hir_host(host_name)]
   ```

   참고: 대상 디렉터리를 지정하지 않으면 백업이 다음 디렉터리에 지정됩니다.

   ACInstallDir\data\policies_backup\pmdbName

5. 다음 명령을 사용하여 PMDB 를 잠금 해제합니다.

   ```
   pmd pmdb_name unlock
   ```

   PMDB 가 잠금 해제되며, 이제 구독자에게 명령을 보낼 수 있습니다.

정책 모델 복원

정책 모델이 복원되면 CA Access Control 은 백업 PMDB 파일을 지정된 디렉터리에 복사합니다. 다음을 포함하여 원래 PMDB 파일에 있던 모든 항목이 새 PMDB 디렉터리에 복사됩니다.

- 정책 정보
- 정책 모델 구독자의 목록
- 구성 설정
- 레지스트리 항목
- updates.dat 파일

대상 디렉터리에 기존 PMBD 가 있는 경우 CA Access Control 은 복원 파일을 대상 디렉터리에 복사하기 전에 기존 파일을 삭제합니다.

다른 플랫폼, 운영 체제, CA Access Control 버전을 사용하는 백업 파일로부터 PMDB 를 복원할 수 없습니다. 정책 모델은 반드시 동일한 플랫폼, 운영 체제, CA Access Control 버전을 실행하는 호스트에 백업해야 합니다.

PMDB 복원

PMDB 를 복원할 때 CA Access Control 은 PMDB 백업 파일의 데이터를 사용자가 지정한 디렉터리로 복사합니다. 복원을 수행하는 터미널에서 CA Access Control 이 실행 중이어야 합니다.

참고: 다른 터미널에서 PMDB 를 백업 및 복원하는 경우 PMDB 는 복원된 PMDB 데이터베이스에서 터미널 리소스를 자동으로 업데이트하지 않습니다. 복원된 PMDB 에 새 터미널 리소스를 추가해야 합니다. 새 터미널 리소스를 추가하려면 복원된 PMDB 를 중지하고, *selang -p pmdb* 명령을 실행한 다음, 복원된 PMDB 를 시작하십시오.

PMDB 를 복원하려면 PMDB 를 복원할 터미널에서 다음 중 *하나*를 실행하십시오.

- sepmd -restore 유틸리티
- selang restore pmd 명령

참고: 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오. selang 명령에 대한 자세한 내용은 *selang 참조 안내서*를 참조하십시오.

PMDB 와 Unicenter 통합

PMDB 를 Unicenter TNG 와 통합하면 PMDB 를 사용하여 Unicenter TNG 개체가 명령 프로세서, Event Management 및 Workload Management 등의 여러 Unicenter TNG 구성 요소에 의해 조작되지 않도록 안전하게 보호하는 규칙을 작성할 수 있습니다.

통합을 수동으로 수행해야 합니다.

PMDB 를 Unicenter TNG 와 통합하려면

1. PMDB 를 작성합니다.
2. 다음 명령을 사용하여 Unicenter Security 옵션을 PMDB 로 마이그레이션합니다.

   `MigOpts` *`pmdb-name`*

   여기서 *pmdb-name* 은 PMDB 의 이름입니다.

   참고: 이 단계는 Unicenter Security 를 사용하고 CA Access Control 설치 도중에 Unicenter 통합에서 Security 데이터 마이그레이션을 선택한 경우에만 필요합니다. Unicenter Security 를 사용하지 않은 경우에는 어떤 보안 옵션도 설정하지 않은 것이므로 PMDB 로 마이그레이션할 내용이 없습니다.
3. 다음 명령을 사용하여 사용자 정의 Unicenter TNG 자산 유형에 대한 클래스를 작성합니다.

   `defclass.bat.` *`pmdb-name`*

   여기서 *pmdb-name* 은 PMDB 의 이름입니다.

   참고: 이 단계는 Unicenter Security 를 사용하고 사용자 정의 자산 유형을 만든 경우에만 필요합니다. CA Access Control 설치 중에 Unicenter 통합을 선택하면 모든 새 PMDB 에 Unicenter TNG 자산 유형이 자동으로 정의됩니다.

메인프레임 암호 동기화

CA Access Control 은 CA Access Control 을 실행하는 Windows 또는 UNIX 컴퓨터와 CA Top Secret, CA ACF2 또는 RACF 보안 제품(및 CA Common Services CAICCI 패키지)을 실행하는 메인프레임 간 암호 동기화를 지원합니다. 동기화는 표준 CA Access Control 암호 정책 모델 방법을 사용하여 수행됩니다.

메인프레임 사용자의 암호 변경 사항은 암호 정책 모델 계층 구조의 모든 컴퓨터에 전파됩니다.

메인프레임 암호 동기화 필수 구성 요소

TNG/TND/NSM 이 설치된 서버에서 "메인프레임 암호 동기화"를 사용하려면 CA Access Control 에서 필수 구성 요소 "TNG/TND/NSM 픽스 - T129430"이 요구됩니다. 이 픽스를 얻으려면 지원 담당자에게 문의하십시오.

제 11 장: 일반 보안 기능

이 섹션은 다음 항목을 포함하고 있습니다.

유지 관리 모드 보호(자동 모드)

CA Access Control 에는 유지 관리를 위해 CA Access Control 서비스가 종료될 때 보호하기 위한 자동 모드라고 하는 유지 관리 모드가 있습니다. 이 모드에서 이러한 서비스가 종료된 동안 CA Access Control 은 이벤트를 거부합니다.

CA Access Control 은 실행 중일 때 보안상 중요한 이벤트를 차단하고 이벤트가 허용되었는지 검사합니다. 유지 관리 모드가 활성화되지 않은 상태에서 CA Access Control 서비스가 종료되면 모든 이벤트가 허용됩니다. 유지 관리 모드가 활성화된 상태에서 CA Access Control 서비스가 종료되면 이벤트가 거부되어, 시스템 유지 관리가 수행되는 동안 사용자 활동이 중지됩니다.

유지 관리 모드는 기본적으로 비활성화되어 있으며, 필요할 때 활성화할 수 있습니다.

CA Access Control 보안 서비스가 종료되었을 때:

- 유지 관리 모드가 활성화되어 있으면 유지 관리 사용자가 실행한 이벤트와 특별한 경우를 제외하고 보안에 민감한 이벤트가 모두 거부됩니다.
- 유지 관리 모드가 비활성화되어 있으면 CA Access Control 이 아무 작업도 중단하지 않고 실행이 운영 체제로 전달됩니다.

유지 관리 모드가 활성화되고 보안이 종료된 경우 금지된 이벤트가 감사 로그 파일에 기록되지 않습니다.

유지 관리 모드를 활성화하려면 다음 단계를 수행하십시오.

1. CA Access Control 서비스가 종료되었는지 확인합니다.
2. 레지스트리 편집기를 사용하여 다음 레지스트리 키로 이동합니다.

   `\HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\FsiDrv`

   다음 값을 변경합니다.

   - SilentModeEnabled = 1
   - SilentModeAdmins = *special_admins*

     special_admins 변수는 CA Access Control 서비스가 종료된 동안 컴퓨터에 액세스할 수 있는 사용자 이름 목록을 정의합니다.

     각 사용자에 대해 새 줄을 사용합니다. 지정 여부와 상관없이 *SYSTEM* 은 언제나 유지 관리 모드 사용자입니다.

     참고: Windows 2000 및 Windows NT 에서는 regedit 을 사용하여 SilentModeAdmins 키를 편집할 수 없습니다. 대신 Regedt32.exe 를 사용하십시오.
3. 명령 셸에서 "seosd -start" 명령을 사용하거나 Windows 시작 메뉴에서 옵션을 사용하여 CA Access Control 서비스를 시작합니다.

이제 CA Access Control 서비스가 종료될 경우 SilentModeAdmins 레지스트리 키 아래 나열된 사용자만 컴퓨터에 액세스할 수 있으며 다른 모든 사용자는 액세스를 시도할 때 거부 메시지를 수신합니다.

드라이버 무시

특정 드라이버가 CA Access Control 권한 부여 검사 없이 동작하도록 지정하려면 이러한 드라이버를 무시하도록 정의하십시오. 예를 들어, 바이러스 백신 프로그램 드라이버를 무시하도록 정의하여 CA Access Control 권한 부여 검사 없이 파일을 열어 스캔하도록 할 수 있습니다. 이때 무시하도록 정의하지 않으면 해당 드라이버와 CA Access Control 사이에 충돌이 발생합니다.

참고: Trend Micro™ PC-cillin Antivirus 의 현재 버전은 기본적으로 무시하도록 설정되어 있습니다.

드라이버를 무시하려면

1. 무시하도록 정의할 드라이버의 수로 BypassDriversCount 레지스트리 항목 값을 설정합니다.

   이 항목은 CA Access Control 레지스트리의 FsiDrv 키에서 찾을 수 있습니다.

   참고: CA Access Control 레지스트리 항목을 변경하려면 우선 CA Access Control 을 중지해야 합니다.

2. 무시할 각 드라이버에 대해 다음을 수행합니다.

   a. DriverName_*drvNumber* 란 이름으로 REG_SZ 유형의 레지스트리 항목을 만듭니다.

      첫 번째 항목은 DriverName_0 이어야 하고 마지막 항목은 DriverName_*X* 여야 합니다. 여기서 *X* 는 BypassDriversCount - 1 입니다.

   b. 각 DriverName_*drvNumber* 항목을 편집하여 무시할 프로그램 드라이버의 이름으로 항목 값을 설정합니다.

      이 값에는 드라이버의 이름(예: thisdrv.sys)만 사용해야 합니다.

3. CA Access Control 을 다시 시작합니다.

   그러면 CA Access Control 이 다시 로드되고 레지스트리에 정의한 드라이버를 무시합니다.

예제: 호환성 문제를 해결하기 위해 드라이버 무시

이 예제는 바이러스 백신 드라이버(avDriverA.sys 및 avDriverB.sys)를 무시하도록 정의하여 바이러스 백신 제품과 CA Access Control 간의 호환성 문제를 해결합니다. CA Access Control 레지스트리 트리에 있는 아래의 FsiDrv 키에 드라이버를 무시하기 위한 레지스트리 항목을 설정합니다.

```
HKLM\SOFTWARE\ComputerAssociates\AccessControl\FsiDrv
```

다음과 같이 레지스트리 항목을 설정하십시오.

이름	유형	데이터
BypassDriversCount	REG_DWORD	2
DriverName_0	REG_SZ	avDriverA.sys
DriverName_1	REG_SZ	avDriverB.sys

BypassDriversCount 레지스트리 항목 값 2 는 CA Access Control 이 두 개의 드라이버를 무시한다는 것을 의미합니다. 각 DriverName_*drvNumber* 레지스트리 항목 값은 무시할 하나의 드라이버를 정의합니다.

드라이버 차단 전환

CA Access Control 필터 드라이버의 차단을 활성화 또는 비활성화할 수 있습니다.

참고: 차단이 비활성화되는 경우 필터 드라이버에 의해 적용되지 않는 CA Access Control 보호 기능은 여전히 적용됩니다. 여기에는 암호 품질 확인, 로그인 이벤트, Windows 서비스 이벤트, STOP 등이 포함됩니다.

차단을 활성화하려면 UseFsiDrv 를 1 로 설정하고, 비활성화하려면 UseFsiDrv 를 0 으로 설정하십시오.

이 구성 설정은 CA Access Control 레지스트리의 AccessControl 섹션에서 찾을 수 있습니다.

이 레지스트리 값을 변경한 후 CA Access Control 서비스를 다시 시작합니다.

CA Access Control 커널 차단 비활성화

커널 수준에서 다음 CA Access Control 차단을 비활성화할 수 있습니다.

- 네트워크 차단
- 프로세스 차단
- 레지스트리 차단
- 파일 차단

네트워크, 프로세스, 레지스트리 및 파일 클래스가 비활성화되어 있고 커널 활동을 차단하는데 이러한 클래스를 사용하지 않는 경우에도 네트워크, 프로세스, 레지스트리 및 파일 차단 처리 코드는 부팅 시 시작되어 런타임에 작동하므로 성능에 영향을 미칩니다. 성능 향상을 위해 하나 이상의 차단을 부팅 시 시작되지 않도록 비활성화할 수 있습니다.

커널 수준에서 CA Access Control 차단을 비활성화하려면

1. REG_DWORD 형식의 다음 레지스트리 항목을 하나 이상 만들고 하나 이상의 항목 값을 1 로 설정합니다.
   - DisableNetworkInterception - 네트워크 차단 비활성화
   - DisableProcessInterception - 프로세스 차단 비활성화
   - DisableRegistryInterception - 레지스트리 차단 비활성화
   - DisableFileInterception - 파일 차단 비활성화

   항목은 다음 레지스트리 키 아래에 만들어야 합니다.

   ```
   HKLM\SYSTEM\CurrentControlSet\Services\drveng\Parameters
   ```
2. 컴퓨터를 다시 시작합니다.

   비활성화된 차단 유형을 시작하지 않고 CA Access Control 이 다시 로드됩니다.

스택 오버플로 보호

STOP(스택 오버플로 보호)은 해커가 스택 오버플로를 작성하고 사용하여 시스템에 침입하는 것을 금지하는 기능입니다. 스택 오버플로를 사용하면 해커가 원격 또는 로컬 시스템에서 관리자처럼 원하는 만큼 임의의 명령을 실행할 수 있습니다. 해커는 운영 체제 또는 다른 프로그램의 버그를 이용하여 이 작업을 수행합니다. 이와 같이 특수한 유형의 버그를 통해 사용자는 프로그램 스택을 덮어쓸 수 있으며 다음에 실행될 명령을 변경할 수 있습니다.

STOP 은 컴퓨터의 각 응용 프로그램에 대한 중요한 운영 체제 호출을 차단합니다. 그런 다음 각 호출에 대해 초기 분석을 수행한 후, 호출이 의심스러운지 확인하는 추가 분석을 위해 전송됩니다. 추가 분석은 STOP 구성 및 서명 파일의 데이터를 사용하여 수행됩니다.

STOP 활성화

STOP 을 사용하면 시스템에 침투하기 위해 스택 오버플로를 만들고 악용하는 해커의 활동을 차단할 수 있습니다. CA Access Control 을 설치할 때 STOP 을 활성화할 수 있습니다. 또는 수동으로 STOP 을 활성화할 수도 있습니다.

STOP 을 활성화하려면

1. 다음 명령을 입력합니다.

   ```
   secons -s
   ```

   CA Access Control 을 종료합니다.

2. STOP *OperationMode* 레지스트리 항목을 1 로 설정합니다.

   레지스트리 항목은 다음 키에서 찾을 수 있습니다.

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\
   PlugIns\StopPlg
   ```

   CA Access Control 이 시작되면 STOP 모듈이 로드되고 컴퓨터에서 STOP 이 활성화됩니다.

3. (선택 사항) 다음 키에 있는 레지스트리 항목을 사용하여 STOP 구성을 조정합니다.

   HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\StopPlg

   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\STOP

   참고: STOP 레지스트리 설정에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

4. 다음 명령을 입력합니다.

   ```
   seosd -start
   ```

   CA Access Control 이 시작됩니다.

서명 파일 업데이트를 수신하도록 STOP 구성

작업 환경의 모든 컴퓨터가 스택 오버플로 차단을 위해 필요한 최신 STOP 정보를 가지고 있는지 확인할 수 있습니다. 중앙 컴퓨터에 있는 STOP 서명 파일을 업데이트하고 이 파일을 정기적으로 가져오도록 다른 컴퓨터들을 설정하면 됩니다.

서명 파일 업데이트를 수신하도록 STOP 을 구성하려면

1. 다음 명령을 입력합니다.

   ```
   secons -s
   ```

   CA Access Control 을 종료합니다.

2. CA Access Control 이 서명 파일을 가져오도록 할 컴퓨터의 호스트 이름으로 *STOPSignatureBrokerName* 레지스트리 항목을 설정합니다.

   레지스트리 항목은 다음 키에서 찾을 수 있습니다.

   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\STOP

   CA Access Control 을 시작하면 CA Access Control 은 정의된 간격으로 지정된 컴퓨터에서 STOP 서명 파일을 가져옵니다.

3. 서명 파일을 업데이트할 간격으로 *STOPUpdateInterval* 레지스트리 항목을 설정합니다.

   CA Access Control 은 지정된 간격으로 지정된 컴퓨터에서 서명 파일을 가져옵니다.

4. (선택 사항) 다음 키에 있는 레지스트리 항목을 사용하여 STOP 구성을 조정합니다.

   ```
   HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\STOP
   ```

   참고: STOP 레지스트리 설정에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

5. 다음 명령을 입력합니다.

   ```
   seosd -start
   ```

   CA Access Control 이 시작됩니다.

참고: eACSigUpdate 유틸리티를 사용하면 어떤 호스트에서든 서명 파일을 가져올 수 있습니다. 이 유틸리티에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

제 12 장: 설정 구성

CA Access Control 에서는 CA Access Control 끝점 구성 설정을 원격으로 관리할 수 있습니다. 이렇게 하려면 CA Access Control 끝점 관리 또는 selang 구성 환경을 사용합니다.

이 섹션은 다음 항목을 포함하고 있습니다.

구성 설정

CA Access Control 에서는 사용되는 끝점 및 정책 모델 구성 설정을 다음 위치에 저장합니다.

- Windows 컴퓨터의 경우 Windows 레지스트리
- UNIX 컴퓨터의 경우 초기화 파일(.ini)

참고: 작성할 수 있는 구성 설정과 그 의미에 대한 자세한 내용은 *참조 안내서*를 참조하십시오.

구성 설정 변경

CA Access Control 및 정책 모델이 작동하는 방식을 변경하려면 구성 설정을 변경해야 합니다.

구성 설정을 변경하려면

1. CA Access Control 끝점 관리에서 다음을 수행하십시오.
   a. "구성"을 클릭합니다.
   b. "원격 구성"을 클릭합니다.

   "원격 구성" 페이지가 나타납니다.
2. "원격 구성" 섹션 페이지의 왼쪽에서 필요한 경우 구성 트리를 확장하여 수정할 구성 설정이 있는 섹션을 표시한 다음 해당 섹션을 선택합니다.

   "섹션: *sectionName* 시스템 토큰" 페이지가 나타나고 모든 구성 설정이 표시됩니다.
3. 필요한 구성 설정을 찾아서 편집한 다음 "토큰 저장"을 클릭합니다.

   변경된 구성 설정이 저장됩니다.

감사 구성 설정 변경

CA Access Control 이 감사 레코드를 생성하고 저장하는 방법에 수정하려면 감사 구성 파일의 설정을 변경해야 합니다. selang 명령을 사용하여 감사 구성 파일의 설정을 변경할 수 있습니다.

감사 구성 설정을 변경하려면

1. (선택 사항) selang 을 사용하여 원격 호스트에 연결하려는 경우 다음 명령으로 호스트에 연결합니다.

   ```
   host host_name
   ```
2. 다음 명령을 사용하여 구성 환경으로 이동합니다.

   ```
   env config
   ```
3. editres 구성 명령을 사용하여 구성 설정을 필요에 따라 수정합니다.

   감사 구성 설정이 변경됩니다.

예: 감사 구성 파일 수정

다음 예제는 감사 구성 파일에 한 줄을 추가합니다.

```
er CONFIG audit.cfg line+("FILE;*;Administrator;*;R;P")
```