Oracle® Solaris 11.2의 사용자 및 프로세스 보안

부품 번호: E53955
2014년 7월

ORACLE®

목차

코드 예

이 설명서 사용

- **개요** - 사용자에게 추가 권한을 지정하고, 역할을 만들고 사용하며, Oracle Solaris 시스템의 프로그램 및 특정 리소스에 대한 권한을 지정하는 방법을 설명합니다.
- **대상** - 보안 관리자
- **필요한 지식** - 사이트 보안 요구 사항

제품 설명서 라이브러리

이 제품에 대한 최신 정보 및 알려진 문제는 설명서 라이브러리(http://www.oracle.com/pls/topic/lookup?ctx=E56343)에서 확인할 수 있습니다.

Oracle 지원 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info를 참조하거나, 청각 장애가 있는 경우 http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs를 방문하십시오.

피드백

http://www.oracle.com/goto/docfeedback에서 이 설명서에 대한 피드백을 보낼 수 있습니다.

♦ ♦ ♦ 1 장

1 권한을 사용하여 사용자 및 프로세스 제어 정보

Oracle Solaris에서는 사용자, 역할, 프로세스 및 선택한 리소스에 지정될 수 있는 권한을 제공합니다. 이러한 권한은 수퍼 유저 모델보다 더 안전한 관리 방법을 제공합니다.

이 장에서는 사용자 및 프로세스 권한(right) 관리를 지원하는 요소에 대한 정보를 제공하고 사용자 권한(right)을 확장하고, 사용자 권한(right)을 제한하고, 명령에 권한(privilege)을 추가하고, 필요한 권한(privilege)으로만 응용 프로그램을 제한하는 방법을 설명합니다.

Oracle Solaris 11.2 권한의 새로운 기능

이 절에서는 기존 고객을 위해 사용자 권한(right)(RBAC(역할 기반 액세스 제어)라고도 함) 및 프로세스 권한(권한(privilege)이라고도 함)의 새로운 중요 기능에 대한 정보를 강조합니다.

- 관리자가 인증된 권한 프로파일로 지정한 권한 프로파일에서는 사용자가 권한 있는 명령을 실행하기 전에 암호를 제공하도록 강제합니다. 사용자가 암호를 제공하지 않으면 명령이 권한 없이 실행됩니다. 암호는 구성 가능한 기간 동안 유효한 상태로 유지됩니다. 예 3-11. "사용자가 DHCP를 관리하기 전에 암호를 입력하도록 요구"을 참조하십시오.

  `policy.conf` 파일의 `AUTH_PROFS_GRANTED` 키워드 값으로 프로파일을 추가하면 시스템에 로그인하는 모든 사용자에게 인증된 권한 프로파일을 지정할 수 있습니다.
- `access_times` 및 `access_tz` 권한을 지정하여 시간 및 시간대별로 호스트에 대한 사용자 및 그룹 액세스를 제한할 수 있습니다. 예는 `user_attr`(4) 매뉴얼 페이지를 참조하십시오.
- Oracle Solaris에서는 `armor` 패키지에 표준화된 역할의 ARMOR(Authorization Roles Managed on RBAC) 세트를 제공합니다. 자세한 내용은 "사용자 및 프로세스 권한은 수퍼 유저 모델의 대안을 제공함" [14] 및 예 3-1. "ARMOR 역할 사용"을 참조하십시오.
- 사용자 관리자 GUI는 사용자 및 역할의 권한을 관리하는 데 사용할 수 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 계정 및 사용자 환경 관리 "의 3 장, "User Manager GUI를 사용하여 사용자 계정 관리"를 참조하십시오.

사용자 권한 관리

사용자 권한 관리는 보통 root 역할로 제한되는 작업에 대한 사용자 액세스를 제어하기 위한 보안 기능입니다. 프로세스 및 사용자에 보안 속성, 즉 권한*(right)*을 적용하여 사이트에서 여러 관리자 간에 수퍼 유저 권한(privilege)을 나눌 수 있습니다. 프로세스 권한 관리는 권한을 통해 구현됩니다. 사용자 권한 관리는 사용자나 역할에 지정되는 권한을 수집하는 권한 프로파일을 통해 구현됩니다. 키오스크, 게스트 사용자 등의 사용자 권한을 제한할 수도 있습니다.

- 커널 프로세스의 권한에 대한 자세한 내용은 "프로세스 권한 관리" [22]를 참조하십시오.
- 권한 관리 절차는 3장. Oracle Solaris에서 권한 지정을 참조하십시오.
- 참조 정보는 8장. Oracle Solaris 권한에 대한 참조를 참조하십시오.

사용자 및 프로세스 권한은 수퍼 유저 모델의 대안을 제공함

전통적인 UNIX 시스템에서 root 사용자는 수퍼유저라고도 하며 전권을 갖습니다. 많은 setuid 프로그램과 마찬가지로 root로 실행되는 프로그램도 전권을 갖습니다. root 사용자는 모든 파일을 읽거나 쓰고, 모든 프로그램을 실행하며, 모든 프로세스에 종료 신호를 보낼 수 있습니다. 실질적으로, 수퍼유저가 될 수 있는 사람이면 누구나 사이트의 방화벽을 수정하고, 감사 증적을 변경하고, 기밀 레코드를 읽고, 전체 네트워크를 종료할 수 있습니다. setuid root 프로그램을 하이재킹할 경우 시스템에서 모든 작업을 수행할 수 있습니다.

사용자, 리소스 및 프로세스에 권한 지정은 all-or-nothing 수퍼 유저 모델보다 강력한 대안을 제공합니다. 권한을 사용하면 더욱 세분화된 레벨에서 보안 정책을 적용할 수 있습니다. 권한(right)은 최소 권한*(privilege)*의 보안 원칙을 따릅니다. 최소 권한이란 사용자가 정확하게 작업 수행에 필요한 privilege(권한)만 할당받는 것을 의미합니다. 일반 사용자 권한으로 응용 프로그램 사용, 작업 상태 확인, 파일 인쇄, 새 파일 만들기 등을 충분히 수행할 수 있습니다. 일반 사용자 권한 밖의 권한은 권한 프로파일로 그룹화됩니다. 수퍼 유저 권한이 필요한 작업을 수행해야 하는 사용자에게 권한 프로파일을 지정할 수 있습니다.

프로파일로 그룹화된 권한을 사용자에게 직접 지정할 수 있습니다. 역할이라는 특수 계정을 만들어 간접적으로 지정할 수도 있습니다. 그러면 사용자가 관리 권한(privilege)이 필요한 작업을 수행하는 역할을 맡을 수 있습니다. Oracle Solaris에서는 미리 정의된 많은 권한(right) 프로파일을 제공합니다. 역할을 만들고 프로파일을 지정합니다.

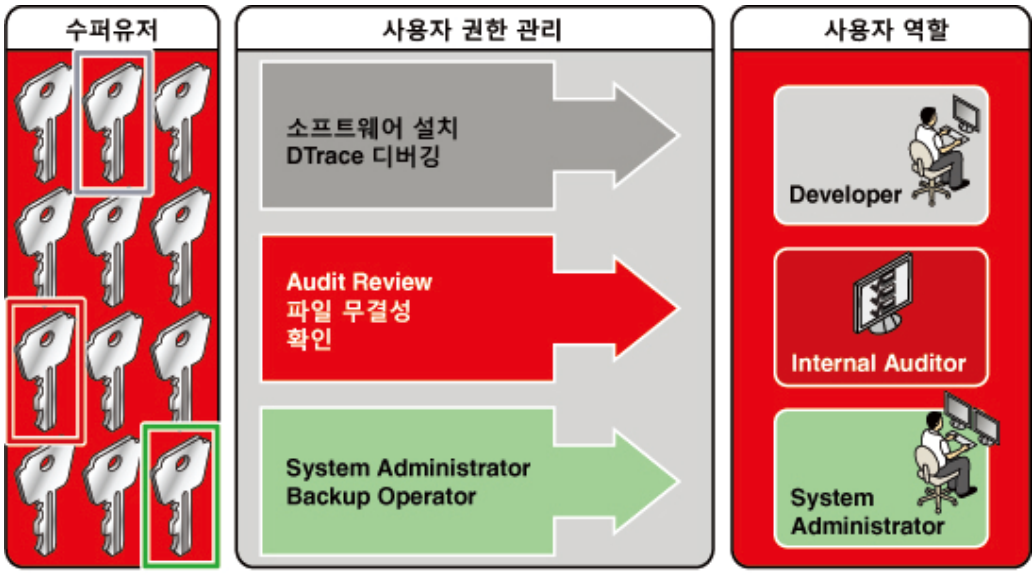
ARMOR 패키지는 표준화된 역할 세트를 제공합니다. 이 패키지를 자동으로 설치하고 사용자에게 역할을 지정하면 부트 시 책임 구분을 제공하는 시스템을 만들 수 있습니다. 자세한 내용은 Authorization Rules Managed On RBAC (ARMOR), "선택한 권한 모델 따르기" [38] 및 예 3-1. "ARMOR 역할 사용"을 참조하십시오.

권한 프로파일은 광범위한 관리 권한을 제공할 수 있습니다. 예를 들어, System Administrator 권한 프로파일을 통해 프린터 관리, cron 작업 관리 등 보안과 관련이 없는

작업을 수행할 수 있습니다. 권한 프로파일을 좁게 정의할 수도 있습니다. 예를 들어, Cron Management 권한 프로파일은 at 및 cron 작업을 관리합니다. 역할을 만들 때 역할에 광범위한 관리 권한이나 제한적인 권한을 지정할 수 있습니다.

다음 그림은 Oracle Solaris에서 역할을 만들어 trusted users(신뢰할 수 있는 사용자)에 권한을 분배할 수 있는 방법을 보여줍니다. 수퍼 유저는 신뢰할 수 있는 사용자에게 직접 권한 프로파일을 지정하여 권한을 분배할 수도 있습니다.

그림 1-1 권한 분배



그림에 표시된 권한 모델에서 수퍼 유저는 3개 역할을 만듭니다. 역할은 권한 프로파일을 기반으로 합니다. 그런 다음 수퍼유저가 작업을 수행하도록 신뢰된 사용자에게 역할을 지정합니다. 사용자가 사용자 이름으로 로그인합니다. 로그인 후에 사용자는 관리 명령 및 GUI(그래픽 사용자 인터페이스) 도구를 실행할 수 있는 역할을 맡습니다.

역할 설정의 유연성 덕분에 다양한 보안 정책이 가능합니다. Oracle Solaris와 함께 제공되는 역할은 몇 개 없지만 역할이 쉽게 구성됩니다. 예 3-1. “ARMOR 역할 사용”에서는 ARMOR 표준을 기반으로 하는 역할을 사용하는 방법을 보여줍니다. ARMOR 역할 외에 또는 ARMOR 역할 대신 Oracle Solaris에서 제공하는 권한 프로파일을 기준으로 고유한 역할을 만들 수 있습니다.

- **root** - `root` 사용자와 같은 강력한 역할입니다. 그러나 모든 역할과 마찬가지로 `root` 역할은 로그인할 수 없습니다. 일반 사용자가 로그인한 후에 지정된 `root` 역할을 맡아야 합니다. 이 역할은 기본적으로 초기 사용자에게 구성 및 지정됩니다.
- **System Administrator** - 보안과 관련이 없는 관리를 위한 덜 강력한 역할입니다. 이 역할은 파일 시스템, 메일 및 소프트웨어 설치를 관리할 수 있습니다. 그러나 이 역할은 암호를 설정할 수 없습니다.
- **Operator** - 백업, 프린터 관리 등의 작업을 위한 하급 관리자 역할입니다.

참고 - Media Backup 권한 프로파일은 전체 루트 파일 시스템에 액세스할 수 있습니다. 따라서 Media Backup 및 Operator 권한 프로파일은 하급 관리자용이긴 하지만 신뢰할 수 있는 사용자인지 확인해야 합니다.

하나 이상의 보안 역할을 구성하고 싶을 수 있습니다. Information Security, User Security, Zone Security라는 세 개의 권한 프로파일과 그 보충 프로파일이 보안을 처리합니다. Network Security는 Information Security 권한 프로파일의 보충 프로파일입니다.

역할을 구현할 필요는 없습니다. 역할은 조직의 보안 요구와 상관 관계가 있습니다. 하나의 전략은 보안, 네트워킹, 방화벽 관리와 같은 분야에 특수 목적의 관리자용 역할을 설정하는 것입니다. 또 다른 전략은 단일의 강력한 관리자 역할을 고급 사용자 역할과 함께 만드는 것입니다. 고급 사용자 역할은 고유 시스템의 일부를 수정하도록 허가된 사용자입니다. 사용자에게 직접 권한 프로파일을 지정하고 역할을 만들지 않을 수도 있습니다.

수퍼 유저 모델과 권한 모델이 공존할 수 있습니다. 다음 표에는 수퍼 유저부터 제한된 일반 사용자까지 권한 모델에서 사용 가능한 단계적 등급이 요약되어 있습니다. 양쪽 모델에서 추적할 수 있는 관리 작업이 포함됩니다. 프로세스 권한(right), 즉 권한*(privilege)*의 영향 요약은 표 1-2. “권한 있는 시스템과 권한 없는 시스템 사이의 눈에 띄는 차이점”를 참조하십시오.

표 1-1 수퍼 유저 모델과 권한 모델 비교

시스템에서 사용자 능력	수퍼 유저 모델	권한 모델
전체 수퍼 유저 권한을 가진 수퍼 유저로 전환할 수 있음	실행 가능	실행 가능
전체 사용자 권한을 가진 사용자로 로그인할 수 있음	실행 가능	실행 가능
제한된 권한을 가진 수퍼 유저로 전환할 수 있음	실행 불가능	실행 가능
사용자로 로그인할 수 있고, 때때로 수퍼 유저 권한을 가질 수 있음	실행 가능, `setuid root` 프로그램만 사용	실행 가능, `setuid root` 프로그램 및 권한 사용
관리 권한(right)을 가졌으나 전체 수퍼 유저 권한(privilege)은 없는 사용자로 로그인할 수 있음	실행 불가능	실행 가능, 권한(right) 프로파일, 역할, 직접 지정된 권한(privilege) 및 권한 부여 사용
일반 사용자보다 적은 권한을 가진 사용자로 로그인할 수 있음	실행 불가능	실행 가능, 권한 제거
수퍼유저 작업을 추적할 수 있음	실행 가능, `su` 명령 감사	실행 가능, `pfexec()` 호출 감사

시스템에서 사용자 능력	수퍼 유저 모델	권한 모델
		또한 root 역할을 맡은 사용자의 이름이 감사 증적에 있음

사용자 및 프로세스 권한의 기본 사항

권한*(privilege)* 없음 또는 권한*(right)* 없이 용어는 Oracle Solaris에서 적용되지 않습니다. 일반 사용자 프로세스를 포함하여 Oracle Solaris의 모든 프로세스에는 최소한 일부 권한(privilege)이나 권한 부여 등의 다른 사용자 권한(right)이 있습니다. Oracle Solaris에서 모든 UNIX 프로세스에 부여하는 기본 권한(privilege) 세트에 대한 자세한 내용은 “프로세스 권한 관리” [22]를 참조하십시오.

다음 요소는 Oracle Solaris에서 사용자 권한을 적용합니다. 허용적 보안 정책이나 제한적 보안 정책을 적용하도록 이러한 권한을 구성할 수 있습니다.

- **권한 부여** - 사용자나 역할이 추가 권한을 필요로 하는 작업 클래스를 수행할 수 있게 하는 권한입니다. 예를 들어, 기본 보안 정책은 콘솔 사용자에게 `solaris.device.cdrw` 권한 부여를 제공합니다. 이 권한 부여를 통해 사용자는 CD-ROM 장치를 읽고 쓸 수 있습니다. 권한 부여 목록은 `auths list` 명령을 사용합니다. 권한 부여는 커널이 아니라 사용자 응용 프로그램 레벨에서 적용됩니다. “사용자 권한 부여에 대한 추가 정보” [20]를 참조하십시오.
- **권한(privilege)** - 명령, 사용자, 역할 또는 특정 리소스(예: 포트, SMF 메소드)에 부여할 수 있는 권한(right)입니다. 권한은 커널에서 구현됩니다. 예를 들어, `proc_exec` 권한을 통해 프로세스가 `execve()`를 호출할 수 있습니다. 일반 사용자는 기본 권한을 갖습니다. 기본 권한을 보려면 `ppriv -vl basic` 명령을 실행합니다. 자세한 내용은 “프로세스 권한 관리” [22]를 참조하십시오.
- **보안 속성** - 프로세스가 작업 또는 권한 구현을 수행할 수 있게 하는 속성입니다. 전형적인 UNIX 환경에서 보안 속성을 통해 프로세스가 일반 사용자에게 금지된 작업을 수행할 수 있습니다. 예를 들어, `setuid` 및 `setgid` 프로그램에는 보안 속성이 있습니다. 권한(right) 모델에서 권한 부여 및 권한(privilege)은 `setuid` 및 `setgid` 프로그램과 더불어 보안 속성입니다. 이러한 속성 또는 권한을 사용자에게 지정할 수 있습니다. 예를 들어, `solaris.device.allocate` 권한이 부여된 사용자는 장치에 배타적 사용을 할당할 수 있습니다. 권한을 프로세스에 둘 수 있습니다. 예를 들어, `file_flag_set` 권한을 가진 프로세스는 immutable, no-unlink, append-only 파일 속성을 설정할 수 있습니다.

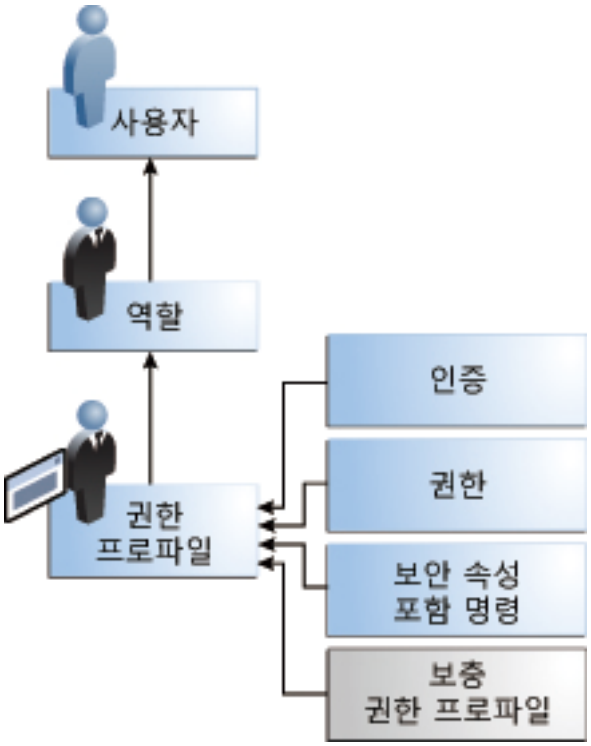
  보안 속성은 권한을 제한할 수도 있습니다. 예를 들어, `access_times` 및 `access_tz` 보안 속성은 보안과 관련된 특정 작업이 허용되는 요일과 시간, 그리고 선택적으로 시간대를 설정합니다. 이러한 키워드가 포함된 인증된 권한 프로파일을 지정하거나 직접 사용자를 제한할 수 있습니다. 자세한 내용은 `user_attr(4)` 매뉴얼 페이지를 참조하십시오.
- **권한(privilege) 있는 응용 프로그램** - 권한(right)을 검사하여 시스템 컨트롤을 대체할 수 있는 응용 프로그램 또는 명령입니다. 자세한 내용은 “권한을 검사하는 응용 프로그램” [33] 및 “Developer’s Guide to Oracle Solaris 11 Security ”를 참조하십시오.
- **권한 프로파일** - 역할이나 사용자에게 지정할 수 있는 권한 모음입니다. 권한(right) 프로파일에는 권한 부여, 직접 지정된 권한(privilege), 보안 속성 포함 명령 및 기타 권한

(right) 프로파일이 포함될 수 있습니다. 다른 프로파일 내의 프로파일을 보충 권한 프로파일이라고 합니다. 권한 프로파일은 권한을 그룹화하는 편리한 방법입니다. 역할이라는 특수 계정이나 사용자에게 직접 지정할 수 있습니다. 프로세스에서 권한을 인식하는 경우에만 권한 프로파일의 명령을 사용할 수 있습니다. 암호를 제공해야 할 수도 있습니다. 또는 기본적으로 암호 인증을 제공할 수 있습니다. "권한 프로파일에 대한 추가 정보" [20]를 참조하십시오.

- **역할** - 권한 있는 응용 프로그램을 실행하기 위한 특수 ID입니다. 특수한 신원은 지정된 사용자만 맡을 수 있습니다. 역할에 의해 실행되는 시스템에서는 초기 구성 후에 수퍼 유저가 필요 없을 수 있습니다. "역할에 대한 추가 정보" [21]를 참조하십시오.

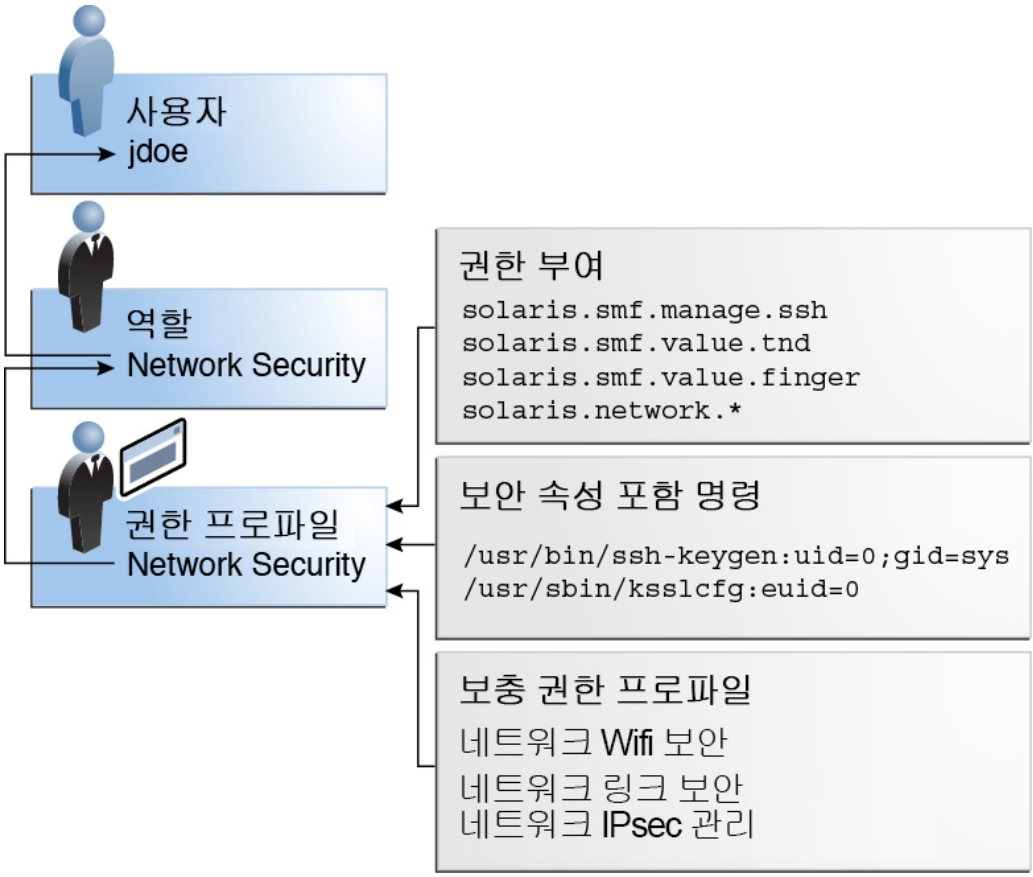
다음 그림은 사용자 권한 및 프로세스 권한이 함께 작동하는 방식을 보여줍니다.

그림 1-2 사용자 권한 및 프로세스 권한 연동



다음 그림은 Network Security 역할과 Network Security 권한 프로파일을 사용하여 지정된 권한의 작동 방식을 보여줍니다.

그림 1-3 사용자 권한 및 프로세스 권한 지정 예



Network Security 역할은 IPsec, Wifi 및 네트워크 링크를 관리하는 데 사용됩니다. 역할이 사용자 `jdoe`에 지정됩니다. `jdoe`가 해당 역할로 전환한 후 역할 암호를 제공하면 역할을 맡을 수 있습니다. 관리자는 역할이 역할 암호가 아니라 사용자 암호로 인증하도록 설정할 수 있습니다.

그림에서는 Network Security 권한 프로파일이 Network Security 역할에 지정됩니다. Network Security 권한 프로파일은 Network Wifi Security, Network Link Security, Network IPsec Management 순으로 평가되는 보충 프로파일을 포함합니다. 이러한 보충 프로파일에는 역할의 기본 작업을 완료하는 권한이 포함되어 있습니다.

Network Security 권한(right) 프로파일에는 직접 지정된 권한 부여 3개, 직접 지정된 권한(privilege) 0개, 보안 속성 포함 명령 2개가 있습니다. 보충 권한 프로파일에는 직접 지정된 권한 부여가 있고, 이들 중 2개에는 보안 속성 포함 명령이 있습니다.

jdoe가 Network Security 역할을 맡고 있는 경우 셸이 profile shell(프로파일 셸)로 변경됩니다. 프로파일 셸 프로세스에서 권한 사용을 평가할 수 있으므로 jdoe가 네트워크 보안을 관리할 수 있습니다.

사용자 권한에 대한 추가 정보

이 절에서는 사용자 레벨의 구현 및 권한 사용에 대한 세부 정보를 제공합니다.

사용자 권한 부여에 대한 추가 정보

권한 부여는 역할, 프로그램, 영역 또는 사용자에게 부여할 수 있는 권한입니다. 권한 부여는 사용자 응용 프로그램 레벨에서 정책을 시행합니다. 권한(privilege)과 마찬가지로 권한 부여를 잘못 지정하면 원래 의도한 것보다 많은 권한(right)이 부여될 수 있습니다. 자세한 내용은 “권한(privilege) 에스컬레이션 및 사용자 권한(right)” [30]을 참조하십시오.

권한 부여와 권한의 차이점은 보안 정책을 시행하는 레벨에 있습니다. 적절한 권한 없이 프로세스는 커널을 통해 권한 있는 작업을 수행하는 것을 금지할 수 있습니다. 적절한 권한 부여가 없을 경우 사용자가 privileged application(권한 있는 응용 프로그램)을 사용하거나 권한 있는 응응 프로그램 내에서 보안이 중요한 작업을 수행하지 못하도록 금지될 수 있습니다. 권한(privilege)에 대한 자세한 내용은 “프로세스 권한 관리” [22]를 참조하십시오.

권한 준수 응용 프로그램은 응용 프로그램이나 응용 프로그 내의 특정 작업에 대한 액세스를 부여하기 전에 사용자의 권한 부여를 검사할 수 있습니다. 이 검사는 UID=0에 대한 기존 UNIX 응용 프로그램의 검사를 대체합니다.

권한 부여에 대한 자세한 내용은 다음 절을 참조하십시오.

- “권한 부여 참조” [104]
- “auth_attr 데이터베이스” [108]
- “권한 부여가 필요한 선택된 명령” [110]

권한 프로파일에 대한 추가 정보

권한 프로파일은 관리 권한이 필요한 작업을 수행하기 위해 역할이나 사용자에게 지정할 수 있는 권한 모음입니다. 권한 프로파일에는 권한 부여, 권한, 지정된 보안 속성 포함 명령 및 기타 권한 프로파일이 포함될 수 있습니다. 권한(right) 프로파일에 초기 상속 가능한 권한(privilege) 세트를 감소/확장하거나 제한 세트를 감소하는 항목이 포함될 수도 있습니다.

인증된 권한 프로파일은 사용자가 암호를 제공하거나 다시 인증해야 하는 권한 프로파일입니다. 관리자는 다시 인증하지 않고 사용할 수 있는 프로파일을 결정합니다. 다시 인증할 필

요가 없는 프로파일의 예로 Basic Solaris User 권한 프로파일이 있습니다. 사이트 보안 요구 사항에 따라 보안이 중요한 작업에 대한 권한 프로파일의 경우 다시 인증이 필요할 수도 있습니다.

권한 프로파일에 대한 참조 정보는 다음 절을 참조하십시오.

- "권한 프로파일 참조" [103]
- "prof_attr 데이터베이스" [108]
- "exec_attr 데이터베이스" [108]

역할에 대한 추가 정보

역할은 권한 있는 응용 프로그램을 실행할 수 있는 특수한 유형의 사용자 계정입니다. 역할은 사용자 계정과 동일한 방법으로 만듭니다. 역할에는 홈 디렉토리, 그룹 지정, 암호 등이 있습니다. 권한 프로파일 및 권한 부여는 역할에 관리 권한을 제공합니다. 역할은 다른 권한이나 역할을 맡은 사용자로부터 권한을 상속받을 수 없습니다. 역할은 수퍼 유저 권한을 분배하므로 보다 안전한 관리 방식이 가능합니다.

역할을 여러 사용자에 지정할 수 있습니다. 동일한 역할을 맡은 모든 사용자는 동일한 역할 홈 디렉토리를 사용하고, 동일한 환경에서 작동하며, 동일한 파일에 액세스할 수 있습니다. 사용자는 명령줄에서 `su` 명령을 실행하고 역할 이름과 역할 암호를 제공하여 역할을 맡을 수 있습니다. 관리자는 사용자의 암호를 제공하여 인증되도록 시스템을 구성할 수 있습니다. 예 3-16. "사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정"을 참조하십시오.

역할은 직접 로그인할 수 없습니다. 사용자가 로그인 후에 역할을 맡습니다. 역할을 맡고 나면 현재 역할을 끝내기 전까지 다른 역할을 맡을 수 없습니다.

또한 권한 프로파일은 사용자 환경에 권한을 추가하는 반면, 역할은 해당 역할을 맡을 수 있는 다른 사용자와 공유되는 클린 실행 환경을 사용자에게 제공합니다. 사용자가 역할로 전환하면 사용자의 권한 부여 또는 권한 프로파일이 역할에 적용되지 않습니다.

`passwd`, `shadow` 및 `user_attr` 데이터베이스는 정적 역할 정보를 저장합니다. 역할의 작업을 감사할 수 있고 감사해야 합니다.

역할 설정에 대한 자세한 내용은 다음 절을 참조하십시오.

- "선택한 권한 모델 따르기" [38]
- "사용자에게 권한 지정" [41]

`root`는 Oracle Solaris의 역할이므로 익명의 `root` 로그인을 금지합니다. 프로파일 셸 명령 `pfexec`를 감사하는 경우 감사 추적에 로그인 사용자의 실제 UID, 사용자가 맡은 모든 역할 및 수행된 권한 있는 작업이 포함됩니다. 시스템에서 권한 있는 작업을 감사하려면 "관리 작업 감사" [77]를 참조하십시오.

프로세스 권한 관리

Oracle Solaris의 프로세스 권한(right) 관리는 권한*(privilege)*을 통해 구현됩니다. 권한을 사용하면 명령, 사용자, 역할 및 특정 시스템 리소스 레벨에서 프로세스를 제한할 수 있습니다. 권한을 사용하면 시스템에서 전체 수퍼 유저 권한을 보유한 하나의 사용자나 프로세스와 연관된 보안 위험을 줄일 수 있습니다. 프로세스 권한 및 사용자 권한은 기존 수퍼 유저 모델의 강력한 대체 모델을 제공합니다.

일반적으로 권한(privilege)은 권한(right)을 추가하는 데 사용됩니다. 그러나 권한(privilege)을 사용하여 권한(right)을 제한할 수도 있습니다. 예를 들어, `setuid root` 프로그램을 권한(privilege) 인식 프로그램으로 변경합니다. 또한 확장 권한 정책을 사용하면 관리자가 파일 객체, 사용자 ID 또는 포트에 지정된 권한만 사용하도록 허용할 수 있습니다. 이 세분화된 권한 지정은 이러한 리소스에 대한 기본 권한을 제외하고 다른 모든 권한을 거부합니다.

- 확장 권한 정책 및 제한적 권한에 대한 자세한 내용은 "확장 권한 정책을 사용하여 권한 사용 제한" [30]을 참조하십시오.
- 사용자 권한에 대한 자세한 내용은 "사용자 권한 관리" [14]를 참조하십시오.
- 권한 관리 방법에 대한 자세한 내용은 3장. Oracle Solaris에서 권한 지정을 참조하십시오.
- 권한에 대한 참조 정보는 "권한 참조" [111]를 참조하십시오.

권한으로 커널 프로세스 보호

권한(privilege)은 프로세스에서 작업을 수행하는 데 필요한 권한(right)입니다. 권한은 커널에서 시행됩니다. 권한의 기본 세트의 한도 내에서 운영되는 프로그램은 시스템 보안 정책의 한도 내에서 운영됩니다. 시스템 보안 정책의 한도 밖에서 운영되는 프로그램의 예로 `setuid root` 프로그램이 있습니다. 권한을 사용할 경우 프로그램이 `setuid root`를 호출할 필요가 없습니다.

권한은 시스템에서 가능한 작업 종류를 열거합니다. 정확히 프로그램 성공을 위해 필요한 권한만으로 프로그램을 실행할 수 있습니다. 예를 들어, 파일을 조작하는 프로그램에 `file_dac_write` 및 `file_flag_set` 권한이 필요할 수 있습니다. 프로세스에 대한 이러한 권한으로 인해 프로그램을 `root`로 실행할 필요가 없습니다.

이전에는 시스템이 "사용자 및 프로세스 권한의 기본 사항" [17]에 소개된 대로 privilege model(권한 모델) 또는 권한(right) 모델을 따르지 않았습니다. 오히려 수퍼유저 모델을 사용했습니다. 수퍼 유저 모델에서는 프로세스가 `root` 또는 사용자로 실행되었습니다. 사용자 프로세스는 사용자의 디렉토리와 파일에서 작동하도록 제한되었습니다. `root` 프로세스는 시스템의 어디든지 디렉토리 및 파일을 만들 수 있습니다. 사용자의 디렉토리 밖에 디렉토리를 만들어야 하는 프로세스는 `UID=0`, 즉 `root`로 실행됩니다. 시스템 파일을 보호하

기 위해 보안 정책이 DAC(임의 액세스 제어)에 의존했습니다. 장치 노드가 DAC로 보호되었습니다. 예를 들어, sys 그룹이 소유한 장치는 해당 그룹의 구성원만 열 수 있었습니다.

그러나 setuid 프로그램, 파일 사용 권한 및 관리 계정은 오용되기 쉽습니다. setuid 프로세스가 허가한 동작이 작업 완료를 위해 필요한 것보다 훨씬 많습니다. 그러면 전권의 root 사용자로 실행된 침입자에 의해 setuid root 프로그램이 손상될 수 있습니다. 마찬가지로, root 암호에 액세스할 수 있는 사용자가 전체 시스템을 손상시킬 수 있습니다.

이와 반대로, 권한(privilege)으로 정책을 적용하는 시스템은 사용자 권한(right)과 root 권한(right) 사이에 단계적 등급을 허용합니다. 일반 사용자 권한을 벗어난 작업을 수행하는 권한을 사용자에게 부여할 수 있고, root가 현재 보유한 권한보다 적은 권한으로 root를 제한할 수 있습니다. 권한(right)을 사용하여 권한(privilege)으로 실행되는 명령을 권한 프로파일에 격리하고 하나의 사용자나 역할에 지정할 수 있습니다. 표 1-1. “수퍼 유저 모델과 권한 모델 비교”에는 권한(right) 모델이 제공하는 사용자 권한(right)과 root 권한(privilege) 사이의 단계적 등급이 요약되어 있습니다.

권한(right) 모델이 수퍼 유저 모델보다 훨씬 안전합니다. 프로세스에서 제거된 권한은 악용될 수 없습니다. 액세스에 악용될 수 있는 DAC 보호와 달리 프로세스 권한(privilege)은 중요한 파일 및 장치에 추가 보호 조치를 제공할 수 있습니다.

그런 다음 권한(privilege)은 프로그램 및 프로세스를 프로그램에 필요한 권한(right)만으로 제한할 수 있습니다. 최소 권한을 구현하는 시스템에서는 프로세스를 탈취하는 침입자가 프로세스가 가진 권한에만 액세스할 수 있습니다. 시스템의 나머지는 손상될 수 없습니다.

권한 설명

권한은 권한 영역을 기준으로 논리적으로 그룹화됩니다.

- **FILE 권한** - file 문자열로 시작하는 권한은 파일 시스템 객체에서 작동합니다. 예를 들어, file_dac_write 권한은 파일에 쓰는 동안 모든 액세스 제어를 대체합니다.
- **IPC 권한** - ipc 문자열로 시작하는 권한은 IPC 객체 액세스 제어를 대체합니다. 예를 들어, ipc_dac_read 권한을 통해 프로세스가 DAC로 보호된 원격 공유 메모리를 읽을 수 있습니다.
- **NET 권한** - net 문자열로 시작하는 권한은 특정 네트워크 기능에 대한 액세스를 제공합니다. 예를 들어, net_rawaccess 권한을 통해 장치가 네트워크에 연결할 수 있습니다.
- **PROC 권한** - proc 문자열로 시작하는 권한을 통해 프로세스 자체의 제한된 등록 정보를 수정할 수 있습니다. PROC 권한은 매우 제한된 효과를 가진 권한입니다. 예를 들어, proc_clock_highres 권한을 통해 프로세스가 고해상도 타이머를 사용할 수 있습니다.
- **SYS 권한** - sys 문자열로 시작하는 권한은 다양한 시스템 등록 정보에 대한 무제한 액세스를 프로세스에 제공합니다. 예를 들어, sys_linkdir 권한을 통해 프로세스가 디렉토리에 대한 하드 링크를 만들고 중단할 수 있습니다.

기타 논리적 그룹에는 CONTRACT, CPC, DTRACE, GRAPHICS, VIRT, WIN 등이 있습니다.

일부 권한은 시스템에 제한된 효과를 미치고, 일부는 광범위한 효과를 미칩니다. proc_taskid 권한의 정의는 제한된 효과를 나타냅니다.

```
proc_taskid
        Allows a process to assign a new task ID to the calling process.
```

net_rawaccess 권한의 정의는 광범위한 효과를 나타냅니다.

```
net_rawaccess
        Allows a process to have direct access to the network layer.
```

privileges(5) 매뉴얼 페이지는 모든 권한에 대해 설명합니다. “권한 목록” [91]도 참조하십시오.

권한 있는 시스템의 관리상 차이점

권한이 있는 시스템과 권한이 없는 시스템 간에는 몇 가지 눈에 띄는 차이점이 있습니다. 다음 표는 일부 차이점을 나열합니다.

표 1-2 권한 있는 시스템과 권한 없는 시스템 사이의 눈에 띄는 차이점

기능	권한 없음	권한
데몬	데몬이 root로 실행됩니다.	데몬이 사용자 daemon으로 실행됩니다. 예를 들어, 제한된 권한이 지정되고 daemon으로 실행되는 데몬에는 lockd 및 rpcbind가 있습니다.
로그 파일 소유권	root가 로그 파일을 소유합니다.	로그 파일을 만든 daemon이 로그 파일을 소유합니다. root 사용자는 파일을 소유하지 않습니다.
오류 메시지	오류 메시지가 수퍼유저를 참조합니다. 예를 들어, chroot: not superuser입니다.	오류 메시지가 권한 사용을 반영합니다. 예를 들어, chroot 실패에 해당하는 오류 메시지는 chroot: exec failed입니다.
setuid 프로그램	프로그램이 setuid root를 사용하여 일반 사용자가 수행할 수 없는 작업을 완료합니다.	많은 setuid root 프로그램은 필요한 권한만으로 실행됩니다. 예를 들어, 권한을 사용하는 명령에는 audit, ikeadm, ipadm, ipsecconf, ping, traceroute, newtask 등이 있습니다.
파일 권한	장치 사용 권한을 DAC로 제어합니다. 예를 들어, sys 그룹의 구성원이 /dev/ip를 열 수 있습니다.	파일 사용 권한(DAC)이 장치를 열 수 있는 사람을 예측하지 않습니다. 장치는 DAC 및 장치 정책으로 보호됩니다. 예를 들어, /dev/ip 파일에 666 권한(permission)이 있지만 적절한 권한(privilege)을 가진 프로세스만 장치를 열 수 있습니다.
감사 이벤트	su 명령 사용의 감사에 많은 관리 기능이 관여합니다.	권한 사용의 감사에 대부분의 관리 기능이 관여합니다. cusa 감사 클래스에는 관리 기능을 모니터링하는 감사 이벤트가 포함됩니다.
프로세스	프로세스는 프로세스 소유자 권한으로 보호됩니다.	권한으로 프로세스를 보호합니다. 프로세스 권한 및 프로세스 플래그는 /proc/<pid>/priv 디렉토리에서 새 항목으로 표시됩니다.
디버깅	코어 덤프에 권한에 대한 참조가 없습니다.	코어 덤프의 ELF 노트 섹션은 NT_PRPRIV 및 NT_PRPRIVINFO 노트에 프로세스 권한 및 플래그에 대한 정보를 포함합니다. ppriv 명령 및 기타 명령은 적절히 크기 조정된 세트의 적절한 개수를 보여줍니다. 정확히 비트 세트의 비트를 권한 이름에 매핑합니다.

권한에 대한 추가 정보

이 절에서는 권한 구현, 사용 및 지정에 대한 세부 사항을 다룹니다.

권한이 구현되는 방법

각 프로세스에는 프로세스가 특정 권한을 사용할 수 있는지 여부를 결정하는 4개의 권한 세트가 있습니다. 커널이 자동으로 권한의 유효 세트를 계산합니다. 권한의 초기 상속 가능한 세트를 수정할 수 있습니다. 권한을 사용하도록 코딩된 프로그램은 권한의 허가된 세트를 줄일 수 있습니다. 권한의 제한 세트를 축소할 수 있습니다.

- **유효 권한 세트 또는 E** - 현재 유효한 권한 세트입니다. 프로세스는 허가된 세트에 속한 권한을 유효 세트에 추가할 수 있습니다. 또한 E에서 권한을 제거할 수도 있습니다.
- **허가된 권한 세트 또는 P** - 사용할 수 있는 권한 세트입니다. 상속 또는 지정을 통해 얻은 권한을 프로그램에 사용할 수 있습니다. 실행 프로파일은 프로그램에 권한을 지정하는 하나의 방법입니다. `setuid` 명령은 `root`가 가진 모든 권한을 프로그램에 지정합니다. 허가된 세트에서 권한을 제거할 수 있지만 추가할 수는 없습니다. P에서 제거된 권한은 자동으로 E에서 제거됩니다.

  권한 인식 프로그램은 프로그램에서 사용하지 않는 권한을 프로그램의 허가된 세트에서 제거합니다. 이렇게 하면 프로그램이나 악의적 프로세스에서 불필요한 권한을 악용할 수 없습니다. privilege-aware(권한 인식) 프로그램에 대한 자세한 내용은 “Developer’s Guide to Oracle Solaris 11 Security ”의 2 장, “Developing Privileged Applications”을 참조하십시오.
- **상속 가능한 권한 세트 또는 I** - `exec` 호출을 통해 프로세스가 상속할 수 있는 권한 세트입니다. `exec` 호출 이후 상속된 권한이 허가된 세트 및 유효 세트에 배치되므로 `setuid` 프로그램의 특수한 경우를 제외하고 이러한 세트가 동일하게 설정됩니다.

  `setuid` 프로그램의 경우 `exec`의 호출 후에 상속 가능한 세트가 먼저 제한 세트로 제약됩니다. 그런 다음, 상속된 권한 세트(I)에서 제한 세트에 속한 권한(L)을 뺀 값이 해당 프로세스의 P 및 E에 지정됩니다.
- **제한 권한 세트 또는 L** - 프로세스와 해당 자식에서 사용할 수 있는 권한의 외부 제한을 정의하는 세트입니다. 기본적으로 제한 세트는 모든 권한입니다. 프로세스가 제한 세트를 축소할 수 있지만 제한 세트를 확장할 수는 없습니다. L은 I를 제한하는 데 사용됩니다. 결과적으로, L은 `exec` 시간에 P 및 E를 제한합니다.

  사용자가 권한 지정 프로그램을 포함하는 프로파일에 지정된 경우 대개 해당 프로그램을 실행할 수 있습니다. 수정되지 않은 시스템에서 프로그램의 지정된 권한은 사용자의 제한 세트 내에 있습니다. 프로그램에 지정된 권한은 사용자의 허가된 세트의 일부가 됩니다. 권한이 지정된 프로그램을 실행하려면 사용자가 profile shell(프로파일 셸)에서 프로그램을 실행해야 합니다.

커널은 기본 권한 세트를 인식합니다. 수정되지 않은 시스템에서 각 사용자의 초기 상속 가능한 세트는 로그인 시 기본 세트와 같습니다. 기본 세트를 수정할 수 없는 반면, 사용자가 기본 세트에서 상속한 권한은 수정할 수 있습니다.

수정되지 않은 시스템에서 로그인 시 사용자의 권한 세트는 다음과 비슷합니다.

```
E (Effective): basic
I (Inheritable): basic
P (Permitted): basic
L (Limit): all
```

로그인 시 모든 사용자는 상속 가능한 세트, 허가된 세트, 유효 세트에 기본 세트를 포함합니다. 사용자의 제한 세트는 (전역 또는 비전역) 영역의 기본 제한 세트와 같습니다.

사용자 또는 보다 정확하게 사용자 로그인 프로세스에 직접적으로, 권한(right) 프로파일을 통해 많은 사용자에게 간접적으로, 사용자에게 권한 있는 명령을 지정하여 간접적으로 추가 권한(privilege)을 지정할 수 있습니다. 사용자의 기본 세트에서 권한을 제거할 수도 있습니다. 절차 및 예는 3장. Oracle Solaris에서 권한 지정을 참조하십시오.

권한 사용 방법

권한은 Oracle Solaris에 기본 제공됩니다. 이 절에서는 Oracle Solaris에서 장치, 리소스 관리 및 레거시 응용 프로그램과 함께 권한을 사용하는 방법을 설명합니다.

프로세스가 권한을 얻는 방법

프로세스는 권한을 상속하거나 권한이 지정될 수 있습니다. 프로세스는 부모 프로세스에서 권한을 상속합니다. 로그인 시, 사용자의 초기 상속 가능한 권한 세트에 따라 사용자 프로세스에서 사용할 수 있는 권한이 결정됩니다. 사용자 초기 로그인의 모든 자식 프로세스는 해당 세트를 상속합니다.

프로그램, 사용자, 역할 및 특정 리소스에 직접 권한을 지정할 수도 있습니다. 프로그램에 권한이 필요할 때 권한 프로파일에서 프로그램의 실행 파일에 권한을 지정합니다. 프로그램을 실행하도록 허가된 사용자나 역할이 프로그램을 포함하는 프로파일에 지정됩니다. 로그인 시 또는 프로파일 셸을 열 때, 프로그램의 실행 파일을 프로파일 셸에 입력하면 프로그램이 권한으로 실행됩니다. 예를 들어, Object Access Management 프로파일을 포함하는 역할은 `chmod` 명령을 `file_chown` 권한으로 실행할 수 있으므로 역할이 소유하지 않은 파일의 소유권을 변경할 수 있습니다.

역할/사용자가 추가 권한이 직접 지정된 프로그램을 실행할 때 지정된 권한이 역할/사용자의 상속 가능한 세트에 추가됩니다. 권한이 지정된 프로그램의 자식 프로세스는 부모의 권한을 상속합니다. 자식 프로세스에 부모 프로세스보다 더 많은 권한이 필요한 경우 자식 프로세스에 해당 권한을 직접 지정해야 합니다.

권한을 사용하도록 코딩된 프로그램을 privilege-aware(권한 인식) 프로그램이라고 합니다. 권한 인식 프로그램은 프로그램 실행 중 권한 사용을 사용 및 사용 안함으로 설정합니다. 운용 환경에서 성공하려면 프로그램이 사용 및 사용 안함으로 설정하는 권한을 프로그램에 지

정해야 합니다. 권한 인식 프로그램을 제공하기 전에 프로그램에 필요한 권한만 실행 파일에 지정합니다. 그런 다음 프로그램을 테스트하여 프로그램이 작업 수행을 성공하는지 확인합니다. 또한 프로그램이 권한 사용을 오용하지 않는지 검사합니다.

권한 인식 코드의 예는 "Developer's Guide to Oracle Solaris 11 Security "의 2 장, "Developing Privileged Applications"를 참조하십시오. 권한이 필요한 프로그램에 권한을 지정하려면 예 4-1. "레거시 응용 프로그램에 보안 속성 지정" 및 예 5-7. "권한 있는 명령을 포함하는 권한 프로파일 만들기"을 참조하십시오.

권한 및 장치

수퍼 유저 모델에서 파일 권한(permission)만으로 보호되는 시스템 인터페이스가 권한(right) 모델에서는 권한(privilege)으로 보호됩니다. 권한 있는 시스템에서 파일 사용 권한은 인터페이스를 보호하기에 너무 약합니다. `proc_owner`와 같은 권한(privilege)은 파일 권한(permission)을 대체하고 시스템에 대한 전체 액세스 권한을 얻을 수 있습니다.

따라서 Oracle Solaris에서 장치 디렉토리의 소유권은 장치를 열기에 충분하지 않습니다. 예를 들어 `sys` 그룹의 구성원은 더 이상 자동으로 `/dev/ip` 장치를 열도록 허용되지 않습니다. `/dev/ip`의 파일 권한(permission)은 `0666`이지만, 장치를 열려면 `net_rawaccess` 권한(privilege)이 필요합니다.

장치 정책은 권한(privilege)으로 제어되기 때문에 장치를 여는 권한을 보다 유연하게 부여할 수 있습니다. 장치 정책 및 드라이버에 대한 권한 요구 사항을 적절히 구성할 수 있습니다. 장치 드라이버를 설치, 추가 또는 업데이트할 때 권한 요구 사항을 구성할 수 있습니다.

자세한 내용은 `add_drv`(1M), `devfsadm`(1M), `getdevpolicy`(1M) 및 `update_drv`(1M) 매뉴얼 페이지를 참조하십시오.

권한 및 리소스 관리

Oracle Solaris에서 `project.max-locked-memory` 및 `zone.max-locked-memory` 리소스 컨트롤을 사용하여 `PRIV_PROC_LOCK_MEMORY` 권한에 지정된 프로세스의 메모리 소비를 제한할 수 있습니다. 이 권한으로 프로세스가 물리적 메모리의 페이지를 잠글 수 있습니다.

`PRIV_PROC_LOCK_MEMORY` 권한을 권한 프로파일에 지정하면 이 권한을 가진 프로세스에 모든 메모리를 잠그는 능력을 제공할 수 있습니다. 보호 조치로, 권한 사용자가 모든 메모리를 잠그지 못하도록 리소스 컨트롤을 설정하십시오. 비전역 영역에서 실행되는 권한 있는 프로세스의 경우 `zone.max-locked-memory` 리소스 컨트롤을 설정합니다. 시스템에서 실행되는 권한 있는 프로세스의 경우 프로젝트를 만들고 `project.max-locked-memory` 리소스 컨트롤을 설정합니다. 이러한 리소스 컨트롤에 대한 자세한 내용은 "Oracle Solaris 11.2의 리소스 관리 "의 6 장, "리소스 제어 정보" 및 "Oracle Solaris 영역 소개 "의 2 장, "비전역 영역 구성 개요"를 참조하십시오.

레거시 응용 프로그램 및 권한 사용

레거시 응용 프로그램을 수용하기 위해 권한(privilege) 구현은 수퍼 유저 및 권한(right) 모델 둘 다로 작동합니다. 커널은 `PRIV_AWARE` 플래그를 자동으로 추적하여 프로그램이 권한과 연동하도록 설계되었는지 나타냅니다. 권한을 인식하지 못하는 자식 프로세스를 고려하십시오. 부모 프로세스에서 상속한 권한은 자식의 허가된 세트와 유효 세트에서 사용할 수 있습니다. 자식 프로세스가 UID를 `0`으로 설정하면 전체 수퍼 유저 권한을 얻지 못할 수도 있습니다. 프로세스의 유효 세트와 허가된 세트는 자식의 제한 세트의 권한으로 제약됩니다. 따라서 권한 인식 프로세스의 제한 세트는 권한을 인식하지 못하는 자식 프로세스의 루트 권한을 제약합니다.

권한 사용 디버그

Oracle Solaris는 권한 실패를 디버깅하는 도구를 제공합니다. `ppriv` 명령과 `truss` 명령은 디버깅 출력을 제공합니다. 예는 `ppriv`(1) 매뉴얼 페이지를 참조하십시오. 예는 “권한 문제 해결” [95]을 참조하십시오. `dtrace` 명령을 사용할 수도 있습니다. 자세한 내용은 `dtrace`(1M) 매뉴얼 페이지 및 “Oracle Solaris 11.2 Dynamic Tracing Guide ”를 참조하십시오.

권한 지정

"privilege(권한)(privilege)"이라는 용어는 일반적으로 권한(right) 증가를 나타냅니다. Oracle Solaris 시스템의 모든 프로세스는 일부 권한(right)으로 실행되므로 권한(privilege)을 제거하여 프로세스에 대한 권한(right)을 줄일 수 있습니다. 이 릴리스에서는 확장 권한 정책을 사용하여 기본적으로 특정 리소스에 제공되는 권한을 제외하고 대부분의 권한을 제거할 수도 있습니다.

사용자 및 프로세스에 권한 지정

보안 관리자의 자격으로 권한을 지정할 책임이 있습니다. 기존 권한(right) 프로파일은 프로파일의 명령에 권한(privilege)이 이미 지정되어 있습니다. 그런 다음 역할 또는 사용자에 권한 프로파일을 지정합니다.

사용자, 역할 또는 권한(right) 프로파일에 직접 권한(privilege)을 지정할 수도 있습니다. 세션 동안 책임감 있게 권한을 사용할 것으로 신뢰되는 일부 사용자에게는 권한을 직접 지정할 수 있습니다. 직접 지정의 좋은 후보는 `proc_clock_highres`와 같은 제한된 영향을 미치는 권한입니다. 직접 지정의 나쁜 후보는 `file_dac_write`와 같이 보다 광범위한 영향을 미치는 권한입니다. 자세한 내용은 “권한 지정 시 보안 고려 사항” [35]을 참조하십시오.

사용자, 역할 또는 프로세스에 권한이 거부될 수도 있습니다. 사용자나 역할의 초기 상속 가능한 세트 또는 제한 세트에서 권한을 제거할 때는 주의해야 합니다.

사용자 또는 역할의 권한 확장

사용자 및 역할은 상속 가능한 권한 세트를 갖습니다. 제한 세트는 초기에 모든 권한이므로 줄일 수만 있습니다. 상속 가능한 세트에 속하지 않는 권한을 지정하여 사용자, 역할 및 프로세스에 대해 초기 상속 가능한 세트를 확장할 수 있습니다.

3가지 방법으로 사용 가능한 권한을 확장할 수 있습니다.

- 초기 상속 가능한 세트에는 없지만 제한 세트에 포함된 권한을 사용자와 역할에 지정할 수 있습니다. 권한 프로파일의 권한 있는 명령을 통해 간접적으로 지정하거나 직접 지정할 수 있습니다.
- 스크립트나 응용 프로그램에 권한 추가 등 상속 가능한 세트에 없는 권한을 프로세스에 명시적으로 지정할 수 있습니다.
- 상속 가능한 세트에는 없지만 제한 세트에 포함된 권한은 네트워크 포트, UID 또는 파일 객체에 명시적으로 지정할 수 있습니다. 권한 사용을 확장 권한 정책이라고 하며, 사용 가능한 권한을 제한하는 수단이기도 합니다. 자세한 내용은 "확장 권한 정책을 사용하여 권한 사용 제한" [30]을 참조하십시오.

권한이 필요한 관리 작업에만 권한을 지정하는 것이 사용자 또는 역할의 권한을 확장하는 가장 정확한 방법입니다. 필요한 권한(privilege)으로 명령 또는 스크립트가 포함된 권한(right) 프로파일을 만듭니다. 그런 다음 사용자 또는 역할에 이 권한 프로파일을 지정합니다. 이러한 지정을 통해 사용자 또는 역할이 해당 권한 있는 명령을 실행할 수 있습니다. 그렇지 않으면 사용자가 권한을 사용할 수 없습니다.

사용자 또는 역할에 대한 초기 상속 가능한 권한 세트를 확장하는 것은 덜 바람직한 권한 지정 방법입니다. 상속 가능한 세트의 모든 권한은 허가된 세트와 유효 세트에 속합니다. 사용자나 역할이 셸에 입력하는 모든 명령은 직접 지정된 권한을 사용할 수 있습니다. 자세한 내용은 "권한 지정 시 보안 고려 사항" [35]을 참조하십시오.

불필요한 권한 가용성을 줄이기 위해 네트워크 포트, UID 및 파일 객체에 확장 권한을 지정할 수 있습니다. 이러한 지정은 확장 권한 지정에 없는 권한을 유효 세트에서 제거합니다. 자세한 내용은 "확장 권한 정책을 사용하여 권한 사용 제한" [30]을 참조하십시오.

사용자 또는 역할의 권한 제한

신뢰할 수 없는 사용자에게 권한(privilege) 및 권한 프로파일을 적용하여 권한(right)을 제한할 수도 있습니다. 권한을 제거하면 사용자와 역할이 특정 작업을 수행하지 못하도록 금지할 수 있습니다. 초기 상속 가능한 세트 및 제한 세트에서 권한을 제거할 수 있습니다. 초기 상속 가능한 세트 또는 제한 세트가 기본 세트보다 작은 경우 세트를 분배하기 전에 권한 제거를 주의 깊게 테스트해야 합니다. 초기 상속 가능한 세트에서 권한을 제거하면 사용자의 로그인을 막을 수 있습니다. 제한 세트에서 권한을 제거할 때 기존의 `setuid root` 프로그램에 제거된 권한이 필요한 경우 프로그램이 실패할 수 있습니다. 권한 제거의 예는 예 3-21. "사용자의 제한 세트에서 권한 제거" 및 예 5-6. "Sun Ray 사용자 권한 프로파일 만들기"을 참조하십시오.

사용자 ID, 포트 또는 파일 객체가 사용할 수 있는 권한을 제한하려면 “확장 권한 정책을 사용하여 권한 사용 제한” [30]을 참조하십시오.

스크립트에 권한 지정

스크립트는 명령처럼 실행 파일입니다. 따라서 권한 프로파일에서 명령에 권한을 추가하듯이 스크립트에 권한을 추가할 수 있습니다. 권한 프로파일에 지정된 사용자/역할이 프로파일 셸에서 스크립트를 실행할 때 추가된 명령으로 스크립트가 실행됩니다. 스크립트에 권한이 필요한 명령이 포함된 경우 추가된 권한을 가진 명령 역시 지정된 권한 프로파일에 있어야 합니다. 예는 “응용 프로그램 및 스크립트에 대한 권한 지정” [59]을 참조하십시오.

확장 권한 정책을 사용하여 권한 사용 제한

확장 권한 정책은 기본 권한과 명시적으로 부여한 권한을 제외하고 포트, 사용자 ID 또는 파일 객체에 대한 액세스를 제한할 수 있습니다. 권한이 거의 없으므로 리소스를 쉽게 사용하여 시스템을 공격할 수 없습니다. 실제로 사용자는 악성 프로세스에서 액세스하지 못하도록 소유한 파일과 디렉토리를 보호할 수 있습니다. 확장 권한 정책의 예는 “응용 프로그램, 스크립트 및 리소스를 특정 권한으로 제한” [59]을 참조하십시오.

권한(privilege) 에스컬레이션 및 사용자 권한(right)

Oracle Solaris는 보안을 구성할 때 관리자에게 커다란 유연성을 부여합니다. 설치 시 소프트웨어는 권한 에스컬레이션을 금지합니다. 사용자나 프로세스가 부여하려는 것보다 많은 관리 권한(right)을 얻을 경우 권한*(privilege)* 에스컬레이션이 발생합니다. 이런 의미에서 "권한(privilege)"은 커널 권한(privilege)뿐 아니라 모든 권한(right)을 의미합니다. “권한 에스컬레이션 및 커널 권한” [31]을 참조하십시오.

Oracle Solaris 소프트웨어에는 `root` 역할에만 지정된 권한이 있습니다. 다른 보안 보호를 그대로 둔 채, 관리자는 `root` 역할용으로 설계된 속성을 다른 계정에 지정할 수 있지만, 이러한 지정 작업은 몹시 주의를 기울여야 합니다.

다음 권한(right) 프로파일과 권한 부여 세트는 비root 계정의 권한(privilege)을 에스컬레이션할 수 있습니다.

- **Media Restore 권한 프로파일** - 이 프로파일은 다른 권한 프로파일에 속하지 않습니다. Media Restore는 전체 루트 파일 시스템에 대한 액세스를 제공하므로 사용하면 권한 에스컬레이션이 가능합니다. 고의로 수정된 파일이나 대체 매체를 복원할 수 있습니다. 기본적으로 `root` 역할에 이 권한 프로파일이 포함됩니다.
- `solaris.*.assign` **권한 부여** - 이러한 권한 부여는 권한 프로파일에 지정되지 않습니다. `solaris.*.assign` 권한이 부여된 계정은 계정 자체에 지정되지 않은 권한을 다른 사용자에게 지정할 수 있습니다. 예를 들어, `solaris.profile.assign` 권한을 가진 역할은 역

할 자체에 지정되지 않은 권한 프로파일을 다른 계정에 지정할 수 있습니다. 기본적으로 `root` 역할에만 `solaris.*.assign` 권한이 있습니다.

`solaris.*.assign` 권한 부여 대신 `solaris.*.delegate` 권한 부여를 지정합니다. `solaris.*.delegate` 권한 부여를 사용하면 위임자가 보유한 권한만 다른 계정에 지정할 수 있습니다. 예를 들어, `solaris.profile.delegate` 권한이 지정된 역할은 역할 자체에 지정된 권한 프로파일을 다른 사용자와 역할에 지정할 수 있습니다.

커널 권한 에스컬레이션을 금지하려면 "권한 에스컬레이션 및 커널 권한" [31]을 참조하십시오.

권한 에스컬레이션 및 커널 권한

커널은 privilege escalation(권한 에스컬레이션)을 금지합니다. 프로세스가 의도한 것보다 많은 권한을 얻지 못하도록 커널에서 취약한 시스템 수정 시 전체 권한 세트가 있는지 검사합니다. 예를 들어, `root`(`UID=0`)가 소유한 파일 또는 프로세스는 전체 권한 세트를 가진 프로세스만 변경할 수 있습니다. `root` 계정이 `root` 소유의 파일을 변경하는 데에는 권한이 필요하지 않습니다. 그러나 비루트 사용자가 `root` 소유의 파일을 변경하려면 모든 권한이 있어야 합니다.

마찬가지로, 장치 액세스를 제공하는 작업은 유효 세트의 모든 권한이 필요합니다.

특히, `file_chown_self` 및 `proc_owner` 권한에는 권한 에스컬레이션이 적용됩니다.

- `file_chown_self` 권한을 통해 프로세스가 해당 파일을 제공할 수 있습니다. `proc_owner` 권한을 통해 프로세스가 소유하지 않은 프로세스를 검사할 수 있습니다.

  `file_chown_self` 권한은 `rstchown` 시스템 변수로 제한됩니다. `rstchown` 변수가 `0`으로 설정된 경우 `file_chown_self` 권한이 모든 시스템 이미지 사용자의 초기 상속 가능한 세트에서 제거됩니다. `rstchown` 시스템 변수에 대한 자세한 내용은 chown(1) 매뉴얼 페이지를 참조하십시오.

  `file_chown_self` 권한(privilege)은 특정 명령, 권한(right) 프로파일에 배치된 명령 및 역할이나 신뢰할 수 있는 사용자에게 지정된 프로파일에 가장 안전하게 지정됩니다.
- `proc_owner` 권한은 프로세스 UID를 `0`으로 전환하기에 충분하지 않습니다. 모든 UID에서 `UID=0`으로 프로세스를 전환하려면 모든 권한이 필요합니다. `proc_owner` 권한은 시스템의 모든 파일에 무제한 읽기 액세스를 제공하므로 권한이 특정 명령, 프로파일에 배치된 명령 및 역할에 지정된 프로파일에 가장 안전하게 지정됩니다.

주의 - 사용자의 초기 상속 가능한 세트에 `file_chown_self` 권한이나 `proc_owner` 권한을 포함하도록 사용자 계정을 구성할 수 있습니다. 그러나 이러한 강력한 권한을 사용자 또는 역할의 상속 가능한 세트에 배치하려면 대체 보안 이유가 있어야 합니다.

장치에 대한 권한 에스컬레이션을 금지하는 방법에 대한 자세한 내용은 "권한 및 장치" [27]를 참조하십시오. 일반적인 내용은 `privileges`(5) 매뉴얼 페이지를 참조하십시오.

권한 확인

프로세스가 실행되는 셸, 이름 지정 서비스의 범위 및 검색 순서는 지정된 권한을 평가할지 여부에 영향을 줄 수 있습니다. 권한을 평가할 수 없는 프로세스는 실패합니다. 권한 지정 검사 방법에 대한 자세한 내용은 "권한 문제 해결" [95]을 참조하십시오.

프로파일 셸 및 권한 확인

사용자 및 역할은 프로파일 셸에서 권한 있는 응용 프로그램을 실행할 수 있습니다. 프로파일 셸은 권한을 인식하는 특수 셸입니다. 관리자가 사용자에게 프로파일 셸을 로그인 셸로 지정할 수 있습니다. 또는 사용자가 역할을 맡기 위해 `pfexec` 명령 또는 `su` 명령을 실행할 때 프로파일 셸이 시작됩니다. Oracle Solaris에서는 모든 셸에 대응하는 프로파일 셸이 있습니다. 프로파일 셸 목록은 `pfexec`(1) 매뉴얼 페이지를 참조하십시오.

권한 프로파일에 직접 지정되고 로그인 셸이 프로파일 셸이 아닌 사용자는 지정된 권한 있는 명령을 실행하기 위해 프로파일 셸을 열어야 합니다. 인증된 권한 프로파일이 지정된 사용자와 역할에는 인증하라는 메시지가 표시됩니다. 즉, 명령을 실행하려면 암호를 제공해야 합니다. 유용성 및 보안 고려 사항은 "권한 지정 시 고려 사항" [34]을 참조하십시오.

이름 서비스 범위 및 권한 확인

이름 서비스 범위는 지정된 권한을 사용할 수 있는 경우에 영향을 줍니다. 역할의 범위를 개별 호스트로 제한할 수 있습니다. 다른 방법으로, LDAP과 같은 이름 지정 서비스로 제공된 모든 호스트를 범위에 포함할 수 있습니다. 시스템의 이름 서비스 범위는 이름 스위치 서비스 `svc:/system/name-service/switch`에 지정됩니다. 첫번째 일치 시 조회를 중지합니다. 예를 들어, 권한 프로파일이 두 이름 서비스 범위에 존재하는 경우 첫번째 이름 서비스 범위의 항목만 사용됩니다. `files`가 첫번째 일치일 경우 역할의 범위가 로컬 호스트로 제한됩니다. 이름 지정 서비스에 대한 자세한 내용은 `nsswitch.conf`(4) 매뉴얼 페이지, "Oracle Solaris 11.2의 이름 지정 및 디렉토리 서비스 작업: DNS 및 NIS " 및 "Oracle Solaris 11.2의 이름 지정 및 디렉토리 서비스 작업: LDAP "를 참조하십시오.

지정된 권한 검색 순서

직접 또는 권한 프로파일을 통해 사용자나 역할에 security attributes(보안 속성)을 지정할 수 있습니다. 검색 순서는 사용되는 보안 속성 값에 영향을 미칩니다. 첫번째 발견된 속성 인스턴스의 값이 사용됩니다.

참고 - 권한 부여 순서는 중요하지 않습니다. 권한 부여는 누적형입니다.

사용자가 로그인하면 다음 검색 순서로 권한이 지정됩니다.

- useradd 및 usermod 명령으로 사용자에게 직접 지정된 **권한**. 가능한 권한 지정 목록은 "user_attr 데이터베이스" [106]를 참조하십시오.
- **useradd** 및 usermod 명령으로 사용자에게 지정된 권한 프로파일. 이러한 지정 항목은 순차적으로 검색됩니다.
  - 먼저 인증된 권한 프로파일이 검색됩니다.

    순서는 인증된 프로파일 목록의 첫번째 프로파일, 해당 보충 프로파일, 인증된 프로파일 목록의 두번째 프로파일, 해당 보충 프로파일 등입니다. auths 값(누적형)을 제외하면, 값의 첫번째 인스턴스는 시스템이 사용하는 것입니다. 권한 프로파일에 지정할 수 있는 속성에는 사용자 및 보충 프로파일에 지정할 수 있는 모든 권한이 포함됩니다. 목록은 "user_attr 데이터베이스" [106]를 참조하십시오.
  - 그런 다음 다시 인증할 필요가 없는 권한 프로파일이 동일한 방식으로 검색됩니다.
- **Console User 권한 프로파일** 값. 자세한 내용은 "권한 프로파일 참조" [103]를 참조하십시오.
- **Stop 권한 프로파일**이 지정된 경우 보안 속성의 평가가 중지됩니다. Stop 프로파일이 지정된 후에는 어떤 속성도 지정되지 않습니다. Stop 프로파일은 Console User 권한 프로파일을 평가한 후 policy.conf 파일의 기타 보안 속성(AUTHS_GRANTED 포함)을 평가하기 전에 평가됩니다. 자세한 내용은 "권한 프로파일 참조" [103]를 참조하십시오.
- **policy.conf** 파일의 Basic Solaris User 권한 프로파일 값.
- policy.conf 파일의 AUTHS_GRANTED 값.
- policy.conf 파일의 AUTH_PROFS_GRANTED 값.
- policy.conf 파일의 PROFS_GRANTED 값.
- policy.conf 파일의 PRIV_DEFAULT 값.
- policy.conf 파일의 PRIV_LIMIT 값.

권한을 검사하는 응용 프로그램

시스템 컨트롤을 대체할 수 있는 응용 프로그램 및 명령은 권한 있는 응응 프로그램으로 간주됩니다. UID=0과 같은 보안 속성, 권한 및 권한 부여는 응용 프로그램에 권한을 부여합니다.

UID 및 GID를 검사하는 응용 프로그램

root(UID=0) 또는 다른 특수한 UID/GID를 검사하는 권한 있는 응용 프로그램이 UNIX 환경에 오랫동안 존재해 왔습니다. 권한 프로파일 방식을 통해 특수한 ID가 필요한 명령을 격리할 수 있습니다. 누구나 액세스할 수 있는 명령의 ID를 변경하는 대신, 권한 프로파일에 지정된 UID의 명령을 배치할 수 있습니다. 그러면 해당 권한 프로파일을 가진 사용자 또는 역할이 수퍼 유저가 되지 않고도 해당 UID로 프로그램을 실행할 수 있습니다.

실제 또는 유효로 ID를 지정할 수 있습니다. 유효 ID를 지정하는 것이 실제 ID를 지정하는 것보다 선호됩니다. 유효 ID는 파일 사용 권한 비트의 setuid 기능과 같습니다. 유효 ID는 감

사용 UID를 식별하기도 합니다. 그러나 일부 셸 스크립트 및 프로그램은 `root`의 실제 UID가 필요하므로 실제 UID도 설정할 수 있습니다. 예를 들어, `reboot` 명령은 유효 UID보다 실제 UID가 필요합니다.

작은 정보 - 유효 ID가 명령을 실행하기에 부족한 경우 실제 ID를 명령에 지정합니다.

권한을 검사하는 응용 프로그램

권한 있는 응용 프로그램은 권한 사용을 검사할 수 있습니다. 권한(right) 프로파일 방식을 통해 보안 속성이 필요한 특정 명령에 대한 권한(privilege)을 지정할 수 있습니다. 그런 다음, 권한 프로파일에서 지정된 보안 속성 포함 명령을 격리할 수 있습니다. 그러면 해당 권한(right) 프로파일을 가진 사용자 또는 역할이 명령에 필요한 권한(privilege)만으로 명령을 실행할 수 있습니다.

권한을 검사하는 명령은 다음과 같습니다.

- Kerberos 명령 - `kadmin`, `kprop`, `kdb5_util`
- 네트워크 명령 - `ipadm`, `routeadm`, `snoop`
- 파일 및 파일 시스템 명령 - `chmod`, `chgrp`, `mount`
- 프로세스를 제어하는 명령 - `kill`, `pcred`, `rcapadm`

권한(privilege) 포함 명령을 권한(right) 프로파일에 추가하려면 권한 프로파일을 만드는 방법 [78] 및 `profiles`(1) 매뉴얼 페이지를 참조하십시오. 특정 프로파일에서 권한을 검사하는 명령을 확인하려면 6장. Oracle Solaris의 권한 목록을 참조하십시오.

권한 부여를 검사하는 응용 프로그램

일부 Oracle Solaris 명령은 다음을 포함하여 권한 부여를 검사합니다.

- 감사 관리 명령 - `auditconfig`, `auditreduce`
- 프린터 관리 명령 - `cupsenable`, `lpadmin`
- 일괄 처리 작업 명령 - `at`, `atq`, `batch`, `crontab`
- 장치 지향적 명령 - `allocate`, `deallocate`, `list_devices`, `cdrw`.

권한 부여를 위해 스크립트 또는 프로그램을 검사하는 방법에 대한 자세한 내용은 예 4-3. “스크립트 또는 프로그램에서 권한 부여 검사”을 참조하십시오. 권한 부여가 필요한 프로그램을 작성하려면 “Developer’s Guide to Oracle Solaris 11 Security ”의 “About Authorizations”를 참조하십시오.

권한 지정 시 고려 사항

보안 및 유용성 문제는 관리자가 권한을 지정하는 방법에 영향을 줄 수 있습니다.

권한 지정 시 보안 고려 사항

일반적으로 사용자나 역할은 권한 프로파일을 통해 관리 권한을 얻지만 직접 권한을 지정할 수도 있습니다.

- 사용자와 역할에 직접 권한을 지정할 수 있습니다.

  직접 권한 지정은 안전한 방법이 아닙니다. 직접 지정된 권한을 가진 사용자와 역할은 커널을 통해 이 권한이 필요한 어디서든 보안 정책을 대체할 수 있습니다. 사용자 또는 역할의 프로세스를 손상시키는 악성 프로세스도 커널을 통해 필요한 어디서든 이 권한을 사용할 수 있습니다.

  더 안전한 방법은 권한 프로파일에서 명령의 보안 속성으로 권한을 지정하는 것입니다. 그런 다음, 해당 권한 프로파일을 가진 누구나 해당 명령에만 권한을 사용할 수 있습니다.
- 사용자와 역할에 직접 권한 부여를 지정할 수 있습니다.

  권한 부여는 사용자 레벨에서 평가되므로 직접 권한 부여 지정은 직접 권한 지정보다 덜 위험할 수 있습니다. 그러나 권한 부여를 통해 사용자가 감사 플래그 지정과 같은 고도의 보안 작업을 수행할 수 있습니다. 보안 강화를 위해 인증된 권한 프로파일에서 권한 부여를 지정합니다. 이 경우 사용자가 암호를 제공해야 명령을 실행할 수 있습니다.

권한 지정 시 유용성 고려 사항

직접 권한 지정은 유용성에 영향을 줄 수 있습니다.

- 직접 지정된 권한 부여와 사용자 권한 프로파일의 명령 및 권한 부여가 유효하려면 프로파일 셸에서 해석해야 합니다. 기본적으로 사용자에는 프로파일 셸이 지정되지 않습니다. 따라서 사용자가 프로파일 셸을 열고 해당 셸에서 명령을 실행해야 합니다.
- 권한 부여의 개별적 지정은 확장 불가능합니다. 직접 지정된 권한 부여가 작업을 수행하기에 부족할 수도 있습니다. 작업에 권한 있는 명령이 필요할 수 있습니다.

  권한 프로파일은 권한 부여와 권한 있는 명령을 함께 묶도록 설계되었습니다. 또한 사용자 그룹에 맞게 잘 확장됩니다.

♦ ♦ ♦ 2 장

2 관리 권한 구성 계획

이 장에서는 시스템을 관리할 때 기존 권한 모델을 사용할지 또는 Oracle Solaris 권한 모델을 완전히 활용할지 결정하는 데 도움이 되는 정보를 제공합니다. 이 장에서는 다음 항목을 다룹니다.

- "관리에 사용할 권한 모델 결정" [37]
- "선택한 권한 모델 따르기" [38]

권한에 대한 개요는 "사용자 권한 관리" [14]를 참조하십시오. 참조 정보는 8장. Oracle Solaris 권한에 대한 참조를 참조하십시오.

관리에 사용할 권한 모델 결정

Oracle Solaris의 권한에는 권한(right) 프로파일, 권한 부여 및 권한(privilege)이 포함됩니다. Oracle Solaris에서는 시스템에 관리 권한(right)을 구성하는 여러 가지 방법을 제공합니다.

다음 목록에는 수퍼 유저 모델이 가장 안전한 모델부터 덜 안전한 기존 모델 순으로 정렬되어 있습니다.

1. 각각 제한된 권한을 가진 여러 trusted users(신뢰할 수 있는 사용자)에게 관리 작업을 분배합니다. 이 접근 방식은 Oracle Solaris 권한 모델입니다.

   이 접근 방식을 따르는 방법에 대한 자세한 내용은 "선택한 권한 모델 따르기" [38]를 참조하십시오.

   이 접근 방식의 이점에 대한 자세한 내용은 1장. 권한을 사용하여 사용자 및 프로세스 제어 정보를 참조하십시오.
2. 기본 권한 구성을 사용합니다. 이 접근 방식은 권한 모델을 사용하지만 사이트에 맞게 사용자 정의하지 않습니다.

   기본적으로 초기 사용자는 일부 관리 권한을 가지며 `root` 역할을 맡을 수 있습니다. 선택적으로, `root` 역할은 신뢰할 수 있는 다른 사용자에게 `root` 역할을 지정할 수 있습니다. 보안 강화를 위해 `root` 역할은 관리 명령을 감사할 수 있습니다.

   이 모델을 사용하는 관리자에게 유용한 작업은 다음과 같습니다.

   - "지정된 관리 권한 사용" [74]
   - "사용자에게 권한 지정" [41]

- “관리 작업 감사” [77]
- “역할 암호 변경” [48]
- 6장. Oracle Solaris의 권한 목록

3. sudo 명령을 사용합니다.

   sudo 명령을 잘 아는 관리자는 sudo를 구성하고 사용할 수 있습니다. 선택적으로, sudo 사용자가 정해진 기간 동안 다시 인증하지 않고도 관리 명령을 실행할 수 있도록 /etc/sudoers 파일을 구성할 수 있습니다.

   sudo 사용자에게 유용한 작업은 다음과 같습니다.

   - “지정된 관리 권한 사용” [74]
   - “관리 작업 감사” [77]
   - 인증 캐싱 - 예 5-2. “역할 사용의 편의성을 위해 인증 캐싱”

   sudo 명령은 권한 프로파일처럼 커널에 연결되어 있지 않습니다. 명령이 모든 권한(privilege)을 가진 root로 실행되므로 현재 사용자의 /etc/sudoers 파일에서 각 프로그램에 대해 지정된 권한(right)을 부여할 수 있습니다. sudo는 프로그램의 후속 자식 프로세스 속성을 지정할 수 없지만 실행을 차단할 수 있습니다. Oracle Solaris 버전의 sudo는 프로세스에서 PRIV_PROC_EXEC 권한을 제거합니다. 자세한 내용은 Oracle Solaris 버전의 sudo(1M) 매뉴얼 페이지를 참조하십시오.

4. root 역할을 사용자로 변경하여 수퍼 유저 모델을 사용합니다.

   기존의 UNIX 모델을 사용하는 관리자는 root 역할을 사용자로 변경하는 방법 [84]을 완료해야 합니다. 선택적으로, root 사용자는 감사를 구성할 수 있습니다.

선택한 권한 모델 따르기

사용자 및 프로세스 권한 관리는 시스템 배포 관리의 핵심 부분일 수 있습니다. 조직의 보안 요구 사항에 대한 지식과 Oracle Solaris의 권한 이해를 바탕으로 계획을 수립해야 합니다. 이 절에서는 사이트의 권한 사용을 계획하는 일반 프로세스에 대해 설명합니다.

1. 권한에 대한 기본 개념을 알아봅니다.

   1장. 권한을 사용하여 사용자 및 프로세스 제어 정보를 참조하십시오. 권한을 사용한 시스템 관리는 기존의 UNIX 관리 방법과 매우 다릅니다.

2. 보안 정책을 조사합니다.

   조직의 보안 정책은 시스템의 잠재적 위협을 기술하고, 각 위협의 위험도를 측정하고, 이러한 위협에 맞서는 전략을 제시합니다. 권한을 통해 보안 관련 작업을 격리시키는 것도 전략의 일부일 수 있습니다.

   예를 들어, 사이트에서 보안 관리와 비보안 관리를 구분해야 할 수도 있습니다. 책임 구분을 구현하려면 예 3-3. “책임 구분용 역할 만들기”을 참조하십시오.

   보안 정책이 ARMOR(Authorization Rules Managed On RBAC)를 사용하는 경우 ARMOR 패키지를 설치 및 사용해야 합니다. Oracle Solaris에서 사용하려면 예 3-1. “ARMOR 역할 사용”을 참조하십시오.

3. 기본 권한 프로파일을 검토합니다.

   기본 권한 프로파일은 작업을 완료하는 데 필요한 권한을 수집합니다. 사용 가능한 권한 프로파일을 검토하려면 "권한 프로파일 목록" [88]을 참조하십시오.

4. 역할을 사용할지 또는 사용자에게 직접 권한 프로파일을 지정할지 결정합니다.

   역할을 사용하여 권한을 쉽게 관리할 수 있습니다. 역할 이름은 역할이 수행할 수 있는 작업을 식별하고 사용자 권한에서 역할 권한을 격리합니다. 역할을 사용하려는 경우 다음 3 가지 옵션이 있습니다.

   - ARMOR(Authorization Roles Managed on RBAC) 표준이 정의하는 7개 역할을 설치하는 ARMOR 패키지를 설치할 수 있습니다. 예 3-1. "ARMOR 역할 사용"을 참조하십시오.
   - 고유한 역할을 정의하고 ARMOR 역할도 사용합니다. "역할 만들기" [42] 및 예 3-1. "ARMOR 역할 사용"을 참조하십시오.
   - 고유한 역할을 정의하고 ARMOR 역할을 사용하지 않습니다. "역할 만들기" [42]를 참조하십시오.

   사이트에서 역할이 필요하지 않은 경우 사용자에게 직접 권한 프로파일을 지정할 수 있습니다. 사용자가 권한 프로파일에서 관리 작업을 수행할 때 암호를 요구하려면 인증된 권한 프로파일을 사용합니다. 예 3-11. "사용자가 DHCP를 관리하기 전에 암호를 입력하도록 요구"을 참조하십시오.

5. 추가 권한 프로파일을 만들어야 하는지 여부를 결정합니다.

   사이트에서 제한된 액세스를 이용할 수 있는 다른 응용 프로그램이나 응용 프로그램 제품군을 찾아보십시오. 보안에 영향을 주거나, 서비스 거부 문제를 일으킬 수 있거나, 특수한 관리자 교육이 필요한 응용 프로그램이 권한 사용의 좋은 후보입니다. 예를 들어, Sun Ray 시스템의 사용자는 일부 기본 권한만 필요합니다. 사용자를 제한하는 권한 프로파일의 예는 예 3-22. "권한 프로파일에서 기본 권한 제거"를 참조하십시오.

   a. 새 작업에 필요한 권한을 결정합니다.
   b. 기존 권한 프로파일이 이 작업에 적합한지 여부를 결정합니다.
   c. 권한(right) 프로파일의 순서를 지정하여 필요한 권한(privilege)으로 명령이 실행되도록 합니다.

      순서 지정에 대한 자세한 내용은 "지정된 권한 검색 순서" [32]를 참조하십시오.

6. 어떤 사용자에게 어떤 권한을 지정할지 결정합니다.

   최소 권한의 원칙에 따라 사용자의 신뢰 레벨에 적합한 역할에 사용자를 지정합니다. 사용자가 수행할 필요가 없는 작업을 실행하지 못하게 금지하면 잠재적 문제를 줄일 수 있습니다.

참고 - 시스템 이미지의 모든 사용자에게 적용되는 권한은 `/etc/security/policy.conf` 파일에서 지정됩니다.

계획이 있으면 권한 프로파일이나 역할이 지정될 수 있는 trusted users(신뢰할 수 있는 사용자)에 대한 로그인을 만듭니다. 사용자를 만드는 방법에 대한 자세한 내용은 "Oracle Solaris

11.2의 사용자 계정 및 사용자 환경 관리 ”의 “CLI를 사용하여 사용자 계정을 설정 및 관리하기 위한 작업 맵”을 참조하십시오.

권한을 지정하려면 “사용자에게 권한 지정” [41]의 절차로 시작합니다. 이후 절에서는 권한 확장, 권한 제한, 리소스에 권한 지정 및 권한 지정 문제 해결의 예를 제공합니다.

♦ ♦ ♦ 3 장

3 Oracle Solaris에서 권한 지정

이 장에서는 사용자와 역할에 권한을 지정하기 위한 작업을 설명합니다. 이 장에서는 다음 항목을 다룹니다.

- "사용자에게 권한 지정" [41]
- "사용자 권한 확장" [48]
- "사용자 권한 제한" [53]

권한에 대한 개요는 "사용자 권한 관리" [14]를 참조하십시오. 참조 정보는 8장. Oracle Solaris 권한에 대한 참조를 참조하십시오.

사용자에게 권한 지정

Oracle Solaris의 권한은 모든 프로세스에 있습니다. 사용자와 역할에 권한을 추가하고 권한을 제거할 수 있습니다. 권한(right)에는 사용자 프로세스의 권한(privilege), 사용자가 실행하는 명령의 권한(privilege) 또는 특수 ID, 특정 작업을 수행하는 권한 부여가 포함됩니다. 권한을 지정하는 관리 작업이 용이하도록 Oracle Solaris에서는 서비스 및 관리 작업에 대한 권한을 권한 프로파일로 수집합니다. 사용자와 역할에 개별 권한(right)을 지정하는 대신 관리 작업에 필요한 모든 권한 부여 및 권한(privilege)이 포함된 관리(right) 프로파일을 지정할 수 있습니다.

역할은 사용자가 수행할 수 있는 관리 작업에 이름(예: `auditadm`)을 지정합니다. 관리 작업을 수행하기 위해 사용자는 지정된 역할을 맡아 작업을 수행합니다. 역할은 보안 정책에 필요할 수도 있고 편의상 사용될 수도 있습니다. 역할을 만들거나, 역할 7개와 해당 로컬 홈 디렉토리를 만드는 `armor` 패키지를 설치할 수 있습니다. 역할에 대한 자세한 내용은 "사용자 및 프로세스 권한은 수퍼 유저 모델의 대안을 제공함" [14]을 참조하십시오.

권한을 지정할 수 있는 사람

처음에는 추가된 권한으로 사용자를 만들기 위해 `root` 역할을 맡아야 합니다.

`root` 역할에 신뢰할 수 있는 사용자로 또는 역할을 지정하여 분배된 관리 작업이 있는 경우 다음과 같은 권한 프로파일 지정을 사용하여 사용자와 역할을 만들거나 권한을 지정할 수 있습니다.

- 사용자나 역할을 만들려면 User Management 권한 프로파일이 지정된 관리자로 전환해야 합니다.
- 사용자나 역할에 가장 많은 권한을 지정하려면 User Security 권한 프로파일이 지정된 관리자로 전환해야 합니다.

  감사 플래그를 지정할 수 없습니다. `root` 역할만 사용자나 역할에 감사 플래그를 지정할 수 있습니다.

  역할의 암호는 변경할 수 없습니다. `root` 역할만 역할의 암호를 변경할 수 있습니다.

관리 권한이 지정된 경우 관리 명령을 실행하기 전에 "지정된 관리 권한 사용" [74]을 참조하십시오.

사용자와 역할에 권한 지정

이 절에서는 역할과 사용자를 만들고 수정하는 명령에 대해 설명합니다. 권한 프로파일을 만들거나 수정하려면 권한 프로파일을 만드는 방법 [78] 및 시스템 권한 프로파일을 복제하고 수정하는 방법 [80]을 참조하십시오.

역할에 대한 자세한 내용은 "사용자 및 프로세스 권한의 기본 사항" [17]을 참조하십시오.

역할과 사용자를 만들고 수정하는 경우의 기본 작업은 다음과 같습니다.

- 역할 만들기
- 신뢰할 수 있는 사용자 만들기
- 역할의 권한 수정
- 사용자의 권한 수정
- 사용자가 고유한 암호를 사용하여 역할을 맡을 수 있도록 설정
- 역할 암호 변경
- 역할 삭제

역할 만들기

역할을 사용하려는 경우 여러 가지 옵션이 있습니다. ARMOR에서 미리 정의된 역할을 설치하고 배타적으로 사용할 수 있습니다. 또한 역할을 만들고 암호를 제공할 수 있습니다. 만든 역할과 함께 ARMOR 역할을 사용할 수 있습니다.

ARMOR 역할을 사용하려면 예 3-1. "ARMOR 역할 사용"을 참조하십시오.

고유한 역할을 만들려면 `roleadd` 명령을 사용합니다. 이 명령에 대한 인수의 전체 목록은 `roleadd`(1M) 매뉴얼 페이지를 참조하십시오.

예를 들어, 다음 명령은 홈 디렉토리와 `pfbash` 로그인 셸을 사용하여 User Administrator 역할을 만들고 역할의 암호를 만듭니다.

```
# roleadd -c "User Administrator role, local" \
-m -K profiles="User Security,User Management"  useradm
80 blocks
# ls /export/home/useradm
local.bash_profile     local.login     local.profile
# passwd useradm
Password: xxxxxxxx
Confirm Password: xxxxxxxx
```

여기서 각 요소는 다음을 나타냅니다.

`-c` *comment* — 역할을 설명합니다.

`-m` — 홈 디렉토리를 만듭니다.

`-K profiles=` — 역할에 권한 프로파일을 하나 이상 지정합니다. 권한 프로파일 목록은 "권한 프로파일 목록" [88]을 참조하십시오.

rolename — 역할의 이름입니다. 허용 가능한 문자열의 제한 사항은 `roleadd`(1M) 매뉴얼 페이지를 참조하십시오.

참고 - 두 명 이상의 사용자에게 역할 계정을 지정할 수 있습니다. 따라서 관리자는 대개 역할 암호를 만들어서 대역 외에서 사용자에게 역할 암호를 알려줍니다. 역할 암호의 대체 방법은 "사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정" [47], 예 3-16. "사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정" 및 예 3-17. "사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 권한 프로파일 수정"을 참조하십시오.

예 3-1 ARMOR 역할 사용

이 예에서 보안 관리자는 ARMOR 표준에서 정의된 역할을 설치합니다. 관리자는 먼저 역할 이름이 기존 계정과 충돌하지 않는지 확인하고 패키지를 설치한 다음 역할 정의를 보고 신뢰할 수 있는 사용자에게 역할을 지정합니다.

먼저 관리자는 이름 지정 서비스에 다음 UID 및 이름이 없는지 확인합니다.

- `57 auditadm`
- `55 fsadm`
- `58 pkgadm`
- `53 secadm`
- `56 svcadm`
- `59 sysop`
- `54 useradm`

관리자는 UID 및 이름이 사용되고 있지 않은지 확인한 후 패키지를 설치합니다.

```
# pkg install system/security/armor
```

패키지는 /export/home 디렉토리에 역할 및 로컬 홈 디렉토리를 7개 만듭니다.

각 역할의 권한을 보기 위해 관리자는 각 역할에 지정된 프로파일을 나열할 수 있습니다.

```
# profiles auditadm
# profiles fsadm
# profiles pkgadm
# profiles secadm
# profiles svcadm
# profiles sysop
# profiles useradm
```

이러한 권한 지정은 수정할 수 없습니다. 다른 권한 구성을 만들려면 새 역할을 만든 다음 시스템 권한 프로파일을 복제하고 수정하는 방법 [80]의 단계에 따라 새 권한 프로파일을 만들어야 합니다.

마지막으로, 관리자는 신뢰할 수 있는 사용자에게 역할을 지정합니다. 사용자의 고유한 암호는 역할에 인증하는 데 사용됩니다. 일부 사용자에게는 두 개 이상의 역할이 지정됩니다. 시간이 중요한 작업의 역할은 두 명 이상의 신뢰할 수 있는 사용자에게 지정됩니다.

```
# usermod -R=auditadm adal
# usermod -R=fsadm,pkgadm bdewey
# usermod -R=secadm,useradm cfoure
# usermod -R=svcadm ghamada
# usermod -R=svcadm yjones
# usermod -R=sysop hmurtha
# usermod -R=sysop twong
```

예 3-2 LDAP 저장소에 User Administrator 역할 만들기

관리자는 LDAP에 User Administrator 역할을 만듭니다. 사용자가 역할을 맡을 때 암호를 제공하면 개별 명령에 대해 암호를 제공할 필요가 없습니다.

```
# roleadd -c "User Administrator role, LDAP" -m -S ldap \
-K profiles="User Security,User Management"  useradm
```

예 3-3 책임 구분용 역할 만들기

관리자는 역할 2개를 만듭니다. usermgt 역할은 사용자를 만들고, 홈 디렉토리를 제공하고, 기타 비보안 작업을 수행할 수 있습니다. usersec 역할은 사용자를 만들 수 없지만 암호를 지정하고 다른 권한 지정을 변경할 수 있습니다. 두 역할은 모두 사용자 또는 역할에 대해 감사 플래그를 설정하거나 역할의 암호를 변경할 수 없습니다. 이러한 작업은 root 역할로 수행해야 합니다.

```
# roleadd -c "User Management role, LDAP" -s /usr/bin/pfksh \
-m -S ldap -K profiles="User Management"  usermgt
# roleadd -c "User Security role, LDAP" -s /usr/bin/pfksh \
```

```
-m -S ldap -K profiles="User Security"  usersec
```

관리자는 예 3-5. “사용자에게 역할 추가”에서 각 일반 사용자를 만들려면 두 사람이 필요하도록 설정합니다.

예 3-4 암호화 서비스를 관리하는 역할 만들기 및 지정

이 예에서 LDAP 네트워크의 관리자가 암호화 프레임워크를 관리하는 역할을 만들어서 UID 1111에 할당합니다.

```
# roleadd -c "Cryptographic Services manager" \
-g 14 -m -u 104 -S ldap -K profiles="Crypto Management" cryptmgt
# passwd cryptmgt
New Password: xxxxxxxx
Confirm password: xxxxxxxx
# usermod -u 1111 -R +cryptmgt
```

UID가 1111인 사용자가 로그인하여 역할을 맡고 지정된 보안을 표시합니다.

```
% su - cryptmgt
Password: xxxxxxxx
# profiles -l
      Crypto Management
          /usr/bin/kmfcfg            euid=0
          /usr/sbin/cryptoadm        euid=0
          /usr/sfw/bin/CA.pl         euid=0
          /usr/sfw/bin/openssl       euid=0
#
```

암호화 프레임워크에 대한 자세한 내용은 “Oracle Solaris 11.2의 암호화 및 인증서 관리 ”의 1 장, “암호화 프레임워크”를 참조하십시오. 프레임워크를 관리하려면 “Oracle Solaris 11.2의 암호화 및 인증서 관리 ”의 “암호화 프레임워크 관리”를 참조하십시오.

신뢰할 수 있는 사용자에 대한 로그인 만들기

useradd 명령을 사용하여 로그인을 만듭니다. useradd 명령에 대한 인수의 전체 목록은 useradd(1M) 매뉴얼 페이지를 참조하십시오. 명령에 대한 권한 관련 인수는 -R *rolename* 옵션이 추가된 roleadd 명령과 유사합니다.

사용자에게 역할을 지정하면 역할을 맡은 후 역할의 권한을 사용할 수 있습니다. 예를 들어, 다음 명령은 “신뢰할 수 있는 사용자에 대한 로그인 만들기” [45]에서 만든 useradm 역할을 맡을 수 있는 신뢰할 수 있는 사용자를 만듭니다.

```
# useradd -c "Trusted Assistant User Manager user" -m -R useradm jdoe
80 blocks
# ls /export/home/jdoe
```

```
local.bash_profile      local.login      local.profile
```

여기서 각 요소는 다음을 나타냅니다.

-s *shell* — *username*에 대한 로그인 셸을 결정합니다. 이 셸은 프로파일 셸(예: pfbash)일 수 있습니다. 신뢰할 수 있는 사용자에게 프로파일 셸을 지정하는 이유는 "권한 지정 시 유용성 고려 사항" [35]을 참조하십시오. 프로파일 셸 목록은 pfexec(1) 매뉴얼 페이지를 참조하십시오.

-R *rolename* — 기존 역할의 이름을 지정합니다.

추가 예는 "Oracle Solaris 11.2의 사용자 계정 및 사용자 환경 관리 "의 "CLI를 사용하여 사용자 계정을 설정 및 관리하기 위한 작업 맵"을 참조하십시오.

사용자 권한 수정

usermod 명령을 사용하여 사용자 계정을 수정합니다. usermod 명령에 대한 인수의 전체 목록은 usermod(1M) 매뉴얼 페이지를 참조하십시오. 명령에 대한 권한 관련 인수는 useradd 명령과 유사합니다.

사용자에게 권한 프로파일을 지정하면 사용자가 프로파일 셸을 연 후 권한을 사용할 수 있습니다. 예를 들어, 사용자에게 권한 프로파일을 지정합니다.

```
# usermod -K profiles="User Management" kdoe
```

변경 사항은 사용자가 다음에 로그인할 때 적용됩니다. 사용자가 지정된 권한을 사용하는 방법을 알아보려면 "지정된 관리 권한 사용" [74]을 참조하십시오.

예 3-5 사용자에게 역할 추가

이 예에서 관리자는 일반 사용자를 만들려면 신뢰할 수 있는 사용자 2명이 필요하도록 설정합니다. 예 3-3. "책임 구분용 역할 만들기"에서 역할이 생성되었습니다.

```
# usermod -R +useradm jdoe
# usermod -R +usersec mdoe
```

역할 권한 수정

rolemod 명령을 사용하여 역할 계정을 수정합니다. rolemod 명령에 대한 인수의 전체 목록은 rolemod(1M) 매뉴얼 페이지를 참조하십시오. 명령에 대한 권한 관련 인수는 roleadd 명령과 유사합니다.

key=*value* 쌍, -A, -P 및 -R 옵션 값은 빼기(-) 또는 더하기(+) 기호로 수정할 수 있습니다. - 기호는 현재 지정된 값에서 해당 값을 뺍니다. + 기호는 현재 지정된 값에 해당 값을 더하는 것을 나타냅니다. 권한 프로파일의 경우 현재 프로파일 목록 앞에 값이 추가됩니다. 이전 권한 프로파일이 되는 경우의 영향은 "지정된 권한 검색 순서" [32]를 참조하십시오.

예 3-6 권한 프로파일을 역할의 첫번째 권한 프로파일로 추가

예를 들어, useradm 역할 앞에 권한 프로파일을 추가합니다.

```
# rolemod -K profiles+="Device Management" useradm
# profiles useradm
useradm:
Device Management
User Management
User Security
```

예 3-7 로컬 역할의 지정된 프로파일 바꾸기

이 예에서는 보안 관리자가 Printer Management 프로파일 뒤에 VSCAN Management 관리 프로파일이 포함되도록 prtmgt 역할을 수정합니다.

```
# rolemod -c "Handles printers and virus scanning" \
-P "Printer Management,VSCAN Management,All" prtmgt
```

예 3-8 역할에 직접 권한 지정

이 예에서는 보안 관리자가 시스템 시간에 영향을 미치는 매우 특정한 권한으로 realtime 역할을 신뢰합니다. 사용자에게 권한을 지정하려면 예 3-14. "사용자에 직접 권한 지정"를 참조하십시오.

```
# rolemod -K defaultpriv+='proc_clock_highres' realtime
```

defaultpriv 키워드의 값은 항상 역할의 프로세스에서 권한 목록에 속합니다.

사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정

사용자가 역할을 맡을 때 역할 암호가 아니라 고유한 암호를 사용할 수 있도록 설정하려면 역할을 수정합니다.

다음 명령은 useradm 역할이 지정된 모든 사용자가 useradm 역할을 포함하여 지정된 역할을 맡을 때 고유한 암호를 사용할 수 있도록 설정합니다.

```
# rolemod -K roleauth=user useradm
```

역할 암호 변경

역할은 여러 사용자에게 지정할 수 있기 때문에 역할이 지정된 사용자가 역할 암호를 변경할 수 없습니다. 역할 암호를 변경하려면 root 역할을 맡아야 합니다.

```
# passwd useradm
Enter useradm's password: xxxxxxxx
New: xxxxxxxx
Confirm: xxxxxxxx
```

저장소를 지정하지 않으면 모든 저장소에서 암호가 변경됩니다.

추가 명령 옵션은 passwd(1) 매뉴얼 페이지를 참조하십시오.

예 3-9 특정 저장소의 역할 암호 변경

다음 예에서 root 역할은 로컬 devadmin 역할의 암호를 변경합니다.

```
# passwd -r files  devadmin
New password: xxxxxxxx
Confirm password: xxxxxxxx
```

다음 예에서 root 역할은 LDAP 이름 지정 서비스의 devadmin 역할 암호를 변경합니다.

```
# passwd -r ldap devadmin
New password: xxxxxxxx
Confirm password: xxxxxxxx
```

역할 삭제

역할을 삭제하면 즉시 사용할 수 없게 됩니다.

```
# roledel useradm
```

현재 역할에서 관리 작업을 수행 중인 사용자는 계속할 수 없습니다. profiles 명령은 다음 출력을 보여줍니다.

```
useradm # profiles
Unable to get user name
```

사용자 권한 확장

이 절의 작업과 예에서는 기본적으로 사용자가 받는 권한에 권한을 추가합니다. 권한에 대한 자세한 내용은 1장. 권한을 사용하여 사용자 및 프로세스 제어 정보를 참조하십시오.

- 신뢰할 수 있는 사용자에게 역할 지정 - 예 3-1. “ARMOR 역할 사용”, 예 3-4. “암호화 서비스를 관리하는 역할 만들기 및 지정”, 예 3-5. “사용자에게 역할 추가”
- 신뢰할 수 있는 사용자에게 권한 프로파일 지정 - 예 3-10. “DHCP를 관리할 수 있는 사용자 만들기”, 예 3-19. “신뢰할 수 있는 사용자가 확장 계정 파일을 읽을 수 있도록 설정”, 예 4-1. “레거시 응용 프로그램에 보안 속성 지정”
- 신뢰할 수 있는 사용자에게 인증된 권한 프로파일 지정 - 예 3-11. “사용자가 DHCP를 관리하기 전에 암호를 입력하도록 요구”, 예 4-2. “지정된 권한으로 응용 프로그램 실행”
- 신뢰할 수 있는 사용자 또는 역할에 권한 부여 지정 - 예 3-12. “사용자에 직접 권한 부여 지정”, 예 3-13. “역할에 권한 부여 지정”
- 사용자 또는 역할에 직접 권한 지정 - 예 3-8. “역할에 직접 권한 지정”, 예 3-14. “사용자에 직접 권한 지정”, 예 3-15. “역할의 기본 권한에 추가”

주의 - 직접 지정된 권한 및 권한 부여를 부적절하게 사용하면 의도하지 않게 보안 위반이 발생할 수 있습니다. 자세한 내용은 “권한 지정 시 보안 고려 사항” [35]을 참조하십시오.

- 사용자가 역할을 맡을 때 고유한 암호를 사용할 수 있도록 설정 - 예 3-16. “사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정”, 예 3-17. “사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 권한 프로파일 수정”
- 권한 프로파일 수정 - 예 3-22. “권한 프로파일에서 기본 권한 제거”
- 권한 프로파일의 명령에 보안 속성 추가 - 예 3-26. “선택한 응용 프로그램이 새 프로세스를 생성하지 못하도록 금지”, 예 3-27. “게스트가 편집기 하위 프로세스를 생성하지 못하도록 금지”, 예 5-7. “권한 있는 명령을 포함하는 권한 프로파일 만들기”
- 사용자가 `root` 소유 파일을 읽을 수 있도록 설정 - 예 3-19. “신뢰할 수 있는 사용자가 확장 계정 파일을 읽을 수 있도록 설정”, 예 3-20. “비`root` 계정이 `root` 소유 파일을 읽을 수 있도록 설정”
- 사용자 또는 역할이 `root` 소유 파일을 편집할 수 있도록 설정 - 예 5-9. “권한 프로파일에서 선택한 권한 복제 및 제거”
- 새 권한 부여가 포함된 권한 프로파일 지정 - 예 5-11. “권한 부여를 권한 프로파일에 추가”

예 3-10 DHCP를 관리할 수 있는 사용자 만들기

보안 관리자는 DHCP를 관리할 수 있는 사용자를 만듭니다.

```
# useradd -P "DHCP Management" -s /usr/bin/pfbash -S ldap  jdoe
```

사용자에게 로그인 셸로 `pfbash`가 지정되었기 때문에 DHCP Management 관리 프로파일의 권한은 항상 평가되므로 DHCP 관리 명령이 성공합니다.

예 3-11 사용자가 DHCP를 관리하기 전에 암호를 입력하도록 요구

이 예에서 보안 관리자는 `jdoe`가 DHCP를 관리하기 전에 암호를 제공하도록 요구합니다.

```
# usermod -K auth_profiles="DHCP Management" profiles="Edit Administrative Files" jdoe
```

jdoe가 DHCP 명령을 입력하면 암호 프롬프트가 나타납니다. jdoe를 인증한 후 DHCP 명령이 완료됩니다. 검색 순서에서 인증된 권한 프로파일이 일반 프로파일보다 먼저 처리됩니다.

```
jdoe% dhcpconfig -R 120.30.33.7,120.30.42.132
Password: xxxxxxxx
    /** Command completes **/
```

예 3-12 사용자에 직접 권한 부여 지정

이 예에서 보안 관리자가 화면 밝기를 조절할 수 있는 로컬 사용자를 만듭니다.

```
# useradd -c "Screened KDoe, local" -s /usr/bin/pfbash \
-A solaris.system.power.brightness  kdoe
```

이 권한 부여는 사용자의 기존 권한 부여 지정에 추가됩니다.

예 3-13 역할에 권한 부여 지정

이 예에서 보안 관리자는 DNS 서버 서비스에 대한 구성 정보를 변경할 수 있는 역할을 만듭니다.

```
# roleadd -c "DNS administrator role" -m -A solaris.smf.manage.bind" dnsadmin
```

예 3-14 사용자에 직접 권한 지정

이 예에서는 보안 관리자가 시스템 시간에 영향을 미치는 매우 특정한 권한으로 사용자 kdoe를 신뢰합니다. 역할에 권한을 지정하려면 예 3-8. “역할에 직접 권한 지정”을 참조하십시오.

```
# usermod -K defaultpriv='basic,proc_clock_highres' kdoe
```

defaultpriv 키워드의 값이 기존 값을 대체합니다. 따라서 basic 권한을 보유할 사용자에 대해 basic 값이 지정됩니다. 기본 구성에서 모든 사용자는 기본 권한을 갖습니다. 기본 권한 목록은 “권한 목록” [91]을 참조하십시오.

사용자는 추가된 권한 및 해당 정의를 볼 수 있습니다.

```
kdoe% ppriv -v $$
1800:   pfksh
flags = <none>
        E: file_link_any,…,proc_clock_highres,sys_ib_info
        I: file_link_any,…,proc_clock_highres,sys_ib_info
        P: file_link_any,…,proc_clock_highres,sys_ib_info
        L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,win_upgrade_sl
```

```
% ppriv -vl proc_clock_highres
        Allows a process to use high resolution timers.
```

예 3-15 역할의 기본 권한에 추가

다음 예에서는 날짜 및 시간 프로그램을 처리하기 위해 `realtime` 역할에 직접 권한이 지정되었습니다. 예 3-8. “역할에 직접 권한 지정”에서는 `proc_clock_highres`를 `realtime`에 지정했습니다.

```
# rolemod -K defaultpriv='basic,sys_time' realtime

% su - realtime
Password: xxxxxxxx
# ppriv -v $$
1600:   pfksh
flags = <none>
        E: file_link_any,...,proc_clock_highres,sys_ib_info,sys_time
        I: file_link_any,...,proc_clock_highres,sys_ib_info,sys_time
        P: file_link_any,...,proc_clock_highres,sys_ib_info,sys_time
        L: cpc_cpu,dtrace_kernel,dtrace_proc,dtrace_user,...,sys_time
```

예 3-16 사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정

기본적으로 사용자가 역할을 맡으려면 역할의 암호를 입력해야 합니다. 관리자는 사용자 암호를 요구하여 Linux 환경에서 역할을 맡듯이 Oracle Solaris에서 역할을 맡을 수 있게 합니다.

```
# rolemod -K roleauth=user auditrev
```

이 역할을 맡으려면 지정된 사용자가 역할을 위해 특별히 만든 암호가 아니라 고유한 암호를 사용할 수 있습니다.

사용자에게 다른 역할이 지정된 경우 사용자 암호는 해당 역할에도 인증합니다.

예 3-17 사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 권한 프로파일 수정

```
# profiles -p "Local System Administrator"
profiles:Local System Administrator> set roleauth="user"
profiles:Local System Administrator> end
profiles:Local System Administrator> exit
```

Local System Administrator 권한 프로파일에 지정된 사용자가 역할을 맡을 때 암호를 묻는 메시지가 나타납니다. 다음 시퀀스에서 역할 이름은 `admin`입니다.

```
% su - admin
Password: xxxxxxxx
#       /** You are now in a profile shell with administrative rights**/
```

예 3-18 LDAP 저장소에서 역할에 대한 roleauth의 값 변경

이 예에서 root 역할을 통해 secadmin 역할을 맡을 수 있는 모든 사용자가 역할을 맡을 때 고유의 암호를 사용하도록 합니다. 이 기능은 LDAP 서버에서 관리되는 모든 시스템에 대해 해당 사용자에게 부여됩니다.

```
# rolemod -S ldap -K roleauth=user secadmin
```

예 3-19 신뢰할 수 있는 사용자가 확장 계정 파일을 읽을 수 있도록 설정

신뢰할 수 있는 사용자 또는 사용자 그룹이 root 계정에 의해 소유된 파일을 읽을 수 있도록 설정할 수 있습니다. 이 권한은 root 소유 파일이 포함된 관리 응용 프로그램을 실행할 수 있는 사용자에게 유용할 수 있습니다. 이 예에서는 Extended Accounting Net Management 권한 프로파일에 Perl 스크립트를 하나 이상 추가합니다.

root 역할을 맡은 후 관리자는 이름이 network로 시작하는 계정 파일을 읽을 수 있는 기능을 추가하는 관리 프로파일을 만듭니다.

다음 프로파일은 확장 권한 정책을 사용하여 /var/adm/exacct/network* 파일만 액세스할 수 있는 file_dac_read 권한을 스크립트에 부여합니다. 이 프로파일은 기존 Extended Accounting Net Management 관리 프로파일을 보충 프로파일로 추가합니다.

```
# profiles -p "Extended Accounting Perl Scripts"
profiles:Extended Accounting Perl Scripts >
set desc="Perl Scripts for Extended Accounting"
... Scripts> add profiles="Extended Accounting Net Management"
... Scripts> add cmd=/usr/local/bin/exacctdisp.pl
... Scripts:exacctdisp.pl> set privs={file_dac_read}:/var/adm/exacct/network*
... Scripts:exacctdisp.pl> end
... Scripts> commit
... Scripts> exit
```

샘플 스크립트는 “Oracle Solaris 11.2의 리소스 관리 ”의 “libexacct에 대한 Perl 인터페이스 사용”을 참조하십시오.

권한 프로파일 항목에서 오타 오류, 생략, 반복 등의 오류를 검토한 후 관리자는 역할이나 사용자에게 Extended Accounting Perl Scripts 권한 프로파일을 지정합니다.

```
# profiles -p "Extended Accounting Perl Scripts" info
Found profile in files repository.
name=Extended Accounting Perl Scripts
desc=Perl Scripts for Extended Accounting
profiles=Extended Accounting Net Management
cmd=/usr/local/bin/exacctdisp.pl
privs={file_dac_read}:/var/adm/exacct/network*

# rolemod -K profiles+="Extended Accounting Perl Scripts" rolename
# usermod -K profiles+="Extended Accounting Perl Scripts" username
```

예 3-20 비root 계정이 root 소유 파일을 읽을 수 있도록 설정

이 예에서 관리자는 확장 권한(privilege) 정책을 사용하여 권한이 부여된 사용자와 역할이 root가 소유한 /var/adm/sulog 파일을 읽을 수 있도록 설정하는 권한(right) 프로파일을 만듭니다. 관리자는 사용자가 파일을 읽는 데 사용할 수 있는 명령을 추가합니다. head 명령 등 목록에 없는 명령은 사용할 수 없습니다.

```
# profiles -p "Read sulog File"
profiles:Read sulog File
set desc="Read sulog File"
... File> add profiles="Read Log Files"
... File> add cmd=/usr/bin/cat
... File:cat> set privs={file_dac_read}:/var/adm/sulog
... File:cat> end
... File> add cmd=/usr/bin/less
... File:less> set privs={file_dac_read}:/var/adm/sulog
... File:less> end
... File> add cmd=/usr/bin/more
... File:more> set privs={file_dac_read}:/var/adm/sulog
... File:more> end
... File> add cmd=/usr/bin/page
... File:page> set privs={file_dac_read}:/var/adm/sulog
... File:page> end
... File> add cmd=/usr/bin/tail
... File:tail> set privs={file_dac_read}:/var/adm/sulog
... File:tail> end
... File> add cmd=/usr/bin/view
... File:head> set privs={file_dac_read}:/var/adm/sulog
... File:head> end
... File> commit
... File> exit
```

view 명령을 통해 사용자는 파일을 읽을 수 있지만 편집할 수는 없습니다.

사용자 권한 제한

이 절의 예에서는 일반 사용자의 권한을 제한하거나 관리자에서 일부 관리 권한을 제거합니다. 이 예에서는 사용자, 역할 및 권한 프로파일을 수정하는 방법을 보여줍니다. 권한에 대한 자세한 내용은 1장. 권한을 사용하여 사용자 및 프로세스 제어 정보를 참조하십시오.

- 사용자에서 제한 권한 제거 - 예 3-21. “사용자의 제한 세트에서 권한 제거”
- 사용자에서 기본 권한 제거 - 예 3-22. “권한 프로파일에서 기본 권한 제거”
- 고유한 셸 프로세스에서 기본 권한 제거 - 예 3-23. “자신에서 기본 권한 제거”
- 제한된 사용을 위한 시스템 만들기 - 예 3-24. “사용자에 제공되는 권한을 제한하도록 시스템 수정”

- 관리자를 명시적으로 지정된 권한으로 제한 - 예 3-25. “관리자를 명시적으로 지정된 권한으로 제한”
- 시스템의 모든 사용자에서 권한 제거 - 예 3-24. “사용자에 제공되는 권한을 제한하도록 시스템 수정”, 예 3-28. “모든 사용자에게 Editor Restrictions 권한 프로파일 지정”
- 응용 프로그램이 하위 프로세스를 만들지 못하도록 금지 - 예 3-26. “선택한 응용 프로그램이 새 프로세스를 생성하지 못하도록 금지”
- 사용자 프로세스에서 하위 프로세스를 생성하지 못하도록 금지 - 예 3-27. “게스트가 편집기 하위 프로세스를 생성하지 못하도록 금지”
- 게스트용 제한된 편집기 만들기 - 예 3-27. “게스트가 편집기 하위 프로세스를 생성하지 못하도록 금지”
- 공용 시스템에 제한된 편집기 지정 - 예 3-28. “모든 사용자에게 Editor Restrictions 권한 프로파일 지정”
- 권한(right) 프로파일의 제한 세트에서 권한(privilege) 제거 - 예 5-6. “Sun Ray 사용자 권한 프로파일 만들기”
- Sun Ray 사용자에 대한 권한 프로파일 만들기 - 예 5-6. “Sun Ray 사용자 권한 프로파일 만들기”
- 권한 프로파일에서 권한 제거 - 예 5-6. “Sun Ray 사용자 권한 프로파일 만들기”, 예 5-9. “권한 프로파일에서 선택한 권한 복제 및 제거”
- 사용자에서 권한 부여 제거 - 예 5-10. “새 권한 부여 테스트”
- 역할 지정 제거 - 예 5-13. “`root` 역할이 시스템 유지 관리에 사용되지 않도록 금지”

예 3-21 사용자의 제한 세트에서 권한 제거

다음 예에서 `jdoe`의 초기 로그인에서 시작된 모든 세션은 `sys_linkdir` 권한을 사용하지 못하도록 금지됩니다. 사용자는 `su` 명령을 실행한 후에도 디렉토리에 하드 링크를 만들거나 디렉토리 링크를 해제할 수 없습니다.

```
# usermod -K 'limitpriv=all,!sys_linkdir' jdoe
# userattr limitpriv jdoe
all,!sys_linkdir
```

예 3-22 권한 프로파일에서 기본 권한 제거

다음 예제에서는 철저한 테스트를 거친 후 보안 관리자가 Sun Ray Users 권한 프로파일에서 다른 기본 권한을 제거합니다. 관리자가 예 5-6. “Sun Ray 사용자 권한 프로파일 만들기”에서 프로파일을 만든 경우 제한 세트에서 권한 1개를 제거했습니다. 이번에는 기본 권한 2개를 제거합니다. 이 프로파일이 지정된 사용자는 자신의 현재 세션 외부의 프로세스를 검사할 수 없으며 다른 세션을 추가할 수 없습니다.

```
# profiles -p "Sun Ray Users"
profiles:Sun Ray Users> set defaultpriv="basic,!proc_session,!proc_info"
profiles:Sun Ray Users> end
profiles:Sun Ray Users> exit
```

예 3-23 자신에서 기본 권한 제거

다음 예에서는 일반 사용자가 `.bash_profile`을 수정하여 `proc_info` 기본 권한을 제거합니다. `ps`, `prstat` 등의 프로그램 출력에는 사용자의 고유한 프로세스만 포함되며 유용한 정보를 강조할 수 있습니다.

```
##  .bash_profile
## Remove proc_info privilege from my shell
##
ppriv -s EI-proc_info $$
```

`ppriv` 라인은 현재 셸 프로세스(`$$`)에서 사용자의 유효 및 상속 가능한 권한 세트(`EI-`)에 있는 `proc_info` 권한을 제거합니다.

다음 `prstat` 출력에서는 합계가 74개에서 3개 프로세스로 감소합니다.

```
## With all basic privileges
Total: 74 processes, 527 lwps, load averages: 0.01, 0.00, 0.00

## With proc_info removed from the effective and inheritable set
Total: 3 processes, 3 lwps, load averages: 0.00, 0.00, 0.00
```

예 3-24 사용자에 제공되는 권한을 제한하도록 시스템 수정

이 예에서 관리자가 네트워크 관리에만 사용되는 시스템을 만듭니다. 관리자가 Basic Solaris User 권한 프로파일과 모든 권한 부여를 `policy.conf` 파일에서 제거합니다. Console User 권한 프로파일은 제거되지 않습니다. `policy.conf` 결과 파일에서 영향을 받는 라인은 다음과 같습니다.

```
...
##AUTHS_GRANTED=
##AUTH_PROFS_GRANTED=
##PROFS_GRANTED=Basic Solaris User
CONSOLE_USER=Console User
...
```

권한 부여, 명령, 권한 프로파일이 명시적으로 지정된 사용자만 이 시스템을 사용할 수 있습니다. 로그인 후에 권한이 부여된 사용자가 관리 업무를 수행할 수 있습니다. 권한이 부여된 사용자가 시스템 콘솔 앞에 앉으면 Console User의 권한을 갖습니다.

예 3-25 관리자를 명시적으로 지정된 권한으로 제한

다음 두 가지 방법으로 역할이나 사용자를 제한된 수의 관리 작업으로 제한할 수 있습니다.

- Stop 권한 프로파일을 사용자 프로파일 목록의 마지막 프로파일로 지정합니다.

  Stop 권한 프로파일은 제한된 셸을 만드는 가장 간단한 방법입니다. `policy.conf` 파일의 권한 부여 및 권한 프로파일은 사용자나 역할에 지정되지 않습니다.

- 시스템에서 `policy.conf` 파일을 수정하고 역할이나 사용자가 관리 작업에 해당 시스템을 사용하도록 요구합니다. 예 3-24. “사용자에 제공되는 권한을 제한하도록 시스템 수정”를 참조하십시오.

다음 명령은 감사 검토만 수행하도록 `auditrev` 역할을 제한합니다.

```
# rolemod -P "Audit Review,Stop" auditrev
```

`auditrev` 역할에 Console User 권한 프로파일이 없기 때문에 감사자가 시스템을 종료할 수 없습니다. 이 역할에 `solaris.device.cdrw` 권한 부여가 없기 때문에 감사자가 CD-ROM 드라이브에서 읽기/쓰기를 수행할 수 없습니다. 이 역할에 Basic Solaris User 권한 프로파일이 없기 때문에 이 역할에서 해당 프로파일의 명령을 실행할 수 없습니다. All 권한 프로파일은 지정되지 않았으므로 `ls` 명령이 실행되지 않습니다. 역할은 File Browser(파일 브라우저)를 사용하여 검토할 감사 파일을 선택합니다.

자세한 내용은 “지정된 권한 검색 순서” [32] 및 “권한 프로파일 참조” [103]를 참조하십시오.

예 3-26 선택한 응용 프로그램이 새 프로세스를 생성하지 못하도록 금지

이 예에서 관리자는 올바른 작업을 위해 하위 프로세스가 필요 없는 응용 프로그램에 대한 권한 프로파일을 만듭니다. 편의상 관리자는 이러한 실행 파일을 저장할 디렉토리를 만듭니다. 하위 프로세스가 필요 없는 새 응용 프로그램이 추가되면 이 디렉토리에 실행 파일이 추가될 수 있습니다. 또는 실행 파일이 특정 디렉토리에 있어야 하는 경우 관리자가 `/opt/local/noex/`*app-executable*에서 파일에 연결할 수 있습니다.

```
# profiles -p "Prevent App Subprocess"
profiles:Prevent App Subprocess> set desc="Keep apps from execing processes"
profiles:Prevent App Subprocess> add cmd=/opt/local/noex/mkmod
... Subprocess:mkmod> set limitprivs=all,!proc_exec
... Subprocess:mkmod> end
... Subprocess> add cmd=/opt/local/noex/gomap
... Subprocess:gomap> set limitprivs=all,!proc_exec
... Subprocess:gomap> end
... Subprocess> commit
... Subprocess> exit
```

예 3-27 게스트가 편집기 하위 프로세스를 생성하지 못하도록 금지

이 예에서는 관리자가 편집기 명령에서 `proc_exec` 기본 권한을 제거하여 사용자가 하나 이상의 편집기에서 하위 셸을 만들지 못하도록 금지합니다.

1. `vim` 편집기의 제한 권한(privilege) 세트에서 `proc_exec`를 제거하는 권한(right) 프로파일을 만듭니다.

   ```
   # profiles -p -S ldap "Editor Restrictions"
   profiles:Editor Restrictions> set desc="Site Editor Restrictions"
   ```

```
... Restrictions> add cmd=/usr/bin/vim
... Restrictions:vim> set limitprivs=all,!proc_exec
... Restrictions:vim> end
... Restrictions> commit
... Restrictions> exit
```

2. 또한 널리 사용되는 다른 편집기를 권한 프로파일에 추가합니다.

```
# profiles -p "Editor Restrictions"
profiles:Editor Restrictions> add cmd=/usr/bin/gedit
... Restrictions:gedit> set limitprivs=all,!proc_exec
... Restrictions:gedit> end
... Restrictions> add cmd=/usr/bin/gconf-editor
... Restrictions:gconf-editor> set limitprivs=all,!proc_exec
... Restrictions:gconf-editor> end
... Restrictions> add cmd=/usr/bin/ed
... Restrictions:ed> set limitprivs=all,!proc_exec
... Restrictions:ed> end
... Restrictions> add cmd=/usr/bin/ex
... Restrictions:ex> set limitprivs=all,!proc_exec
... Restrictions:ex> end
... Restrictions> add cmd=/usr/bin/edit
... Restrictions:edit> set limitprivs=all,!proc_exec
... Restrictions:edit> end
... Restrictions> commit
... Restrictions> exit
```

3. 권한 프로파일 항목에서 오타 오류, 생략, 반복 등의 오류를 검토합니다.

```
# profiles -p "Editor Restrictions" info
Found profile in files repository.
name=Editor Restrictions
desc=Site Editor Restrictions
cmd=/usr/bin/vim
limitprivs=all,!proc_exec
...
```

4. guest 사용자에게 Editor Restrictions 권한 프로파일을 지정합니다.

```
# usermod -K profiles+="Editor Restrictions" guest
```

profiles+를 사용하여 계정의 현재 권한 프로파일에 이 권한 프로파일을 추가합니다.

5. 편집기 권한이 제한되었는지 확인하기 위해 관리자가 편집기를 열고 새 창에서 편집기 프로세스에 대한 권한을 검사합니다.

```
# ppriv -S $(pgrep vi)
2805:   vi .bash_profile
flags = PRIV_PFEXEC       User is running a profile shell
        E: basic,!proc_info       proc_info is removed from basic set
```

```
I: basic,!proc_info
P: basic,!proc_info
L: all,!proc_exec       proc_exec is removed from limit set
```

예 3-28 모든 사용자에게 Editor Restrictions 권한 프로파일 지정

이 예에서는 관리자가 `policy.conf` 파일에 Editor Restrictions 권한 프로파일을 추가합니다. 관리자는 게스트가 로그인할 수 있는 모든 공용 시스템에 이 파일이 배포되도록 합니다.

```
# cd /etc/security; cp policy.conf policy.conf.orig
# pfedit /etc/security/policy.conf
...
AUTHS_GRANTED=
AUTH_PROFS_GRANTED=
#PROFS_GRANTED=Basic Solaris User
PROFS_GRANTED=Editor Restrictions,Basic Solaris User
```

User Security 관리자가 모든 사용자에게 프로파일 셸을 지정했습니다. 이유 및 절차는 "사용자에게 권한 지정" [41]을 참조하십시오.

♦ ♦ ♦ 4 장

4 응용 프로그램, 스크립트 및 리소스에 대한 권한 지정

이 장에서는 사용자, 포트 및 응용 프로그램에 권한(privilege), 확장 권한 정책 및 기타 권한(right)을 적용하는 작업을 다룹니다.

권한에 대한 개요는 “사용자 권한 관리” [14]를 참조하십시오.

응용 프로그램, 스크립트 및 리소스를 특정 권한으로 제한

이 절의 작업과 예에서는 실행 파일 및 시스템 리소스에 대한 권한을 지정합니다. 일반적으로 실행 파일에 대한 권한을 지정하여 신뢰할 수 있는 사용자가 해당 실행 파일을 실행할 수 있게 합니다. “응용 프로그램 및 스크립트에 대한 권한 지정” [59]에서는 권한 지정을 사용하여 신뢰할 수 있는 사용자가 프로파일 셸에서 응용 프로그램 또는 스크립트를 실행할 수 있게 합니다. “확장 권한을 사용하여 리소스 잠금” [62]에서는 확장 권한 정책을 통해 사용자 ID, 포트 또는 파일 객체를 기본 유효 세트보다 작은 권한 세트로 제한합니다. 지정되지 않은 권한은 해당 사용자의 프로세스, 포트 또는 객체에 대해 거부됩니다. 이러한 지정은 최소 권한 정책에 근접합니다.

응용 프로그램 및 스크립트에 대한 권한 지정

응용 프로그램과 스크립트는 명령 1개나 일련의 명령을 실행합니다. 권한(right)을 지정하려면 권한 프로파일의 각 명령에 대해 세트 ID, 권한(privilege) 등의 보안 속성을 설정합니다. 해당하는 경우 응용 프로그램이 권한 부여를 검사할 수 있습니다.

참고 - 스크립트의 명령이 성공하는 데 `setuid` 비트 또는 `setgid` 비트 세트가 필요한 경우 스크립트 실행 파일 및 명령에 대한 보안 속성이 권한 프로파일에 추가되어 있어야 합니다. 프로파일 셸에서 스크립트를 실행하면 보안 속성으로 명령이 실행됩니다.

- 권한이 필요한 스크립트 실행 - 권한 있는 명령으로 셸 스크립트를 실행하는 방법 [60]
- root가 아닌 사용자가 권한 인식 응용 프로그램을 실행할 수 있도록 허용 - 예 4-1. “레거시 응용 프로그램에 보안 속성 지정”
- root가 아닌 사용자가 root 소유 응용 프로그램을 실행할 수 있도록 허용 - 예 4-2. “지정된 권한으로 응용 프로그램 실행”
- 스크립트에서 권한 부여 검사 - 예 4-3. “스크립트 또는 프로그램에서 권한 부여 검사”

▼ 권한 있는 명령으로 셸 스크립트를 실행하는 방법

권한 있는 셸 스크립트를 실행하려면 스크립트 및 스크립트의 명령에 권한을 추가합니다. 그런 다음 해당 권한(right) 프로파일에 권한(privilege)이 지정된 명령을 포함해야 합니다.

시작하기 전에 root 역할을 맡아야 합니다. 자세한 내용은 “지정된 관리 권한 사용” [74]을 참조하십시오.

1. **첫번째 라인에서 /bin/pfsh 또는 다른 프로파일 셸로 스크립트를 만듭니다.**

```
#!/bin/pfsh
# Copyright (c) 2013 by Oracle
```

2. **일반 사용자로 스크립트의 명령에 필요한 권한을 확인합니다.**

   권한 없이 스크립트를 실행하면 ppriv 명령에 대한 debug 옵션이 누락된 권한을 나열합니다.

```
% ppriv -eD script-full-path
```

   자세한 내용은 프로그램에 필요한 권한을 확인하는 방법 [100]을 참조하십시오.

3. **스크립트에 대한 권한 프로파일을 만들거나 수정합니다.**

   셸 스크립트와 셸 스크립트의 명령을 필요한 보안 속성으로 권한 프로파일에 추가합니다. 권한 프로파일을 만드는 방법 [78]을 참조하십시오.

4. **신뢰할 수 있는 사용자 또는 역할에 권한 프로파일을 지정합니다.**

   예는 “사용자에게 권한 지정” [41]을 참조하십시오.

5. **스크립트를 실행하려면 다음 중 하나를 수행합니다.**

   - **사용자로 스크립트가 지정된 경우 프로파일 셸을 열고 스크립트를 실행합니다.**

```
% pfexec script-full-path
```

   - **역할로 스크립트가 지정된 경우 역할을 맡고 스크립트를 실행합니다.**

```
% su - rolename
```

```
Password: xxxxxxxx
# script-full-path
```

예 4-1 레거시 응용 프로그램에 보안 속성 지정

레거시 응용 프로그램은 권한(privilege)을 인식하지 못하므로 관리자가 권한(right) 프로파일에서 응용 프로그램 실행 파일에 `euid=0` 보안 속성을 지정합니다. 그런 다음 신뢰할 수 있는 사용자에게 지정합니다.

```
# profiles -p LegacyApp
profiles:LegacyApp> set desc="Legacy application"
profiles:LegacyApp> add cmd=/opt/legacy-app/bin/legacy-cmd
profiles:LegacyApp:legacy-cmd> set euid=0
profiles:LegacyApp:legacy-cmd> end
profiles:LegacyApp> exit
# profiles -p LegacyApp 'select cmd=/opt/legacy-app/bin/legacy-cmd;info;end'
 id=/opt/legacy-app/bin/legacy-cmd
 euid=0

# usermod -K profiles+="Legacy application" jdoe
```

예 4-2 지정된 권한으로 응용 프로그램 실행

이 예에서는 관리자가 예 5-7. “권한 있는 명령을 포함하는 권한 프로파일 만들기”의 권한 프로파일을 신뢰할 수 있는 사용자에게 지정합니다. 사용자가 스크립트를 실행할 때 암호를 제공해야 합니다.

```
# usermod -K auth_profiles+="Site application" jdoe
```

예 4-3 스크립트 또는 프로그램에서 권한 부여 검사

권한 부여를 검사하려면 `auths` 명령을 기반으로 하는 테스트를 작성합니다. 이 명령에 대한 자세한 내용은 `auths`(1) 매뉴얼 페이지를 참조하십시오.

예를 들어, 다음 라인은 사용자에게 `$1` 인수로 제공된 권한 부여가 있는지 테스트합니다.

```
if [ `/usr/bin/auths|/usr/xpg4/bin/grep $1` ]; then
        echo Auth granted
else
        echo Auth denied
fi
```

보다 완벽한 테스트에는 와일드카드 사용을 검사하는 논리가 포함됩니다. 예를 들어, 사용자에게 `solaris.system.date` 권한 부여가 있는지 테스트하려면 다음 문자열을 검사해야 합니다.

- `solaris.system.date`
- `solaris.system.*`

- `solaris.*`

프로그램을 작성하는 경우 `getauthattr()` 함수를 사용하여 권한 부여를 테스트합니다.

확장 권한을 사용하여 리소스 잠금

확장 권한 정책은 응용 프로그램에 대한 공격이 성공한 경우 시스템에 대한 공격자 액세스를 제한할 수 있습니다. 확장 정책 규칙은 권한 지정의 영향 범위를 규칙에 포함된 리소스로만 제한합니다. 확장 정책 규칙은 권한을 중괄호로 묶고 뒤에 콜론 및 관련 리소스를 추가하여 표현합니다. 자세한 내용은 "사용자 또는 역할의 권한 확장" [29]을 참조하십시오. 구문의 예는 `ppriv`(1) 및 `privileges`(5) 매뉴얼 페이지를 참조하십시오.

관리자와 일반 사용자 모두 확장 권한을 사용하여 리소스를 잠글 수 있습니다. 관리자는 사용자, 포트 및 응용 프로그램에 대한 확장 권한 규칙을 만들 수 있습니다. 일반 사용자는 명령줄을 사용하거나 `ppriv -r` 명령을 사용하는 스크립트를 작성하여 응용 프로그램이 사용자 지정 디렉토리 외부에 파일을 쓰지 못하도록 할 수 있습니다.

- 진입하는 악의적인 사용자가 사용할 수 있는 액세스를 포트로 제한 - 확장 권한 정책을 포트에 적용하는 방법 [62]
- `root`가 아닌 데몬으로 데이터베이스 실행 - MySQL 서비스를 잠그는 방법 [63]
- `root`가 아닌 데몬으로 Apache 웹 서버 실행 - Apache 웹 서버에 특정 권한을 지정하는 방법 [66]
- Apache 웹 서버가 권한을 사용하여 실행 중인지 확인 - Apache 웹 서버에서 사용 중인 권한을 확인하는 방법 [67]
- Firefox가 시스템의 디렉토리에 쓰지 못하도록 금지 - 예 4-4. "보호된 환경에서 브라우저 실행"
- 응용 프로그램을 시스템의 특정 디렉토리로 제한 - 예 4-5. "응용 프로그램 프로세스에서 시스템의 디렉토리 보호"

▼ 확장 권한 정책을 포트에 적용하는 방법

NTP(Network Time Protocol) 서비스는 `udp` 트래픽에 대해 권한 있는 포트 `123`을 사용합니다. 이 서비스를 실행하려면 권한이 필요합니다. 이 절차 예에서는 서비스 매니페스트를 수정하여 이 포트에 지정된 권한을 얻었을 수도 있는 악의적인 사용자가 다른 포트에 액세스하지 못하도록 보호합니다.

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 "지정된 관리 권한 사용" [74]을 참조하십시오.

1. **포트에 대한 기본 서비스 매니페스트 항목을 읽습니다.**

다음 /lib/svc/manifest/network/ntp.xml start 메소드 항목에서 net_privaddr, proc_lock_memory 및 sys_time 권한은 다른 프로세스에서 사용할 수 있습니다.

```
privileges='basic,!file_link_any,!proc_info,!proc_session,net_privaddr,
proc_lock_memory,sys_time'
```

!file_link_any,!proc_info,!proc_session에서 지정한 제거된 권한은 서비스가 다른 프로세스를 관찰하거나 신호를 보낼 수 없도록 방지하고 파일 이름 바꾸기 방식으로 하드 링크를 만들지 못하도록 방지합니다. 즉, 서비스로 시작된 프로세스는 다른 권한 있는 포트가 아니라 NTP 포트인 123에만 바인딩할 수 있습니다.

해커가 서비스를 악용하여 다른 프로세스를 시작할 수 있는 경우 해당 프로세스도 유사하게 제한됩니다.

2. **start 및 restart 메소드를 수정하여 net_privaddr 권한을 이 포트로만 제한합니다.**

```
# svccfg -s ntp editprop
```

   a. **net_privaddr 문자열을 검색합니다.**

   b. **net_privaddr이 포함된 항목의 주석 처리를 해제합니다.**

   c. **두 항목에서 모두 net_privaddr을 {net_privaddr}:123/udp로 바꿉니다.**
      확장 권한 정책은 지정된 권한 및 지정되지 않은 기본 권한을 제외한 모든 권한을 이 서비스에서 제거합니다. 따라서 잠재적으로 악용 가능한 80개 이상의 권한 세트가 8개 미만으로 줄어듭니다.

3. **확장 권한 정책을 사용하려면 서비스를 다시 시작합니다.**

```
# svcadm restart ntp
```

4. **서비스가 확장 권한을 사용 중인지 확인합니다.**

```
# svccfg -s ntp listprop | grep privileges
start/privileges    astring  basic,!file_link_any,!proc_info,!proc_session,
                    {net_privaddr}:123/udp,proc_lock_memory,sys_time
restart/privileges  astring  basic,!file_link_any,!proc_info,!proc_session,
                    {net_privaddr}:123/udp,proc_lock_memory,sys_time
```

▼ MySQL 서비스를 잠그는 방법

설치 시 MySQL 데이터베이스는 보호되지 않은 포트를 통해 root의 전체 권한으로 실행되도록 구성됩니다. 이 작업에서는 권한(right) 프로파일을 통해 MySQL 서비스에 확장 권한(privilege) 정책을 지정합니다. 권한 프로파일이 서비스의 exec 메소드가 되면 MySQL이 보호된 포트를 통해 사용자 mysql로 실행되고 비MySQL 프로세스에 의한 데이터베이스 액세스가 제한됩니다.

시작하기 전에 초기 사용자는 패키지를 설치할 수 있습니다. 나머지 단계는 root 역할이 수행해야 합니다. 자세한 내용은 "지정된 관리 권한 사용" [74]을 참조하십시오.

1. **MySQL 패키지를 설치합니다.**

```
# pkg search basename:mysql
...
basename ... pkg:/database/mysql-51@version
# pfexec pkg install mysql-51
```

참고 - MySQL 데이터베이스 버전 5.5로 업그레이드하는 경우 5.1 및 51 대신 5.5 및 55를 사용하도록 모든 단계를 수정합니다.

2. **MySQL 서비스의 FMRI 및 상태를 표시합니다.**

```
# svcs mysql
STATE          STIME    FMRI
disabled       May_15   svc:/application/database/mysql:version_51
```

3. **서비스의 실행 메소드를 수정하는 권한 프로파일을 만듭니다.**
이 서비스에 대한 서비스 매니페스트는 실행 메소드를 셸 스크립트 래퍼 /lib/svc/method/mysql_51로 지정합니다.

```
# svccfg -s mysql listprop | grep manifest
... astring     /lib/svc/manifest/application/database/mysql_51.xml
# grep exec= /lib/svc/manifest/application/database/mysql_51.xml
                exec='/lib/svc/method/mysql_51 start'
                exec='/lib/svc/method/mysql_51 stop'
```

프로파일의 명령에 /lib/svc/method/mysql_51 래퍼를 사용합니다.

```
% su -
Password: xxxxxxxx
# profiles -p "MySQL Service"
MySQL Service> set desc="Locking down the MySQL Service"
MySQL Service> add cmd=/lib/svc/method/mysql_51
MySQL Service:mysql_51> set privs=basic
MySQL Service:mysql_51> add privs={net_privaddr}:3306/tcp
MySQL Service:mysql_51> add privs={file_write}:/var/mysql/5.1/data/*
MySQL Service:mysql_51> add privs={file_write}:/tmp/mysql.sock
MySQL Service:mysql_51> add privs={file_write}:/var/tmp/ib*
MySQL Service:mysql_51> end
MySQL Service> set uid=mysql
MySQL Service> set gid=mysql
MySQL Service> exit
```

file_write 권한은 기본적으로 모든 프로세스에 부여되는 기본 권한입니다. 쓰기 가능 경로를 명시적으로 열거하면 쓰기 액세스가 해당 경로로만 제한됩니다. 이 제약 조건은 지정한 실행 파일 및 자식 프로세스에 적용됩니다.

4. **MySQL의 기본 포트를 권한 있는 포트로 만듭니다.**

```
# ipadm set-prop -p extra_priv_ports+=3306 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY            PERM CURRENT     PERSISTENT   DEFAULT    POSSIBLE
tcp   extra_priv_ports    rw   2049,4045,  3306         2049,4045  1-65535
                                 3306
```

권한 있는 포트에 바인딩하려면 `net_privaddr` 권한이 필요합니다. MySQL의 경우 기본 포트 번호 3306에 바인딩할 때는 일반적으로 이 권한이 필요하지 않습니다.

5. **MySQL 서비스에 권한 프로파일을 지정하고 서비스에서 사용하도록 지시합니다.**

```
# svccfg -s mysql:version_51
...version_51> setprop method_context/profile="MySQLService"
...version_51> setprop method_context/use_profile=true
...version_51> refresh
...version_51> exit
```

6. **서비스를 사용으로 설정합니다.**

   FMRI의 마지막 구성 요소인 `mysql:version_51`은 서비스를 고유하게 지정하는 데 충분합니다.

```
# svcadm enable mysql:version_5
```

7. **(옵션) 서비스가 MySQL Service 권한 프로파일에 지정된 권한으로 실행되고 있는지 확인합니다.**

```
# ppriv $(pgrep mysql)
103697:   /usr/mysql/5.1/bin/mysqld --basedir=/usr/mysql/5.1
                                    --datadir=/var/mysql/5.1/data
flags =   PRIV_XPOLICY
    Extended policies:
             {net_privaddr}:3306/tcp
             {file_write}:/var/mysql/5.1/data/*
             {file_write}:/tmp/mysql.sock
             {file_write}:/var/tmp/ib*
        E: basic,!file_write
        I: basic,!file_write
        P: basic,!file_write
        L: all
103609:  /bin/sh /usr/mysql/5.1/bin/mysqld_safe --user=mysql
                               --datadir=/var/mysql/5.1/data
flags =  PRIV_XPOLICY
    Extended policies:
             {net_privaddr}:3306/tcp
             {file_write}:/var/mysql/5.1/data/*
             {file_write}:/tmp/mysql.sock
             {file_write}:/var/tmp/ib*
        E: basic,!file_write
        I: basic,!file_write
        P: basic,!file_write
        L: all
```

▼ Apache 웹 서버에 특정 권한을 지정하는 방법

이 절차에서는 필요한 권한만 지정하여 웹 서버 데몬을 잠급니다. 웹 서버는 포트 80에만 바인딩할 수 있으며 webservd 데몬이 소유한 파일에만 쓸 수 있습니다. root로 실행되는 apache22 서비스 프로세스는 없습니다.

시작하기 전에 root 역할을 맡아야 합니다. 자세한 내용은 "지정된 관리 권한 사용" [74]을 참조하십시오.

1. **Web Server 권한 프로파일을 만듭니다.**

```
# profiles -p "Apache2"
profiles:Apache2> set desc="Apache Web Server Extended Privilege"
profiles:Apache2> add cmd=/lib/svc/method/http-apache22
profiles:Apache2:http-apache22> add privs={net_privaddr}:80/tcp
...http-apache22> add privs={zone}:/system/volatile/apache2
...http-apache22> add privs={zone}:/var/apache2/2.2/logs/*
...http-apache22> add privs={zone}:/var/user
...http-apache22> add privs={file_write}:/var/user/webserv*
...http-apache22> add privs={file_write}:/tmp/*
...http-apache22> add privs={file_write}:/system/volatile/apache*
...http-apache22> add privs={file_write}:/proc/*
...http-apache22> add privs=basic,proc_priocntl
...http-apache22> set uid=webservd
...http-apache22> set gid=webservd
...http-apache22> end
---Apache2> exit
```

2. **(옵션) Apache2와 함께 SSL 커널 프록시를 사용 중인 경우 webservd 확장 정책에 SSL 포트를 추가해야 합니다.**

```
# profiles -p "Apache2"
profiles:Apache2> add privs={net_privaddr}:443/tcp
profiles:Apache2> add privs={net_privaddr}:8443/tcp
profiles:Apache2:http-apache22> end
```

SSL 커널 프록시 절차는 "Oracle Solaris 11.2의 네트워크 보안 "의 "SSL 커널 프록시를 사용하도록 Apache 2.2 웹 서버를 구성하는 방법"에서 설명합니다.

3. **apache22 SMF 시작 메소드에 권한 프로파일을 추가합니다.**

```
# svccfg -s apache22
svc:/network/http:Apache2> listprop start/exec
start/exec  astring "/lib/svc/method/http-apache22 start"
...
svc:/network/http:Apache2> setprop start/profile="Apache2"
svc:/network/http:Apache2> setprop start/use_profile=true
svc:/network/http:Apache2> refresh
svc:/network/http:Apache2> exit
```

apache22 서비스가 사용으로 설정된 경우 Apache2 프로파일이 사용됩니다.

4. **apache22 서비스를 사용으로 설정합니다.**

```
# svcadm enable apache22
```

5. **웹 서버가 작동 중인지 확인합니다.**
   브라우저를 열고 Firefox URL 필드에 localhost를 입력합니다.

다음 순서 권한이 올바르게 적용되었는지 확인하려면 Apache 웹 서버에서 사용 중인 권한을 확인하는 방법 [67]을 계속합니다.

▼ Apache 웹 서버에서 사용 중인 권한을 확인하는 방법

이 작업에서는 Apache2 권한(right) 프로파일의 디버그 버전을 만들어 웹 서버에서 사용 중인 권한(privilege)을 확인합니다.

시작하기 전에 Apache 웹 서버에 특정 권한을 지정하는 방법 [66]을 완료했습니다. apache22 서비스가 사용 안함으로 설정되었습니다. root 역할을 맡고 있습니다.

1. **Apache2 프로파일을 복제하여 다른 명령을 호출합니다.**
   명령 디버그는 SMF 서비스 디버그보다 더 간단합니다. apachectl 명령은 Apache 서비스를 대화식으로 시작합니다.

```
# profiles -p "Apache2"
profiles:Apache2> set name="Apache-debug"
profiles:Apache-debug> sel <Tab><Tab>
profiles:Apache-debug:http-apache22> set id=/usr/apache2/2.2/bin/apachectl
profiles:Apache-debug:apachectl> end
profiles:Apache-debug> exit
```

   자세한 내용은 apachectl(8) 매뉴얼 페이지를 참조하십시오.

2. **복제된 프로파일을 webservd 계정에 지정합니다.**

```
# usermod -K profiles+=Apache-debug webservd
```

3. **webservd ID로 전환합니다.**

```
# su - webservd
```

4. **(옵션) ID를 확인합니다.**

```
# id
uid=80(webservd) gid=80(webservd)
```

5. **프로파일 셸에서 디버그 모드로 웹 서비스를 시작합니다.**
   SMF를 직접 사용하지 마십시오. Apache-debug 권한 프로파일에 명령을 사용합니다.

```
% pfbash
# ppriv -De /usr/apache2/2.2/bin/apachectl start
```

6. **root 역할에서 첫번째 http 데몬의 권한을 검사합니다.**

```
# ppriv $(pgrep httpd|head -1)
2999:   httpd
flags = PRIV_DEBUG|PRIV_XPOLICY|PRIV_EXEC
  5         Extended policies:
  6                 {net_privaddr}:80/tcp
  7                 {zone}:/system/volatile/apache2
  8                 {zone}:/var/apache2/2.2/logs/*
  9                 {zone}:/var/user
 10                 {file_write}:/var/user/webserv*
 11                 {file_write}:/tmp/*
 12                 {file_write}:/system/volatile/apache*
 13                 {file_write}:/proc/*
 14         E: basic,!file_write,!proc_info,proc_priocntl
 15         I: basic,!file_write,!proc_info,proc_priocntl
 16         P: basic,!file_write,!proc_info,proc_priocntl
 17         L: all
```

사용자가 실행하는 응용 프로그램 잠금

사용자는 확장 권한 정책을 사용하여 응용 프로그램에서 기본 권한을 제거할 수 있습니다. 정책은 응용 프로그램이 액세스하면 안되는 디렉토리에 대한 액세스를 금지합니다.

참고 - 순서가 중요합니다. 대부분의 `$HOME/.*` 디렉토리에 대해 보다 제한적인 권한을 지정한 후 `$HOME/Download*` 등의 디렉토리에 대해 보다 광범위한 권한을 지정해야 합니다.

예 4-4 보호된 환경에서 브라우저 실행

이 예에서는 사용자가 보호된 환경에서 Firefox 브라우저를 실행할 수 있는 방법을 보여줍니다. 이 구성에서는 사용자의 `Documents` 디렉토리가 Firefox에서 숨겨집니다.

사용자는 다음 명령을 사용하여 `/usr/bin/firefox` 명령에서 기본 권한을 제거합니다. `ppriv -r` 명령에 대한 확장 권한 인수는 브라우저에서 사용자가 지정한 디렉토리만 읽고 쓸 수 있도록 제한합니다. `-e` 옵션 및 해당 인수는 확장 권한 정책으로 브라우저를 엽니다.

```
% ppriv -r "\
{file_read}:/dev/*,\
{file_read}:/etc/*,\
{file_read}:/lib/*,\
{file_read}:/usr/*,\
{file_read}:/var/*,\
```

```
{file_read}:/proc,\
{file_read}:/proc/*,\
{file_read}:/system/volatile/*,\
{file_write}:$HOME,\
{file_read}:$HOME/.*,\
{file_read,file_write}:$HOME/.mozill*,\
{file_read,file_write}:$HOME/.gnome*,\
{file_read,file_write}:$HOME/Downloa*,\
{file_read,file_write}:/tmp,\
{file_read,file_write}:/tmp/*,\
{file_read,file_write}:/var/tmp,\
{file_read,file_write}:/var/tmp/*,\
{proc_exec}:/usr/*\
" -e /usr/bin/firefox file:///$HOME/Desktop
```

확장 정책에서 `file_read` 및 `file_write` 권한이 사용되는 경우 읽거나 써야 하는 모든 파일에 대한 명시적 액세스를 부여해야 합니다. 이러한 정책에는 와일드카드 문자 `*`를 사용해야 합니다.

자동 마운트된 홈 디렉토리를 처리하기 위해 사용자가 자동 마운트 경로에 대한 명시적 항목을 추가합니다. 예를 들면 다음과 같습니다.

```
{file_read,file_write}:/export/home/$USER
```

사이트에서 `automount` 기능을 사용하지 않는 경우 보호된 디렉토리의 초기 목록이면 충분합니다.

사용자는 셸 스크립트를 만들어 이 명령줄로 보호된 브라우저를 자동 마운트할 수 있습니다. 그런 다음 브라우저를 실행하기 위해 사용자는 `/usr/bin/firefox` 명령이 아니라 스크립트를 호출합니다.

예 4-5 응용 프로그램 프로세스에서 시스템의 디렉토리 보호

이 예에서는 일반 사용자가 셸 스크립트 래퍼를 사용하여 응용 프로그램에 대한 sandbox를 만듭니다. 스크립트의 첫번째 부분은 응용 프로그램을 특정 디렉토리로 제한합니다. Firefox 등의 예외는 스크립트의 뒷부분에서 처리됩니다. 스크립트 뒤에는 스크립트 요소에 대한 설명이 나옵니다.

```
1 #!/bin/bash
2
3 # Using bash because ksh misinterprets extended policy syntax
4
5 PATH=/usr/bin:/usr/sbin:/usr/gnu/bin
6
7 DENY=file_read,file_write,proc_exec,proc_info
8
9 SANDBOX="\
10 {file_read}:/dev/*,\
11 {file_read}:/etc/*,\
```

```
12 {file_read}:/lib/*,\
13 {file_read,file_write}:/usr/*,\
14 {file_read}:/proc,\
15 {file_read,file_write}:/proc/*,\
16 {file_read}:/system/volatile/*,\
17 {file_read,file_write}:/tmp,\
18 {file_read,file_write}:/tmp/*,\
19 {file_read,file_write}:/var/*,\
20 {file_write}:$HOME,\
21 {file_read}:$HOME/.*,\
22 {file_read,file_write}:$PWD,\
23 {file_read,file_write}:$PWD/*,\
24 {proc_exec}:/usr/*\
25 "
26
27 # Default program is restricted bash shell
28
29 if [[ ! -n $1 ]]; then
30     program="/usr/bin/bash --login --noprofile
        --restricted"
31 else
32     program="$@"
33 fi
34
35
36 # Firefox needs more file and network access
37 if [[ "$program" =~ firefox ]]; then
38     SANDBOX+=",\
39 {file_read,file_write}:$HOME/.gnome*,\
40 {file_read,file_write}:$HOME/.mozill*,\
41 {file_read,file_write}:$HOME/.dbu*,\
42 {file_read,file_write}:$HOME/.puls*\
43 "
44
45 else
46     DENY+=",net_access"
47 fi
48
49 echo Starting $program in sandbox
50 ppriv -s I-$DENY -r $SANDBOX -De $program
```

특정 응용 프로그램에 더 많거나 적은 액세스를 허용하도록 정책을 조정할 수 있습니다. 하나의 조정은 라인 38-42에 있으며, 사용자의 홈 디렉토리에서 세션 정보를 유지 관리하는 여러 dot 파일에 대한 쓰기 액세스가 Firefox에 부여됩니다. 또한 Firefox에는 네트워크 액세스를 제거하는 라인 46이 적용되지 않습니다. 그러나 Firefox는 사용자의 홈 디렉토리에서 임의 파일을 읽을 수 없도록 제한되며 현재 디렉토리에만 파일을 저장할 수 있습니다.

추가 레벨의 보호를 위해 라인 30의 기본 프로그램은 제한된 Bash 셸입니다. 제한된 셸은 현재 디렉토리를 변경하거나 사용자의 dot 파일을 실행할 수 없습니다. 따라서 이 셸에서 시작된 모든 명령은 유사하게 sandbox에 잠깁니다.

스크립트의 최종 라인에서는 `ppriv` 명령에 `$DENY` 및 `$SANDBOX`의 권한 세트 2개가 셸 변수로 전달됩니다.

첫번째 세트인 `$DENY`는 프로세스에서 파일 읽기 또는 쓰기, 하위 프로세스 실행, 다른 사용자의 프로세스 관찰 및 (조건부) 네트워크 액세스를 수행하지 못하도록 금지합니다. 이러한 제한은 너무 엄격하므로 두번째 세트인 `$SANDBOX`에서 읽고 쓰고 실행할 수 있는 디렉토리를 열거하여 정책을 세분화합니다.

또한 라인 50에서 디버그 옵션 `-D`가 지정됩니다. 액세스 실패는 터미널 창에 실시간으로 표시되며 명명된 객체 및 성공에 필요한 해당 권한을 포함합니다. 이 디버깅 정보는 사용자가 다른 응용 프로그램에 대한 정책을 사용자 정의하는 데 유용할 수 있습니다.

♦ ♦ ♦ 5 장

5 권한 사용 관리

이 장에서는 권한 모델을 관리에 사용하는 시스템을 유지 관리하는 작업을 다룹니다. 여러 작업을 통해 새 권한 프로파일 및 권한 부여를 만들어 Oracle Solaris에서 제공하는 권한을 확장합니다.

이 장에서는 다음 항목을 다룹니다.

권한에 대한 자세한 내용은 1장. 권한을 사용하여 사용자 및 프로세스 제어 정보를 참조하십시오. 사용자 및 역할에 지정된 권한을 유지 관리하는 방법에 대한 자세한 내용은 3장. Oracle Solaris에서 권한 지정을 참조하십시오.

권한 사용 관리

이 절의 작업과 예에서는 지정된 권한을 사용하는 방법 및 기본적으로 제공되는 권한 구성을 변경하는 방법을 설명합니다.

참고 - 문제 해결 방법에 대한 자세한 내용은 "권한 문제 해결" [95]을 참조하십시오.

- 지정된 권한 사용 - "지정된 관리 권한 사용" [74]
- 관리 작업 감사 - 예 5-5. "두 역할을 사용하여 감사 구성"
- 권한 프로파일 및 권한 부여 추가 - "권한 프로파일 및 권한 부여 만들기" [78]
- root를 사용자로 구성 - root 역할을 사용자로 변경하는 방법 [84]
- root를 다시 역할로 변경 - 예 5-12. "root 사용자를 root 역할로 변경"
- root가 시스템을 관리하지 못하도록 금지 - 예 5-13. "root 역할이 시스템 유지 관리에 사용되지 않도록 금지"

지정된 관리 권한 사용

root 역할에서 초기 사용자는 모든 관리 권한을 갖습니다. root로서 이 사용자는 역할, 권한(right) 프로파일, 특정 권한(privilege)과 권한 부여 등의 관리 권한(right)을 신뢰할 수 있는 사용자에게 지정할 수 있습니다. 이 절에서는 이러한 사용자가 지정된 권한을 사용할 수 있는 방법을 설명합니다.

참고 - Oracle Solaris에서는 관리 파일용 특수 편집기를 제공합니다. 관리 파일을 편집하는 경우 pfedit 명령을 사용합니다. 예 5-1. “시스템 파일 편집”에서는 비root 사용자가 지정된 시스템 파일을 편집할 수 있게 하는 방법을 보여줍니다.

관리 작업을 수행하려면 터미널 창을 열고 다음 옵션에서 선택합니다.

- sudo를 사용 중인 경우 sudo 명령을 입력합니다.

  sudo 명령을 잘 아는 관리자의 경우 sudoers 파일에서 지정된 관리 명령의 이름으로 명령을 실행합니다. 자세한 내용은 sudo(1M) 및 sudoers(4) 매뉴얼 페이지를 참조하십시오.
- 작업에 수퍼 유저 권한이 필요한 경우 root로 전환합니다.

```
% su -
Password: xxxxxxxx
#
```

참고 - 이 명령은 root가 사용자인지 또는 역할인지에 관계없이 작동합니다. 파운드 기호(#) 프롬프트는 지금 root임을 나타냅니다.

- 작업이 역할에 지정된 경우 해당 작업을 수행할 수 있는 역할을 맡습니다.

  다음 예에서는 감사 구성 역할을 맡습니다. 이 역할에는 Audit Configuration 권한 프로파일이 포함됩니다. 관리자로부터 역할 암호를 받았습니다.

```
% su - audadmin
Password: xxxxxxxx
#
```

작은 정보 - 역할 암호를 받지 못한 경우 관리자가 사용자 암호를 요구하도록 역할을 구성한 것입니다. 역할을 맡으려면 사용자 암호를 입력합니다. 이 옵션에 대한 자세한 내용은 예 3-16. “사용자가 역할 암호에 고유한 암호를 사용할 수 있도록 설정”을 참조하십시오.

이 명령을 입력한 셸이 이제 프로파일 셸입니다. 이 셸에서 auditconfig 명령을 실행할 수 있습니다. 프로파일 셸에 대한 자세한 내용은 “프로파일 셸 및 권한 확인” [32]을 참조하십시오.

작은 정보 - 역할의 권한을 보려면 "권한 프로파일 목록" [88]을 참조하십시오.

- 사용자로 작업이 직접 지정된 경우 다음 방법 중 하나로 프로파일 셸을 만듭니다.
  - `pfbash` 명령을 사용하여 관리 권한을 평가하는 셸을 만듭니다.

    다음 예에서는 Audit Configuration 권한 프로파일이 직접 지정되었습니다. 다음 명령 세트를 사용하면 `pfbash` 프로파일 셸에서 감사 사전 선택 값과 감사 정책을 볼 수 있습니다.

    ```
    % pfbash
    # auditconfig -getflags
    active user default audit flags = ua,ap,lo(0x45000,0x45000)
    configured user default audit flags = ua,ap,lo(0x45000,0x45000)
    # auditconfig -getpolicy
    configured audit policies = cnt
    active audit policies = cnt
    ```

  - `pfexec` 명령을 사용하여 관리 명령 1개를 실행합니다.

    다음 예에서는 Audit Configuration 권한 프로파일이 인증된 권한 프로파일로 직접 지정되었습니다. 권한 있는 명령 이름과 `pfexec` 명령을 함께 사용하여 이 프로파일에서 해당 명령을 실행할 수 있습니다. 예를 들어, 사용자의 사전 선택된 감사 플래그를 볼 수 있습니다.

    ```
    % pfexec auditconfig -getflags
    Enter password:       Type your user password
    active user default audit flags = ua,ap,lo(0x45000,0x45000)
    configured user default audit flags = ua,ap,lo(0x45000,0x45000)
    ```

    일반적으로 권한(right)에 포함된 다른 권한 있는 명령을 실행하려면 권한 있는 명령을 입력하기 전에 `pfexec`를 다시 입력해야 합니다. 자세한 내용은 `pfexec`(1) 매뉴얼 페이지를 참조하십시오. 암호 캐싱으로 구성된 경우 예 5-2. "역할 사용의 편의성을 위해 인증 캐싱"와 같이 암호를 제공하지 않고 구성 가능한 간격 내에 후속 명령을 실행할 수 있습니다.

예 5-1 시스템 파일 편집

UID가 `0`인 `root`가 아닌 경우 기본적으로 시스템 파일을 편집할 수 없습니다. 하지만 `solaris.admin.edit/`*path-to-system-file* 권한 부여가 지정된 경우 *system-file*을 편집할 수 있습니다. 예를 들어, `solaris.admin.edit/etc/security/audit_warn` 권한 부여가 지정된 경우 `pfedit` 명령을 사용하여 `audit_warn` 파일을 편집할 수 있습니다.

```
# pfedit /etc/security/audit_warn
```

자세한 내용은 `pfedit`(4) 매뉴얼 페이지를 참조하십시오. 이 명령은 모든 관리자가 사용합니다.

예 5-2 역할 사용의 편의성을 위해 인증 캐싱

이 예에서는 관리자가 감사 구성을 관리하는 역할을 구성하지만 사용자 인증을 캐시하여 사용 편의성을 제공합니다. 먼저, 관리자가 역할을 만들고 지정합니다.

```
# roleadd -K roleauth=user -P "Audit Configuration" audadmin
# usermod -R +audadmin jdoe
```

jdoe가 역할로 전환할 때 -c 옵션을 사용하는 경우 auditconfig 출력 표시에 앞서 암호를 요구합니다.

```
% su - audadmin -c auditconfig option
Password: xxxxxxxx
     auditconfig output
```

인증이 캐시되지 않는 경우 jdoe가 명령을 다시 실행하면 암호 프롬프트가 나타납니다.

관리자는 pam.d 디렉토리에 인증 캐싱을 사용으로 설정하는 su 스택을 포함할 파일을 만듭니다. 인증이 캐시되는 경우 처음에는 암호가 필수이지만, 그 후에는 특정 시간이 경과할 때까지 필수가 아닙니다.

```
# pfedit /etc/pam.d/su
## Cache authentication for switched user
#
auth required           pam_unix_cred.so.1
auth sufficient         pam_tty_tickets.so.1
auth requisite          pam_authtok_get.so.1
auth required           pam_dhkeys.so.1
auth required           pam_unix_auth.so.1
```

파일을 만든 후 관리자가 항목에 오타, 누락, 반복이 있는지 검사합니다.

관리자는 전체 선행 su 스택을 제공해야 합니다. pam_tty_tickets.so.1 모듈은 캐시를 구현합니다. PAM에 대한 자세한 내용은 pam_tty_tickets(5) 및 pam.conf(4) 매뉴얼 페이지와 “Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리 ”의 1 장, “플러그 가능한 인증 모듈 사용”을 참조하십시오.

관리자가 su PAM 파일을 추가하고 시스템을 재부트한 후에는 일련의 명령을 실행할 때 audadmin 역할을 포함하는 모든 역할에 대해 암호를 묻는 프롬프트가 한 번만 표시됩니다.

```
% su - audadmin -c auditconfig option
Password: xxxxxxxx
     auditconfig output
% su - audadmin -c auditconfig option
     auditconfig output
...
```

예 5-3 root 역할 맡기

다음 예에서 초기 사용자가 root 역할을 맡고 역할의 셸에 권한을 나열합니다.

```
% roles
root
% su - root
Password: xxxxxxxx
#       Prompt changes to root prompt
# ppriv $$
1200:   pfksh
flags = <none>
        E: all
        I: basic
        P: all
        L: all
```

권한에 대한 자세한 내용은 “프로세스 권한 관리” [22] 및 ppriv(1) 매뉴얼 페이지를 참조하십시오.

예 5-4 ARMOR 역할 맡기

이 예에서 사용자는 관리자가 지정한 ARMOR 역할을 맡습니다.

사용자는 터미널 창에서 지정된 역할을 확인합니다.

```
% roles
fsadm
sysop
```

그런 다음 fsadm 역할을 맡고 사용자 암호를 제공합니다.

```
% su - fsadm
Password: xxxxxxxx
#
```

su - *rolename* 명령은 터미널 셸을 프로파일 셸로 변경합니다. 이 터미널 창에서 사용자가 이제 fsadm 역할입니다.

이 역할에서 실행할 수 있는 명령을 확인하기 위해 사용자는 “권한 프로파일 목록” [88]의 지침을 따릅니다.

관리 작업 감사

사이트 보안 정책에서는 대체로 관리 작업을 감사하도록 요구합니다. 116:AUE_PFEXEC:execve(2) with pfexec enabled:ps,ex,ua,as 감사 이벤트는 이러한 작업을 캡처합니다. 역할에 사용하기에 적합한 이벤트 그룹을 제공하는 cusa 메타 클래스는 관리 작업을 감사할 때의 또 다른 옵션입니다. 자세한 내용은 /etc/security/audit_class 파일의 주석을 검토하십시오.

예 5-5 두 역할을 사용하여 감사 구성

이 예에서는 두 관리자가 사이트 보안 담당자의 감사 구성 계획을 구현합니다. 계획은 모든 사용자에 대해 `pf` 클래스를 사용하고 개별 역할에 대해 `cusa` 메타 클래스를 지정하는 것입니다. `root` 역할이 역할에 감사 플래그를 지정합니다. 첫번째 관리자가 감사를 구성하고 두번째 관리자가 새 구성을 사용으로 설정합니다.

첫번째 관리자에게 Audit Configuration 권한 프로파일이 지정됩니다. 이 관리자는 현재 감사 구성을 봅니다.

```
# auditconfig -getflags
active user default audit flags = lo(0x1000,0x1000)
configured user default audit flags = lo(0x1000,0x1000)
```

`pf` 클래스에는 `lo` 클래스가 포함되어 있지 않으므로 관리자가 시스템 구성에 클래스를 추가합니다.

```
# auditconfig -setflags lo,pf
```

새 감사 구성을 커널로 읽어오기 위해 Audit Control 권한 프로파일이 지정된 관리자가 감사 서비스를 새로 고칩니다.

```
# audit -s
```

권한 프로파일 및 권한 부여 만들기

제공된 권한 프로파일에 필요한 권한 모음이 없는 경우 권한 프로파일을 만들거나 변경할 수 있습니다. 제한된 권한을 가진 사용자, 새 응용 프로그램 또는 다른 여러 가지 이유로 권한 프로파일을 만들 수 있습니다.

Oracle Solaris에서 제공하는 권한 프로파일은 읽기 전용입니다. 권한 모음이 충분하지 않으면 수정을 위해 제공된 권한 프로파일을 복제할 수 있습니다. 예를 들어, `solaris.admin.edit/`*path-to-system-file* 권한 부여를 제공된 권한 프로파일에 추가해야 할 수 있습니다. 배경 정보는 "권한 프로파일에 대한 추가 정보" [20]를 참조하십시오.

제공된 권한 부여가 권한 있는 응용 프로그램에 코딩된 권한 부여를 포함하지 않는 경우 권한 부여를 만들 수 있습니다. 기존 권한 부여는 변경할 수 없습니다. 배경 정보는 "사용자 권한 부여에 대한 추가 정보" [20]를 참조하십시오.

▼ 권한 프로파일을 만드는 방법

시작하기 전에 권한 프로파일을 만들려면 File Security 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "지정된 관리 권한 사용" [74]을 참조하십시오.

1. **권한 프로파일을 만듭니다.**

```
# profiles -p [-S repository] profile-name
```

설명에 대한 프롬프트가 표시됩니다.

2. **권한 프로파일에 내용을 추가합니다.**

   `set desc`와 같이 단일 값을 갖는 프로파일 등록 정보에 대해 `set` 하위 명령을 사용합니다. `add cmd`와 같이 값이 두 개 이상일 수 있는 등록 정보에 대해 `add` 하위 명령을 사용합니다.

   "Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리 "의 "수정된 PAM 정책을 지정하는 방법"에서 다음 명령은 사용자 정의 PAM 권한 프로파일을 만듭니다. 이름은 표시 목적으로 줄여서 표시됩니다.

```
# profiles -p -S LDAP "Site PAM LDAP"
profiles:Site PAM LDAP> set desc="Profile which sets pam_policy=ldap"
...LDAP> set pam_policy=ldap
...LDAP> commit
...LDAP> end
...LDAP> exit
```

예 5-6 Sun Ray 사용자 권한 프로파일 만들기

이 예에서는 관리자가 LDAP 저장소에 Sun Ray users에 대한 권한 프로파일을 만듭니다. 관리자가 이미 Basic Solaris User 권한 프로파일의 Sun Ray 버전을 만들고 Sun Ray 서버의 `policy.conf` 파일에서 모든 권한 프로파일을 제거했습니다.

```
# profiles -p -S LDAP "Sun Ray Users"
profiles:Sun Ray Users> set desc="For all users of Sun Rays"
... Ray Users> add profiles="Sun Ray Basic User"
... Ray Users> set defaultpriv="basic,!proc_info"
... Ray Users> set limitpriv="basic,!proc_info"
... Ray Users> end
... Ray Users> exit
```

관리자가 해당 내용을 확인합니다.

```
# profiles -p "Sun Ray Users" info
Found profile in LDAP repository.
        name=Sun Ray Users
        desc=For all users of Sun Rays
        defaultpriv=basic,!proc_info,
        limitpriv=basic,!proc_info,
        profiles=Sun Ray Basic User
```

예 5-7 권한 있는 명령을 포함하는 권한 프로파일 만들기

이 예에서 보안 관리자는 관리자가 만드는 권한 프로파일의 응용 프로그램에 권한을 추가합니다. 응용 프로그램은 권한 인식형입니다.

```
# profiles -p SiteApp
```

```
profiles:SiteApp> set desc="Site application"
profiles:SiteApp> add cmd="/opt/site-app/bin/site-cmd"
profiles:SiteApp:site-cmd> add privs="proc_fork,proc_taskid"
profiles:SiteApp:site-cmd> end
profiles:SiteApp> exit
```

확인을 위해 관리자가 `site-cmd`를 선택합니다.

```
# profiles -p SiteApp "select cmd=/opt/site-app/bin/site-cmd; info;end"
Found profile in files repository.
  id=/opt/site-app/bin/site-cmd
  privs=proc_fork,proc_taskid
```

다음 순서 신뢰할 수 있는 사용자 또는 역할에 권한 프로파일을 지정합니다. 예는 예 3-10. “DHCP를 관리할 수 있는 사용자 만들기” 및 예 3-19. “신뢰할 수 있는 사용자가 확장 계정 파일을 읽을 수 있도록 설정”를 참조하십시오.

참조 권한 지정 문제를 해결하려면 권한 지정 문제를 해결하는 방법 [95]을 참조하십시오. 배경 정보는 “지정된 권한 검색 순서” [32]를 참조하십시오.

▼ 시스템 권한 프로파일을 복제하고 수정하는 방법

시작하기 전에 권한 프로파일을 만들거나 변경하려면 File Security 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 “지정된 관리 권한 사용” [74]을 참조하십시오.

1. **기존 프로파일에서 새 권한 프로파일을 만듭니다.**

   ```
   # profiles -p [-S repository] existing-profile-name
   ```

   - **기존 권한 프로파일에 내용을 추가하려면 새 프로파일을 만듭니다.**
     기존 권한 프로파일을 새 프로파일에 보충 권한 프로파일로 추가한 후 향상된 기능을 추가합니다. 예 5-8. “Network IPsec Management 권한 프로파일 복제 및 향상”을 참조하십시오.

   - **기존 권한 프로파일에서 내용을 제거하려면 프로파일을 복제하고 이름을 바꾼 후 수정합니다.**
     예 5-9. “권한 프로파일에서 선택한 권한 복제 및 제거”를 참조하십시오.

2. **보충 권한 프로파일, 권한 부여 및 기타 권한을 추가하거나 제거하여 새 권한 프로파일을 수정합니다.**

예 5-8 Network IPsec Management 권한 프로파일 복제 및 향상

이 예에서 관리자는 `root` 역할이 필수가 아니도록 `solaris.admin.edit` 권한 부여를 사이트 IPsec Management 관리 프로파일에 추가합니다. 이 권한 프로파일은 `/etc/hosts` 파일을 수정할 수 있도록 신뢰된 사용자에만 지정됩니다.

1. 관리자가 Network IPsec Management 권한 프로파일을 수정할 수 없는지 확인합니다.

```
# profiles -p "Network IPsec Management"
profiles:Network IPsec Management> add auths="solaris.admin.edit/etc/hosts"
Cannot add. Profile cannot be modified
```

2. 관리자가 Network IPsec Management 프로파일을 포함하는 권한 프로파일을 만듭니다.

```
# profiles -p "Total IPsec Mgt"
... IPsec Mgt> set desc="Network IPsec Mgt plus /etc/hosts"
... IPsec Mgt> add profiles="Network IPsec Management"
... IPsec Mgt> add auths="solaris.admin.edit/etc/hosts"
... IPsec Mgt> end
... IPsec Mgt> exit
```

3. 관리자가 해당 내용을 확인합니다.

```
# profiles -p "Total IPsec Mgt" info
        name=Total IPsec Mgt
        desc=Network IPsec Mgt plus /etc/hosts
        auths=solaris.admin.edit/etc/hosts
        profiles=Network IPsec Management
```

예 5-9 권한 프로파일에서 선택한 권한 복제 및 제거

이 예에서는 관리자가 서비스를 사용 및 사용 안함으로 설정하는 기능으로부터 VSCAN 서비스의 등록 정보 관리를 분리합니다.

먼저, 관리자가 Oracle Solaris에서 제공되는 권한 프로파일의 내용을 나열합니다.

```
# profiles -p "VSCAN Management" info
        name=VSCAN Management
        desc=Manage the VSCAN service
        auths=solaris.smf.manage.vscan,solaris.smf.value.vscan,
              solaris.smf.modify.application
        help=RtVscanMngmnt.html
```

그런 다음 서비스를 사용 및 사용 안함으로 설정할 수 있는 권한 프로파일을 만듭니다.

```
# profiles -p "VSCAN Management"
profiles:VSCAN Management> set name="VSCAN Control"
profiles:VSCAN Control> set desc="Start and stop the VSCAN service"
... VSCAN Control> remove auths="solaris.smf.value.vscan"
... VSCAN Control> remove auths="solaris.smf.modify.application"
... VSCAN Control> end
... VSCAN Control> exit
```

그런 다음 관리자가 서비스의 등록 정보를 변경할 수 있는 권한 프로파일을 만듭니다.

```
# profiles -p "VSCAN Management"
```

```
profiles:VSCAN Management> set name="VSCAN Properties"
profiles:VSCAN Properties> set desc="Modify VSCAN service properties"
... VSCAN Properties> remove auths="solaris.smf.manage.vscan"
... VSCAN Properties> end
... VSCAN Properties> exit
```

관리자가 새 권한 프로파일의 내용을 확인합니다.

```
# profiles -p "VSCAN Control" info
        name=VSCAN Control
        desc=Start and stop the VSCAN service
        auths=solaris.smf.manage.vscan
# profiles -p "VSCAN Properties" info
        name=VSCAN Properties
        desc=Modify VSCAN service properties
        auths=solaris.smf.value.vscan,solaris.smf.modify.application
```

다음 순서 신뢰할 수 있는 사용자 또는 역할에 권한 프로파일을 지정합니다. 예는 예 3-10. “DHCP를 관리할 수 있는 사용자 만들기” 및 예 3-19. “신뢰할 수 있는 사용자가 확장 계정 파일을 읽을 수 있도록 설정”를 참조하십시오.

참조 권한 지정 문제를 해결하려면 권한 지정 문제를 해결하는 방법 [95]을 참조하십시오. 배경 정보는 “지정된 권한 검색 순서” [32]를 참조하십시오.

▼ 권한 부여를 만드는 방법

시작하기 전에 개발자가 설치 중인 응용 프로그램에 권한 부여를 정의하고 사용했습니다. 자세한 내용은 “Developer’s Guide to Oracle Solaris 11 Security ” 및 “Developer’s Guide to Oracle Solaris 11 Security ”의 “About Authorizations”를 참조하십시오.

1. **(옵션) 새 권한 부여에 대한 도움말 파일을 만듭니다.**

   예를 들어, 사용자가 응용 프로그램에서 데이터를 수정할 수 있도록 권한 부여에 대한 도움말 파일을 만듭니다.

```
# pfedit /docs/helps/NewcoSiteAppModData.html
<HTML>
-- Copyright 2013 Newco.  All rights reserved.
-- NewcoSiteAppModData.html
-->
<HEAD>
     <TITLE>NewCo Modify SiteApp Data Authorization</TITLE>
</HEAD>
<BODY>
The com.newco.siteapp.data.modify authorization authorizes you
to modify existing data in the application.
<p>
Only authorized accounts are permitted to modify data.
Use this authorization with care.
<p>
```

```
</BODY>
</HTML>
```

2. **auths add 명령을 사용하여 권한 부여를 만듭니다.**

   예를 들어, 다음 명령은 로컬 시스템에서 com.newco.siteapp.data.modify 권한 부여를 만듭니다.

   ```
   # auths add -t "SiteApp Data Modify Authorized" \
   -h /docs/helps/NewcoSiteAppModData.html com.newco.siteapp.data.modify
   ```

   이제 권한 부여를 테스트하고 권한 프로파일에 추가한 다음 프로파일을 역할 또는 사용자에게 지정할 수 있습니다.

예 5-10 새 권한 부여 테스트

이 예에서 관리자는 예 5-7. “권한 있는 명령을 포함하는 권한 프로파일 만들기”의 SiteApp 권한 프로파일을 사용하여 com.newco.siteapp.data.modify 권한 부여를 테스트합니다.

```
# usermod -A com.newco.siteapp.data.modify -P SiteApp tester1
```

테스트에 성공하면 관리자가 권한 부여를 제거합니다.

```
# rolemod -A-=com.newco.siteapp.data.modify siteapptester
```

유지 관리하기 쉽도록 관리자가 예 5-11. “권한 부여를 권한 프로파일에 추가”의 SiteApp 권한 프로파일에 권한 부여를 추가합니다.

예 5-11 권한 부여를 권한 프로파일에 추가

권한 부여가 제대로 작동하는지 테스트한 후 보안 관리자는 기존 권한 프로파일에 com.newco.siteapp.data.modify 권한 부여를 추가합니다. 예 5-7. “권한 있는 명령을 포함하는 권한 프로파일 만들기”에서는 관리자가 프로파일을 만든 방법을 보여줍니다.

```
# profiles -p "SiteApp"
profiles:SiteApp> add auths="com.newco.siteapp.data.modify"
profiles:SiteApp> end
profiles:SiteApp> exit
```

확인을 위해 관리자가 프로파일의 내용을 나열합니다.

```
# profiles -p SiteApp
Found profile in files repository.
  id=/opt/site-app/bin/site-cmd
  auths=com.newco.siteapp.data.modify
```

다음 순서 신뢰할 수 있는 사용자 또는 역할에 권한 프로파일을 지정합니다. 예는 예 3-10. “DHCP를 관리할 수 있는 사용자 만들기” 및 예 3-19. “신뢰할 수 있는 사용자가 확장 계정 파일을 읽을 수 있도록 설정”를 참조하십시오.

참조 권한 지정 문제를 해결하려면 권한 지정 문제를 해결하는 방법 [95]을 참조하십시오. 배경 정보는 “지정된 권한 검색 순서” [32]를 참조하십시오.

root가 사용자인지 또는 역할인지 변경

기본적으로 root는 Oracle Solaris의 역할입니다. 사용자로 변경하거나, 다시 역할로 변경하거나, 사용되지 않도록 제거할 수 있습니다.

Oracle Enterprise Manager를 사용 중이거나 권한 모델 대신 기존의 수퍼 유저 관리 모델을 따르는 경우 root를 사용자로 변경해야 합니다. 배경 정보는 “관리에 사용할 권한 모델 결정” [37]을 참조하십시오.

권한 모델을 따르는 경우 네트워크에서 제거된 시스템의 서비스를 해제할 때 root를 사용자로 변경할 수 있습니다. 이 시나리오에서 root로 시스템에 로그인하면 간단히 정리됩니다.

참고 - root 역할을 사용하여 원격으로 관리하는 경우 보안 원격 로그인 지침은 “Oracle Solaris 11.2의 보안 셸 액세스 관리 ”의 “보안 셸을 사용하여 ZFS를 원격으로 관리하는 방법”을 참조하십시오.

일부 사이트에서는 root가 생산 시스템의 적합한 계정이 아닙니다. root가 사용되지 않도록 제거하려면 예 5-13. “root 역할이 시스템 유지 관리에 사용되지 않도록 금지”을 참조하십시오.

▼ root 역할을 사용자로 변경하는 방법

이 절차는 root가 시스템에 직접 로그인할 수 있어야 하는 시스템에서 필요합니다.

시작하기 전에 root 역할을 맡아야 합니다.

1. **root 역할 지정을 로컬 사용자로부터 제거합니다.**
   예를 들어, 두 사용자로부터 역할 지정을 제거합니다.

```
% su -
Password: xxxxxxxx
# roles jdoe
root
# roles kdoe
root
# roles ldoe
secadmin
# usermod -R "" jdoe
# usermod -R "" kdoe
#
```

2. **root 역할을 사용자로 변경합니다.**

```
# rolemod -K type=normal root
```

현재 root 역할에 속한 사용자는 그대로 남고, 루트 액세스를 가진 다른 사용자는 root에 su를 실행하거나 root 사용자로 시스템에 로그인할 수 있습니다.

3. **변경 사항을 확인합니다.**
다음 명령 중 하나를 사용할 수 있습니다.

- **root에 대한 user_attr 항목을 검사합니다.**

```
# getent user_attr root
root::::auths=solaris.*;profiles=All;audit_flags=lo\:no;lock_after_retries=no;
min_label=admin_low;clearance=admin_high
```

type 키워드가 출력에서 누락되거나 normal과 같은 경우 계정은 역할이 아닙니다.

- **userattr 명령의 출력을 봅니다.**

```
# userattr type root
```

출력이 비어 있거나 normal을 나열하는 경우 계정은 역할이 아닙니다.

예 5-12 root 사용자를 root 역할로 변경

이 예에서 root 사용자는 root 사용자를 다시 역할로 전환합니다.

먼저, root 사용자가 root 계정을 역할로 변경하고 변경 사항을 확인합니다.

```
# usermod -K type=role root
# getent user_attr root
root::::type=role...
```

그런 다음, root가 root 역할을 로컬 사용자에 지정합니다.

```
# usermod -R root jdoe
```

예 5-13 root 역할이 시스템 유지 관리에 사용되지 않도록 금지

이 예에서 사이트 보안 정책에 따라 root 계정이 시스템을 유지 관리하지 못하도록 해야 합니다. 관리자가 시스템을 유지 관리하는 역할을 만들고 테스트했습니다. 이러한 역할에는 모든 보안 프로파일과 System Administrator 권한 프로파일이 포함됩니다. 신뢰된 사용자에게 백업을 복원할 수 있는 역할이 지정되었습니다. 사용자, 역할 또는 권한 프로파일에 대한 감사 플래그를 변경하거나 역할의 암호를 변경할 수 있는 역할은 없습니다.

root 계정이 시스템 유지 관리에 사용되지 못하도록 하려면 보안 관리자가 root 역할 지정을 제거합니다. root 계정이 단일 사용자 모드로 시스템에 로그인할 수 있어야 하므로 계정이 암호를 유지합니다.

```
# usermod -K roles= jdoe
# userattr roles jdoe
```

일반 오류 데스크탑 환경에서 root가 역할이면 root로 직접 로그인할 수 없습니다. 진단 메시지는 root가 시스템의 역할임을 나타냅니다.

root 역할을 맡을 수 있는 로컬 계정이 없는 경우 다음 단계를 수행합니다.

- 단일 사용자 모드에서 시스템에 root로 로그인하고 로컬 사용자 계정과 암호를 만듭니다.
- 새 계정에 root 역할을 지정합니다.
- 새 사용자로 로그인하고 root 역할을 맡습니다.

♦ ♦ ♦ 6 장

6 Oracle Solaris의 권한 목록

이 장에서는 시스템의 모든 권한, 특정 사용자에게 지정된 권한 및 고유한 권한을 나열하는 방법을 설명합니다.

권한에 대한 개요는 “사용자 권한 관리” [14]를 참조하십시오. 참조 정보는 8장. Oracle Solaris 권한에 대한 참조를 참조하십시오.

권한 및 해당 정의 목록

이 절의 명령을 사용하면 시스템에서 정의된 권한을 찾고 사용자 프로세스에 적용된 권한을 나열할 수 있습니다.

이 절의 명령에 대한 자세한 내용은 다음 매뉴얼 페이지를 참조하십시오.

- `auths`(1)
- `getent`(1M)
- `ppriv`(1)
- `profiles`(1)
- `privileges`(5)
- `roles`(1)

권한 부여 목록

- `auths` - 현재 사용자의 권한 부여를 나열합니다.
- `auths list` - 현재 사용자의 권한 부여를 나열합니다.

- `auths list -u` *username* - *username*의 권한 부여를 나열합니다.
- `auths list -x` - 인증이 필요한 현재 사용자의 권한 부여를 나열합니다.
- `auths list -xu` *username* - 인증이 필요한 *username* 권한 부여를 나열합니다.
- `auths info` - 이름 지정 서비스의 모든 권한 부여 이름을 나열합니다.
- `getent auth_attr` - 이름 지정 서비스의 모든 권한 부여에 대한 전체 정의를 나열합니다.

예 6-1 모든 권한 부여 목록

```
$ auths info
    solaris.account.activate
    solaris.account.setpolicy
    solaris.admin.edit
...
    solaris.zone.login
    solaris.zone.manage
```

예 6-2 권한 부여 데이터베이스의 내용 목록

```
$ getent auth_attr | more
solaris.:::All Solaris Authorizations::help=AllSolAuthsHeader.html
solaris.account.:::Account Management::help=AccountHeader.html
...
solaris.zone.login:::Zone Login::help=ZoneLogin.html
solaris.zone.manage:::Zone Deployment::help=ZoneManage.html
```

예 6-3 사용자의 기본 권한 부여 목록

다음 권한 부여는 기본적으로 모든 사용자에게 지정되는 권한 프로파일에 포함됩니다.

```
$ auths
solaris.device.cdrw,solaris.device.mount.removable,solaris.mail.mailq
solaris.network.autoconf.read,solaris.admin.wusb.read
solaris.smf.manage.vbiosd,solaris.smf.value.vbiosd
```

권한 프로파일 목록

- `profiles` - 현재 사용자의 권한 프로파일을 나열합니다.
- `profiles -a` - 모든 권한 프로파일 이름을 나열합니다.
- `profiles -l` - 현재 사용자의 권한 프로파일에 대한 전체 정의를 나열합니다.
- `profiles` *username* - *username*의 권한 프로파일을 나열합니다.
- `profiles -x` - 인증이 필요한 현재 사용자의 권한 프로파일을 나열합니다.
- `profiles -x` *username* - 인증이 필요한 *username*의 권한 프로파일을 나열합니다.
- `profiles -p` *profile-name* `info` - 지정된 권한 프로파일의 내용을 예쁘게 인쇄합니다.

- `getent prof_attr` - 이름 지정 서비스의 모든 권한 프로파일에 대한 전체 정의를 나열합니다.

예 6-4 모든 권한 프로파일의 이름 목록

```
$ profiles -a
        Console User
        CUPS Administration
        Desktop Removable Media User
...
        VSCAN Management
        WUSB Management
```

예 6-5 권한 프로파일 데이터베이스의 내용 목록

```
$ getent prof_attr | more
All:::Execute any command as the user or role:help=RtAll.html
Audit Configuration:::Configure Solaris Audit:auths=solaris.smf.value.audit;
help=RtAuditCfg.html
...
Zone Management:::Zones Virtual Application Environment Administration:
help=RtZoneMngmnt.html
Zone Security:::Zones Virtual Application Environment Security:auths=solaris.zone.*,
solaris.auth.delegate;help=RtZoneSecurity.html ...
```

예 6-6 사용자의 기본 권한 프로파일 목록

내 권한 프로파일을 나열합니다. 다음 권한 프로파일은 기본적으로 모든 사용자에 지정됩니다.

```
$ profiles
Basic Solaris User
All
```

예 6-7 초기 사용자의 권한 프로파일 목록

초기 사용자에게는 여러 권한 프로파일이 지정됩니다.

```
$ profiles Initial user
System Administrator
Audit Review
...
CPU Power Management
Basic Solaris User
All
```

초기 사용자의 프로파일에 지정된 모든 보안 속성을 표시하려면 `-l` 옵션을 사용합니다.

```
$ profiles -l Initial user | more
```

Initial user:
```
System Administrator
 profiles=Install Service Management,Audit Review,Extended Accounting
Flow Management,Extended Accounting Net Management,Extended Accounting Process
Management,Extended Accounting Task Management,Printer Management,Cron Managem
ent,Device Management,File System Management,Log Management,Mail Management,
Maintenance and Repair,Media Catalog,Name Service Management,Network Management,
Project Management,RAD Management,Service Operator,Shadow Migration Monitor,So
Software Installation,System Configuration,User Management,ZFS Storage Management
          /usr/sbin/gparted          uid=0
Install Service Management
 auths=solaris.autoinstall.service
 profiles=Install Manifest Management,Install Profile Management,
Install Client Management
...
```

예 6-8 지정된 권한 프로파일의 내용 목록

초기 사용자는 Audit Review 프로파일을 통해 부여된 권한을 나열합니다.

```
$ profiles -l
Audit Review
  solaris.audit.read

  /usr/sbin/auditreduce  euid=0
  /usr/sbin/auditstat    privs=proc_audit
  /usr/sbin/praudit      privs=file_dac_read
```

예 6-9 권한 프로파일에 포함된 명령의 보안 속성 목록

`profiles` 명령의 이 변형은 지정되지 않은 권한 프로파일에 포함된 명령의 보안 속성을 보는 데 유용합니다.

먼저 프로파일의 명령을 나열합니다.

```
% profiles -p "Audit Review" info
 name=Audit Review
 desc=Review Solaris Auditing logs
 help=RtAuditReview.html
 cmd=/usr/sbin/auditreduce
 cmd=/usr/sbin/auditstat
 cmd=/usr/sbin/praudit
```

그런 다음 프로파일에 포함된 명령 중 하나의 보안 속성을 나열합니다.

```
% profiles -p "Audit Review" "select cmd=/usr/sbin/praudit ; info; end;"
select: command is read-only
 id=/usr/sbin/praudit
 privs=file_dac_read
end: command is read-only
```

예 6-10 최근에 생성된 권한 프로파일의 내용 목록

`less` 옵션은 가장 최근에 추가된 권한 프로파일을 먼저 표시합니다. `profiles` 명령의 이 변형은 사이트에서 권한 프로파일을 만들거나 수정할 때 유용합니다. 다음 출력은 예 4-1. “레거시 응용 프로그램에 보안 속성 지정”에서 추가된 프로파일의 내용을 보여줍니다. 일반 사용자는 이 명령을 실행할 수 있습니다.

```
$ profiles -la | less
LegacyApp
         /opt/legacy-app/bin/legacy-cmd
                                   euid=0
OpenLDAP...
```

역할 목록

- `roles` - 현재 사용자의 역할을 나열합니다.
- `roles` *username* - *username*의 역할을 나열합니다.
- `logins -r` - 사용 가능한 모든 역할을 나열합니다.

예 6-11 지정된 역할 목록

`root` 역할은 기본적으로 초기 사용자에게 지정됩니다. `No roles`는 역할이 지정되지 않음을 나타냅니다.

```
$ roles
root
```

권한 목록

- `man privileges` - 개발자가 사용하는 권한 정의 및 해당 이름을 나열합니다.
- `ppriv -vl` - 관리자가 사용하는 권한 정의 및 해당 이름을 나열합니다.
- `ppriv -vl basic` - 기본 권한 세트에 포함된 권한의 이름 및 정의를 나열합니다.
- `ppriv $$` - 현재 셸(`$$`)의 권한을 나열합니다.
- `getent exec_attr` - 보안 속성(`setuid` 또는 권한(privilege))이 포함된 모든 명령을 권한(right) 프로파일 이름으로 나열합니다.

```
$ getent exec_attr | more
All:solaris:cmd:::*:
Audit Configuration:solaris:cmd:::/usr/sbin/auditconfig:privs=sys_audit
...
```

```
Zone Security:solaris:cmd:::/usr/sbin/txzonemgr:uid=0
Zone Security:solaris:cmd:::/usr/sbin/zonecfg:uid=0 ...
```

예 6-12 모든 권한 및 해당 정의 목록

privileges(5) 매뉴얼 페이지에 설명된 권한 형식은 개발자가 사용합니다.

```
$ man privileges
Standards, Environments, and Macros                 privileges(5)

NAME
     privileges - process privilege model
...
     The defined privileges are:

     PRIV_CONTRACT_EVENT

         Allow a process to request reliable delivery  of  events
         to an event endpoint.

         Allow a process to include events in the critical  event
         set  term  of  a  template  which  could be generated in
         volume by the user.
...
```

예 6-13 권한 지정에 사용된 권한 목록

ppriv 명령은 모든 권한을 이름으로 나열합니다. 정의에는 -v 옵션을 사용합니다.

이 권한(privilege) 형식은 useradd, roleadd, usermod, rolemod 명령을 사용하여 사용자 및 역할에 권한(privilege)을 지정하고 profiles 명령을 사용하여 권한(right) 프로파일에 권한을 지정합니다.

```
$ ppriv -lv | more
contract_event
 Allows a process to request critical events without limitation.
 Allows a process to request reliable delivery of all events on
 any event queue.
...
win_upgrade_sl
        Allows a process to set the sensitivity label of a window
        resource to a sensitivity label that dominates the existing
        sensitivity label.
        This privilege is interpreted only if the system is configured
        with Trusted Extensions.
```

예 6-14 현재 셸의 권한 목록

모든 사용자는 기본적으로 기본 권한 세트에 지정됩니다. 기본 제한 세트는 모든 권한입니다.

출력의 단문자는 다음 권한 세트를 가리킵니다.

E 유효 권한 세트

I 상속 가능한 권한 세트

P 허가된 권한 세트

L 제한 권한 세트

```
$ ppriv $$
1200:   -bash
flags = <none>
        E: basic
        I: basic
        P: basic
        L: all
$ ppriv -v $$
1200:   -bash
flags = <none>
 E: file_link_any,file_read,file_write,net_access,proc_exec,proc_fork,
          proc_info,proc_session,sys_ib_info
 I: file_link_any,file_read,...,sys_ib_info
 P: file_link_any,file_read,...,sys_ib_info
 L: contract_event,contract_identity,...,sys_time
```

이중 달러 기호($$)는 부모 셸의 프로세스 번호를 명령에 전달합니다. 지정된 권한(right) 프로파일의 명령으로 제한된 권한(privilege)은 이 목록에 포함되지 않습니다.

예 6-15 기본 권한 및 해당 정의 목록

```
$ ppriv -vl basic
file_link_any
 Allows a process to create hardlinks to files owned by a uid
 different from the process' effective uid.
file_read
 Allows a process to read objects in the filesystem.
file_write
 Allows a process to modify objects in the filesystem.
net_access
 Allows a process to open a TCP, UDP, SDP or SCTP network endpoint.
proc_exec
 Allows a process to call execve().
proc_fork
 Allows a process to call fork1()/forkall()/vfork()
proc_info
 Allows a process to examine the status of processes other
 than those it can send signals to.  Processes which cannot
 be examined cannot be seen in /proc and appear not to exist.
proc_session
 Allows a process to send signals or trace processes outside its
```

```
 session.
sys_ib_info
 Allows a process to perform read InfiniBand MAD (Management Datagram)
 operations.
```

예 6-16 권한 프로파일의 보안 속성 포함 명령 목록

Basic Solaris User 프로파일은 사용자가 CD-ROM을 읽고 쓸 수 있는 명령을 포함합니다.

```
$ profiles -l
  Basic Solaris User
...
   /usr/bin/cdrecord.bin   privs=file_dac_read,sys_devices,
     proc_lock_memory,proc_priocntl,net_privaddr
   /usr/bin/readcd.bin     privs=file_dac_read,sys_devices,net_privaddr
   /usr/bin/cdda2wav.bin   privs=file_dac_read,sys_devices,
     proc_priocntl,net_privaddr
  All
   *
```

한정 속성 목록

- `man user_attr` - 보안 속성의 한정자를 정의합니다.
- `getent` - 명령이 실행된 시스템에 있는 사용자 또는 역할의 한정 보안 속성을 나열합니다.
- `ldapaddent` - 사용자 또는 역할의 모든 한정 보안 속성을 나열합니다.

예 6-17 이 시스템에 있는 사용자의 한정 속성 목록

```
machine1$ getent user_attr | jdoe:
jdoe:machine1:::profiles=System Administrator
```

예 6-18 LDAP에 있는 사용자의 모든 한정 속성 목록

```
machine1$ ldapaddent -d user_attr | grep ^jdoe:
jdoe:machine1:::profiles=System Administrator
jdoe:sysopgroup:::profiles=System Operator
```

♦ ♦ ♦ 7 장

7 Oracle Solaris에서 권한 문제 해결

이 장에서는 Oracle Solaris에서 관리 권한을 관리 및 사용하는 경우의 문제 해결 제안 사항을 제공합니다.

권한 사용에 대한 자세한 내용은 다음 정보를 참조하십시오.

- 3장. Oracle Solaris에서 권한 지정
- “권한을 지정할 수 있는 사람” [41]
- “사용자 권한 관리” [14]
- “프로세스 권한 관리” [22]

권한 문제 해결

이 절의 작업과 예에서는 권한 지정 문제를 해결하는 방법을 제안합니다. 배경 정보는 “권한 확인” [32]을 참조하십시오.

▼ 권한 지정 문제를 해결하는 방법

권한이 평가되고 올바르게 적용되지 않는 이유에 영향을 주는 요소에는 여러 가지가 있습니다. 이 절차는 사용자, 역할 또는 프로세스에서 지정된 권한을 사용할 수 없는 이유를 디버그하는 데 도움이 됩니다. 일부 단계는 “지정된 권한 검색 순서” [32]를 기반으로 합니다.

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 “지정된 관리 권한 사용” [74]을 참조하십시오.

1. **이름 지정 서비스를 확인하고 다시 시작합니다.**

   a. **사용자나 역할에 대한 보안 지정이 시스템에서 사용으로 설정된 이름 지정 서비스에 속하는지 확인합니다.**

      ```
      # svccfg -s name-service/switch
      ```

```
svc:/system/name-service/switch>
listprop config

config                      application
config/value_authorization  astring  solaris.smf.value.name-service.switch
config/default              astring  files ldap
config/host                 astring  "files dns mdns ldap"
config/netgroup             astring  ldap
config/printer              astring  "user files"
```

이 출력에서 명시적으로 언급되지 않은 모든 서비스는 기본 `files ldap`의 값을 상속합니다. 따라서 `passwd` 및 관련 속성 데이터베이스인 `user_attr`, `auth_attr` 및 `prof_attr`이 파일에서 먼저 검색된 후 LDAP에서 검색됩니다.

b. **이름 서비스 캐시 `svc:/system/name-service/cache`를 다시 시작합니다.**

`nscd` 데몬의 TTL(time-to-live) 간격이 길어질 수 있습니다. 데몬을 다시 시작하여 이름 지정 서비스를 현재 데이터로 업데이트합니다.

```
# svcadm restart name-service/cache
```

2. **`userattr -v` 명령을 실행하여 사용자에게 권한이 지정된 위치를 확인합니다.**

예를 들어, 다음 명령은 사용자 `jdoe`에 대해 지정된 권한과 지정된 위치를 나타냅니다. `jdoe`가 기본값을 사용 중임을 나타내는 출력은 없습니다.

```
% userattr -v access_times jdoe
% userattr -v access_tz jdoe
% userattr -v auth_profiles jdoe
% userattr -v defaultpriv jdoe
% userattr -v limitpriv jdoe
% userattr -v idlecmd jdoe
% userattr -v idletime jdoe
% userattr -v lock_after_retries jdoe
% userattr -v pam_policy jdoe

% userattr -v auths jdoe       Output indicates authorizations from rights profiles
Basic Solaris User :solaris.mail.mailq,solaris.network.autoconf.read,
solaris.admin.wusb.read
Console User :solaris.system.shutdown,solaris.device.cdrw,
solaris.device.mount.removable,solaris.smf.manage.vbiosd,solaris.smf.value.vbiosd
% userattr -v audit_flags jdoe
user_attr: fw:no      Output indicates jdoe is individually assigned audit flags
# userattr -v profiles jdoe
user_attr: Audit Review,Stop       Output indicates two assigned rights profiles
# userattr roles jdoe
user_attr : cryptomgt,infosec       Output indicates two assigned roles
```

출력 결과는 `jdoe`에게 감사 플래그, 두 개의 권한 프로파일 및 두 개의 역할이 직접 지정되었음을 보여줍니다. 지정된 권한 부여는 `policy.conf` 파일의 기본 권한 프로파일에서 가져옵니다.

- jdoe에 감사 플래그가 직접 지정되었으므로 권한 프로파일의 감사 플래그 값은 사용되지 않습니다.
- 권한 프로파일은 순서대로 먼저 Audit Review 권한 프로파일이 평가된 후 Stop 프로파일이 평가됩니다.
- 다른 모든 권한은 cryptomgt 및 infosec 역할에서 jdoe에 지정됩니다. 이러한 권한을 보려면 jdoe가 각 역할을 맡은 후 권한을 나열해야 합니다.

사용자에게 직접 권한이 지정되지 않은 경우 다음 검사를 계속합니다.

3. **지정된 권한 부여의 철자가 올바른지 확인합니다.**

   권한 부여가 사용자에 대해 누적되므로 권한 부여 지정의 소스는 중요하지 않습니다. 하지만 철자가 잘못된 권한 부여는 자동으로 실패합니다.

4. **직접 만든 권한 프로파일의 경우 해당 프로파일의 명령에 적절한 보안 속성을 지정했는지 확인합니다.**

   예를 들어, 일부 명령이 성공하려면 euid=0이 아니라 uid=0이 필요합니다. 명령이나 해당 옵션에 권한 부여가 필요한지 확인하는 명령은 매뉴얼 페이지를 참조하십시오.

5. **사용자의 권한 프로파일에서 권한을 검사합니다.**

   a. **순서대로, 인증된 권한 프로파일 목록에서 권한을 검사합니다.**

      목록에서 가장 빠른 권한 프로파일의 속성 값이 커널의 값입니다. 이 값이 올바르지 않으면 해당 권한 프로파일의 값을 변경하거나 프로파일을 올바른 순서로 다시 지정합니다. 지정된 권한 순서를 조정하는 방법 [100]을 참조하십시오.

      권한 있는 명령의 경우 defaultpriv 또는 limitpriv 키워드에서 권한이 제거되지 않았는지 확인합니다.

   b. **순서대로, 일반 권한 프로파일 목록에서 권한을 검사합니다.**

      인증된 권한 프로파일에 대해 수행한 것과 동일한 검사를 따르십시오.

   c. **검색 중인 권한이 나열되지 않는 경우 사용자에게 지정된 역할을 확인합니다.**

      권한이 역할에 지정된 경우 권한을 얻으려면 사용자가 역할을 맡아야 합니다.

6. **실패한 명령이 성공하려면 권한 부여가 필요한지 확인합니다.**

   a. **기존 권한 프로파일에 필요한 권한 부여가 포함되어 있는지 확인합니다.**

      프로파일이 존재하면 사용합니다. 인증된 권한 프로파일이나 일반 권한 프로파일로 사용자에게 지정합니다. 성공하려면 이 권한 부여가 필요한 명령이 포함된 다른 권한 프로파일보다 이전 순서를 프로파일에 지정합니다.

   b. **명령에 대한 옵션에 권한 부여가 필요한지 확인합니다.**

권한(privilege)이 필요한 명령에 권한(privilege)을 지정하고, 필요한 권한 부여를 추가하고, 명령 및 권한 부여를 권한(right) 프로파일에 배치하고, 프로파일을 사용자에게 지정합니다.

7. **사용자에 대해 명령이 계속 실패하는 경우 사용자가 프로파일 셸에서 명령을 실행 중인지 확인합니다.**

   관리 명령은 프로파일 셸에서 실행해야 합니다. 예 7-1. "프로파일 셸을 사용 중인지 확인"에서는 프로파일 셸이 있는지 테스트하는 방법을 보여줍니다.

   사용자 오류 가능성을 줄이려면 다음을 시도할 수 있습니다.

   - 프로파일 셸을 사용자 로그인 셸로 지정합니다.
   - 사용자에게 모든 권한 있는 명령 앞에 `pfexec` 명령을 배치하도록 지시합니다.
   - 사용자에게 프로파일 셸에서 관리 명령을 실행하도록 미리 알립니다.
   - 사이트에서 역할을 사용 중인 경우 사용자에게 관리 명령을 실행하기 전에 역할을 맡도록 미리 알립니다. 사용자가 아니라 역할로서 성공한 명령 실행의 예는 예 7-3. "역할의 권한 있는 명령 실행"을 참조하십시오.

8. **역할에 대해 명령이 실패한 경우 역할을 맡고 사용자 권한을 검사할 때 수행한 것과 동일한 단계를 수행합니다.**

예 7-1 프로파일 셸을 사용 중인지 확인

권한 있는 명령이 작동하지 않는 경우 사용자가 `PRIV_PFEXEC` 플래그를 테스트한 후 명령을 실행합니다. 오류 메시지에 문제가 권한 문제라고 표시되지 않을 수도 있습니다.

```
% praudit 20120814200247.20120912213421.example-system
praudit: Cannot associate stdin with 20120814200247.20120912213421.example-system:
Permission denied

% ppriv $$
107219: bash
flags = <none>
...

% pfbash
# ppriv $$
1072232: bash
flags = PRIV_PFEXEC
...

# praudit 20120814200247.20120912213421.example-system
     /** Command succeeds **/
```

예 7-2 역할의 권한 있는 명령 확인

이 예에서는 사용자가 지정된 역할을 맡고 권한 프로파일 중 하나에 포함된 권한을 나열합니다. 명령을 강조하기 위해 권한이 잘렸습니다.

```
% roles
devadmin

% su - devadmin
Password: xxxxxxxx

# profiles -l
Device Security
        ...
 profiles=Service Configuration
          /usr/sbin/add_drv          uid=0
          /usr/sbin/devfsadm         uid=0
                                     privs=sys_devices,sys_config,
                                           sys_resource,file_owner,
                                           file_chown,file_chown_self,
                                           file_dac_read
          /usr/sbin/eeprom           uid=0
          /usr/bin/kbd
          /usr/sbin/list_devices     euid=0
          /usr/sbin/rem_drv          uid=0
          /usr/sbin/strace           euid=0
          /usr/sbin/update_drv       uid=0
          /usr/sbin/add_allocatable  euid=0
          /usr/sbin/remove_allocatable   euid=0
Service Configuration
          /usr/sbin/svcadm
          /usr/sbin/svccfg
```

예 7-3 역할의 권한 있는 명령 실행

다음 예에서는 admin 역할이 useful.script 파일에 대한 권한을 변경할 수 있습니다.

```
% whoami
jdoe
% ls -l useful.script
-rwxr-xr-- 1 elsee eng 262 Apr 2 10:52 useful.script

% chgrp admin useful.script
chgrp: useful.script: Not owner

% su - admin
Password: xxxxxxxx

# chgrp admin useful.script
# chown admin useful.script
# ls -l useful.script
-rwxr-xr-- 1 admin admin 262 Apr 2 10:53 useful.script
```

▼ 지정된 권한 순서를 조정하는 방법

사용자에 대해 권한 있는 버전이 아니라 권한 없는 명령이 적용되는 경우 사용자의 권한 프로파일 지정 순서를 조정해야 합니다. 자세한 내용은 "지정된 권한 검색 순서" [32]를 참조하십시오.

시작하기 전에 사용자는 User Security 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "지정된 관리 권한 사용" [74]을 참조하십시오.

1. **사용자 또는 역할에 현재 지정된 권한 프로파일의 목록을 봅니다.**
   목록은 순서대로 표시됩니다.

   ```
   % profiles username | rolename
   ```

2. **권한 프로파일을 올바른 순서로 지정합니다.**

   ```
   # usermod | rolemod -P "list-of-profiles"
   ```

예 7-4 특정 순서로 권한 프로파일 지정

이 예에서 관리자는 권한 있는 명령을 포함하는 권한 프로파일이 `devadmin` 역할의 All 권한 프로파일 다음에 나열되는지 확인합니다.

```
# profiles devadmin

    Basic Solaris User
    All
    Device Management
```

따라서 `devadmin` 역할은 역할의 지정된 권한으로 장치 관리 명령을 실행할 수 없습니다.

관리자가 권한 프로파일을 `devadmin`에 다시 지정합니다. 새로운 지정 순서에서는 장치 관리 명령이 지정된 권한으로 실행됩니다.

```
# rolemod -P "Device Management,Basic Solaris User,All"

# profiles devadmin

    Device Management
    Basic Solaris User
    All
```

▼ 프로그램에 필요한 권한을 확인하는 방법

명령이나 프로세스가 실패하는 경우 이 디버깅 절차를 사용합니다. 첫번째 권한 실패를 찾아서 수정한 후 `ppriv -eD` *command* 명령을 다시 실행하여 추가 권한 요구 사항을 찾아야 할 수도 있습니다.

1. **실패하는 명령을 `ppriv` 디버깅 명령에 대한 인수로 입력합니다.**

```
% ppriv -eD touch /etc/acct/yearly

touch[5245]: missing privilege "file_dac_write"
     (euid = 130, syscall = 224) needed at zfs_zaccess+0x258
touch: cannot create /etc/acct/yearly: Permission denied
```

2. **디버깅 출력의 `syscall` 번호를 사용하여 실패하는 시스템 호출을 확인합니다.**

   `/etc/name_to_sysnum` 파일에서 `syscall` 번호의 이름을 찾습니다.

```
% grep 224 /etc/name_to_sysnum

creat64                 224
```

   이 예에서는 `creat64()` 호출이 실패합니다. 성공하려면 `/etc/acct/yearly` 디렉토리에 파일을 만드는 권한을 프로세스에 지정해야 합니다.

예 7-5 `truss` 명령을 사용하여 권한 사용 조사

`truss` 명령은 일반 셸에서 권한 사용을 디버그할 수 있습니다. 예를 들어, 다음 명령은 실패하는 `touch` 프로세스를 디버그합니다.

```
% truss -t creat touch /etc/acct/yearly

creat64("/etc/acct/yearly", 0666)
                      Err#13 EACCES [file_dac_write
]
touch: /etc/acct/yearly cannot create
```

확장된 `/proc` 인터페이스가 `truss` 출력에서 오류 코드 뒤에 `file_dac_write` 권한이 누락되었다고 보고합니다.

예 7-6 `ppriv` 명령을 사용하여 프로파일 셸의 권한 사용 조사

이 예에서 `jdoe` 사용자는 `objadmin` 역할을 맡을 수 있습니다. `objadmin` 역할에는 Object Access Management 권한 프로파일이 포함됩니다. 이 권한 프로파일을 통해 `objadmin` 역할은 `objadmin`이 소유하지 않은 파일에 대한 권한을 변경할 수 있습니다.

다음 발췌 부분에서 `jdoe`가 `useful.script` 파일에 대한 권한 변경을 실패합니다.

```
jdoe% ls -l useful.script

-rw-r--r--  1 aloe  staff  2303 Apr 10 10:10 useful.script
jdoe%
chown objadmin useful.script

chown: useful.script: Not owner
jdoe%
ppriv -eD chown objadmin useful.script
```

```
chown[11444]: missing privilege "file_chown"
            (euid = 130, syscall = 16) needed at zfs_zaccess+0x258
chown: useful.script: Not owner
```

jdoe가 objadmin 역할을 맡을 때 파일에 대한 사용 권한이 변경됩니다.

```
jdoe% su - objadmin
Password: xxxxxxxx

# ls -l useful.script
-rw-r--r--  1 aloe  staff  2303 Apr 10 10:10 useful.script

# chown objadmin useful.script
# ls -l useful.script
-rw-r--r--  1 objadmin  staff  2303 Apr 10 10:10 useful.script
# chgrp admin useful.script

# ls -l objadmin.script
-rw-r--r--  1 objadmin  admin  2303 Apr 10 10:11 useful.script
```

예 7-7 root 사용자가 소유한 파일 변경

이 예는 권한 에스컬레이션에 대한 보호 조치를 보여줍니다. 자세한 내용은 “권한 에스컬레이션 및 커널 권한” [31]을 참조하십시오. 파일은 root 사용자가 소유합니다. 비교적 덜 강력한 역할인 objadmin이 파일 소유권을 변경하려면 모든 권한이 필요하므로 작업을 실패합니다.

```
jdoe% su - objadmin
Password: xxxxxxxx

# cd /etc; ls -l system
-rw-r--r--  1 root  sys   1883 Oct 10 10:20 system

# chown objadmin system
chown: system: Not owner
# ppriv -eD chown objadmin system
chown[11481]: missing privilege "ALL"
     (euid = 101, syscall = 16) needed at zfs_zaccess+0x258
chown: system: Not owner
```

♦ ♦ ♦ 8 장

8 Oracle Solaris 권한에 대한 참조

이 장에서는 Oracle Solaris의 관리 권한 사용에 대한 참조 자료를 제공합니다.

권한(privilege)을 비롯한 권한(right) 사용 방법에 대한 자세한 내용은 3장. Oracle Solaris에서 권한 지정을 참조하십시오. 개요 정보는 "사용자 권한 관리" [14] 및 "프로세스 권한 관리" [22]를 참조하십시오.

권한 프로파일 참조

이 절에서는 일반적인 권한 프로파일에 대해 설명합니다. 권한 프로파일은 권한 부여 및 기타 보안 속성, 보안 속성 포함 명령, 보충 권한 프로파일을 간편하게 모은 것입니다. Oracle Solaris는 많은 권한 프로파일을 제공합니다. 이들이 사용자 요구를 충족하지 않으면 기존 것을 수정하여 새로 만들 수 있습니다.

가장 강력한 권한 프로파일에서 가장 약한 순으로 지정되어야 합니다. 자세한 내용은 "지정된 권한 검색 순서" [32]를 참조하십시오.

다음 권한 프로파일의 내용을 보려면 "권한 프로파일의 내용 보기" [104]를 참조하십시오.

- **System Administrator 권한 프로파일** - 보안과 관련이 없는 대부분의 작업에 대한 액세스를 제공합니다. 이 프로파일에는 강력한 역할을 만들기 위한 여러 다른 프로파일이 포함됩니다. All 권한 프로파일은 보충 권한 프로파일 목록 끝에 지정됩니다.
- **Operator 권한 프로파일** - 파일 및 오프라인 매체를 관리하기 위한 제한된 권한을 제공합니다. 이 프로파일에는 단순한 역할을 만들기 위한 보충 권한 프로파일이 포함됩니다.
- **Printer Management 권한 프로파일** - 인쇄를 처리하기 위한 제한된 수의 명령 및 권한 부여를 제공합니다. 이 프로파일은 관리 분야를 다루는 여러 프로파일 중 하나입니다.
- **Basic Solaris User 권한 프로파일** - 사용자가 보안 정책의 범위 내에서 시스템을 사용할 수 있게 합니다. 이 프로파일은 `policy.conf` 파일에 기본적으로 나열됩니다. Basic Solaris User 권한 프로파일에서 제공하는 편의성은 사이트 보안 요구 사항과 균형을 이

루어야 합니다. 더 엄격한 보안이 필요한 사이트는 `policy.conf` 파일에서 이 프로파일을 제거하거나 Stop 권한 프로파일을 지정하는 것이 좋습니다. Basic Solaris User 권한 프로파일의 구현은 예 6-16. “권한 프로파일의 보안 속성 포함 명령 목록”을 참조하십시오.

- **Console User 권한 프로파일** - 워크스테이션 소유자에 대해 컴퓨터에 앉은 사용자의 권한 부여, 명령 및 작업 액세스를 제공합니다.
- **All 권한 프로파일** - 역할에 대해 보안 속성이 없는 명령 액세스를 제공합니다. 이 프로파일은 제한된 권한을 가진 사용자에 적합할 수 있습니다.
- **Stop 권한 프로파일** - 더 이상의 프로파일 평가를 중지하는 특수 권한 프로파일입니다. 이 프로파일은 `policy.conf` 파일에서 `AUTHS_GRANTED`, `PROFS_GRANTED`, `CONSOLE_USER` 변수의 평가를 금지합니다. 이 프로파일로 역할 및 사용자에게 제한된 프로파일 셸을 제공할 수 있습니다.

참고 - Stop 프로파일은 권한 지정에 간접적인 영향을 미칩니다. Stop 프로파일 후에 나열된 권한 프로파일은 평가되지 않습니다. 따라서 이러한 프로파일의 권한을 가진 명령은 효력이 없습니다. 예 3-25. “관리자를 명시적으로 지정된 권한으로 제한”를 참조하십시오.

각 권한 프로파일에는 연관된 도움말 파일이 있습니다. 도움말 파일은 HTML 형식이고 사용자 정의할 수 있습니다. 파일은 `/usr/lib/help/profiles/locale/C` 디렉토리에 상주합니다.

권한 프로파일의 내용 보기

권한 프로파일의 내용을 보기 위한 보기 3개가 있습니다.

- `getent` 명령으로 시스템에 있는 모든 권한 프로파일의 내용을 볼 수 있습니다. 샘플 출력은 6장. Oracle Solaris의 권한 목록을 참조하십시오.
- `profiles -p "`*Profile Name*`" info` 명령으로 특정 권한 프로파일의 내용을 볼 수 있습니다.
- `profiles -l` *account-name* 명령으로 특정 사용자나 역할에 지정된 권한 프로파일의 내용을 볼 수 있습니다.

자세한 내용은 6장. Oracle Solaris의 권한 목록, `getent`(1M) 및 `profiles`(1) 매뉴얼 페이지를 참조하십시오.

권한 부여 참조

권한 부여는 역할이나 사용자에게 부여할 수 있는 개별 권한입니다. 사용자가 응용 프로그램이나 응용 프로그램 내의 특정 작업에 액세스하기 전에 호환 응용 프로그램에서 권한 부여를 검사합니다.

권한 부여는 사용자 레벨이므로 확장 가능합니다. 권한 부여가 필요한 프로그램을 작성하고, 권한 부여를 시스템에 추가하고, 이러한 권한 부여에 대한 권한 프로파일을 만들고, 프로그램 사용이 허용된 사용자나 역할에 권한 프로파일을 지정할 수 있습니다.

권한 부여 이름 지정 규약

권한 부여는 내부적으로 사용되는 이름이 있습니다. 예를 들어, `solaris.system.date`는 권한 부여의 이름입니다. 권한 부여에는 GUI(그래픽 사용자 인터페이스)에 나타나는 간단한 설명이 있습니다. 예를 들어, `Set Date & Time`은 `solaris.system.date` 권한의 설명입니다.

규약상 권한 부여 이름은 인터넷 공급자 이름, 주제 영역, 하위 영역, 기능의 역순으로 구성됩니다. 권한 부여 이름의 부분은 점으로 구분됩니다. 그 예로 `com.xyzcorp.device.access`가 있습니다. 이 규약의 예외는 Oracle의 권한 부여로, 인터넷 이름 대신 접두어 `solaris`를 사용합니다. 이름 지정 규약에 따라 관리자는 계층적 방식으로 권한 부여를 적용할 수 있습니다. 와일드카드 문자(`*`)는 점 오른쪽의 문자열을 나타낼 수 있습니다.

권한 부여 사용 방법의 예로, Network Link Security 권한 프로파일에는 `solaris.network.link.security` 권한 부여만 포함되고 Network Security 권한 프로파일에는 보충 프로파일로 Network Link Security 프로파일뿐 아니라 `solaris.network.*` 및 `solaris.smf.manage.ssh` 권한 부여가 포함됩니다.

권한 부여의 위임 기관

접미어 `delegate`로 끝나는 권한 부여를 통해 사용자나 역할이 동일한 접두어로 시작하는 지정된 권한 부여를 다른 사용자에게 위임할 수 있습니다.

`solaris auth.delegate` 권한 부여는 사용자나 역할에 지정된 권한 부여를 다른 사용자에게 위임할 수 있습니다. 예를 들어, `solaris auth.delegate` 및 `solaris.network.wifi.wep` 권한 부여를 가진 역할은 `solaris.network.wifi.wep` 권한 부여를 다른 사용자나 역할에 위임할 수 있습니다.

권한 데이터베이스

다음 데이터베이스는 Oracle Solaris의 권한에 대한 데이터를 저장합니다.

- **확장된 사용자 속성 데이터베이스** (`user_attr`) - 다른 키워드 중에서 권한 부여, 권한(privilege) 및 권한(right) 프로파일과 사용자 및 역할을 연관시킵니다.
- **권한 프로파일 속성 데이터베이스** (`prof_attr`) - 권한(right) 프로파일을 정의하고 프로파일의 지정된 권한 부여, 권한(privilege), 키워드를 나열하고 연관된 도움말 파일을 식별합니다.

- **권한 부여 속성 데이터베이스** (auth_attr) - 권한 부여와 해당 속성을 정의하고 연관된 도움말 파일을 식별합니다.
- **실행 속성 데이터베이스** (exec_attr) - 특정 권한 프로파일에 지정된 보안 속성 포함 명령을 식별합니다.

policy.conf 데이터베이스에는 모든 사용자에게 적용되는 권한 부여, 권한(privilege) 및 권한(right) 프로파일이 포함됩니다. 자세한 내용은 "policy.conf 파일" [108]을 참조하십시오.

권한 데이터베이스 및 이름 지정 서비스

권한 데이터베이스의 이름 서비스 범위는 이름 지정 서비스 스위치 svc:/system/name-service/switch에 대한 SMF 서비스에 정의됩니다. 권한 데이터베이스에 대한 이 서비스의 등록 정보는 auth_attr, password, prof_attr입니다. password 등록 정보는 passwd 및 user_attr 데이터베이스에 대한 이름 지정 서비스 우선 순위를 설정합니다. prof_attr 등록 정보는 prof_attr 및 exec_attr 데이터베이스에 대한 이름 지정 서비스 우선 순위를 설정합니다.

다음 출력에서 auth_attr, password, prof_attr 항목이 나열되지 않습니다. 따라서 권한 데이터베이스는 files 이름 지정 서비스를 사용하고 있습니다.

```
# svccfg -s name-service/switch listprop config
config                          application
config/value_authorization      astring       solaris.smf.value.name-service.switch
config/default                  astring       files
config/host                     astring       "files ldap dns"
config/printer                  astring       "user files ldap"
```

user_attr 데이터베이스

user_attr 데이터베이스는 passwd 및 shadow 데이터베이스를 보충하는 사용자 및 역할 정보를 포함합니다. *attr* 필드에는 보안 속성이 포함되고 *qualifier* 필드에는 보안 속성의 영향을 시스템 또는 시스템 그룹으로 한정하거나 제한하는 속성이 포함됩니다.

attr 필드의 보안 속성은 roleadd, rolemod, useradd, usermod, profiles 명령으로 설정할 수 있습니다. LDAP 이름 지정 범위와 로컬에서 설정할 수 있습니다.

- 사용자의 경우 roles 키워드가 하나 이상의 정의된 역할을 지정합니다.
- 역할의 경우 roleauth 키워드의 user 값을 사용하여 역할 암호가 아니라 사용자 암호로 인증할 수 있습니다. 기본적으로 값은 role입니다.
- 사용자 또는 역할의 경우 다음 속성을 설정할 수 있습니다.
  - access_times 키워드 - 지정한 응용 프로그램과 서비스에 액세스할 수 있는 요일 및 시간을 지정합니다. 자세한 내용은 getaccess_times(3C) 매뉴얼 페이지를 참조하십시오.

- access_tz 키워드 - access_times 항목의 시간을 해석할 때 사용할 시간대를 지정합니다. 자세한 내용은 pam_unix_account(5) 매뉴얼 페이지를 참조하십시오.
- audit_flags 키워드 - 감사 마스크를 수정합니다. 자세한 내용은 audit_flags(5) 매뉴얼 페이지를 참조하십시오.
- auths 키워드 - 권한 부여를 지정합니다. 자세한 내용은 auths(1) 매뉴얼 페이지를 참조하십시오.
- auth_profiles 키워드 - 인증된 권한 프로파일을 지정합니다. 참조는 profiles(1) 매뉴얼 페이지를 참조하십시오.
- defaultpriv 키워드 - 기본 권한 세트에서 권한을 추가하거나 제거합니다.
- limitpriv 키워드 - 기본값의 제한 권한 세트에서 권한을 추가하거나 제거합니다.

  defaultpriv 및 limitpriv 권한은 사용자의 초기 프로세스에 지정되므로 항상 적용됩니다. 자세한 내용은 privileges(5) 매뉴얼 페이지와 "권한이 구현되는 방법" [25]을 참조하십시오.
- idlecmd 키워드 - idletime에 도달한 후 사용자를 로그아웃하거나 화면을 잠급니다.
- idletime 키워드 - 키보드 활동이 없는 시점 이후 시스템을 사용할 수 있는 시간을 설정합니다. idlecmd 값을 지정할 때 idletime을 설정합니다.
- lock_after_retries 키워드 - 값이 yes인 경우 재시도 횟수가 /etc/default/login 파일에 허용된 수를 초과하면 시스템이 잠깁니다. 자세한 내용은 login(1) 매뉴얼 페이지를 참조하십시오.
- profiles 키워드 - 권한 프로파일을 지정합니다. 자세한 내용은 profiles(1) 매뉴얼 페이지를 참조하십시오.
- project 키워드 - 기본 프로젝트를 추가합니다. 자세한 내용은 project(4) 매뉴얼 페이지를 참조하십시오.

참고 - access_times 및 access_tz 속성은 PAM 속성이므로 인증 중에 확인됩니다. 따라서 사용자나 역할에 직접 또는 인증된 권한 프로파일에서 지정해야 합니다. 일반적인 권한 프로파일에서는 무시됩니다.

LDAP 이름 지정 범위에서만 사용자와 역할에 대해 한정 속성을 설정할 수 있습니다. 이러한 한정자는 권한 프로파일과 같은 사용자 또는 역할의 속성 지정을 하나 이상의 시스템으로 제한합니다. 예는 useradd(1M) 및 user_attr(4) 매뉴얼 페이지를 참조하십시오.

한정자는 host 및 netgroup입니다.

- host 한정자 - 사용자나 역할이 지정된 작업을 수행할 수 있는 시스템을 식별합니다.
- netgroup 한정자 - 사용자나 역할이 지정된 작업을 수행할 수 있는 시스템을 나열합니다. host 지정이 netgroup 지정보다 우선합니다.

자세한 내용은 user_attr(4) 매뉴얼 페이지를 참조하십시오. 이 데이터베이스의 내용을 보려면 getent user_attr 명령을 사용하십시오. 자세한 내용은 getent(1M) 매뉴얼 페이지와 6장, Oracle Solaris의 권한 목록을 참조하십시오.

auth_attr 데이터베이스

auth_attr 데이터베이스는 권한 부여 정의를 저장합니다. 권한 부여는 사용자, 역할 또는 권한 프로파일에 지정할 수 있습니다. 선호 방법은 권한 프로파일에 권한 부여를 배치한 후 역할이나 사용자에게 권한 프로파일을 지정하는 것입니다.

이 데이터베이스의 내용을 보려면 getent auth_attr 명령을 사용하십시오. 자세한 내용은 getent(1M) 매뉴얼 페이지와 6장. Oracle Solaris의 권한 목록을 참조하십시오.

prof_attr 데이터베이스

prof_attr 데이터베이스는 권한 프로파일에 지정된 이름, 설명, 도움말 파일 위치, 권한 및 권한 부여를 저장합니다. 권한 프로파일에 지정된 명령 및 보안 속성은 exec_attr 데이터베이스에 저장됩니다. 자세한 내용은 "exec_attr 데이터베이스" [108]를 참조하십시오.

자세한 내용은 prof_attr(4) 매뉴얼 페이지를 참조하십시오. 이 데이터베이스의 내용을 보려면 getent exec_attr 명령을 사용하십시오. 자세한 내용은 getent(1M) 매뉴얼 페이지와 6장. Oracle Solaris의 권한 목록을 참조하십시오.

exec_attr 데이터베이스

exec_attr 데이터베이스는 성공을 위해 보안 속성이 필요한 명령을 정의합니다. 명령은 권한 프로파일의 일부입니다. 보안 속성 포함 명령은 프로파일을 지정받은 역할이나 사용자가 실행할 수 있습니다.

자세한 내용은 exec_attr(4) 매뉴얼 페이지를 참조하십시오. 이 데이터베이스의 내용을 보려면 getent 명령을 사용하십시오. 자세한 내용은 getent(1M) 매뉴얼 페이지와 6장. Oracle Solaris의 권한 목록을 참조하십시오.

policy.conf 파일

/etc/security/policy.conf 파일은 특정 권한(right) 프로파일, 특정 권한 부여, 특정 권한(privilege)을 시스템의 모든 사용자에게 부여하는 방법을 제공합니다. 파일의 관련 항목은 키=값 쌍으로 구성됩니다.

- AUTHS_GRANTED=*authorizations* - 하나 이상의 권한 부여를 가리킵니다.
- AUTH_PROFS_GRANTED=*rights profiles* - 하나 이상의 인증된 권한 프로파일을 가리킵니다.

- PROFS_GRANTED=*rights profiles* - 인증되지 않은 하나 이상의 권한 프로파일을 가리킵니다.
- CONSOLE_USER=Console User - Console User 권한 프로파일을 가리킵니다. 이 프로파일은 콘솔 사용자를 위한 편리한 권한 부여 세트와 함께 제공됩니다. 이 프로파일을 사용자 정의할 수 있습니다.
- PRIV_DEFAULT=*privileges* - 하나 이상의 권한을 가리킵니다.
- PRIV_LIMIT=*privileges* - 모든 권한을 가리킵니다.

다음 예에서는 policy.conf 데이터베이스의 일부 권한 값을 보여줍니다.

```
##
AUTHS_GRANTED=
AUTH_PROFS_GRANTED=
CONSOLE_USER=Console User
PROFS_GRANTED=Basic Solaris User
#PRIV_DEFAULT=basic
#PRIV_LIMIT=all
```

권한 관리 명령

이 절에서는 권한 관리에 사용되는 명령을 나열합니다. 또한 권한 부여로 액세스를 제어할 수 있는 명령 표가 포함되어 있습니다.

권한 부여, 권한 프로파일 및 역할을 관리하는 명령

다음 표에 나열된 명령은 사용자 프로세스에 대한 권한을 검색 및 설정합니다.

표 8-1 권한 관리 명령

명령	설명
auths(1)	사용자에 대한 권한 부여를 표시합니다. 새 권한 부여를 만듭니다.
getent(1M)	권한 데이터베이스 내용을 나열합니다.
nscd(1M)	이름 서비스 캐시 데몬으로, 권한 데이터베이스를 캐싱하는 데 유용합니다. 데몬을 다시 시작하려면 svcadm 명령을 사용합니다.
pam_roles(5)	PAM용 역할 계정 관리 모듈입니다. 역할을 맡기 위해 권한 부여를 검사합니다.
pam_unix_account(5)	PAM용 UNIX 계정 관리 모듈입니다. 시간 제한, 비활성 등의 계정 제한 사항을 검사합니다.
pfbash(1)	권한을 평가할 수 있는 프로파일 셸 프로세스를 만드는 데 사용됩니다.
pfedit(1M)	관리 파일을 편집하는 데 사용됩니다.
pfexec(1)	보안 속성 포함 명령을 실행하는 데 사용됩니다.

명령	설명
policy.conf(4)	시스템 보안 정책에 대한 구성 파일입니다. 부여된 권한 부여, 부여된 권한 및 기타 보안 정보를 나열합니다.
profiles(1)	지정된 사용자에 대한 권한 프로파일을 표시합니다. 권한 프로파일을 만들거나 수정합니다.
roles(1)	지정된 사용자가 맡을 수 있는 역할을 표시합니다.
roleadd(1M)	로컬 시스템 또는 LDAP 네트워크에 역할을 추가합니다.
roleadd(1M)	로컬 시스템 또는 LDAP 네트워크에 역할을 추가합니다.
rolemod(1M)	로컬 시스템 또는 LDAP 네트워크에서 역할의 등록 정보를 수정합니다.
userattr(1)	사용자나 역할 계정에 지정된 특정 권한의 값을 표시합니다.
useradd(1M)	시스템 또는 LDAP 네트워크에 사용자 계정을 추가합니다. -R 옵션은 사용자의 계정에 역할을 지정합니다.
userdel(1M)	시스템 또는 LDAP 네트워크에서 사용자 로그인을 삭제합니다.
usermod(1M)	시스템에서 사용자의 계정 등록 정보를 수정합니다.

권한 부여가 필요한 선택된 명령

다음 표는 Oracle Solaris 시스템에서 명령 옵션을 제한하기 위해 권한 부여가 사용되는 방법의 예를 제공합니다. 권한 부여에 대한 자세한 내용은 “권한 부여 참조” [104]를 참조하십시오.

표 8-2 명령 및 연관된 권한 부여

명령	권한 부여 요구 사항
at(1)	solaris.jobs.user - 모든 옵션에 필요합니다(at.allow 또는 at.deny 파일이 존재하지 않는 경우).
atq(1)	solaris.jobs.admin - 모든 옵션에 필요합니다.
cdrw(1)	solaris.device.cdrw - 모든 옵션에 필요하고 policy.conf 파일에서 기본적으로 부여됩니다.
crontab(1)	solaris.jobs.user - 작업을 제출하는 옵션에 필요합니다(crontab.allow 파일 또는 crontab.deny 파일이 존재하지 않는 경우). solaris.jobs.admin - 다른 사용자의 crontab 파일을 나열/수정하는 옵션에 필요합니다.
allocate(1)	solaris.device.allocate(또는 기타 device_allocate 파일에 지정된 권한 부여) - 장치를 할당하려면 필요합니다. solaris.device.revoke(또는 기타 device_allocate 파일에 지정된 권한 부여) - 다른 사용자에게 장치를 할당하는 데 필요합니다(-F 옵션).
deallocate(1)	solaris.device.allocate(또는 기타 device_allocate 파일에 지정된 권한 부여) - 다른 사용자의 장치 할당을 해제하는 데 필요합니다. solaris.device.revoke(또는 기타 device_allocate 파일에 지정된 권한 부여) - 지정된 장치(-F 옵션) 또는 모든 장치(-I 옵션)를 강제 할당 해제하려면 필요합니다.
list_devices(1)	solaris.device.revoke - 다른 사용자의 장치를 나열하는 데 필요합니다(-U 옵션).

명령	권한 부여 요구 사항
roleadd(1M)	solaris.user.manage - 역할을 만드는 데 필요합니다. solaris.account.activate - 초기 암호를 설정하는 데 필요합니다. solaris.account.setpolicy - 계정 잠금 및 암호 에이징과 같은 암호 정책을 설정하는 데 필요합니다.
roledel(1M)	solaris.passwd.assign 권한 - 암호를 삭제하려면 필요합니다.
rolemod(1M)	solaris.passwd.assign 권한 - 암호를 변경하려면 필요합니다. solaris.account.setpolicy - 계정 잠금 및 암호 에이징과 같은 암호 정책을 변경하려면 필요합니다.
sendmail(1M)	solaris.mail - 메일 부속 시스템 기능에 액세스하는 데 필요합니다. solaris.mail.mailq - 메일 대기열을 보는 데 필요합니다.
useradd(1M)	solaris.user.manage - 사용자를 만드는 데 필요합니다. solaris.account.activate - 초기 암호를 설정하는 데 필요합니다. solaris.account.setpolicy - 계정 잠금 및 암호 에이징과 같은 암호 정책을 설정하는 데 필요합니다.
userdel(1M)	solaris.passwd.assign 권한 - 암호를 삭제하려면 필요합니다.
usermod(1M)	solaris.passwd.assign 권한 - 암호를 변경하려면 필요합니다. solaris.account.setpolicy - 계정 잠금 및 암호 에이징과 같은 암호 정책을 변경하려면 필요합니다.

권한 참조

권한 제약 프로세스는 커널에서 구현되며 명령, 사용자, 역할, 시스템 레벨에서 프로세스를 제약할 수 있습니다.

권한 처리용 명령

다음 표에는 권한 처리에 사용할 수 있는 명령이 나열되어 있습니다.

표 8-3 권한 처리용 명령

용도	명령	매뉴얼 페이지
권한 실패 디버그	ppriv -eD *failed-operation*	ppriv(1)
시스템에 권한 나열	ppriv -l	ppriv(1)
권한 및 해당 설명 나열	ppriv -lv *priv*	ppriv(1)
UID, 프로세스 또는 포트에 확장 권한 정책을 나열합니다	ppriv -lv *extended-policy*	ppriv(1)
프로세스 권한 조사	ppriv -v *pid*	ppriv(1)
UID, 프로세스 또는 포트에 확장 권한 정책 추가	ppriv -r *rule*	privileges(5)
프로세스 권한 설정	ppriv -s *spec*	ppriv(1)
확장 권한 정책 규칙 제거	ppriv -X *rule*	privileges(5)
권한 프로파일에 권한 지정	profiles -p *profile-name*	profiles(1)
새 역할에 권한 지정	roleadd -K defaultpriv=	roleadd(1M)

용도	명령	매뉴얼 페이지
기존 역할에 권한 추가	`rolemod -K defaultpriv+=`	`rolemod`(1M)
새 사용자에 권한 지정	`useradd -K defaultpriv=`	`useradd`(1M)
기존 사용자에 권한 추가	`usermod -K defaultpriv+=`	`usermod`(1M)
장치에 장치 정책 추가	`add_drv -p` *policy driver*	`add_drv`(1M)
장치 정책 설정	`devfsadm`	`devfsadm`(1M)
장치 정책 보기	`getdevpolicy`	`getdevpolicy`(1M)
열린 장치에서 장치 정책 업데이트	`update_drv -p` *policy driver*	`update_drv`(1M)

권한 정보를 포함하는 파일

`policy.conf` 및 `syslog.conf` 파일에는 권한 정보가 포함되어 있습니다.

- `/etc/security/policy.conf`에는 다음 권한 정보가 포함되어 있습니다.
  - `PRIV_DEFAULT` - 시스템의 상속 가능한 권한 세트
  - `PRIV_LIMIT` - 시스템의 제한 권한 세트

  자세한 내용은 `policy.conf`(4) 매뉴얼 페이지를 참조하십시오.
- `/etc/syslog.conf`는 권한 디버깅과 관련된 디버그 메시지에 대한 시스템 로그 파일입니다. 디버그 메시지의 경로는 `priv.debug` 항목에서 설정됩니다.

  자세한 내용은 `syslog.conf`(4) 매뉴얼 페이지를 참조하십시오.

감사 레코드의 권한 있는 작업

권한 사용을 감사할 수 있습니다. 프로세스가 권한을 사용할 때 언제든지 `upriv` 감사 토큰의 감사 증적에 권한 사용이 기록됩니다. 권한 이름이 레코드의 일부일 때 텍스트 표현이 사용됩니다. 다음 감사 이벤트는 권한 사용을 기록합니다.

- `AUE_SETPPRIV` **감사 이벤트** - 권한 세트를 변경할 때 감사 레코드를 생성합니다. `AUE_SETPPRIV` 감사 이벤트는 `pm` 클래스에 속합니다.
- `AUE_MODALLOCPRIV` **감사 이벤트** - 커널 밖에서 권한을 추가할 때 감사 레코드를 생성합니다. `AUE_MODALLOCPRIV` 감사 이벤트는 `ad` 클래스에 속합니다.
- `AUE_MODDEVPLCY` **감사 이벤트** - 장치 정책을 변경할 때 감사 레코드를 생성합니다. `AUE_MODDEVPLCY` 감사 이벤트는 `ad` 클래스에 속합니다.
- `AUE_PFEXEC` **감사 이벤트** - `pfexec()`가 사용으로 설정된 채 `execve()`를 호출할 때 감사 레코드를 생성합니다. `AUE_PFEXEC` 감사 이벤트는 `as`, `ex`, `ps`, `ua` 감사 클래스에 속합니다. 권한의 이름이 감사 레코드에 포함됩니다.

기본 세트에 속하는 권한의 성공적 사용은 감사되지 않습니다. 사용자의 기본 세트에서 제거된 기본 권한을 사용하려는 시도는 감사됩니다.

보안 용어

공개 키 기술에 대한 정책 키 관리 프레임워크(KMF)에서 정책은 인증서 사용을 관리합니다. KMF 정책 데이터베이스는 KMF 라이브러리에서 관리되는 키 및 인증서 사용을 제약할 수 있습니다.

공개 키 암호화 각 사용자가 두 개의 키, 즉 하나의 공개 키와 하나의 개인 키를 사용하는 암호화 체계입니다. 공개 키 암호화에서 발신자는 수신자의 공개 키를 사용하여 메시지를 암호화하고, 수신자는 개인 키를 사용하여 암호를 해독합니다. Kerberos 서비스는 개인 키 시스템입니다. private-key encryption(개인 키 암호화)도 참조하십시오.

관리자 주체 *username*`/admin`(예: `jdoe/admin`) 형식의 이름을 가진 사용자 주체입니다. 관리자 주체는 일반 사용자 주체보다 더 많은 권한(예: 정책 변경)을 가질 수 있습니다. principal name(주체 이름), user principal(사용자 주체)도 참조하십시오.

기본 주체 이름의 첫번째 부분입니다. 인스턴스, principal name(주체 이름), realm(영역)도 참조하십시오.

네트워크 애플리케이션 서버 `ftp`와 같은 네트워크 응용 프로그램을 제공하는 서버입니다. 영역에 여러 네트워크 애플리케이션 서버를 포함할 수 있습니다.

네트워크 정책 네트워크 트래픽을 보호하기 위해 네트워크 유틸리티가 구성하는 설정입니다. 네트워크 보안에 대한 자세한 내용은 "Oracle Solaris 11.2의 네트워크 보안 "을 참조하십시오.

다이제스트 message digest(메시지 다이제스트)를 참조하십시오.

동기 감사 이벤트 감사 이벤트의 다수를 차지합니다. 이러한 이벤트는 시스템의 프로세스와 연관됩니다. 프로세스와 연관된 출처를 알 수 없는 이벤트는 실패한 로그인과 같은 동기 이벤트입니다.

마스터 KDC 각 영역의 주 KDC로, Kerberos 관리 서버인 `kadmind`와 인증 및 티켓 부여 데몬인 `krb5kdc`를 포함합니다. 각 영역에는 적어도 하나의 마스터 KDC가 있어야 하고, 클라이언트에 인증 서비스를 제공하는 많은 중복된 슬레이브 KDC를 포함할 수 있습니다.

무결성 사용자 인증과 더불어, 암호화 체크섬을 통해 전송된 데이터의 유효성을 제공하는 보안 서비스입니다. authentication(인증), privacy(프라이버시)도 참조하십시오.

문장암호 개인 키를 문장암호 사용자가 만들었는지 확인하는 데 사용되는 문구입니다. 좋은 문장암호는 10-30자 길이로, 영문자와 숫자를 섞어서 만들고 단순한 문구와 단순한 이름을 피합니다. 통신을 암호화 및 해독하기 위해 개인 키 사용을 인증하려면 문장암호를 묻는 메시지가 나타납니다.

보안 서비스 서비스를 참조하십시오.

보안 정책 policy(정책)을 참조하십시오.

상속 가능한 세트 exec의 호출에서 프로세스가 상속할 수 있는 권한 세트입니다.

서버 주체 (RPCSEC_GSS API) 서비스를 제공하는 주체입니다. 서버 주체는 *service@host* 형식으로 ASCII 문자열로 저장됩니다. 클라이언트 주체도 참조하십시오.

서비스 1. 종종 여러 대의 서버에 의해 네트워크 클라이언트에 제공된 리소스입니다. 예를 들어, `central.example.com` 시스템에 `rlogin`을 제공하는 경우 해당 시스템은 `rlogin` 서비스를 제공하는 서버입니다.

2. 인증 외의 보호 레벨을 제공하는 보안 서비스(무결성 또는 프라이버시)입니다. 무결성 및 privacy(프라이버시)를 참조하십시오.

서비스 주체 서비스에 Kerberos 인증을 제공하는 주체입니다. 서비스 주체의 경우 기본 이름은 `ftp`와 같은 서비스 이름이고, 해당 인스턴스는 서비스를 제공하는 시스템의 정규화된 호스트 이름입니다. host principal(호스트 주체), user principal(사용자 주체)도 참조하십시오.

서비스 키 서비스 주체 및 KDC에서 공유되고 시스템 한도 밖에서 배포되는 암호화 키입니다. key(키)를 참조하십시오.

수퍼 유저 모델 컴퓨터 시스템의 전형적인 UNIX 보안 모델입니다. 수퍼 유저 모델에서 관리자는 all-or-nothing 방식으로 시스템을 제어합니다. 일반적으로 시스템을 관리하려는 경우 사용자는 수퍼 유저(`root`)로 로그인하여 모든 관리 작업을 수행할 수 있습니다.

알고리즘 암호화 알고리즘입니다. 입력을 암호화하거나 해시하는 확립된 순환적 계산 프로시저입니다.

암호 정책 암호를 생성하는 데 사용할 수 있는 암호화 알고리즘입니다. 암호 변경 주기, 암호 시도 허용 회수, 기타 보안 고려 사항 등 암호와 관련한 일반적인 사안이라고 할 수 있습니다. 보안 정책에 암호가 필요합니다. 암호 정책에서는 암호를 AES 알고리즘으로 암호화해야 하고, 추가로 암호 강도와 관련된 요구 사항이 필요할 수 있습니다.

암호화 알고리즘 알고리즘을 참조하십시오.

암호화 프레임워크의 정책 Oracle Solaris의 암호화 프레임워크 기능에서 정책은 기존 암호화 방식을 사용 안함으로 설정합니다. 그러면 방식을 사용할 수 없습니다. 암호화 프레임워크의 정책은 DES와 같은 공급자가 `CKM_DES_CBC`와 같은 특정 방식을 사용하는 것을 금지할 수 있습니다.

액세스 제어 목록(ACL) 액세스 제어 목록(ACL)은 전통적인 UNIX 파일 보호보다 좀 더 세부적인 파일 보안을 제공합니다. 예를 들어, ACL을 사용하여 파일에 그룹 읽기 액세스를 허용하면서 해당 그룹의 한 멤버만 파일 쓰기를 허용할 수 있습니다.

유효 세트 현재 프로세스에 발효 중인 권한 세트입니다.

이름 서비스 범위 역할이 작동하도록 허가된 범위입니다. 즉, NIS LDAP와 같은 지정된 이름 지정 서비스에서 제공하는 개별 호스트 또는 모든 호스트를 말합니다.

인스턴스 주체 이름의 두번째 부분으로, 인스턴스는 주체의 기본 부분을 한정합니다. 서비스 주체의 경우 인스턴스는 필수 사항입니다. 인스턴스는 `host/central.example.com`과 같이 호스트의 정규화된 도메인 이름입니다. 사용자 주체의 경우 인스턴스는 선택 사항입니다. 그러나 `jdoe` 및 `jdoe/admin`은 고유한 주체입니다. 기본, principal name(주체 이름), 서비스 주체, user principal(사용자 주체)도 참조하십시오.

잘못된 티켓 아직 사용할 수 없는 후일자 티켓입니다. 잘못된 티켓은 유효해질 때까지 애플리케이션 서버에서 거부합니다. 유효화하려면 시작 시간이 지난 후에 `VALIDATE` 플래그 세트를 사용하여 TGS 요청의 클라이언트가 KDC에 잘못된 티켓을 제시해야 합니다. postdated ticket(후일자 티켓)도 참조하십시오.

장치 정책 커널 레벨의 장치 보호입니다. 장치 정책은 장치에 두 개의 권한 세트로 구현됩니다. 첫번째 권한 세트는 장치의 읽기 액세스를 제어합니다. 두번째 권한 세트는 장치의 쓰기 액세스를 제어합니다. policy(정책)을 참조하십시오.

장치 할당 사용자 레벨의 장치 보호입니다. 장치 할당은 하나의 장치를 한번에 한 사용자만 배타적으로 사용하도록 합니다. 장치 재사용 전에 장치 데이터를 비웁니다. 권한 부여를 사용하여 장치 할당이 허가된 사용자를 제한할 수 있습니다.

제한 세트 프로세스와 그 자식에 사용 가능한 권한에 대한 외부 제한입니다.

주체 1. 네트워크 통신에 참여하는 고유한 이름이 지정된 클라이언트/사용자 또는 서버/서비스입니다. Kerberos 트랜잭션에는 주체들(서비스 주체 및 사용자 주체) 간의 상호 작용 또는 주체와 KDC 간의 상호 작용이 관여합니다. 다신 말해서, 주체는 Kerberos가 티켓을 지정할 수 있는 고유한 개체입니다. principal name(주체 이름), 서비스 주체, user principal(사용자 주체)도 참조하십시오.

2. (RPCSEC_GSS API) 클라이언트 주체, 서버 주체를 참조하십시오.

초기 티켓 (기존 TGT(티켓 부여 티켓)에 기반하지 않고) 직접 발행된 티켓입니다. 암호를 변경하는 응용 프로그램과 같은 일부 서비스는 `initial`로 표시된 티켓이 필요할 수 있으므로 클라이언트가 보안 키를 알고 있다는 것을 스스로 보증해야 합니다. 초기 티켓은 클라이언트가 (오랫동안 존재해 왔던 TGT(티켓 부여 티켓)에 의존하는 대신) 최근에 자체 인증되었음을 나타내므로 이 보증은 매우 중요합니다.

최소 권한의 원칙 최소한의 특권을 참조하십시오.

최소한의 특권 지정된 프로세스를 일부 수퍼 유저 권한에만 제공하는 보안 모델입니다. 최소 권한 모델은 일반 사용자가 파일 시스템 마운트 및 파일 소유권 변경과 같은 개인적인 관리 작업을 수행할 수 있도록 충분한 권한을 지정합니다. 반면에 프로세스는 완전한 수퍼 유저 권한(즉 모든 권한)이 아닌, 작업 완료에 필요한 권한으로만 실행됩니다. 버퍼 오버플로우 같은 프로그래밍 오류로 인한 손해는, 보호된 시스템 파일 읽기/쓰기 또는 시스템 정지 같은 중요한 능력에 액세스할 수 없는 비루트 사용자에 국한될 수 있습니다.

최소화 서버 실행에 필요한 최소 운영 체제를 설치합니다. 서버 작동에 직접적인 관련이 없는 소프트웨어는 설치되지 않거나 설치 후 삭제됩니다.

출처를 알 수 없는 감사 이벤트 AUE_BOOT 이벤트와 같이 개시자가 결정할 수 없는 감사 이벤트입니다.

클라이언트 좁은 의미로, 사용자 대신 네트워크 서비스를 이용하는 프로세스입니다. 예를 들어, `rlogin`을 사용하는 응용 프로그램이 있습니다. 어떤 경우 서버 자체가 다른 서버나 서비스의 클라이언트가 될 수 있습니다.

더 넓은 의미로, a) Kerberos 자격 증명을 수신하고 b) 서버에서 제공한 서비스를 이용하는 호스트입니다.

간단히 말하면, 서비스를 이용하는 주체입니다.

클라이언트 주체 (RPCSEC_GSS API) RPCSEC_GSS로 보안된 네트워크 서비스를 사용하는 클라이언트(사용자 또는 응용 프로그램)입니다. 클라이언트 주체 이름은 `rpc_gss_principal_t` 구조 형태로 저장됩니다.

클럭 불균형 Kerberos 인증 시스템에 참여하는 모든 호스트의 내부 시스템 클럭에 차이가 날 수 있는 최대 시간입니다. 참여하는 호스트 사이에 클럭 불균형을 초과할 경우 요청이 거부됩니다. 클럭 불균형은 `krb5.conf` 파일에 지정할 수 있습니다.

티켓 사용자의 신원을 서버나 서비스로 안전하게 전달하는 데 사용되는 정보 패킷입니다. 티켓은 단일 클라이언트에만, 그리고 특정 서버의 특정 서비스에만 유효합니다. 티켓에는 서비스의 주체 이름, 사용자의 주체 이름, 사용자 호스트의 IP 주소, 시간 기록, 티켓의 수명을 정의하는 값이 포함됩니다. 클라이언트 및 서비스에서 사용할 무작위 세션 키로 티켓이 생성됩니다. 일단 티켓이 만들어지면 만료될 때까지 재사용할 수 있습니다. 티켓은 새로운 인증자와 함께 제시될 때 클라이언트 인증에만 사용됩니다. authenticator(인증자), credential(자격 증명), 서비스, session key(세션 키)를 참조하십시오.

티켓 파일 credential cache(자격 증명 캐시)를 참조하십시오.

AES Advanced Encryption Standard의 머리글자어로, 고급 암호화 표준입니다. 대칭 128비트 블록 데이터 암호화 기술입니다. 미국 정부는 2000년 10월 알고리즘의 Rijndael 변형을 암호화 표준으로 채택했습니다. AES가 정부 표준으로 user principal(사용자 주체) 암호화를 대체합니다.

application server(애플리케이션 서버) 네트워크 애플리케이션 서버를 참조하십시오.

asynchronous audit event(비동기 감사 이벤트) 비동기 이벤트는 시스템 이벤트의 소수를 차지합니다. 이러한 이벤트는 어떤 프로세스와 연관되지 않으므로 프로세스를 차단했다가 나중에 깨울 수 없습니다. 초기 시스템 부트 및 PROM 진입/종료 이벤트가 비동기 이벤트의 예입니다.

audit files(감사 파일) 이진 감사 로그입니다. 감사 파일은 감사 파일 시스템에 별도로 저장됩니다.

audit policy(감사 정책) 어떤 감사 이벤트를 기록할지 결정하는 전역 사용자별 설정입니다. 감사 서비스에 적용되는 전역 설정은 일반적으로 감사 추적에 포함할 선택적 정보 조각에 영향을 미칩니다. 두 설정 `cnt` 및 `ahlt`는 감사 대기열을 채울 때 시스템의 작업에 영향을 미칩니다. 예를 들어, 감사 정책에서 시퀀스 번호가 모든 감사 레코드에 속하도록 요구할 수 있습니다.

audit trail(감사 증적) 모든 호스트의 모든 감사 파일 모음입니다.

authenticated rights profile(인증된 권한 프로파일) 지정된 사용자나 역할이 프로파일에서 작업을 실행하기 전에 암호를 입력하도록 요구하는 rights profile(권한 프로파일)입니다. 이 동작은 `sudo` 동작과 비슷합니다. 암호가 유효한 기간은 구성 가능합니다.

authentication(인증) 주체의 제시된 신원을 확인하는 프로세스입니다.

authenticator(인증자) 인증자는 티켓(KDC에서) 및 서비스(서버에서)를 요청할 때 클라이언트에 의해 전달됩니다. 최근 시점에서 확인할 수 있는 클라이언트 및 서버에만 알려진 세션 키를 사용하여 생성된 정보를 포함하므로 트랜잭션이 안전한 것으로 나타납니다. 티켓과 함께 사용할 경우 인증자는 사용자 주체를 인증할 수 있습니다. 인증자에는 사용자의 주체 이름, 사용자 호스트의 IP 주소, 시간 기록이 포함됩니다. 티켓과 달리, 인증자는 대개 서비스 액세스를 요청할 때 한번만 사용할 수 있습니다. 인증자는 클라이언트 및 서버에 대한 세션 키를 사용하여 암호화됩니다.

authorization(권한 부여) 1. Kerberos에서는 주체가 서비스를 사용할 수 있는지, 어떤 객체에 주체가 액세스할 수 있는지, 각 객체에 대해 허용된 액세스 유형 등을 결정하는 프로세스입니다.

2. 사용자 권한 관리에서는 다른 상황에서 보안 정책에 의해 금지되는 종류의 작업을 수행하기 위해 역할 또는 사용자에 지정하거나 권한 프로파일에 포함할 수 있는 권한입니다. 권한 부여는 커널이 아니라 사용자 응용 프로그램 레벨에서 적용됩니다.

basic set(기본 세트) 로그인 시 사용자 프로세스에 지정되는 권한 세트입니다. 수정되지 않은 시스템에서 각 사용자의 초기 상속 가능한 세트는 로그인 시 기본 세트와 같습니다.

Blowfish 32-448비트의 가변 길이 키를 사용하는 대칭 블록 암호화 알고리즘입니다. 저작자인 Bruce Schneier에 따르면, Blowfish는 키를 자주 바꾸지 않는 응용 프로그램에 최적화되어 있습니다.

confidentiality(가밀성) privacy(프라이버시)를 참조하십시오.

consumer(소비자) Oracle Solaris의 암호화 프레임워크 기능에서 소비자는 공급자로부터 전달된 암호화 서비스의 사용자입니다. 소비자는 응용 프로그램, 최종 사용자 또는 커널 작업일 수 있습니다. 소비자의 예로 Kerberos, IKE, IPsec 등이 있습니다. 공급자의 예는 provider(공급자)를 참조하십시오.

credential cache(자격 증명 캐시) KDC로부터 받은 자격 증명을 포함하는 저장 공간(대개 파일)입니다.

credential(자격 증명) 티켓 및 일치하는 세션 키를 포함하는 정보 패키지입니다. 주체의 신원을 인증하는 데 사용됩니다. 티켓, session key(세션 키)도 참조하십시오.

DES Data Encryption Standard의 머리글자어로, 데이터 암호화 표준입니다. 1975년에 개발되고 1981년에 ANSI에 의해 ANSI X.3.92로 표준화된 대칭 키 암호화 방법입니다. DES에서는 56비트 키를 사용합니다.

Diffie-Hellman 프로토콜 공개 키 암호화라고도 합니다. 1976년 Diffie와 Hellman이 개발한 비대칭 암호화 키 계약 프로토콜입니다. 이 프로토콜을 사용하면 두 사용자가 사전 보안 없이 비보안 매체를 통해 보안 키를 교환할 수 있습니다. Diffie-Hellman은 Kerberos에서 사용됩니다.

DSA Digital Signature Algorithm의 머리글자어로, 디지털 서명 알고리즘입니다. 512-4096비트의 가변 키 크기를 사용하는 공개 키 알고리즘입니다. 미국 정부 표준인 DSS는 1024비트까지 지원합니다. DSA는 입력에 SHA1을 사용합니다.

ECDSA Elliptic Curve Digital Signature Algorithm의 머리글자어로, 타원 곡선 디지털 서명 알고리즘입니다. 타원 곡선 수학을 기반으로 하는 공개 키 알고리즘입니다. ECDSA 키 크기는 동일한 길이의 서명을 생성하는 데 필요한 DSA 공개 키의 크기보다 많이 작습니다.

flavor(종류) 전통적으로 보안 종류와 인증 종류는 인증 유형(AUTH_UNIX, AUTH_DES, AUTH_KERB)을 지칭한 종류로서, 동일한 의미입니다. RPCSEC_GSS는 인증과 더불어 무결성과 프라이버시 서비스를 제공하지만 역시 보안 종류입니다.

forwardable ticket(전달 가능 티켓) 클라이언트가 원격 호스트에서 티켓을 요청하기 위해 전체 인증 프로세스를 거치지 않고도 사용할 수 있는 티켓입니다. 예를 들어, 사용자 `jennifer`의 시스템에 있는 동안 사용자 `david`가 전달 가능 티켓을 얻은 경우 `david`는 새 티켓을 얻지 않고도(다시 인증받을 필요 없이) 자신의 시스템에 로그인할 수 있습니다. proxiable ticket(프록시 가능 티켓)도 참조하십시오.

FQDN 정규화된 도메인 이름입니다. 간단한 `denver`와 대조되는 `central.example.com`을 예로 들 수 있습니다.

GSS-API Generic Security Service Application Programming Interface의 약자. Kerberos 서비스를 포함하여 다양한 모듈형 보안 서비스를 지원하는 네트워크 계층입니다. GSS-API는 보안 인증, 무결성, 프라이버시 서비스를 제공합니다. authentication(인증), 무결성, privacy(프라이버시)를 참조하십시오.

hardening(강화) 호스트에 내재된 보안 취약성을 제거하도록 운영 체제의 기본 구성을 수정한 것입니다.

hardware provider(하드웨어 공급자) Oracle Solaris의 암호화 프레임워크 기능에서 장치 드라이버 및 해당 하드웨어 가속기입니다. 하드웨어 공급자는 컴퓨터 시스템에서 값비싼 암호화 작업 부담을 덜어주므로 CPU 리소스를 확보하여 다른 용도로 사용할 수 있습니다. provider(공급자)도 참조하십시오.

host principal(호스트 주체) `ftp`, `rcp` 또는 `rlogin`과 같은 다양한 네트워크 서비스를 제공하기 위해 주체(기본 이름 `host`로 서명됨)가 설정되는 특정 인스턴스의 서비스 주체입니다. 호스트 주체의 예는 `host/central.example.com@EXAMPLE.COM`입니다. 서버 주체도 참조하십시오.

host(호스트) 네트워크를 통해 액세스 가능한 시스템입니다.

KDC Key Distribution Center의 머리글자어로, 키 배포 센터입니다. 세 가지 Kerberos V5 구성 요소가 있는 시스템입니다.

- 주체 및 키 데이터베이스
- 인증 서비스
- TGS(티켓 부여 서비스)

각 영역에는 마스터 KDC가 있고 하나 이상의 슬레이브 KDC가 있어야 합니다.

Kerberos 인증 서비스, 서비스에서 사용되는 프로토콜 또는 서비스 구현에 사용되는 코드입니다.

Kerberos V5 구현에 가장 근접한 Oracle Solaris의 Kerberos 구현입니다.

"Kerberos"와 "Kerberos V5"는 기술적으로 서로 다르지만 Kerberos 문서에서 종종 바꿔서 사용하기도 합니다.

Kerberos(Cerberus라고도 씀)는 그리스 신화에서 지옥의 문을 지키는 머리가 셋 달린 사나운 개입니다.

Kerberos policy(Kerberos 정책) Kerberos 서비스에서 암호 사용을 통제하는 규칙 세트입니다. 정책을 통해 주체의 액세스나 티켓 수명 매개변수를 규제할 수 있습니다.

key(키) 1. 일반적으로, 두 가지의 주요 키 유형 중 하나입니다.

- 대칭 키 - 암호 해독 키와 똑같은 암호화 키입니다. 대칭 키는 파일을 암호화하는 데 사용됩니다.
- 비대칭 키 또는 공개 키 - Diffie-Hellman 또는 RSA와 같은 공개 키 알고리즘에 사용되는 키입니다. 공개 키에는 한 사용자에만 알려진 개인 키, 서버나 일반 리소스에서 사용되는 공개 키, 그리고 둘을 조합한 개인-공개 키 쌍이 포함됩니다. 개인 키는 보안 키라고도 합니다. 공개 키는 공유 키 또는 공통 키라고도 합니다.

2. keytab 파일의 항목(주체 이름)입니다. keytab file(keytab 파일)도 참조하십시오.

3. Kerberos에서 암호화 키로 사용되며 다음 세 가지 유형이 있습니다.

- 개인 키 - 주체 및 KDC에서 공유되고 시스템 한도 밖에서 배포되는 암호화 키입니다. private key(개인 키)도 참조하십시오.
- 서비스 키 - 이 키는 개인 키와 동일한 목적을 제공하지만, 서버 및 서비스에서 사용됩니다. 서비스 키도 참조하십시오.
- 세션 키 - 단일 로그인 세션 기간으로 제한된 수명 동안 두 주체 간에 사용되는 임시 암호화 키입니다. session key(세션 키)도 참조하십시오.

keystore(키 저장소) 키 저장소는 응용 프로그램별로 검색하기 위한 암호, 문장암호, 인증서 및 기타 인증 객체를 저장합니다. 키 저장소는 일부 응용 프로그램이 사용하는 기술 또는 위치에 따라 달라질 수 있습니다.

keytab file(keytab 파일) 하나 이상의 키(주체)를 포함하는 키 테이블 파일입니다. 사용자가 암호를 사용하는 것처럼 호스트나 서비스는 keytab 파일을 사용합니다.

kvno 키 버전 번호. 생성 순서대로 특정 키를 추적하는 시퀀스 번호입니다. 가장 높은 kvno가 최신의 가장 현재 키입니다.

MAC 1. MAC(메시지 인증 코드)를 참조하십시오.

2. 레이블 지정이라고도 합니다. 정부 보안 기술에서 MAC은 필수 액세스 제어입니다. MAC의 예로 Top Secret 및 Confidential과 같은 레이블이 있습니다. MAC은 DAC(모든 액세스 제어)과 대조를 이룹니다. DAC의 예로 UNIX 사용 권한이 있습니다.

3. 하드웨어에서 LAN의 고유한 시스템 주소입니다. 시스템이 이더넷에 있는 경우 MAC은 이더넷 주소입니다.

MAC(메시지 인증 코드) MAC는 데이터 무결성을 보증하고 데이터 발신을 인증합니다. MAC는 도청에 대해 보호되지 않습니다.

MD5 디지털 서명을 포함하여 메시지 인증용으로 사용되는 반복적인 암호화 해시 함수입니다. 이 기능은 1991년 Rivest가 개발했습니다. 이 기술은 더 이상 사용되지 않습니다.

mechanism(방식) 1. 데이터 인증 또는 기밀성을 이루기 위한 암호화 기법을 지정하는 소프트웨어 패키지입니다. 예: Kerberos V5, Diffie-Hellman 공개 키.

2. Oracle Solaris의 암호화 프레임워크 기능에서 특정 목적을 위한 알고리즘의 구현입니다. 예를 들어, 인증에 적용된 DES 방식(예: CKM_DES_MAC)은 암호화에 적용된 DES 방식(예: CKM_DES_CBC_PAD)과 별도의 방식입니다.

message digest(메시지 다이제스트) 메시지 다이제스트는 메시지에서 계산된 해시 값입니다. 해시 값은 메시지를 거의 고유하게 식별합니다. 다이제스트는 파일 무결성 확인에 유용합니다.

NTP Network Time Protocol의 약자. 네트워크 환경에서 정밀한 시간이나 네트워크 클럭 동기화(또는 둘 다)를 관리할 수 있는 델라웨어 대학교에서 설계한 소프트웨어입니다. NTP를 사용하여 Kerberos 환경에서 클럭 불균형을 유지 관리할 수 있습니다. 클럭 불균형도 참조하십시오.

PAM Pluggable Authentication Module의 약자. 여러 인증 방식에서 서비스를 재컴파일할 필요 없이 사용할 수 있는 프레임워크입니다. PAM은 로그인 시 Kerberos 세션 초기화를 사용으로 설정합니다.

permitted set(허가된 세트) 프로세스에서 사용할 수 있는 권한 세트입니다.

policy(정책) 일반적으로, 의사결정 및 조치를 반영하거나 결정하는 계획이나 행동 방침입니다. 컴퓨터 시스템의 경우 정책은 대개 보안 정책을 의미합니다. 사이트의 보안 정책은 처리 중인 정보의 민감도를 정의하는 규칙 세트이자, 허용되지 않은 액세스로부터 정보를 보호하는 데 사용되는 측정치입니다. 예를 들어, 시스템을 감사하고 장치를 사용할 수 있도록 할당하고 암호를 6주마다 변경하도록 보안 정책을 수립할 수 있습니다.

Oracle Solaris OS의 특정 영역에서 정책을 구현하는 방법은 audit policy(감사 정책), 암호화 프레임워크의 정책, 장치 정책, Kerberos policy(Kerberos 정책), 암호 정책 및 rights policy(권한 정책)을 참조하십시오.

postdated ticket(후일자 티켓) 후일자 티켓은 생성 후 지정된 시간까지 유효해지지 않습니다. 예를 들어, 이러한 티켓은 밤늦게 실행하려는 일괄 처리 작업에 유용합니다. 티켓을 훔친 경우 일괄 처리 작업이 실행될 때까지 티켓을 사용할 수 없기 때문입니다. 후일자 티켓을 발행할 때 `invalid`로 발행되고 a) 시작 시간이 지날 때까지 b) 클라이언트가 KDC에서 검증으로 요청할 때까지 해당 방법을 유지합니다. 후일자 티켓은 보통 TGT(티켓 부여 티켓)의 만료 시간까지 유효합니다. 그러나 후일자 티켓이 `renewable`로 표시된 경우 티켓의 수명이 보통 TGT(티켓 부여 티켓)의 전체 수명 기간과 똑같이 설정됩니다. 잘못된 티켓, renewable ticket(갱신 가능 티켓)도 참조하십시오.

principal name(주체 이름) 1. *primary/instance@REALM* 형식의 주체 이름입니다. 인스턴스, 기본, realm(영역)도 참조하십시오.

2. (RPCSEC_GSS API) 클라이언트 주체, 서버 주체를 참조하십시오.

privacy(프라이버시) 전송된 데이터를 보내기 전에 암호화하는 보안 서비스입니다. 프라이버시에는 데이터 무결성과 사용자 인증도 포함됩니다. authentication(인증), 무결성, 서비스를 참조하십시오.

private key(개인 키) 각 사용자 주체에 제공되며 주체의 사용자와 KDC에만 알려진 키입니다. 사용자 주체의 경우 키는 사용자 암호를 기반으로 합니다. key(키)를 참조하십시오.

private-key encryption(개인 키 암호화) 개인 키 암호화에서 발신자와 수신자는 암호화에 동일한 키를 사용합니다. 공개 키 암호화도 참조하십시오.

privilege escalation(권한 에스컬레이션) 기본값을 대체하는 권한을 포함하여 지정된 권한이 허용하는 리소스 범위 밖에 있는 리소스에 대한 액세스를 얻는 것입니다. 그 결과, 프로세스에서 허용되지 않은 작업을 수행할 수 있습니다.

privilege model(권한 모델) 수퍼 유저 모델보다 더 엄격한 컴퓨터 시스템의 보안 모델입니다. 권한 모델에서 프로세스를 실행하려면 권한이 필요합니다. 시스템 운영은 관리자가 해당 프로세스에 보유한 권한으로 기반으로 별개의 부분으로 나눌 수 있습니다. 관리자의 로그인 프로세스에 권한을 지정할 수 있습니다. 또는 특정 명령에만 효력을 발휘하도록 권한을 지정할 수 있습니다.

privilege set(권한 세트) 권한 모음입니다. 각 프로세스에는 프로세스가 특정 권한을 사용할 수 있는지 여부를 결정하는 4개의 권한 세트가 있습니다. 제한 세트, 유효 세트, permitted set(허가된 세트), 상속 가능한 세트를 참조하십시오.

또한 권한의 basic set(기본 세트)는 로그인 시 사용자의 프로세스에 지정된 권한 모음입니다.

privilege-aware(권한 인식) 코드를 통해 권한 사용을 켜고 끄는 프로그램, 스크립트, 명령입니다. 운용 환경에서 켜져 있는 권한을 프로세스에 제공해야 합니다. 프로그램의 사용자가 권한을 프로그램에 추가한 권한 프로파일을 사용하도록 하면 됩니다. 권한에 대한 자세한 설명은 `privileges`(5) 매뉴얼 페이지를 참조하십시오.

privilege(권한) 1. 일반적으로 컴퓨터 시스템에서 일반 사용자의 권한을 벗어난 작업을 수행할 수 있는 권한 또는 기능입니다. 수퍼 유저 권한은 수퍼 유저에게 부여된 모든 rights(권한)입니다. 권한 있는 사용자 또는 권한 있는 응용 프로그램은 추가 권한이 부여된 사용자나 응용 프로그램입니다.

2. Oracle Solaris 시스템에서 프로세스에 대한 별개의 권한입니다. 권한은 `root`인 프로세스를 좀 더 세부적으로 제어합니다. 권한은 커널에서 정의되고 시행됩니다. 권한을 프로세스 권한 또는 커널 권한이라고도 합니다. 권한에 대한 자세한 설명은 `privileges`(5) 매뉴얼 페이지를 참조하십시오.

privileged application(권한 있는 응용 프로그램) 시스템 컨트롤을 대체할 수 있는 응용 프로그램입니다. 응용 프로그램이 특정 UID, GID, 권한 부여 또는 권한과 같은 보안 속성을 검사합니다.

privileged user(권한 있는 사용자) 컴퓨터 시스템에서 일반 사용자의 권한을 벗어난 권한이 지정된 사용자입니다. trusted users(신뢰할 수 있는 사용자)도 참조하십시오.

profile shell(프로파일 셸) 권한 관리에서는 역할 또는 사용자가 역할의 권한 프로파일에 지정된 권한 있는 응용 프로그램을 명령줄에서 실행할 수 있게 하는 셸입니다. 프로파일 셸 버전은 시스템에서 사용 가능한 셸에 해당합니다(예: `bash`의 `pfbash` 버전).

provider(공급자) Oracle Solaris의 암호화 프레임워크 기능에서 소비자에게 제공된 암호화 서비스입니다. 공급자의 예로 PKCS #11 라이브러리, 커널 암호화 모듈, 하드웨어 가속기가 있습니다. 공급자는 암호화 프레임워크에 플러그인되므로 플러그인이라고도 합니다. 소비자의 예는 consumer(소비자)를 참조하십시오.

proxiable ticket(프록시 가능 티켓) 클라이언트에 작업을 수행하는 대신, 서비스에서 사용할 수 있는 티켓입니다. 따라서 서비스가 클라이언트의 프록시 역할을 한다고 말할 수 있습니다. 티켓을 사용하여 서비스는 클라이언트의 신원을 차용할 수 있습니다. 프록시 가능 티켓을 사용하여 다른 서비스에 대한 서비스 티켓을 얻을 수 있지만, TGT(티켓 부여 티켓)는 얻을 수 없습니다. 프록시 가능 티켓과 전달 가능 티켓의 차이점은, 프록시 가능 티켓은 단일 작업에만 유효하다는 것입니다. forwardable ticket(전달 가능 티켓)도 참조하십시오.

public object(공용 객체) `root` 사용자가 소유하고 어디서든 읽을 수 있는 파일입니다(예: `/etc` 디렉토리의 파일).

QOP Quality of Protection의 머리글자어로, 보호 품질입니다. 무결성 서비스나 프라이버시 서비스와 함께 사용되는 암호화 알고리즘을 선택할 수 있는 매개변수입니다.

RBAC 역할 기반 액세스 제어의 머리글자어로, Oracle Solaris의 사용자 권한 관리 기능입니다. rights(권한)을 참조하십시오.

RBAC policy(RBAC 정책) rights policy(권한 정책)을 참조하십시오.

realm(영역) 1. 단일 Kerberos 데이터베이스와 일련의 KDC(키 배포 센터)에 의해 제공된 논리적 네트워크입니다.

2. 주체 이름의 세번째 부분입니다. 주체 이름 `jdoe/admin@CORP.EXAMPLE.COM`의 경우 영역은 `CORP.EXAMPLE.COM`입니다. principal name(주체 이름)도 참조하십시오.

reauthentication(재인증) 컴퓨터 작업을 수행하기 위해 암호를 제공하도록 요구하는 것입니다. 일반적으로 `sudo` 작업에는 재인증이 필요합니다. 인증된 권한 프로파일에는 재인증이 필요한 명령이 포함될 수 있습니다. authenticated rights profile(인증된 권한 프로파일)을 참조하십시오.

relation(관계) `kdc.conf` 또는 `krb5.conf` 파일에 정의된 구성 변수 또는 관계입니다.

renewable ticket(갱신 가능 티켓) 장시간 존재하는 티켓은 보안 위험이 있으므로 티켓을 `renewable`로 지정할 수 있습니다. 갱신 가능 티켓에는 두 개의 만료 시간 a) 티켓의 현재 인스턴스가 만료되는 시간 b) 티켓의 최대 수명이 있습니다. 클라이언트가 티켓을 계속 사용하려면 첫번째 만료가 발생하기 전에 티켓을 갱신합니다. 예를 들어, 1시간 동안 유효한 티켓이 있고 모든 티켓은 최대 10시간의 수명을 가질 수 있습니다. 티켓을 보유하는 클라이언트가 1시간보다 더 오래 티켓을 보관하려면 티켓을 갱신해야 합니다. 티켓이 최대 티켓 수명에 도달하면 자동으로 만료되어 갱신할 수 없습니다.

rights policy(권한 정책) 명령과 연관된 보안 정책입니다. 현재 Oracle Solaris에 유효한 정책은 `solaris`입니다. `solaris` 정책은 권한과 확장 권한 정책, 권한 부여 및 `setuid` 보안 속성을 인식합니다.

rights profile(권한 프로파일) 프로파일이라고도 합니다. 역할 또는 사용자에 지정할 수 있는 보안 대체 모음입니다. 권한(right) 프로파일에는 권한 부여, 권한(privilege), 보안 속성 포함 명령 및 보충 프로파일이라고 하는 기타 권한(right) 프로파일이 포함될 수 있습니다.

rights(권한) all-or-nothing 수퍼 유저 모델의 대안입니다. 사용자 권한(right) 관리 및 프로세스 권한(right) 관리를 통해 조직은 수퍼 유저의 권한(privilege)을 분담하고 이를 사용자 또는 역할에 지정할 수 있습니다. Oracle Solaris의 권한(right)은 커널 권한(privilege), 권한 부여 및 프로세스를 특정 UID 또는 GID로 실행하는 기능으로 구현됩니다. rights profile(권한 프로파일) 및 role(역할)에서 권한을 수집할 수 있습니다.

role(역할) 지정된 사용자만 맡을 수 있는 권한 있는 응용 프로그램을 실행하기 위한 특수한 신원입니다.

RSA 디지털 서명 및 공개 키 암호화 체계를 얻기 위한 방법입니다. 1978년에 개발자 Rivest, Shamir, Adleman이 처음 기술했습니다.

scan engine(검사 엔진) 파일에 알려진 바이러스가 있는지 조사하는, 외부 호스트에 상주하는 타사 응용 프로그램입니다.

SEAM Solaris 시스템의 Kerberos 초기 버전에 대한 제품 이름입니다. 이 제품은 MIT(Massachusetts Institute of Technology)에서 개발된 Kerberos V5 기술을 기반으로 합니다. 이제 SEAM을 Kerberos 서비스라고 합니다. SEAM은 계속해서 MIT 버전과 약간 다릅니다.

secret key(보안 키) private key(개인 키)를 참조하십시오.

Secure Shell(보안 셸) 비보안 네트워크를 통해 보안 원격 로그인 및 기타 보안 네트워크 서비스를 제공하기 위한 특수한 프로토콜입니다.

security attributes(보안 속성) 수퍼 유저가 아닌 사용자가 관리 명령을 실행할 때 이 명령이 성공하게 해주는 보안 정책의 대체입니다. 수퍼 유저 모델에서는 `setuid root` 및 `setgid` 프로그램이 보안 속성입니다. 이러한 속성을 명령에 적용할 때 누가 명령을 실행하든지 관계없이 명령을 성공합니다. privilege model(권한 모델)에서는 커널 권한 및 기타 rights(권한)이 `setuid root` 프로그램을 보안 속성으로 대체합니다. 권한 모델은 `setuid` 및 `setgid` 프로그램을 보안 속성으로 인식하므로 수퍼 유저 모델과 호환됩니다.

security flavor(보안 종류) flavor(종류)를 참조하십시오.

security mechanism(보안 방식) mechanism(방식)을 참조하십시오.

seed(시드) 무작위 수를 생성하기 위한 숫자 스타터입니다. 스타터가 무작위 소스에서 기원할 때 시드를 무작위 시드라고 합니다.

separation of duty(책임 구분) 최소한의 특권 개념의 일부입니다. 책임을 구분하면 한 사용자가 트랜잭션을 완성하는 모든 작업을 수행하거나 승인할 수 없게 됩니다. 예를 들어, RBAC에서 보안 대체 지정으로부터 로그인 사용자 생성을 분리할 수 있습니다. 한 역할이 사용자를 만듭니다. 별도의 역할이 권한 프로파일, 역할, 기존 사용자의 권한과 같은 보안 속성을 지정할 수 있습니다.

server(서버) 네트워크 클라이언트에 리소스를 제공하는 주체입니다. 예를 들어, `central.example.com` 시스템에 `ssh`를 제공하면 해당 시스템은 `ssh` 서비스를 제공하는 서버입니다. 서비스 주체도 참조하십시오.

session key(세션 키) 인증 서비스 또는 TGS(티켓 부여 서비스)에서 생성된 키입니다. 세션 키는 클라이언트와 서비스 간에 보안 트랜잭션을 제공하기 위해 생성됩니다. 세션 키의 수명은 단일 로그인 세션으로 제한됩니다. key(키)를 참조하십시오.

SHA1 Secure Hashing Algorithm의 머리글자어로, 보안 해시 알고리즘입니다. 이 알고리즘은 2^{64} 미만의 입력 길이에서 작동하여 메시지 다이제스트를 생성합니다. SHA1 알고리즘은 DSA에 입력됩니다.

single-system image(단일 시스템 이미지) 단일 시스템 이미지는 Oracle Solaris 감사에 사용되어 동일한 이름 지정 서비스를 사용하는 감사 시스템 그룹을 설명합니다. 이러한 시스템은 해당 감사 레코드를 중앙 감사 서버로 보내고, 여기서 레코드가 한 시스템에서 나온 것처럼 레코드를 비교할 수 있습니다.

slave KDC(슬레이브 KDC) 마스터 KDC의 복사본으로, 대부분의 마스터 기능을 수행할 수 있습니다. 각 영역에는 대개 여러 개의 슬레이브 KDC(마스터 KDC는 하나만)가 있습니다. KDC, 마스터 KDC도 참조하십시오.

software provider(소프트웨어 공급자) Oracle Solaris의 암호화 프레임워크 기능에서 암호화 서비스를 제공하는 커널 소프트웨어 모듈 또는 PKCS #11 라이브러리입니다. provider(공급자)도 참조하십시오.

stash 파일 stash 파일은 KDC용 마스터 키의 암호화된 복사본을 포함합니다. `kadmind` 및 `krb5kdc` 프로세스를 시작하기 전에 KDC를 자동으로 인증하도록 서버를 재부트할 때 이 마스터 키가 사용됩니다. stash 파일에 마스터 키가 포함되므로 stash 파일과 그 백업을 안전하게 보관해야 합니다. 암호화가 훼손되면 키를 사용하여 KDC 데이터베이스를 액세스하거나 수정해야 합니다.

TGS Ticket-Granting Service의 머리글자어로, 티켓 부여 서비스입니다. 티켓 발행을 담당하는 KDC의 부분입니다.

TGT Ticket-Granting Ticket의 머리글자어로, 티켓 부여 티켓입니다. 클라이언트가 다른 서비스에 대한 티켓을 요청할 수 있는 KDC에서 발행한 티켓입니다.

trusted users(신뢰할 수 있는 사용자) 결정된 사용자는 일부 신뢰 레벨에서 관리 작업을 수행할 수 있습니다. 일반적으로 관리자는 먼저 신뢰할 수 있는 사용자에 대한 로그인을 만들고 사용자의 신뢰 레벨 및 기능과 일치하는 관리 권한을 지정합니다. 이러한 사용자는 시스템을 구성 및 유지 관리하는 데 도움이 됩니다. 권한 있는 사용자라고도 합니다.

user principal(사용자 주체) 특정 사용자에 기인한 주체입니다. 사용자 주체의 기본 이름은 사용자 이름이고, 선택적 인스턴스는 의도한 해당 자격 증명 용도를 설명하는 데 사용되는 이름입니다(예: `jdoe` 또는 `jdoe/admin`). 사용자 인스턴스라고도 합니다. 서비스 주체도 참조하십시오.

VPN(가상 사설망) 암호화를 사용하고 공용 네트워크를 통한 사용자 연결을 터널링하여 보안 통신을 제공하는 네트워크입니다.

색인