Oracle® Solaris 11.2의 감사 관리

부품 번호: E53975
2014년 7월

ORACLE

Copyright © 2002, 2014, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이센스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이센스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이센스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이센스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이센스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이센스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제 3자로부터 제공되는 컨텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 컨텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 컨텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

목차

이 설명서 사용

*Oracle® Solaris 11.2*의 감사 관리에서는 Oracle Solaris의 감사 기능에 대해 설명합니다.

- **개요** - Oracle Solaris 시스템 또는 시스템 네트워크에서 감사를 관리하는 방법을 설명합니다.
- **대상** - 회사에서 보안을 구현해야 하는 시스템 관리자
- **필요한 지식** - 보안 개념 및 용어 관련 전문 지식

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 Oracle Solaris의 감사 정보

Oracle Solaris의 감사 부속 시스템은 시스템이 어떻게 사용되고 있는지에 대한 기록을 유지합니다. 감사 서비스에는 감사 데이터 분석을 도와주는 도구가 포함됩니다.

이 장에서는 Oracle Solaris에서 감사가 어떻게 작동하는지 소개합니다

계획 제안은 2장. 감사 계획을 참조하십시오. 사용자 사이트에서 감사를 구성하는 절차는 다음 장을 참조하십시오.

참조 정보는 7장. 감사 참조를 참조하십시오.

Oracle Solaris 감사 서비스의 새로운 기능

이 절에서는 Oracle Solaris 감사 서비스에서 기존 고객을 위한 중요한 새 기능에 대한 정보를 보여줍니다.

- Oracle Solaris 시스템의 감사 레코드를 Oracle Audit Vault and Database Firewall에 플러그인하면 Oracle Solaris 시스템에서 감사되는 이벤트에 대한 정보를 가져오는 데 사용할 수 있습니다.
- 감사 구성 파일 `audit_class`, `audit_event` 및 `audit_warn`에는 두 개의 패키지 속성 세트가 포함됩니다. `preserve=renamenew` 속성을 사용하면 파일을 수정할 수 있으며, 패키

지를 업데이트하고 수정하더라도 수정 사항이 보존됩니다. `overlay=allow` 속성을 사용하면 파일을 고객이 만드는 패키지의 파일로 바꿀 수 있습니다.

감사란?

감사는 시스템 리소스 사용에 대한 데이터의 모음입니다. 감사 데이터는 보안 관련 시스템 이벤트의 레코드를 제공합니다. 그러면 이 데이터를 사용하여 호스트에서 발생하는 작업에 대한 책임을 지정할 수 있습니다.

성공적인 감사는 식별 및 인증의 두 가지 보안 기능으로 시작됩니다. 각 로그인 시 사용자가 사용자 이름을 제공하고 PAM(플러그인할 수 있는 인증 모듈) 인증을 성공하면 고유하고 변경 불가능한 감사 사용자 *ID*가 생성되어 해당 사용자와 연결되며, 고유한 감사 세션 ID가 생성되고 해당 사용자의 프로세스와 연결됩니다. 감사 세션 ID는 해당 로그인 세션 중 시작된 모든 프로세스에서 상속합니다. 사용자가 다른 사용자로 전환할 경우 모든 사용자 작업은 동일한 감사 사용자 ID로 추적됩니다. ID 전환에 대한 자세한 내용은 `su`(1M) 매뉴얼 페이지를 참조하십시오. 기본적으로 시스템 부트 및 종료와 같은 특정 작업은 항상 감사됩니다.

감사 서비스를 통해 다음과 같은 작업을 수행할 수 있습니다.

- 호스트에서 발생하는 보안 관련 이벤트 모니터링
- 네트워크 전역 감사 추적으로 이벤트 기록
- 잘못된 사용 또는 무단 작업 감지
- 개인 및 객체의 액세스 패턴 및 액세스 내역 검토
- 보호 방식을 우회하려는 시도 감지
- 사용자가 ID를 변경할 때 발생하는 확장된 권한 사용 감지

참고 - 보안 유지 관리를 위해 변경된 암호와 같은 감사된 이벤트 중 일부는 표시되지 않습니다. 자세한 내용은 "감사 레코드 및 감사 토큰" [14]을 참조하십시오.

감사 용어 및 개념

다음 용어는 감사 서비스를 설명하는 데 사용됩니다. 일부 정의에는 좀더 자세한 설명에 대한 포인터가 포함되어 있습니다.

감사 클래스
: 감사 이벤트의 그룹화입니다. 감사 클래스는 감사할 이벤트 그룹을 선택할 수 있는 방법을 제공합니다.

  자세한 내용은 "감사 클래스 및 사전 선택" [13]과 `audit_flags`(5), `audit_class`(4) 및 `audit_event`(4) 매뉴얼 페이지를 참조하십시오.

감사 파일 시스템
: 이진 형식의 감사 파일 저장소입니다.

자세한 내용은 “감사 로그” [15] 및 audit.log(4) 매뉴얼 페이지를 참조하십시오.

감사 이벤트

감사 가능한 보안 관련 시스템 작업입니다. 선택이 용이하도록 이벤트는 감사 클래스로 그룹화됩니다.

자세한 내용은 “감사 이벤트” [12] 및 audit_event(4) 매뉴얼 페이지를 참조하십시오.

감사 플래그

명령 또는 키워드에 대한 인수로 제공되는 감사 클래스입니다. 플래그는 더하기 기호나 빼기 기호를 앞에 붙여 클래스가 성공(+) 또는 실패(-)에 대해 감사되는지 나타낼 수 있습니다. 앞에 캐럿(^)이 있으면 성공이 감사되지 않음(^+) 또는 실패가 감사되지 않음(^-)을 나타냅니다.

자세한 내용은 audit_flags(5) 매뉴얼 페이지 및 “감사 클래스 구문” [111]을 참조하십시오.

감사 플러그인

대기열의 감사 레코드를 지정된 위치로 전송하는 모듈입니다. audit_binfile 플러그인은 이진 감사 파일을 만듭니다. 이진 파일은 감사 파일 시스템에 저장되는 감사 추적을 구성합니다. audit_remote 플러그인은 이진 감사 레코드를 원격 저장소로 보냅니다. audit_syslog 플러그인은 syslog 로그에서 선택한 감사 레코드를 요약합니다.

자세한 내용은 “감사 플러그인 모듈” [15]과 모듈 매뉴얼 페이지 audit_binfile(5), audit_remote(5) 및 audit_syslog(5)를 참조하십시오.

감사 정책

사이트에서 사용 또는 사용 안함으로 설정할 수 있는 감사 옵션 세트입니다. 특정 종류의 감사 데이터를 기록할지 여부 및 감사 대기열이 가득 찼을 때 감사 가능한 작업을 일시 중지할지 여부를 지정할 수 있습니다.

자세한 내용은 “감사 정책 이해” [32] 및 auditconfig(1M) 매뉴얼 페이지를 참조하십시오.

감사 레코드

감사 대기열에 수집되는 감사 데이터입니다. 하나의 감사 레코드는 단일 감사 이벤트를 설명합니다. 각 감사 레코드는 감사 토큰으로 구성됩니다.

자세한 내용은 “감사 레코드 및 감사 토큰” [14] 및 audit.log(4) 매뉴얼 페이지를 참조하십시오.

감사 토큰

감사 레코드 또는 이벤트의 필드입니다. 각 감사 토큰은 사용자, 그룹, 프로그램 또는 기타 객체와 같은 감사 이벤트의 속성을 설명합니다.

자세한 내용은 “감사 토큰 형식” [117] 및 audit.log(4) 매뉴얼 페이지를 참조하십시오.

감사 추적

기본 플러그인 audit_binfile을 사용하는 모든 감사된 시스템의 감사 데이터를 저장하는 하나 이상의 감사 파일 모음입니다.

자세한 내용은 “감사 추적” [115]을 참조하십시오.

로컬 감사
: 로컬 시스템에서 생성되는 감사 레코드를 수집하는 작업입니다. 레코드는 전역 영역 또는 비전역 영역이나 두 영역 모두에서 생성할 수 있습니다.

  자세한 내용은 “감사 플러그인 모듈” [15]을 참조하십시오.

사후 선택
: 감사 추적에서 검사할 감사 이벤트를 선택합니다. 기본 활성 플러그인 `audit_binfile`이 감사 추적을 만듭니다. 사후 선택 도구 `auditreduce` 명령이 감사 추적에서 레코드를 선택합니다.

  자세한 내용은 `auditreduce`(1M) 및 `praudit`(1M) 매뉴얼 페이지를 참조하십시오.

사전 선택
: 모니터할 감사 클래스를 선택합니다. 사전 선택된 감사 클래스의 감사 이벤트는 감사 대기열에 수집됩니다. 사전 선택되지 않은 감사 클래스는 감사되지 않으므로 해당 이벤트가 대기열에 나타나지 않습니다.

  자세한 내용은 “감사 클래스 및 사전 선택” [13]과 `audit_flags`(5) 및 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오.

공용 객체
: `root` 사용자가 소유하고 누구나 읽을 수 있는 파일입니다. 예를 들어, `/etc` 디렉토리 및 `/usr/bin` 디렉토리에 있는 파일은 공용 객체입니다. 공용 객체는 읽기 전용 이벤트에 대해 감사되지 않습니다. 예를 들어, `file_read`(fr) 감사 클래스가 사전 선택되더라도 공용 객체 읽기는 감사되지 않습니다. `public` 감사 정책 옵션을 변경하여 기본값을 대체할 수 있습니다.

원격 감사
: 감사 중이고 활성 `audit_remote` 플러그인으로 구성된 시스템에서 감사 레코드를 수신하고 저장하는 ARS(감사 원격 서버)입니다. 감사되는 시스템을 ARS와 구분하기 위해 감사 시스템을 "로컬로 감사되는 시스템"이라고 부를 수 있습니다

  자세한 내용은 `auditconfig`(1M) 매뉴얼 페이지의 `-setremote` 옵션 및 “감사 원격 서버” [112]를 참조하십시오.

감사 이벤트

감사 이벤트는 시스템에서 감사 가능한 작업을 나타냅니다. 감사 이벤트는 `/etc/security/audit_event` 파일에 나열됩니다. 각 감사 이벤트는 시스템 호출 또는 사용자 명령에 연결되고, 하나 이상의 감사 클래스에 지정됩니다. `audit_event` 파일의 형식에 대한 설명은 `audit_event`(4) 매뉴얼 페이지를 참조하십시오.

예를 들어, `AUE_EXECVE` 감사 이벤트는 `execve()` 시스템 호출을 감사합니다. `auditrecord -e execve` 명령은 이 항목을 표시합니다.

```
# auditrecord -e execve
execve
system call execve              See execve(2)
event ID    23                  AUE_EXECVE
class       ps,ex               (0x0000000040100000)
header
path
[attribute]                 omitted on error
[exec_arguments]            output if argv policy is set
[exec_environment]          output if arge policy is set
subject
[use_of_privilege]
return
```

감사 클래스 ps 또는 감사 클래스 ex를 사전 선택할 경우 모든 execve() 시스템 호출이 감사 대기열에 기록됩니다.

감사는 *attributable* 및 *non-attributable* 이벤트를 처리합니다. 감사 정책은 다음과 같이 이벤트를 *synchronous* 및 *asynchronous* 이벤트로 나눕니다.

- **지정 가능한 이벤트** - 사용자에게 지정할 수 있는 이벤트입니다. execve() 시스템 호출은 사용자에게 지정할 수 있으므로 호출이 지정 가능한 이벤트로 간주됩니다. 모든 지정 가능한 이벤트는 동기 이벤트입니다.
- **지정 불가능한 이벤트** - 커널 인터럽트 레벨에서 발생하거나 사용자가 인증되기 전에 발생하는 이벤트입니다. na 감사 클래스는 지정 불가능한 감사 이벤트를 처리합니다. 예를 들어, 시스템 부트는 지정 불가능한 이벤트입니다. 대부분의 지정 불가능한 이벤트는 비동기 이벤트입니다. 하지만 실패한 로그인과 같이 연결된 프로세스가 있는 지정 불가능한 이벤트는 동기 이벤트입니다.
- **동기 이벤트** - 시스템의 프로세스와 연결된 이벤트입니다. 시스템 이벤트의 대부분은 동기 이벤트입니다.
- **비동기 이벤트** - 프로세스와 연결되지 않아 차단했다가 나중에 시작할 수 있는 프로세스가 없는 이벤트입니다. 초기 시스템 부트 및 PROM 진입/종료 이벤트가 비동기 이벤트의 예입니다.

감사 서비스에서 정의한 감사 이벤트 이외에 타사 응용 프로그램에서 감사 이벤트를 생성할 수 있습니다. 감사 이벤트 번호 32768부터 65535까지 타사 응용 프로그램에 대해 사용할 수 있습니다. 공급업체는 Oracle Solaris 담당자에게 연락하여 이벤트 번호를 예약하고 감사 인터페이스에 대한 액세스 권한을 얻어야 합니다.

감사 클래스 및 사전 선택

각 감사 이벤트는 감사 클래스에 속합니다. 감사 클래스는 많은 수의 감사 이벤트에 대한 편리한 컨테이너입니다. 감사할 클래스를 사전 선택할 경우 해당 클래스의 모든 이벤트가 감사 대기열에 기록됩니다. 예를 들어, ps 감사 클래스를 사전 선택하면 execve(), fork() 및 기타 시스템 호출이 기록됩니다.

시스템의 이벤트 또는 특정 사용자가 시작한 이벤트에 대해 사전 선택할 수 있습니다.

- **시스템 전역 사저 선택** - `auditconfig` 명령에 `-setflags` 및 `-setnaflags` 옵션을 사용하여 감사에 대한 시스템 전역 기본값을 지정합니다.

참고 - `perzone` 정책이 설정된 경우 모든 영역에서 기본 감사 클래스를 지정할 수 있습니다. `perzone` 감사의 경우 기본값은 시스템 전역이 아닌 영역 전역입니다.

- **사용자 특정 사전 선택** - 사용자에 대한 감사 플래그를 구성하여 개별 사용자에 대해 시스템 전역 감사 기본값과 다르게 지정합니다. `useradd`, `roleadd`, `usermod` 및 `rolemod` 명령은 `user_attr` 데이터베이스에 `audit_flags` 보안 속성을 추가합니다. `profiles` 명령은 `prof_attr` 데이터베이스에 권한 프로파일에 대한 감사 플래그를 추가합니다.

  감사 사전 선택 마스크는 사용자에 대해 감사되는 이벤트의 클래스를 결정합니다. 사용자 사전 선택 마스크에 대한 설명은 “프로세스 감사 특성” [114]을 참조하십시오. 사용되는 구성된 감사 플래그는 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 권한 검색 순서”를 참조하십시오.

감사 클래스는 `/etc/security/audit_class` 파일에 정의되어 있습니다. 각 항목은 클래스에 대한 감사 마스크, 클래스에 대한 이름 및 클래스에 대한 설명을 포함합니다. 예를 들어, `lo` 및 `ps` 클래스 정의는 `audit_class` 파일에 다음과 같이 나타납니다.

```
0x0000000000001000:lo:login or logout
0x0000000000100000:ps:process start/stop
```

감사 클래스에는 `all` 및 `no`의 두 전역 클래스가 포함됩니다. 감사 클래스에 대한 설명은 `audit_class`(4) 매뉴얼 페이지를 참조하십시오. 클래스 목록은 `/etc/security/audit_class` 파일을 검토하십시오.

클래스에 대한 감사 이벤트 매핑은 구성 가능합니다. 클래스에서 이벤트를 제거하거나 클래스에 이벤트를 추가하고, 선택한 특정 이벤트에 대해 새 클래스를 만들 수 있습니다. 절차는 감사 이벤트의 클래스 멤버쉽을 변경하는 방법 [52]을 참조하십시오. 클래스에 매핑된 이벤트를 보려면 `auditrecord -c` *class* 명령을 사용합니다.

감사 레코드 및 감사 토큰

각 감사 레코드는 감사된 단일 이벤트의 발생을 기록합니다. 레코드에는 작업을 수행한 사람, 영향을 받는 파일, 시도된 작업, 작업이 발생한 위치 및 시기 등과 같은 정보가 포함됩니다. 다음 예에서는 세 가지 토큰인 `header`, `subject` 및 `return`이 포함된 `login` 감사 레코드를 보여줍니다.

```
header,69,2,login - local,,example_system,2010-10-10 10:10:10.020 -07:00
subject,jdoe,jdoe,staff,jdoe,staff,1210,4076076536,69 2 example_system
return,success,0
```

각 감사 이벤트에 대해 저장되는 정보의 유형은 감사 토큰 세트로 정의됩니다. 이벤트에 대해 감사 레코드가 만들어질 때마다 레코드에는 해당 이벤트에 대해 정의된 토큰의 일부 또는 모두가 포함됩니다. 이벤트의 특성에 따라 기록되는 토큰이 결정됩니다. 위의 예에서 각 행은

감사 토큰의 이름으로 시작합니다. 감사 토큰의 내용은 토큰 이름 다음에 나옵니다. `header`, `subject` 및 `return` 감사 토큰은 함께 `login - local` 감사 레코드를 구성합니다. 감사 레코드를 구성하는 토큰을 표시하려면 `auditrecord -e` *event* 명령을 사용합니다.

참고 - `sensitive` 시스템 속성을 가진 파일은 해당 컨텐츠 또는 컨텐츠 변경 사항이 감사 레코드에 포함되지 않습니다. 이 속성은 특정 파일에 있는 민감한 정보(예: 암호, PIN, 키 등)에 아무나 액세스하지 못하도록 보장합니다. 자세한 내용은 `pfedit`(1M) 매뉴얼 페이지를 참조하십시오.

`praudit` 출력의 예와 함께 각 감사 토큰의 구조에 대한 자세한 설명은 "감사 토큰 형식" [117]을 참조하십시오. 감사 토큰의 이진 스트림에 대한 설명은 `audit.log`(4) 매뉴얼 페이지를 참조하십시오.

감사 플러그인 모듈

감사 플러그인 모듈은 감사 대기열의 감사 레코드를 파일 또는 저장소로 지정합니다. 적어도 하나의 플러그인은 활성화되어야 합니다. 기본적으로 `audit_binfile` 플러그인이 활성화됩니다. `auditconfig -setplugin` *plugin-name* 명령을 사용하여 플러그인을 구성합니다.

감사 서비스는 다음 플러그인을 제공합니다.

- `audit_binfile` 플러그인 - 이진 감사 파일로 감사 대기열의 전달을 처리합니다. 자세한 내용은 `audit.log`(4) 매뉴얼 페이지를 참조하십시오.
- `audit_remote` 플러그인 - 감사 대기열에서 구성된 원격 서버로 이진 감사 레코드의 보안 전달을 처리합니다. `audit_remote` 플러그인은 `libgss()` 라이브러리를 사용하여 서버를 인증합니다. 전송은 개인 정보 및 무결성을 위해 보호됩니다.
- `audit_syslog` 플러그인 - 감사 대기열에서 `syslog` 로그로 선택된 레코드의 전달을 처리합니다.

플러그인 구성 방법에 대한 자세한 내용은 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오. 플러그인 구성의 예는 "감사 로그 구성" [69]의 작업을 참조하십시오. 플러그인에 대한 자세한 내용은 `audit_binfile`(5), `audit_remote`(5) 및 `audit_syslog`(5) 매뉴얼 페이지를 참조하십시오.

감사 로그

감사 레코드는 감사 로그에 수집됩니다. 감사 서비스는 감사 레코드에 대한 세 가지 출력 모드를 제공합니다.

- 감사 파일이라는 로그는 감사 레코드를 이진 형식으로 저장합니다. 시스템 또는 사이트의 감사 파일 세트는 완전한 감사 레코드를 제공합니다. 완전한 감사 레코드를 감사 추

적이라고 합니다. 이러한 로그는 audit_binfile 플러그인으로 만들어지고, praudit 및 auditreduce 사후 선택 명령으로 검토할 수 있습니다.

- audit_remote 플러그인은 감사 레코드를 원격 저장소로 스트리밍합니다. 저장소에서는 감사 추적을 유지 관리하고 사후 선택 도구를 제공합니다.
- syslog 유틸리티는 감사 레코드의 텍스트 요약을 수집하고 저장합니다. syslog 레코드는 안전하지 않습니다. 다음 예는 login 감사 레코드에 대한 syslog 항목을 보여줍니다.

```
Oct 10  10:10:20 example_system auditd: [ID 6472 audit.notice] \
login - login ok session 4076172534 by root as root:other
```

사이트는 모든 형식으로 감사 레코드를 수집하도록 감사를 구성할 수 있습니다. 이진 모드를 로컬에서 사용하거나 이진 파일을 원격 저장소로 보내거나 syslog 모드를 사용하도록 사이트의 시스템을 구성할 수 있습니다. 다음 표는 이진 감사 레코드를 syslog 감사 레코드와 비교한 것입니다.

표 1-1 이진, 원격 및 syslog 감사 레코드의 비교

기능	이진 및 원격 레코드	syslog 레코드
프로토콜	이진 - 파일 시스템에 기록합니다. 원격 - 원격 저장소로 스트리밍합니다.	원격 로깅을 위해 UDP를 사용합니다.
데이터 유형	이진	텍스트
레코드 길이	제한 없음	감사 레코드당 최대 1024자
위치	이진 - 시스템의 zpool에 저장 원격 - 원격 저장소	syslog.conf 파일에 지정된 위치에 저장
구성하는 방법	이진 - audit_binfile 플러그인에서 p_dir 속성을 설정합니다. 원격 - audit_remote 플러그인에서 p_hosts 속성을 설정하고 플러그인을 활성화합니다.	audit_syslog 플러그인을 활성화하고 syslog.conf 파일을 구성합니다.
읽는 방법	이진 - 일반적으로 배치 모드에서 XML로 브라우저 출력 원격 - 저장소에서 절차 결정	실시간으로 또는 syslog에 대해 만든 스크립트로 검색 일반 텍스트 출력
완전성	완전성이 보장되며 올바른 순서로 나타남	완전성이 보장되지 않음
시간 기록	협정 세계시(UTC)	감사되는 시스템의 시간

플러그인 및 감사 로그에 대한 자세한 내용은 다음을 참조하십시오.

- audit_binfile(5) 매뉴얼 페이지
- audit_syslog(5) 매뉴얼 페이지
- audit.log(4) 매뉴얼 페이지
- 감사 추적에 대한 감사 공간을 지정하는 방법 [73]
- syslog 감사 로그를 구성하는 방법 [82]

이진 레코드 정보

이진 레코드가 가장 뛰어난 보안과 완전성을 제공합니다. 이진 출력은 Common Criteria (http://www.commoncriteriaportal.org/) 감사 요구 사항과 같은 보안 자격 증명 요구 사항을 충족합니다.

`audit_binfile` 플러그인은 스누핑으로부터 보호되는 파일 시스템에 레코드를 기록합니다. 단일 시스템에서 모든 이진 레코드는 순서대로 수집되고 표시됩니다. 한 감사 추적의 시스템이 여러 시간대에 분포되어 있는 경우 이진 로그의 UTC 시간 기록을 사용하여 정확한 비교가 가능합니다. `praudit -x` 명령을 사용하여 브라우저에서 XML로 레코드를 볼 수 있습니다. 또한 스크립트를 사용하여 XML 출력을 구문 분석할 수 있습니다.

`audit_remote` 플러그인은 원격 저장소에 레코드를 기록합니다. 저장소는 저장 및 사후 선택을 처리합니다.

`syslog` 감사 레코드 정보

반면, `syslog` 레코드는 높은 편의성과 유연성을 제공할 수 있습니다. 예를 들어, 다양한 소스에서 `syslog` 데이터를 수집할 수 있습니다. 또한 `syslog.conf` 파일에서 `audit.notice` 이벤트를 모니터할 때 `syslog` 유틸리티는 현재 시간 기록과 함께 감사 레코드 요약을 기록합니다. 워크스테이션, 서버, 방화벽 및 라우터를 포함한 다양한 소스에서 `syslog` 메시지에 대해 개발한 동일한 관리 및 분석 도구를 사용할 수 있습니다. 레코드는 실시간으로 보거나 원격 시스템에 저장할 수 있습니다.

`syslog.conf`를 사용하여 감사 레코드를 원격으로 저장하면 공격자가 로그 데이터를 변경하거나 삭제하지 못하도록 보호할 수 있습니다. 하지만 `syslog` 모드에 대한 다음과 같은 결점을 고려해야 합니다.

- 서비스 거부 및 스푸핑된 소스 주소와 같은 네트워크 공격에 레코드가 취약해질 수 있습니다.
- UDP 프로토콜은 패킷을 삭제하거나 패킷을 순서 없이 전달할 수 있습니다.
- `syslog` 항목에 대한 1024자 제한으로 인해 로그에서 일부 감사 레코드가 잘릴 수 있습니다.
- 단일 시스템에서는 일부 감사 레코드가 수집되지 않고 순서대로 표시되지 않을 수 있습니다.
- 각 감사 레코드는 로컬 시스템의 날짜 및 시간으로 기록됩니다. 따라서 여러 시스템에 대해 감사 추적을 구성할 때 시간 기록에 의존할 수 없습니다.

감사 추적 저장 및 관리

`audit_binfile` 플러그인이 활성화되면 감사 파일 시스템이 감사 파일을 이진 형식으로 보관합니다. 일반적인 설치에서는 `/var/audit` 파일 시스템을 사용하며 추가 파일 시스템을 사

용할 수 있습니다. 모든 감사 파일 시스템의 컨텐츠는 감사 추적을 구성합니다. 감사 레코드는 이러한 파일 시스템에 다음 순서대로 저장됩니다.

- **기본 감사 파일 시스템** - `/var/audit` 파일 시스템이며, 시스템에 대한 감사 파일의 기본 파일 시스템입니다.
- **보조 감사 파일 시스템** - 관리자의 지시에 따라 시스템에 대한 감사 파일이 보관되는 파일 시스템입니다.

파일 시스템은 `audit_binfile` 플러그인의 `p_dir` 속성에 인수로 지정됩니다. 목록의 앞에 있는 파일 시스템이 가득 찰 때까지 파일 시스템은 사용되지 않습니다. 파일 시스템 항목 목록의 예는 감사 파일에 대한 ZFS 파일 시스템을 만드는 방법 [70]을 참조하십시오.

기본 감사 루트 디렉토리에 감사 파일을 두면 감사 추적을 검토할 때 감사 검토자에게 도움이 됩니다. `auditreduce` 명령은 감사 루트 디렉토리를 사용하여 감사 추적의 모든 파일을 찾습니다. 기본 감사 루트 디렉토리는 `/var/audit`입니다.

`auditreduce` 명령에는 다음과 같은 옵션을 사용할 수 있습니다.

- `auditreduce` 명령에 대해 `-M` 옵션을 사용하면 특정 시스템에서 감사 파일을 지정할 수 있습니다.
- `-S` 옵션을 사용하면 다른 감사 파일 시스템을 지정할 수 있습니다.

`auditreduce` 명령 사용에 대한 예는 감사 추적에서 감사 파일을 병합하는 방법 [93]을 참조하십시오. 자세한 내용은 `auditreduce`(1M) 매뉴얼 페이지를 참조하십시오.

감사 서비스는 감사 추적의 파일을 결합하고 필터링하기 위한 명령을 제공합니다. `auditreduce` 명령은 감사 추적의 감사 파일을 병합할 수 있습니다. 또한 이 명령은 파일을 필터링하여 특정 이벤트를 찾을 수 있습니다. `praudit` 명령은 이진 파일을 읽습니다. `praudit` 명령에 대한 옵션은 스크립팅 및 브라우저 표시에 적당한 출력을 제공합니다.

신뢰할 수 있는 시간 기록 유지

여러 시스템의 감사 로그를 병합할 때 이러한 시스템의 날짜와 시간은 정확해야 합니다. 마찬가지로, 감사 로그를 원격 시스템에 보낼 때 기록 시스템과 저장소 시스템의 시계가 정확해야 합니다. NTP(Network Time Protocol)는 시스템 시계를 정확하고 알맞게 유지합니다. 자세한 내용은 “Oracle Solaris 11.2 네트워크 서비스 소개 ”의 3 장, “시간 관련 서비스” 및 `xntpd`(1M) 매뉴얼 페이지를 참조하십시오.

원격 저장소 관리

`audit_remote` 플러그인이 구성된 후 원격 저장소가 감사 레코드를 수신합니다. ARS는 감사 레코드에 대한 수신자를 제공합니다. 감사 레코드는 보호 연결을 통해 ARS에 스트리밍되며 로컬로 저장되는 방식과 비슷한 방식으로 저장할 수 있습니다. ARS를 구성하려면 감사 파일에 대한 원격 저장소를 구성하는 방법 [78]을 참조하십시오. ARS에 대한 설명은 “감사 원격 서버” [112] 및 `ars`(5) 매뉴얼 페이지를 참조하십시오.

감사와 보안의 관련성

감사는 시스템 사용에 대한 의심스럽거나 비정상적인 패턴을 밝혀내어 잠재적인 보안 침입을 감지하는 데 도움을 줍니다. 또한 감사는 의심스런 작업을 특정 사용자로 역추적할 수 있는 방법을 제공하므로 침입을 지연시키는 역할도 수행합니다. 자신의 작업이 감사되고 있다는 사실을 알고 있는 사용자는 악의적인 작업을 덜 시도하게 됩니다.

컴퓨터 시스템, 특히 네트워크에 있는 시스템을 보호하기 위해서는 시스템 프로세스나 사용자 프로세스가 시작되기 전에 작업을 제어하는 방식이 필요합니다. 보안을 위해서는 작업이 발생할 때 작업을 모니터하는 도구가 필요합니다. 또한 보안을 위해서는 작업이 발생한 후 작업 보고서가 필요합니다.

대부분의 감사 작업에는 지정된 매개변수를 충족하는 현재 이벤트 모니터링 및 이벤트 보고가 포함되므로 사용자가 로그인하거나 시스템 프로세스가 시작되기 전에 감사 매개변수를 설정합니다. 감사 서비스에서 이러한 이벤트를 모니터하고 보고하는 방법에 대한 자세한 내용은 2장. 감사 계획 및 3장. 감사 서비스 관리를 참조하십시오.

감사는 해커의 무단 침입을 막을 수는 없습니다. 하지만 감사 서비스는 특정 사용자가 특정 작업을 특정 시간과 날짜에 수행했다고 보고할 수 있습니다. 감사 보고서에서는 침입 경로와 사용자 이름으로 사용자를 식별할 수 있습니다. 이러한 정보는 터미널에 즉시 보고되고 나중에 분석을 위해 파일에 저장할 수 있습니다. 따라서 감사 서비스는 다음을 확인하는 데 도움을 주는 데이터를 제공합니다.

- 시스템 보안이 침해된 방법
- 원하는 레벨의 보안 유지를 위해 막아야 하는 보안 허점

감사가 작동하는 방식

감사는 지정된 이벤트가 발생할 때 감사 레코드를 생성합니다. 가장 일반적으로 감사 레코드를 생성하는 이벤트에는 다음이 포함됩니다.

- 시스템 시작 및 시스템 종료
- 로그인 및 로그아웃
- 프로세스 만들기/삭제 또는 스레드 만들기/삭제
- 객체 열기, 닫기, 만들기, 삭제 또는 이름 바꾸기
- 권한 사용
- 식별 작업 및 인증 작업
- 프로세스 또는 사용자에 의한 권한 변경
- 관리 작업(예: 패키지 설치)
- 사이트 특정 응용 프로그램

감사 레코드는 세 가지 소스에서 생성됩니다.

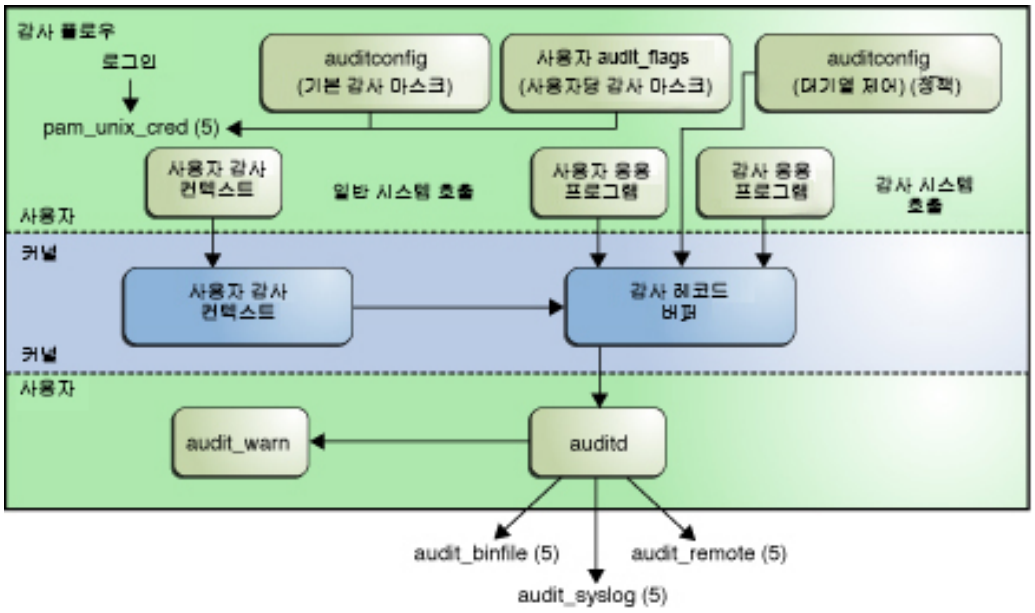
- 응용 프로그램
- asynchronous audit event(비동기 감사 이벤트)의 결과
- 프로세스 시스템 호출의 결과

관련 이벤트 정보가 캡처된 후 정보는 감사 레코드로 만들어집니다. 각 감사 레코드에는 이벤트를 식별하는 정보, 이벤트의 발생 원인, 이벤트의 시간 및 기타 관련 정보가 포함됩니다. 그러면 이 레코드가 감사 대기열에 배치되고 보관을 위한 활성 플러그인에 전송됩니다. 모든 플러그인을 활성화할 수 있지만 적어도 하나의 플러그인은 활성화되어야 합니다. 플러그인에 대한 자세한 내용은 "감사를 구성하는 방법" [20] 및 "감사 플러그인 모듈" [15]을 참조하십시오.

감사를 구성하는 방법

시스템 구성 중 모니터할 감사 레코드의 클래스를 사전 선택하게 됩니다. 또한 개별 사용자에 대해 수행되는 감사의 정도를 세밀하게 조정할 수 있습니다. 다음 그림은 Oracle Solaris에서 감사의 자세한 플로우를 보여줍니다.

그림 1-1 감사 플로우



감사 데이터가 커널에 수집된 후 플러그인은 데이터를 적합한 위치로 분배합니다.

- `audit_binfile` 플러그인은 이진 감사 레코드를 `/var/audit` 파일 시스템에 둡니다. 기본적으로 `audit_binfile` 플러그인이 활성화됩니다. 사후 선택 도구를 사용하여 감사 추적의 관심 있는 부분을 검사할 수 있습니다.

  감사 파일은 하나 이상의 ZFS 풀에 저장할 수 있습니다. 이러한 풀은 서로 다른 시스템 및 서로 다르지만 서로 연결된 네트워크에 있을 수 있습니다. 서로 연결된 감사 파일 모음은 감사 추적으로 간주됩니다.
- `audit_remote` 플러그인은 이진 감사 레코드를 보호된 링크를 통해 원격 저장소로 보냅니다.
- `audit_syslog` 플러그인은 감사 레코드의 텍스트 요약을 `syslog` 유틸리티로 보냅니다.

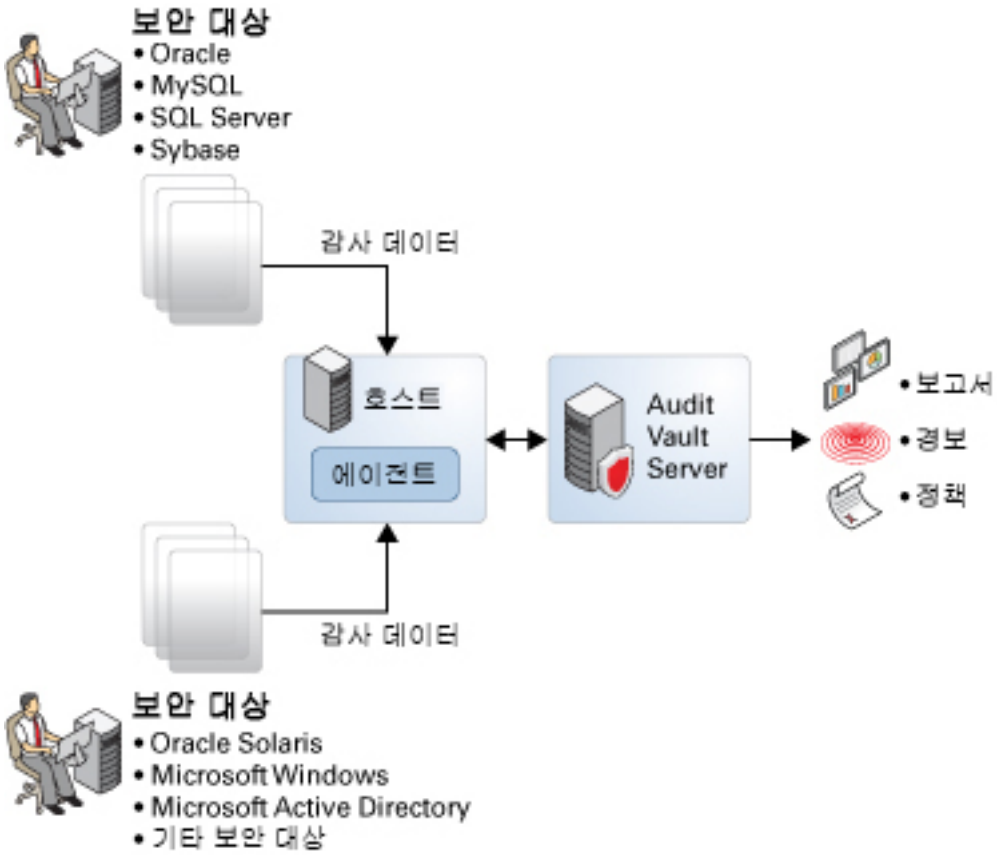
비전역 영역을 설치하는 시스템은 전역 영역의 모든 영역을 동일하게 감사할 수 있습니다. 또한 비전역 영역에서 서로 다른 레코드를 수집하도록 이러한 시스템을 구성할 수 있습니다. 자세한 내용은 "감사 및 Oracle Solaris 영역" [110]을 참조하십시오.

감사 레코드의 저장 및 분석을 위한 Oracle Audit Vault and Database Firewall 사용

Oracle Solaris 시스템의 감사 레코드는 Oracle Audit Vault and Database Firewall, 릴리스 12.1.0.0에 플러그인할 수 있습니다. Oracle Audit Vault and Database Firewall은 Oracle 및 비Oracle 데이터베이스로부터의 감사 데이터 통합 및 모니터링을 자동화합니다. 그런 다음 Oracle Solaris 시스템에서 감사된 이벤트에 대한 분석 및 보고서에 Oracle Audit Vault and Database Firewall을 사용할 수 있습니다. 자세한 내용은 Oracle Audit Vault and Database Firewall (http://www.oracle.com/technetwork/products/audit-vault/overview/index.html)을 참조하십시오.

다음 그림은 Oracle Audit Vault and Database Firewall이 지정된 보안 대상에서 Oracle Solaris 감사 레코드를 수집하는 방법을 보여줍니다. 보안 대상은 감사 레코드 또는 데이터를 저장하는 시스템입니다.

그림 1-2 Oracle Solaris 및 Audit Vault



호스트는 Oracle Audit Vault and Database Firewall과 통신하는 AV 에이전트를 실행하도록 지정됩니다. 에이전트를 통해 Oracle Audit Vault and Database Firewall은 보안 대상으로부터 감사 데이터를 수신하고 처리할 수 있습니다. 에이전트는 보안 대상의 지정된 감사 추적으로부터 감사 레코드를 읽습니다. 이러한 감사 레코드는 고유 이진 형식으로 인코딩됩니다. 에이전트는 Oracle Audit Vault and Database Firewall에서 구문 분석 가능한 형식으로 데이터를 변환합니다. Oracle Audit Vault and Database Firewall은 데이터를 수신하고 필요에 따라 관리자 및 보안 관리자를 위한 보고서를 생성합니다.

에이전트는 별도의 호스트 또는 시스템 대신 보안 대상에 설치할 수 있습니다. 에이전트를 포함하는 다중 호스트를 Audit Vault 서버에 연결되도록 구성할 수도 있습니다. 하지만 보안 대상을 등록할 때는 AV 서버가 감사 데이터를 가져오기 위해 통신하는 특정 호스트를 지정합니다.

Oracle Solaris 보안 대상 및 비Oracle Solaris 보안 대상으로부터 감사 레코드를 모두 수락하도록 Oracle Audit Vault and Database Firewall을 구성하려면 에이전트가 지정된 호스트 시스템에 설치되어 활성화되었는지 확인합니다. 자세한 내용은 Oracle Audit Vault and Database Firewall 설명서 (http://www.oracle.com/technetwork/products/audit-vault/documentation/index.html)를 참조하십시오.

Oracle Solaris 영역이 있는 시스템 감사

영역은 단일 인스턴스의 Oracle Solaris OS 내에 만들어지는 가상화 운영 체제 환경입니다. 감사 서비스는 영역에서의 작업을 포함한 전체 시스템을 감사합니다. 비전역 영역을 설치한 시스템은 단일 감사 서비스를 실행하여 모든 영역을 동일하게 감사할 수 있습니다. 또는 전역 영역을 포함하여 영역당 하나의 감사 서비스를 실행할 수 있습니다.

다음 조건을 충족하는 사이트는 단일 감사 서비스를 실행할 수 있습니다.

- 사이트에 단일 이미지 감사 추적이 필요합니다.
- 비전역 영역이 응용 프로그램 컨테이너로 사용됩니다. 영역이 관리 도메인의 일부입니다. 즉, 사용자 정의된 이름 지정 서비스 파일이 있는 비전역 영역이 없습니다.

  시스템의 모든 영역이 하나의 관리 도메인 내에 있는 경우 `zonename` 감사 정책을 사용하여 서로 다른 영역에서 구성된 감사 이벤트를 구별할 수 있습니다.
- 관리자가 낮은 감사 오버헤드를 원합니다. 전역 영역 관리자가 모든 영역을 동일하게 감사합니다. 또한 전역 영역의 감사 데몬이 시스템의 모든 영역을 서비스합니다.

다음 조건을 충족하는 사이트는 영역당 하나의 감사 서비스를 실행할 수 있습니다.

- 사이트에 단일 이미지 감사 추적이 필요하지 않습니다.
- 비전역 영역에 사용자 정의된 이름 지정 서비스 파일이 있습니다. 이러한 별도의 관리 도메인은 대개 서버로 작동합니다.
- 개별 영역 관리자가 자신이 관리하는 영역의 감사를 제어하고자 합니다. 영역별 감사에서 영역 관리자는 자신이 관리하는 영역에 대한 감사를 사용 또는 사용 안함으로 설정할지 결정할 수 있습니다.

영역별 감사의 장점은 각 영역에 대한 사용자 정의된 감사 추적과 영역을 기준으로 영역에 대한 감사를 사용 안함으로 설정할 수 있는 기능입니다. 이러한 장점은 관리 오버헤드로 약화될 수 있습니다. 각 영역 관리자가 감사를 관리해야 합니다. 각 영역은 고유의 감사 데몬에서 실행되고 고유의 감사 대기열과 감사 로그를 가집니다. 이러한 감사 로그는 관리해야 합니다.

♦ ♦ ♦ 2 장

2 감사 계획

이 장에서는 Oracle Solaris 설치에 대해 감사 서비스 사용자 정의를 계획하는 방법을 설명합니다.

감사 개요는 1장. Oracle Solaris의 감사 정보를 참조하십시오. 사용자 사이트에서 감사를 구성하는 절차는 다음 장을 참조하십시오.

- 3장. 감사 서비스 관리
- 4장. 시스템 작업 모니터링
- 5장. 감사 데이터 작업
- 6장. 감사 서비스 문제 분석 및 해결

참조 정보는 7장. 감사 참조를 참조하십시오.

감사 계획 개념

감사할 작업의 종류에 대해 선별하고자 합니다. 동시에 유용한 감사 정보를 수집하고자 합니다. 그리고 누구를 감사하고 무엇을 감사할지 신중하게 계획해야 합니다. 기본 `audit_binfile` 플러그인을 사용하는 경우 감사 파일이 빠르게 커지면서 사용 가능한 공간을 채우므로 충분한 디스크 공간을 할당해야 합니다.

단일 시스템 감사 추적 계획

참고 - 단일 시스템 감사 추적 구현은 `audit_binfile` 플러그인에만 적용됩니다.

단일 관리 도메인 내의 시스템은 단일 시스템 이미지 감사 추적을 만들 수 있습니다.

사이트에 대해 단일 시스템 이미지 감사 추적을 만들려면 다음 요구 사항을 따릅니다.

- 모든 시스템에 대해 동일한 이름 지정 서비스를 사용합니다.

  감사 레코드의 올바른 구현을 위해서는 `passwd`, `group` 및 `hosts` 파일이 일관적이어야 합니다.
- 모든 시스템에서 동일하게 감사 서비스를 구성합니다. 서비스 설정 표시 및 수정에 대한 자세한 내용은 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오.
- 모든 시스템에 대해 동일한 `audit_warn`, `audit_event` 및 `audit_class` 파일을 사용합니다.

시스템에서 감사를 사용으로 설정할 때의 추가 고려 사항은 감사할 대상(사용자 및 객체)을 계획하는 방법 [28]을 참조하십시오.

영역에서 감사 계획

시스템에 비전역 영역이 포함되어 있는 경우 전역 영역과 동일하게 영역을 감사하거나 각 비전역 영역에 대한 감사 서비스를 별도로 구성, 사용 및 사용 안함으로 설정할 수 있습니다. 예를 들어, 비전역 영역만 감사하고 전역 영역은 감사하지 않을 수 있습니다.

장단점에 대한 자세한 내용은 “Oracle Solaris 영역이 있는 시스템 감사” [23]를 참조하십시오.

영역에서 감사를 구현할 때는 다음과 같은 옵션을 사용할 수 있습니다.

모든 영역에 대해 단일 감사 서비스 구현

모든 영역을 동일하게 감사하면 단일 이미지 감사 추적을 만들 수 있습니다. 단일 이미지 감사 추적은 `audit_binfile` 또는 `audit_remote` 플러그인을 사용할 때 발생하며, 시스템의 모든 영역이 한 관리 도메인의 일부입니다. 그러면 모든 영역의 레코드가 동일한 설정으로 사전 선택되므로 감사 레코드를 쉽게 비교할 수 있습니다.

이 구성은 모든 영역을 한 시스템의 일부로 취급합니다. 전역 영역은 시스템에서 하나의 감사 서비스만 실행하고 모든 영역에 대한 감사 레코드를 수집합니다. 전역 영역에서만 `audit_class` 및 `audit_event` 파일을 사용자 정의한 다음 이러한 파일을 모든 비전역 영역에 복사합니다.

모든 영역에 대해 단일 감사 서비스를 구성할 때 다음과 같은 지침을 따릅니다.

- 모든 영역에 대해 동일한 이름 지정 서비스를 사용합니다.

참고 - 이름 지정 서비스 파일이 비전역 영역에서 사용자 정의되고 `perzone` 정책이 설정되지 않으면 감사 도구를 신중하게 사용하여 유용한 레코드를 선택해야 합니다. 한 영역의 사용자 ID는 다른 영역에서 동일한 ID를 가진 다른 사용자를 가리킬 수 있습니다.

- 감사 레코드에 영역의 이름이 포함되도록 합니다.

  영역 이름을 감사 레코드의 일부로 두려면 전역 영역에서 zonename 정책을 설정합니다. 그러면 auditreduce 명령이 감사 추적에서 영역별로 감사 이벤트를 선택할 수 있습니다. 예는 auditreduce(1M) 매뉴얼 페이지를 참조하십시오.

단일 이미지 감사 추적을 계획하려면 감사할 대상(사용자 및 객체)을 계획하는 방법 [28]을 참조하십시오. 첫번째 단계부터 시작합니다. 또한 전역 영역 관리자는 감사 레코드의 디스크 공간 계획 방법 [30]에 설명된 대로 저장소를 마련해 두어야 합니다.

영역당 하나의 감사 서비스 구현

서로 다른 영역에서 서로 다른 이름 지정 서비스 데이터베이스를 사용하거나 영역 관리자가 해당 영역의 감사를 제어하고자 하는 경우 영역별 감사를 구성하려면 선택합니다.

참고 - 비전역 영역을 감사하려면 perzone 정책을 설정해야 하지만, 전역 영역에서는 감사 서비스를 사용으로 설정하지 않아도 됩니다. 비전역 영역 감사가 구성되고 해당 감사 서비스가 전역 영역과 별도로 사용 및 사용 안함으로 설정됩니다.

- 영역별 감사를 구성할 경우 전역 영역에서 perzone 감사 정책을 설정합니다. 비전역 영역이 처음으로 부트되기 전에 영역별 감사가 설정된 경우 감사는 영역의 최초 부트부터 시작됩니다. 감사 정책을 설정하려면 영역별 감사를 구성하는 방법 [65]을 참조하십시오.
- 각 영역 관리자가 영역에 대한 감사를 구성합니다.

  비전역 영역 관리자는 perzone 및 ahlt를 제외한 모든 정책 옵션을 설정할 수 있습니다.
- 각 영역 관리자는 영역의 감사를 사용 또는 사용 안함으로 설정할 수 있습니다.
- 검토 중 발생 영역으로 추적할 수 있는 레코드를 생성하려면 zonename 감사 정책을 설정합니다.

참고 - 영역별 감사에서 audit_binfile 플러그인이 활성 상태인 경우 각 영역 관리자는 감사 레코드의 디스크 공간 계획 방법 [30]에 설명된 대로 영역마다 저장소를 마련해 두어야 합니다. 추가 계획 지침은 감사할 대상(사용자 및 객체)을 계획하는 방법 [28]을 참조하십시오.

계획 감사

다음 작업 맵에서는 디스크 공간 및 기록할 이벤트를 계획하는 데 필요한 주요 작업을 안내합니다.

표 2-1 감사 계획 작업 맵

작업	수행 방법
감사할 대상(사용자 및 객체)을 결정합니다.	감사할 대상(사용자 및 객체)을 계획하는 방법 [28]

작업	수행 방법
감사 추적에 대한 저장 공간을 계획합니다.	감사 레코드의 디스크 공간 계획 방법 [30]
원격 서버에 대한 감사 추적 전송 계획	감사 레코드를 원격 저장소에 스트리밍하기 위한 준비 방법 [31]

▼ 감사할 대상(사용자 및 객체)을 계획하는 방법

시작하기 전에 비전역 영역을 구현하는 경우 이 절차를 사용하기 전에 "영역에서 감사 계획" [26]을 검토하십시오.

1. **감사 정책을 결정합니다.**

   기본적으로 `cnt` 정책만 사용으로 설정됩니다.

   `auditconfig -lspolicy` 명령을 사용하여 사용 가능한 정책 옵션에 대한 설명을 봅니다.

   - 정책 옵션의 효과는 "감사 정책 이해" [32]를 참조하십시오.
   - `cnt` 정책의 효과는 "비동기 및 동기 이벤트에 대한 감사 정책" [113]을 참조하십시오.
   - 감사 정책을 설정하려면 감사 정책을 변경하는 방법 [46]을 참조하십시오.

2. **이벤트-클래스 매핑의 수정을 원하는지 여부를 결정합니다.**

   거의 모든 상황에서 기본 매핑이면 충분합니다. 하지만 새 클래스를 추가하거나 클래스 정의를 변경하거나 특정 시스템 호출의 레코드가 유용하지 않다고 판단되는 경우 이벤트-클래스 매핑을 수정할 수도 있습니다.

   예는 감사 이벤트의 클래스 멤버쉽을 변경하는 방법 [52]을 참조하십시오.

3. **사전 선택할 감사 클래스를 결정합니다.**

   감사 클래스를 추가하거나 기본 클래스를 변경하는 가장 좋은 시기는 사용자가 시스템에 로그인하기 전입니다.

   `auditconfig` 명령에 `-setflags` 및 `-setnaflags` 옵션을 사용하여 사전 선택하는 감사 클래스는 모든 사용자와 프로세스에 적용됩니다. 성공, 실패 또는 둘 다에 대해 클래스를 사전 선택할 수 있습니다.

   감사 클래스 목록은 `/etc/security/audit_class` 파일을 검토하십시오.

4. **시스템 전역 사전 선택에 대한 사용자 수정을 결정합니다.**

   일부 사용자를 시스템과 다르게 감사하도록 결정할 경우에는 개별 사용자 또는 권한 프로파일에 대한 `audit_flags` 보안 속성을 수정할 수 있습니다. 사용자 사전 선택 마스크는 감사 플래그가 명시적으로 설정되었거나 명시적인 감사 플래그로 권한 프로파일이 지정된 사용자를 위해 수정되었습니다.

   절차는 사용자의 감사 특성을 구성하는 방법 [42]을 참조하십시오. 적용되는 감사 플래그 값은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 권한 검색 순서"를 참조하십시오.

5. **audit_warn 전자 메일 별칭을 어떻게 관리할지 결정합니다.**

   audit_warn 스크립트는 감사 시스템에서 관리 주의가 요구되는 상황을 감지할 때마다 실행됩니다. 기본적으로 audit_warn 스크립트는 전자 메일을 audit_warn 별칭에 보내고 메시지를 콘솔로 보냅니다.

   별칭을 설정하려면 audit_warn 전자 메일 별칭을 구성하는 방법 [50]을 참조하십시오.

6. **감사 레코드를 어떤 형식으로 어디에 수집할지 결정합니다.**

   세 가지 옵션이 있습니다.

   - 기본적으로 이진 감사 레코드를 로컬에 저장합니다. 기본 저장소 디렉토리는 /var/audit.입니다 audit_binfile 플러그인을 추가로 구성하려면 감사 파일에 대한 ZFS 파일 시스템을 만드는 방법 [70]을 참조하십시오.
   - audit_remote 플러그인을 사용하여 이진 감사 레코드를 보호된 원격 저장소로 스트리밍합니다. 레코드에 대한 수신자가 있어야 합니다. 요구 사항은 "원격 저장소 관리" [18]를 참조하십시오. 절차는 원격 저장소에 감사 파일을 보내는 방법 [76]을 참조하십시오.
   - audit_syslog 플러그인을 사용하여 감사 레코드 요약을 syslog에 보냅니다. 절차는 syslog 감사 로그를 구성하는 방법 [82]을 참조하십시오.

     이진과 syslog 형식에 대한 비교는 "감사 로그" [15]를 참조하십시오.

7. **관리자에게 디스크 공간 축소에 대해 언제 경고할지 결정합니다.**

   참고 - 이 단계는 audit_binfile 플러그인에만 적용됩니다.

   감사 파일 시스템의 디스크 공간이 최소 여유 공간 비율 또는 소프트 한계 아래로 떨어지면 감사 서비스는 다음 사용 가능한 감사 디렉토리로 전환합니다. 그런 다음 서비스에서는 소프트 한계를 초과했다는 경고를 보냅니다.

   최소 여유 공간 비율을 설정하는 방법을 보려면 예 4-7. "경고에 대한 소프트 제한 설정"을 참조하십시오.

8. **모든 감사 디렉토리가 가득 찰 경우 어떤 작업을 수행할지 결정합니다.**

   참고 - 이 단계는 audit_binfile 플러그인에만 적용됩니다.

   기본 구성에서는 audit_binfile 플러그인이 활성화되고 cnt 정책이 설정됩니다. 이 구성에서는 커널 감사 대기열이 가득 차면 시스템이 계속 작동합니다. 시스템에서는 삭제되는 감사 레코드 수를 계산하지만 이벤트를 기록하지 않습니다. 더욱 높은 보안을 위해 cnt 정책을 사용 안함으로 설정하고 ahlt 정책을 사용으로 설정할 수 있습니다. 비동기 이벤트를 감사 대기열에 둘 수 없으면 ahlt 정책은 시스템을 중지시킵니다.

   하지만 audit_binfile 대기열이 가득 차고 다른 활성 플러그인에 대한 대기열이 가득 차지 않으면 커널 대기열이 가득 차지 않은 플러그인에 계속해서 레코드를 보냅니다. audit_binfile 대기열에서 다시 레코드를 수신할 수 있게 되면 감사 서비스가 레코드 보내기를 재개합니다.

cnt 및 ahlt 정책 옵션에 대한 자세한 내용은 “비동기 및 동기 이벤트에 대한 감사 정책” [113]을 참조하십시오. 이러한 정책 옵션을 구성하는 방법을 보려면 예 3-10. “ahlt 감사 정책 옵션”을 참조하십시오.

참고 - 적어도 하나의 플러그인에 대한 대기열이 감사 레코드를 수신하지 않으면 cnt 또는 ahlt 정책이 트리거되지 않습니다.

감사 레코드의 디스크 공간 계획

audit_binfile 플러그인은 감사 추적을 만듭니다. 감사 추적에는 전용 파일 공간이 필요합니다. 이 공간은 사용 가능하고 안전해야 합니다. 시스템에서는 초기 저장소에 대해 /var/audit 파일 시스템을 사용합니다. 감사 파일에 대해 추가 감사 파일 시스템을 구성할 수 있습니다. 다음 절차에서는 감사 추적 저장소를 계획할 때 해결해야 하는 문제를 다룹니다.

▼ 감사 레코드의 디스크 공간 계획 방법

시작하기 전에 비전역 영역을 구현하는 경우 이 절차를 사용하기 전에 “영역에서 감사 계획” [26]을 완료하십시오.

이 절차에서는 audit_binfile 플러그인을 사용한다고 가정합니다.

1. **사이트에서 필요한 감사의 양을 결정합니다.**

   사이트의 보안 요구 사항과 감사 추적용 디스크 공간 가용성의 균형을 맞춥니다.

   사이트 보안을 유지하면서 공간 요구 사항을 줄이는 방법과 감사 저장소를 설계하는 방법은 “감사 비용 제어” [34] 및 “효율적으로 감사” [35]를 참조하십시오.

   실제 단계는 “감사 레코드 볼륨이 큼” [102], 전용 파일 시스템에서 감사 파일을 압축하는 방법 [60] 및 예 5-4. “감사 파일 결합 및 줄이기”를 참조하십시오.

2. **감사할 시스템을 결정하고 감사 파일 시스템을 구성합니다.**

   사용할 모든 파일 시스템 목록을 만듭니다. 구성에 대한 자세한 내용은 “감사 추적 저장 및 관리” [17] 및 auditreduce(1M) 매뉴얼 페이지를 참조하십시오. 감사 파일 시스템을 지정하려면 감사 추적에 대한 감사 공간을 지정하는 방법 [73]을 참조하십시오.

3. **모든 시스템의 시계를 동기화합니다.**

   자세한 내용은 “신뢰할 수 있는 시간 기록 유지” [18]를 참조하십시오.

감사 레코드를 원격 저장소에 스트리밍하기 위한 준비

audit_remote 플러그인은 audit_binfile 플러그인이 로컬 감사 파일에 쓰는 것과 동일한 형식으로 ARS에 이진 감사 추적을 보냅니다. audit_remote 플러그인은 libgss 라이브러리를 사용하여 ARS를 인증하고 GSS-API 방식을 사용하여 개인 정보의 전송 및 무결성을 보호합니다. 참조 정보는 "Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리 "의 "Kerberos 서비스란?" 및 "Oracle Solaris 11.2의 Kerberos 및 기타 인증 서비스 관리 "의 "Kerberos 유틸리티"를 참조하십시오.

현재까지 지원되는 유일한 GSS-API 방식은 kerberosv5입니다. 자세한 내용은 mech(4) 매뉴얼 페이지를 참조하십시오.

▼ 감사 레코드를 원격 저장소에 스트리밍하기 위한 준비 방법

참고 - 식별된 ARS(감사 원격 서버)로 구성된 Kerberos 영역이 있고 해당 영역 내에 모든 감사되는 시스템이 포함되는 경우 이 절차를 건너뛸 수 있습니다. ARS 및 감사되는 시스템을 구성하는 단계는 감사 파일에 대한 원격 저장소를 구성하는 방법 [78] 및 원격 저장소에 감사 파일을 보내는 방법 [76]을 참조하십시오.

Kerberos 영역이 구성되었는지 여부를 확인하려면 다음 명령을 실행합니다. 샘플 출력은 Kerberos가 시스템에 설치되지 않았음을 나타냅니다.

```
# pkg info system/security/kerberos-5
pkg: info: no packages matching these patterns are installed on the system.
```

시작하기 전에 이 절차에서는 audit_remote 플러그인을 사용한다고 가정합니다.

1. **마스터 KDC(키 배포 센터) 패키지를 설치합니다.**

   ARS로 작동하는 시스템을 사용하거나 인접 시스템을 사용할 수 있습니다. ARS는 상당히 많은 양의 트래픽을 마스터 KDC로 보냅니다.

   ```
   # pkg install pkg:/system/security/kerberos-5
   ```

   마스터 KDC는 Kerberos kdcmgr 및 kadmin 명령을 사용하여 영역을 관리합니다. 자세한 내용은 kdcmgr(1M) 및 kadmin(1M) 매뉴얼 페이지를 참조하십시오.

2. **감사 레코드를 ARS로 보내는 모든 감사되는 시스템에서 마스터 KDC 패키지를 설치합니다.**

   ```
   # pkg install pkg:/system/security/kerberos-5
   ```

   이 패키지에는 kclient 명령이 포함됩니다. 이러한 시스템에서 kclient 명령을 사용하여 KDC와 연결합니다. 자세한 내용은 kclient(1M) 매뉴얼 페이지를 참조하십시오.

3. **KDC 영역의 클럭을 동기화합니다.**

감사되는 시스템과 ARS 간의 클럭 불균형이 너무 크면 연결 시도가 실패합니다. “이진 감사 파일 이름 지정 규칙” [115]에 설명된 대로 연결이 설정된 후 ARS의 로컬 시간에 따라 저장된 감사 파일의 이름이 결정됩니다.

클럭에 대한 자세한 내용은 “신뢰할 수 있는 시간 기록 유지” [18]를 참조하십시오.

감사 정책 이해

감사 정책은 로컬 시스템에 대한 감사 레코드의 특성을 결정합니다. `auditconfig` 명령을 사용하여 이러한 정책을 설정합니다. 자세한 내용은 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오.

저장소 요구 사항 및 시스템 처리 수요를 최소화하기 위해 대부분의 감사 정책 옵션은 사용 안함으로 설정됩니다. 이러한 옵션은 감사 서비스의 등록 정보이며 시스템 부트 시 적용되는 정책을 결정합니다. 자세한 내용은 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오.

다음 표를 참조하여 사이트의 요구 사항이 하나 이상의 감사 정책 옵션을 사용으로 설정하여 발생하는 추가 오버헤드보다 우선하는지 여부를 결정합니다.

표 2-2 감사 정책 옵션의 효과

정책 이름	설명	정책 고려 사항
`ahlt`	이 정책은 비동기 이벤트에만 적용됩니다. 사용 안함으로 설정할 경우, 이 정책은 감사 레코드를 생성하지 않고 이벤트가 완료되도록 허용합니다. 사용으로 설정할 경우, 이 정책은 감사 대기열이 가득 찰 경우 시스템을 중지시킵니다. 감사 대기열을 정리하고 감사 레코드에 사용 가능한 공간을 만들며 재부트하려면 관리자의 개입이 필요합니다. 이 정책은 전역 영역에서만 사용으로 설정할 수 있습니다. 정책은 모든 영역에 영향을 줍니다.	시스템 가용성이 보안보다 중요할 경우에는 사용 안함 옵션을 선택하는 것이 좋습니다. 보안이 가장 중요한 환경에서는 사용 옵션을 선택하는 것이 좋습니다. 자세한 내용은 “비동기 및 동기 이벤트에 대한 감사 정책” [113]을 참조하십시오.
`arge`	사용 안함으로 설정하면 이 정책은 실행된 프로그램의 환경 변수를 execve 감사 레코드에서 뺍니다. 사용으로 설정된 경우 이 정책은 실행된 프로그램의 환경 변수를 execve 감사 레코드에 추가합니다. 결과 감사 레코드에는 이 정책이 사용 안함으로 설정될 때 더 자세한 정보가 포함됩니다.	사용 안함 옵션으로 설정하면 사용 옵션보다 적은 정보를 수집합니다. 비교는 사용자의 모든 명령을 감사하는 방법 [54]을 참조하십시오. 적은 수의 사용자를 감사할 때는 사용 옵션을 선택하는 것이 좋습니다. 또한 ex 감사 클래스의 프로그램에서 사용되고 있는 환경 변수가 확실하지 않은 경우 이 옵션이 유용합니다.
`argv`	사용 안함으로 설정하면 이 정책은 실행된 프로그램의 인수를 execve 감사 레코드에서 뺍니다. 사용으로 설정하면 이 정책은 실행된 프로그램의 인수를 execve 감사 레코드에 추가합니다. 결과 감사 레코드에는 이 정책이 사용 안함으로 설정될 때 더 자세한 정보가 포함됩니다.	사용 안함 옵션으로 설정하면 사용 옵션보다 적은 정보를 수집합니다. 비교는 사용자의 모든 명령을 감사하는 방법 [54]을 참조하십시오. 적은 수의 사용자를 감사할 때는 사용 옵션을 선택하는 것이 좋습니다. 또한 ex 감사 클래스

정책 이름	설명	정책 고려 사항
		에서 비정상적인 프로그램이 실행되고 있다고 판단되는 경우 이 옵션이 유용합니다.
cnt	사용 안함으로 설정하면 이 정책은 사용자 또는 응용 프로그램 실행을 차단합니다. 감사 대기열이 가득 차서 감사 레코드를 감사 추적에 추가할 수 없으면 차단이 발생합니다. 사용으로 설정하면 이 정책은 감사 레코드를 생성하지 않고 이벤트가 완료되도록 허용합니다. 정책에서는 삭제되는 감사 레코드의 수를 계산합니다.	보안이 가장 중요한 환경에서는 사용 안함 옵션을 선택하는 것이 좋습니다. 시스템 가용성이 보안보다 중요할 경우에는 사용 옵션을 선택하는 것이 좋습니다. 자세한 내용은 "비동기 및 동기 이벤트에 대한 감사 정책" [113]을 참조하십시오.
group	사용 안함으로 설정하면 이 정책은 그룹 목록을 감사 레코드에 추가하지 않습니다. 사용으로 설정된 경우 이 정책은 그룹 목록을 모든 감사 레코드에 특별한 토큰으로 추가합니다.	사용 안함 옵션은 일반적으로 사이트 보안 요구 사항을 충족합니다. 주체가 속하는 보조 그룹을 감사해야 하는 경우 사용 옵션을 선택하는 것이 좋습니다.
path	사용 안함으로 설정하면 이 정책은 시스템 호출 중 사용되는 하나의 경로만 감사 레코드에 기록합니다. 사용으로 설정하면 이 정책은 감사 이벤트와 함께 사용되는 모든 경로를 모든 감사 레코드에 기록합니다.	사용 안함 옵션은 하나의 경로만 감사 레코드에 추가합니다. 사용 옵션은 시스템 호출 중 사용되는 각 파일 이름이나 경로를 감사 레코드에 path 토큰으로 입력합니다.
perzone	사용 안함으로 설정하면 이 정책은 시스템에 대해 단일 감사 구성을 유지합니다. 하나의 감사 서비스가 전역 영역에서 실행됩니다. zonename 감사 토큰이 사전 선택된 경우 특정 영역의 감사 이벤트를 감사 레코드에서 찾을 수 있습니다. 사용으로 설정하면 이 정책은 각 영역에 대해 별도의 감사 구성, 감사 대기열 및 감사 로그를 유지합니다. 감사 서비스가 각 영역에서 실행됩니다. 이 정책은 전역 영역에서만 사용으로 설정할 수 있습니다.	각 영역에 대해 별도의 감사 로그, 대기열 및 데몬을 유지해야 하는 특별한 이유가 있을 경우 사용 안함 옵션이 유용합니다. zonename 감사 토큰으로 감사 레코드를 간단히 검사하여 효과적으로 시스템을 모니터할 수 없는 경우 사용 옵션이 유용합니다.
public	사용 안함으로 설정하면 이 정책은 파일 읽기가 사전 선택되었을 때 공용 객체의 읽기 전용 이벤트를 감사 추적에 추가하지 않습니다. 읽기 전용 이벤트를 포함하는 감사 클래스에는 fr, fa 및 cl이 있습니다. 사용으로 설정된 경우 이 정책은 적합한 감사 클래스가 사전 선택되었을 때 공용 객체의 모든 읽기 전용 감사 이벤트를 기록합니다.	사용 안함 옵션은 일반적으로 사이트 보안 요구 사항을 충족합니다. 사용 옵션은 거의 유용하지 않습니다.
seq	사용 안함으로 설정하면 이 정책은 시퀀스 번호를 모든 감사 레코드에 추가하지 않습니다. 사용으로 설정된 경우 이 정책은 시퀀스 번호를 모든 감사 레코드에 추가합니다. sequence 토큰에 시퀀스 번호가 포함됩니다.	감사가 부드럽게 실행되는 경우 사용 안함 옵션이면 충분합니다. cnt 정책이 사용으로 설정된 경우 사용 옵션을 선택하는 것이 좋습니다. seq 정책을 사용하여 데이터가 언제 폐기되었는지 확인할 수 있습니다. 또는 auditstat 명령을 사용하여 삭제된 레코드를 볼 수 있습니다.
trail	사용 안함으로 설정하면 이 정책은 trailer 토큰을 감사 레코드에 추가하지 않습니다.	사용 안함 옵션은 더 작은 감사 레코드를 만듭니다.

정책 이름	설명	정책 고려 사항
	사용으로 설정된 경우 이 정책은 trailer 토큰을 모든 감사 레코드에 추가합니다.	사용 옵션은 trailer 토큰을 사용하여 각 감사 레코드의 끝을 분명하게 표시합니다. trailer 토큰은 종종 sequence 토큰과 함께 사용됩니다. trailer 토큰은 손상된 감사 추적 복구에 도움이 됩니다.
zonename	사용 안함으로 설정하면 이 정책은 zonename 토큰을 감사 레코드에 포함시키지 않습니다.	영역별로 감사 동작을 추적할 필요가 없는 경우 사용 안함 옵션이 유용합니다.
	사용으로 설정하면 이 정책은 zonename 토큰을 모든 감사 레코드에 포함시킵니다.	영역에 따라 레코드를 사후 선택하여 영역별로 감사 동작을 격리하고 비교하고자 하는 경우 사용 옵션이 유용합니다.

감사 비용 제어

감사는 시스템 리소스를 소모하므로 기록되는 세부 정보의 정도를 제어해야 합니다. 감사할 대상을 결정할 때 다음 감사 비용을 고려하십시오.

- 처리 시간 증가 비용
- 감사 데이터의 분석 비용

기본 플러그인 audit_binfile을 사용하는 경우 감사 데이터의 저장소 비용도 고려해야 합니다.

감사 데이터의 처리 시간 증가 비용

처리 시간 증가 비용은 감사 비용 중 가장 적은 부분을 차지합니다. 일반적으로 감사는 이미지 처리, 복잡한 계산 등과 같이 프로세서를 많이 사용하는 작업 중에 수행되지 않습니다. 또한 audit_binfile 플러그인을 사용하는 경우 감사 관리자가 사후 선택 작업을 감사되는 시스템에서 감사 데이터 분석 전용 시스템으로 이동할 수 있습니다. 마지막으로 커널 이벤트가 사전 선택되지 않았으면 감사 서비스가 시스템 성능에 대해 별다른 영향을 주지 않습니다.

감사 데이터의 분석 비용

분석 비용은 대개 수집되는 감사 데이터의 양에 비례합니다. 분석 비용에는 감사 레코드를 병합하고 검토하는 데 필요한 시간이 포함됩니다.

audit_binfile 플러그인으로 수집된 레코드의 경우, 비용에는 레코드 및 해당 지원 이름 서비스 데이터베이스를 아카이브하고 레코드를 안전한 장소에 보관하는 데 필요한 시간도 포함됩니다. 지원 데이터베이스에는 groups, hosts 및 passwd가 포함됩니다.

생성하는 레코드가 적을수록 감사 추적을 분석하는 데 필요한 시간이 줄어듭니다. “감사 데이터의 저장소 비용” [35] 및 “효율적으로 감사” [35] 절에서는 효율적으로 감사하는

방법에 대해 설명합니다. 효율적인 감사는 감사 데이터의 양을 줄이면서 사이트의 보안 목표를 달성할 수 있는 충분한 정보를 제공합니다.

감사 데이터의 저장소 비용

audit_binfile 플러그인을 사용하는 경우 저장소 비용은 감사 비용 중 가장 많은 부분을 차지합니다. 감사 데이터의 양은 다음에 따라 달라집니다.

- 사용자 수
- 시스템 수
- 사용량
- 필요한 추적 가능 및 책임 가능 정도

이러한 요소는 사이트마다 다르므로 공식으로 디스크 공간의 양을 사전에 결정하여 감사 데이터 저장소를 마련해 둘 수 없습니다. 다음 정보에 따라 작업을 수행합니다.

- 감사 클래스 이해

  감사를 구성하기 전에 클래스에 포함된 이벤트의 유형을 이해해야 합니다. 감사 이벤트-클래스 매핑을 변경하여 감사 레코드 수집을 최적화할 수 있습니다.
- 감사 클래스를 현명하게 사전 선택하여 생성되는 레코드의 양을 줄입니다.

  전체 감사(즉, all 클래스 사용)는 디스크 공간을 빠르게 채웁니다. 프로그램 컴파일과 같은 단순한 작업도 큰 감사 파일을 생성할 수 있습니다. 보통 크기의 프로그램이 1분 내에 수천 개의 감사 레코드를 생성할 수 있습니다.

  예를 들어, file_read 감사 클래스 fr을 빼면 감사 양을 크게 줄일 수 있습니다. 실패한 작업에 대해서만 감사하도록 선택하면 종종 감사 양을 줄일 수 있습니다. 예를 들어, 실패한 file_read 작업 -fr에 대해 감사하면 모든 file_read 이벤트에 대해 감사할 때보다 훨씬 적은 수의 레코드를 생성할 수 있습니다.
- audit_binfile 플러그인을 사용하는 경우 효율적인 감사 파일 관리도 중요합니다. 예를 들어, 감사 파일 전용인 ZFS 파일 시스템을 압축할 수 있습니다.
- 사이트에 대한 감사 철학을 개발합니다.

  사이트에 필요한 추적 가능 양 및 관리하는 사용자 유형과 같은 측정 단위에 맞게 기준을 정합니다.

효율적으로 감사

다음 기술은 조직의 보안 목표를 달성하면서 더욱 효율적으로 감사하는 데 도움이 될 수 있습니다.

- 가능한 한 많은 감사 클래스에 대해 시스템 전역이 아닌 사용자 및 역할에 대한 감사 클래스만 사전 선택합니다.
- 특정 시점에 사용자의 일부만 무작위로 감사합니다.

- audit_binfile 플러그인이 활성화된 경우 파일을 필터링, 병합 및 압축하여 감사에 대한 디스크 저장소 요구 사항을 줄입니다. 파일 아카이브, 이동식 매체로 파일 전송 및 원격으로 파일 저장을 위한 절차를 개발합니다.
- 비정상적인 동작에 대해 감사 데이터를 실시간으로 모니터링합니다.
  - audit_syslog 플러그인 - syslog 파일에서 감사 레코드를 처리하기 위해 이미 개발한 관리 및 분석 도구를 확장할 수 있습니다.
  - audit_binfile 플러그인 - 특정 작업에 대한 감사 추적을 모니터링하기 위한 절차를 설정할 수 있습니다. 비정상적인 이벤트 감지 시 특정 사용자나 특정 시스템에 대한 감사 자동 증가를 트리거하는 스크립트를 작성할 수 있습니다.

    예를 들어, 다음을 수행하는 스크립트를 작성할 수 있습니다.

    1. 감사되는 시스템에서 감사 파일 만들기를 모니터합니다.
    2. tail 명령을 사용하여 감사 파일을 처리합니다.

       tail -0f 명령에서 praudit 명령을 통한 출력 파이프는 레코드가 생성될 때 감사 레코드 스트림을 반환할 수 있습니다. 자세한 내용은 tail(1) 매뉴얼 페이지를 참조하십시오.
    3. 비정상적인 메시지 유형 또는 기타 지표에 대해 이 스트림을 분석하고, 감사자에게 분석을 전달합니다.

       또는 스크립트를 사용하여 자동 응답을 트리거할 수 있습니다.
    4. 감사 파일 시스템에서 새로운 not_terminated 감사 파일 출현을 지속적으로 모니터합니다.
    5. 해당 파일이 더 이상 쓰여지지 않는 경우 남아 있는 tail 프로세스를 종료합니다.

♦ ♦ ♦ 3 장

3 감사 서비스 관리

이 장에서는 Oracle Solaris 시스템에서 감사를 구성하고 관리하는 데 유용한 절차를 제공합니다. 이 장에서는 다음 작업을 다룹니다.

- "감사 서비스의 기본 구성" [37]
- "감사 서비스 구성" [40]
- "감사 대상 사용자 정의" [53]
- "영역에서 감사 서비스 구성" [62]
- "예제: Oracle Solaris 감사 구성" [66]

또한 다음 장에서는 기타 감사 관리 작업에 대해 설명합니다.

- 4장. 시스템 작업 모니터링
- 5장. 감사 데이터 작업
- 6장. 감사 서비스 문제 분석 및 해결

감사 서비스에 대한 개요는 1장. Oracle Solaris의 감사 정보를 참조하십시오. 계획 제안은 2장. 감사 계획을 참조하십시오. 참조 정보는 7장. 감사 참조를 참조하십시오.

감사 서비스의 기본 구성

감사 서비스는 기본 구성을 포함하며 Oracle Solaris 11.2 설치 후 전역 영역에서 즉시 작동 가능합니다. 서비스를 사용으로 설정하거나 사용하도록 구성하기 위해 추가 작업이 필요하지 않습니다. 기본 구성에서는 감사 서비스가 다음 작업을 기록합니다.

- 로그인 및 로그아웃 작업
- su 명령 사용
- 화면 잠금 및 화면 잠금 해제 작업

서비스의 기본 구성은 시스템에 성능 영향을 주지 않기 때문에 성능 향상을 위해 이 서비스를 사용 안함으로 설정할 필요가 없습니다.

Audit Review 권한 프로파일과 같은 적합한 감사 관련 권한이 있으면 감사 로그를 볼 수 있습니다. 로그는 `/var/audit/`*hostname*에 저장됩니다. 이러한 파일은 `praudit`

및 auditreduce 명령을 사용하여 볼 수 있습니다. 자세한 내용은 “감사 추적 데이터 표시” [85]를 참조하십시오.

이 장의 이후 절에서는 기본 구성만으로 부족한 경우 감사 서비스 구성을 사용자 정의하기 위한 지침을 제공합니다.

감사 서비스 기본값 표시

감사 서비스는 다음 매개변수에 따라 제어됩니다.

- 지정 가능한 이벤트 및 지정 불가능한 이벤트의 클래스
- 감사 정책
- 감사 플러그인
- 대기열 제어

감사 서비스 기본값을 표시하려면 일반적으로 auditconfig -get* 하위 명령을 사용합니다. 이 하위 명령은 -getflags -getpolicy 또는 -getqctrl과 같이 별표(*)로 표시된 매개변수의 현재 구성을 보여줍니다. 지정 불가능한 이벤트의 클래스에 대한 정보를 표시하려면 auditconfig -getnaflags 하위 명령을 사용합니다.

auditconfig 명령에 대한 자세한 내용은 auditconfig(1M) 매뉴얼 페이지를 참조하십시오.

참고 - 감사 서비스 구성을 표시하려면 Audit Configuration 또는 Audit Control 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

다음 예제는 기본 감사 구성 설정을 표시하기 위해 사용할 적합한 명령 구문을 보여줍니다.

예 3-1 이벤트에 대한 기본 클래스 표시

이 예제에서는 두 개의 하위 명령을 사용해서 지정 가능한 이벤트 및 지정 불가능한 이벤트에 대해 사전 선택된 클래스를 표시합니다. 클래스에 지정된 이벤트 및 이에 따라 기록되는 이벤트를 보려면 auditrecord -c *class* 명령을 실행합니다.

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

lo는 login/logout 감사 클래스에 대한 플래그입니다. 마스크 출력의 형식은 (*success*,*failure*)입니다.

```
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
```

```
configured non-attributable audit flags = lo(0x1000,0x1000)
```

예 3-2 기본 감사 정책 표시

```
$ auditconfig -getpolicy
configured audit policies = cnt
active audit policies = cnt
```

active 정책은 현재 정책이지만, 정책 값은 감사 서비스에서 저장하지 않습니다. *configured* 정책은 감사 서비스에서 저장하므로 감사 서비스를 다시 시작하면 정책이 복원됩니다.

예 3-3 기본 감사 플러그인 표시

```
$ auditconfig -getplugin
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=0;p_minfree=1;

Plugin: audit_syslog (inactive)
Attributes: p_flags=;

Plugin: audit_remote (inactive)
Attributes: p_hosts=;p_retries=3;p_timeout=5;
```

`audit_binfile` 플러그인은 기본적으로 활성화됩니다.

예 3-4 감사 대기열 제어 표시

```
$ auditconfig -getqctrl
no configured audit queue hiwater mark
no configured audit queue lowater mark
no configured audit queue buffer size
no configured audit queue delay
active audit queue hiwater mark (records) = 100
active audit queue lowater mark (records) = 10
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

active 대기열 제어는 커널에서 현재 사용하고 있는 대기열 제어입니다. `no configured` 문자열은 시스템에서 기본값을 사용 중임을 나타냅니다.

감사 서비스를 사용/사용 안함으로 설정

감사 서비스는 기본적으로 사용으로 설정됩니다. `perzone` 감사 정책이 설정된 경우 영역 관리자가 필요에 따라 각 비전역 영역에서 감사 서비스를 사용/사용 안함으로 설정하거나 새로 고쳐야 합니다. `perzone` 감사 정책이 설정되지 않은 경우 전역 영역에서 감사 서비스를 사용/사용 안함으로 설정하거나 새로 고치면 모든 비전역 영역에 영향을 줍니다.

감사 서비스를 사용 또는 사용 안함으로 설정하려면 Audit Control 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

감사 서비스를 사용 안함으로 설정하려면 다음 명령을 사용합니다.

```
# audit -t
```

감사 서비스를 사용으로 설정하려면 다음 명령을 사용합니다.

```
# audit -s
```

감사 서비스가 실행 중인지 확인하려면 다음 명령을 사용합니다.

```
# auditconfig -getcond
audit condition = auditing
```

perzone 감사 정책이 설정된 경우, 감사를 사용으로 설정한 비전역 영역에서 이 확인을 수행해야 합니다.

자세한 내용은 audit(1M) 및 auditd(1M) 매뉴얼 페이지를 참조하십시오.

감사 서비스 구성

네트워크에서 감사를 사용으로 설정하기 전에 해당 사이트 감사 요구 사항을 충족하도록 기본값을 수정할 수 있습니다. 가장 좋은 방법은 처음 사용자가 로그인하기 전에 가능한 많이 감사 구성을 사용자 정의하는 것입니다.

영역을 구현한 경우 전역 영역에서 모든 영역을 감사하거나 비전역 영역을 개별적으로 감사하도록 선택할 수 있습니다. 개요는 “감사 및 Oracle Solaris 영역” [110]을 참조하십시오. 계획은 “영역에서 감사 계획” [26]을 참조하십시오. 절차는 “영역에서 감사 서비스 구성” [62]을 참조하십시오.

감사 서비스를 구성하려면 일반적으로 auditconfig 하위 명령을 사용합니다. 이러한 하위 명령으로 설정된 구성은 전체 시스템에 적용됩니다.

- auditconfig -get*은 “감사 서비스 기본값 표시” [38] 예제에 표시된 것처럼 별표(*)로 표시된 매개변수의 현재 구성을 보여줍니다.
- auditconfig -set*는 -setflags, -setpolicy 또는 -setqctrl과 같이 별표(*)로 표시된 매개변수에 값을 지정합니다. 지정 불가능한 이벤트에 대한 클래스를 구성하려면 auditconfig setnaflags 하위 명령을 사용합니다.

또한 전체 시스템이 아닌 사용자 또는 프로파일에 적용하도록 감사를 사용자 정의할 수도 있습니다. 각 사용자에 대한 감사 클래스 사전 선택은 audit_flags 보안 속성으로 지정됩니다. “프로세스 감사 특성” [114]에 설명된 대로 이러한 사용자 특정 값은 시스템에 대해 사전 선택된 클래스와 함께 사용자의 감사 마스크를 결정합니다.

시스템별 기준이 아닌 사용자별 기준으로 클래스를 사전 선택하면 시스템 성능에 대한 감사의 영향을 줄일 수 있는 경우가 있습니다. 또한 시스템과 약간 다르게 특정 사용자를 감사할 수도 있습니다.

사용자 또는 프로파일에 적용되는 감사를 구성하려면 다음 명령을 사용합니다.

- `usrattr`은 사용자에 대해 설정된 `audit_flags` 값을 표시합니다. 기본적으로 사용자는 시스템 전역 설정에 대해서만 감사됩니다.
- `usermod -K`는 사용자에게 적용되는 플래그를 설정합니다.
- `profile`은 프로파일에 적용되는 플래그를 설정합니다.

`userattr` 명령에 대한 설명은 `userattr`(1) 매뉴얼 페이지를 참조하십시오. `audit_flags` 키워드에 대한 설명은 `user_attr`(4) 매뉴얼 페이지를 참조하십시오.

다음 작업 맵에서는 감사 구성을 위한 절차를 안내합니다. 모든 작업은 선택 사항입니다.

표 3-1 감사 서비스 구성 작업 맵

작업	설명	수행 방법
감사되는 이벤트를 선택합니다.	시스템 전역 감사 클래스를 사전 선택합니다. 이벤트가 지정 가능한 경우 모든 사용자가 이 이벤트에 대해 감사됩니다.	감사 클래스를 사전 선택하는 방법 [41]
특정 사용자에 대해 감사되는 이벤트를 선택합니다.	시스템 전역 감사 클래스에서 사용자별 차이점을 설정합니다.	사용자의 감사 특성을 구성하는 방법 [42]
감사 정책을 지정합니다.	사이트에서 요구하는 추가 감사 데이터를 정의합니다.	감사 정책을 변경하는 방법 [46]
대기열 제어를 지정합니다.	기본 버퍼 크기, 대기열의 감사 레코드 및 버퍼에 감사 레코드 쓰기 간격을 수정합니다.	감사 대기열 제어를 변경하는 방법 [49]
`audit_warn` 전자 메일 별칭을 만듭니다.	감사 서비스에 주의가 필요할 때 전자 메일 경고를 받는 사람을 정의합니다.	`audit_warn` 전자 메일 별칭을 구성하는 방법 [50]
감사 로그를 구성합니다.	각 플러그인에 대한 감사 레코드의 위치를 구성합니다.	"감사 로그 구성" [69]
감사 클래스를 추가합니다.	중요 이벤트를 포함할 새 감사 클래스를 만들어 감사 레코드의 수를 줄입니다.	감사 클래스를 추가하는 방법 [51]
이벤트-클래스 매핑을 변경합니다.	이벤트-클래스 매핑을 변경하여 감사 레코드의 수를 줄입니다.	감사 이벤트의 클래스 멤버쉽을 변경하는 방법 [52]

▼ 감사 클래스를 사전 선택하는 방법

모니터할 이벤트를 포함하는 감사 클래스를 사전 선택합니다. 사전 선택된 클래스에 없는 이벤트는 기록되지 않습니다.

시작하기 전에 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **현재 사전 선택된 클래스를 결정합니다.**

```
# auditconfig -getflags
...

# auditconfig -getnaflags
,,,
```

출력에 대한 설명은 "감사 서비스 기본값 표시" [38]를 참조하십시오.

2. **지정 가능한 클래스를 사전 선택합니다.**

```
# auditconfig -setflags lo,ps,fw
user default audit flags = ps,lo,fw(0x101002,0x101002)
```

이 명령은 `login/logout`, `process start/stop` 및 `file write` 클래스에서 이벤트의 성공 및 실패에 대해 감사합니다.

참고 - `auditconfig -setflags` 명령은 현재 사전 선택을 바꾸므로 사전 선택할 모든 클래스를 지정해야 합니다.

---

3. **지정 불가능한 클래스를 사전 선택합니다.**

`na` 클래스에는 대표적으로 PROM, 부트 및 지정 불가능한 마운트 이벤트가 포함됩니다.

```
# auditconfig -setnaflags lo,na
non-attributable audit flags = lo,na(0x1400,0x1400)
```

`lo` 및 `na`는 `-setnaflags` 옵션에 유일하게 유용한 인수입니다.

참고 - `auditconfig -setnaflags` 명령은 현재 사전 선택을 바꾸므로 사전 선택할 모든 클래스를 지정해야 합니다.

▼ 사용자의 감사 특성을 구성하는 방법

이 절차에서 설정하는 이러한 사용자별 감사 특성은 시스템에 대해 사전 선택된 클래스와 결합됩니다. "프로세스 감사 특성" [114]에 설명된 대로 이러한 특성을 통해 사용자의 감사 마스크를 확인할 수 있습니다.

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안"의 "지정된 관리 권한 사용"을 참조하십시오.

1. **(옵션) 기존 사용자에 대해 현재 선택된 감사 클래스를 표시합니다.**

   a. **사용자 목록을 표시합니다.**

```
# who
```

```
adoe    pts/1        Oct 10 10:20 (:0.0)
adoe    pts/2        Oct 10 10:20 (:0.0)
jdoe    pts/5        Oct 12 12:20 (:0.0)
jdoe    pts/6        Oct 12 12:20 (:0.0)
...
```

b. **각 사용자의 `audit_flags` 속성값을 표시합니다.**

```
# userattr audit_flags adoe
# userattr audit_flags jdoe
```

2. **`user_attr` 또는 `prof_attr` 데이터베이스에서 감사 플래그를 설정합니다.**

예를 들어, 사용자의 하위 세트에 대한 권한을 정의하는 권한 프로파일을 만들 수 있습니다. 해당 권한 프로파일이 지정된 사용자는 동일하게 감사됩니다.

- **사용자에 대한 감사 플래그를 설정하려면 `usermod` 명령을 사용합니다.**

```
# usermod -K audit_flags=fw:no jdoe
```

`audit_flags` 키워드는 *always-audit*:*never-audit*입니다.

always-audit	이 사용자에 대해 감사되는 감사 클래스를 나열합니다. 시스템 전역 클래스에 대한 수정 앞에는 캐럿(^)이 붙습니다. 시스템 전역 클래스에 추가된 클래스 앞에는 캐럿이 붙지 않습니다.
never-audit	이러한 감사 이벤트가 시스템 전역으로 감사되더라도 사용자에 대해 감사되지 않는 감사 클래스를 나열합니다. 시스템 전역 클래스에 대한 수정 앞에는 캐럿(^)이 붙습니다.

여러 감사 클래스를 지정하려면 클래스를 콤마로 구분합니다. 자세한 내용은 `audit_flags`(5) 매뉴얼 페이지를 참조하십시오.

- **권한 프로파일에 대한 감사 플래그를 설정하려면 `profiles` 명령을 사용합니다.**

```
# profiles -p "System Administrator"
profiles:System Administrator> set name="Audited System Administrator"
profiles:Audited System Administrator> set always_audit=fw,as
profiles:Audited System Administrator> end
profiles:Audited System Administrator> exit
```

Audited System Administrator 권한 프로파일을 사용자나 역할에 지정할 경우 해당 사용자나 역할은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 권한 검색 순서"에 설명된 검색 순서에 따라 이러한 플래그에 대해 감사됩니다.

예 3-5 한 사용자에 대해 감사되는 이벤트 변경

이 예에서는 모든 사용자에 대한 감사 사전 선택 마스크가 다음과 같습니다.

```
# auditconfig -getflags
```

```
active user default audit flags = ss,lo(0x11000,0x11000)
configured user default audit flags = ss,lo(0x11000,0x11000)
```

관리자를 제외하고 로그인된 사용자가 없습니다.

시스템 리소스에 대한 AUE_PFEXEC 감사 이벤트의 영향을 줄이기 위해 관리자는 이 이벤트를 시스템 레벨에서 감사하지 않습니다. 대신 관리자는 사용자 jdoe에 대해 pf 클래스를 사전 선택합니다. pf 클래스는 예 3-15. “새 감사 클래스 만들기”에서 만들어졌습니다.

```
# usermod -K audit_flags=pf:no jdoe
```

userattr 명령은 추가를 표시합니다.

```
# userattr audit_flags jdoe
pf:no
```

사용자 jdoe가 로그인할 때 jdoe의 감사 사전 선택 마스크는 시스템 기본값과 audit_flags 값의 조합입니다. 289는 jdoe 로그인 셸의 PID입니다.

```
# auditconfig -getpinfo 289
audit id = jdoe(1234)
process preselection mask = ss,pf,lo(0x0100000008011000,0x0100000008011000)
terminal id (maj,min,host) = 242,511,example1(192.168.160.171)
audit session id = 103203403
```

예 3-6 한 사용자에 대한 감사 사전 선택 예외 사항 수정

이 예에서는 모든 사용자에 대한 감사 사전 선택 마스크가 다음과 같습니다.

```
# auditconfig -getflags
active user default audit flags = ss,lo(0x11000,0x11000)
configured user default audit flags = ss,lo(0x11000,0x11000)
```

관리자를 제외하고 로그인된 사용자가 없습니다.

관리자는 jdoe 사용자에 대해 실패한 ss 이벤트를 수집하지 않도록 결정합니다.

```
# usermod -K audit_flags=^-ss:no jdoe
```

userattr 명령은 예외 사항을 표시합니다.

```
# userattr audit_flags jdoe
^-ss:no
```

사용자 jdoe가 로그인할 때 jdoe의 감사 사전 선택 마스크는 시스템 기본값과 audit_flags 값의 조합입니다. 289는 jdoe 로그인 셸의 PID입니다.

```
# auditconfig -getpinfo 289
audit id = jdoe(1234)
process preselection mask = +ss,lo(0x11000,0x1000)
terminal id (maj,min,host) = 242,511,example1(192.168.160.171)
```

```
audit session id = 103203403
```

예 3-7 선택한 사용자 감사, 시스템 전역 감사 없음

이 예에서는 4명의 선택된 사용자에 대한 로그인 및 역할 작업이 시스템에서 감사됩니다. 시스템에 대해 사전 선택된 감사 클래스는 없습니다.

먼저 관리자는 모든 시스템 전역 플래그를 제거합니다.

```
# auditconfig -setflags no
user default audit flags = no(0x0,0x0)
```

그런 다음 관리자는 4명의 사용자에 대해 2개의 감사 클래스를 사전 선택합니다. `pf` 클래스는 예 3-15. “새 감사 클래스 만들기”에서 만들어졌습니다.

```
# usermod -K audit_flags=lo,pf:no jdoe
# usermod -K audit_flags=lo,pf:no kdoe
# usermod -K audit_flags=lo,pf:no pdoe
# usermod -K audit_flags=lo,pf:no zdoe
```

그런 다음 관리자는 `root` 역할에 대해 `pf` 클래스를 사전 선택합니다.

```
# userattr audit_flags root
# rolemod -K audit_flags=lo,pf:no root
# userattr audit_flags root
lo,pf:no
```

무단 침입 기록을 계속하기 위해 관리자는 지정 불가능한 로그인의 감사를 변경하지 않습니다.

```
# auditconfig -getnaflags
active non-attributable audit flags = lo(0x1000,0x1000)
configured non-attributable audit flags = lo(0x1000,0x1000)
```

예 3-8 사용자의 감사 플래그 제거

다음 예에서는 관리자가 모든 사용자별 감사 플래그를 제거합니다. 현재 로그인된 사용자의 기존 프로세스는 계속 감사됩니다.

관리자는 `audit_flags` 키워드를 `no` 값으로 설정하여 `usermod` 명령을 실행합니다.

```
# usermod -K audit_flags= jdoe
# usermod -K audit_flags= kdoe
# usermod -K audit_flags= ldoe
```

그런 다음 관리자는 제거를 확인합니다.

```
# userattr audit_flags jdoe
# userattr audit_flags kdoe
# userattr audit_flags ldoe
```

예 3-9 사용자 그룹에 대한 권한 프로파일 만들기

관리자는 사이트의 모든 관리 권한 프로파일이 `pf` 클래스를 명시적으로 감사하도록 하고자 합니다. 지정할 모든 권한 프로파일에 대해 관리자는 감사 플래그가 포함된 LDAP의 사이트별 버전을 만듭니다.

먼저, 관리자는 기존 권한 프로파일을 복제한 다음 이름을 변경하고 감사 플래그를 추가합니다.

```
# profiles -p "Network Wifi Management" -S ldap
profiles: Network Wifi Management> set name="Wifi Management"
profiles: Wifi Management> set desc="Audited wifi management"
profiles: Wifi Management> set audit_always=pf
profiles: Wifi Management> exit
```

사용할 모든 권한 프로파일에 대해 이 절차를 반복한 후 관리자는 Wifi Management 프로파일의 정보를 나열합니다.

```
# profiles -p "Wifi Management" -S ldap info
name=Wifi Management
desc=Audited wifi management
auths=solaris.network.wifi.config
help=RtNetWifiMngmnt.html
always_audit=pf
```

▼ 감사 정책을 변경하는 방법

기본 감사 정책을 변경하여 감사된 명령에 대한 자세한 정보를 기록하거나 영역 이름을 모든 레코드에 추가하거나 기타 사이트 보안 요구 사항을 충족할 수 있습니다.

시작하기 전에 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

1. **현재 감사 정책을 봅니다.**

   ```
   $ auditconfig -getpolicy
   ...
   ```

   출력에 대한 설명은 “감사 서비스 기본값 표시” [38]를 참조하십시오.

2. **사용 가능한 감사 옵션을 봅니다.**

   ```
   $ auditconfig -lspolicy
   policy string    description:
   ahlt             halt machine if it can not record an async event
   all              all policies for the zone
   arge             include exec environment args in audit recs
   ```

```
argv            include exec command line args in audit recs
cnt             when no more space, drop recs and keep a cnt
group           include supplementary groups in audit recs
none            no policies
path            allow multiple paths per event
perzone         use a separate queue and auditd per zone
public          audit public files
seq             include a sequence number in audit recs
trail           include trailer token in audit recs
windata_down    include downgraded window information in audit recs
windata_up      include upgraded window information in audit recs
zonename        include zonename token in audit recs
```

참고 - `perzone` 및 `ahlt` 정책 옵션은 전역 영역에서만 설정할 수 있습니다. 특정 정책 옵션을 사용하는 장단점은 "감사 정책 이해" [32]를 참조하십시오.

3. **선택한 감사 정책 옵션을 사용 또는 사용 안함으로 설정합니다.**

```
# auditconfig [ -t ] -setpolicy [prefix]policy[,policy...]
```

`-t` 선택 사항. 임시(또는 활성) 정책을 만듭니다. 디버깅 또는 테스트 목적으로 임시 정책을 설정할 수 있습니다.

임시 정책은 감사 서비스가 새로 고쳐질 때까지 또는 정책이 `auditconfig -setpolicy` 명령으로 수정될 때까지 유효합니다.

prefix +의 *prefix* 값은 정책 목록을 현재 정책에 추가합니다. -의 *prefix* 값은 정책 목록을 현재 정책에서 제거합니다. 접두어가 없으면 감사 정책이 재설정됩니다. 이 옵션을 사용하여 현재 감사 정책을 유지할 수 있습니다.

policy 사용으로 설정하거나 사용 안함으로 설정할 정책을 선택합니다.

예 3-10 `ahlt` 감사 정책 옵션

이 예에서 엄격한 사이트 보안을 위해서는 `ahlt` 정책이 필요합니다.

```
# auditconfig -setpolicy -cnt
# auditconfig -setpolicy +ahlt
```

`ahlt` 정책 앞의 더하기 기호(+)는 정책을 현재 정책 설정에 추가합니다. 더하기 기호가 없으면 `ahlt` 정책이 현재 모든 감사 정책을 바꿉니다.

예 3-11 임시 감사 정책 설정

이 예에서는 `ahlt` 감사 정책이 구성되었습니다. 디버깅을 위해 관리자는 `trail` 감사 정책을 활성 정책(`+trail`)에 임시로(`-t`) 추가합니다. `trail` 정책은 손상된 감사 추적을 복구하는 데 유용합니다.

```
$ auditconfig -setpolicy ahlt
$ auditconfig -getpolicy
configured audit policies = ahlt
active audit policies = ahlt
$ auditconfig -t -setpolicy +trail
configured audit policies = ahlt
active audit policies = ahlt,trail
```

디버깅이 완료되면 관리자는 `trail` 정책을 사용 안함으로 설정합니다.

```
$ auditconfig -setpolicy -trail
$ auditconfig -getpolicy
configured audit policies = ahlt
active audit policies = ahlt
```

`audit -s` 명령을 실행하여 감사 서비스를 새로 고쳐도 감사 서비스의 다른 임시 값과 함께 이 임시 정책이 제거됩니다. 다른 임시 값의 예는 감사 대기열 제어를 변경하는 방법 [49]을 참조하십시오.

예 3-12 `perzone` 감사 정책 설정

이 예에서는 `perzone` 감사 정책이 전역 영역의 기존 정책에 추가됩니다. `perzone` 정책 설정은 영구 등록 정보로 저장되므로 `perzone` 정책은 세션 동안 및 감사 서비스가 다시 시작되어도 유효합니다. 해당 영역에서 정책은 다음에 영역을 부트할 때 사용할 수 있습니다.

```
$ auditconfig -getpolicy
configured audit policies = cnt
active audit policies = cnt
$ auditconfig -setpolicy +perzone
$ auditconfig -getpolicy
configured audit policies = perzone,cnt
active audit policies = perzone,cnt
```

예 3-13 외부 감사자에 대한 감사 레코드 수집

이 예제에서 관리자는 외부 감사자의 요구 사항을 충족시키기 위한 감사 레코드를 수집합니다. 관리자는 ARS(감사 원격 서버)를 사용해서 관리 작업에 대한 정보를 수집하도록 결정합니다. 또한 관리자는 부트와 같이 사용자에게 지정할 수 없는 작업도 수집합니다.

관리자가 ARS를 설정합니다. `cusa` 클래스 감사 외에도 관리자는 감사 구성에 정책을 추가합니다.

```
# auditconfig -setflags cusa
user default audit flags = ex,xa,ua,as,ss,ap,lo,ft(0x80475080,0x80475080)
# auditconfig -setpolicy ahlt,argv,argeauditconfig # auditconfig -getpolicy
configured audit policies = ahlt,arge,argv
active audit policies = ahlt,arge,argv
# auditconfig -setnaflags lo,na
non-attributable audit flags = lo,na(0x1400,0x1400)
```

관리자가 audit_remote 플러그인을 사용으로 설정하고 감사 서비스를 새로 고치면 레코드가 수집됩니다.

▼ 감사 대기열 제어를 변경하는 방법

감사 서비스는 감사 대기열 매개변수에 대한 기본값을 제공합니다. auditconfig 명령을 사용하여 이러한 값을 검사, 변경 및 영구적으로 변경할 수 있습니다.

시작하기 전에 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **감사 대기열 제어의 현재 값을 봅니다.**

```
$ auditconfig -getqctrl
...
```

출력에 대한 설명은 "감사 서비스 기본값 표시" [38]를 참조하십시오.

2. **선택한 감사 대기열 제어를 수정합니다.**

감사 대기열 제어의 예 및 설명은 auditconfig(1M) 매뉴얼 페이지를 참조하십시오.

- 일부 또는 모든 감사 대기열 제어를 수정하려면 -setqctrl 옵션을 사용합니다.

```
# auditconfig [ -t ] -setqctrl hiwater lowater bufsz interval
```

고수위(hiwater) 및 저수위(lowater) 값은 프로세스가 각각 일시 중지되고 재개되는 지점을 나타냅니다. 각 지점은 배달되지 않은 감사 레코드 수로 측정됩니다. 버퍼 크기(bufsz)는 대기열의 버퍼 크기를 나타냅니다. Interval은 감사 출력 생성 사이의 지연 시간을 나타내며, 틱 수로 측정됩니다.

예를 들어, 다른 제어를 설정하지 않고 *interval* 값을 10으로 설정합니다.

```
# auditconfig -setqctrl 0 0 0 10
```

- 특정 감사 대기열 제어를 수정하려면 해당 옵션을 지정합니다. **auditconfig -setqdelay 10**에서와 같이 -setqdelay 옵션은 -setqctrl 0 0 0 *interval*과 동일합니다.

```
# auditconfig [ -t ] -setqhiwater value
# auditconfig [ -t ] -setqlowater value
# auditconfig [ -t ] -setqbufsz value
# auditconfig [ -t ] -setqdelay value
```

예 3-14 감사 대기열 제어를 기본값으로 재설정

관리자는 모든 감사 대기열 제어를 설정한 다음 저장소의 *lowater* 값을 기본값으로 다시 변경합니다.

```
# auditconfig -setqctrl 200 5 10216 10
# auditconfig -setqctrl 200 0 10216 10
configured audit queue hiwater mark (records) = 200
no configured audit queue lowater mark
configured audit queue buffer size (bytes) = 10216
configured audit queue delay (ticks) = 10
active audit queue hiwater mark (records) = 200
active audit queue lowater mark (records) = 5
active audit queue buffer size (bytes) = 10216
active audit queue delay (ticks) = 10
```

나중에 관리자는 *lowater* 값을 현재 세션에 대한 기본값으로 설정합니다.

```
# auditconfig -setqlowater 10
# auditconfig -getqlowater
configured audit queue lowater mark (records) = 10
active audit queue lowater mark (records) = 10
```

▼ audit_warn 전자 메일 별칭을 구성하는 방법

/etc/security/audit_warn 스크립트는 관리자에게 주의가 필요할 수 있는 감사 이벤트를 알려주는 메일을 생성합니다. 이 스크립트를 사용자 정의하고 root 이외의 계정에 메일을 보낼 수 있습니다.

perzone 정책이 설정된 경우 비전역 관리자는 비전역 영역에서 audit_warn 전자 메일 별칭을 구성해야 합니다.

시작하기 전에 사용자는 solaris.admin.edit/etc/security/audit_warn 권한 부여가 지정된 관리자여야 합니다. 기본적으로 root 역할에만 이 권한 부여가 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- **audit_warn 전자 메일 별칭을 구성합니다.**

  다음 옵션 중 하나를 선택합니다.

  - audit_warn 스크립트에서 audit_warn 전자 메일 별칭을 다른 전자 메일 계정으로 바꿉니다.

    스크립트의 ADDRESS 행에서 audit_warn 전자 메일 별칭을 다른 주소로 변경합니다.

    ```
    #ADDRESS=audit_warn            # standard alias for audit alerts
    ADDRESS=audadmin               # role alias for audit alerts
    ```

참고 - 감사 구성 파일 수정의 영향에 대한 자세한 내용은 "감사 구성 파일 및 패키징" [110]을 참조하십시오.

  - audit_warn 전자 메일을 다른 메일 계정으로 재지정합니다.

audit_warn 전자 메일 별칭을 적합한 메일 별칭 파일에 추가합니다. 로컬 /etc/mail/aliases 파일이나 이름 공간의 mail_aliases 데이터베이스에 별칭을 추가할 수 있습니다. root 및 audadmin 전자 메일 계정이 audit_warn 전자 메일 별칭의 구성원으로 추가된 경우 /etc/mail/aliases 항목은 다음 예와 유사합니다.

```
audit_warn: root,audadmin
```

그런 다음 newaliases 명령을 실행하여 aliases 파일에 대한 모든 액세스 데이터베이스를 재구축합니다.

```
# newaliases
/etc/mail/aliases: 14 aliases, longest 10 bytes, 156 bytes total
```

▼ 감사 클래스를 추가하는 방법

고유의 감사 클래스를 만들 때 해당 사이트에 대해 감사하고자 하는 감사 이벤트를 추가하면 됩니다. 이 전략은 수집되는 레코드 수를 줄이고 감사 추적의 노이즈를 줄일 수 있습니다.

한 시스템에서 클래스를 추가하는 경우 감사하는 모든 시스템에 변경 사항을 복사합니다. 가장 좋은 방법은 첫번째 사용자가 로그인하기 전에 감사 클래스를 만드는 것입니다.

감사 구성 파일 수정의 영향에 대한 자세한 내용은 "감사 구성 파일 및 패키징" [110]을 참조하십시오.

작은 정보 - Oracle Solaris에서는 파일이 포함된 고유한 패키지를 만들고 사이트에서 사용자 정의한 파일로 Oracle Solaris 패키지를 바꿀 수 있습니다. 패키지에서 preserve 속성을 true로 설정하면 pkg 하위 명령(예: verify, fix, revert 등)이 사용자의 패키지를 기반으로 실행됩니다. 자세한 내용은 pkg(1) 및 pkg(5) 매뉴얼 페이지를 참조하십시오.

시작하기 전에 고유 항목에 대한 여유 비트를 선택합니다. 고객이 사용 가능한 비트를 /etc/security/audit_class 파일에서 확인합니다.

사용자는 solaris.admin.edit/etc/security/audit_class 권한 부여가 지정된 관리자여야 합니다. 기본적으로 root 역할에만 이 권한 부여가 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **(옵션) audit_class 파일의 백업 복사본을 저장합니다.**

   ```
   # cp /etc/security/audit_class /etc/security/audit_class.orig
   ```

2. **새 항목을 audit_class 파일에 추가합니다.**

   각 항목의 형식은 다음과 같습니다.

   0x64bitnumber:flag:description

필드에 대한 설명은 audit_class(4) 매뉴얼 페이지를 참조하십시오. 기존 클래스 목록은 /etc/security/audit_class 파일을 참조하십시오.

예 3-15 새 감사 클래스 만들기

이 예에서는 역할에서 실행되는 관리 명령을 포함하는 클래스를 만듭니다. audit_class 파일에 추가된 항목은 다음과 같습니다.

```
0x0100000000000000:pf:profile command
```

항목은 새 pf 감사 클래스를 만듭니다. 예 3-16. "기존 감사 이벤트를 새 클래스에 매핑"은 새로운 감사 클래스를 채우는 방법을 보여줍니다.

일반 오류 audit_class 파일을 사용자 정의한 경우 사용자 또는 권한 프로파일에 직접 지정된 감사 플래그가 새 감사 클래스와 일관성이 있는지 확인합니다. audit_flags 값이 audit_class 파일의 일부가 아닌 경우 오류가 발생합니다.

▼ 감사 이벤트의 클래스 멤버쉽을 변경하는 방법

감사 이벤트의 클래스 멤버쉽을 변경하여 기존 감사 클래스의 크기를 줄이거나 이벤트를 고유의 클래스에 추가할 수 있습니다.

주의 - audit_event 파일에서 이벤트를 주석 처리하지 마십시오. 이 파일은 praudit 명령에서 이진 감사 파일을 읽는 데 사용됩니다. 아카이브된 감사 파일은 파일에 나열된 이벤트를 포함할 수 있습니다.

한 시스템에서 감사 이벤트-클래스 매핑을 재구성하는 경우 감사하는 모든 시스템에 변경 사항을 복사합니다. 가장 좋은 방법은 첫번째 사용자가 로그인하기 전에 이벤트-클래스 매핑을 변경하는 것입니다.

참고 - 감사 구성 파일 수정의 영향에 대한 자세한 내용은 "감사 구성 파일 및 패키징" [110]을 참조하십시오.

작은 정보 - Oracle Solaris에서는 파일이 포함된 고유한 패키지를 만들고 사이트에서 사용자 정의한 파일로 Oracle Solaris 패키지를 바꿀 수 있습니다. 패키지에서 preserve 속성을 true로 설정하면 pkg 하위 명령(예: verify, fix, revert 등)이 사용자의 패키지를 기반으로 실행됩니다. 자세한 내용은 pkg(1) 및 pkg(5) 매뉴얼 페이지를 참조하십시오.

시작하기 전에 사용자는 solaris.admin.edit/etc/security/audit_event 권한 부여가 지정된 관리자여야 합니다. 기본적으로 root 역할에만 이 권한 부여가 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **(옵션) `audit_event` 파일의 백업 복사본을 저장합니다.**

```
# cp /etc/security/audit_event /etc/security/audit_event.orig
```

2. **이벤트의 *class-list*를 변경하여 특정 이벤트가 속한 클래스를 변경합니다.**

각 항목의 형식은 다음과 같습니다.

number:*name*:*description*:*class-list*

number	감사 이벤트 ID입니다.
name	감사 이벤트의 이름입니다.
description	일반적으로 감사 레코드 만들기를 트리거하는 시스템 호출 또는 실행 파일입니다.
class-list	쉼표로 구분된 감사 클래스 목록입니다.

예 3-16 기존 감사 이벤트를 새 클래스에 매핑

이 예에서는 기존 감사 이벤트를 예 3-15. “새 감사 클래스 만들기”에서 만들어진 새 클래스에 매핑합니다. 기본적으로 `AUE_PFEXEC` 감사 이벤트는 여러 감사 클래스에 매핑됩니다. 새 클래스를 만들면 관리자는 다른 클래스의 이벤트를 감사하지 않고 `AUE_PFEXEC` 이벤트를 감사할 수 있습니다.

```
# grep pf /etc/security/audit_class
0x0100000000000000:pf:profile command
# grep AUE_PFEXEC /etc/security/audit_event
116:AUE_PFEXEC:execve(2) with pfexec enabled:ps,ex,ua,as,cusa
# pfedit /etc/security/audit_event
#116:AUE_PFEXEC:execve(2) with pfexec enabled:ps,ex,ua,as,cusa
116:AUE_PFEXEC:execve(2) with pfexec enabled:pf
# auditconfig -setflags lo,pf
user default audit flags = pf,lo(0x0100000000001000,0x0100000000001000)
```

감사 대상 사용자 정의

다음 작업 맵에서는 사용자 요구와 관련된 감사를 구성하는 절차를 안내합니다.

표 3-2 감사 사용자 정의 작업 맵

작업	설명	수행 방법
사용자가 시스템에서 수행하는 모든 작업을 감사합니다.	모든 명령에 대해 하나 이상의 사용자를 감사합니다.	사용자의 모든 명령을 감사하는 방법 [54]

작업	설명	수행 방법
기록되는 감사 이벤트를 변경하고 변경 사항을 기존 세션에 적용합니다.	사용자의 사전 선택 마스크를 업데이트합니다.	로그인한 사용자의 사전 선택 마스크를 업데이트하는 방법 [57]
특정 파일에 대한 수정 사항을 찾습니다.	파일 수정을 감사한 다음 `auditreduce` 명령을 사용하여 특정 파일을 찾습니다.	특정 파일에 대한 변경 사항 감사 레코드를 찾는 방법 [56]
감사 파일에 대해 파일 시스템 공간을 덜 사용합니다.	ZFS 할당량 및 압축을 사용합니다.	전용 파일 시스템에서 감사 파일을 압축하는 방법 [60]
`audit_event` 파일에서 감사 이벤트를 제거합니다.	`audit_event` 파일을 올바르게 업데이트합니다.	특정 이벤트의 감사를 막는 방법 [59]

▼ 사용자의 모든 명령을 감사하는 방법

보안 정책의 일부로 일부 사이트에서는 `root` 계정 및 관리 역할에서 실행하는 모든 명령의 감사 레코드를 요구합니다. 일부 사이트에서는 모든 사용자가 실행하는 모든 명령에 대한 감사 레코드를 요구할 수 있습니다. 또한 사이트에서는 명령 인수 및 환경이 기록되도록 요구할 수 있습니다.

시작하기 전에 감사 클래스를 사전 선택하고 감사 정책을 설정하려면 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 감사 플래그를 사용자, 역할, 권한 프로파일에 지정하려면 `root` 역할을 맡아야 합니다.

1. **`lo` 및 `ex` 클래스에 대한 사용자 레벨 이벤트 정보를 표시합니다.**

   `ex` 클래스는 `exec()` 및 `execve()` 함수에 대한 모든 호출을 감사합니다.

   `lo` 클래스는 로그인, 로그아웃 및 화면 잠금을 감사합니다. 다음 출력은 `ex` 및 `lo` 클래스의 모든 이벤트를 나열합니다.

```
% auditconfig -lsevent | grep " lo "
AUE_login                       6152 lo login - local
AUE_logout                      6153 lo logout
AUE_telnet                      6154 lo login - telnet
AUE_rlogin                      6155 lo login - rlogin
AUE_rshd                        6158 lo rsh access
AUE_su                          6159 lo su
AUE_rexecd                      6162 lo rexecd
AUE_passwd                      6163 lo passwd
AUE_rexd                        6164 lo rexd
AUE_ftpd                        6165 lo ftp access
AUE_ftpd_logout                 6171 lo ftp logout
AUE_ssh                         6172 lo login - ssh
AUE_role_login                  6173 lo role login
AUE_newgrp_login                6212 lo newgrp login
AUE_admin_authenticate          6213 lo admin login
AUE_screenlock                  6221 lo screenlock - lock
AUE_screenunlock                6222 lo screenlock - unlock
AUE_zlogin                      6227 lo login - zlogin
```

```
AUE_su_logout                   6228 lo su logout
AUE_role_logout                 6229 lo role logout
AUE_smbd_session                6244 lo smbd(1m) session setup
AUE_smbd_logoff                 6245 lo smbd(1m) session logoff
AUE_ClientConnect               9101 lo client connection to x server
AUE_ClientDisconnect            9102 lo client disconn. from x server

% auditconfig -lsevent | egrep " ex |,ex |ex,"
AUE_EXECVE                       23 ex,ps execve(2)
```

2. **lo 및 ex 클래스를 감사합니다.**

   - **관리 역할에 대해 이러한 클래스를 감사하려면 역할의 보안 속성을 수정합니다.**

     다음 예에서 root는 역할입니다. 사이트에서는 sysadm, auditadm 및 netadm의 세 역할을 만들었습니다. 모든 역할은 ex 및 lo 클래스에 있는 이벤트의 성공 및 실패에 대해 감사됩니다.

     ```
     # rolemod -K audit_flags=lo,ex:no root

     # rolemod -K audit_flags=lo,ex:no sysadm

     # rolemod -K audit_flags=lo,ex:no auditadm

     # rolemod -K audit_flags=lo,ex:no netadm
     ```

   - **모든 사용자에 대해 이러한 클래스를 감사하려면 시스템 전역 플래그를 설정합니다.**

     ```
     # auditconfig -setflags lo,ex
     ```

     출력은 다음과 유사하게 나타납니다.

     ```
     header,129,2,AUE_EXECVE,,mach1,2010-10-14 12:17:12.616 -07:00
     path,/usr/bin/ls
     attribute,100555,root,bin,21,320271,18446744073709551615
     subject,jdoe,root,root,root,root,2486,50036632,82 0 mach1
     return,success,0
     ```

3. **명령 사용에 대해 기록할 추가 정보를 지정합니다.**

   - **명령에 대한 인수를 기록하려면 argv 정책을 추가합니다.**

     ```
     # auditconfig -setpolicy +argv
     ```

     exec_args 토큰은 명령 인수를 기록합니다.

     ```
     header,151,2,AUE_EXECVE,,mach1,2010-10-14 12:26:17.373 -07:00
     path,/usr/bin/ls
     attribute,100555,root,bin,21,320271,18446744073709551615
     exec_args
     ,2,ls,/etc/security
     subject,jdoe,root,root,root,root,2494,50036632,82 0 mach1
     return,success,0
     ```

■ **명령이 실행되는 환경을 기록하려면 `arge` 정책을 추가합니다.**

```
# auditconfig -setpolicy +arge
```

exec_env 토큰은 명령 환경을 기록합니다.

```
header,1460,2,AUE_EXECVE,,mach1,2010-10-14 12:29:39.679 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
exec_args,2,ls,/etc/security
exec_env
,49,MANPATH=/usr/share/man,USER=jdoe,GDM_KEYBOARD_LAYOUT=us,EDITOR=gedit,
LANG=en_US.UTF-8,GDM_LANG=en_US.UTF-8,PS1=#,GDMSESSION=gnome,SESSIONTYPE=1,SHLVL=2,
HOME=/home/jdoe,LOGNAME=jdoe,G_FILENAME_ENCODING=@locale,UTF-8,
PRINTER=example-dbl,...,_=/usr/bin/ls
subject,jdoe,root,root,root,root,2502,50036632,82 0 mach1
return,success,0
```

▼ 특정 파일에 대한 변경 사항 감사 레코드를 찾는 방법

목적이 /etc/passwd 및 /etc/default 디렉토리의 파일과 같이 제한된 수의 파일에 대한 파일 쓰기를 기록하는 것이라면 auditreduce 명령을 사용하여 파일을 찾을 수 있습니다.

시작하기 전에 root 역할은 이 절차의 모든 작업을 수행할 수 있습니다.

관리 권한이 조직에 분산되어 있는 경우 다음을 참조하십시오.

- Audit Configuration 권한 프로파일이 있는 관리자는 auditconfig 명령을 실행할 수 있습니다.
- Audit Review 권한 프로파일이 있는 관리자는 auditreduce 명령을 실행할 수 있습니다.
- root 역할만 감사 플래그를 지정할 수 있습니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

1. **파일 변경 사항을 감사하기 위해 다음 단계 중 하나를 수행합니다.**

   ■ fw 클래스를 감사합니다.

   사용자나 역할의 감사 플래그에 fw 클래스를 추가하면 시스템 전역 감사 사전 선택 마스크에 클래스를 추가할 때보다 적은 레코드가 생성됩니다. 다음 단계 중 하나를 수행합니다.

   ■ fw 클래스를 특정 역할에 추가합니다.

   ```
   # rolemod -K audit_flags=fw:no root
   # rolemod -K audit_flags=fw:no sysadm
   # rolemod -K audit_flags=fw:no auditadm
   ```

```
# rolemod -K audit_flags=fw:no netadm
```

- `fw` 클래스를 시스템 전역 플래그에 추가합니다.

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

# auditconfig -setflags lo,fw
user default audit flags = lo,fw(0x1002,0x1002)
```

- 성공한 파일 쓰기를 감사합니다.

  성공을 감사하면 실패 및 성공을 감사할 때보다 적은 레코드가 생성됩니다. 다음 단계 중 하나를 수행합니다.

  - `+fw` 플래그를 특정 역할에 추가합니다.

```
# rolemod -K audit_flags=+fw:no root
# rolemod -K audit_flags=+fw:no sysadm
# rolemod -K audit_flags=+fw:no auditadm
# rolemod -K audit_flags=+fw:no netadm
```

  - `+fw` 플래그를 시스템 전역 플래그에 추가합니다.

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

# auditconfig -setflags lo,+fw
user default audit flags = lo,+fw(0x1002,0x1000)
```

2. **`auditreduce` 명령으로 특정 파일에 대한 감사 레코드를 가져옵니다.**

```
# auditreduce -o file=/etc/passwd,/etc/default -O filechg
```

`auditreduce` 명령은 `file` 인수의 모든 인스턴스에 대한 감사 추적을 검색합니다. 명령은 관심 파일의 경로가 포함된 모든 레코드를 포함하는 `filechg` 접미어의 이진 파일을 만듭니다. `-o file=` *pathname* 옵션의 구문은 `auditreduce`(1M) 매뉴얼 페이지를 참조하십시오.

3. **`praudit` 명령을 사용해서 `filechg` 파일을 읽습니다.**

```
# praudit *filechg
```

▼ 로그인한 사용자의 사전 선택 마스크를 업데이트하는 방법

이 절차에서는 시스템 전역 감사 사전 선택 마스크의 변경 사항에 대해 이미 로그인한 사용자를 감사하는 방법을 설명합니다. 이 작업은 일반적으로 사용자에게 로그아웃하고 다시 로그

인하도록 지시하여 수행할 수 있습니다. 또는 Process Management 권한 프로파일이 지정된 역할의 경우 `kill` 명령을 사용해서 활성 세션을 수동으로 종료할 수 있습니다. 새 세션은 새로운 사전 선택 마스크를 상속합니다.

하지만 사용자 세션 종료는 적합하지 않을 수 있습니다. 또는 `auditconfig` 명령을 사용해서 각 로그인한 사용자의 사전 선택 마스크를 동적으로 변경할 수 있습니다.

다음 절차에서는 다음 명령을 실행하여 시스템 전역 감사 사전 선택 마스크를 `lo`에서 `lo,ex`로 변경했다고 가정합니다.

```
# auditconfig -setflags lo,ex
```

시작하기 전에 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 사용자 세션을 종료하려면 Process Management 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **로그인한 일반 사용자 및 프로세스 ID를 나열합니다.**

```
# who -a
jdoe  - vt/2         Jan 25 07:56  4:10   1597   (:0)
jdoe  + pts/1        Jan 25 10:10   .     1706   (:0.0)
...
jdoe  + pts/2        Jan 25 11:36  3:41   1706   (:0.0)
```

2. **나중에 비교를 위해 각 사용자의 사전 선택 마스크를 표시합니다.**

```
# auditconfig -getpinfo 1706
audit id = jdoe(1234)
process preselection mask = lo(0x1000,0x1000)
terminal id (maj,min,host) = 9426,65559,mach1(192.168.123.234)
audit session id = 103203403
```

3. **다음 명령 중 하나 이상을 실행하여 적합한 사전 선택 마스크를 수정합니다.**

```
# auditconfig -setpmask 1706 lo,ex            /* for this process */

# auditconfig -setumask jdoe lo,ex            /* for this user */

# auditconfig -setsmask 103203403 lo,ex       /* for this session */
```

4. **사용자에 대한 사전 선택 마스크가 변경되었는지 확인합니다.**

   예를 들어, 마스크를 변경하기 전에 있었던 프로세스를 확인합니다.

```
# auditconfig -getpinfo 1706
audit id = jdoe(1234)
process preselection mask = ex,lo(0x40001000,0x40001000)
terminal id (maj,min,host) = 9426,65559,mach1(192.168.123.234)
audit session id = 103203403
```

▼ 특정 이벤트의 감사를 막는 방법

유지 관리를 목적으로 때때로 사이트에서 이벤트가 감사되지 않도록 막을 수 있습니다.

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안"의 "지정된 관리 권한 사용"을 참조하십시오.

1. **이벤트의 클래스를 `no` 클래스로 변경합니다.**

   참고 - 감사 구성 파일 수정의 영향에 대한 자세한 내용은 "감사 구성 파일 및 패키징" [110]을 참조하십시오.

   예를 들어, 이벤트 26 및 27은 `pm` 클래스에 속해 있습니다.

   ```
   ## audit_event file
   ...
   25:AUE_VFORK:vfork(2):ps
   26:AUE_SETGROUPS:setgroups(2):pm
   27:AUE_SETPGRP:setpgrp(2):pm
   28:AUE_SWAPON:swapon(2):no
   ...
   ```

   이러한 이벤트를 `no` 클래스로 변경합니다.

   ```
   ## audit_event file
   ...
   25:AUE_VFORK:vfork(2):ps
   26:AUE_SETGROUPS:setgroups(2):no
   27:AUE_SETPGRP:setpgrp(2):no
   28:AUE_SWAPON:swapon(2):no
   ...
   ```

   `pm` 클래스가 현재 감사되는 경우 기존 세션에서는 이벤트 26 및 27을 여전히 감사합니다. 이러한 이벤트의 감사를 중지하려면 로그인한 사용자의 사전 선택 마스크를 업데이트하는 방법 [57]의 지침을 따라 사용자의 사전 선택 마스크를 업데이트해야 합니다.

   주의 - `audit_event` 파일에서 이벤트를 주석 처리하지 마십시오. 이 파일은 `praudit` 명령에서 이진 감사 파일을 읽는 데 사용됩니다. 아카이브된 감사 파일은 파일에 나열된 이벤트를 포함할 수 있습니다.

2. **커널 이벤트를 새로 고칩니다.**

   ```
   # auditconfig -conf
   Configured 283 kernel events.
   ```

▼ 전용 파일 시스템에서 감사 파일을 압축하는 방법

감사 파일은 커질 수 있습니다. 예 4-3. “audit_binfile 플러그인에 대한 파일 크기 제한”에 나온 대로 파일 크기에 대한 상한을 설정할 수 있습니다. 이 절차에서는 압축을 사용하여 크기를 줄입니다.

시작하기 전에 ZFS File System Management 및 ZFS Storage Management 권한 프로파일이 지정된 관리자여야 합니다. ZFS Storage Management 권한 프로파일을 사용하여 저장소 풀을 만들 수 있습니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 관리 권한 사용”을 참조하십시오.

1. **감사 파일에 대한 전용 ZFS 파일 시스템을 만듭니다.**
   절차는 감사 파일에 대한 ZFS 파일 시스템을 만드는 방법 [70]을 참조하십시오.

2. **다음 옵션 중 하나를 사용하여 ZFS 저장소 풀을 압축합니다.**
   두 옵션 모두 감사 파일 시스템을 압축합니다. 감사 서비스를 새로 고치면 압축률이 표시됩니다.

   다음 예에서 ZFS 풀 auditp/auditf는 데이터 세트입니다.

   - **기본 압축 알고리즘을 사용합니다.**

     ```
     # zfs set compression=on auditp/auditf
     # audit -s
     # zfs get compressratio auditp/auditf
     NAME           PROPERTY       VALUE  SOURCE
     auditp/auditf  compressratio  4.54x  -
     ```

   - **더 높은 압축 알고리즘을 사용합니다.**

     ```
     # zfs set compression=gzip-9 auditp/auditf
     # zfs get compression auditp/auditf
     NAME           PROPERTY     VALUE     SOURCE
     auditp/auditf  compression  gzip-9    local
     ```

     gzip-9 압축 알고리즘을 사용하면 기본 압축 알고리즘인 lzjb보다 1/3 적은 공간을 차지하는 파일이 생성됩니다. 자세한 내용은 “Oracle Solaris 11.2의 ZFS 파일 시스템 관리 ”의 5 장, “Oracle Solaris ZFS 파일 시스템 관리”를 참조하십시오.

3. **감사 서비스를 새로 고칩니다.**

   ```
   # audit -s
   ```

4. **(옵션) 새로운 압축 설정을 확인합니다.**
   예를 들어, 높은 압축 알고리즘을 사용한 경우 정보가 다음과 비슷할 수 있습니다.

   ```
   # zfs get compressration auditp/auditf
   ```

```
NAME           PROPERTY       VALUE  SOURCE
auditp/auditf  compressratio  16.89x  -
```

▼ FTP 및 SFTP 파일 전송을 감사하는 방법

FTP 서비스는 파일 전송 로그를 만듭니다. `ssh` 프로토콜로 실행되는 SFTP 서비스는 `ft` 감사 클래스를 사전 선택하여 감사할 수 있습니다. 두 서비스에 대한 로그인을 감사할 수 있습니다.

참고 - FTP 서비스의 명령 및 파일 전송을 기록하는 방법은 `proftpd`(8) 매뉴얼 페이지를 참조하십시오.

사용 가능한 로깅 옵션은 ProFTPD Logging (http://www.proftpd.org/docs/howto/Logging.html)을 참조하십시오.

- **SFTP 또는 FTP를 감사할지 여부에 따라 다음 중 하나를 수행합니다.**
  - `sftp` 액세스 및 파일 전송을 기록하려면 `ft` 클래스를 편집합니다.

    `ft` 클래스에는 다음 SFTP 트랜잭션이 포함됩니다.

    ```
    % auditrecord -c ft
    file transfer: chmod ...
    file transfer: chown ...
    file transfer: get ...
    file transfer: mkdir ...
    file transfer: put ...
    file transfer: remove ...
    file transfer: rename ...
    file transfer: rmdir ...
    file transfer: session start ...
    file transfer: session end ...
    file transfer: symlink ...
    file transfer: utimes
    ```

  - FTP 서버에 대한 액세스를 기록하려면 `lo` 클래스를 감사합니다.

    다음 샘플 출력에 나온 대로 `proftpd` 데몬의 로그인 및 로그아웃으로 감사 레코드가 생성됩니다.

    ```
    % auditrecord -c lo | more
    ...
    FTP server login
    program     proftpd             See in.ftpd(1M)
    event ID    6165                AUE_ftpd
    class       lo                  (0x0000000000001000)
    ```

```
header
subject
[text]                      error message
return

FTP server logout
program     proftpd             See in.ftpd(1M)
event ID    6171                AUE_ftpd_logout
class       lo                  (0x0000000000001000)
header
subject
return
...
```

영역에서 감사 서비스 구성

감사 서비스는 영역의 감사 이벤트를 포함한 전체 시스템을 감사합니다. 비전역 영역을 설치한 시스템은 모든 영역을 동일하게 감사하거나 영역별로 감사를 구성할 수 있습니다. 자세한 내용은 “영역에서 감사 계획” [26]을 참조하십시오.

전역 영역 감사와 동일하게 비전역 영역을 감사할 경우 비전역 영역의 관리자가 감사 레코드에 액세스하지 못할 수 있습니다. 또한 전역 영역 관리자는 비전역 영역에 있는 사용자의 감사 사전 선택 마스크를 수정할 수 있습니다.

비전역 영역을 개별적으로 감사할 경우에는 감사 레코드가 비전역 영역 및 비전역 영역 루트의 전역 영역에 표시됩니다.

▼ 감사를 위해 동일하게 모든 영역을 구성하는 방법

이 절차에서는 모든 영역을 동일하게 감사할 수 있습니다. 이 방법에는 가장 작은 컴퓨터 오버헤드와 관리 리소스가 요구됩니다.

시작하기 전에 `root` 역할을 맡아야 합니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

1. **감사를 위해 전역 영역을 구성합니다.**

   다음 예외 사항을 적용하여 “감사 서비스 구성” [40]에서 작업을 완료합니다.

   - `perzone` 감사 정책을 사용으로 설정하지 않습니다.
   - `zonename` 정책을 설정합니다. 이 정책은 영역의 이름을 모든 감사 레코드에 추가합니다.

```
# auditconfig -setpolicy +zonename
```

2. **감사 구성 파일을 수정한 경우 전역 영역에서 모든 비전역 영역으로 복사합니다.**

   audit_class 또는 audit_event 파일을 수정한 경우 두 가지 방법 중 하나로 복사합니다.

   - 파일을 루프백 마운트할 수 있습니다.
   - 파일을 복사할 수 있습니다.

   비전역 영역이 실행되고 있어야 합니다.

   - **변경된 audit_class 및 audit_event 파일을 루프백 파일 시스템(lofs)으로 마운트합니다.**

     a. **전역 영역에서 비전역 영역을 정지합니다.**

     ```
     # zoneadm -z non-global-zone halt
     ```

     b. **전역 영역에서 수정한 모든 감사 구성 파일에 대해 읽기 전용 루프백 마운트를 만듭니다.**

     ```
     # zonecfg -z non-global-zone
     zone: add fs
     zone/fs: set special=/etc/security/audit-file
     zone/fs: set dir=/etc/security/audit-file
     zone/fs: set type=lofs
     zone/fs: add options [ro,nodevices,nosetuid]
     zone/fs: commit
     zone/fs: end
     zone: exit
     #
     ```

     c. **변경 사항을 적용하려면 비전역 영역을 부트합니다.**

     ```
     # zoneadm -z non-global-zone boot
     ```

     나중에 전역 영역에서 감사 구성 파일을 수정할 경우 각 영역을 재부트하여 비전역 영역에서 루프백 마운트된 파일을 새로 고칩니다.

   - **파일을 복사합니다.**

     a. **전역 영역에서 각 비전역 영역의 /etc/security 디렉토리를 나열합니다.**

     ```
     # ls /zone/zonename/root/etc/security/
     ```

     b. **변경된 audit_class 및 audit_event 파일을 각 영역의 /etc/security 디렉토리에 복사합니다.**

     ```
     # cp /etc/security/audit-file /zone/zonename/root/etc/security/audit-file
     ```

나중에 전역 영역에서 이러한 파일 중 하나를 변경할 경우 변경된 파일을 비전역 영역에 복사해야 합니다.

전역 영역에서 감사 서비스를 다시 시작하거나 영역이 재부트될 때 비전역 영역이 감사됩니다.

예 3-17 감사 구성 파일을 영역의 루프백 마운트로 마운트

이 예에서는 시스템 관리자가 `audit_class` , `audit_event` 및 `audit_warn` 파일을 수정했습니다.

`audit_warn` 파일은 전역 영역에서만 읽히므로 비전역 영역에 마운트할 필요가 없습니다.

이 시스템 `machine1`에서 관리자는 `machine1-webserver` 및 `machine1-appserver`의 두 비전역 영역을 만들었습니다. 관리자는 감사 구성 파일 수정을 마쳤습니다. 관리자가 나중에 파일을 수정할 경우 영역을 재부트하여 루프백 마운트를 다시 읽어야 합니다.

```
# zoneadm -z machine1-webserver halt
# zoneadm -z machine1-appserver halt
# zonecfg -z machine1-webserver
webserver: add fs
webserver/fs: set special=/etc/security/audit_class
webserver/fs: set dir=/etc/security/audit_class
webserver/fs: set type=lofs
webserver/fs: add options [ro,nodevices,nosetuid]
webserver/fs: commit
webserver/fs: end
webserver: add fs
webserver/fs: set special=/etc/security/audit_event
webserver/fs: set dir=/etc/security/audit_event
webserver/fs: set type=lofs
webserver/fs: add options [ro,nodevices,nosetuid]
webserver/fs: commit
webserver/fs: end
webserver: exit
#

# zonecfg -z machine1-appserver
appserver: add fs
appserver/fs: set special=/etc/security/audit_class
appserver/fs: set dir=/etc/security/audit_class
appserver/fs: set type=lofs
appserver/fs: add options [ro,nodevices,nosetuid]
appserver/fs: commit
appserver/fs: end
appserver: exit
```

비전역 영역이 재부트되면 `audit_class` 및 `audit_event` 파일은 영역에서 읽기 전용입니다.

▼ 영역별 감사를 구성하는 방법

이 절차에서는 별도의 영역 관리자가 해당 영역에서 감사 서비스를 제어할 수 있습니다. 전체 정책 옵션 목록은 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오.

시작하기 전에 감사를 구성하려면 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 감사 서비스를 사용으로 설정하려면 Audit Control 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **전역 영역에서 감사를 구성합니다.**

   a. **"감사 서비스 구성" [40]의 작업을 완료합니다.**

   b. **`perzone` 감사 정책을 추가합니다. 명령은 예 3-12. "`perzone` 감사 정책 설정"를 참조하십시오.**

   참고 - 전역 영역에서 감사 서비스를 사용으로 설정할 필요는 없습니다.

2. **감사하고자 하는 각 비전역 영역에서 감사 파일을 구성합니다.**

   a. **"감사 서비스 구성" [40]의 작업을 완료합니다.**

   b. **시스템 전역 감사 설정은 구성하지 않습니다.**
      특히 `perzone` 또는 `ahlt` 정책을 비전역 영역에 추가하지 않습니다.

3. **영역에서 감사를 사용으로 설정합니다.**

```
myzone# audit -s
```

예 3-18 비전역 영역에서 감사를 사용 안함으로 설정

이 예는 `perzone` 감사 정책이 설정된 경우에 작동합니다. `noaudit` 영역의 영역 관리자는 해당 영역에 대한 감사를 사용 안함으로 설정합니다.

```
noauditzone # auditconfig -getcond
audit condition = auditing
noauditzone # audit -t
noauditzone # auditconfig -getcond
audit condition = noaudit
```

예제: Oracle Solaris 감사 구성

이 절에서는 Oracle Solaris 감사를 구성 및 구현하는 방법에 대한 예를 제공합니다. 이 예는 특정 요구 및 요구 사항에 따라 서비스의 여러 속성에 대한 구성으로 시작됩니다. 구성이 완료된 다음에는 감사 서비스가 시작되어 구성 설정이 적용됩니다. 새로운 요구 사항을 수용하기 위해 기존 감사 구성을 개정해야 할 때마다 이 예제와 동일한 작업 순서를 따릅니다.

1. 감사 매개변수를 구성합니다.
2. 감사 서비스를 새로 고칩니다.
3. 새 감사 구성을 확인합니다.

- 먼저, 관리자는 임시 정책을 추가합니다.

```
# auditconfig -t -setpolicy +zonename
# auditconfig -getpolicy
configured audit policies = ahlt,arge,argv,perzone
active audit policies = ahlt,arge,argv,perzone,zonename
```

- 그런 다음 관리자는 대기열 제어를 지정합니다.

```
# auditconfig -setqctrl 200 20 0 0
# auditconfig -getqctrl
configured audit queue hiwater mark (records) = 200
configured audit queue lowater mark (records) = 20
configured audit queue buffer size (bytes) = 8192
configured audit queue delay (ticks) = 20
active audit queue hiwater mark (records) = 200
active audit queue lowater mark (records) = 20
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

- 그런 다음 관리자는 플러그인 속성을 지정합니다.
  - `audit_binfile` 플러그인에 대해 관리자는 `qsize` 값을 제거합니다.

```
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/audit/sys1.1,/var/audit;
p_minfree=2;p_fsize=4G;
Queue size: 200
# auditconfig -setplugin audit_binfile  "" 0
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/audit/sys1.1,/var/audit
p_minfree=2;p_fsize=4G;
```

  - `audit_syslog` 플러그인에 대해 관리자는 성공한 로그인/로그아웃 이벤트 및 실패한 실행 파일이 `syslog`에 보내지도록 구성합니다. 이 플러그인에 대한 `qsize`는 150으로 설정되었습니다.

```
# auditconfig -setplugin audit_syslog active p_flags=+lo,-ex 150
# auditconfig -getplugin audit_syslog
auditconfig -getplugin audit_syslog
Plugin: audit_syslog
Attributes: p_flags=+lo,-ex;
Queue size: 150
```

  - 관리자는 audit_remote 플러그인을 구성하거나 사용하지 않습니다.
- 그런 다음 관리자는 감사 서비스를 새로 고치고 구성을 확인합니다.
  - 임시 zonename 정책은 더 이상 설정되지 않습니다.

```
# audit -s
# auditconfig -getpolicy
configured audit policies = ahlt,arge,argv,perzone
active audit policies = ahlt,arge,argv,perzone
```

  - 대기열 제어는 동일하게 유지됩니다.

```
# auditconfig -getqctrl
configured audit queue hiwater mark (records) = 200
configured audit queue lowater mark (records) = 20
configured audit queue buffer size (bytes) = 8192
configured audit queue delay (ticks) = 20
active audit queue hiwater mark (records) = 200
active audit queue lowater mark (records) = 20
active audit queue buffer size (bytes) = 8192
active audit queue delay (ticks) = 20
```

  - audit_binfile 플러그인에는 지정된 대기열 크기가 없습니다. audit_syslog 플러그인에는 지정된 대기열 크기가 있습니다.

```
# auditconfig -getplugin
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;

Plugin: audit_syslog
Attributes: p_flags=+lo,-ex;
Queue size: 50
...
```

♦ ♦ ♦ 4 장

4 시스템 작업 모니터링

이 장에서는 시스템에서 작업을 모니터링할 수 있도록 감사 로그를 구성하는 절차를 설명합니다. 또한 다음 장에서는 기타 감사 관리 작업에 대해 설명합니다.

- 3장. 감사 서비스 관리
- 5장. 감사 데이터 작업
- 6장. 감사 서비스 문제 분석 및 해결

감사 서비스에 대한 개요는 1장. Oracle Solaris의 감사 정보를 참조하십시오. 계획 제안은 2장. 감사 계획을 참조하십시오. 참조 정보는 7장. 감사 참조를 참조하십시오.

감사 로그 구성

`audit_binfile` 및 `audit_syslog`의 두 감사 플러그인은 로컬 감사 로그를 만들 수 있습니다. 다음 작업은 이러한 로그를 구성하는 방법을 설명합니다.

감사 로그 구성

다음 작업 맵에서는 다양한 플러그인에 대한 감사 로그를 구성하기 위한 절차를 안내합니다. `audit_binfile` 플러그인의 로그 구성은 선택 사항입니다. 다른 플러그인의 로그는 관리자가 구성해야 합니다.

표 4-1 감사 로그 구성 작업 맵

작업	설명	수행 방법
`audit_binfile` 플러그인에 대한 로컬 저장소를 추가합니다	감사 파일에 대한 추가 디스크 공간을 만들고 파일 권한으로 보호합니다	감사 파일에 대한 ZFS 파일 시스템을 만드는 방법 [70]
`audit_binfile` 플러그인에 대한 저장소를 지정합니다	이진 감사 레코드에 대한 디렉토리를 식별합니다	감사 추적에 대한 감사 공간을 지정하는 방법 [73]

작업	설명	수행 방법
원격 시스템에 대한 감사 레코드의 스트리밍을 구성합니다	보호 방식을 통해 원격 저장소로 감사 레코드를 보낼 수 있습니다	원격 저장소에 감사 파일을 보내는 방법 [76]
감사 파일에 대한 원격 저장소를 구성합니다	원격 시스템에서 감사 레코드를 수신할 수 있게 해줍니다	감사 파일에 대한 원격 저장소를 구성하는 방법 [78]
`audit_syslog` 플러그인에 대한 저장소를 구성합니다.	감사 이벤트를 텍스트 형식으로 `syslog`에 스트리밍할 수 있습니다.	`syslog` 감사 로그를 구성하는 방법 [82]

▼ 감사 파일에 대한 ZFS 파일 시스템을 만드는 방법

다음 절차에서는 감사 파일에 대한 ZFS 풀과 해당하는 파일 시스템 및 마운트 지점을 만드는 방법을 설명합니다. 기본적으로 `/var/audit` 파일 시스템에는 `audit_binfile` 플러그인에 대한 감사 파일이 포함됩니다.

시작하기 전에 ZFS File System Management 및 ZFS Storage Management 권한 프로파일이 지정된 관리자여야 합니다. ZFS Storage Management 권한 프로파일을 사용하여 저장소 풀을 만들 수 있습니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **필요한 디스크 공간 크기를 결정합니다.**

   호스트당 200MB 이상의 디스크 공간을 지정합니다. 하지만 필요한 감사의 양에 따라 디스크 공간 요구 사항이 결정됩니다. 사용자의 디스크 공간 요구 사항은 이 그림보다 훨씬 높을 수 있습니다.

   참고 - 기본 클래스 사전 선택은 `lo` 클래스의 모든 기록되는 이벤트 인스턴스(로그인, 로그아웃, 역할 지정 등)에 대해 약 80바이트씩 증가하는 파일을 `/var/audit`에 만듭니다.

2. **미러링되는 ZFS 저장소 풀을 만듭니다.**

   `zpool create` 명령은 ZFS 파일 시스템에 대한 컨테이너인 저장소 풀을 만듭니다. 자세한 내용은 "Oracle Solaris 11.2의 ZFS 파일 시스템 관리 "의 1 장, "Oracle Solaris ZFS 파일 시스템(소개)"을 참조하십시오.

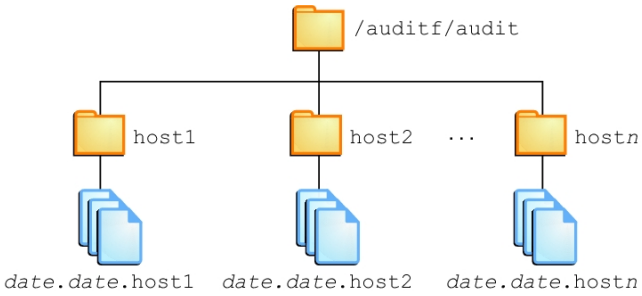
   ```
   # zpool create audit-pool mirror disk1 disk2
   ```

   예를 들어, `c3t1d0` 및 `c3t2d0`의 두 디스크에서 `auditp` 풀을 만들고 미러링합니다.

   ```
   # zpool create auditp mirror c3t1d0 c3t2d0
   ```

3. **감사 파일에 대한 ZFS 파일 시스템 및 마운트 지점을 만듭니다.**

   하나의 명령으로 파일 시스템 및 마운트 지점을 만듭니다. 생성 시 파일 시스템이 마운트됩니다. 예를 들어, 다음 그림은 호스트 이름으로 저장되는 감사 추적 저장소를 보여줍니다.



참고 - 파일 시스템을 암호화하려는 경우 생성 시 파일 시스템을 암호화해야 합니다. 예는 예 4-1. "감사 파일에 대한 암호화된 파일 시스템 만들기"를 참조하십시오.

암호화에는 관리가 필요합니다. 예를 들어, 마운트 시 문장암호가 필요합니다. 자세한 내용은 "Oracle Solaris 11.2의 ZFS 파일 시스템 관리 "의 "ZFS 파일 시스템 암호화"를 참조하십시오.

```
# zfs create -o mountpoint=/mountpoint  audit-pool/mountpoint
```

예를 들어, `auditf` 파일 시스템에 대한 `/audit` 마운트 지점을 만듭니다.

```
# zfs create -o mountpoint=/audit auditp/auditf
```

4. **감사 파일에 대한 ZFS 파일 시스템을 만듭니다.**

```
# zfs create -p auditp/auditf/system
```

예를 들어, `sys1` 시스템에 대한 암호화되지 않은 ZFS 파일 시스템을 만듭니다.

```
# zfs create -p auditp/auditf/sys1
```

5. **(옵션) 감사 파일에 대한 추가 파일 시스템을 만듭니다.**
추가 파일 시스템을 만드는 한 가지 이유는 감사 오버플로우를 막기 위함입니다. 8단계에 나온 대로 파일 시스템당 ZFS 할당량을 설정할 수 있습니다. `audit_warn` 전자 메일 별칭은 각 할당량에 도달하면 알려줍니다. 공간을 확보하기 위해 닫힌 감사 파일을 원격 서버로 이동할 수 있습니다.

```
# zfs create -p auditp/auditf/sys1.1

# zfs create -p auditp/auditf/sys1.2
```

6. **상위 감사 파일 시스템을 보호합니다.**
다음 ZFS 등록 정보는 풀의 모든 파일 시스템에 대해 `off`로 설정됩니다.

```
# zfs set devices=off auditp/auditf
```

```
# zfs set exec=off auditp/auditf

# zfs set setuid=off auditp/auditf
```

7. **풀의 감사 파일을 압축합니다.**

   일반적으로 압축은 ZFS의 파일 시스템 레벨에서 설정됩니다. 하지만 이 풀의 모든 파일 시스템에는 감사 파일이 포함되므로 압축은 풀에 대한 최상위 레벨 데이터 세트에서 설정됩니다.

   ```
   # zfs set compression=on auditp
   ```

   또한 "Oracle Solaris 11.2의 ZFS 파일 시스템 관리 "의 "ZFS 압축, 중복 제거 및 암호화 등록 정보 간의 상호 작용"을 참조하십시오.

8. **할당량을 설정합니다.**

   상위 파일 시스템, 종속 파일 시스템 또는 둘 다에서 할당량을 설정할 수 있습니다. 상위 감사 파일 시스템에서 할당량을 설정할 경우 종속 파일 시스템에 대한 할당량을 설정하면 제한이 추가됩니다.

   a. **상위 감사 파일 시스템에서 할당량을 설정합니다.**

      다음 예에서는 `auditp` 풀의 두 디스크가 모두 할당량에 도달하면 `audit_warn` 스크립트가 감사 관리자에게 알려줍니다.

      ```
      # zfs set quota=510G auditp/auditf
      ```

   b. **종속 감사 파일 시스템에서 할당량을 설정합니다.**

      다음 예에서는 `auditp/auditf/`*system* 파일 시스템에 대한 할당량에 도달하면 `audit_warn` 스크립트가 감사 관리자에게 알려줍니다.

      ```
      # zfs set quota=170G auditp/auditf/sys1

      # zfs set quota=170G auditp/auditf/sys1.1

      # zfs set quota=165G auditp/auditf/sys1.2
      ```

9. **대량 풀의 경우 감사 파일의 크기를 제한합니다.**

   기본적으로 감사 파일은 풀의 크기까지 커질 수 있습니다. 관리 용이성을 위해 감사 파일의 크기를 제한합니다. 예 4-3. "`audit_binfile` 플러그인에 대한 파일 크기 제한"를 참조하십시오.

예 4-1 감사 파일에 대한 암호화된 파일 시스템 만들기

사이트 보안 요구 사항을 준수하기 위해 관리자는 다음과 같은 단계를 수행합니다.

1. 필요한 경우 암호화된 감사 로그를 저장하기 위한 새로운 ZFS 풀을 만듭니다.
2. 암호화 키를 생성합니다.

3. 감사 로그를 저장하기 위해 암호화가 설정된 감사 파일 시스템을 만들고 마운트 지점을 설정합니다.
4. 암호화된 디렉토리를 사용하도록 감사를 구성합니다.
5. 감사 서비스를 새로 고쳐서 새로운 구성 설정을 적용합니다.

```
# zpool create auditp mirror disk1 disk2
```

```
# pktool genkey keystore=file outkey=/filename keytype=aes keylen=256

# zfs create -o encryption=aes-256-ccm \
-o keysource=raw,file:///filename \
-o compression=on -o mountpoint=/audit auditp/auditf

# auditconfig -setplugin audit_binfile p_dir=/audit/

# audit -s
```

예제의 *filename*과 같이 키가 저장된 위치의 파일을 백업하고 보호해야 합니다.

관리자가 `auditf` 파일 시스템에 추가 파일 시스템을 만들면 이러한 종속 파일 시스템도 암호화됩니다.

예 4-2 `/var/audit` 디렉토리에서 할당량 설정

이 예에서는 관리자가 기본 감사 파일 시스템에서 할당량을 설정합니다. 이 할당량에 도달하면 `audit_warn` 스크립트가 감사 관리자에게 경고합니다.

```
# zfs set quota=252G rpool/var/audit
```

▼ 감사 추적에 대한 감사 공간을 지정하는 방법

이 절차에서는 `audit_binfile` 플러러그인에 대한 속성을 사용하여 감사 추적에 추가 디스크 공간을 지정합니다.

시작하기 전에 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **`audit_binfile` 플러그인에 대한 속성을 결정합니다.**
   `audit_binfile`(5) 매뉴얼 페이지의 `OBJECT ATTRIBUTES` 절을 참조하십시오.

```
# man audit_binfile

...
OBJECT ATTRIBUTES
The p_dir attribute specifies where the audit files will be created.
```

```
The directories are listed in the order in which they are to be used.

The p_minfree attribute defines the percentage of free space that the
audit system requires before the audit daemon invokes the audit_warn
script.

The p_fsize attribute defines the maximum size that an audit
file can become before it is automatically closed and a new
audit file is opened. ... The format of the p_fsize value can
be specified as an exact value in bytes or in a human-readable
form with a suffix of  B,  K, M, G, T, P, E, Z (for bytes,
kilobytes, megabytes, gigabytes, terabytes,  petabytes, exabytes,
or zettabytes, respectively). Suffixes of KB, MB, GB, TB, PB, EB,
and ZB are also accepted.
```

2. **감사 추적에 디렉토리를 추가하려면 `p_dir` 속성을 지정합니다.**

   기본 파일 시스템은 `/var/audit`입니다.

   ```
   # auditconfig -setplugin audit_binfile p_dir=/audit/sys1.1,/var/audit
   ```

   위의 명령은 `/audit/sys1.1` 파일 시스템을 감사 파일에 대한 기본 디렉토리로 설정하고 기본 `/var/audit` 파일 시스템을 보조 디렉토리로 설정합니다. 이 시나리오에서는 `/var/audit`가 마지막 의존 디렉토리입니다. 이 구성이 성공하려면 `/audit/sys1.1` 파일 시스템이 존재해야 합니다.

   감사 파일에 대한 ZFS 파일 시스템을 만드는 방법 [70]에서 유사한 파일 시스템을 만들었습니다.

3. **감사 서비스를 새로 고칩니다.**

   `auditconfig -setplugin` 명령은 구성된 값을 설정합니다. 이 값은 감사 서비스의 등록 정보이므로 서비스를 새로 고치거나 다시 시작해도 복원됩니다. 감사 서비스가 새로 고쳐지거나 다시 시작되면 구성된 값이 활성이 됩니다. 구성된 값 및 활성 값에 대한 자세한 내용은 `auditconfig`(1M) 매뉴얼 페이지를 참조하십시오.

   ```
   # audit -s
   ```

예 4-3 `audit_binfile` 플러그인에 대한 파일 크기 제한

다음 예에서는 이진 감사 파일의 크기가 특정 크기로 설정됩니다. 크기는 메가바이트로 지정됩니다.

```
# auditconfig -setplugin audit_binfile p_fsize=4M

# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=4M;p_minfree=1;
```

기본적으로 감사 파일은 무제한으로 커질 수 있습니다. 더 작은 감사 파일을 만들기 위해 관리자는 4MB의 파일 크기 제한을 지정합니다. 제한 크기에 도달하면 감사 서비스는 새 파일을 만듭니다. 파일 크기 제한은 관리자가 감사 서비스를 새로 고친 후 적용됩니다.

```
# audit -s
```

예 4-4 로그 교체를 위한 시간 지정

다음 예제에서는 감사 파일에 대한 시간 제한이 설정되어 있습니다. 시간 제한은 시간, 일, 주, 월 또는 연 단위로 지정됩니다.

```
# auditconfig -setplugin audit_binfile "p_age=1w"

# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_age=1w;
Queue size: 200
```

기본적으로 감사 파일은 시간 제한이 없습니다. 파일은 외부 작업으로 인해 파일 교체가 수행될 때까지 무기한 열린 상태로 유지됩니다. 관리자는 파일의 시간 제한을 1주로 설정합니다. 이 기간이 지나면 새 감사 파일이 열립니다. 새로운 시간 제한을 구현하려면 관리자가 감사 서비스를 새로 고칩니다.

```
# audit -s
```

예 4-5 감사 플러그인에 여러 변경 사항 지정

다음 예에서는 처리량이 많고 ZFS 풀이 큰 시스템의 관리자가 `audit_binfile` 플러그인에 대한 대기열 크기, 이진 파일 크기 및 소프트 제한 경고를 변경합니다. 관리자는 감사 파일이 4GB까지 커질 수 있도록 허용하고, ZFS 풀의 2%가 남으면 경고를 받으며, 허용된 할당량 크기를 두 배로 늘립니다. 기본 대기열 크기는 `active audit queue hiwater mark (records) = 100`과 같이 커널 감사 대기열에 대한 고수위 마크인 `100`입니다. 또한 감사 파일의 시간 제한은 2주로 설정됩니다.

```
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=2G;p_minfree=1;

# auditconfig -setplugin audit_binfile \
      "p_minfree=2;p_fsize=4G;p_age=2w" 200

# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;p_age=2w;
Queue size: 200
```

변경된 지정 사항은 관리자가 감사 서비스를 새로 고친 후 적용됩니다.

```
# audit -s
```

예 4-6 감사 플러그인에 대한 대기열 크기 제거

다음 예에서는 `audit_binfile` 플러그인에 대한 대기열 크기가 제거됩니다.

```
# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;
Queue size: 200

# auditconfig -setplugin audit_binfile "" 0

# auditconfig -getplugin audit_binfile
Plugin: audit_binfile
Attributes: p_dir=/var/audit;p_fsize=4G;p_minfree=2;
```

비어 있는 따옴표("")는 현재 속성 값을 보존합니다. 마지막 `0`은 플러그인의 대기열 크기를 기본값으로 설정합니다.

플러그인에 대한 `qsize` 지정 변경 사항은 관리자가 감사 서비스를 새로 고친 후 적용됩니다.

```
# audit -s
```

예 4-7 경고에 대한 소프트 제한 설정

이 예에서는 모든 감사 파일 시스템에 대한 최소 사용 가능 공간 레벨을 설정하여 파일 시스템의 2%를 아직 사용할 수 있을 때 경고를 보냅니다.

```
# auditconfig -setplugin audit_binfile p_minfree=2
```

기본값은 1%입니다. 대형 ZFS 풀의 경우 적당히 낮은 백분율을 선택합니다. 예를 들어, 16TB 풀의 10%는 16GB이므로 충분한 디스크 공간이 남아 있을 때 감사 관리자에게 경고를 보내게 됩니다. 값이 2이면 약 2GB의 디스크 공간이 남아 있을 때 `audit_warn` 메시지를 보냅니다.

`audit_warn` 전자 메일 별칭이 경고를 수신합니다. 별칭을 설정하려면 `audit_warn` 전자 메일 별칭을 구성하는 방법 [50]을 참조하십시오.

또한 대형 풀의 경우 관리자는 파일 크기를 3GB로 제한할 수 있습니다.

```
# auditconfig -setplugin audit_binfile p_fsize=3G
```

플러그인에 대한 `p_minfree` 및 `p_fsize` 지정 사항은 관리자가 감사 서비스를 새로 고친 후 적용됩니다.

```
# audit -s
```

▼ 원격 저장소에 감사 파일을 보내는 방법

이 절차에서는 `audit_remote` 플러그인의 속성을 사용하여 원격 감사 저장소에 감사 추적을 보냅니다. Oracle Solaris 시스템에서 원격 저장소를 구성하려면 감사 파일에 대한 원격 저장소를 구성하는 방법 [78]을 참조하십시오.

시작하기 전에 원격 저장소에서 감사 파일의 수신자여야 합니다. Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안"의 "지정된 관리 권한 사용"을 참조하십시오.

1. **audit_remote 플러그인의 속성을 결정합니다.**

   audit_remote(5) 매뉴얼 페이지의 OBJECT ATTRIBUTES 절을 참조하십시오.

   ```
   # man audit_remote

   ...
   OBJECT ATTRIBUTES
   The p_hosts attribute specifies the remote servers.
   You can also specify the port number and the GSS-API
   mechanism.

   The p_retries attribute specifies the number of retries for
   connecting and sending data. The default is 3.

   The p_timeout attribute specifies the number of seconds
   in which a connection times out.
   ```

   기본 포트는 solaris_audit IANA 지정 포트인 16162/tcp입니다. 기본 방식은 kerberos_v5입니다. 시간 초과 기본값은 5초입니다. 플러그인에 대한 대기열 크기도 지정할 수 있습니다.

2. **원격 수신 시스템을 지정하려면 p_hosts 속성을 사용합니다.**

   이 예에서 수신 시스템은 서로 다른 포트를 사용합니다.

   ```
   # auditconfig -setplugin audit_remote \
        p_hosts=ars.example.com:16088:kerberos_v5
   ```

3. **변경하려는 플러그인의 다른 속성을 지정합니다.**

   예를 들어, 다음 명령은 모든 선택적 속성에 대한 값을 지정합니다.

   ```
   # auditconfig -setplugin audit_remote "p_retries=;p_timeout=3" 300
   ```

4. **값을 확인한 후 플러그인을 활성화합니다.**

   예를 들어, 다음 명령은 플러그인의 값을 지정하고 확인합니다.

   ```
   # auditconfig -getplugin audit_remote
   Plugin: audit_remote (inactive)
   Attributes: p_hosts=ars.example.com:16088:kerberos_v5;p_retries=5;p_timeout=3;
   Queue size: 300

   # auditconfig -setplugin audit_remote active
   ```

5. **감사 서비스를 새로 고칩니다.**

   감사 서비스는 새로 고쳐질 때 감사 플러그인 변경 사항을 읽습니다.

   ```
   # audit -s
   ```

예 4-8 감사 대기열 버퍼 크기 조정

이 예에서 감사 대기열은 audit_remote 플러그인 사용 시 꽉 찹니다. 이 감사되는 시스템은 여러 클래스를 감사하도록 구성되었으며 트래픽이 많고 속도가 느린 네트워크를 사용하는 중입니다. 관리자는 플러그인의 버퍼 크기를 확대하여 감사 대기열이 증가할 수 있도록 하되 대기열에서 레코드를 제거하기 전에 버퍼 제한을 초과하지 않도록 할 수 있습니다.

```
audsys1 # auditconfig -setplugin audit_remote "" 1000

audsys1 # audit -s
```

▼ 감사 파일에 대한 원격 저장소를 구성하는 방법

이 절차에서는 원격 시스템 ARS(감사 원격 서버)가 하나 이상의 감사되는 시스템에서 감사 레코드를 수신하고 저장하도록 구성합니다. 그런 후 원격 서버에서 감사 데몬을 활성화합니다.

구성은 두 가지입니다. 첫째, 감사 데이터를 보안 방식으로 전송하도록 기본 보안 방식을 구성합니다. 즉, KDC를 구성합니다. 둘째, 감사되는 시스템과 ARS 모두에서 감사 서비스를 구성합니다. 이 절차에서는 ARS와 KDC가 동일한 서버에 있는 하나의 감사 클라이언트 및 하나의 ARS가 포함된 시나리오를 보여줍니다. 마찬가지로 보다 복잡한 시나리오도 구성할 수 있습니다. 처음 4개 단계에서는 KDC의 구성을 설명하고, 마지막 단계에서는 감사 서비스의 구성을 설명합니다.

시작하기 전에 다음을 완료했는지 확인합니다. root 역할을 맡아야 합니다.

- root 역할을 맡았습니다.
- 감사 레코드를 원격 저장소에 스트리밍하기 위한 준비 방법 [31]에 설명된 대로 Kerberos 패키지를 설치했습니다.
- 원격 저장소에 감사 파일을 보내는 방법 [76]에 설명된 대로 감사되는 시스템을 구성한 관리자와 함께 작업합니다.

1. **사이트에 아직 KDC가 구성되지 않았으면 구성합니다.**

   감사되는 시스템 및 ARS가 모두 사용할 수 있는 시스템에 KDC가 필요하며, 각 시스템에 호스트 주체가 있어야 하고 audit 서비스 주체가 필요합니다. 다음 예제는 KDC 구성 전략을 보여줍니다.

   ```
   arstore # kdcmgr -a audr/admin -r EXAMPLE.COM create master
   ```

   이 명령은 관리 주체인 audr/admin을 사용하여 EXAMPLE.COM 영역에 마스터 KDC를 만들고 마스터 KDC를 사용으로 설정한 후 Kerberos 서비스를 시작합니다.

2. **KDC를 사용할 수 있는지 확인합니다.**

   자세한 내용은 kdcmgr(1M) 매뉴얼 페이지를 참조하십시오.

```
# kdcmgr status

KDC Status Information
------------------------------------------
svc:/network/security/krb5kdc:default (Kerberos key distribution center)
State: online since Wed Feb 29 01:59:27 2012
See: man -M /usr/share/man -s 1M krb5kdc
See: /var/svc/log/network-security-krb5kdc:default.log
Impact: None.

KDC Master Status Information
------------------------------------------
svc:/network/security/kadmin:default (Kerberos administration daemon)
State: online since Wed Feb 29 01:59:28 2012
See: man -M /usr/share/man -s 1M kadmind
See: /var/svc/log/network-security-kadmin:default.log
Impact: None.

Transaction Log Information
------------------------------------------

Kerberos update log (/var/krb5/principal.ulog)
Update log dump :
Log version # : 1
Log state : Stable
Entry block size : 2048
Number of entries : 13
First serial # : 1
Last serial # : 13
First time stamp : Wed Feb 29 01:59:27 2012
Last time stamp : Mon Mar 5 19:29:28 2012


Kerberos Related File Information
------------------------------------------
(Displays any missing files)
```

3. **audit 서비스 주체를 KDC keytab 파일에 추가합니다.**
   KDC 시스템에서 `kadmin.local` 명령을 입력하여 주체를 추가할 수 있습니다. 또는 `kadmin` 명령을 사용하고 암호를 제공하여 주체를 원격으로 추가할 수 있습니다. 이 예에서 `arstore` 시스템은 KDC를 실행합니다.

   ```
   # kadmin -p audr/admin

   kadmin: addprinc -randkey audit/arstore.example.com@EXAMPLE.COM

   kadmin: ktadd audit/arstore.example.com@EXAMPLE.COM
   ```

4. **감사되는 각 시스템에서 키를 추가합니다.**
   수신자와 발신자 모두 키가 있어야 합니다.

   ```
   enigma # kclient
   ```

```
.. Enter the Kerberos realm:
EXAMPLE.COM

.. KDC hostname for the above realm:
arstore.example.com

.. Will this client need service keys ? [y/n]:
y
```

5. **ARS에서 감사 서비스를 구성합니다.**

   - **Kerberos 영역의 모든 감사되는 시스템에서 감사 레코드를 받아들이는 그룹을 만들려면 연결 그룹을 지정합니다.**

     ```
     # auditconfig -setremote group create Bank_A
     ```

     Bank_A는 연결 그룹입니다. hosts 속성이 정의되지 않았기 때문에 이 그룹은 모든 연결을 받아들이는 와일드카드 그룹입니다. 이 Kerberos 영역에서 audit_remote 플러그인이 올바르게 구성된 모든 감사되는 시스템은 이 ARS에 연결할 수 있습니다.

   - **이 그룹에 대한 연결을 제한하려면 이 저장소를 사용할 수 있는 감사되는 시스템을 지정합니다.**

     ```
     # auditconfig -setremote group Bank_A "hosts=enigma.example.com"
     ```

     연결 그룹 Bank_A는 이제 enigma 시스템의 연결만 받아들입니다. 다른 호스트의 연결은 거절됩니다.

   - **이 그룹의 감사 파일이 너무 커지지 않도록 방지하려면 최대 크기를 설정합니다.**

     ```
     # auditconfig -setremote group Bank_A "binfile_fsize=4GB"

     # auditconfig -getremote
     Audit Remote Server
     Attributes: listen_address=;login_grace_time=30;max_startups=10;listen_port=0;
     Connection group: Bank_A (inactive)
     Attributes: binfile_dir=/var/audit;binfile_fsize=4GB;binfile_minfree=1;
     hosts=enigma.example.com;
     ```

6. **감사되는 시스템에서 감사 서비스를 구성합니다.**
   ARS를 지정하려면 p_hosts 속성을 사용합니다.

   ```
   enigma # auditconfig -setplugin audit_remote \
        active p_hosts=arstore.example.com

   enigma # auditconfig -getplugin audit_remote
   Plugin: audit_remote
   Attributes: p_retries=3;p_timeout=5;p_hosts=arstore.example.com;
   ```

7. **감사 서비스를 새로 고칩니다.**

감사 서비스는 새로 고쳐질 때 감사 플러그인 변경 사항을 읽습니다.

```
# audit -s
```

KDC는 이제 감사되는 시스템 enigma와 ARS 사이의 연결을 관리합니다.

예 4-9 감사 레코드를 동일 ARS의 다른 파일 위치로 스트리밍

이 예에서는 이 절차의 예를 확장합니다. 관리자는 두 개의 연결 그룹을 만들어서 ARS의 호스트별로 감사 레코드를 구분합니다.

audsys1의 감사 파일은 이 ARS에서 Bank_A 연결 그룹으로 스트리밍됩니다.

```
arstore # auditconfig -setremote group create Bank_A

arstore # auditconfig -setremote group active Bank_A "hosts=audsys1" \
   "hosts=audsys1;binfile_dir=/var/audit/audsys1;binfile_fsize=4M;"
```

audsys2의 감사 파일은 Bank_B 연결 그룹으로 스트리밍됩니다.

```
arstore # auditconfig -setremote group create Bank_B

arstore # auditconfig -setremote group active Bank_B \
"hosts=audsys2;binfile_dir=/var/audit/audsys2;binfile_fsize=4M;"
```

유지 관리를 쉽게 하기 위해 관리자는 다른 속성 값을 동일하게 설정합니다.

```
arstore # auditconfig -getremote
Audit Remote Server
Attributes: listen_address=;login_grace_time=30;max_startups=10;listen_port=0;

Connection group: Bank_A
Attributes: binfile_dir=/var/audit/audsys1;binfile_fsize=4M;binfile_minfree=1;
hosts=audsys1

Connection group: Bank_B
Attributes: binfile_dir=/var/audit/audsys2;binfile_fsize=4M;binfile_minfree=1;
hosts=audsys2
```

예 4-10 KDC와 다른 시스템에 ARS 배치

이 예에서 관리자는 KDC와 다른 시스템에 ARS를 배치합니다. 첫째, 관리자가 마스터 KDC를 만들고 구성합니다.

```
kserv # kdcmgr -a audr/admin -r EXAMPLE.COM create master

kserv # kadmin.local -p audr/admin

kadmin: addprinc -randkey \
audit/arstore.example.com@EXAMPLE.COM
```

```
kadmin: ktadd -t /tmp/krb5.keytab.audit \
     audit/arstore.example.com@EXAMPLE.COM
```

/tmp/krb5.keytab.audit 파일을 ARS arstore로 안전하게 전송한 후 관리자가 파일을 올바른 위치로 이동합니다.

```
arstore # chown root:root krb5.keytab.audit

arstore # chmod 600 krb5.keytab.audit

arstore # mv krb5.keytab.audit /etc/krb5/krb5.keytab
```

파일을 다시 작성하는 대신 관리자는 ARS에서 ktutil 명령을 사용하여 KDC krb5.keytab.audit 파일을 arstore의 /etc/krb5/krb5.keytab 파일에 있는 기존 키와 병합할 수도 있습니다.

마지막으로 관리자가 감사되는 시스템에서 키를 생성합니다.

```
enigma # kclient

.. Enter the Kerberos realm: EXAMPLE.COM

.. KDC hostname for the above realm: kserv.example.com

.. Will this client need service keys ? [y/n]: y
```

▼ syslog 감사 로그를 구성하는 방법

감사 서비스에서 감사 대기열의 감사 레코드 중 일부나 모두를 syslog 유틸리티에 복사하도록 지시할 수 있습니다. 이진 감사 데이터와 텍스트 요약을 모두 기록할 경우 이진 데이터는 완전한 감사 레코드를 제공하고, 요약은 실시간 검토를 위해 데이터를 필터링합니다.

시작하기 전에 audit_syslog 플러그인을 구성하려면 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. syslog 유틸리티를 구성하고 auditlog 파일을 만들려면 root 역할을 맡아야 합니다.

1. **audit_syslog 플러그인에 전송할 감사 클래스를 선택하고 플러그인을 활성화합니다.**

   참고 - p_flags 감사 클래스는 시스템 기본값으로 또는 사용자나 권한 프로파일의 감사 플래그로 사전 선택되어야 합니다. 사전 선택되지 않은 클래스에 대한 레코드는 수집되지 않습니다.

   ```
   # auditconfig -setplugin audit_syslog \
        active p_flags=lo,+as,-ss
   ```

2. **syslog 유틸리티를 구성합니다.**

a. **audit.notice 항목을 syslog.conf 파일에 추가합니다.**
항목에는 로그 파일의 위치가 포함됩니다.

```
# cat /etc/syslog.conf

…
audit.notice       /var/adm/auditlog
```

b. **로그 파일을 만듭니다.**

```
# touch /var/adm/auditlog
```

c. **로그 파일의 권한을 640으로 설정합니다.**

```
# chmod 640 /var/adm/auditlog
```

d. **시스템에서 실행 중인 system-log 서비스 인스턴스를 확인합니다.**

```
# svcs system-log

STATE        STIME      FMRI
online       Nov_27     svc:/system/system-log:default
disabled     Nov 27     svc:/system/system-log:rsyslog
```

e. **활성 syslog 서비스 인스턴스에 대한 구성 정보를 새로 고칩니다.**

```
# svcadm refresh system/system-log:default
```

3. **감사 서비스를 새로 고칩니다.**
감사 서비스는 새로 고쳐질 때 감사 플러그인 변경 사항을 읽습니다.

```
# audit -s
```

4. **정기적으로 syslog 로그 파일을 아카이브합니다.**
감사 서비스는 확장 출력을 생성할 수 있습니다. 로그를 관리하려면 logadm(1M) 매뉴얼 페이지를 참조하십시오.

예 4-11 syslog 출력에 대한 감사 클래스 지정

다음 예에서는 syslog 유틸리티가 사전 선택된 감사 클래스의 하위 세트를 수집합니다. pf 클래스는 예 3-15. “새 감사 클래스 만들기”에서 만들어졌습니다.

```
# auditconfig -setnaflags lo,na

# auditconfig -setflags lo,ss

# usermod -K audit_flags=pf:no jdoe
```

```
# auditconfig -setplugin audit_syslog \
    active p_flags=lo,+na,-ss,+pf
```

auditconfig 명령에 대한 인수는 시스템에서 모든 로그인/로그아웃, 지정 불가능 및 시스템 상태 감사 레코드의 변경 사항을 수집하도록 지시합니다. audit_syslog 플러그인 항목은 syslog 유틸리티에서 모든 로그인, 성공한 지정 불가능 이벤트 및 시스템의 상태의 실패한 변경 사항을 수집하도록 지시합니다.

jdoe 사용자의 경우 이진 유틸리티가 pfexec 명령에 대한 성공 및 실패한 호출을 수집합니다. syslog 유틸리티는 pfexec 명령에 대해 성공한 호출을 수집합니다.

예 4-12 원격 시스템에 syslog 감사 레코드 두기

syslog.conf 파일의 audit.notice 항목이 원격 시스템을 가리키도록 변경할 수 있습니다. 예를 들어, 로컬 시스템의 이름은 sys1.1입니다. 원격 시스템은 remote1입니다.

```
sys1.1 # cat /etc/syslog.conf

…
audit.notice       @remote1
```

remote1 시스템에 있는 syslog.conf 파일의 audit.notice 항목은 로그 파일을 가리킵니다.

```
remote1 # cat /etc/syslog.conf

…
audit.notice       /var/adm/auditlog
```

♦ ♦ ♦ 5 장

5 감사 데이터 작업

이 장에서는 여러 다른 로컬 시스템에서 생성되는 감사 데이터를 사용하는 데 도움이 되는 절차를 제공합니다. 이 장에서는 다음 내용을 다룹니다.

- “감사 추적 데이터 표시” [85]
- “로컬 시스템에서 감사 레코드 관리” [93]

또한 다음 장에서는 기타 감사 관리 작업에 대해 설명합니다.

- 3장. 감사 서비스 관리
- 4장. 시스템 작업 모니터링
- 6장. 감사 서비스 문제 분석 및 해결

감사 서비스에 대한 개요는 1장. Oracle Solaris의 감사 정보를 참조하십시오. 계획 제안은 2장. 감사 계획을 참조하십시오. 참조 정보는 7장. 감사 참조를 참조하십시오.

감사 추적 데이터 표시

기본 플러그인 `audit_binfile`은 감사 추적을 만듭니다. 추적은 대량의 데이터를 포함할 수 있습니다. 다음 절에서는 이 데이터의 사용 방법에 대해 설명합니다.

감사 레코드 정의 표시

감사 레코드 정의를 표시하려면 `auditrecord` 명령을 사용합니다. 정의는 감사 이벤트 번호, 감사 클래스, 선택 마스크 및 감사 이벤트의 레코드 형식을 제공합니다.

```
% auditrecord -options
```

명령으로 생성되는 화면 출력은 다음 일부 목록에 표시된 것처럼 사용하는 옵션에 따라 달라집니다.

- `-p` 옵션은 프로그램의 감사 레코드 정의를 표시합니다.
- `-c` 옵션은 감사 클래스의 감사 레코드 정의를 표시합니다.
- `-a` 옵션은 모든 감사 이벤트 정의를 나열합니다.

또한 표시된 출력을 파일로 인쇄할 수 있습니다.

자세한 내용은 auditrecord(1M) 매뉴얼 페이지를 참조하십시오.

예 5-1 프로그램의 감사 레코드 정의 표시

이 예에서는 login 프로그램으로 생성되는 모든 감사 레코드의 정의가 표시됩니다. 로그인 프로그램에는 rlogin, telnet, newgrp 및 Oracle Solaris의 Secure Shell 기능이 포함됩니다.

```
% auditrecord -p login
...
login: logout
program     various              See login(1)
event ID    6153                 AUE_logout
class       lo                  (0x0000000000001000)
...
newgrp
program     newgrp               See newgrp login
event ID    6212                 AUE_newgrp_login
class       lo                  (0x0000000000001000)
...
rlogin
program     /usr/sbin/login      See login(1) - rlogin
event ID    6155                 AUE_rlogin
class       lo                   (0x0000000000001000)
...
/usr/lib/ssh/sshd
program     /usr/lib/ssh/sshd    See login - ssh
event ID    6172                 AUE_ssh
class       lo                   (0x0000000000001000)
...
telnet login
program     /usr/sbin/login      See login(1) - telnet
event ID    6154                 AUE_telnet
class       lo                   (0x0000000000001000)
…
```

예 5-2 감사 클래스의 감사 레코드 정의 표시

이 예에서는 Example 3–15에서 만들어진 예 3-15. “새 감사 클래스 만들기” 클래스의 모든 감사 레코드 정의가 표시됩니다.

```
% auditrecord -c pf
pfexec
system call pfexec               See execve(2) with pfexec enabled
event ID    116                  AUE_PFEXEC
class       pf                   (0x0100000000000000)
header
path                    pathname of the executable
```

```
path                    pathname of working directory
[privileges]            privileges if the limit or inheritable set are changed
[privileges]            privileges if the limit or inheritable set are changed
[process]               process if ruid, euid, rgid or egid is changed
exec_arguments
[exec_environment]      output if arge policy is set
subject
[use_of_privilege]
return
```

use_of_privilege 토큰은 권한이 사용될 때마다 기록됩니다. privileges 토큰은 제한 또는 상속 가능한 설정이 변경될 경우 기록됩니다. process 토큰은 ID가 변경될 경우 기록됩니다. 이러한 토큰이 레코드에 포함되기 위해 필요한 정책 옵션은 없습니다.

예 5-3 감사 레코드 정의를 파일로 인쇄

이 예제에서는 모든 감사 레코드 정의를 HTML 형식의 파일에 넣기 위해 -h 옵션이 추가되었습니다. 브라우저에서 HTML 파일을 표시하면 브라우저의 Find(찾기) 도구를 사용하여 특정 감사 레코드 정의를 찾습니다.

```
% auditrecord -ah > audit.events.html
```

표시할 감사 이벤트 선택

Audit Review 권한 프로파일이 지정된 관리자는 auditreduce 명령을 사용하여 검사할 감사 레코드를 필터링할 수 있습니다. 이 명령은 입력 파일을 결합할 때 관심이 적은 레코드를 없앨 수 있습니다.

auditreduce -*option argument* **[***optional-file***]**

여기서 *argument*는 옵션에 필요한 특정 인수입니다.

다음은 *record selection* 옵션의 일부 목록 및 해당 인수입니다.

-c 감사 클래스를 선택합니다. 여기서 *argument*는 ua와 같은 감사 클래스입니다.

-d 특정 날짜의 모든 이벤트를 선택합니다. *argument*에 대한 날짜 형식은 *yyymmdd*입니다. -b 및 -a와 같은 다른 날짜 옵션은 각각 특정 날짜 전후의 이벤트를 선택합니다.

-u 특정 사용자에게 지정 가능한 모든 이벤트를 선택합니다. 이 옵션의 경우 사용자 이름을 지정합니다. 다른 사용자 옵션인 -e는 유효 사용자 ID에 지정 가능한 모든 이벤트를 선택합니다.

-g 특정 그룹에 지정 가능한 모든 이벤트를 선택합니다. 이 옵션의 경우 그룹 이름을 지정합니다.

-c 사전 선택된 감사 클래스의 모든 이벤트를 선택합니다. 이 옵션을 사용하려면 감사 클래스 이름을 지정합니다.

-m 특정 감사 이벤트의 모든 인스턴스를 선택합니다.

-o 객체 유형별로 선택합니다. 파일, 그룹, 파일 소유자, FMRI, PID 및 기타 객체 유형별로 선택하려면 이 옵션을 사용합니다.

optional-file 감사 파일의 이름입니다.

명령에는 또한 다음 예제에 표시된 것처럼 모두 대문자로 된 *file selection* 옵션이 사용됩니다. 전체 옵션 목록은 auditreduce(1M) 매뉴얼 페이지를 참조하십시오.

예 5-4 감사 파일 결합 및 줄이기

이 예제에서는 1개월이 지난 감사 파일의 로그인 및 로그아웃 레코드만 보존됩니다. 이 예제에서는 현재 날짜가 9월 27일이라고 가정합니다. 전체 감사 추적을 검색해야 하는 경우 백업 매체에서 추적을 복구할 수 있습니다. -O 옵션은 명령 출력을 lo.summary 파일로 지정합니다.

```
# cd /var/audit/audit_summary
# auditreduce -O lo.summary -b 20100827 -c lo; compress *lo.summary
```

예 5-5 한 사용자의 감사 레코드를 요약 파일로 복사

이 예에서는 특정 사용자의 이름을 포함하는 감사 추적의 레코드가 병합됩니다. -e 옵션은 유효 사용자를 찾습니다. -u 옵션은 로그인 사용자를 찾습니다. -O 옵션은 출력을 tamiko 파일로 지정합니다.

```
# cd /var/audit/audit_summary
# auditreduce -e tamiko -O tamiko
```

표시되는 정보 범위를 더욱 좁힐 수 있습니다. 다음 예제에서는 다음 항목이 필터링되고 tamikolo라는 파일로 인쇄됩니다.

- -c 옵션으로 지정된 사용자 로그인 및 로그아웃 시간
- -d 옵션으로 지정된 날짜 2013년 9월 7일. 날짜의 짧은 형식은 *yyyymmdd*입니다.
- -u 옵션으로 지정된 사용자 이름 tamiko
- -M 옵션으로 지정된 시스템 이름

```
# auditreduce -M tamiko -O tamikolo -d 20130907 -u tamiko -c lo
```

예 5-6 선택한 레코드를 단일 파일로 병합

이 예에서는 특정 일에 대한 로그인 및 로그아웃 레코드가 감사 추적에서 선택됩니다. 레코드는 대상 파일로 병합됩니다. 대상 파일은 감사 루트 디렉토리를 포함하는 파일 시스템 이외의 파일 시스템에 쓰여집니다.

```
# auditreduce -c lo -d 20130827 -O /var/audit/audit_summary/logins

# ls /var/audit/audit_summary/*logins
/var/audit/audit_summary/20130827183936.20130827232326.logins
```

이진 감사 파일의 컨텐츠 보기

Audit Review 권한 프로파일이 지정된 관리자는 praudit 명령을 사용하여 이진 감사 파일의 컨텐츠를 볼 수 있습니다.

```
# praudit options
```

다음은 일부 옵션 목록입니다. 이러한 옵션 중 하나를 -l 옵션과 결합하여 각 레코드를 한 행에 표시할 수 있습니다.

-s 행당 토큰 하나씩 짧은 형식으로 감사 레코드를 표시합니다.

-r 행당 토큰 하나씩 원시 형식으로 감사 레코드를 표시합니다.

-x 행당 토큰 하나씩 XML 형식으로 감사 레코드를 표시합니다. 이 옵션은 추가 처리에 유용합니다.

또한 auditreduce 명령의 praudit 출력을 파이프하여 auditreduce 및 praudit 명령을 함께 사용할 수 있습니다.

예 5-7 짧은 형식으로 감사 레코드 표시

이 예제에서는 auditreduce 명령으로 추출된 로그인 및 로그아웃 이벤트가 짧은 형식으로 표시됩니다.

```
# auditreduce -c lo | praudit -s

header,69,2,AUE_screenlock,,mach1,2010-10-14 08:02:56.348 -07:00
subject,jdoe,root,staff,jdoe,staff,856,50036632,82 0 mach1
return,success,0
sequence,1298
```

예 5-8 원시 형식으로 감사 레코드 표시

이 예제에서는 auditreduce 명령으로 추출된 로그인 및 로그아웃 이벤트가 원시 형식으로 표시됩니다.

```
# auditreduce -c lo | praudit -r

21,69,2,6222,0x0000,10.132.136.45,1287070091,698391050
36,26700,0,10,26700,10,856,50036632,82 0 10.132.136.45
39,0,0
47,1298
```

예 5-9 감사 레코드를 XML 형식으로 표시

이 예에서는 감사 레코드가 XML 형식으로 변환됩니다.

```
# praudit -x 20100827183214.20100827215318.logins > 20100827.logins.xml
```

마찬가지로, auditreduce 명령으로 필터링된 감사 레코드를 XML 형식으로 표시할 수 있습니다.

```
# auditreduce -c lo | praudit -x
<record version="2" event="screenlock - unlock" host="mach1"
iso8601="2010-10-14 08:28:11.698 -07:00">
<subject audit-uid="jdoe" uid="root" gid="staff" ruid="jdoe
rgid="staff" pid="856" sid="50036632" tid="82 0 mach1"/>
<return errval="success" retval="0"/>
<sequence seq-num="1298"/>
</record>
```

파일의 내용은 스크립트로 관련 정보를 추출하여 작업할 수 있습니다.

예 5-10 XML 형식의 감사 레코드를 브라우저에서 읽을 수 있도록 설정

xsltproc 도구를 사용해서 모든 브라우저에서 읽을 수 있도록 XML 파일의 레코드 형식을 조정할 수 있습니다. 이 도구는 스타일시트 정의를 파일 컨텐츠에 적용합니다. 형식이 조정된 컨텐츠를 별도의 파일로 저장하려면 다음을 입력합니다.

```
# auditreduce -c lo | praudit -x | xsltproc - > logins.html
```

브라우저에서 logins.html 컨텐츠는 다음과 비슷한 형식으로 표시됩니다.

```
                    Audit Trail Data

File: time: 2013-11-04 12:54:28.000 -08:00

Event: login - local
time: 2013-11-04 12:54:28.418 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jdoe uid: jdoe gid: staff ruid: jdoe rgid: staff
      pid: 1534 sid: 3583012893 tid: 0 0 host
RETURN errval: success retval: 0

Event: connect to RAD
time: 2013-11-04 12:54:52.029 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jdoe uid: jdoe gid: staff ruid: jdoe rgid: staff
      pid: 1835 sid: 3583012893 tid: 0 0 host
RETURN errval: success retval: 0

Event: role login
time: 2013-11-08 08:42:52.286 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jdoe uid: root gid: root ruid: root rgid: root
      pid: 4265 sid: 3583012893 tid: 0 0 host
RETURN errval: success retval: 0

Event: role logout
```

```
time: 2013-11-08 08:43:37.125 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jdoe uid: root gid: root ruid: root rgid: root
      pid: 4265 sid: 3583012893 tid: 0 0 host
RETURN errval: success retval: 0

Event: login - ssh
time: 2013-12-23 12:24:37.292 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jsmith uid: jsmith gid: staff ruid: jsmith rgid: staff
      pid: 2002 sid: 39351741 tid: 14632 202240 host.example.com
RETURN errval: success retval: 0

Event: role login
time: 2013-12-23 12:25:07.345 -08:00 vers: 2 mod: fe host: host
SUBJECT audit-uid: jsmith uid: root gid: root ruid: root rgid: root
      pid: 2023 sid: 39351741 tid: 14632 202240 host.example.com
RETURN errval: failure retval: Permission denied

Event: su
time: 2013-12-23 17:19:24.031 -08:00 vers: 2 mod: na host: host
RETURN errval: success retval: 0

Event: su logout
time: 2013-12-23 17:19:24.362 -08:00 vers: 2 mod: na host: host
RETURN errval: success retval: 0

Event: login - ssh
time: 2013-12-23 17:27:21.306 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jsmith uid: jsmith gid: staff ruid: jsmith rgid: staff
      pid: 2583 sid: 3401970889 tid: 13861 5632 host.example.com
RETURN errval: success retval: 0

Event: role login
time: 2013-12-23 17:27:28.361 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jsmith uid: root gid: root ruid: root rgid: root
      pid: 2593 sid: 3401970889 tid: 13861 5632 host.example.com
RETURN errval: success retval: 0

Event: role logout
time: 2013-12-23 17:30:39.029 -08:00 vers: 2 mod: host: host
SUBJECT audit-uid: jsmith uid: root gid: root ruid: root rgid: root
      pid: 2593 sid: 3401970889 tid: 13861 5632 host.example.com
RETURN errval: success retval: 0
```

Other events

예 5-11 `pfedit` 레코드만 표시

필터를 사용해서 감사 추적에서 특정 레코드만 추출하고 표시할 수 있습니다. 이 예제에서는 `pfedit` 명령 사용을 캡처하는 레코드가 필터링됩니다. 여기에서는 요약 파일이 `20130827183936.20130827232326.logins`라고 가정합니다. `pfedit` 명령을 사용하면 `AUE_admin_edit` 이벤트가 생성됩니다. 따라서 `pfedit` 레코드를 추출하려면 다음 명령을 실행합니다.

```
auditreduce -m AUE_admin_edit 20130827183936.20130827232326.logins | praudit
```

예 5-12 전체 감사 추적 인쇄

인쇄 명령에 파이프를 사용하면 전체 감사 추적이 프린터로 출력됩니다. 보안상 이유로 프린터는 제한적인 액세스 권한을 가집니다.

```
# auditreduce | praudit | lp -d example.protected.printer
```

예 5-13 특정 감사 파일 보기

이 예에서는 요약 로그인 파일이 터미널 창에서 검사됩니다.

```
# cd /var/audit/audit_summary/logins

# praudit 20100827183936.20100827232326.logins | more
```

예 5-14 스크립트를 사용하여 praudit 출력 처리

praudit 명령의 출력을 텍스트 행으로 처리하고자 할 수 있습니다. 예를 들어, auditreduce 명령에서 선택할 수 없는 레코드를 선택하고자 할 수 있습니다. 간단한 셸 스크립트를 사용하여 praudit 명령의 출력을 처리할 수 있습니다. 다음 샘플 스크립트는 하나의 감사 레코드를 한 행에 표시하고 사용자 지정 문자열을 검색한 다음 감사 파일을 원래 형식으로 반환합니다.

```
#!/bin/sh
#
## This script takes an argument of a user-specified string.
#  The sed command prefixes the header tokens with Control-A
#  The first tr command puts the audit tokens for one record
#  onto one line while preserving the line breaks as Control-A
#
praudit | sed -e '1,2d' -e '$s/^file.*$//' -e 's/^header/^aheader/' \\
| tr '\\012\\001' '\\002\\012' \\
| grep "$1" \\
```
Finds the user-specified string

```
| tr '\\002' '\\012'
```
Restores the original newline breaks

스크립트의 ^a는 ^과 a의 두 문자가 아닌 Ctrl-A입니다. 접두어는 텍스트로 나타날 수 있는 header 문자열에서 header 토큰을 구분합니다.

다음과 유사한 메시지는 praudit 명령을 사용할 수 있는 충분한 권한이 없음을 나타냅니다.

```
praudit: Can't assign 20090408164827.20090408171614.sys1.1 to stdin.
```

프로파일 셸에서 `praudit` 명령을 실행합니다. Audit Review 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

로컬 시스템에서 감사 레코드 관리

다음 작업 맵에서는 감사 레코드를 선택, 분석 및 관리하기 위한 절차를 안내합니다.

표 5-1 로컬 시스템에서 감사 레코드 관리 작업 맵

작업	설명	수행 방법
감사 레코드를 병합합니다.	여러 시스템의 감사 파일을 하나의 감사 추적으로 합칩니다.	감사 추적에서 감사 파일을 병합하는 방법 [93]
잘못 이름 지정된 감사 파일을 정리합니다.	감사 서비스에서 실수로 열어 둔 감사 파일에 종료 시간 기록을 제공합니다.	`not_terminated` 감사 파일을 정리하는 방법 [95]
감사 추적 오버플로우를 막습니다.	감사 파일 시스템이 가득 차지 않도록 막습니다.	"감사 추적 오버플로우 방지" [96]

▼ 감사 추적에서 감사 파일을 병합하는 방법

모든 감사 디렉토리의 감사 파일을 결합하면 전체 감사 추적의 내용을 분석할 수 있습니다.

참고 - 감사 추적의 시간 기록은 협정 세계시(UTC)로 되어 있으므로 의미를 가지려면 날짜와 시간을 현재 시간대로 변환해야 합니다. `auditreduce` 명령이 아닌 표준 파일 명령으로 이러한 파일을 조작할 때는 항상 이 사항을 염두에 두십시오.

시작하기 전에 Audit Review 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **병합된 감사 파일을 저장할 파일 시스템을 만듭니다.**

   디스크 공간 제한에 도달할 수 있는 가능성을 줄이기 위해서는 이 파일 시스템이 원래 파일을 저장하기 위해 *How to Create ZFS File Systems for Audit Files*에서 만든 파일 시스템과 감사 파일에 대한 ZFS 파일 시스템을 만드는 방법 [70]에 있어야 합니다.

2. **감사 추적의 감사 레코드를 병합합니다.**

   병합된 감사 파일을 저장할 디렉토리로 이동합니다. 이 디렉토리에서 감사 레코드를 이름이 지정된 접미어가 있는 파일로 병합합니다. 로컬 시스템에서 감사 추적의 모든 디렉토리가 병합되고 이 디렉토리에 배치됩니다.

```
# cd audit-storage-directory
# auditreduce -Uppercase-option -O suffix
```

auditreduce 명령의 대문자 옵션은 감사 추적의 파일을 조작합니다. 대문자 옵션에는 다음이 포함됩니다.

-A	감사 추적의 모든 파일을 선택합니다.
-C	완전한 파일만 선택합니다.
-M	특정 접미어가 있는 파일을 선택합니다. 접미어는 시스템 이름이거나 요약 파일에 대해 지정한 접미어일 수 있습니다.
-O	현재 디렉토리에서 *suffix* 접미어를 사용하여 시작 시간과 종료 시간 모두에 대해 14자의 시간 기록을 가진 감사 파일을 만듭니다.
-R *pathname*	대체 감사 루트 디렉토리인 *pathname*에서 감사 파일을 읽도록 지정합니다.
-S *server*	지정된 서버에서 감사 파일을 읽도록 지정합니다.

전체 옵션 목록은 auditreduce(1M) 매뉴얼 페이지를 참조하십시오.

예 5-15 감사 파일을 요약 파일로 복사

다음 예에서는 System Administrator 권한 프로파일이 지정된 관리자가 감사 추적의 모든 파일을 다른 파일 시스템의 병합된 파일로 복사합니다. /var/audit/storage 파일 시스템은 감사 루트 파일 시스템인 /var/audit 파일 시스템과 다른 별도의 디스크에 있습니다.

```
$ cd /var/audit/storage
$ auditreduce -A -O All
$ ls /var/audit/storage/*All
20100827183214.20100827215318.All
```

다음 예에서는 완전한 파일만 감사 추적에서 병합된 파일로 복사됩니다. 전체 경로는 -O 옵션의 값으로 지정됩니다. 경로의 마지막 구성 요소인 Complete는 접미어로 사용됩니다.

```
$ auditreduce -C -O /var/audit/storage/Complete

$ ls /var/audit/storage/*Complete
20100827183214.20100827214217.Complete
```

다음 예에서는 -D 옵션을 추가하여 원래 감사 파일이 삭제됩니다.

```
$ auditreduce -C -O daily_sys1.1 -D sys1.1

$ ls *sys1.1
20100827183214.20100827214217.daily_sys1.1
```

▼ not_terminated 감사 파일을 정리하는 방법

비정상적인 시스템 중단이 발생할 경우 감사 파일이 열린 상태에서 감사 서비스가 종료됩니다. 또는 파일 시스템에 액세스할 수 없게 되고 시스템이 강제로 새로운 파일 시스템으로 전환됩니다. 이러한 경우 감사 파일이 더 이상 감사 레코드에 사용되지 않더라도 감사 파일은 종료 시간 기록으로 not_terminated 문자열을 가집니다. auditreduce -O 명령을 사용하여 파일에 올바른 시간 기록을 지정합니다.

시작하기 전에 Audit Review 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

1. **감사 파일 시스템에 not_terminated 문자열이 있는 파일을 만들어진 순서대로 나열합니다.**

   # ls -R1t *audit-directory* ***/* | grep not_terminated**

   -R 하위 디렉토리의 파일을 나열합니다.

   -t 최근에서 가장 오래된 순서로 파일을 나열합니다.

   -1 파일을 한 열로 나열합니다.

2. **오래된 not_terminated 파일을 정리합니다.**

   auditreduce -O 명령에 오래된 파일의 이름을 지정합니다.

   # auditreduce -O *system-name old-not-terminated-file*

3. **오래된 not_terminated 파일을 제거합니다.**

   # rm *system-name old-not-terminated-file*

예 5-16 닫힌 not_terminated 감사 파일 정리

다음 예에서는 not_terminated 파일을 찾고 이름을 바꾼 다음 원본을 제거합니다.

```
ls -R1t */* | grep not_terminated
…/egret.1/20100908162220.not_terminated.egret
…/egret.1/20100827215359.not_terminated.egret

# cd */egret.1
# auditreduce -O egret 20100908162220.not_terminated.egret
# ls -1t
20100908162220.not_terminated.egret      Current audit file

20100827230920.20100830000909.egret      Cleaned-up audit file

20100827215359.not_terminated.egret      Input (old) audit file

# rm 20100827215359.not_terminated.egret
```

```
# ls -1t
20100908162220.not_terminated.egret      Current audit file

20100827230920.20100830000909.egret      Cleaned-up audit file
```

새 파일의 시작 시간 기록에는 not_terminated 파일에서 첫 감사 이벤트의 시간이 반영됩니다. 종료 시간 기록에는 파일에서 마지막 감사 이벤트의 시간이 반영됩니다.

감사 추적 오버플로우 방지

보안 정책에 따라 모든 감사 데이터를 저장해야 할 경우에는 다음 방법에 따라 감사 레코드 손실을 방지합니다.

참고 - root 역할을 맡아야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

- audit_binfile 플러그인에서 최소 사용 가능 크기를 설정합니다.

  p_minfree 속성을 사용합니다.

  audit_warn 전자 메일 별칭은 디스크 공간이 최소 사용 가능 크기까지 채워지면 경고를 보냅니다. 예 4-7. "경고에 대한 소프트 제한 설정"을 참조하십시오.
- 감사 파일을 정기적으로 아카이브하도록 일정을 설정합니다.

  파일을 오프라인 매체에 백업하여 감사 파일을 아카이브합니다. 또한 아카이브 파일 시스템으로 파일을 이동할 수도 있습니다.

  syslog 유틸리티를 사용하여 텍스트 감사 로그를 수집하는 경우 텍스트 로그를 아카이브합니다. 자세한 내용은 logadm(1M) 매뉴얼 페이지를 참조하십시오.
- 아카이브된 감사 파일을 감사 파일 시스템에서 삭제하도록 일정을 설정합니다.
- 보조 정보를 저장하고 보관합니다.

  감사 추적과 함께 감사 레코드를 해석하는 데 필요한 정보를 아카이브합니다. 최소한 passwd, group 및 hosts 파일을 저장합니다. 또한 audit_event 및 audit_class 파일을 아카이브할 수 있습니다.
- 어떤 감사 파일이 아카이브되었는지 기록합니다.
- 아카이브된 매체를 적절히 보관합니다.
- ZFS 압축을 사용으로 설정하여 필요한 파일 시스템 용량을 줄입니다.

  감사 파일 전용 ZFS 파일 시스템에서 압축을 사용하면 파일이 크게 줄어듭니다. 예는 전용 파일 시스템에서 감사 파일을 압축하는 방법 [60]을 참조하십시오.

  또한 "Oracle Solaris 11.2의 ZFS 파일 시스템 관리 "의 "ZFS 압축, 중복 제거 및 암호화 등록 정보 간의 상호 작용"을 참조하십시오.
- 요약 파일을 만들어 저장하는 감사 데이터의 양을 줄입니다.

  auditreduce 명령에 대한 옵션을 사용하여 감사 추적에서 요약 파일을 추출할 수 있습니다. 요약 파일에는 지정된 유형의 감사 이벤트에 대한 레코드만 포함됩니다. 요약 파일을

추출하려면 예 5-4. “감사 파일 결합 및 줄이기” 및 예 5-6. “선택한 레코드를 단일 파일로 병합”을 참조하십시오.

♦ ♦ ♦ 6 장

6 감사 서비스 문제 분석 및 해결

이 장에서는 감사 관련 문제를 해결하는 데 도움이 되는 절차를 제공합니다. 또한 다음 장에서는 기타 감사 관리 작업에 대해 설명합니다.

- 3장. 감사 서비스 관리
- 4장. 시스템 작업 모니터링
- 5장. 감사 데이터 작업

감사 서비스에 대한 개요는 1장. Oracle Solaris의 감사 정보를 참조하십시오. 계획 제안은 2장. 감사 계획을 참조하십시오. 참조 정보는 7장. 감사 참조를 참조하십시오.

감사 서비스 문제 해결

이 절에서는 다양한 감사 오류 메시지, 기본 설정 및 감사 문제 디버그를 위해 다른 도구에서 제공하는 감사에 대해 다룹니다.

일반적으로는 감사 서비스에서 발생한 오류를 알리기 위해 여러 통지가 전송됩니다. 감사 서비스에 문제가 있다고 생각될 경우 전자 메일 및 로그 파일을 검토하십시오.

- `audit_warn` 별칭으로 전송된 전자 메일을 읽습니다.

  `audit_warn` 스크립트는 경보 메시지를 `audit_warn` 전자 메일 별칭으로 보냅니다. 올바르게 구성된 별칭이 없을 경우 메시지가 `root` 별칭으로 보내집니다.
- 감사 서비스에 대한 로그 파일을 검토합니다.

  `svcs -s auditd` 명령의 출력은 감사 서비스에서 생성하는 감사 로그에 대한 전체 경로를 나열합니다.
- 시스템 로그 파일을 검토합니다.

  `audit_warn` 스크립트는 `daemon.alert` 메시지를 `/var/log/syslog` 파일에 씁니다.

  `/var/adm/messages` 파일에는 정보가 포함되어 있을 수 있습니다.

문제를 찾아 수정한 다음 감사 서비스를 사용으로 설정하거나 다시 시작합니다.

```
# audit -s
```

다음 절에서는 발생 가능한 문제 사례 및 해결 단계에 대해 설명합니다.

참고 - 문제 해결 작업을 수행하기 전에 적합한 권한 부여가 있는지 확인합니다. 예를 들어, 감사를 구성하려면 Audit Configuration 권한 프로파일이 지정된 관리자여야 합니다. 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "지정된 관리 권한 사용"을 참조하십시오.

감사 레코드가 기록되지 않음

감사 기능은 기본적으로 사용으로 설정됩니다. 감사가 사용 안함으로 설정되지 않았는데 감사 레코드가 활성 플러그인으로 전송되지 않는다고 여겨질 경우 문제 원인은 이 절에 설명된 다음 요소 중 하나 또는 이러한 요소의 조합으로 인한 것일 수 있습니다. 파일 시스템을 수정하려면 사용자에게 `solaris.admin.edit/`*path-to-system-file* 권한 부여가 지정되어 있어야 합니다. 기본적으로 root 역할에는 이 권한 부여가 있습니다.

감사 서비스가 실행되지 않음

감사가 실행 중인지 확인하려면 다음 방법 중 하나를 사용합니다.

- 현재 감사 조건을 확인합니다.

  다음 출력은 감사가 실행 중이 아님을 나타냅니다.

  ```
  # auditconfig -getcond
  audit condition = noaudit
  ```

  다음 출력은 감사가 실행 중임을 나타냅니다.

  ```
  # auditconfig -getcond
  audit condition = auditing
  ```

- 감사 서비스가 실행 중인지 확인합니다.

  다음 출력은 감사가 실행 중이 아님을 나타냅니다.

  ```
  # svcs -x auditd

  svc:/system/auditd:default (Solaris audit daemon)
  State: disabled since Sun Oct 10 10:10:10 2010
  Reason: Disabled by an administrator.
  See: http://support.oracle.com/msg/SMF-8000-05
  See: auditd(1M)
  See: audit(1M)
  See: auditconfig(1M)
  See: audit_flags(5)
  See: audit_binfile(5)
  See: audit_syslog(5)
  See: audit_remote(5)
  ```

```
See: /var/svc/log/system-auditd:default.log
Impact: This service is not running.
```

다음 출력은 감사 서비스가 실행 중임을 나타냅니다.

```
# svcs auditd
STATE          STIME    FMRI
online         10:10:10 svc:/system/auditd:default
```

감사 서비스가 실행 중이 아닌 경우 사용으로 설정합니다. 절차는 "감사 서비스를 사용/사용 안함으로 설정" [39]을 참조하십시오.

활성 상태의 감사 플러그인이 없음

다음 명령을 사용해서 플러그인이 활성 상태인지 확인합니다. 감사 서비스가 작동하려면 하나 이상의 플러그인이 활성 상태여야 합니다.

```
# audit -v
audit: no active plugin found
```

활성화된 플러그인이 없는 경우 활성화합니다.

```
# auditconfig -setplugin audit_binfile active
# audit -v
configuration ok
```

감사 클래스가 정의되지 않음

정의되지 않은 감사 클래스를 사용하려고 시도하는 중일 수 있습니다. `pf` 클래스를 만드는 자세한 내용은 감사 클래스를 추가하는 방법 [51]을 참조하십시오.

예를 들어, 다음 플래그 목록은 Oracle Solaris 소프트웨어에서 제공하지 않은 `pf` 클래스를 포함합니다.

```
# auditconfig -getflags
active user default audit flags = pf,lo(0x0100000000000000,00x0100000000001000)
configured user default audit flags = pf,lo(0x0100000000000000,00x0100000000001000)
```

클래스를 정의하지 않으려면 `auditconfig -setflags` 명령을 유효한 값과 함께 실행하여 현재 플래그를 재설정합니다. 그렇지 않으면 클래스를 정의할 때 다음을 확인합니다.

- 감사 클래스가 `audit_class` 파일에 정의되어 있습니다.

  ```
  # grep pf /etc/security/audit_class
      Verify class exists


  0x0100000000000000:pf:profile
  ```

- 마스크가 고유합니다. 고유하지 않으면 마스크를 바꿉니다.

```
# grep 0x0100000000000000 /etc/security/audit_class
     Ensure mask is unique

0x0100000000000000:pf:profile
```

감사 클래스에 이벤트가 지정되지 않음

사용 중인 사용자 정의된 클래스는 정의되었다 해도 클래스에 지정된 이벤트가 없을 수 있습니다.

이벤트가 사용자 정의된 클래스에 지정되었는지 여부를 확인하려면 다음 방법 중 하나를 사용합니다.

```
# auditconfig -lsevent | egrep " pf|,pf|pf,"
AUE_PFEXEC      116 pf execve(2) with pfexec enabled

# auditrecord -c pf
     List of audit events assigned to pf class
```

이벤트가 클래스에 지정되지 않은 경우 적당한 이벤트를 이 클래스에 지정합니다.

감사 레코드 볼륨이 큼

사이트에서 감사해야 하는 이벤트를 확인한 후에는 다음 제안에 따라 필요한 정보만 포함된 감사 파일을 만듭니다. 플래그를 사용자, 역할, 권한 프로파일에 지정하려면 `root` 역할을 맡아야 합니다.

- 특히, 이벤트 및 감사 토큰을 감사 추적에 추가하지 마십시오. 다음 정책은 감사 추적의 크기를 늘립니다.

`arge`	환경 변수를 `execv` 감사 이벤트에 추가합니다. `execv` 이벤트 감사는 비용이 많이 들 수 있지만 감사 레코드에 변수를 추가하는 것은 비용이 많이 들지 않습니다.
argv	명령 매개변수를 `execv` 감사 이벤트에 추가합니다. 감사 레코드에 대한 명령 매개변수 추가는 비용이 많이 들지 않습니다.
group	선택적 `newgroups` 토큰이 포함된 감사 이벤트에 그룹 토큰을 추가합니다.
path	선택적 `path` 토큰이 포함된 감사 이벤트에 `path` 토큰을 추가합니다.
public	파일 이벤트가 감사되는 경우 감사 가능한 이벤트가 public object(공용 객체)에 발생할 때마다 감사 추적에 이벤트를 추가합

니다. 파일 클래스에는 fa, fc, fd, fm, fr, fw 및 cl이 포함됩니다. 공용 파일에 대한 정의는 "감사 용어 및 개념" [10]을 참조하십시오.

seq — 모든 감사 이벤트에 시퀀스 토큰을 추가합니다.

trail — 모든 감사 이벤트에 트레일러 토큰을 추가합니다.

windata_down — Trusted Extensions로 구성된 시스템에서 레이블이 있는 창의 정보가 다운그레이드될 때 이벤트를 추가합니다.

windata_up — Trusted Extensions로 구성된 시스템에서 레이블이 있는 창의 정보가 업그레이드될 때 이벤트를 추가합니다.

zonename — 모든 감사 이벤트에 영역 이름을 추가합니다. 전역 영역이 유일하게 구성된 영역인 경우 모든 감사 이벤트에 zone, global 문자열을 추가합니다.

다음 감사 레코드는 ls 명령의 사용을 보여줍니다. ex 클래스가 감사되고 기본 정책을 사용 중입니다.

```
header,129,2,AUE_EXECVE,,mach1,2010-10-14 11:39:22.480 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
subject,jdoe,root,root,root,root,2404,50036632,82 0 mach1
return,success,0
```

다음은 모든 정책이 설정되었을 때 나타나는 동일한 레코드입니다.

```
header,1578,2,AUE_EXECVE,,mach1,2010-10-14 11:45:46.658 -07:00
path,/usr/bin/ls
attribute,100555,root,bin,21,320271,18446744073709551615
exec_args,2,ls,/etc/security
exec_env,49,MANPATH=/usr/share/man,USER=jdoe,GDM_KEYBOARD_LAYOUT=us,EDITOR=gedit,
LANG=en_US.UTF-8,GDM_LANG=en_US.UTF-8,PS1=#,GDMSESSION=gnome,SESSIONTYPE=1,SHLVL=2,
HOME=/home/jdoe,LOGNAME=jdoe,G_FILENAME_ENCODING=@locale,UTF-8, PRINTER=example-dbl,
...
path,/lib/ld.so.1
attribute,100755,root,bin,21,393073,18446744073709551615
subject,jdoe,root,root,root,root,2424,50036632,82 0 mach1
group,root,other,bin,sys,adm,uucp,mail,tty,lp,nuucp,daemon
return,success,0
zone,global
sequence,197
trailer,1578
```

- audit_syslog 플러그인을 사용하여 일부 감사 이벤트를 syslog로 보냅니다.

이러한 감사 이벤트를 audit_binfile 또는 audit_remote 플러그인으로 보내지 마십시오. 이 전략은 syslog 로그로 보내는 감사 이벤트의 이진 레코드를 보관할 필요가 없을 경우에만 유효합니다.

- 더 적은 시스템 전역 감사 플래그를 설정하고 개별 사용자를 감사합니다.

  시스템 전역으로 감사되는 감사 클래스의 수를 줄여 모든 사용자에 대한 감사의 양을 줄입니다.

  roleadd, rolemod, useradd 및 usermod 명령에 audit_flags 키워드를 사용하여 특정 사용자 및 역할에 대한 이벤트를 감사합니다. 예는 예 4-11. “syslog 출력에 대한 감사 클래스 지정” 및 usermod(1M) 매뉴얼 페이지를 참조하십시오.

  profiles 명령의 always_audit 및 never_audit 등록 정보를 사용하여 특정 권한 프로파일에 대한 이벤트를 감사합니다. 자세한 내용은 profiles(1) 매뉴얼 페이지를 참조하십시오.

참고 - 다른 보안 속성과 마찬가지로 감사 플래그는 검색 순서의 영향을 받습니다. 자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안 ”의 “지정된 권한 검색 순서”를 참조하십시오.

- 고유의 사용자 정의된 감사 클래스를 만듭니다.

  해당 사이트에서 감사 클래스를 만들 수 있습니다. 모니터해야 하는 감사 이벤트만 이러한 감사 클래스에 추가합니다. 절차는 감사 클래스를 추가하는 방법 [51]을 참조하십시오.

참고 - 감사 구성 파일 수정의 영향에 대한 자세한 내용은 “감사 구성 파일 및 패키징” [110]을 참조하십시오.

이진 감사 파일 크기 무제한 증가

Audit Review 권한 프로파일이 지정된 관리자는 아카이브 및 검색을 쉽게 수행할 수 있도록 이진 파일의 크기를 제한할 수 있습니다. 또한 이 절에 설명된 옵션 중 하나를 사용해서 원본 파일로부터 비슷한 이진 파일을 만들 수도 있습니다.

- p_fsize 속성을 사용하여 개별 이진 감사 파일의 크기를 제한합니다.

  p_fsize 속성에 대한 설명은 audit_binfile(5) 매뉴얼 페이지의 OBJECT ATTRIBUTES 섹션을 참조하십시오.

  예제는 예 4-3. “audit_binfile 플러그인에 대한 파일 크기 제한”를 참조하십시오.
- 추가 분석을 위해 auditreduce 명령을 사용하여 레코드를 선택하고 이러한 레코드를 더 작은 파일에 씁니다.

  auditreduce -*lowercase* 옵션은 특정 레코드를 찾습니다.

`auditreduce -`*Uppercase* 옵션은 선택 항목을 파일에 씁니다. 자세한 내용은 `auditreduce`(1M) 매뉴얼 페이지를 참조하십시오. 또한 "감사 추적 데이터 표시" [85]를 참조하십시오.

다른 운영 체제에서의 로그인이 감사되지 않음

Oracle Solaris OS는 소스와 상관없이 모든 로그인을 감사할 수 있습니다. 로그인이 감사되지 않을 경우, 지정 가능한 이벤트 및 지정 불가능한 이벤트 모두에 대해 `lo` 클래스가 설정되지 않았을 수 있습니다. 이 클래스는 로그인, 로그아웃 및 화면 잠금을 감사합니다. 이러한 클래스는 기본적으로 감사됩니다.

참고 - `ssh` 로그인을 감사하려면 시스템에서 Oracle Solaris의 `ssh` 데몬을 실행하고 있어야 합니다. 이 데몬은 Oracle Solaris 시스템에서 감사 서비스에 대해 수정됩니다. 자세한 내용은 "Oracle Solaris 11.2의 보안 셸 액세스 관리 "의 "보안 셸 및 OpenSSH 프로젝트"를 참조하십시오.

예 6-1 로그인이 감사되는지 확인

이 예제에서 처음 두 개의 명령 출력은 지정 가능한 이벤트 및 지정 불가능한 이벤트에 대해 `lo` 클래스가 설정되지 않았음을 보여줍니다. 그런 후 마지막 두 개의 명령은 로그인 이벤트의 감사를 사용으로 설정하도록 `lo` 클래스를 설정합니다.

```
# auditconfig -getflags
active user default audit flags = as,st(0x20800,0x20800)
configured user default audit flags = as,st(0x20800,0x20800)

# auditconfig -getnaflags
active non-attributable audit flags = na(0x400,0x400)
configured non-attributable audit flags = na(0x400,0x400)

# auditconfig -setflags lo,as,st
user default audit flags = as,lo,st(0x21800,0x21800)

# auditconfig -setnaflags lo,na
non-attributable audit flags = lo,na(0x1400,0x1400)
```

♦ ♦ ♦ 7 장

7 감사 참조

이 장에서는 중요한 감사 구성 요소에 대해 설명하고 다음 항목을 다룹니다.

감사 개요는 1장. Oracle Solaris의 감사 정보를 참조하십시오. 계획 제안은 2장. 감사 계획을 참조하십시오. 사용자 사이트에서 감사를 구성하는 절차는 다음 장을 참조하십시오.

- 3장. 감사 서비스 관리
- 4장. 시스템 작업 모니터링
- 5장. 감사 데이터 작업
- 6장. 감사 서비스 문제 분석 및 해결

Audit Service

감사 서비스 `auditd`는 기본적으로 사용으로 설정됩니다. 서비스를 사용으로 설정하거나, 새로 고치거나, 사용 안함으로 설정하는 방법은 “감사 서비스를 사용/사용 안함으로 설정” [39]을 참조하십시오.

사용자 구성이 없으면 다음 기본값이 설정됩니다.

- 모든 로그인 이벤트가 감사됩니다.

  성공 및 실패 로그인 시도가 모두 감사됩니다.
- 모든 사용자가 로그인 및 로그아웃 이벤트(역할 맡기 및 화면 잠금 포함)에 대해 감사됩니다.
- `audit_binfile` 플러그인이 활성화됩니다. `/var/audit` 디렉토리가 감사 레코드를 저장하고, 감사 파일의 크기는 무제한이며, 대기열 크기는 레코드 100개입니다.
- `cnt` 정책이 설정됩니다.

  감사 레코드가 사용 가능한 디스크 공간을 채우면 시스템에서 삭제된 감사 레코드의 수를 추적합니다. 사용 가능한 디스크 공간의 1%가 남으면 경고가 발생합니다.
- 다음 감사 대기열 제어가 설정됩니다.
  - 레코드 잠금을 생성하기 전 감사 대기열의 최대 레코드 수는 100입니다.
  - 차단된 감사 프로세스가 차단 해제되기 전 감사 대기열의 최대 레코드 수는 10입니다.
  - 감사 대기열에 대한 버퍼 크기는 8192바이트입니다.
  - 감사 추적에 감사 레코드 쓰기 간격은 20초입니다.

기본값을 표시하려면 "감사 서비스 기본값 표시" [38]를 참조하십시오.

감사 서비스를 사용하여 임시(또는 활성) 값을 설정할 수 있습니다. 이러한 값은 구성된(또는 등록 정보) 값과 다를 수 있습니다.

- 임시 값은 감사 서비스를 새로 고치거나 다시 시작할 때 복원되지 않습니다.

  감사 정책 및 감사 대기열 제어는 임시 값을 사용할 수 있습니다. 감사 플래그에는 임시 값이 없습니다.
- 구성된 값은 서비스의 등록 정보 값으로 저장되므로 감사 서비스를 새로 고치거나 다시 시작할 때 복원됩니다.

권한 프로파일은 감사 서비스를 관리할 수 있는 사용자를 제어합니다. 자세한 내용은 "감사 관리를 위한 권한 프로파일" [110]을 참조하십시오.

기본적으로 모든 영역은 동일하게 감사됩니다. "감사 및 Oracle Solaris 영역" [110]을 참조하십시오.

감사 서비스 매뉴얼 페이지

다음 표에서는 감사 서비스에 대한 주요 관리 매뉴얼 페이지를 요약합니다.

매뉴얼 페이지	요약
audit(1M)	감사 서비스의 작업을 제어하는 명령입니다. `audit -n`은 `audit_binfile` 플러그인에 대한 새로운 감사 파일을 시작합니다. `audit -s`는 감사를 사용으로 설정하고 새로 고칩니다.

매뉴얼 페이지	요약
	`audit -t`는 감사를 사용 안함으로 설정합니다. `audit -v`는 적어도 하나의 플러그인이 활성화되었는지 확인합니다.
`audit_binfile`(5)	감사 레코드를 이진 파일로 보내는 기본 감사 플러그인입니다. "감사 플러그인" [112]도 참조하십시오.
`audit_remote`(5)	감사 레코드를 원격 수신자에게 보내는 감사 플러그인입니다.
`audit_syslog`(5)	감사 레코드의 텍스트 요약을 `syslog` 유틸리티로 보내는 감사 플러그인입니다.
`audit_class`(4)	감사 클래스의 정의를 포함하는 파일입니다. 8개 상위 순서 비트는 고객이 새 감사 클래스를 만드는 데 사용할 수 있습니다. 시스템 업그레이드 시 이 파일 수정의 효과에 대한 자세한 내용은 감사 클래스를 추가하는 방법 [51]을 참조하십시오.
`audit_event`(4)	감사 이벤트의 정의를 포함하고 이벤트를 감사 클래스에 매핑하는 파일입니다. 매핑은 수정할 수 있습니다. 시스템 업그레이드 시 이 파일 수정의 효과에 대한 자세한 내용은 감사 이벤트의 클래스 멤버쉽을 변경하는 방법 [52]을 참조하십시오.
`audit_flags`(5)	감사 클래스 사전 선택의 구문, 실패한 이벤트만 또는 성공한 이벤트만 선택하기 위한 접두어 및 기존 사전 선택을 수정하는 접두어를 설명합니다.
`audit.log`(4)	이진 감사 파일의 이름 지정, 파일의 내부 구조 및 모든 감사 토큰의 구조를 설명합니다.
`audit_warn`(1M)	감사 서비스에서 감사 레코드를 작성할 때 비정상적인 조건이 발견될 경우 전자 메일 별칭에 이를 알려주는 스크립트입니다. 사이트에 대해 이 스크립트를 사용자 정의하여 수동 개입이 필요할 수 있는 조건을 경고하거나 이러한 조건을 자동으로 처리하는 방법을 지정할 수 있습니다.
`auditconfig`(1M)	감사 구성 매개변수를 검색하는 명령입니다. 검색 및 설정할 수 있는 매개변수 목록을 표시하려면 이 `auditconfig`를 옵션 없이 실행합니다.
`auditrecord`(1M)	`/etc/security/audit_event` 파일의 감사 이벤트 정의를 표시하는 명령입니다. 샘플 출력은 "감사 레코드 정의 표시" [85]를 참조하십시오.
`auditreduce`(1M)	이진 형식으로 저장된 감사 레코드를 사후 선택하고 병합하는 명령입니다. 명령은 하나 이상의 입력 감사 파일에서 감사 레코드를 병합할 수 있습니다. 레코드는 이진 형식으로 유지됩니다. 대문자 옵션은 파일 선택에 영향을 미칩니다. 소문자 옵션은 레코드 선택에 영향을 미칩니다.
`auditstat`(1M)	커널 감사 통계를 표시하는 명령입니다. 예를 들어, 명령은 커널 감사 대기열의 레코드 수, 삭제된 레코드 수 및 사용자 프로세스가 시스템 호출 결과로 커널에서 생성한 감사 레코드 수를 표시할 수 있습니다.
`praudit`(1M)	표준 입력에서 이진 형식의 감사 레코드를 읽고 사전 선택 가능한 형식으로 레코드를 표시하는 명령입니다. 입력은 `auditreduce` 명령 또는 단일 감사 파일이나 감사 파일 목록에서 파이프할 수 있습니다. 또한 입력은 현재 감사 파일에 대해 `tail -0f` 명령으로 생성할 수 있습니다. 샘플 출력은 "이진 감사 파일의 컨텐츠 보기" [89]을 참조하십시오.
`syslog.conf`(4)	`audit_syslog` 플러그인에 대해 감사 레코드의 텍스트 요약을 `syslog` 유틸리티로 보내도록 구성된 파일입니다.

감사 관리를 위한 권한 프로파일

Oracle Solaris는 감사 서비스 구성, 서비스 사용/사용 안함으로 설정 및 감사 추적 분석을 위한 권한 프로파일을 제공합니다. 감사 구성 파일을 편집하려면 `root` 권한이 있어야 합니다.

- **Audit Configuration** - 관리자가 감사 서비스의 매개변수를 구성하고 `auditconfig` 명령을 실행할 수 있도록 합니다.
- **Audit Control** - 관리자가 감사 서비스를 시작, 새로 고침 및 사용 안함으로 설정하고 `audit` 명령을 실행하여 서비스를 시작, 새로 고침 또는 중지할 수 있도록 합니다.
- **Audit Review** - 관리자가 감사 레코드를 분석할 수 있도록 합니다. 이 권한 프로파일은 `praudit` 및 `auditreduce` 명령으로 감사 레코드를 읽을 수 있는 권한을 부여합니다. 또한 이 관리자는 `auditstat` 명령을 실행할 수 있습니다.
- **System Administrator** - Audit Review 권한 프로파일을 포함합니다. System Administrator 권한 프로파일을 가진 관리자는 감사 레코드를 분석할 수 있습니다.

감사 서비스 처리 역할을 구성하려면 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "역할 만들기"를 참조하십시오.

감사 및 Oracle Solaris 영역

비전역 영역은 전역 영역과 동일하게 감사하거나 고유의 플래그, 저장소 및 감사 정책을 설정할 수 있습니다.

모든 영역이 동일하게 감사되는 경우 전역 영역의 `audit_class` 및 `audit_event` 파일이 모든 영역에서 감사를 위한 클래스-이벤트 매핑을 제공합니다. `+zonename` 정책 옵션은 영역 이름으로 레코드를 사후 선택하는 데 유용합니다.

영역은 개별적으로 감사할 수 있습니다. 정책 옵션 `perzone`이 전역 영역에서 설정된 경우 각 비전역 영역은 고유의 감사 서비스를 실행하고, 고유의 감사 대기열을 처리하며, 해당 감사 레코드의 내용과 위치를 지정합니다. 또한 비전역 영역은 대부분의 감사 정책 옵션을 설정할 수 있습니다. 전체 시스템에 영향을 미치는 정책은 설정할 수 없으므로 비전역 영역은 `ahlt` 또는 `perzone` 정책을 설정할 수 없습니다. 자세한 내용은 "Oracle Solaris 영역이 있는 시스템 감사" [23] 및 "영역에서 감사 계획" [26]을 참조하십시오.

영역에 대해 알아보려면 "Oracle Solaris 영역 소개 "를 참조하십시오.

감사 구성 파일 및 패키징

Oracle Solaris의 감사 구성 파일은 패키지에 `preserve=renamenew` 패키지 속성으로 표시됩니다. 이 속성은 업데이트 간에 파일에 적용하는 모든 수정 사항을 보존합니다. `preserve` 값의 영향에 대한 자세한 내용은 `pkg`(5) 매뉴얼 페이지를 참조하십시오.

이러한 구성 파일은 overlay=allow 패키지 속성으로도 표시됩니다. 이 속성을 사용하면 이러한 파일을 포함하는 사용자 고유의 패키지를 만들고 Oracle Solaris 파일을 사용자 패키지의 파일로 바꿀 수 있습니다. 패키지에서 overlay 속성을 true로 설정하면 pkg 하위 명령(예: verify, fix, revert 등)이 패키지에 대한 결과를 반환합니다. 자세한 내용은 pkg(1) 및 pkg(5) 매뉴얼 페이지를 참조하십시오.

감사 클래스

Oracle Solaris는 많은 수의 감사 이벤트에 대한 편리한 컨테이너로 감사 클래스를 정의합니다.

감사 클래스를 재구성하고 새 감사 클래스를 만들 수 있습니다. 감사 클래스 이름은 최대 8자까지 가능합니다. 클래스 설명은 72자로 제한됩니다. 숫자 및 영숫자 이외의 문자가 허용됩니다. 자세한 내용은 audit_class(4) 매뉴얼 페이지 및 감사 클래스를 추가하는 방법 [51]을 참조하십시오.

주의 - all 클래스는 많은 양의 데이터를 생성하고 디스크를 빠르게 채울 수 있습니다. 모든 작업을 감사해야 하는 특별한 이유가 있을 경우에만 all 클래스를 사용하십시오.

감사 클래스 구문

감사 클래스의 이벤트는 성공, 실패 또는 둘 다에 대해 감사할 수 있습니다.

- 접두어가 없으면 이벤트 클래스가 성공 및 실패에 대해 감사됩니다.
- 더하기(+) 접두어가 있으면 이벤트 클래스가 성공에 대해서만 감사됩니다.
- 빼기(-) 접두어가 있으면 이벤트 클래스가 실패에 대해서만 감사됩니다.
- 현재 사전 선택을 수정하려면 접두어 또는 감사 플래그 앞에 캐럿(^)을 추가합니다. 예를 들면 다음과 같습니다.
  - ot가 시스템에 대해 사전 선택되고 사용자의 사전 선택이 ^ot인 경우 해당 사용자는 other 클래스의 이벤트에 대해 감사되지 않습니다.
  - +ot가 시스템에 대해 사전 선택되고 사용자의 사전 선택이 ^+ot인 경우 해당 사용자는 other 클래스의 성공 이벤트에 대해 감사되지 않습니다.
  - -ot가 시스템에 대해 사전 선택되고 사용자의 사전 선택이 ^-ot인 경우 해당 사용자는 other 클래스의 실패 이벤트에 대해 감사되지 않습니다.

감사 클래스 사전 선택 구문을 검토하려면 audit_flags(5) 매뉴얼 페이지를 참조하십시오.

감사 클래스 및 해당 접두어는 다음 명령에서 지정할 수 있습니다.

- auditconfig 명령 옵션 -setflags 및 -setnaflags에 대한 인수로 지정합니다.

- audit_syslog 플러그인의 p_flags 속성에 대한 값으로 지정합니다. auditconfig -setplugin audit_syslog active 명령에 대한 옵션으로 속성을 지정합니다.
- useradd, usermod, roleadd 및 rolemod 명령의 -K audit_flags=*always-audit-flags*:*never-audit-flags* 옵션에 대한 값으로 지정합니다.
- profiles 명령의 -always_audit 및 -never_audit 등록 정보에 대한 값으로 지정합니다.

감사 플러그인

감사 플러그인은 감사 대기열의 감사 레코드를 어떻게 처리할지 지정합니다. 감사 플러그인은 audit_binfile, audit_remote 및 audit_syslog 이름을 사용하여 auditconfig -setplugin 명령에 대한 인수로 지정됩니다. 플러그인은 다음 속성으로 추가 지정할 수 있습니다.

- audit_binfile 플러그인

  p_dir 속성 - 이진 데이터를 보낼 위치

  p_minfree 속성 - 관리자에게 경고를 표시하기 전까지 디스크에 남은 최소 공간

  p_fsize 속성 - 감사 파일의 최대 크기
- audit_remote 플러그인

  p_hosts 속성 - 이진 감사 데이터를 보낼 원격 인증된 감사 서버

  p_retries 속성 - 원격 인증된 감사 서버에 연결하기 위한 시도 횟수

  p_timeout 속성 - 원격 인증된 감사 서버에 연결하기 위한 시도 간격(초)
- audit_syslog 플러그인

  p_flags 속성 - syslog로 전송할 감사 레코드의 텍스트 요약 선택
- (모든 플러그인에 대상) 플러그인에 대해 대기되는 최대 감사 레코드 수 – qsize 속성

audit_binfile(5), audit_remote(5), audit_syslog(5) 및 auditconfig(1M) 매뉴얼 페이지를 참조하십시오.

감사 원격 서버

ARS(감사 원격 서버)는 감사되는 시스템에서 보안 링크를 통해 감사 레코드를 수신하고 레코드를 저장합니다.

수신을 위해 다음과 같은 구성이 필요합니다.

- 특정 감사 주체 및 GSS-API 방식의 Kerberos 영역
- 최소한 하나 이상의 연결 그룹이 구성되고 활성 상태인 ARS
- 연결 그룹에 최소한 하나 이상의 감사되는 시스템이 있고, 구성되어 활성 상태인 audit_remote 플러그인

연결 그룹은 ARS의 group 등록 정보에 지정됩니다. 파일 관리를 위해 group은 감사 파일의 크기를 제한하고 최소 여유 공간을 지정할 수 있습니다. 다른 연결 그룹을 지정하는 주요 이유는 예 4-9. "감사 레코드를 동일 ARS의 다른 파일 위치로 스트리밍"에 표시된 대로 ARS에서 다른 저장소 위치를 지정하기 위한 것입니다.

ARS에 대한 자세한 내용은 ars(5) 매뉴얼 페이지를 참조하십시오. ARS 구성 정보는 auditconfig(1M) 매뉴얼 페이지에서 -setremote 옵션을 참조하십시오.

감사되는 시스템을 구성하려면 audit_remote(5) 매뉴얼 페이지 및 auditconfig(1M) 매뉴얼 페이지의 -setplugin 옵션을 참조하십시오.

감사 정책

감사 정책은 추가 정보가 감사 추적에 추가되는지 여부를 결정합니다.

arge, argv, group, path, seq, trail, windata_down, windata_up 및 zonename 정책은 감사 레코드에 토큰을 추가합니다. windata_down 및 windata_up 정책은 Oracle Solaris의 Trusted Extensions 기능에서 사용됩니다. 자세한 내용은 "Trusted Extensions 구성 및 관리 "의 22장, "Trusted Extensions와 감사"를 참조하십시오.

나머지 정책은 토큰을 추가하지 않습니다. public 정책은 공용 파일의 감사를 제한합니다. perzone 정책은 비전역 영역에 대해 별도의 감사 대기열을 설정합니다. ahlt 및 cnt 정책은 감사 레코드를 전달할 수 없을 경우 어떻게 되는지 결정합니다. 자세한 내용은 "비동기 및 동기 이벤트에 대한 감사 정책" [113]을 참조하십시오.

서로 다른 감사 정책 옵션의 효과는 "감사 정책 이해" [32]를 참조하십시오. 감사 정책 옵션에 대한 설명은 auditconfig(1M) 매뉴얼 페이지의 -setpolicy 옵션을 참조하십시오. 사용 가능한 정책 옵션 목록을 표시하려면 auditconfig -lspolicy 명령을 실행합니다. 현재 정책을 표시하려면 auditconfig -getpolicy 명령을 실행합니다.

비동기 및 동기 이벤트에 대한 감사 정책

ahlt 정책과 cnt 정책은 함께 감사 대기열이 가득 차서 더 이상 이벤트를 수신할 수 없을 때 어떻게 되는지 제어합니다.

참고 - 적어도 하나의 플러그인에 대한 대기열이 감사 레코드를 수신할 수 있으면 cnt 또는 ahlt 정책이 트리거되지 않습니다.

---

cnt 및 ahlt 정책은 서로 독립적이면서 관련되어 있습니다. 이 정책의 조합은 다음 효과를 가집니다.

- -ahlt +cnt는 제공되는 기본 정책입니다. 이 기본값은 이벤트를 기록할 수 없더라도 감사된 이벤트가 처리되도록 합니다.

-ahlt 정책은 비동기 이벤트의 감사 레코드를 커널 감사 대기열에 둘 수 없는 경우 시스템에서 이벤트 수를 계산하고 처리를 계속하도록 지시합니다. 전역 영역에서 as_dropped 카운터가 수를 기록합니다.

+cnt 정책은 동기 이벤트가 도착하고 이벤트를 커널 감사 대기열에 둘 수 없는 경우 시스템에서 이벤트 수를 계산하고 처리를 계속하도록 지시합니다. 영역의 as_dropped 카운터가 수를 기록합니다.

-ahlt +cnt 구성은 처리를 계속하면 감사 레코드가 손실되더라도 처리를 계속해야 하는 사이트에서 일반적으로 사용됩니다. auditstat drop 필드는 영역에서 삭제된 감사 레코드의 수를 표시합니다.

- +ahlt -cnt 정책은 비동기 이벤트를 커널 감사 대기열에 추가할 수 없는 경우 처리를 정지하도록 지시합니다.

  +ahlt 정책은 비동기 이벤트의 감사 레코드를 커널 감사 대기열에 둘 수 없는 경우 모든 처리가 중지되도록 지시합니다. 시스템 패닉이 발생합니다. 비동기 이벤트가 감사 대기열에 들어가지 않으며 호출 스택의 포인터에서 복구해야 합니다.

  -cnt 정책은 동기 이벤트를 커널 감사 대기열에 둘 수 없는 경우 이벤트 전달을 시도하는 스레드가 차단되도록 지시합니다. 감사 공간을 사용할 수 있을 때까지 스레드는 일시 정지 대기열에 있습니다. 수가 계산되지 않습니다. 감사 공간을 사용할 수 있을 때까지 프로그램이 멈춘 것처럼 보일 수 있습니다.

  +ahlt -cnt 구성은 모든 감사 이벤트의 레코드가 시스템 가용성보다 우선하는 사이트에서 일반적으로 사용됩니다. auditstat wblk 필드는 스레드가 차단된 횟수를 표시합니다.

  하지만 비동기 이벤트가 발생할 경우 시스템 패닉이 발생하고 더 이상 작동하지 않습니다. 감사 이벤트의 커널 대기열은 저장된 충돌 덤프에서 수동으로 복구할 수 있습니다. 비동기 이벤트가 감사 대기열에 들어가지 않으며 호출 스택의 포인터에서 복구해야 합니다.
- -ahlt -cnt 정책은 비동기 이벤트를 커널 감사 대기열에 둘 수 없는 경우 이벤트 수를 계산하고 처리를 계속하도록 지시합니다. 동기 이벤트를 커널 감사 대기열에 둘 수 없는 경우 이벤트 전달을 시도하는 스레드가 차단됩니다. 감사 공간을 사용할 수 있을 때까지 스레드는 일시 정지 대기열에 있습니다. 수가 계산되지 않습니다. 감사 공간을 사용할 수 있을 때까지 프로그램이 멈춘 것처럼 보일 수 있습니다.

  -ahlt -cnt 구성은 모든 동기 감사 이벤트의 기록이 비동기 감사 레코드의 일부 손실보다 우선하는 사이트에서 일반적으로 사용됩니다. auditstat wblk 필드는 스레드가 차단된 횟수를 표시합니다.
- +ahlt +cnt 정책은 비동기 이벤트를 커널 감사 대기열에 둘 수 없는 경우 시스템 패닉이 발생하도록 지시합니다. 동기 이벤트를 커널 감사 대기열에 둘 수 없는 경우 시스템에서 이벤트 수를 계산하고 처리를 계속합니다.

프로세스 감사 특성

다음 감사 특성은 최초 로그인 시 설정됩니다.

- **프로세스 사전 선택 마스크** - 사용자 감사 마스크가 지정된 경우 시스템 전역 감사 마스크와 사용자 특정 감사 마스크의 조합입니다. 사용자가 로그인하면 로그인 프로세스가 사전

선택된 클래스를 결합하여 사용자의 프로세스에 대한 프로세스 사전 선택 마스크를 설정합니다. 프로세스 사전 선택 마스크는 감사 레코드를 생성하는 이벤트를 지정합니다.

다음 알고리즘은 시스템에서 사용자의 프로세스 사전 선택 마스크를 어떻게 얻는지 설명합니다.

`(system-wide default flags  +` *always-audit-classes*`) -` *never-audit-classes*

`auditconfig -getflags` 명령 결과의 시스템 전역 감사 클래스를 사용자의 *always_audit* 키워드에 대한 `always-audit-classes` 값의 클래스에 추가합니다. 그런 다음 전체에서 사용자의 *never-audit-classes*의 클래스를 뺍니다. 또한 `audit_flags`(5) 매뉴얼 페이지를 참조하십시오.

- **감사 사용자 ID** - 사용자가 로그인할 때 프로세스는 변경 불가능한 감사 사용자 ID를 얻습니다. 이 ID는 사용자의 초기 프로세스로 시작된 모든 하위 프로세스에서 상속됩니다. 감사 사용자 ID는 책임을 적용하는 데 도움이 됩니다. 사용자가 역할을 맡은 후에도 감사 사용자 ID는 동일하게 유지됩니다. 각 감사 레코드에 저장된 감사 사용자 ID를 사용하여 특정 작업을 로그인 사용자로 항상 역추적할 수 있습니다.
- **감사 세션 ID** - 감사 세션 ID는 로그인 시에 지정됩니다. 이 ID는 모든 하위 프로세스에서 상속됩니다.
- **터미널 ID** - 로컬 로그인의 경우, 터미널 ID는 로컬 시스템의 IP 주소와 사용자가 로그인한 물리적 장치를 식별하는 고유한 번호로 구성됩니다. 대부분의 경우 로그인은 콘솔을 통해 이루어집니다. 콘솔 장치에 해당하는 번호는 `0,0`입니다. 원격 로그인의 경우, 터미널 ID는 원격 호스트의 IP 주소와 원격 포트 번호 및 로컬 포트 번호로 구성됩니다.

감사 추적

감사 추적에는 이진 감사 파일이 포함됩니다. 추적은 `audit_binfile` 플러그인으로 만들어집니다. 감사 서비스는 감사 대기열의 레코드를 수집하고 디스크에 기록하도록 플러그인에 보냅니다.

이진 감사 파일 이름 지정 규칙

`audit_binfile` 플러그인은 이진 감사 파일을 만듭니다. 각 이진 감사 파일은 자체적으로 레코드의 모음입니다. 파일의 이름은 레코드가 생성된 기간과 레코드를 생성한 시스템을 식별합니다. 기간을 나타내는 시간 기록은 협정 세계시(UTC)로 지정되어 서로 다른 시간대에서도 올바른 순서로 정렬되도록 합니다.

자세한 내용은 `audit.log`(4) 매뉴얼 페이지를 참조하십시오. 열리고 닫힌 감사 파일 이름의 예는 `not_terminated` 감사 파일을 정리하는 방법 [95]을 참조하십시오.

감사 레코드 구조

감사 레코드는 감사 토큰의 시퀀스입니다. 각 감사 토큰에는 사용자 ID, 시간 및 날짜와 같은 이벤트 정보가 포함됩니다. `header` 토큰은 감사 레코드를 시작하고, 선택적 `trailer` 토큰은 레코드를 종료합니다. 기타 감사 토큰에는 감사 이벤트와 관련된 정보가 포함됩니다. 다음 그림은 일반적인 커널 감사 레코드 및 일반적인 사용자 레벨 감사 레코드를 보여줍니다.

그림 7-1 일반적인 감사 레코드 구조

감사 레코드 분석

감사 레코드 분석에는 감사 추적에서 레코드 사후 선택이 포함됩니다. 두 가지 방법 중 하나를 사용하여 수집된 이진 데이터를 구문 분석할 수 있습니다.

- `praudit` 명령을 사용할 수 있습니다. 명령에 대한 옵션은 서로 다른 텍스트 출력을 제공합니다. 예를 들어, `praudit -x` 명령은 스크립트 및 브라우저에 입력을 위한 XML을 제공합니다. `praudit` 출력에는 필드가 포함되지 않으며, 유일한 목적은 이진 데이터의 구문 분석을 돕는 것입니다. `praudit` 출력의 순서 및 형식은 Oracle Solaris 릴리스 사이에 보증되지 않습니다.

  `praudit` 출력의 예는 "이진 감사 파일의 컨텐츠 보기" [89]를 참조하십시오.

  각 감사 토큰에 대한 `praudit` 출력의 예는 "감사 토큰 형식" [117]의 개별 토큰을 참조하십시오.
- 이진 데이터 스트림 구문 분석을 위한 프로그램을 작성할 수 있습니다. 프로그램에서는 감사 레코드의 변형을 고려해야 합니다. 예를 들어, `ioctl()` 시스템 호출은 "잘못된 파

일 이름"에 대한 감사 레코드를 만듭니다 이 레코드에는 “잘못된 파일 설명자”에 대한 `ioctl()` 감사 레코드와 다른 토큰이 포함됩니다

- 각 감사 토큰에서 이진 데이터 순서에 대한 설명은 `audit.log`(4) 매뉴얼 페이지를 참조하십시오.
- 매니페스트 값은 `/usr/include/bsm/audit.h` 파일을 참조하십시오.
- 감사 레코드에서 토큰의 순서를 보려면 `auditrecord` 명령을 사용합니다. `auditrecord` 명령의 출력에는 서로 다른 매니페스트 값에 대해 서로 다른 토큰이 포함됩니다. 대괄호(`[]`)는 감사 토큰이 선택 사항임을 나타냅니다. 자세한 내용은 `auditrecord`(1M) 매뉴얼 페이지를 참조하십시오.

감사 토큰 형식

각 감사 토큰은 토큰 유형 식별자와 토큰에 대한 데이터로 구성됩니다. 다음 표는 각 토큰의 간략한 설명과 함께 토큰 이름을 나타냅니다. 더 이상 사용되지 않는 토큰은 이전 Solaris 릴리스와 호환성을 위해 유지됩니다.

표 7-1 감사에 대한 감사 토큰

토큰 이름	설명	자세한 정보
`acl`	액세스 제어 항목(ACE) 및 액세스 제어 목록(ACL) 정보	“`acl` 토큰” [118]
`arbitrary`	형식 및 유형 정보가 있는 데이터	`audit.log`(4) 매뉴얼 페이지
`argument`	시스템 호출 인수 값	“`argument` 토큰” [119]
`attribute`	파일 `vnode` 정보	“`attribute` 토큰” [119]
`cmd`	명령 인수 및 환경 변수	“`cmd` 토큰” [119]
`exec_args`	Exec 시스템 호출 인수	“`exec_args` 토큰” [119]
`exec_env`	Exec 시스템 호출 환경 변수	“`exec_env` 토큰” [120]
`exit`	프로그램 종료 정보	`audit.log`(4) 매뉴얼 페이지
`file`	감사 파일 정보	“`file` 토큰” [120]
`fmri`	프레임워크 관리 리소스 표시기	“`fmri` 토큰” [120]
`group`	프로세스 그룹 정보	“`group` 토큰” [121]
`header`	감사 레코드의 시작을 나타냄	“`header` 토큰” [121]
`ip`	IP 헤더 정보	`audit.log`(4) 매뉴얼 페이지
`ip address`	인터넷 주소	“`ip address` 토큰” [121]
`ip port`	인터넷 포트 주소	“`ip port` 토큰” [122]
`ipc`	시스템 V IPC 정보	“`ipc` 토큰” [122]
`IPC_perm`	시스템 V IPC 객체 액세스 정보	“`IPC_perm` 토큰” [122]
`opaque`	구조화되지 않은 데이터(지정되지 않은 형식)	`audit.log`(4) 매뉴얼 페이지
`path`	경로 정보	“`path` 토큰” [123]
`path_attr`	액세스 경로 정보	“`path_attr` 토큰” [123]

토큰 이름	설명	자세한 정보
권한	권한 세트 정보	“privilege 토큰” [123]
process	프로세스 정보	“process 토큰” [123]
return	시스템 호출의 상태	“return 토큰” [124]
sequence	시퀀스 번호	“sequence 토큰” [124]
socket	소켓 유형 및 주소	“socket 토큰” [124]
subject	주체 정보(process와 동일한 형식)	“subject 토큰” [125]
text	ASCII 문자열	“text 토큰” [125]
trailer	감사 레코드의 끝을 나타냄	“trailer 토큰” [125]
use of authorization	권한 부여 사용	“use of authorization 토큰” [126]
use of privilege	권한 사용	“use of privilege 토큰” [126]
user	사용자 ID 및 사용자 이름	“user 토큰” [126]
xclient	X 클라이언트 식별	“xclient 토큰” [126]
zonename	영역의 이름	“zonename 토큰” [126]
Trusted Extensions 토큰	label 및 X 창 시스템 정보	“Trusted Extensions 구성 및 관리 ”의 “Trusted Extensions 감사 참조”

다음 토큰은 더 이상 사용되지 않습니다.

- liaison
- host
- tid

더 이상 사용되지 않는 토큰에 대한 정보는 해당 토큰이 포함된 릴리스의 참조 자료를 참조하십시오.

감사 레코드는 항상 감사 레코드가 감사 추적에서 시작되는 위치를 나타내는 header 토큰으로 시작합니다. 지정 가능한 이벤트의 경우 subject 및 process 토큰이 이벤트를 일으킨 프로세스의 값을 가리킵니다. 지정 불가능한 이벤트의 경우 process 토큰이 시스템을 가리킵니다.

acl 토큰

acl 토큰에는 ZFS 파일 시스템의 ACE(액세스 제어 항목) 및 레거시 UFS 파일 시스템의 ACL(액세스 제어 목록)에 대한 정보를 기록하는 데 서로 다른 형식이 사용됩니다.

acl 토큰이 UFS 파일 시스템에 대해 기록되는 경우 praudit -x 명령은 다음과 같이 필드를 표시합니다.

```
<acl type="1" value="root" mode="6"/>
```

acl 토큰이 ZFS 데이터 세트에 대해 기록되는 경우 praudit -x 명령은 다음과 같이 필드를 표시합니다.

```
<acl who="root" access_mask="default" flags="-i,-R" type="2"/>
```

argument 토큰

argument 토큰에는 시스템 호출의 인수에 대한 정보(시스템 호출의 인수 수, 인수 값 및 선택적 설명)가 포함됩니다. 이 토큰은 감사 레코드에서 32비트 정수 시스템 호출 인수를 허용합니다.

praudit -x 명령은 argument 토큰의 필드를 다음과 같이 표시합니다.

```
<argument arg-num="2" value="0x5401" desc="cmd"/>
```

attribute 토큰

attribute 토큰에는 파일 vnode의 정보가 포함됩니다.

attribute 토큰은 대개 path 토큰과 함께 사용됩니다. attribute 토큰은 경로 검색 중 생성됩니다. 경로 검색 오류가 발생할 경우 vnode를 사용해서 필요한 파일 정보를 가져올 수 없습니다. 따라서 attribute 토큰이 감사 레코드의 일부로 포함되지 않습니다. praudit -x 명령은 attribute 토큰의 필드를 다음과 같이 표시합니다.

```
<attribute mode="20620" uid="root" gid="tty" fsid="0" nodeid="9267" device="108233"/>
```

cmd 토큰

cmd 토큰은 명령과 연결된 인수 목록 및 환경 변수 목록을 기록합니다.

praudit -x 명령은 cmd 토큰의 필드를 표시합니다. 다음 예제는 잘린 cmd 토큰입니다. 행은 표시 목적으로 줄바꿈되었습니다.

```
<cmd><arge>WINDOWID=6823679</arge>
<arge>COLORTERM=gnome-terminal</arge>
<arge>...LANG=C</arge>...<arge>HOST=machine1</arge>
<arge>LPDEST=printer1</arge>...</cmd>
```

exec_args 토큰

exec_args 토큰은 exec() 시스템 호출의 인수를 기록합니다.

praudit -x 명령은 exec_args 토큰의 필드를 다음과 같이 표시합니다.

```
<exec_args><arg>/usr/bin/sh</arg><arg>/usr/bin/hostname</arg></exec_args>
```

참고 - exec_args 토큰은 argv 감사 정책 옵션이 활성화된 경우에만 출력됩니다.

exec_env 토큰

exec_env 토큰은 exec() 시스템 호출의 현재 환경 변수를 기록합니다.

praudit -x 명령은 exec_env 토큰의 필드를 표시합니다. 다음 예제의 행은 표시 목적으로 줄바꿈되었습니다.

```
<exec_env><env>_=/usr/bin/hostname</env>
<env>LANG=C</env><env>PATH=/usr/bin</env>
<env>LOGNAME=jdoe</env><env>USER=jdoe</env>
<env>DISPLAY=:0</env><env>SHELL=/bin/csh</env>
<env>HOME=/home/jdoe</env><env>PWD=/home/jdoe</env><env>TZ=US/Pacific</env>
</exec_env>
```

참고 - exec_env 토큰은 arge 감사 정책 옵션이 활성화된 경우에만 출력됩니다.

file 토큰

file 토큰은 새 감사 파일의 시작과 이전 파일이 비활성화된 경우 이전 감사 파일의 끝을 표시하는 특수 토큰입니다. 처음 file 토큰은 감사 추적에서 이전 파일을 식별합니다. 최종 file 토큰은 감사 추적에서 다음 파일을 식별합니다. 이러한 토큰은 연속된 감사 파일을 하나의 감사 추적으로 연결합니다.

praudit -x 명령은 file 토큰의 필드를 표시합니다. 다음 예제의 행은 표시 목적으로 줄바꿈되었습니다.

```
<file iso8601="2009-04-08 14:18:26.200 -07:00">
/var/audit/machine1/files/20090408211826.not_terminated.machine1</file>
```

fmri 토큰

fmri 토큰은 결함 관리 리소스 표시기(FMRI)의 사용을 기록합니다. 자세한 내용은 smf(5) 매뉴얼 페이지를 참조하십시오.

praudit -x 명령은 fmri 토큰의 내용을 다음과 같이 표시합니다.

```
<fmri service_instance="svc:/system/cryptosvc"</fmri>
```

group 토큰

group 토큰은 프로세스 자격 증명의 그룹 항목을 기록합니다. group 토큰은 group 감사 정책 옵션이 활성화된 경우에만 출력됩니다.

praudit -x 명령은 group 토큰의 필드를 다음과 같이 표시합니다.

```
<group><gid>staff</gid><gid>other</gid></group>
```

header 토큰

header 토큰은 감사 레코드의 시작을 표시하는 특수 토큰입니다. header 토큰은 trailer 토큰과 결합하여 레코드의 다른 모든 토큰을 괄호로 묶습니다.

드물게, header 토큰이 하나 이상의 이벤트 수정자를 포함할 수 있습니다.

- fe는 실패한 감사 이벤트를 나타냅니다.
- fp는 실패한 권한 사용을 나타냅니다.
- na는 지정 불가능한 이벤트를 나타냅니다.

  ```
  header,52,2,system booted,na,mach1,2011-10-10 10:10:20.564 -07:00
  ```
- rd는 객체에서 데이터 읽기를 나타냅니다.
- sp는 성공한 권한 사용을 나타냅니다.

  ```
  header,120,2,exit(2),sp,mach1,2011-10-10 10:10:10.853 -07:00
  ```
- wr은 객체에 데이터 쓰기를 나타냅니다.

praudit 명령은 header 토큰을 다음과 같이 표시합니다.

```
header,756,2,execve(2),,machine1,2010-10-10 12:11:10.209 -07:00
```

praudit -x 명령은 header 토큰의 필드를 감사 레코드의 시작에 표시합니다. 다음 예제의 행은 표시 목적으로 줄바꿈되었습니다.

```
<record version="2" event="execve(2)" host="machine1"
iso8601="2010-10-10 12:11:10.209 -07:00">
```

ip address 토큰

ip address 토큰에는 인터넷 프로토콜 주소(IP 주소)가 포함됩니다. IP 주소는 IPv4 또는 IPv6 형식으로 표시될 수 있습니다. IPv4 주소는 4바이트를 사용합니다. IPv6 주소는 1바이트를 사용하여 주소 유형을 설명하고, 16바이트를 사용하여 주소를 설명합니다.

praudit -x 명령은 ip address 토큰의 내용을 다음과 같이 표시합니다.

```
<ip_address>machine1</ip_address>
```

ip port 토큰

ip port 토큰에는 TCP 또는 UDP 포트 주소가 포함됩니다.

praudit 명령은 ip port 토큰을 다음과 같이 표시합니다.

```
ip port,0xf6d6
```

ipc 토큰

ipc 토큰에는 호출자가 특정 IPC 객체를 식별하는 데 사용되는 시스템 V IPC 메시지 핸들, 세마포어 핸들 또는 공유 메모리 핸들이 포함됩니다.

IPC 객체 식별자는 감사 토큰의 컨텍스트 없는 성질에 위배됩니다. 전역 "이름"은 IPC 객체를 고유하게 식별하지 않습니다. 대신 IPC 객체가 핸들로 식별됩니다. 핸들은 IPC 객체가 활성화된 시간 동안에만 유효합니다. 하지만 IPC 객체의 식별이 문제가 되면 안됩니다. 시스템 V IPC 방식은 거의 사용되지 않으며, 방식은 모두 동일한 감사 클래스를 공유합니다.

다음 표는 IPC 객체 유형 필드에 가능한 값을 나타냅니다. 값은 /usr/include/bsm/audit.h 파일에 정의되어 있습니다.

표 7-2 IPC 객체 유형 필드에 대한 값

이름	값	설명
AU_IPC_MSG	1	IPC 메시지 객체
AU_IPC_SEM	2	IPC 세마포어 객체
AU_IPC_SHM	3	IPC 공유 메모리 객체

praudit -x 명령은 ipc 토큰의 필드를 다음과 같이 표시합니다.

```
<IPC ipc-type="shm" ipc-id="15"/>
```

IPC_perm 토큰

IPC_perm 토큰에는 시스템 V IPC 액세스 권한의 복사본이 포함됩니다. 이 토큰은 IPC 공유 메모리 이벤트, IPC 세마포어 이벤트 및 IPC 메시지 이벤트로 생성되는 감사 레코드에 추가됩니다.

praudit -x 명령은 IPC_perm 토큰의 필드를 표시합니다. 다음 예제의 행은 표시 목적으로 줄 바꿈되었습니다.

```
<IPC_perm uid="jdoe" gid="staff" creator-uid="jdoe"
creator-gid="staff" mode="100600" seq="0" key="0x0"/>
```

값은 IPC 객체와 연결된 IPC_perm 구조에서 가져옵니다.

path 토큰

path 토큰에는 객체에 대한 액세스 경로 정보가 포함됩니다.

praudit -x 명령은 path 토큰의 내용을 다음과 같이 표시합니다.

```
<path>/export/home/srv/.xsession-errors</path>
```

path_attr 토큰

path_attr 토큰에는 객체에 대한 액세스 경로 정보가 포함됩니다. 액세스 경로는 path 토큰 객체 아래에 있는 속성 파일 객체의 시퀀스를 지정합니다. openat()와 같은 시스템 호출이 속성 파일에 액세스합니다. 속성 파일 객체에 대한 자세한 내용은 fsattr(5) 매뉴얼 페이지를 참조하십시오.

praudit 명령은 다음과 같이 path_attr 토큰을 표시합니다.

```
path_attr,1,attr_file_name
```

privilege 토큰

privilege 토큰은 프로세스에 대한 권한 사용을 기록합니다. privilege 토큰은 기본 세트의 권한에 대해서는 기록되지 않습니다. 권한이 관리 작업으로 기본 세트에서 제거된 경우에는 해당 권한의 사용이 기록됩니다. 권한에 대한 자세한 내용은 "Oracle Solaris 11.2의 사용자 및 프로세스 보안 "의 "프로세스 권한 관리"를 참조하십시오.

praudit -x 명령은 privilege 토큰의 필드를 표시합니다.

```
<privilege set-type="Inheritable">ALL</privilege>
```

process 토큰

process 토큰에는 신호의 수신자와 같이 프로세스와 연결된 사용자에 대한 정보가 포함됩니다.

praudit -x 명령은 process 토큰의 필드를 표시합니다. 다음 예제의 행은 표시 목적으로 줄 바꿈되었습니다.

```
<process audit-uid="-2" uid="root" gid="root" ruid="root"
rgid="root" pid="567" sid="0" tid="0 0 0.0.0.0"/>
```

return 토큰

return 토큰에는 시스템 호출의 반환 상태(u_error) 및 프로세스 반환 값(u_rval1)이 포함됩니다.

return 토큰은 항상 시스템 호출에 대해 커널에서 생성한 감사 레코드의 일부로 반환됩니다. 응용 프로그램 감사에서 이 토큰은 종료 상태 및 기타 반환 값을 나타냅니다.

praudit 명령은 시스템 호출에 대한 return 토큰을 다음과 같이 표시합니다.

```
return,failure: Operation now in progress,-1
```

praudit -x 명령은 return 토큰의 필드를 다음과 같이 표시합니다.

```
<return errval="failure: Operation now in progress" retval="-1/">
```

sequence 토큰

sequence 토큰에는 시퀀스 번호가 포함됩니다. 시퀀스 번호는 감사 레코드가 감사 추적에 추가될 때마다 증분됩니다. sequence 토큰은 seq 감사 정책 옵션이 활성화된 경우에만 출력됩니다. 이 토큰은 디버깅에 유용합니다.

praudit -x 명령은 sequence 토큰의 내용을 표시합니다.

```
<sequence seq-num="1292"/>
```

socket 토큰

socket 토큰에는 인터넷 소켓을 설명하는 정보가 포함됩니다. 일부 인스턴스에서 토큰에는 원격 포트 및 원격 IP 주소만 포함됩니다.

praudit 명령은 socket 토큰의 인스턴스를 다음과 같이 표시합니다.

```
socket,0x0002,0x83b1,localhost
```

확장된 토큰은 소켓 유형 및 로컬 포트 정보를 포함한 정보를 추가합니다.

praudit -x 명령은 socket 토큰의 인스턴스를 다음과 같이 표시합니다. 다음 예제의 행은 표시 목적으로 줄바꿈되었습니다.

```
<socket sock_domain="0x0002" sock_type="0x0002" lport="0x83cf"
laddr="example1" fport="0x2383" faddr="server1.Subdomain.Domain.COM"/>
```

subject 토큰

subject 토큰은 작업을 수행하거나 시도하는 사용자를 설명합니다. 형식은 process 토큰과 동일합니다.

subject 토큰은 항상 시스템 호출에 대해 커널에서 생성한 감사 레코드의 일부로 반환됩니다. praudit 명령은 subject 토큰을 다음과 같이 표시합니다.

```
subject,jdoe,root,root,root,root,1631,1421584480,8243 65558 machine1
```

praudit -x 명령은 subject 토큰의 필드를 표시합니다. 다음 예제의 행은 표시 목적으로 줄바꿈되었습니다.

```
<subject audit-uid="jdoe" uid="root" gid="root" ruid="root"
rgid="root" pid="1631" sid="1421584480" tid="8243 65558 machine1"/>
```

text 토큰

text 토큰에는 텍스트 문자열이 포함됩니다.

praudit -x 명령은 text 토큰의 내용을 다음과 같이 표시합니다.

```
<text>booting kernel</text>
```

trailer 토큰

header 및 trailer의 두 토큰은 감사 레코드의 시작 및 끝점을 구별하고 다른 모든 토큰을 괄호로 묶는 특수 토큰입니다. header 토큰은 감사 레코드를 시작합니다. trailer 토큰은 감사 레코드를 끝냅니다. trailer 토큰은 선택적 토큰이며, trail 감사 정책 옵션이 설정된 경우에만 각 레코드의 마지막 토큰으로 추가됩니다.

트레일러가 설정된 상태에서 감사 레코드가 생성된 경우 auditreduce 명령은 trailer 토큰이 레코드 헤더를 올바르게 가리키는지 확인할 수 있습니다. trailer 토큰은 감사 추적의 역추적을 지원합니다.

praudit 명령은 trailer 토큰을 다음과 같이 표시합니다.

```
trailer,136
```

use of authorization 토큰

use of authorization 토큰은 권한 부여 사용을 기록합니다.

praudit 명령은 use of authorization 토큰을 다음과 같이 표시합니다.

```
use of authorization,solaris.role.delegate
```

use of privilege 토큰

use of privilege 토큰은 권한 사용을 기록합니다.

praudit -x 명령은 use of privilege 토큰의 필드를 다음과 같이 표시합니다.

```
<use_of_privilege result="successful use of priv">proc_setid</use_of_privilege>
```

user 토큰

user 토큰은 사용자 이름 및 사용자 ID를 기록합니다. 이 토큰은 사용자 이름이 호출자와 다른 경우에만 존재합니다.

praudit -x 명령은 user 토큰의 필드를 다음과 같이 표시합니다.

```
<user uid="123456" username="tester1"/>
```

xclient 토큰

xclient 토큰에는 X 서버에 대한 클라이언트 연결 수가 포함됩니다.

praudit -x 명령은 xlient 토큰의 내용을 다음과 같이 표시합니다.

```
<X_client>15</X_client>
```

zonename 토큰

zonename 토큰은 감사 이벤트가 발생한 영역을 기록합니다. 문자열 “global”은 전역 영역에서 발생한 감사 이벤트를 나타냅니다.

praudit -x 명령은 zonename 토큰의 내용을 다음과 같이 표시합니다.

```
<zone name="graphzone"/>
```

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 로그. 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 추적) 모든 호스트의 모든 감사 파일 모음입니다.

authenticated rights profile(인증된 권한 프로파일) 프로파일에서 작업을 수행하기 전에 지정된 사용자 또는 역할이 암호를 입력하도록 지정하는 rights profile(권한 프로파일)입니다. 이 동작은 `sudo` 동작과 비슷합니다. 암호가 유효한 기간을 구성할 수 있습니다.

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 관리에서 보안 정책에 의해 금지된 일련의 작업을 수행하기 위해 역할/사용자에 지정할 수 있는(또는 권한 프로파일에 포함할 수 있는) 권한입니다. 권한 부여는 커널이 아닌 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 인증 받을 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. 또한 provider(공급자)를 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, 사용할 장치를 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 포함해서 사용자에게 지정된 권한이 허용하는 리소스 범위 밖의 리소스에 대한 액세스 권한을 갖습니다. 그 결과, 프로세스가 권한이 없는 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 일반 사용자의 능력을 벗어나서 컴퓨터 시스템에서 작업을 수행하기 위한 기능 또는 능력입니다. 수퍼 유저 권한은 수퍼 유저에 부여되는 모든 rights(권한)입니다. 권한 있는 사용자 또는 권한 있는 응용 프로그램은 추가 권한이 부여된 사용자 또는 응용 프로그램입니다.

2. Oracle Solaris 시스템에서 프로세스에 대한 별개의 권한입니다. 권한은 `root`인 프로세스를 좀 더 세부적으로 제어합니다. 권한은 커널에서 정의되고 시행됩니다. 권한은 또한 프로세스 권한 또는 커널 권한이라고도 부릅니다. 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 벗어나는 권한이 지정된 사용자입니다. 또한 trusted users(신뢰된 사용자)를 참조하십시오.

profile shell(프로파일 셸)

권한 관리에서 역할(또는 사용자)이 권한 프로파일에 지정된 권한 있는 응용 프로그램을 명령줄에서 실행할 수 있는 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 약어입니다. 무결성 서비스나 프라이버시 서비스와 함께 사용되는 암호화 알고리즘을 선택할 수 있는 매개변수입니다.

RBAC Oracle Solaris의 사용자 권한 관리 기능인 역할 기반 액세스 제어입니다. 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 제공해야 하는 요구 사항입니다. 일반적으로 `sudo` 작업에 다시 인증이 필요합니다. 인증된 권한 프로파일은 다시 인증이 필요한 명령을 포함할 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. 현재까지는 `solaris`가 Oracle Solaris에 대해 유효한 정책입니다. `solaris` 정책은 권한, 확장된 권한 정책, 권한 부여 및 `setuid` 보안 속성을 인식합니다.

rights profile(권한 프로파일) 프로파일이라고도 합니다. 역할 또는 사용자에게 지정할 수 있는 보안 대체 모음입니다. 권한 프로파일은 권한 부여, 권한, 보안 속성 포함 명령 및 보조 프로파일이라고 부르는 기타 권한 프로파일을 포함할 수 있습니다.

rights(권한) all-or-nothing 수퍼 유저 모델의 대안입니다. 사용자 권한 관리 및 프로세스 권한 관리를 통해 조직은 수퍼 유저의 권한을 분할하고 이를 사용자 또는 역할에 지정할 수 있습니다. Oracle Solaris에서 권한은 커널 권한, 권한 부여 및 프로세스를 특정 UID 또는 GID로 실행할 수 있는 기능으로 구현됩니다. 권한은 rights profile(권한 프로파일) 및 role(역할)에서 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 시스템에서 초기 버전의 Kerberos에 대한 제품 이름입니다. 이 제품은 Massachusetts Institute of Technology에서 개발된 Kerberos V5 기술을 기반으로 합니다. SEAM은 이제 Kerberos 서비스라고 부릅니다. 이 기술은 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 일반 사용자가 관리 명령을 실행할 때 명령을 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 보안 속성으로서 `setuid root` 프로그램을 대신합니다. 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. 책임 구분 하에서는, 한 사용자가 트랜잭션을 완성하는 모든 작업을 수행하거나 승인할 수 없습니다. 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 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. 또한 provider(공급자)를 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 Service(티켓 부여 서비스)의 약어입니다. 티켓 발행을 담당하는 KDC의 부분입니다.

TGT Ticket-Granting Ticket(티켓 부여 티켓)의 약어입니다. 클라이언트가 다른 서비스에 대한 티켓을 요청할 수 있는 KDC에서 발행한 티켓입니다.

trusted users(신뢰된 사용자) 일정한 트러스트 레벨에서 관리 작업을 수행할 수 있도록 결정된 사용자입니다. 일반적으로 관리자는 먼저 신뢰할 수 있는 사용자에 대한 로그인을 만들고 사용자의 트러스트 및 기능 레벨에 따라 관리 권한을 지정합니다. 이러한 사용자는 시스템을 구성 및 유지 관리하는 데 도움을 줄 수 있습니다. 또한 권한 있는 사용자라고도 부릅니다.

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

ㄱ