Sun Java System Access Manager 7.1 관리 설명서

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

부품 번호: 820–0841

070428@17039

목차

머리말

Sun Java System Access Manager 7 관리 설명서에서는 명령줄 인터페이스를 통해 사용자와 서비스 데이터를 관리하고 Sun Java™ System Access Manager 콘솔을 사용하는 방법에 대해 설명합니다.

Access Manager는 Sun Java Enterprise System(Java ES)의 구성 요소로 네트워크 또는 인터넷 환경 전체에 배포되는 기업 응용 프로그램 지원에 필요한 서비스를 제공하는 일련의 소프트웨어 구성 요소입니다.

본 설명서의 대상

본 설명서는 Sun Java System 서버 및 소프트웨어를 사용하여 웹 액세스 플랫폼을 구현하는 IT 관리자 및 소프트웨어 개발자를 대상으로 제작되었습니다.

본 설명서를 읽기 전에

본 설명서를 읽는 사용자는 다음 구성 요소와 개념에 대해 알고 있어야 합니다.

- **Sun Java System Access Manager 7.1 Technical Overview**에 설명된 Access Manager 기술 개념
- 배포 플랫폼: Solaris™ 또는 Linux 운영 체제
- Access Manager를 실행할 웹 컨테이너: Sun Java System Application Server, Sun Java System Web Server, BEA WebLogic 또는 IBM WebSphere Application Server
- 기술 개념: Lightweight Directory Access Protocol(LDAP), Java 기술, JavaServer Pages™(JSP) 기술, HyperText Transfer Protocol(HTTP), HyperText Markup Language(HTML) 및 eXtensible Markup Language(XML)

관련 문서

사용할 수 있는 관련 문서는 다음과 같습니다.

Access Manager 핵심 설명서

Access Manager 핵심 설명서 세트에 포함된 항목은 다음과 같습니다.

- **Sun Java System Access Manager 7.1 릴리스 노트**는 제품 출시 후 온라인으로 제공됩니다. 여기에는 이 릴리스의 새로운 기능에 대한 설명, 알려진 문제점과 제한 사항, 설치 주의 사항, 소프트웨어 또는 설명서에 관한 문제를 보고하는 방법 등의 최신 정보가 포함되어 있습니다.
- **Sun Java System Access Manager 7.1 Technical Overview**는 Access Manager 구성 요소를 결합하여 액세스 제어 기능을 통합하고 기업 자산 및 웹 기반 응용 프로그램을 보호하는 방법에 대한 개요를 제공합니다. 또한 기본적인 Access Manager 개념 및 용어에 대해서 설명합니다.
- **Sun Java System Access Manager 7.1 Deployment Planning Guide**는 솔루션 수명 주기에 따라 Sun Java System Access Manager를 계획하고 배치하는 방법을 제공합니다.
- **Sun Java System Access Manager 7.1 Postinstallation Guide**는 Access Manager를 설치한 후 구성하는 방법에 대한 정보를 제공합니다.
- **Sun Java System Access Manager 7.1 Performance Tuning Guide**는 최적의 성능을 위해 Access Manager 및 관련 구성 요소를 조정하는 방법을 제공합니다.
- **Sun Java System Access Manager 7.1 관리 설명서**는 명령줄 인터페이스를 통해 사용자 및 서비스를 관리하고 Access Manager 콘솔을 사용하는 방법에 대해 설명합니다.
- **Sun Java System Access Manager 7.1 Federation and SAML Administration Guide**는 Liberty Alliance Project 사양에 따른 연합 모듈에 대한 정보를 제공합니다. 이 사양에 따른 통합 서비스 정보, Liberty 기반 환경 사용 지침 및 프레임워크 확장을 위한 API(Application Programming Interface) 요약 정보도 포함합니다.
- **Sun Java System Access Manager 7.1 Developer’s Guide**는 Access Manager를 사용자 정의하고 Access Manager 기능을 조직의 현재 기술 인프라에 통합하는 방법을 제공합니다. 또한 제품과 해당 API의 프로그램 사양에 대한 정보를 제공합니다.
- **Sun Java System Access Manager 7.1 C API Reference**는 공용 Access Manager C API를 구성하는 데이터 유형, 구조 및 기능에 대한 요약 정보를 제공합니다.
- **Java API Reference**는 Access Manager에서의 Java 패키지 구현에 대한 정보를 제공합니다.
- **Sun Java System Access Manager Policy Agent 2.2 User’s Guide**는 Access Manager에 적용 가능한 정책 기능 및 정책 에이전트에 대한 개요를 제공합니다.

릴리스 노트 업데이트 및 핵심 설명서 수정 링크는 Sun Java Enterprise System 설명서 웹 사이트의 Access Manager 페이지에서 찾을 수 있습니다. 업데이트된 문서에는 개정 날짜가 표시됩니다.

Sun Java Enterprise System 제품 설명서

다음 제품에 대한 설명서에서 유용한 정보를 찾을 수 있습니다.

- Directory Server
- Web Server
- Application Server
- Web Proxy Server

관련된 타사 웹 사이트 참조

이 설명서에 있는 타사 URL을 참조하여 추가 관련 정보를 살펴 보십시오.

주 – Sun은 본 설명서에서 언급된 타사 웹 사이트의 가용성 여부에 대해 책임을 지지 않습니다. 또한 해당 사이트나 리소스를 통해 제공되는 내용, 광고, 제품 및 기타 자료에 대해 어떠한 보증도 하지 않으며 그에 대한 책임도 지지 않습니다. 따라서 타사 웹 사이트의 내용, 제품 또는 리소스의 사용으로 인해 발생한 실제 또는 주장된 손상이나 피해에 대해서도 책임을 지지 않습니다.

문서, 지원 및 교육

Sun 웹 사이트에서는 다음과 같은 추가 자원에 관한 정보가 제공됩니다.

- 문서 (`http://www.sun.com/documentation/`)
- 지원 (`http://kr.sun.com/support/`)
- 교육 (`http://kr.sun.com/korea/`)

활자체 규약

다음 표에는 이 책에 사용된 활자체 규약이 나와 있습니다.

표 P–1 활자체 규약

활자체	의미	예
`AaBbCc123`	명령 이름, 파일, 디렉토리 및 화면 상의 컴퓨터 출력	`.login` 파일을 편집합니다. 모든 파일을 나열하려면 `ls -a`를 사용합니다. `machine_name% you have mail.`
`AaBbCc123`	컴퓨터 화면상의 출력에 대하여 입력할 내용	`machine_name%` **`su`** **`비밀번호:`**
aabbcc123	자리 표시자: 실제 이름이나 값으로 대체됩니다.	파일을 제거하는 명령은 `rm` *filename*입니다.
AaBbCc123	책 제목, 새로 나오는 용어, 강조 표시할 단어입니다.	**사용 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하면 **안 됩니다**. **주:** 일부 강조된 항목은 온라인에서 굵은 글씨로 표시됩니다.

명령 예의 쉘 프롬프트

C 쉘, Bourne 쉘 및 Korn 쉘에 대한 기본 UNIX® 시스템 프롬프트 및 수퍼유저 프롬프트는 다음 표와 같습니다.

표 P–2 쉘 프롬프트

쉘	프롬프트
C 쉘	`machine_name%`
C 쉘 수퍼유저	`machine_name#`
Bourne 쉘 및 Korn 쉘	`$`
Bourne 쉘 및 Korn 쉘 수퍼유저	`#`

Sun은 여러분의 의견을 환영합니다.

Sun은 설명서의 내용을 개선하기 위해 노력하고 있으며 사용자의 의견 및 제안을 환영합니다.

사용자 의견을 보내시려면 http://docs.sun.com에서 의견 보내기를 누릅니다. 온라인 양식에 설명서 제목과 부품 번호를 입력합니다. 부품 번호는 해당 설명서의 제목 페이지나 문서 맨 위에 있으며 일반적으로 7자리 또는 9자리 숫자입니다.

예를 들어, 본 설명서의 제목은 **Sun Java System Access Manager 7.1 관리 설명서**이며 부품 번호는 820-0841입니다. 사용자 의견을 제출할 때 해당 양식에 영문 설명서 제목과 부품 번호를 입력해야 할 수도 있습니다. 본 설명서의 영문 부품 번호와 제목은 819-4670, **Sun Java SystemAccess Manager 7.1 Administration Guide**입니다 .

1

액세스 제어

Sun Java System Access Manager™ 7.1 관리 설명서의 제1부입니다. Access Control 인터페이스는 인증 및 권한 부여 서비스를 만들고 및 관리하는 방법을 제공하여 영역 기반 자원을 보호하고 규제합니다. 기업 사용자가 정보를 요청하면 Access Manager는 사용자의 아이디를 확인하고 요청한 특정 자원에 액세스할 수 있는 권한을 부여합니다. 제2부는 다음 내용으로 구성되어 있습니다.

◆ ◆ ◆ 1 장

1 Access Manager 콘솔

Access Manager 콘솔은 다양한 액세스 수준을 가진 관리자가 이를 사용하여 영역 및 조직을 생성하고 이 영역에서 사용자를 생성 및 삭제하며, 영역의 자원을 보호하고 이에 대한 액세스를 제한하기 위한 적용 정책을 설정할 수 있는 웹 인터페이스입니다. 또한 관리자는 현재 사용자 세션을 확인 및 종료할 수 있으며 사용자의 연합 구성(인증 도메인 및 공급자 생성, 삭제 및 수정)을 관리할 수 있습니다. 반면 관리 권한이 없는 사용자는 개인 정보(이름, 전자 메일 주소, 전화 번호 등)를 관리하고 비밀번호를 변경하며, 그룹에 가입 및 탈퇴하고 자신의 역할을 확인할 수 있습니다. Access Manager에는 다음과 같은 두 개의 기본 보기가 있습니다.

관리 보기

관리 역할이 있는 사용자가 Access Manager에 인증하는 경우 기본 보기는 관리 보기입니다. 이 보기에서 관리자는 Access Manager와 관련된 대부분의 관리 작업을 수행할 수 있습니다. Access Manager는 영역 모드와 레거시 모드, 두 개의 다른 모드로 설치할 수 있습니다. 각 모드에는 고유의 콘솔이 있습니다. 영역 모드와 레거시 모드에 대한 자세한 내용은 **Sun Java System Access Manager 7.1 Technical Overview**를 참조하십시오.

주 – Access Manager 7.1을 영역 모드로 설치하면 레거시 모드로 되돌릴 수 없습니다. 하지만 Access Manager를 레거시 모드로 설치하는 경우 amadmin 명령을 사용하면 영역 모드로 되돌릴 수 있습니다. 자세한 내용은 **Access Manager Administration Reference**의 Changing from Legacy Mode to Realm Mode를 참조하십시오.

영역 모드 콘솔

영역 모드의 관리 콘솔에서 관리자는 영역 기반 액세스 제어, 기본 서비스 구성, 웹 서비스 및 연합을 관리할 수 있습니다. 관리자 로그인 화면에 액세스하려면 브라우저에서 다음 주소 구문을 사용하십시오.

protocol://*servername*/amserver/UI/Login

protocol은 배포 방법에 따라 http: 또는https입니다.

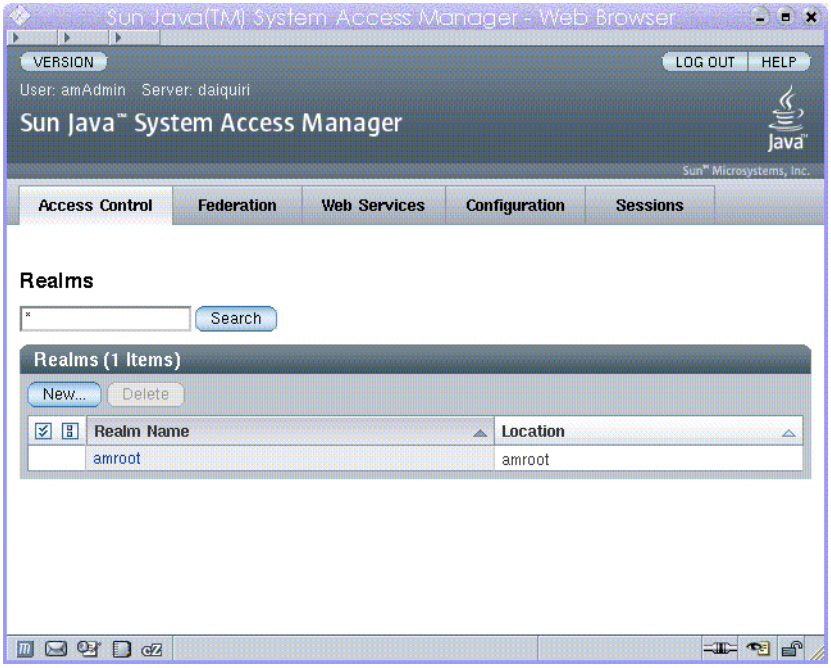

그림 1–1 영역 모드 관리 보기

레거시 모드 콘솔

레거시 모드 콘솔은 Access Manager 6.3 아키텍처를 기반으로 합니다. 레거시 Access Manager 아키텍처는 Sun Java System Directory Server와 함께 제공되는 LDAP 디렉토리 정보 트리(DIT)를 사용합니다. 레거시 모드에서 사용자 정보 및 액세스 제어 정보는 모두 LDAP 조직에 저장됩니다. 레거시 모드를 선택하는 경우 LDAP 조직은 액세스 제어 영역에 해당합니다. 영역 정보는 LDAP 조직 내에 통합됩니다. 레거시 모드에서는 Access Manager 기반의 identity 관리에 대해 디렉토리 관리 탭을 사용할 수 있습니다.

관리자 로그인 화면에 액세스하려면 브라우저에서 다음 주소 구문을 사용하십시오.

protocol://*servername*/amserver/console

protocol은 배포 방법에 따라 http: 또는 https:입니다.

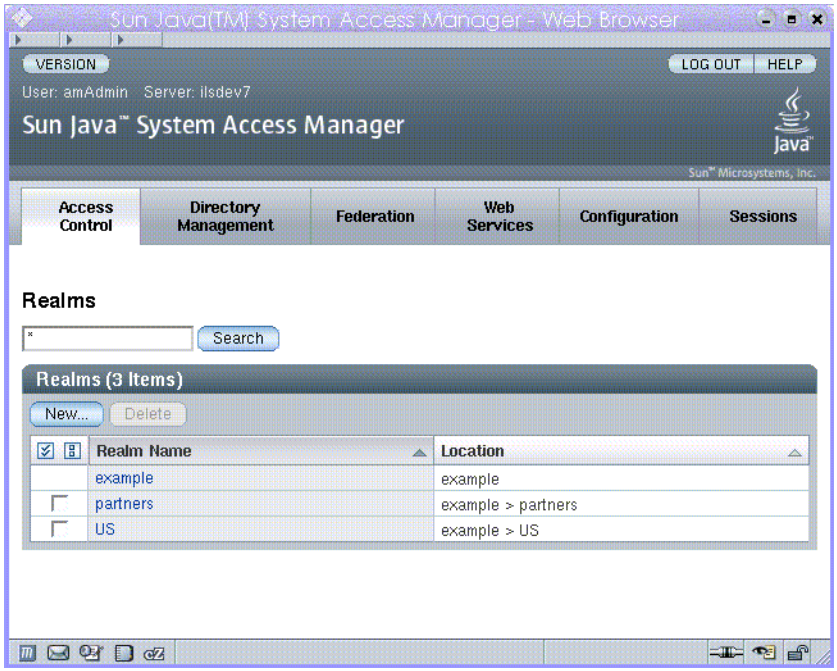

그림 1–2 레거시 모드 관리 보기

레거시 모드 6.3 콘솔

Access Manager 6.3의 일부 기능은 Access Manager 7.1 콘솔에서 사용할 수 없습니다. 이런 이유로 관리자는 7.1 레거시 배포를 통해 6.3 콘솔에 로그인할 수 있습니다. 이 콘솔은 Access Manager가 Sun Java System Portal Server 또는 Sun Java System Directory Server를 중앙 아이디 저장소로 사용해야 하는 기타 Sun Java System 통신 제품 상에 구축된 경우 일반적으로 사용됩니다. Delegated Administration 및 Class of Service와 같은 다른 기능은 이 콘솔을 통해서만 액세스할 수 있습니다.

주 – 6.3과 7.1 레거시 모드 콘솔을 번갈아 사용하지 마십시오.

6.3 콘솔에 액세스하려면 브라우저에서 다음 주소 구문을 사용합니다.

protocol://*servername*/amconsole

protocol은 배포 방법에 따라 http: 또는 https:입니다.

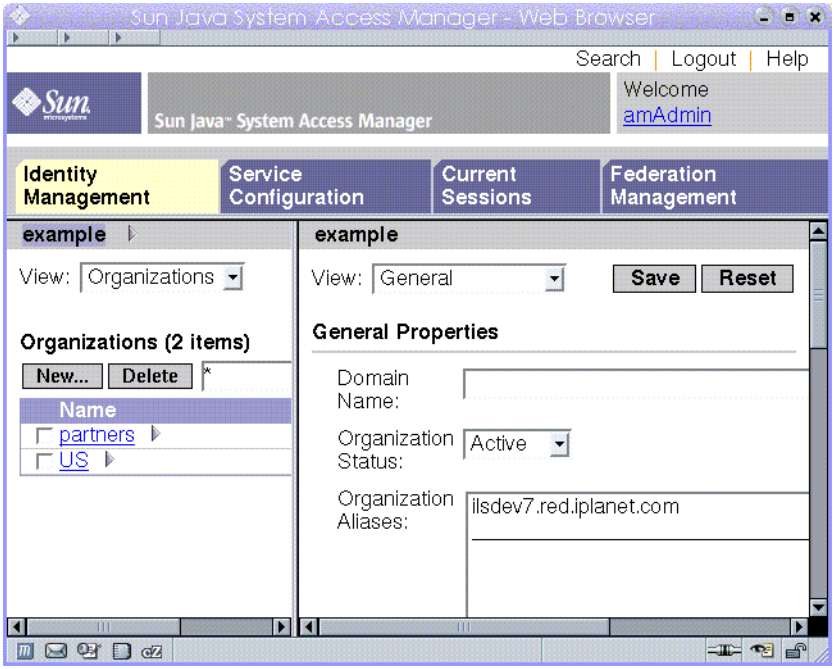

그림 1–3 Legacy 6.3 기반 콘솔

사용자 프로필 보기

관리 역할이 할당되지 않은 사용자가 Access Manager에 대해 인증을 수행할 때는 사용자 자신의 사용자 프로필이 기본 보기가 됩니다. 사용자 프로필 보기는 영역 또는 레거시 모드에서 액세스할 수 있습니다. 사용자는 이 보기에 액세스하려면 로그인 페이지에서 자신의 사용자 이름과 비밀번호를 입력해야 합니다.

이 보기에서 사용자는 개인 프로필 특정의 속성 값을 수정할 수 있습니다. 여기에는 이름, 주소(집), 비밀번호 등이 포함될 수 있지만 이에 제한되지는 않습니다. 사용자 프로필 보기에 표시되는 속성은 확장할 수 있습니다.

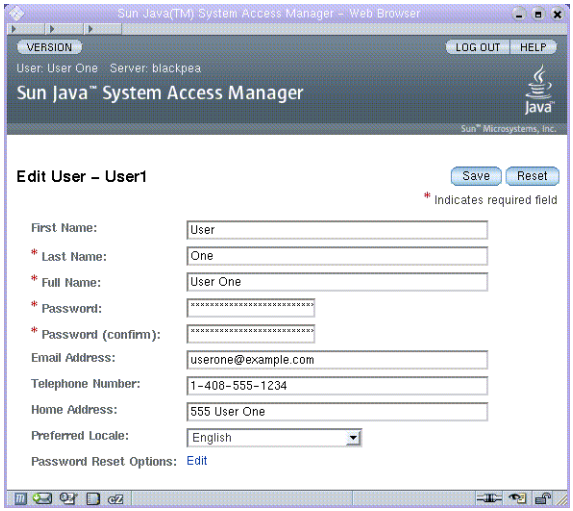

그림 1–4 사용자 프로필 보기

◆ ◆ ◆ 2 장

2 영역 관리

액세스 제어 영역은 사용자 또는 사용자의 그룹에 연결할 수 있는 인증 등록 정보 및 권한 부여 정책의 그룹입니다. 영역 데이터는 Access Manager에서 사용자가 지정한 데이터 저장소 내에 만든 소유 정보 트리에 저장됩니다. Access Manager 프레임워크는 Access Manager 정보 트리 내의 각 영역에 있는 정책과 속성을 종합합니다. 기본적으로 Access Manager는 사용자 데이터와는 별도로 Access Manager 정보 트리를 특수 분기로 Sun Java Enterprise System Directory Server에 자동으로 삽입합니다. 어떤 LDAPv3 데이터베이스를 사용하는 중이라도 액세스 제어 영역을 사용할 수 있습니다.

영역에 대한 자세한 내용은 **Sun Java System Access Manager 7.1 Technical Overview**를 참조하십시오.

[영역] 탭에서 액세스 제어에 대해 다음과 같은 등록 정보를 구성할 수 있습니다.

영역 만들기 및 관리

이 절에서는 영역을 만들고 관리하는 방법을 설명합니다.

▼ 새 영역 만들기

1 **[액세스 제어] 탭의 [영역] 목록에서 [새로 만들기]를 선택합니다.**

2 **다음과 같은 일반 속성을 정의합니다.**

이름 영역 이름을 입력합니다.

부모 만드는 영역의 위치를 정의합니다. 새 영역이 위치할 부모 영역을 선택합니다.

3 **다음과 같은 영역 속성을 지정합니다.**

영역 상태 — 활성 또는 비활성 상태를 선택합니다. 기본값은 활성입니다. 이 값은 영역의 수명 동안 등록 정보] 아이콘을 선택하여 언제든지 변경할 수 있습니다. 비활성을 선택하면 로그인 시 사용자 액세스를 사용할 수 없습니다.

영역/DNS 별칭 — 영역의 DNS 이름에 대한 별칭 이름을 추가할 수 있습니다. 이 속성에는 "실제" 도메인 별칭(임의의 문자열은 허용 안 됨)만 허용됩니다.

4 **저장하려면 [확인]을 누르고 이전 페이지로 돌아가려면 [취소]를 누릅니다.**

일반 등록 정보

일반 등록 정보 페이지에는 영역에 대한 기본 속성이 표시됩니다. 이 등록 정보를 수정하려면 [액세스 제어] 탭의 [영역 이름] 목록에서 해당 영역을 누릅니다. 그리고 다음 등록 정보를 편집합니다.

영역 상태 — 활성 또는 비활성 상태를 선택합니다. 기본값은 활성입니다. 이 값은 영역의 수명 동안 [등록 정보] 아이콘을 선택하여 언제든지 변경할 수 있습니다. 비활성을 선택하면 로그인 시 사용자 액세스를 사용할 수 없습니다.

영역/DNS 별칭 — 영역의 DNS 이름에 대한 별칭 이름을 추가할 수 있습니다. 이 속성에는 "실제" 도메인 별칭(임의의 문자열은 허용 안 됨)만 허용됩니다.

등록 정보를 편집한 다음 [저장]을 누릅니다.

주 – `AMAdmin.dtd`의 `recursive=true` 플래그는 영역 모드에서 하위 영역의 객체를 검색하는 경우 제대로 작동하지 않습니다. 모든 하위 조직이 같은 루트 접미어 아래에 있으므로 이 플래그는 레거시 모드에서만 작동합니다. 영역 모드에서 각 하위 영역은 서로 다른 루트 접미어를 가지며 다른 서버에 있을 수도 있습니다. 하위 영역에서 그룹과 같은 객체를 검색할 경우 XML 데이터 파일에서 검색할 하위 영역을 지정해야 합니다.

인증

사용자가 다른 인증 모듈을 사용하여 로그인하기 전에 일반 인증 서비스를 영역에 대한 서비스로 등록해야 합니다. 핵심 인증 서비스를 사용하면 Access Manager 관리자가 영역의 인증 매개 변수에 대한 기본값을 지정할 수 있습니다. 지정된 인증 모듈에 정의된 대체 값이 없는 경우 이러한 값을 사용할 수 있습니다. 핵심 인증 서비스의 기본값은 `amAuth.xml` 파일에 정의되며 설치 후에 Directory Server에 저장됩니다.

자세한 내용은 인증 관리를 참조하십시오.

서비스

Access Manager에서 서비스는 Access Manager 콘솔에서 함께 관리되는 속성의 그룹입니다. 속성은 직원 이름, 직위 및 전자 메일 주소와 같은 관련 정보입니다. 그러나 속성은 일반적으로 메일 응용 프로그램이나 급여 서비스와 같은 소프트웨어 모듈의 구성 매개 변수로 사용됩니다.

[서비스] 탭을 사용하여 몇 가지 Access Manager 기본 서비스를 영역에 추가하고 구성할 수 있으며, 다음과 같은 서비스를 추가할 수 있습니다.

- 관리
- 검색 서비스
- 국제화 설정
- 비밀번호 재설정
- 세션
- 사용자

주 – Access Manager에서 서비스 XML 파일의 필수 속성에는 반드시 기본값이 있어야 합니다. 서비스의 필수 속성에 값이 없으면 기본값을 추가하고 해당 서비스를 다시 로드해야 합니다.

▼ 영역에 서비스를 추가하려면

1. **새 서비스를 추가할 영역 이름을 누릅니다.**
2. **[서비스] 탭을 선택합니다.**
3. **서비스 목록에서 [추가]를 누릅니다.**
4. **영역에 추가할 서비스를 선택합니다.**
5. **[다음]을 누릅니다.**
6. **영역 속성을 정의하여 서비스를 구성합니다. 서비스 속성에 대한 설명은 온라인 도움말에서 구성 부분을 참조하십시오.**
7. **[마침]을 누릅니다.**

8 서비스의 등록 정보를 편집하려면 [서비스] 목록에서 해당 이름을 누릅니다.

권한

Access Manager의 위임 모델은 관리자에게 할당된 권한 또는 자격을 바탕으로 합니다. 권한은 정책 객체에 대한 READ 작업처럼 자원에 대해 수행할 수 있는 작업입니다. 정의되는 작업 집합은 READ, MODIFY 및 DELEGATE입니다. 자원은 작업을 수행할 수 있는 객체로, 구성 객체 또는 Identity 객체일 수 있습니다.

구성 객체의 예로는 인증 구성, 정책, 데이터 저장소 등이 있으며, Identity 객체의 예로는 사용자, 그룹, 규칙 및 에이전트가 있습니다. 권한 집합은 동적으로 만들어 Access Manager에 동적으로 추가할 수 있지만 Access Manager가 제대로 실행되도록 설치 중에 소규모의 권한 집합이 추가됩니다. 로드된 권한은 역할과 그룹에 할당할 수 있습니다. 이러한 역할과 그룹에 속한 사용자가 위임된 관리자가 되어 할당된 작업을 수행할 수 있습니다. 기본적으로 관리자는 하나 이상의 권한이 할당된 역할과 그룹에 속하는 사용자입니다.

Access Manager 7.1을 사용하면 다음과 같은 관리자 유형에 대해 권한을 구성할 수 있습니다.

- 영역 관리자 — 모든 객체(구성 및 Identity 객체 모두)에 대한 모든 READ, MODIFY 및 DELEGATE 작업 권한을 가집니다. Unix 시스템에서 영역 관리자는 "root"로 간주될 수 있습니다. 영역 관리자는 하위 영역을 만들고, 모든 서비스에 대한 구성을 수정하고, 사용자, 그룹, 역할 및 에이전트를 작성, 수정 및 삭제할 수도 있습니다.
- 정책 관리자 — 정책과 정책 서비스 구성만 관리할 수 있는 권한을 가집니다. 정책 관리자는 규칙, 주제, 조건 및 응답 속성으로 구성된 정책을 작성, 수정 및 삭제할 수 있습니다. 하지만 정책을 관리하기 위해서는 Identity 저장소 주제 및 인증 구성에 대한 읽기 권한도 필요합니다. 정책 관리자는 이 권한을 사용하여 Identity와 인증 구성을 볼 수 있습니다.
- 로그 관리자 — 감염된 응용 프로그램에서 감사 로그를 악의적으로 사용하지 못하도록 보호하는 데 사용할 수 있는 로그 레코드를 읽고 쓰는 권한을 가집니다. 로깅 인터페이스가 공용이므로 인증된 모든 사용자가 로그 레코드를 읽고 쓸 수 있습니다. 따라서 이러한 오용을 방지하기 위해 이 권한이 추가됩니다. 로깅 인터페이스를 주로 사용하는 J2EE 및 웹 에이전트에는 MODIFY 권한만 필요하며 READ 권한은 허용되지 않습니다. 이와 비슷하게 로그를 보는 관리자에게는 READ 권한만 있어야 하며 MODIFY 권한은 허용되지 않습니다. 이러한 사용 유형을 지원하기 위해 로깅 권한을 다음과 같이 보다 자세히 분류할 수 있습니다.
  - 쓰기 액세스 권한을 가진 로그 관리자 – 모든 로그 파일을 쓸 수 있는 권한을 가집니다.
  - 읽기 액세스 권한을 가진 로그 관리자 – 모든 로그 파일을 읽을 수 있는 권한을 가집니다.

- 읽기/쓰기 액세스 권한을 가진 로그 관리자 – 모든 로그 파일을 읽고 쓸 수 있는 권한을 가집니다.

Access Manager 7.1 권한 정의

새로운 Access Manager 7.1 설치 인스턴스는 정책 관리자, 영역 관리자(또는 레거시 모드의 조직 관리자) 및 로그 관리자에 대해 액세스 권한을 제공합니다. 권한을 할당하거나 수정하려면 편집할 역할 또는 그룹 이름을 누릅니다. 그러면 다음 중에서 선택할 수 있습니다.

모든 로그 파일에 대한 읽기 및 쓰기 액세스
: 로그 관리자의 읽기 및 쓰기 액세스 권한을 정의합니다.

모든 로그 파일에 대한 쓰기 액세스
: 로그 관리자의 쓰기 액세스 권한만 정의합니다.

모든 로그 파일에 대한 읽기 액세스
: 로그 관리자의 읽기 액세스 권한만 정의합니다.

정책 등록 정보 전용의 읽기 및 쓰기 액세스
: 정책 관리자의 읽기 및 쓰기 액세스 권한을 정의합니다.

모든 영역 및 정책 등록 정보에 대한 읽기 및 쓰기 액세스
: 영역 관리자의 읽기 및 쓰기 액세스 권한을 정의합니다.

Access Manager 7.0에서 7.1로의 업그레이드 권한 정의

버전 7.0에서 버전 7.1로 Access Manager를 업그레이드한 경우 권한 구성은 새로 설치한 Access Manager 7.1의 권한 구성과 다르지만 정책 관리자, 영역 관리자 및 로그 관리자에 대한 권한은 그대로 지원됩니다. 권한을 할당하거나 수정하려면 편집할 역할 또는 그룹 이름을 누릅니다. 그러면 다음 중에서 선택할 수 있습니다.

데이터 저장소에 대한 읽기 전용 권한
: 데이터 저장소에 대한 정책 관리자의 읽기 액세스 권한을 정의합니다.

모든 로그 파일에 대한 읽기 및 쓰기 액세스
: 로그 관리자의 읽기 및 쓰기 액세스 권한을 정의합니다.

모든 로그 파일에 대한 쓰기 액세스
: 로그 관리자의 쓰기 액세스 권한만 정의합니다.

모든 로그 파일에 대한 읽기 액세스
: 로그 관리자의 읽기 액세스 권한만 정의합니다.

정책 등록 정보 전용의 읽기 및 쓰기 액세스
: 정책 관리자의 읽기 및 쓰기 액세스 권한을 정의합니다.

모든 영역 및 정책 등록 정보에 대한 읽기 및 쓰기 액세스
영역 관리자의 읽기 및 쓰기 액세스 권한을 정의합니다.

모든 등록 정보 및 서비스에 대한 읽기 전용 권한
모든 등록 정보와 서비스에 대한 정책 관리자의 읽기 액세스 권한을 정의합니다.

Access Manager에서는 다음 정의를 개별적으로 또는 함께 사용할 수 없습니다.

- 데이터 저장소에 대한 읽기 전용 권한
- 모든 등록 정보 및 서비스에 대한 읽기 전용 권한

이러한 권한 정의는 정책 관리자의 위임 제어를 정의할 때 "정책 등록 정보 전용의 읽기 및 쓰기 액세스" 정의와 함께 사용되어야 합니다.

◆ ◆ ◆ 3 장

3 데이터 저장소

데이터 저장소는 사용자 속성과 사용자 구성 데이터를 저장할 수 있는 데이터베이스입니다. Access Manager는 LDAPv3 Identity 저장소 프레임워크에 연결할 Identity 저장소 플러그인을 제공합니다. 이러한 플러그인을 사용하면 기존 사용자 데이터베이스를 변경하지 않고도 Access Manager 사용자 정보를 확인하고 검색할 수 있습니다. Access Manager 프레임워크는 아이디 저장소 플러그인에서 얻은 데이터를 다른 Access Manager 플러그인에서 얻은 데이터와 통합하여 각 사용자의 가상 아이디를 구성합니다. Access Manager는 이제 두 개 이상의 아이디 저장소 간의 인증 및 권한 부여 과정에 이러한 범용 아이디를 사용할 수 있습니다. 가상 사용자 아이디는 사용자 세션이 종료되면 소멸됩니다.

Access Manager 데이터 저장소 유형

이 절에서는 구성할 수 있는 데이터 저장소 유형을 설명하고 새로운 데이터 저장소 유형을 만드는 단계와 그 구성 방법도 제공합니다.

다음 데이터 저장소 유형의 데이터 저장소 인스턴스를 새로 만들 수 있습니다.

Access Manager 저장소 플러그인

이 데이터 저장소 유형은 Sun Java System Directory Server 인스턴스에 있으며 Access Manager 정보 트리를 저장합니다. 이 저장소 유형은 LDAP 버전 3 사양에는 포함되지 않은 역할 및 서비스 클래스 등의 Directory Server 기능을 사용하며 이전 버전의 Access Manager와 호환됩니다.

활성 디렉토리

이 데이터 유형은 LDAP 버전 3 사양을 사용하여 Microsoft Active Directory 인스턴스에 Identity 데이터를 기록합니다.

플랫 파일 저장소

이 저장소를 사용하면 별도의 데이터 저장소를 만들지 않고도 Access Manager의 로컬 설치 인스턴스에 데이터 및 Identity를 플랫 DIT 구조로 저장할 수 있습니다. 이 저장소는 테스트 및 배포 검증에 널리 사용됩니다.

일반 LDAPv3

이 데이터 저장소 유형을 사용하면 Identity 데이터를 모든 LDAPv3 호환 데이터베이스에 기록할 수 있습니다. 사용 중인 LDAPv3 데이터베이스가 영구 검색을 지원하지 않는 경우 캐시 기능을 사용할 수 없습니다.

Access Manager 스키마를 사용하는 Sun Directory Server

이 데이터 저장소 유형은 Sun Java System Directory Server 인스턴스에 있으며 Access Manager 정보 트리를 저장합니다. Access Manager 저장소 플러그인과 달리 더 많은 구성 속성을 사용하여 데이터 저장소를 보다 잘 사용자 정의할 수 있습니다.

▼ 새 데이터 저장소 만들기

다음 절에서는 데이터 저장소를 연결하는 단계에 대해 설명합니다.

1. **새 데이터 저장소를 추가할 영역을 선택합니다.**
2. **[데이터 저장소] 탭을 누릅니다.**
3. **데이터 저장소 목록에서 [새로 만들기]를 누릅니다.**
4. **데이터 저장소의 이름을 입력합니다.**
5. **만들 데이터 저장소 유형을 선택합니다.**
6. **[다음]을 누릅니다.**
7. **적절한 속성 값을 입력하여 데이터 저장소를 구성합니다.**
8. **[마침]을 누릅니다.**

데이터 저장소 속성

이 절에서는 새 Access Manager 데이터 저장소를 각각 구성하는 데 사용되는 속성을 설명합니다. 저장소 속성은 다음과 같습니다.

주 – Active Directory, 일반 LDAPv3 및 Access Manager 스키마를 사용하는 Sun Directory Server 데이터 저장소 유형은 같은 기본 플러그인을 공유하므로 구성 속성도 서로 같습니다. 그러나 일부 속성의 기본값은 각 데이터 저장소 유형마다 다르며 이에 따라 Access Manager 콘솔에서도 다르게 표시됩니다.

Access Manager 저장소 속성

다음과 같은 속성을 사용하여 Access Manager 저장소 플러그인을 구성할 수 있습니다.

클래스 이름

Access Manager 저장소 플러그인을 구현할 클래스 파일의 위치를 지정합니다.

Access Manager 지원 유형 및 작업

해당 LDAP 서버상에서 허용되거나 수행할 수 있는 작업을 지정합니다. 해당 LDAPv3 저장소 플러그인이 지원하는 작업만 기본 작업입니다. LDAPv3 저장소 플러그인이 지원하는 작업은 다음과 같습니다.

- 그룹 — 읽기, 만들기, 편집, 삭제
- 사용자 — 읽기, 만들기, 편집, 삭제, 서비스
- 에이전트 — 읽기, 만들기, 편집, 삭제

LDAP 서버 설정에 따라 위 목록에서 권한을 제거할 수 있지만 권한을 더 추가할 수는 없습니다.

구성된 LDAPv3 저장소 플러그인이 Sun Java System Directory Server의 인스턴스를 가리키면 `role` 유형에 대한 권한을 추가할 수 있습니다. 그렇지 않으면 다른 데이터 저장소에서 역할을 지원하지 않을 수 있으므로 권한을 추가할 수 없습니다. 'role' 유형의 권한은 다음과 같습니다.

- 역할 — 읽기, 만들기, 편집, 삭제

`user` 유형이 LDAPv3 저장소에 대해 지원되는 경우 해당 사용자에 대해 읽기, 만들기, 편집 및 삭제 서비스 작업이 가능합니다. 즉, `user` 유형이 지원되는 경우 읽기, 편집, 만들기 및 삭제 작업을 사용하여 Identity 저장소에서 사용자 항목을 읽고 편집하고

만들고 삭제할 수 있습니다. user=service 작업을 사용하면 Access Manager 서비스가 사용자 항목의 속성에 액세스할 수 있습니다. 사용자가 속한 영역이나 역할에 서비스가 할당된 경우 이 사용자는 동적인 서비스 속성에 액세스할 수 있습니다.

또한 모든 할당된 서비스에 대한 사용자 속성도 관리할 수 있습니다. 사용자에게 service가 작업으로 지정(user=service)되어 있는 경우 지원되는 모든 서비스 관련 작업을 지정합니다. 이러한 작업으로는 assignService, unassignService, getAssignedServices, getServiceAttributes, removeServiceAttributes 및 modifyService가 있습니다.

조직 DN 값

Access Manager가 관리할 Directory Server 내의 조직을 가리키는 DN을 정의합니다. 이 DN은 데이터 저장소에서 수행되는 모든 작업의 기본 DN이 됩니다.

사용자 컨테이너 이름 지정 속성

사용자가 사용자 컨테이너에 있는 경우 사용자 컨테이너의 이름 지정 속성을 지정합니다. 사용자가 사용자 컨테이너에 없으면 이 필드는 비어 있습니다.

사용자 컨테이너 값

사용자 컨테이너의 값을 지정합니다. 기본값은 people입니다.

에이전트 컨테이너 이름 지정 속성

에이전트가 에이전트 컨테이너에 있는 경우 에이전트 컨테이너의 이름을 지정합니다. 에이전트가 에이전트 컨테이너에 없으면 이 필드는 비어 있습니다.

에이전트 컨테이너 값

에이전트 컨테이너 값을 지정합니다. 기본값은 agents입니다.

재귀적 검색

이 옵션이 활성화되면 Access Manager 저장소에서 지정된 Identity에 대해 재귀적으로 검색이 수행됩니다. 다음 데이터 구조에서 재귀적 검색을 수행하는 경우를 예로 들어 보겠습니다.

루트
영역1
  하위 영역11
    사용자5
  하위 영역12
    사용자56
영역2

사용자1
사용자2
하위 영역21
  사용자3
  사용자4

그러면 다음과 같은 결과가 나옵니다.

- 루트에서 검색을 수행하는 경우 이 수준에 정의된 사용자(amadmin 및 anonymous 제외)가 없으면 검색에서 사용자 1–6을 반환합니다.
- **영역1**에서 검색을 수행하는 경우 정의된 사용자가 없으면 검색에서 **사용자5** 및 **사용자6**을 반환합니다.
- **영역2**(두 사용자가 정의됨)에서 검색을 수행하는 경우 검색에서 사용자 1–4를 반환합니다.

복사 영역 구성

영역 모드 설치에서 이 속성을 사용하면 Access Manager가 저장소에 있는 각 영역 및 하위 영역과 동일한 조직 및 하위 조직을 만듭니다. 또한 영역/하위 영역에 등록된 서비스가 새로 만들어진 조직/하위 조직에도 등록됩니다. 그러면 영역 DIT와 조직 DIT가 모두 데이터 저장소 내에 존재하게 됩니다.

플랫 파일 저장소 속성

다음 속성을 사용하여 플랫 파일 저장소를 구성할 수 있습니다.

파일 저장소 플러그인 클래스 이름

이 속성은 플랫 파일에 대한 구현을 제공하는 Java 클래스 파일을 지정합니다. 이 속성을 수정해서는 안 됩니다.

파일 저장소 디렉토리

Identity 및 해당 속성을 저장할 기본 디렉토리를 정의합니다.

캐시

사용하는 경우(기본값) Identity 및 해당 속성이 캐싱됩니다. 후속 요청은 파일 시스템에 액세스하지 않습니다.

캐시 업데이트 시간

캐싱을 사용하면 이 속성으로 파일 시스템이 변경된 경우 캐시의 해당 항목을 검사하기까지의 시간 간격(분 단위)를 확인할 수 있습니다. 이 검사 메커니즘은 타임스탬프를 기반으로 합니다.

파일 사용자 객체 클래스

사용자를 만들 때 해당 사용자에게 자동으로 추가되는 객체 클래스를 정의합니다.

비밀번호 속성

인증에 사용되는 비밀번호를 포함하는 속성 이름을 제공합니다. 이 속성은 데이터 저장소 인증 모듈을 사용하는 경우 사용자를 인증하는 데 사용됩니다.

상태 속성

Identity의 상태를 저장하는 속성 이름을 제공합니다. 상태 속성 값으로는 `active` 또는 `inactive`를 사용할 수 있습니다. 이 속성은 Identity 인증에 사용되며 Identity가 `inactive`인 경우 사용자는 인증에 사용되지 않습니다.

해시된 속성

값이 해시되어 파일에 저장된 속성 목록을 제공합니다. 해시된 후에는 원래 값을 얻을 수 없으며 해시된 값만 검색됩니다. 영구적으로 저장되지는 않지만 검증에 사용되는 특정 속성의 기밀을 보장하는 데 사용됩니다. 이러한 속성 유형의 예로는 Identity의 비밀번호 속성이 있습니다.

암호화된 속성

값이 암호화되어 파일에 저장된 속성 목록을 제공합니다. 암호화되어 저장되어 있지만 Identity 저장소 API를 호출하면 원래의 암호화되지 않은 값이 반환됩니다. 이 속성은 사용자가 파일 시스템에 직접 액세스하여 중요한 속성을 읽지 못하도록 합니다.

LDAPv3 속성

다음과 같은 속성을 사용하여 LDAPv3 저장소 플러그인을 구성할 수 있습니다.

LDAP 서버

연결할 LDAP 서버의 이름을 입력하며 `hostname.domainname:portnumber` 형식이어야 합니다.

두 개 이상의 `host:portnumber` 항목을 입력한 경우 목록의 첫 번째 호스트로 연결이 시도됩니다. 현재 호스트에 대한 연결 시도가 실패한 경우에만 목록의 다음 항목에 대한 연결을 시도합니다.

LDAP 바인드 DN

현재 사용자가 연결된 LDAP 서버에 대한 인증에 Access Manager가 사용할 DN 이름을 지정합니다. DN 이름이 바인딩된 사용자는 38 페이지 “LDAPv3 플러그인 지원 유형 및 작업” 속성에서 구성한 올바른 추가, 수정 및 삭제 권한을 가져야 합니다.

LDAP 바인드 비밀번호

현재 사용자가 연결된 LDAP 서버에 대한 인증에 Access Manager가 사용할 DN 비밀번호를 지정합니다.

LDAP 바인드 비밀번호(확인)

비밀번호를 확인합니다.

LDAP 조직 DN

해당 데이터 저장소의 저장소가 매핑될 DN으로 데이터 저장소에서 수행되는 모든 작업의 기본 DN이 됩니다.

LDAP SSL

활성화된 경우 Access Manager는 HTTPS 프로토콜을 사용하여 기본 서버에 연결합니다.

LDAP 연결 풀 최소 크기

연결 풀의 초기 연결 수를 지정합니다. 연결 풀을 사용하면 매번 새로 연결할 필요가 없습니다.

LDAP 연결 풀 최대 크기

허용된 최대 연결 수를 지정합니다.

검색에서 반환되는 최대 결과 수

검색 작업에서 반환되는 최대 항목 수를 지정합니다. 해당 제한값에 도달하면 Directory Server는 검색 요청과 일치하는 모든 항목을 반환합니다.

검색 시간 초과

검색 요청에 할당할 최대 시간(초)을 지정합니다. 해당 제한값에 도달하면 Directory Server는 검색 요청과 일치하는 모든 항목을 반환합니다.

LDAP에서 참조를 따름

이 옵션이 활성화되면 다른 LDAP 서버로의 참조를 자동으로 따라갑니다.

LDAPv3 저장소 플러그인 클래스 이름

LDAPv3 저장소를 구현할 클래스 파일의 위치를 지정합니다.

일반 속성 이름 매핑

프레임워크에 알려진 공통 속성을 기본 데이터 저장소에 매핑할 수 있도록 합니다. 예를 들어 프레임워크에서 사용자 상태를 결정하는 데 `inetUserStatus`를 사용한다면 기본 데이터 저장소에서는 `userStatus`를 사용할 수 있습니다. 속성 정의는 대소문자를 구분합니다.

LDAPv3 플러그인 지원 유형 및 작업

해당 LDAP 서버상에서 허용되거나 수행할 수 있는 작업을 지정합니다. 해당 LDAPv3 저장소 플러그인이 지원하는 작업만 기본 작업입니다. LDAPv3 저장소 플러그인이 지원하는 작업은 다음과 같습니다.

- 그룹 — 읽기, 만들기, 편집, 삭제
- 사용자 — 읽기, 만들기, 편집, 삭제, 서비스
- 에이전트 — 읽기, 만들기, 편집, 삭제

LDAP 서버 설정에 따라 위 목록에서 권한을 제거할 수 있지만 권한을 더 추가할 수는 없습니다.

구성된 LDAPv3 저장소 플러그인이 Sun Java System Directory Server의 인스턴스를 가리키면 `role` 유형에 대한 권한을 추가할 수 있습니다. 그렇지 않으면 다른 데이터 저장소에서 역할을 지원하지 않을 수 있으므로 권한을 추가할 수 없습니다. 'role' 유형의 권한은 다음과 같습니다.

- 역할 — 읽기, 만들기, 편집, 삭제

`user` 유형이 LDAPv3 저장소에 대해 지원되는 경우 해당 사용자에 대해 읽기, 만들기, 편집 및 삭제 서비스 작업이 가능합니다. 즉 `user` 유형이 지원되는 경우 읽기, 편집, 만들기 및 삭제 작업을 사용하여 Identity 저장소에서 사용자 항목을 읽고 편집하고 만들고 삭제할 수 있습니다. `user=service` 작업을 사용하면 Access Manager 서비스가 사용자 항목의 속성에 액세스할 수 있습니다. 사용자가 속한 영역이나 역할에 서비스가 할당된 경우 이 사용자는 동적인 서비스 속성에 액세스할 수 있습니다.

또한 모든 할당된 서비스에 대한 사용자 속성도 관리할 수 있습니다. 사용자에게 `service`가 작업으로 지정(`user=service`)되어 있는 경우 지원되는 모든 서비스 관련 작업을 지정합니다. 이러한 작업으로는 `assignService`, `unassignService`, `getAssignedServices`, `getServiceAttributes`, `removeServiceAttributes` 및 `modifyService`가 있습니다.

LDAPv3 플러그인 검색 범위

LDAPv3 플러그인 항목을 찾는 데 사용되는 범위를 정의합니다. 이 범위는 다음 중 하나가 되어야 합니다.

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB(기본값)

LDAP 사용자 검색 속성

이 필드는 사용자에 대해 검색을 수행하는 속성 유형을 정의합니다. 예를 들어 사용자 DN이 `uid=user1,ou=people,dc=iplanet,dc=com`이면 이름 지정 속성은 `uid`입니다.

LDAP 사용자 검색 필터

사용자 항목을 찾는 데 사용되는 검색 필터를 지정합니다.

LDAP 사용자 객체 클래스

사용자를 위한 객체 클래스를 지정합니다. 사용자가 생성되면 사용자 객체 클래스의 해당 목록이 사용자의 속성 목록에 추가됩니다.

LDAP 사용자 속성

사용자와 연결된 속성의 목록을 정의합니다. 해당 목록에 없는 사용자 속성에 대한 읽기/쓰기 시도는 허용되지 않습니다. 속성은 대소문자를 구분합니다. 객체 클래스 및 속성 스키마는 사용자가 객체 클래스 및 속성 스키마를 지정하기 전에 Directory Server에서 정의되어야 합니다.

LDAP 사용자 만들기 속성 매핑

사용자를 만들 때 필요한 속성을 지정합니다. 이 속성은 다음 구문을 사용합니다.

```
DestinationAttributeName=SourceAttributeName
```

소스 속성 이름이 없는 경우 기본값은 사용자 아이디(`uid`)입니다. 예를 들면 다음과 같습니다.

```
cn
sn=givenName
```

사용자 프로필을 만들려면 `cn`과 `sn`이 모두 필요합니다. `cn`에는 `uid`로 이름 지정된 속성의 값이 들어가고 `sn`에는 `givenName`으로 이름 지정된 속성의 값이 들어갑니다.

사용자 상태 속성

사용자의 상태를 나타내는 속성 이름을 지정합니다.

사용자 상태 활성 값

활성인 사용자 상태에 대한 속성 이름을 지정합니다. 기본값은 `active`입니다.

사용자 상태 비활성 값

비활성인 사용자 상태에 대한 속성 이름을 지정합니다. 기본값은 `inactive`입니다.

LDAP 그룹 검색 속성

이 필드는 그룹에서 검색을 수행하는 속성 유형을 정의합니다. 기본값은 cn입니다.

LDAP 그룹 검색 필터

그룹 항목을 찾는 데 사용되는 검색 필터를 지정합니다. 기본값은 (objectclass=groupOfUniqueNames)입니다.

LDAP 그룹 컨테이너 이름 지정 속성

그룹이 컨테이너에 있는 경우 그룹 컨테이너의 이름 지정 속성을 지정합니다. 그렇지 않으면 이 속성은 비어 있습니다. 예를 들어 cn=group1,ou=groups,dc=iplanet,dc=com의 그룹 DN이 ou=groups에 있는 경우 그룹 컨테이너 이름 지정 속성은 ou입니다.

LDAP 그룹 컨테이너 값

그룹 컨테이너 값을 지정합니다. 예를 들어 cn=group1,ou=groups,dc=iplanet,dc=com의 그룹 DN이 ou=groups에 있는 경우 그룹 컨테이너 값은 groups입니다.

LDAP 그룹 객체 클래스

그룹에 대한 객체 클래스를 지정합니다. 그룹이 생성되면 그룹 객체 클래스의 해당 목록이 그룹의 속성 목록에 추가됩니다.

LDAP 그룹 속성

그룹과 연결된 속성의 목록을 정의합니다. 목록에 없는 그룹 속성에 대한 읽기/쓰기 시도는 허용되지 않습니다. 속성은 대소문자를 구분합니다. 객체 클래스 및 속성 스키마는 사용자가 객체 클래스 및 속성 스키마를 지정하기 전에 Directory Server에서 정의되어야 합니다.

그룹 구성원 속성

DN이 속한 모든 그룹 이름을 값으로 가지는 속성의 이름을 지정합니다. 기본값은 memberOf입니다.

고유 구성원 속성

해당 그룹에 속한 DN을 값으로 가지는 속성의 이름을 지정합니다. 기본값은 uniqueMember입니다.

그룹 구성원 URL 속성

해당 그룹에 속한 구성원을 확인하는 LDAP URL을 값으로 가지는 속성의 이름을 지정합니다. 기본값은 memberUrl입니다.

LDAP 사용자 컨테이너 이름 지정 속성

사용자가 사용자 컨테이너에 있는 경우 사용자 컨테이너의 이름 지정 속성을 지정합니다. 사용자가 사용자 컨테이너에 없으면 이 필드는 비어 있습니다.

LDAP 사용자 컨테이너 값

사용자 컨테이너의 값을 지정합니다. 기본값은 `people`입니다.

LDAP 에이전트 검색 속성

이 필드는 에이전트에서 검색을 수행하는 속성 유형을 정의합니다. 기본값은 `uid`입니다.

LDAP 에이전트 컨테이너 이름 지정 변수

에이전트가 에이전트 컨테이너에 있는 경우 에이전트 컨테이너의 이름을 지정합니다. 에이전트가 에이전트 컨테이너에 없으면 이 필드는 비어 있습니다.

LDAP 에이전트 컨테이너 값

에이전트 컨테이너 값을 지정합니다. 기본값은 `agents`입니다.

LDAP 에이전트 검색 필터

에이전트 검색에 사용되는 필터를 정의합니다. 이 필드에 LDAP 에이전트 검색 변수를 추가하여 실제 에이전트 검색 필터를 만듭니다.

예를 들어 LDAP 에이전트 검색 속성이 `uid`이고 LDAP 사용자 검색 필터는 `(objectClass=sunIdentityServerDevice)`인 경우 실제 사용자 검색 필터는`(&(uid=*)(objectClass=sunIdentityServerDevice))`입니다.

LDAP 에이전트 객체 클래스

에이전트에 대한 객체 클래스를 지정합니다. 에이전트가 생성되면 사용자 객체 클래스가 에이전트의 속성 목록에 추가됩니다.

LDAP 에이전트 속성

에이전트와 연결된 속성의 목록을 정의합니다. 목록에 없는 에이전트 속성에 대한 읽기/쓰기 시도는 허용되지 않습니다. 속성은 대소문자를 구분합니다. 객체 클래스 및 속성 스키마는 사용자가 객체 클래스 및 속성 스키마를 지정하기 전에 Directory Server에서 정의되어야 합니다.

인증 가능한 Identity 유형

영역에 대한 인증 모듈 모드가 데이터 저장소로 설정된 경우 이 데이터 저장소에서 사용자 및 에이전트 Identity 유형을 인증할 수 있음을 지정합니다.

영구 검색 기본 DN

영구 검색에 사용할 기본 DN을 지정합니다. 일부 LDAPv3 서버는 루트 접미사 수준의 영구 검색만 지원합니다.

영구 검색 필터

Directory Server 항목에 대한 특정 변경 내용을 반환할 필터를 정의합니다. 데이터 저장소에서는 정의된 필터와 일치하는 변경 내용만 수신합니다.

시스템 재시작 전의 최대 유휴 시간 영구 검색

영구 검색을 다시 시작하기 전 최대 유휴 시간을 지정합니다. 1보다 큰 값을 사용해야 합니다. 값이 1 이하인 경우 연결 유휴 시간에 관계없이 검색을 다시 시작합니다.

Access Manager가 로드 밸런서와 함께 배포된 경우 지정된 시간 동안 유휴 상태이면 일부 로드 밸런서에서 시간 초과가 발생합니다. 이 경우 [시스템 재시작 전의 최대 유휴 시간 영구 검색]을 로드 밸런서에 지정한 값보다 작은 값으로 설정해야 합니다.

오류 코드 후의 최대 재시도 횟수

[재시도할 LDAP 예외 오류 코드]에서 지정된 오류 코드가 발생하는 경우 영구 검색 작업에 대한 최대 재시도 횟수를 지정합니다.

재시도 사이의 지연 시간

각 재시도 전 대기 시간을 지정합니다. 영구 검색 연결에만 적용됩니다.

재시도할 LDAP 예외 오류 코드

영구 검색 작업을 재시도할 오류 코드를 지정합니다. 이 속성은 영구 검색에만 적용되며 모든 LDAP 작업에 적용되지는 않습니다.

캐싱

이 속성이 활성화되면 Access Manager에서 데이터 저장소에서 가져온 데이터를 캐싱할 수 있습니다.

캐시된 항목의 최대 기간

데이터를 제거하기 전에 캐시에 저장되어 있는 최대 시간을 지정하며이 값은 초 단위로 정의됩니다.

캐시의 최대 크기

캐시의 최대 크기를 지정합니다. 값이 클수록 더 많은 데이터를 저장할 수 있지만 많은 메모리가 필요합니다. 이 값은 바이트 단위로 정의됩니다.

◆ ◆ ◆ 4 장

4 인증 관리

인증 서비스는 Access Manager 배포 시 설치되는 모든 기본 인증 유형에 사용할 웹 기반 사용자 인터페이스를 제공합니다. 이 인터페이스는 액세스 요청한 사용자에게 (호출된 인증 모듈에 따라) 로그인 요구 사항 화면을 표시함으로써 인증 자격 증명을 수집하는 동적/사용자 정의 가능 수단을 제공합니다. 이러한 인터페이스는 Sun Java System™ Application Framework(*JATO*라고도 함)를 사용하여 구축되는데, 이는 개발자들이 기능적 웹 응용 프로그램 구축 시 사용하는 J2EE(Java 2 Enterprise Edition) 표현 프레임워크입니다.

인증 구성

이 절에서는 배포를 위한 인증을 구성하는 방법에 대해 설명합니다. 첫 번째 절에서는 기본 인증 모듈에 대해 간략히 설명하고 필요한 구성 전 지침을 제공합니다. 영역, 사용자, 역할 등에 대해 같은 인증 모듈 유형의 여러 구성 인스턴스를 구성할 수 있습니다. 또한 인증 체인을 추가해서 성공적인 인증을 위해 인증이 여러 인스턴스의 기준을 통과하도록 할 수 있습니다. 이 절에는 다음 내용이 포함되어 있습니다.

인증 모듈 유형

인증 모듈은 사용자 아이디, 비밀번호와 같은 사용자 정보를 수집하고 이 정보를 데이터베이스의 항목과 비교하여 확인하는 플러그인입니다. 사용자가 인증 기준에 맞는 정보를 제공하면 요청한 자원에 대한 액세스 권한이 허용됩니다. 사용자가 인증 기준에 맞지 않는 정보를 제공하면 요청한 자원에 대한 액세스가 거부됩니다. Access Manager는 다음 15가지 유형의 인증 모듈과 함께 설치됩니다.

주 – 일부 인증 모듈의 경우 인증 인스턴스로 사용할 수 있으려면 사전 구성이 필요합니다. 필요한 경우 구성 단계가 모듈 유형 설명에 나옵니다.

핵심

Access Manager는 기본적으로 핵심 인증 모듈과 15개의 다른 인증 모듈을 제공합니다. 핵심 인증 모듈은 인증 모듈에 대한 전체 구성을 제공합니다. 활성 디렉토리, 익명, 인증서 기반, HTTP 기본, JDBC, LDAP 및 인증 모듈을 추가하고 활성화하기 전에 먼저 핵심 인증을 추가하고 활성화해야 합니다. 핵심 및 LDAP 인증 모듈은 모두 기본 영역에 대해 자동으로 활성화됩니다.

[고급 등록 정보] 버튼을 누르면 영역에 대해 정의할 수 있는 핵심 인증 속성이 표시됩니다. 전역 속성은 영역에 적용되지 않으므로 표시되지 않습니다.

활성 디렉토리

활성 디렉토리 인증 모듈은 LDAP 모듈과 비슷한 방법으로 인증을 수행하지만 LDAP 인증 모듈의 Directory Server와 반대되는 Microsoft의 Active Directory™ 서버를 사용합니다. 활성 디렉토리 서버에 대해 LDAP 인증 모듈을 구성할 수는 있지만 이 모듈을 사용하면 LDAP 및 활성 디렉토리 인증이 모두 같은 영역 아래에 있게 됩니다.

주 – 이 릴리스의 경우 활성 디렉토리 인증 모듈만 사용자 인증을 지원합니다. 비밀번호 정책은 LDAP 인증 모듈에서만 지원됩니다.

익명

기본적으로 이 모듈이 활성화되면 사용자는 Access Manager에 **익명** 사용자로 로그인할 수 있습니다. 또한 유효한 익명 사용자 목록 속성을 구성하여 이 모듈에 대한 익명 사용자 목록을 정의할 수 있습니다. 익명 액세스를 허용한다는 것은 비밀번호를 입력하지 않고 액세스할 수 있다는 의미입니다. 특정 액세스 유형(예: 읽기 액세스, 검색 액세스) 또는 디렉토리 내의 개별 항목이나 특정 하위 트리로 익명 액세스를 제한할 수 있습니다.

인증서

인증서 기반 인증에는 PDC(Personal Digital Certificate)를 사용한 사용자 식별 및 인증이 포함됩니다. Directory Server에 저장된 PDC에 대한 일치 및 인증서 해지 목록에 대한 확인을 수행하도록 PDC를 구성할 수 있습니다.

인증서 기반 인증 모듈을 영역에 추가하기 전에 여러가지 작업을 수행해야 합니다. 먼저, Access Manager와 함께 설치되는 웹 컨테이너를 보호하고 인증서 기반 인증에 맞게 구성해야 합니다.

주 – SSL 사용 가능 Sun Java System Web Server 6.1 인스턴스를 사용하여 Access Manager 인증서 인증을 구성하고 인증서 기반 인증 요청 및 비인증서 기반 인증 요청을 모두 수락하도록 Web Server를 정의하려면 Web Server의 `obj.conf` 파일에 다음 값을 설정해야 합니다.

```
PathCheck fn="get-client-cert" dorequest="1" require="0"
```

이는 이 동작에 대한 선택적 속성을 설정할 때 Web Server 콘솔의 제한 사항으로 인한 것입니다.

인증서 기반 모듈을 활성화하기 전에 이 초기 Web Server 구성 단계에 대한 내용을 보려면 **Sun ONE Web Server 6.1 관리자 설명서**의 6장, “인증서 및 키 사용”을 참조하십시오. 이 문서는 다음 위치에서 확인할 수 있습니다.

```
http://docs.sun.com/db/prod/s1websrv#hic
```

또는 다음 위치에서 **Sun ONE Application Sever 관리자 보안 설명서**를 참조하십시오.

http://docs.sun.com/db/prod/s1appsrv#hic (`http://docs.sun.com/db/prod/s1appsrv#hic`)

주 – 인증서 기반 모듈을 사용하여 인증할 각 사용자는 사용자의 브라우저에 대한 PDC를 요청해야 합니다. 지침은 사용되는 브라우저에 따라 다릅니다. 자세한 내용은 해당 브라우저의 설명서를 참조하십시오.

이 모듈을 추가하려면 Access Manager에 영역 관리자로 로그인해야 하며 Access Manager와 웹 컨테이너에서 SSL을 구성하고 클라이언트 인증을 활성화해야 합니다. 자세한 내용은 **Access Manager Post Installation Guide**의 Configuring Access Manager in SSL Mode를 참조하십시오.

데이터 저장소

데이터 저장소 인증 모듈을 사용하면 영역의 Identity 저장소를 사용한 로그인 시 사용자를 인증할 수 있습니다. 데이터 저장소 모듈을 사용하면 같은 데이터 저장소에 대해 인증해야 하는 경우에도 인증 플러그인 모듈을 작성하고 인증 모듈을 로드 및 구성할 필요가 없습니다. 또한 해당 영역에서 플랫 파일 인증이 각 저장소에 필요한 경우 사용자 정의 인증 모듈을 작성할 필요가 없습니다.

Access Manager 인증을 구성할 때 이 인증 유형을 사용하면 어느 정도 편리합니다. Access Manager 7.1 이전 릴리스에서는 LDAPv3 데이터 저장소의 사용자가 해당 영역에서 인증되려면 다음을 수행해야 했습니다.

- LDAPv3 데이터 저장소 구성
- 동일한 영역 주제를 참조하도록 LDAP 인증 모듈 인스턴스 구성

데이터 저장소 인증 모듈을 사용하면 영역의 Identity 저장소에 정의된 사용자를 인증할 수 있습니다. 이 경우 어떤 LDAP 인증 구성도 필요하지 않습니다. 예를 들어 영역의 Identity 저장소에 LDAPv3 데이터 저장소가 있고 같은 영역에서 데이터 저장소 인증을 사용한다고 가정해 보겠습니다. 이 경우 Identity 저장소에 정의된 사용자는 모두 해당 영역에서 인증될 수 있습니다.

HTTP 기본

이 모듈은 HTTP 프로토콜에서 지원하는 기본 제공 인증인 기본 인증을 사용합니다. Web Server는 아이디 및 비밀번호에 대한 클라이언트 요청을 발급하고, 해당 정보를 인증된 요청에 포함하여 서버로 다시 보냅니다. Access Manager는 사용자 아이디와 비밀번호를 수신한 다음 LDAP 인증 모듈에 대해 사용자를 내부적으로 인증합니다. HTTP 기본이 제대로 작동하게 하려면 LDAP 인증 모듈을 추가해야 합니다(HTTP 기본 모듈만 추가하면 작동되지 않음). 성공적으로 인증한 사용자는 사용자 아이디와 비밀번호를 묻는 메시지를 표시하지 않고 다시 인증할 수 있습니다.

JDBC

JDBC(Java Database Connectivity) 인증 모듈은 Access Manager가 JDBC 기술 사용 드라이버를 제공하는 SQL 데이터베이스를 통해 사용자를 인증하는 기법을 지원합니다. SQL 데이터베이스는 JDBC 드라이버나 JNDI 연결 풀을 통해 직접 연결할 수 있습니다.

주 – 이 모듈은 MySQL4.0 및 Oracle 8i에서 테스트되었습니다.

LDAP

LDAP 인증 모듈에서는 사용자가 로그인할 때 특정 사용자 DN 및 비밀번호를 사용하여 LDAP Directory Server에 바인드해야 합니다. 이는 모든 영역 기반 인증에 대한 기본 인증 모듈입니다. 사용자는 Directory Server에 있는 사용자 아이디와 비밀번호를 입력하여 유효한 Access Manager 세션에 액세스할 수 있으며 해당 세션을 사용하여 사용자를 설정할 수 있습니다. 기본 영역에서는 핵심 및 LDAP 인증 모듈을 모두 자동으로 사용할 수 있게 됩니다.

구성원

구성원 인증은 `my.site.com`, `mysun.sun.com` 등과 같은 사용자 설정 사이트와 비슷하게 구현됩니다. 이 모듈이 사용 가능한 경우 사용자는 관리자의 도움 없이 계정을 만들어 사용자 설정할 수 있습니다. 사용자는 이 새 계정에 추가된 사용자로 액세스할 수 있습니다. 또한, 사용자 프로필 데이터베이스에 인증 데이터 및 사용자 기본 설정으로 저장된 뷰어 인터페이스에 액세스할 수 있습니다.

MSISDN

MSISDN(Mobile Station Integrated Services Digital Network) 인증 모듈을 사용하면 휴대전화와 같은 장치의 이동 가입자 ISDN을 사용하여 인증할 수 있습니다. 이 모듈은 비대화식 모듈입니다. 가입자 ISDN을 검색하고 이를 Directory Server에서 검증하여 번호에 맞는 사용자를 찾습니다.

RADIUS

Access Manager를 구성하여 이미 설치된 RADIUS 서버에서 작업할 수 있습니다. 이렇게 하면 회사에서 레거시 RADIUS 서버를 사용하여 인증하는 경우에 유용합니다. RADIUS 인증 모듈을 활성화하려면 다음 2단계 프로세스를 거쳐야 합니다.

1. RADIUS 서버를 구성합니다.

   자세한 내용은 RADIUS 서버 설명서를 참조하십시오.
2. RADIUS 인증 모듈을 등록하여 사용 가능하게 합니다.

Sun Java System Application Server에서 RADIUS 구성

RADUIS 클라이언트에서 이 서버에 소켓 연결을 수행할 경우 기본적으로 Application Server의 `server.policy` 파일에 SocketPermissions 연결 권한만 허용됩니다. RADUIS 인증이 제대로 작동하게 하려면 다음 작업에 대한 권한을 허용해야 합니다.

- 적용
- 연결
- 수신
- 결정

소켓 연결 권한을 부여하려면 Application Server의 `server.policy` 파일에 항목을 추가해야 합니다. SocketPermission은 호스트 사양과 해당 호스트에 연결하는 방법을 지정하는 작업 집합으로 구성됩니다. 호스트를 지정하는 구문은 다음과 같습니다.

```
host = hostname | IPaddress:portrange:portrange = portnumber
| -portnumberportnumber-portnumber
```

host는 DNS 이름, 숫자 IP 주소 또는 로컬 호스트(로컬 시스템의 경우)로 표현됩니다. 와일드카드 "*"는 DNS 이름 호스트 규격에 한 번 포함될 수 있습니다. 와일드카드가 포함되는 경우 가장 왼쪽 위치(예: `*.example.com`)에 와일드카드가 있어야 합니다.

포트(또는 포트 범위)는 선택 사항입니다. `N-` 형식의 포트 사양은 번호가 `N` 이상인 모든 포트를 나타냅니다. 여기서 `N`은 포트 번호입니다. `-N` 형식의 사양은 번호가 `N` 이하인 모든 포트를 나타냅니다.

수신 작업은 로컬 호스트에서 사용될 때만 적용됩니다. **결정**(호스트/IP 이름 서비스 조회 결정) 작업은 다른 작업이 있을 때 적용됩니다.

예를 들어, SocketPermissions을 만들 때 일부 코드에 다음 권한이 허용되는 경우 해당 코드를 `machine1.example.com`의 `port 1645`에 연결하고 해당 포트에서 연결을 적용할 수 있습니다.

```
permission java.net.SocketPermission machine1.example.com:1645, "connect,accept";
```

마찬가지로 일부 코드에 다음 권한이 허용되는 경우 해당 코드를 사용하여 로컬 호스트에서 1024에서 65535 사이의 포트에서 연결을 적용, 연결 또는 수신할 수 있습니다.

```
permission java.net.SocketPermission "machine1.example.com:1645", "connect,accept";
permission java.net.SocketPermission "localhost:1024-", "accept,connect,listen";
```

주 – 원격 호스트에 연결을 적용하거나 연결하도록 코드 권한을 허용하면 유해 코드를 통해 해당 데이터에 대한 액세스 권한이 없는 당사자 간에 기밀 데이터를 쉽게 전송 및 공유할 수 있기 때문에 문제가 발생할 수 있습니다. 포트 번호의 범위 대신 정확한 포트 번호를 지정하여 해당 사용 권한만 부여해야 합니다.

SafeWord

Access Manager를 구성하여 Secure Computing의 SafeWord™ 또는 SafeWord PremierAccess™ 인증 서버에 대한 SafeWord 인증 요청을 처리할 수 있습니다. Access Manager는 SafeWord 인증의 클라이언트 부분을 제공합니다. SafeWord 서버는 Access Manager가 설치되는 시스템이나 별도의 시스템에 위치할 수 있습니다.

Sun Java System Application Server에서 SafeWord 구성

SafeWord 클라이언트에서 이 서버에 소켓 연결을 수행할 경우 기본적으로 Application Server'의 `server.policy` 파일에 SocketPermissions **연결** 권한만 허용됩니다. SafeWord 인증이 제대로 작동하게 하려면 다음 작업에 대한 권한을 허용해야 합니다.

- 적용
- 연결
- 수신
- 결정

소켓 연결 권한을 부여하려면 Application Server의 `server.policy` 파일에 항목을 추가해야 합니다. SocketPermission은 호스트 사양과 해당 호스트에 연결하는 방법을 지정하는 작업 집합으로 구성됩니다. 호스트를 지정하는 구문은 다음과 같습니다.

```
host = (hostname | IPaddress)[:portrange] portrange =
portnumber | -portnumberportnumber-[portnumber]
```

host는 DNS 이름, 숫자 IP 주소 또는 로컬 호스트(로컬 시스템의 경우)로 표현됩니다. 와일드카드 "*"는 DNS 이름 호스트 규격에 한 번 포함될 수 있습니다. 와일드카드가 포함되는 경우 가장 왼쪽 위치(예: `*.example.com`)에 와일드카드가 있어야 합니다.

포트(또는 포트 범위)는 선택 사항입니다. `N-` 형식의 포트 사양은 번호가 `N` 이상인 모든 포트를 나타냅니다. 여기서 `N`은 포트 번호입니다. `-N` 형식의 사양은 번호가 `N` 이하인 모든 포트를 나타냅니다.

수신 작업은 로컬 호스트에서 사용될 때만 적용됩니다. **결정**(호스트/IP 이름 서비스 조회 결정) 작업은 다른 작업이 있을 때 적용됩니다.

예를 들어, SocketPermissions을 만들 때 일부 코드에 다음 권한이 허용되는 경우 해당 코드를 `machine1.example.com`의 `port 1645`에 연결하고 해당 포트에서 연결을 적용할 수 있습니다.

```
permission java.net.SocketPermission machine1.example.com:5030, "connect,accept";
```

마찬가지로 일부 코드에 다음 권한이 허용되는 경우 해당 코드를 사용하여 로컬 호스트에서 1024에서 65535 사이의 포트에서 연결을 적용, 연결 또는 수신할 수 있습니다.

```
permission java.net.SocketPermission "machine1.example.com:5030", "connect,accept";
permission java.net.SocketPermission "localhost:1024-", "accept,connect,listen";
```

주 – 원격 호스트에 연결을 적용하거나 연결하도록 코드 권한을 허용하면 유해 코드를 통해 해당 데이터에 대한 액세스 권한이 없는 당사자 간에 기밀 데이터를 쉽게 전송 및 공유할 수 있기 때문에 문제가 발생할 수 있습니다. 포트 번호의 범위 대신 정확한 포트 번호를 지정하여 해당 사용 권한만 부여해야 합니다.

SAML

SAML(Security Assertion Markup Language) 인증 모듈은 대상 서버에서 SAML 명제를 받아 검증합니다. SAML SSO는 업그레이드(예: Access Manager 2005Q4를 Access Manager 7.1로) 이후를 포함하여 이 모듈을 대상 컴퓨터에 구성한 경우에만 작동됩니다.

SecurID

Access Manager를 구성하여 RSA의 ACE/Server 인증 서버에 대한 SecurID 인증 요청을 처리할 수 있습니다. Access Manager는 SecurID 인증의 클라이언트 부분을 제공합니다. ACE/Server는 Access Manager가 설치되는 시스템이나 별도의 시스템에 위치할 수 있습니다. 로컬로 관리되는 사용자 아이디(`admintool`(1M) 참조)를 인증하려면 루트로 액세스해야 합니다.

SecurID 인증에서는 `amsecuridd` 인증 **도우미**가 사용됩니다. 이 프로세스는 Access Manager 주 프로세스와 다른 별도의 프로세스입니다. 시작 시에 이 도우미는 하나의 포트에서 구성 정보를 수신합니다. Access Manager를 설치하여 `nobody` 또는 루트가 아닌 사용자 아이디로 실행할 경우에도 *AccessManager-base*`/SUNWam/share/bin/amsecuridd` 프로세스는 여전히 루트로 실행되어야 합니다. `amsecuridd` 도우미에 대한 자세한 내용은 Access Manager Administration Reference의 "The amSecurID Helper"를 참조하십시오.

주 – 이 릴리스의 Access Manager에서는 Linux 또는 Solaris x86 플랫폼에 대해 SecurID 인증 모듈을 사용할 수 없으며, 이 두 플랫폼에서 등록, 구성 및 사용해서는 안됩니다. SPARC 시스템에만 사용할 수 있습니다.

UNIX

Access Manager를 구성하여 Access Manager가 설치된 Solaris 또는 Linux 시스템에 알려진 Unix 사용자 아이디와 비밀번호에 대한 인증 요청을 처리할 수 있습니다. 영역 속성은 하나만 있지만 Unix 인증을 위한 전역 속성이 여러 개인 경우 몇 가지 시스템 고려 사항이 있습니다. 로컬로 관리되는 사용자 아이디(`admintool`(1M) 참조)를 인증하려면 루트로 액세스해야 합니다.

Unix 인증에서는 `amunixd` 인증 **도우미**가 사용됩니다. 이 프로세스는 Access Manager 주 프로세스와 다른 별도의 프로세스입니다. 시작 시에 이 도우미는 하나의 포트에서 구성 정보를 수신합니다. 각 Access Manager에는 모든 영역에 서비스를 제공하는 Unix 도우미가 하나씩만 있습니다.

Access Manager를 설치하여 `nobody` 또는 루트가 아닌 사용자 아이디로 실행할 경우에도 *AccessManager-base*`/SUNWam/share/bin/amunixd` 프로세스는 여전히 루트로 실행되어야 합니다. Unix 인증 모듈은 `localhost:58946`에 대한 소켓을 열어 `amunixd` 데몬을 호출하여 Unix 인증 요청을 수신합니다. 기본 포트에서 `amunixd` 도우미 프로세스를 실행하려면 다음 명령을 입력합니다.

```
./amunixd
```

기본 포트가 아닌 포트에서 `amunixd`를 실행하려면 다음 명령을 입력합니다.

```
./amunixd [-c portnm] [ipaddress]
```

ipaddress 및 portnumber는 AMConfig.properties의 UnixHelper.ipadrs(IPV4 형식) 및 UnixHelper.port 속성에 있습니다. amserver 명령줄 유틸리티를 통해 amunixd를 실행할 수 있습니다. amserver는 프로세스를 자동으로 실행하여 AMConfig.properties에서 포트 번호와 IP 주소를 검색합니다.

/etc/nsswitch.conf 파일의 passwd 항목에 따라 인증에 /etc/passwd 및 /etc/shadow 파일을 참조하는지 NIS를 참조하는지가 결정됩니다.

Windows 데스크탑 SSO

Windows 데스크탑 SSO 인증 모듈은 Windows 2000™에 사용되는 커버로스 기반 인증 플러그인 모듈입니다. 이 모듈을 사용하면 KDC(Kerberos Distribution Center)에 대해 이미 인증을 받은 사용자는 로그인 조건(단일 사인온)을 다시 제출하지 않고도 Access Manager에 대해 인증을 받을 수 있습니다.

사용자는 SPNEGO(Simple and Protected GSS-API Negotiation Mechanism) 프로토콜을 통해 Access Manager에 커버로스 토큰을 제공합니다. 이 인증 모듈을 통해 Access Manager에 커버로스 기반 단일 사인온을 수행하려면 클라이언트측에서 사용자를 인증하도록 SPNEGO 프로토콜을 지원해야 합니다. 일반적으로 이 프로토콜을 지원하는 사용자는 이 모듈을 사용하여 Access Manager에 인증할 수 있습니다. 클라이언트측 토큰의 가용성에 따라 이 모듈은 SPENGO 토큰 또는 커버로스 토큰(두 경우 모두 동일한 프로토콜)을 제공합니다. Windows 2000 이상에서 실행되는 Microsoft Internet Explorer(5.01 이상)는 현재 이 프로토콜을 지원합니다. 또한 Solaris(9 및 10)의 Mozilla 1.4도 SPNEGO 지원 기능이 있지만 Solaris에서 SPNEGO를 지원하지 않으므로 반환되는 토큰은 커버로스 토큰뿐입니다.

주 – 커버로스 V5 인증 모듈의 새 기능을 활용하려면 JDK 1.4 이상을 사용하고 이 SPNEGO 모듈에서 커버로스 기반 SSO를 수행하려면 Java GSS API를 사용해야 합니다.

Internet Explorer의 알려진 제한 사항

WindowsDesktopSSO 인증용으로 Microsoft Internet Explorer 6.x를 사용하고, 브라우저에 WindowsDesktopSSO 모듈에서 구성된 KDC 영역과 일치하는 사용자의 커버로스/SPNEGO 토큰에 대한 액세스 권한이 없는 경우 브라우저가 WindowsDesktopSSO 모듈 인증에 실패하면 다른 모듈에 대해 올바르게 작동하지 않습니다. 이 문제의 직접적인 원인은 Internet Explorer에서 WindowsDesktopSSO 모듈 실패 후 다른 모듈의 콜백이 프롬프트에 표시되는 경우에도 브라우저에서 이를 Access Manager에 전달할 수 없기 때문이며, 이러한 불능 상태는 브라우저를 다시 시작할 때까지 계속됩니다. 즉 null 사용자 자격 증명 때문에 WindowsDesktopSSO 실패 후 수신한 모든 모듈도 실패합니다.

자세한 내용은 다음 설명서를 참조하십시오.

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074
(http://support.microsoft.com/default.aspx?scid=kb;en-us;308074)

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM
(http://support.microsoft.com/default.aspx?scid=kb;en-us;308074)

주 – 이번 Access Manager 릴리스부터 Microsoft에서 이러한 제한 사항을 해결했습니다. 자세한 내용은 다음 설명서를 참조하십시오.

http:// http://www.microsoft.com/technet/security/bulletin/ms06-042.mspx

---

Windows 데스크탑 SSO 구성

Windows 데스크탑 SSO 인증을 활성화하는 2단계 프로세스는 다음과 같습니다.

1. Windows 2000 도메인 컨트롤러에서 사용자를 만듭니다.
2. Internet Explorer를 설정합니다.

▼ Windows 2000 도메인 컨트롤러에서 사용자를 만들려면

1 도메인 컨트롤러에서 Access Manager 인증 모듈에 사용할 사용자 계정을 만듭니다.

a. 시작 메뉴에서 프로그램>관리 도구로 이동합니다.

b. 활성 디렉토리 사용자 및 컴퓨터를 선택합니다.

c. [컴퓨터]>[새로 만들기]>[컴퓨터]로 이동하고 클라이언트 컴퓨터의 이름을 추가합니다. Windows XP를 사용하는 경우 이 단계는 도메인 컨트롤러 계정 구성 시 자동으로 수행됩니다.

d. [사용자]>[새로 만들기]>[사용자]로 이동한 다음 Access Manager 호스트 이름을 사용하여 새 사용자를 사용자 아이디(로그인 이름)로 만듭니다. Access Manager 호스트 이름에는 도메인 이름이 포함되지 않아야 합니다.

2 사용자 계정을 서비스 공급자 이름과 연결하고 Access Manager가 설치된 시스템으로 키탭 파일을 내보냅니다. 그러려면 다음 명령을 실행합니다.

```
ktpass -princ host/hostname.domainname@DCDOMAIN -pass password -mapuser userName-out
hostname.host.keytab
ktpass -princ HTTP/hostname.domainname@DCDOMAIN -pass
password -mapuser userName-out hostname
.HTTP.keytab
```

주 – ktpass 유틸리티는 Windows 2000 서버의 일부로 설치되지 않으므로 설치 CD에서 c:\program files\support 도구 디렉토리로 설치해야 합니다.

---

ktpass 명령에는 다음과 같은 매개 변수가 사용됩니다.

hostname. Access Manager를 실행하는 호스트 이름(도메인 이름 없음)입니다.

domainname. Access Manager 도메인 이름입니다.

DCDOMAIN. 도메인 컨트롤러의 도메인 이름입니다. Access Manager 도메인 이름과 다를 수 있습니다.

password . 사용자 계정의 비밀번호입니다. `ktpass`에서는 비밀번호를 확인하지 않으므로 비밀번호가 정확한지 확인합니다.

userName. 사용자 계정 아이디입니다. 호스트 이름과 같아야 합니다.

주 – 두 키탭 파일이 모두 안전하게 보존되는지 확인합니다.

서비스 템플리트 값은 다음 예와 비슷해야 합니다.

서비스 기본: `HTTP/machine1.EXAMPLE.COM@ISQA.EXAMPLE.COM`

키탭 파일 이름:`/tmp/machine1.HTTP.keytab`

커버로스 영역: `ISQA.EXAMPLE.COM`

커버로스 서버 이름:`machine2.EXAMPLE.com`

도메인 이름과 함께 기본 반환:`false`

인증 수준:22

주 – Windows 2003 또는 Windows 2003 서비스 팩을 사용하는 경우 다음 `ktpass` 명령 구문을 사용합니다.

```
ktpass /out filename /mapuser username /princ HTTP/hostname.domainname
     /crypto encryptiontype /rndpass /ptype principaltype /target domainname
```

예를 들면 다음과 같습니다.

```
ktpass /out demo.HTTP.keytab /mapuser http
/princ HTTP/demo.identity.sun.com@IDENTITY.SUN.COM /crypto RC4-HMAC-NT
/rndpass /ptype KRB5_NT_PRINCIPAL /target IDENTITY.SUN.COM
```

구문 정의에 대한 자세한 내용은 `http://technet2.microsoft.com/WindowsServer/en/library/64042138-9a5a-4981-84e9-d576a8db0d051033.mspx?mfr=true` 웹 사이트를 참조하십시오.

---

3 **서버를 다시 시작합니다.**

▼ Internet Explorer를 설정하려면

이 단계는 Microsoft Internet Explorer™ 6 이상에 적용됩니다. 이전 버전을 사용하는 경우 Access Manager가 브라우저의 인터넷 영역에 있고 고유한 Windows 인증을 사용하는지 확인하십시오.

1 **[도구] 메뉴에서 [인터넷 옵션] > [고급/보안] > [보안]으로 이동합니다.**

2 **[통합된 Windows 인증 사용] 옵션을 선택합니다.**

3 **[보안] > [로컬 인터넷]으로 이동합니다.**

   a. **[사용자 지정 수준]을 선택합니다. [사용자 인증/로그온] 창에서 [인트라넷 영역에서만 자동으로 로그온] 옵션을 선택합니다.**

   b. **사이트로 가서 옵션을 모두 선택합니다.**

   c. **[고급]을 누르고 로컬 영역에 Access Manager를 추가합니다(아직 추가되지 않은 경우).**

Windows NT

이미 설치된 Windows NT/Windows 2000 서버에서 작업하도록 Access Manager를 구성할 수 있습니다. Access Manager는 NT 인증의 클라이언트 부분을 제공합니다.

1. NT 서버를 구성합니다. 자세한 내용은 Windows NT 서버 설명서를 참조하십시오.
2. Windows NT 인증 모듈을 추가하여 사용 가능하게 하려면 Solaris 시스템의 Access Manager와 통신하도록 Samba 클라이언트를 설치해야 합니다.

Samba 클라이언트 설치

Windows NT 인증 모듈을 활성화하려면 Samba Client 2.2.2를 다운로드하여 다음 디렉토리에 설치해야 합니다.

```
AccessManager-base/SUNWam/bin
```

Samba 클라이언트는 별도의 Windows NT/2000 Server 없이도 Windows 시스템과 UNIX 시스템을 혼합하여 사용할 수 있는 파일 및 인쇄 서버입니다. 자세한 내용을 보거나 Samba 클라이언트를 다운로드하려면 http://wwws.sun.com/software/download/products/3e3af224.html에 액세스하십시오.

Red Hat Linux에서는 Samba 클라이언트가 다음 디렉토리에 제공됩니다.

```
/usr/bin
```

Linux용 Windows NT 인증 모듈을 사용하여 인증하려면 다음 Access Manager 디렉토리에 클라이언트 바이너리를 복사합니다.

AccessManager-base/sun/identity/bin

주 – 인터페이스가 여러 개인 경우에는 추가 구성이 필요합니다. smb.conf 파일의 구성에서 여러 인터페이스를 설정할 수 있으므로 mbclient로 전달됩니다.

인증 모듈 인스턴스

기본 인증 모듈을 기반으로 영역에 대해 여러 인증 모듈 인스턴스를 만들 수 있습니다. 개별적으로 구성된 같은 인증 모듈의 여러 인스턴스를 추가할 수 있습니다.

▼ 새 인증 모듈 인스턴스 만들기

1 **새 인증 모듈 인스턴스를 추가할 영역의 이름을 누릅니다.**

2 **[인증] 탭을 선택합니다.**

주 – [관리자 인증 구성] 버튼은 관리자에 대해서만 인증 서비스를 정의합니다. 관리자의 인증 모듈이 최종 사용자의 모듈과 달라야 하는 경우 이 속성을 사용할 수 있습니다. 이 속성에 구성된 모듈은 Access Manager 콘솔에 액세스할 때 선택됩니다.

---

3 **모듈 인스턴스 목록에서 [새로 만들기]를 누릅니다.**

4 **인증 모듈 인스턴스의 이름을 입력합니다. 이름은 고유해야 합니다.**

5 **영역에 대한 인증 모듈의 유형을 선택합니다.**

6 **[만들기]를 누릅니다.**

7 **새로 만든 모듈 인스턴스의 이름을 누르고 해당 모듈의 등록 정보를 편집합니다. 각 모듈 유형의 등록 정보에 대한 정의는 온라인 도움말의 인증 절을 참조하십시오.**

8 **이러한 단계를 반복하여 여러 개의 모듈 인스턴스를 추가합니다.**

인증 연결

인증을 하나 이상 구성할 수 있으므로 사용자는 모든 인증에 인증 자격 증명을 전달해야 합니다. 이를 **인증 연결**이라고 합니다. Access Manager의 인증 연결은 인증 서비스에 통합된 JAAS 프레임워크를 사용하여 수행됩니다.

▼ 새 인증 체인 만들기

1 **새 인증 체인을 추가할 영역의 이름을 누릅니다.**

2 **[인증] 탭을 선택합니다.**

3 **인증 연결 목록에서 [새로 만들기]를 누릅니다.**

4 **인증 체인의 이름을 입력합니다.**

5 **[만들기]를 누릅니다.**

6 **[추가]를 눌러 체인에 포함할 인증 모듈 인스턴스를 정의합니다. 이를 수행하려면 인스턴스 목록에서 모듈 인스턴스 이름을 선택합니다. 이 목록에 표시된 모듈 인스턴스 이름은 모듈 인스턴스 속성에서 만들어집니다.**

7 **체인의 기준을 선택합니다. 이러한 플래그는 플래그가 정의된 인증 모듈에 대한 적용 기준을 설정하며적용을 위한 계층이 있습니다.[필수]가 가장 높고[옵션]이 가장 낮습니다.**

필요 모듈 인스턴스가 성공적이어야 합니다. 성공한 경우 인증 연결 목록의 그 다음 항목에 대해 인증이 계속됩니다. 실패한 경우 제어가 응용 프로그램에 즉시 반환되며 인증 연결 목록의 그 다음 항목에 대해 인증이 진행되지 않습니다.

필수 이 모듈에 대한 인증이 성공적이어야 합니다. 체인의 필수 모듈 중 하나라도 실패하면 결과적으로 전체 인증 체인이 실패합니다. 그러나 필수 모듈이 성공하든 실패하든 제어는 체인에서 그 다음 모듈에 대해 계속 진행됩니다.

충분 모듈 인스턴스가 반드시 성공적이지 않아도 됩니다. 성공한 경우 제어가 즉시 응용 프로그램에 반환되며 인증 모듈 목록의 그 다음 항목에 대해 인증이 진행되지 않습니다. 실패한 경우 인증 연결 목록의 그 다음 항목에 대해 인증이 계속됩니다.

옵션 모듈 인스턴스가 반드시 성공적이지 않아도 됩니다. 성공 또는 실패한 경우 인증 연결 목록의 그 다음 항목에 대해 인증이 계속 진행됩니다.

8 **체인에 대한 옵션을 입력합니다. 이렇게 하면 키=값 쌍으로 모듈에 대한 추가 옵션을 허용합니다. 여러 옵션을 사용할 경우 공백으로 구분합니다.**

9 **다음 속성을 정의합니다.**

성공한 로그인 URL	인증 성공 시 사용자가 리디렉션되는 URL을 지정합니다.
실패한 로그인 URL	인증 실패 시 사용자가 리디렉션되는 URL을 지정합니다.
인증 사후 처리 클래스	로그인 성공 또는 실패 후에 인증 사후 처리를 사용자 정의하는 데 사용되는 Java 클래스의 이름을 정의합니다.

10 [저장]을 누릅니다.

인증 유형

인증 서비스는 여러 가지 인증 적용 방법을 제공합니다. 로그인 URL 매개 변수를 지정하거나 인증 API 를 통해 이러한 인증 방법에 액세스할 수 있습니다. 자세한 내용은 **Sun Java System Access Manager 7.1 Developer's Guide**의 2 장, "Using Authentication APIs and SPIs"를 참조하십시오. 인증 모듈을 구성하기 전에 특정 인증 모듈 이름을 포함하도록 핵심 인증 서비스 속성인 영역 인증 모듈을 수정해야 합니다.

인증 구성 서비스는 다음 인증 유형에 대한 인증 모듈을 정의하는 데 사용됩니다.

- 59 페이지 "영역 기반 인증"
- 61 페이지 "조직 기반 인증"
- 63 페이지 "역할 기반 인증"
- 66 페이지 "서비스 기반 인증"
- 69 페이지 "사용자 기반 인증"
- 71 페이지 "인증 수준 기반 인증"
- 73 페이지 "모듈 기반 인증"

이러한 인증 유형 중 하나에 대해 인증 모듈을 정의한 경우, 인증 프로세스의 성공 또는 실패 여부에 따라 사후 처리 Java 클래스 사양뿐만 아니라 리디렉션 URL을 제공하도록 해당 모듈을 구성할 수 있습니다.

인증 유형에 따른 액세스 결정 방법

이러한 방법마다 사용자 인증이 성공하기도 하고 실패하기도 합니다. 그러나 방법이 결정된 다음에는 다음 절차를 따르게 됩니다. 1-3단계는 인증 성공 후에 나타나고, 4단계는 인증 성공과 인증 실패 후에 나타납니다.

1. Access Manager는 인증된 사용자가 Directory Server 데이터 저장소에 정의되어 있고 프로필이 활성 상태인지 여부를 확인합니다.

   핵심 인증 모듈의 사용자 프로필 속성은 **필수**, **동적**, **동적 (사용자 별칭과 함께 사용)** 또는 **무시** 중 하나로 정의할 수 있습니다. 인증 성공 후 Access Manager는 인증된 사용자가 Directory Server 데이터 저장소에 정의되어 있는지 확인하고, 사용자 프로필 값이 **필수**인 경우 해당 프로필이 활성 상태인지 확인합니다(기본적인 경우). 사용자 프로필이 **동적으로 구성된** 경우에는 인증 서비스에서 Directory Server 데이터 저장소에 사용자 프로필을 작성합니다. 사용자 프로필이 **무시**로 설정되면 사용자 검증이 수행되지 않습니다.
2. 인증 사후 처리 SPI의 실행이 완료되었습니다.

핵심 인증 모듈에는 인증 사후 처리 클래스 이름이 자체 값으로 포함되는 인증 사후 처리 클래스 속성이 들어 있습니다. `AMPostAuthProcessInterface`는 사후 처리 인터페이스로서인증 성공/실패 시 또는 로그아웃 시 실행될 수 있습니다.

3. 다음 등록 정보가 세션 토큰에 추가되거나 세션 토큰에서 업데이트된 후 사용자의 세션이 활성화됩니다.

   realm. 사용자가 속한 영역의 DN입니다.

   Principal. 사용자의 DN입니다.

   Principals. 사용자가 인증한 이름의 목록입니다. (이 등록 정보에는 세로줄(|)로 구분한 목록으로 정의된 둘 이상의 값이 있을 수 있습니다.)

   UserId. 모듈에서 반환하는 사용자의 DN이거나 LDAP 또는 구성원 이외의 모듈의 경우 사용자 이름입니다. (모든 Principal은 동일한 사용자에 매핑되어야 합니다. UserID는 Principal이 매핑되는 사용자 DN입니다.)

   주 – 이 등록 정보는 DN이 아닌 값일 수 있습니다.

   UserToken. 사용자 이름입니다. (모든 Principal은 동일한 사용자에 매핑되어야 합니다. UserToken은 Principal이 매핑된 사용자 이름입니다.)

   Host. 클라이언트의 호스트 이름이나 IP 주소입니다.

   authLevel. 사용자의 최고 인증 수준입니다.

   AuthType. 사용자가 인증한 인증 모듈을 세로줄(|)로 구분한 목록(예: `module1|module2|module3`)입니다.

   clientType. 클라이언트 브라우저의 장치 유형입니다.

   Locale. 클라이언트의 로켈입니다.

   CharSet. 클라이언트에 대해 설정된 문자 집합입니다.

   Role. 역할 기반의 인증에만 적용될 수 있으며 사용자가 속한 역할입니다.

   Service. 서비스 기반의 인증에만 적용될 수 있으며 사용자가 속한 서비스입니다.

4. Access Manager에서는 인증 성공 또는 실패 후 사용자를 리디렉션할 위치에 대한 정보를 찾습니다.

   URL 리디렉션 위치는 Access Manager 페이지 또는 URL이 될 수 있습니다. 리디렉션은 Access Manager가 인증 방법을 기준으로 찾은 리디렉션의 우선 순위와 해당 인증이 성공 또는 실패했는지 여부에 따라 달라집니다. 이러한 순서는 다음 인증 방법 절에서 URL 리디렉션 부분에 자세히 설명되어 있습니다.

URL 리디렉션

인증 구성 서비스에서 성공적인 인증 또는 실패한 인증에 대한 URL 리디렉션을 할당할 수 있습니다. URL은 이 서비스의 로그인 성공 URL 및 로그인 실패 URL 속성에 자동으로 정의됩니다. URL 리디렉션을 사용 가능하게 하려면 영역에 인증 구성 서비스를 추가하여 해당 서비스를 역할, 영역 또는 사용자에 대해 구성 가능하게 만들어야 합니다. 인증 구성 서비스를 추가할 경우 LDAP - 필수와 같은 인증 모듈을 추가해야 합니다.

영역 기반 인증

이 인증 방법을 사용하면 영역 또는 하위 영역에 대해 인증할 수 있습니다. Access Manager에 대한 기본 인증 방법입니다. 영역 인증 방법은 핵심 인증 모듈을 영역에 등록하고 영역 인증 구성 속성을 정의함으로써 설정됩니다.

영역 기반 인증 로그인 URL

인증 영역은 `realm` 매개 변수 또는 `domain` 매개 변수를 정의하여 사용자 인터페이스 로그인 URL에서 지정될 수 있습니다. 인증 요청 영역은 다음 매개 변수/속성에 의해 여기에 표시된 순서대로 결정됩니다.

1. `domain` 매개 변수
2. `realm` 매개 변수
3. 관리자 서비스의 `DNS` **별칭 이름** 속성 값

   영역을 정확하게 호출한 다음에는 사용자가 인증할 인증 모듈을 핵심 인증 서비스의 영역 인증 구성 속성에서 검색합니다. 영역 기반 인증을 지정하고 초기화하는 데 사용하는 로그인 URL은 다음과 같습니다.

   ```
   http://server_name.domain_name:port/amserver/UI/Login
   http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
   http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name
   ```

   정의된 매개 변수가 없을 때는 로그인 URL에 지정된 서버 호스트와 도메인으로부터 영역이 결정됩니다.

주 – 사용자가 특정 영역의 구성원이고 이 영역에서 인증된 경우 다른 영역에 대해 인증을 받으려고 하면 두 매개 변수(`realm` 및 `module`)만 전달됩니다. 예를 들어 `User1`이 `realmA`의 구성원이고 이 영역에서 인증된 경우 `realmB`로 전환하거나 인증을 받으려고 하면 이 사용자는 `realmB`에 대해 지정된 모듈 인스턴스를 사용하여 `realmB` 인증을 새로 시작하거나 `realmA`의 기존 인증 세션으로 돌아가도록 요청하는 경고 페이지를 받게 됩니다. 사용자가 `realmB`에 대해 인증을 받으려는 경우 영역 이름과 모듈 이름(지정된 경우)만 전달되어 새 인증 프로세스를 결정하는 데 사용됩니다.

영역 기반 인증 리디렉션 URL

조직 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 영역 기반 인증 리디렉션 URL

성공한 영역 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `goto` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-success-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
9. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
10. 전역 기본값으로 `iplanet-am-auth-login-success-url` 속성에 설정된 URL

실패한 영역 기반 인증 리디렉션 URL

실패한 영역 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `gotoOnFail` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-failure-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL

9. 사용자 영역 항목의 iplanet-am-auth-login-failure-url 속성에 설정된 URL
10. 전역 기본값으로 iplanet-am-auth-login-failure-url 속성에 설정된 URL

영역 기반 인증을 구성하려면

먼저 핵심 인증 서비스를 영역에 추가하여 영역에 인증 모듈을 설정합니다.

▼ 영역의 인증 속성을 구성하려면

1 인증 체인을 추가할 영역으로 이동합니다.

2 [인증] 탭을 누릅니다.

3 [기본 인증 체인]을 선택합니다.

4 풀다운 메뉴에서 [관리자 인증 체인]을 선택합니다. 관리자의 인증 모듈이 최종 사용자의 모듈과 달라야 하는 경우 이 속성을 사용할 수 있습니다. 기본 인증 모듈은 LDAP입니다.

5 인증 체인을 정의한 후 [저장]을 누릅니다.

조직 기반 인증

이 인증 유형은 레거시 모드로 설치된 Access Manager 배포에만 적용됩니다.

이 인증 방법을 사용하면 조직 또는 하위 조직에 인증할 수 있습니다. 이는 Access Manager 인증의 기본 방법입니다. 조직 인증 방법은 핵심 인증 모듈을 조직에 등록하고 조직 인증 구성 속성을 정의함으로써 설정됩니다.

조직 기반 인증 로그인 URL

인증 조직은 사용자 인터페이스 로그인 URL에서 org 매개 변수나 domain 매개 변수를 정의하는 방법으로 지정할 수 있습니다. 인증 요청 조직은 여기에 표시된 순서대로 다음 매개 변수/속성에 의해 결정됩니다.

1. domain 매개 변수
2. org 매개 변수
3. 관리 서비스의 DNS Alias Names(조직 별칭 이름) 속성 값

   조직을 정확하게 호출한 다음에는 사용자가 인증할 인증 모듈을 핵심 인증 서비스의 조직 인증 구성 속성에서 검색합니다. 조직 기반 인증을 지정하고 시작하는 데 사용되는 로그인 URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login
http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
http://server_name.domain_name:port/amserver/UI/Login?org=org_name
```

정의된 매개 변수가 없을 때는 로그인 URL에 지정된 서버 호스트와 도메인으로부터 조직이 결정됩니다.

주 – 사용자가 특정 조직의 구성원이고 이 조직에서 인증된 경우 다른 조직의 인증을 받으려고 하면 두 매개 변수(`org` 및 `module`)만 전달됩니다. 예를 들어 `User1`이 `orgA`의 구성원이고 이 영역에서 인증된 경우 `orgB`로 전환하거나 인증을 받으려고 하면 이 사용자는 `orgB`에 대해 지정된 모듈 인스턴스에서 `orgB` 인증을 새로 시작하거나 `orgA`의 기존 인증 세션으로 돌아가도록 요청하는 경고 페이지를 받게 됩니다. 사용자가 `orgB`에 대해 인증을 받으려고 하면 영역 이름과 모듈 이름(지정된 경우)만 전달되어 새 인증 프로세스를 결정하는 데 적용됩니다.

조직 기반 인증 리디렉션 URL

조직 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 조직 기반 인증 리디렉션 URL

성공한 조직 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `goto` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 조직 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-success-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
9. 사용자 조직 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
10. 전역 기본값으로 `iplanet-am-auth-login-success-url` 속성에 설정된 URL

실패한 조직 기반 인증 리디렉션 URL

실패한 조직 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `gotoOnFail` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 조직 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-failure-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
9. 사용자 조직 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
10. 전역 기본값으로 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL

조직 기반 인증을 구성하려면

먼저 핵심 인증 서비스를 조직에 추가하여 조직에 인증 모듈을 설정합니다.

▼ 조직의 인증 속성을 구성하려면

1 인증 체인을 추가할 조직으로 이동합니다.

2 [인증] 탭을 누릅니다.

3 [기본 인증 체인]을 선택합니다.

4 풀다운 메뉴에서 [관리자 인증 체인]을 선택합니다. 관리자의 인증 모듈이 최종 사용자의 모듈과 달라야 하는 경우 이 속성을 사용할 수 있습니다. 기본 인증 모듈은 LDAP입니다.

5 인증 체인을 정의한 후 [저장]을 누릅니다.

역할 기반 인증

이 인증 방법을 사용하면 영역이나 하위 영역 내의 정적 또는 필터링된 역할에 인증할 수 있습니다.

주 – 인증 구성 서비스를 역할의 인스턴스로 등록하기 전에 먼저 영역에 등록해야 합니다.

인증이 성공하려면 사용자는 해당 역할에 속하고 이 역할에 구성된 인증 구성 서비스 인스턴스에 정의된 모듈마다 인증해야 합니다. 역할 기반 인증의 인스턴스마다 다음 속성을 지정할 수 있습니다.

충돌 해결 수준. 같은 사용자의 서로 다른 역할에 정의된 인증 구성 서비스 인스턴스에 대해 우선 순위 수준을 설정합니다. 예를 들어, `User1`이 `Role1` 및 `Role2`에 모두 지정되고 `Role1`에 더 높은 충돌 해결 수준이 설정되면 사용자가 인증을 시도할 때 성공 또는 실패 리디렉션과 인증 사후 프로세스에 대해 `Role1`에 더 높은 우선 순위가 적용됩니다.

인증 구성. 역할의 인증 프로세스에 구성된 인증 모듈을 정의합니다.

로그인 성공 URL. 성공한 인증에서 사용자가 리디렉션될 URL을 정의합니다.

로그인 실패 URL. 실패한 인증에서 사용자가 리디렉션될 URL을 정의합니다.

인증 사후 처리 클래스. 인증 사후 인터페이스를 정의합니다.

역할 기반 인증 로그인 URL

역할 기반 인증은 role 매개 변수를 정의하는 방법으로 사용자 인터페이스 로그인 URL에서 지정할 수 있습니다. 역할을 정확하게 호출한 다음에는 사용자가 인증할 인증 모듈을 해당 역할에 대해 정의된 인증 구성 서비스 인스턴스에서 검색합니다.

이 역할 기반 인증을 지정하고 초기화하는 데 사용되는 URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?role=role_name
http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name&role=role_name
```

realm 매개 변수가 구성되어 있지 않은 경우 역할이 속한 영역은 로그인 URL 자체에 지정된 서버 호스트와 도메인으로 결정됩니다.

역할 기반 인증 리디렉션 URL

역할 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 역할 기반 인증 리디렉션 URL

성공한 역할 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `goto` 로그인 URL 매개 변수에서 설정한 URL

3. 사용자 프로필(amUser.xml)의 iplanet-am-user-success-url에 대해 clientType 사용자 정의 파일에 설정된 URL
4. 사용자가 인증한 역할의 iplanet-am-auth-login-success-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
5. 인증된 사용자에 대한 다른 역할 항목의 iplanet-am-auth-login-success-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL(이전 리디렉션 URL이 실패한 경우 이 옵션으로 대체됩니다.)
6. 사용자 영역 항목의 iplanet-am-auth-login-success-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
7. iplanet-am-auth-login-success-url 속성에 대해 전역 기본값으로 clientType 사용자 정의 파일에 설정된 URL
8. 사용자 프로필(amUser.xml)의 iplanet-am-user-success-url 속성에 설정된 URL
9. 사용자가 인증한 역할의 iplanet-am-auth-login-success-url 속성에 설정된 URL
10. 인증된 사용자의 다른 역할 항목의 iplanet-am-auth-login-success-url 속성에 설정된 URL(이전 리디렉션 URL이 실패한 경우 이 옵션으로 대체됩니다.)
11. 사용자 영역 항목의 iplanet-am-auth-login-success-url 속성에 설정된 URL
12. 전역 기본값으로 iplanet-am-auth-login-success-url 속성에 설정된 URL

실패한 역할 기반 인증 리디렉션 URL

실패한 역할 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. goto 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(amUser.xml)의 iplanet-am-user-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
4. 사용자가 인증한 역할의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
5. 인증된 사용자에 대한 다른 역할 항목의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL(이전 리디렉션 URL이 실패한 경우 이 옵션으로 대체됩니다.)
6. 사용자 영역 항목의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
7. iplanet-am-auth-login-failure-url 속성에 대해 전역 기본값으로 clientType 사용자 정의 파일에 설정된 URL
8. 사용자 프로필(amUser.xml)의 iplanet-am-user-failure-url 속성에 설정된 URL
9. 사용자가 인증한 역할의 iplanet-am-auth-login-failure-url 속성에 설정된 URL

10. 인증된 사용자에 대한 다른 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL(이전 리디렉션 URL이 실패한 경우 이 옵션으로 대체됩니다.)
11. 사용자 영역 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
12. 전역 기본값으로 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL

▼ 역할 기반 인증을 구성하려면

1 **인증 구성 서비스를 추가할 영역(또는 조직)으로 이동합니다.**

2 **[주제] 탭을 누릅니다.**

3 **[필터링된 역할] 또는 [역할]을 누릅니다.**

4 **인증 구성을 설정할 역할을 선택합니다.**

5 **활성화할 [기본 인증 체인]을 선택합니다.**

6 **[저장]을 누릅니다.**

주 – 새 역할을 만들 경우 인증 구성 서비스가 해당 역할에 자동으로 할당되지 않습니다. 새 역할을 만들기 전에 역할 프로필 페이지의 위쪽에 있는 인증 구성 서비스 옵션을 선택하십시오.

역할 기반 인증이 사용 가능한 경우 구성원을 구성할 필요가 없으므로 LDAP 인증 모듈을 기본값으로 그대로 사용할 수 있습니다.

서비스 기반 인증

이 인증 방법을 사용하면 영역 또는 하위 영역에 등록된 특정 서비스나 응용 프로그램에 인증할 수 있습니다. 이러한 서비스는 인증 구성 서비스 내에 서비스 인스턴스로 구성되고 인스턴스 이름과 연관됩니다. 인증이 성공하려면 사용자는 해당 서비스에 구성된 인증 구성 서비스 인스턴스에 정의된 모듈마다 인증해야 합니다. 서비스 기반 인증의 인스턴스마다 다음 속성을 지정할 수 있습니다.

인증 구성. 서비스의 인증 프로세스에 구성된 인증 모듈을 정의합니다.

로그인 성공 URL. 성공한 인증에서 사용자가 리디렉션될 URL을 정의합니다.

로그인 실패 URL. 실패한 인증에서 사용자가 리디렉션될 URL을 정의합니다.

인증 사후 처리 클래스. 인증 사후 인터페이스를 정의합니다.

서비스 기반 인증 로그인 URL

서비스 기반 인증은 service 매개 변수를 정의하는 방법으로 사용자 인터페이스 로그인 URL에서 지정할 수 있습니다. 서비스를 호출한 다음에는 해당 서비스에 대해 정의된 인증 구성 서비스로부터 사용자가 인증할 인증 모듈을 검색합니다.

서비스 기반 인증을 지정하고 초기화하는 데 사용되는 URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/
Login?service=auth-chain-name
```

및

```
http://server_name.domain_name:port/amserver
     /UI/Login?realm=realm_name&service=auth-chain-name
```

org 매개 변수를 지정하지 않은 경우에는 로그인 URL 자체에 지정된 서버 호스트와 도메인으로부터 영역이 결정됩니다.

서비스 기반 인증 리디렉션 URL

서비스 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 서비스 기반 인증 리디렉션 URL

성공한 서비스 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. goto 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(amUser.xml)의 iplanet-am-user-success-url에 대해 clientType 사용자 정의 파일에 설정된 URL
4. 사용자가 인증한 서비스의 iplanet-am-auth-login-success-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
5. 사용자 역할 항목의 iplanet-am-auth-login-success-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
6. 사용자 영역 항목의 iplanet-am-auth-login-success-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
7. iplanet-am-auth-login-success-url 속성에 대해 전역 기본값으로 clientType 사용자 정의 파일에 설정된 URL
8. 사용자 프로필(amUser.xml)의 iplanet-am-user-success-url 속성에 설정된 URL
9. 사용자가 인증한 서비스의 iplanet-am-auth-login-success-url 속성에 설정된 URL

10. 사용자 역할 항목의 iplanet-am-auth-login-success-url 속성에 설정된 URL
11. 사용자 영역 항목의 iplanet-am-auth-login-success-url 속성에 설정된 URL
12. 전역 기본값으로 iplanet-am-auth-login-success-url 속성에 설정된 URL

실패한 서비스 기반 인증 리디렉션 URL

실패한 서비스 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. goto 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(amUser.xml)의 iplanet-am-user-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
4. 사용자가 인증한 서비스의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
5. 사용자 역할 항목의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
6. 사용자 영역 항목의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
7. iplanet-am-auth-login-failure-url 속성에 대해 전역 기본값으로 clientType 사용자 정의 파일에 설정된 URL
8. 사용자 프로필(amUser.xml)의 iplanet-am-user-failure-url 속성에 설정된 URL
9. 사용자가 인증한 서비스의 iplanet-am-auth-login-failure-url 속성에 설정된 URL
10. 사용자 역할 항목의 iplanet-am-auth-login-failure-url 속성에 설정된 URL
11. 사용자 영역 항목의 iplanet-am-auth-login-failure-url 속성에 설정된 URL
12. 전역 기본값으로 iplanet-am-auth-login-failure-url 속성에 설정된 URL

▼ 서비스 기반 인증을 구성하려면

인증 구성 서비스를 추가한 다음 서비스에 대한 인증 모듈을 설정합니다. 수행 방법은 다음과 같습니다.

1 서비스 기반 인증을 구성할 영역을 선택합니다.

2 [인증] 탭을 누릅니다.

3 인증 모듈 인스턴스를 만듭니다.

4 인증 체인을 만듭니다.

5 **[저장]을 누릅니다.**

6 **영역에 대한 서비스 기반 인증에 액세스하려면 다음 주소를 입력합니다.**

`http://`*server_name.domain_name:port*`/amserver/UI/Login?realm=`*realm_name*`&service=`*auth-chain-name*

사용자 기반 인증

이 인증 방법을 사용하면 사용자에 대해 특별히 구성된 인증 프로세스를 인증할 수 있습니다. 프로세스는 사용자 프로필의 사용자 인증 구성 속성 값으로 구성됩니다. 인증이 성공하려면 정의된 모듈마다 인증해야 합니다.

사용자 기반 인증 로그인 URL

사용자 기반 인증은 사용자 인터페이스 로그인 URL에서 user 매개 변수를 정의하는 방법으로 지정할 수 있습니다. 사용자를 정확하게 호출한 다음에는 사용자가 인증할 인증 모듈을 정의된 사용자 인증 구성 인스턴스에서 검색합니다.

이 역할 기반 인증을 지정하고 초기화하는 데 사용되는 URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?user=user_name
http://server_name.domain_name:port/amserver/UI/Login?org=org_name&user=user_name
```

realm 매개 변수를 지정하지 않은 경우 역할이 속한 영역은 로그인 URL 자체에 지정된 서버 호스트와 도메인에서 결정됩니다.

사용자 별칭 목록 속성

사용자 기반 인증에 대한 요청을 받으면 인증 서비스에서는 먼저 사용자가 유효한 사용자인지 확인하고 그에 대한 인증 구성 데이터를 검색합니다. 사용자 로그인 URL 매개 변수 값과 관련하여 유효한 사용자 프로필이 둘 이상 있는 경우에는 모든 프로필이 지정된 사용자에 매핑되어야 합니다. 사용자 프로필의 사용자 별칭 속성(`iplanet-am-user-alias-list`)은 해당 사용자에 속한 다른 프로필을 정의할 수 있는 위치입니다. 매핑이 실패하면 사용자는 유효한 세션에서 거부됩니다. 사용자 중 하나가 최상위 관리자이므로 사용자 매핑 검증이 수행되지 않고 사용자가 최상위 관리자 권한을 가진 경우는 예외가 될 수 있습니다.

사용자 기반 인증 리디렉션 URL

사용자 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 사용자 기반 인증 리디렉션 URL

성공한 사용자 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `goto` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-success-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
9. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
10. 전역 기본값으로 `iplanet-am-auth-login-success-url` 속성에 설정된 URL

실패한 사용자 기반 인증 리디렉션 URL

실패한 사용자 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `gotoOnFail` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-failure-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
9. 사용자 영역 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
10. 전역 기본값으로 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL

▼ 사용자 기반 인증을 구성하려면

1 **사용자에 대해 인증을 구성할 영역으로 이동합니다.**

2 **[주제] 탭을 누르고 [사용자]를 누릅니다.**

3 **수정할 사용자의 이름을 누릅니다.**
[사용자 프로필]이 표시됩니다.

주 – 새 사용자를 만들 경우 [인증 구성] 서비스가 사용자에 자동으로 할당되지 않습니다. 사용자를 만들기 전에 서비스 프로필에 있는 [인증 구성] 서비스 옵션을 선택하십시오. 이 옵션을 선택하지 않으면 사용자가 해당 역할에 대해 정의된 인증 구성을 상속하지 못합니다.

---

4 **[사용자 인증 구성] 속성에서 적용할 인증 체인을 선택합니다.**

5 **[저장]을 누릅니다.**

인증 수준 기반 인증

각 인증 모듈에 해당 **인증 수준**에 대한 정수 값을 연결할 수 있습니다. 모듈의 [인증 수준] 속성에 해당하는 값을 변경하여 인증 수준을 할당할 수 있습니다. 높은 인증 수준은 사용자가 하나 또는 여러 인증 모듈에 인증을 얻은 후에 사용자에 대해 높은 신뢰도를 정의합니다.

사용자가 모듈에 성공적으로 인증하면 인증 수준이 사용자의 SSO 토큰에 설정됩니다. 사용자가 여러 인증 모듈에 성공적으로 인증되어야 하는 경우 가장 높은 인증 수준 값이 사용자의 SSO 토큰에 설정됩니다.

사용자가 서비스에 대한 액세스를 시도한 경우 서비스는 사용자의 SSO 토큰에서 인증 수준을 확인하여 사용자에게 액세스를 허용할지 여부를 결정할 수 있습니다. 그런 다음 설정된 인증 수준을 사용하여 인증 모듈을 통해 이동하도록 사용자를 리디렉션합니다.

사용자는 특정 인증 수준을 사용하여 인증 모듈에 액세스 할 수도 있습니다. 예를 들어, 다음 구문을 사용하여 로그인을 수행합니다.

```
http://hostname:port/deploy_URI/UI/Login?authlevel=
auth_level_value
```

인증 수준이 *auth_level_value*보다 크거나 같은 모든 모듈은 사용자가 선택할 수 있는 인증 메뉴로 표시됩니다. 일치하는 모듈이 하나이면 이 인증 모듈에 대한 인증 페이지가 직접 표시됩니다.

이 인증 방법을 사용하면 관리자가 아이디로 인증할 수 있는 모듈의 보안 수준을 지정할 수 있습니다. 인증 모듈마다 별도의 인증 수준 속성이 있고 이 속성의 값은 유효한 정수로 정의될 수 있습니다. 인증 수준 기반 인증을 사용하면 인증 서비스에서 인증 모듈을 포함하는 메뉴가 있는 모듈 로그인 페이지를 표시하는데, 이 인증 모듈의 인증 수준은 로그인 URL 매개 변수에서 지정한 값보다 크거나 같습니다. 사용자는 제시된 목록에서 모듈을 선택할 수 있습니다. 모듈을 선택하면 나머지 프로세스는 모듈 기반 인증에 따라 진행됩니다.

인증 수준 기반 인증 로그인 URL

인증 수준 기반 인증은 authlevel 매개 변수를 정의하는 방법으로 사용자 인터페이스 로그인 URL에서 지정할 수 있습니다. 관련된 모듈 목록이 있는 로그인 화면을 호출한 후 사용자는 인증할 모듈을 하나 선택해야 합니다. 인증 수준 기반 인증을 지정하고 초기화하는 데 사용되는 URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=authentication_level
```

및

```
http://server_name.domain_name:port/amserver/UI/
Login?realm=realm_name&authlevel=authentication_level
```

`realm` 매개 변수를 지정하지 않은 경우 사용자가 속한 영역은 로그인 URL 자체에 지정된 서버 호스트와 도메인으로부터 결정됩니다.

인증 수준 기반 인증 리디렉션 URL

인증 수준 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 인증 수준 기반 인증 리디렉션 URL

성공한 인증 수준 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `goto` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL

6. iplanet-am-auth-login-success-url 속성에 대해 전역 기본값으로 clientType 사용자 정의 파일에 설정된 URL
7. 사용자 프로필(amUser.xml)의 iplanet-am-user-success-url 속성에 설정된 URL
8. 사용자 역할 항목의 iplanet-am-auth-login-success-url 속성에 설정된 URL
9. 사용자 영역 항목의 iplanet-am-auth-login-success-url 속성에 설정된 URL
10. 전역 기본값으로 iplanet-am-auth-login-success-url 속성에 설정된 URL

실패한 인증 수준 인증 리디렉션 URL

실패한 인증 수준 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. gotoOnFail 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 항목(amUser.xml)의 iplanet-am-user-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 iplanet-am-auth-login-failure-url 속성에 대해 clientType 사용자 정의 파일에 설정된 URL
6. iplanet-am-auth-login-failure-url 속성에 대해 전역 기본값으로 clientType 사용자 정의 파일에 설정된 URL
7. 사용자 항목(amUser.xml)의 iplanet-am-user-failure-url 속성에 설정된 URL
8. 사용자 역할 항목의 iplanet-am-auth-login-failure-url 속성에 설정된 URL
9. 사용자 영역 항목의 iplanet-am-auth-login-failure-url 속성에 설정된 URL
10. 전역 기본값으로 iplanet-am-auth-login-failure-url 속성에 설정된 URL

모듈 기반 인증

다음 구문을 사용하여 특정 인증 모듈에 액세스할 수 있습니다.

```
http://hostname:port/deploy_URI/UI/Login?module=
module_name
```

인증 모듈에 액세스하기 전에 인증 모듈 이름을 포함하도록 핵심 인증 서비스 속성인 영역 인증 모듈을 수정해야 합니다. 인증 모듈 이름이 이 속성에 없으면 사용자가 인증하려고 시도할 때 "인증 모듈이 거부되었습니다"라는 페이지가 표시됩니다.

이 인증 방법을 사용하면 사용자가 인증할 모듈을 지정할 수 있습니다. 지정된 모듈은 사용자가 액세스하는 영역 또는 하위 영역에 등록되어야 합니다. 이 모듈은 영역의 핵심 인증 서비스의 영역 인증 모듈 속성에서 구성됩니다. 이러한 모듈 기반 인증 요청을 받으면 인증 서비스에서 모듈이 정확하게 구성되었는지 확인하고 모듈이 정의되지 않은 경우에는 사용자 액세스가 거부됩니다.

모듈 기반 인증 로그인 URL

모듈 기반 인증은 사용자 인터페이스 로그인 URL에서 module 매개 변수를 정의하는 방법으로 지정할 수 있습니다. 모듈 기반 인증을 지정하고 초기화하는 데 사용되는 URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?module=authentication_module_name
http://server_name.domain_name:port/amserver/UI/
Login?org=org_name&module=authentication_module_name
```

`org` 매개 변수를 지정하지 않은 경우 사용자가 속한 영역은 로그인 URL 자체에 지정된 서버 호스트와 도메인으로부터 결정됩니다.

모듈 기반 인증 리디렉션 URL

모듈 기반 인증이 성공/실패하면 Access Manager는 사용자를 리디렉션할 위치 정보를 찾습니다. 응용 프로그램에서 이 정보를 찾는 순서는 다음과 같습니다.

성공한 모듈 기반 인증 리디렉션 URL

성공한 모듈 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `goto` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url`에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-success-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 프로필(`amUser.xml`)의 `iplanet-am-user-success-url` 속성에 설정된 URL
8. 사용자 역할 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
9. 사용자 영역 항목의 `iplanet-am-auth-login-success-url` 속성에 설정된 URL
10. 전역 기본값으로 `iplanet-am-auth-login-success-url` 속성에 설정된 URL

실패한 모듈 기반 인증 리디렉션 URL

실패한 모듈 기반 인증의 리디렉션 URL은 다음 정보를 순서대로 확인하여 결정합니다.

1. 인증 모듈에서 설정한 URL
2. `gotoOnFail` 로그인 URL 매개 변수에서 설정한 URL
3. 사용자 항목(`amUser.xml`)의 `iplanet-am-user-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
4. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
5. 사용자 영역 항목의 `iplanet-am-auth-login-failure-url` 속성에 대해 `clientType` 사용자 정의 파일에 설정된 URL
6. `iplanet-am-auth-login-failure-url` 속성에 대해 전역 기본값으로 `clientType` 사용자 정의 파일에 설정된 URL
7. 사용자 역할 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
8. 사용자 영역 항목의 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL
9. 전역 기본값으로 `iplanet-am-auth-login-failure-url` 속성에 설정된 URL

사용자 인터페이스 로그인 URL

인증 서비스 사용자 인터페이스는 웹 브라우저의 위치 표시줄에 로그인 URL을 입력하는 방법으로 액세스할 수 있습니다. 다음과 같이 URL을 입력합니다.

`http://`*AccessManager-root*`/.`*domain_name*`:`*port*` /`*service_deploy_uri*` /UI/Login`

주 – 설치 도중 *service_deploy_uri*가 `amserver`로 구성됩니다. 이러한 기본 서비스 배포 URI은 이 설명서 전반에 걸쳐 사용됩니다.

사용자 인터페이스 로그인 URL에 로그인 URL 매개 변수가 추가되어 특정 인증 방법이나 성공 또는 실패한 인증 리디렉션 URL을 정의합니다.

로그인 URL 매개 변수

URL 매개 변수는 URL의 끝에 추가되는 이름/값 쌍입니다. 매개 변수는 물음표(?)로 시작하며 `name=value` 형식으로 사용됩니다. 다음과 같이 여러 개의 매개 변수를 하나의 로그인 URL로 조합할 수 있습니다.

```
http://server_name.domain_name:port/amserver/UI/
Login?module=LDAP&locale=ja&goto=http://www.sun.com
```

매개 변수가 둘 이상인 경우 앰퍼샌드(&)로 구분됩니다. 그러나 매개 변수 조합은 다음 지침을 지켜야 합니다.

- 각 매개 변수는 하나의 URL에서 한 번만 사용되어야 합니다. 예를 들어, `module=LDAP&module=NT`는 사용할 수 없습니다.
- `org` 매개 변수와 `domain` 매개 변수는 모두 로그인 영역을 결정합니다. 이 경우 로그인 URL에서 두 매개 변수 중 하나만 사용해야 합니다. 두 매개 변수를 모두 사용하면서 우선 순위를 지정하지 않으면 하나만 적용됩니다.
- `user`, `role`, `service`, `module` 및 `authlevel` 매개 변수는 각각의 기준에 따라 인증 모듈을 정의하는 매개 변수입니다. 따라서 로그인 URL에는 이들 중 하나만 사용해야 합니다. 매개 변수를 두 개 이상 사용하면서 우선 순위를 지정하지 않으면 하나만 적용됩니다.

다음 절에서는 사용자 인터페이스 로그인 URL에 추가하여 웹 브라우저의 위치 표시줄에 입력했을 때 여러 가지 인증 기능을 수행하는 매개 변수를 설명합니다.

주 – 영역에 전사적으로 배포할 인증 URL 및 매개 변수를 간소화하기 위해 관리자는 간단한 URL을 사용하여 HTML 페이지를 구성할 수 있는데 이 페이지에는 구성된 모든 인증 방법에서 사용할 복잡한 로그인 URL 링크가 포함됩니다.

---

goto 매개 변수

`goto=`*successful_authentication_URL* 매개 변수는 인증 구성 서비스의 로그인 성공 URL에 정의된 값 대신 사용됩니다. 이 매개 변수는 인증이 성공하면 지정된 URL로 연결됩니다. 마찬가지로 `goto=logout_URL` 매개 변수는 사용자 로그아웃 시 지정된 URL로 연결하기 위해 사용합니다. 성공적인 인증 URL의 예는 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/
UI/Login?goto=http://www.sun.com/homepage.html
```

`goto` 로그아웃 URL의 예는 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/
UI/Logout?goto=http://www.sun.com/logout.html.
```

주 – Access Manager에서 성공적인 인증 리디렉션 URL을 찾을 때 적용하는 우선 순위가 있습니다. 이러한 리디렉션 URL 및 해당 순서는 인증 방법에 따라 다르므로 인증 유형 절에서 이 순서(및 관련 정보)를 자세히 설명합니다.

---

gotoOnFail 매개 변수

gotoOnFail=failed_authentication_URL 매개 변수는 인증 구성 서비스의 로그인 실패 URL에 정의된 값 대신 사용됩니다. 이 매개 변수는 사용자 인증 실패 시 지정된 URL로 연결됩니다. 예를 들어 gotoOnFail URL은 http:// *server_name.domain_name*:*port* /amserver/UI/Login?gotoOnFail=http://www.sun.com/auth_fail.html이 될 수 있습니다.

주 – Access Manager에서 실패한 인증 리디렉션 URL을 찾을 때 적용하는 우선 순위가 있습니다. 이러한 리디렉션 URL 및 해당 순서는 인증 방법에 따라 다르므로 인증 유형 절에서 이 순서(및 관련 정보)를 자세히 설명합니다.

---

realm 매개 변수

org=*realmName* 매개 변수를 사용하면 사용자가 지정된 영역에서 사용자로 인증할 수 있습니다.

주 – 아직 지정된 영역의 구성원이 아닌 사용자가 realm 매개 변수를 사용하여 인증하려고 하면 오류 메시지가 나타납니다. 그러나 다음 사항이 모두 해당되면 Directory Server에서 사용자 프로필을 동적으로 작성할 수 있습니다.

- 핵심 인증 서비스의 사용자 프로필 속성은 **동적** 또는 **동적 (사용자 별칭과 함께 사용)** 으로 설정되어야 합니다.
- 사용자는 필수 모듈에 성공적으로 인증되어야 합니다.
- 사용자는 아직 Directory Server에 프로필이 없습니다.

이 매개 변수를 통해 영역 및 로켈 설정에 따라 정확한 로그인 페이지가 표시됩니다. 이 매개 변수를 설정하지 않은 경우 기본값은 최상위 영역입니다. 예를 들어, 다음은 org URL이 될 수 있습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?realm=sun
```

---

org 매개 변수

org=*orgName* 매개 변수를 사용하면 사용자가 지정된 조직의 사용자로서 인증할 수 있습니다.

주 – 아직 지정된 조직의 구성원이 아닌 사용자가 `org` 매개 변수를 사용하여 인증하려고 하면 오류 메시지가 나타납니다. 그러나 다음 사항이 모두 해당되면 Directory Server에서 사용자 프로필을 동적으로 만들 수 있습니다.

- 핵심 인증 서비스의 사용자 프로필 속성은 **동적** 또는 **동적 (사용자 별칭과 함께 사용)** 으로 설정되어야 합니다.
- 사용자는 필수 모듈에 성공적으로 인증되어야 합니다.
- 사용자는 아직 Directory Server에 프로필이 없습니다.

이 매개 변수를 통해 조직 및 로켈 설정에 따라 정확한 로그인 페이지가 표시됩니다. 이 매개 변수를 설정하지 않은 경우 기본값은 최상위 조직입니다. 예를 들면 `org` URL이 다음과 같을 수 있습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?org=sun
```

user 매개 변수

`user=`*userName* 매개 변수는 시용자 프로필의 사용자 인증 구성 속성에서 구성된 모듈을 기반으로 인증을 실행합니다. 예를 들어, 한 사용자의 프로필은 인증 모듈을 사용하여 인증하도록 구성하는 반면 다른 사용자는 LDAP 모듈을 사용하여 인증하도록 구성할 수 있습니다. 이 매개 변수를 추가하면 사용자의 조직에 구성된 방법이 아닌 사용자 자신이 구성한 인증 프로세스를 따르게 됩니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?user=jsmith
```

role 매개 변수

`role=`roleName 매개 변수는 지정된 역할을 위해 구성된 인증 프로세스를 따르게 합니다. 아직 지정된 역할의 구성원이 아닌 사용자가 이 매개 변수를 사용하여 인증하려고 하면 오류 메시지가 나타납니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?role=manager.
```

locale 매개 변수

Access Manager에는 콘솔 자체는 물론 인증 프로세스에서도 현지화된 화면(영어가 아닌 다른 언어로 번역된 화면)을 표시하는 기능이 있습니다. `locale=`*localeName* 매개 변수를 사용하면 지정된 로켈이 정의된 다른 로켈보다 우선 적용됩니다. 로그인 로켈은 다음 위치에서 순서에 따라 구성을 검색한 후 클라이언트에 표시됩니다.

1. 로그인 URL에서 locale 매개 변수의 값

   `locale=`*localeName* 매개 변수의 값은 정의된 다른 로켈보다 우선 적용됩니다.
2. 사용자 프로필에 정의된 로켈

URL 매개 변수가 없을 때는 사용자 프로필의 사용자 기본 언어 속성에 설정된 값에 따라 로켈이 표시됩니다.

3. HTTP 헤더에 정의된 로켈

   이 로켈은 웹 브라우저에서 설정합니다.

4. 핵심 인증 서비스에 정의된 로켈

   이 로켈은 핵심 인증 모듈의 기본 인증 로켈 속성의 값입니다.

5. 플랫폼 서비스에 정의된 로켈

   이 로켈은 플랫폼 서비스에서 플랫폼 로켈 속성의 값입니다.

운영 체제 로켈

여기서 파생된 로켈은 사용자의 세션 토큰에 저장되며 Access Manager에서는 현지화된 인증 모듈을 로드할 때만 이를 사용합니다. 인증이 성공적으로 수행되면 사용자 프로필의 기본 언어 속성에 정의된 로켈이 사용됩니다. 아무 것도 설정되어 있지 않을 때는 인증에 사용된 로켈이 적용됩니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?locale=ja.
```

주 – Access Manager에서 화면 텍스트와 오류 메시지 현지화 방법에 대한 내용을 참조할 수 있습니다.

---

module 매개 변수

`module=`*moduleName* 매개 변수를 사용하면 지정된 인증 모듈을 통해 인증할 수 있습니다. 모든 인증 모듈은 먼저 사용자가 속한 영역에서 등록되고 핵심 인증 모듈에서 해당 영역의 인증 모듈 중 하나로 선택되어야 지정될 수 있습니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?module=Unix.
```

주 – URL 매개 변수에서 사용되는 인증 모듈 이름은 대소문자를 구분합니다.

---

service 매개 변수

`service=`*serviceName* 매개 변수를 사용하면 사용자는 서비스의 구성된 인증 스키마를 통해 인증할 수 있습니다. 인증 구성 서비스를 사용하여 서비스마다 인증 스키마를 달리 구성할 수 있습니다. 예를 들어, 영역의 직원 디렉토리 응용 프로그램은 LDAP 인증 모듈만 필요한 반면 온라인 급여 응용 프로그램은 보다 안전한 인증서 인증 모듈을 사용하여 인증해야 합니다. 이러한 서비스마다 인증 스키마를 구성하고 이름을 지정할 수 있습니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?service=sv1.
```

주 – 인증 구성 서비스는 서비스 기반의 인증을 위한 스키마 정의에 사용됩니다.

arg 매개 변수

arg=*newsession* 매개 변수는 사용자의 현재 세션을 종료하고 새 세션을 시작하는 데 사용됩니다. 인증 서비스는 사용자의 기존 세션 토큰을 제거하고 요청 시마다 새 로그인을 수행합니다. 이 옵션은 일반적으로 익명 인증 모듈에서 사용됩니다. 사용자는 먼저 익명 세션으로 인증한 다음 등록이나 로그인 링크를 누릅니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?arg=newsession.
```

authlevel 매개 변수

authlevel=*value* 매개 변수는 인증 수준이 지정된 인증 수준 값보다 크거나 같은 모듈을 호출하도록 인증 서비스에 명령합니다. 각 인증 모듈은 고정된 정수 인증 수준으로 정의됩니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=1.
```

주 – 인증 수준은 각 모듈의 특정 프로필에 설정됩니다.

domain 매개 변수

이 매개 변수를 사용하면 지정된 도메인으로 식별된 영역에 로그인할 수 있습니다. 지정된 도메인은 영역 프로필의 도메인 이름 속성에 정의된 값과 일치해야 합니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?domain=sun.com.
```

주 – 아직 지정된 도메인/영역의 구성원이 아닌 사용자가 org 매개 변수를 사용하여 인증하려고 하면 오류 메시지가 나타납니다. 그러나 다음 사항이 모두 해당되면 Directory Server에서 사용자 프로필을 동적으로 작성할 수 있습니다.

- 핵심 인증 서비스의 사용자 프로필 속성은 **동적** 또는 **동적 (사용자 별칭과 함께 사용)** 으로 설정되어야 합니다.
- 사용자는 필수 모듈에 성공적으로 인증되어야 합니다.
- 사용자는 아직 Directory Server에 프로필이 없습니다.

iPSPCookie 매개 변수

iPSPCookie=yes 매개 변수를 사용하면 영구 쿠키를 사용하여 로그인할 수 있습니다. 영구 쿠키는 브라우저 창을 닫은 후에도 계속 존재하는 쿠키를 말합니다. 이 매개 변수를 사용하기 위해서는 사용자가 로그인한 조직의 영구 쿠키가 핵심 인증 모듈에서 활성화되어 있어야 합니다. 사용자가 인증되고 브라우저를 닫은 후에는 다시 인증할 필요 없이 새 브라우저 세션으로 로그인할 수 있으며 콘솔로 직접 이동합니다. 이러한 작업은 핵심 서비스에 지정된 영구 쿠키 최대 시간의 값이 경과할 때까지 가능합니다. 예를 들면 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?org=example&iPSPCookie=yes
```

IDTokenN 매개 변수

이 매개 변수 옵션을 사용하면 URL 또는 HTML 형식으로 인증 자격 증명을 통과할 수 있습니다. IDTokenN=*value* 매개 변수를 사용하는 경우 인증 서비스 사용자 인터페이스에 액세스하지 않고도 인증할 수 있습니다. 이러한 프로세스를 *0* **페이지 로그인**이라고 합니다. 0페이지 로그인은 하나의 로그인 페이지를 사용하는 인증 모듈에서만 작동합니다. IDToken0, IDToken1, ..., IDTokenN 값은 인증 모듈의 로그인 페이지에 있는 필드에 매핑됩니다. 예를 들어 LDAP 인증 모듈은 userID 정보에 IDToken1을 사용하고, 비밀번호 정보에 IDToken2를 사용할 수 있습니다. 이 경우 LDAP 모듈 IDTokenN URL은 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/
Login?module=LDAP&IDToken1=userID&IDToken2=password
```

(LDAP가 기본 인증 모듈인 경우 module=LDAP를 생략할 수 있습니다.)

익명 인증의 경우 로그인 URL 매개 변수는 다음과 같습니다.

```
http://server_name.domain_name:port/amserver/UI/Login?module=Anonymous&IDToken1=anonymousUserID.
```

주 – 이전 릴리스의 토큰 이름인 Login.Token0, Login.Token1, ..., Login.TokenN은 아직 지원되지만 향후 릴리스에서는 사용할 수 없게 됩니다. 새로 제공되는 IDTokenN 매개 변수를 사용하는 것이 좋습니다.

계정 잠금

인증 서비스는 사용자 인증이 n회 실패하면 사용자 인증을 **잠그는** 기능을 제공합니다. 이 기능은 기본적으로 비활성화되어 있지만 Access Manager 콘솔을 사용하여 활성화할 수 있습니다.

주 – 잘못된 비밀번호 예외가 발생하는 모듈만 계정 잠금 기능을 사용할 수 있습니다.

핵심 인증 서비스에는 다음을 포함하여 이 기능을 활성화/사용자 정의하는 속성이 들어 있습니다.

- **로그인 실패 잠금 모드.** 계정 잠금을 활성화합니다.
- **로그인 실패 잠금 수.** 사용자가 잠기기 전에 인증을 시도할 수 있는 횟수를 정의합니다. 이 값은 사용자 아이디에 대해서만 적용됩니다. 동일한 사용자 아이디가 지정된 횟수만큼 실패하면 그 사용자 아이디는 잠겨집니다.
- **로그인 실패 잠금 간격.** 사용자 잠금이 적용되기 전 얼마 동안 로그인 실패 잠금 수의 값이 완료되어야 하는지 분 단위로 정의합니다.
- **잠금 알림을 보낼 전자 메일 주소.** 사용자 잠금 알림을 보낼 전자 메일 주소를 지정합니다.
- **N회 실패 후 사용자에게 경고.** 몇 차례 인증이 실패하면 경고 메시지가 사용자에게 표시되는지 지정합니다. 관리자는 사용자에게 잠금이 임박했음을 경고한 이후의 추가 로그인 시도를 설정할 수 있습니다.
- **로그인 실패 잠금 기간.** 잠금 후 얼마나 대기한 후 다시 인증을 시도할 수 있는지 분 단위로 정의합니다.
- **잠금 속성 이름.** 물리적 잠금에 대해 사용자 프로필 중 어떤 LDAP 속성이 `inactive`로 설정될 것인지 정의합니다.
- **잠금 속성 값. 잠금 속성 이름**에 지정된 LDAP 속성 중 어떤 속성이 `inactive` 또는 `active`로 설정될 것인지 정의합니다.

계정 잠금이 발생하면 관리자에게 전자 메일 알림이 전송됩니다. (계정 잠금 활동도 기록)

주 – Microsoft® Windows 2000 운영 체제에서의 이 기능 사용에 대한 자세한 내용은 부록 A, “AMConfig.properties 파일”에서 “SMTP(Simple Mail Transfer Protocol)”를 참조하십시오.

Access Manager에서는 다음 절에서 정의하는물리적 잠금과 메모리 잠금의 두 가지 계정 잠금 유형을 지원합니다.

물리적 잠금

이 동작은 Access Manager에 대한 기본 잠금 동작입니다. 사용자 프로필의 LDAP 속성 상태를 inactive로 변경하면 이 잠김이 초기화됩니다. **잠금 속성 이름**은 잠금 목적에 따라 사용되는 LDAP 속성을 정의합니다.

주 – 별칭 사용자는 LDAP 프로필에서 사용자 별칭 목록 속성(amUser.xml의 iplanet-am-user-alias-list)을 구성하는 방법으로 기존의 LDAP 사용자 프로필에 매핑된 사용자입니다. 별칭 사용자는 핵심 인증 서비스의 별칭 검색 속성 이름 필드에 iplanet-am-user-alias-list를 추가함으로써 검증할 수 있습니다. 즉 별칭 사용자가 잠긴 경우 해당 사용자가 별칭 처리된 실제 LDAP 프로필도 잠기게 됩니다. 이는 LDAP 및 구성원이 아닌 인증 모듈을 사용하는 물리적 잠금에만 적용됩니다.

메모리 잠금

메모리 잠금은 **로그인 실패 잠금 기간** 속성을 0보다 큰 값으로 변경하는 방법으로 사용할 수 있습니다. 그러면 사용자 계정은 지정된 시간(분) 동안 메모리에서 잠깁니다. 시간이 모두 경과한 후에는 계정의 잠금이 해제됩니다. 메모리 잠금 기능을 사용할 때는 몇 가지 사항에 특별히 주의해야 합니다.

- Access Manager가 다시 시작되면 메모리에서 잠긴 모든 계정이 잠금 해제됩니다.
- 사용자 계정이 메모리에서 잠겨있고 관리자가 계정 잠금 체계를 물리적 잠금으로 변경한 경우(잠금 기간을 다시 0으로 설정) 사용자 계정이 메모리에서 잠금 해제되고 잠금 수가 재설정됩니다.
- 메모리 잠금 후 LDAP 및 구성원을 제외한 인증 모듈을 사용할 때 사용자가 정확한 비밀번호로 로그인을 시도할 경우 **사용자가 활성 상태가 아닙니다**. 오류 대신 **이 조직에 사용자의 프로필이 없습니다**. 오류가반환됩니다.

주 – 사용자 프로필에 실패 URL 속성이 설정된 경우 잠금 경고 메시지나 계정이 잠겨 있음을 나타내는 메시지가 표시되지 않고 정의된 URL로 사용자가 리디렉션됩니다.

인증 서비스 페일오버

인증 서비스 페일오버는 하드웨어나 소프트웨어 문제 때문에 주 서버에 장애가 발생하거나 서버가 일시적으로 다운될 경우 자동으로 인증 요청을 보조 서버로 리디렉션합니다.

인증 서비스를 사용할 수 있는 Access Manager 인스턴스에서 인증 컨텍스트가 먼저 만들어져야 합니다. 이 Access Manager 인스턴스를 사용할 수 없는 경우 인증 페일오버를 통해 다른 Access Manager 인스턴스에서 인증 컨텍스트를 만들 수 있습니다. 인증 컨텍스트는 다음 순서로 서버 가용성을 확인합니다.

1. 인증 서비스 URL이 AuthContext API로 전달됩니다. 예를 들면 다음과 같습니다.

```
AuthContext(orgName, url)
```

이 API가 사용될 경우 URL에 의해 참조되는 서버만 사용합니다. 해당 서버에서 인증 서비스를 사용할 수 있는 경우라도 페일오버는 이루어지지 않습니다.

2. 인증 컨텍스트는 `AMConfig.properties` 파일의 `com.iplanet.am.server*` 속성에 정의된 서버를 검사합니다.
3. 2단계가 실패할 경우 인증 컨텍스트는 이름 지정 서비스를 사용할 수 있는 서버에서 플랫폼 목록을 조회합니다. 이 플랫폼 목록은 하나의 Directory Server 인스턴스를 공유하는 다수의 Access Manager 인스턴스(일반적으로 페일오버 목적)가 설치될 때 자동으로 만들어집니다.

   예를 들어, 플랫폼 목록에 `Server1`, `Server2` 및 `Server3`의 URL이 포함되면 인증 컨텍스트는 그 중 하나에서 인증이 성공할 때까지 `Server1`, `Server2`, `Server3`을 차례로 순환합니다.

   플랫폼 목록은 이름 지정 서비스의 가용성에 따라 다르므로 항상 동일한 서버에서 얻어질 수 있는 것은 아닙니다. 더욱이 이름 지정 서비스 페일오버가 먼저 일어날 수도 있습니다. `AMConfing.properties`의 `com.iplanet.am.naming.url property`에 다수의 이름 지정 서비스 URL이 지정됩니다. 사용할 수 있는 첫 번째 이름 지정 서비스 URL은 인증 페일오버가 이루어지는 서버 목록이 포함된 서버를 식별하는 데 사용됩니다.

정규화된 도메인 이름(FQDN) 매핑

정규화된 도메인 이름(FQDN) 매핑을 사용하면 사용자가 잘못된 URL을 입력하더라도(예: 보호된 자원에 액세스할 때 부분 호스트 이름이나 IP 주소 지정) 인증 서비스에서 수정할 수 있습니다. FQDN 매핑은 `AMConfig.properties` 파일의 `com.sun.identity.server.fqdnMap` 속성을 수정하여 활성화합니다. 이 등록 정보를 지정하는 형식은 다음과 같습니다.

`com.sun.identity.server.fqdnMap[`*invalid-name* `]=`*valid-name*

invalid-name 값은 사용자가 잘못 입력한 FQDN 호스트 이름이 되고 *valid-name*은 필터에서 사용자를 리디렉션할 실제 호스트 이름이 됩니다. 명시된 요구 사항에 부합한다면 횟수 제한 없이 매핑 지정이 가능합니다(코드 예 1-1 참조). 이 등록 정보를 설정하지 않으면 사용자는 `AMConfig.properties` 파일에서도 확인 가능한 `com.iplanet.am.server.host=` *server_name* 등록 정보에 구성된 기본 서버 이름으로 보내집니다.

예 4–1 `AMConfig.properties`의 FQDN 매핑 속성

```
com.sun.identity.server.fqdnMap[isserver]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[isserver.mydomain]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[
            IP address]=isserver.mydomain.com
```

예 4–1 AMConfig.properties의 FQDN 매핑 속성 (계속)

FQDN 매핑의 용도

이 등록 정보는 서버에 호스트된 응용 프로그램에 둘 이상의 호스트 이름으로 액세스 가능할 경우 둘 이상의 호스트 이름에 대해 하나의 매핑을 만드는 데 사용할 수 있습니다. 또한 Access Manager에서 특정 URL에 대해 수정 조치를 취하지 않게 할 때에도 사용할 수 있습니다. 예를 들어, IP 주소를 사용하여 응용 프로그램에 액세스하는 사용자에게 리디렉션이 필요하지 않다면 이 기능은 다음과 같이 매핑 항목을 지정하여 구현할 수 있습니다.

`com.sun.identity.server.fqdnMap[`*IP address* `]=`*IP address.*

주 – 매핑이 둘 이상 정의되어 있을 때는 잘못된 FQDN 이름으로 값이 겹치지 않아야 합니다. 응용 프로그램에서 액세스하지 못할 수도 있습니다.

영구 쿠키

영구 쿠키는 웹 브라우저를 닫은 후에도 계속 존재하는 쿠키로서, 사용자가 이 영구 쿠키를 사용하면 다시 인증할 필요 없이 새 브라우저 세션으로 로그인할 수 있습니다. 이 쿠키의 이름은 `AMConfig.properties`의 `com.iplanet.am.pcookie.name` 등록 정보에서 정의되며 기본값은 DProPCookie입니다. 쿠키 값은 3DES 암호화된 문자열로서 사용자 DN, 영역 이름, 인증 모듈 이름, 최대 세션 시간, 유휴 시간 및 캐시 시간으로 구성됩니다.

▼ 영구 쿠키를 사용하려면

1. **핵심 인증 모듈에서 영구 쿠키 모드를 설정합니다.**
2. **핵심 인증 모듈에서 영구 쿠키 최대 시간 속성에 대한 시간 값을 구성합니다.**
3. **값이 yes인 iPSPCookie 매개 변수를 사용자 인터페이스 로그인 URL에 추가합니다.**
   사용자가 이 URL을 사용하여 인증하면 브라우저를 닫아도 다시 인증할 필요 없이 새 브라우저 창을 열 수 있고 콘솔로 리디렉션하게 됩니다. 영구 쿠키는 2단계에서 정의한 시간이 경과할 때까지 계속 적용됩니다.

   영구 쿠키 모드는 다음 인증 SPI 방법을 사용하여 설정할 수 있습니다.

```
AMLoginModule.setPersistentCookieOn()
```

레거시 모드에서 다중 LDAP 인증 모듈 구성

페일오버의 한 형식으로 또는 Access Manager 콘솔에 값 필드가 하나만 제공되는 경우 하나의 속성에 여러 값을 구성하기 위해 관리자는 하나의 영역에 여러 LDAP 인증 모듈 구성을 정의할 수 있습니다. 이러한 추가 구성은 콘솔에 표시되지 않더라도 사용자의 인증 요청에 대한 초기 검색이 없는 경우에 기본 구성과 함께 사용됩니다. 예를 들어, 한 영역에서 두 개의 서로 다른 도메인에 인증용 LDAP 서버를 통한 검색을 정의하거나 한 도메인에 사용자 이름 지정 속성을 여러 개 구성할 수도 있습니다. 후자는 콘솔에 텍스트 필드를 하나만 갖는 경우이며, 기본 검색 기준을 사용하여 사용자를 찾지 못하면 LDAP 모듈에서 2차 범위를 사용하여 검색하게 됩니다. 다음은 추가 LDAP를 구성하는 단계입니다.

▼ 추가 LDAP 구성을 추가하려면

1 2차(또는 3차) LDAP 인증 구성에 필요한 새 값과 전체 속성 세트를 포함하여 XML 파일을 만듭니다.

`etc/opt/SUNWam/config/xml`에 있는 `amAuthLDAP.xml`을 확인하여 사용 가능한 속성을 참조할 수 있습니다. 그러나 이 단계에서 만든 XML 파일은 `amAuthLDAP.xml`과 달리 `amadmin.dtd` 구조를 기반으로 합니다. 이 파일에 대해 속성을 하나 또는 전부 정의할 수 있습니다. 코드 예 1-2는 LDAP 인증 구성에 사용할 수 있는 모든 속성 값이 포함된 하위 구성 파일의 예입니다.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  Copyright (c) 2002 Sun Microsystems, Inc. All rights reserved.
  Use is subject to license terms.
-->
<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun ONE Access Manager 6.0 Admin CLI DTD//EN"
    "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<!--
  Before adding subConfiguration load the schema with
GlobalConfiguration defined and replace corresponding
 serviceName and subConfigID in this sample file OR load
 serviceConfigurationRequests.xml before loading this sample
-->
<Requests>
<realmRequests DN="dc=iplanet,dc=com">
    <AddSubConfiguration subConfigName = "ssc"
        subConfigId = "serverconfig"
```

```
priority = "0" serviceName="iPlanetAMAuthLDAPService">

      <AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-server"/>
    <Value>vbrao.red.iplanet.com:389</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-base-dn"/>
    <Value>dc=iplanet,dc=com</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="planet-am-auth-ldap-bind-dn"/>
    <Value>cn=amldapuser,ou=DSAME Users,dc=iplanet,dc=com</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-bind-passwd"/>
    <Value>
          plain text password</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-user-naming-attribute"/>
    <Value>uid</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-user-search-attributes"/>
    <Value>uid</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-search-scope"/>
    <Value>SUBTREE</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-ssl-enabled"/>
    <Value>false</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-return-user-dn"/>
    <Value>true</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-auth-level"/>
    <Value>0</Value>
</AttributeValuePair>
<AttributeValuePair>
    <Attribute name="iplanet-am-auth-ldap-server-check"/>
    <Value>15</Value>
</AttributeValuePair>
```

```
    </AddSubConfiguration>

</realmRequests>
</Requests>
```

2 1단계에서 만든 XML 파일에서 iplanet-am-auth-ldap-bind-passwd 값으로 일반 텍스트 비밀번호를 복사합니다.

이 속성 값은 코드 예에 굵은 글씨로 표시되어 있습니다.

3 amadmin 명령줄 도구를 사용하여 XML 파일을 로드합니다.

```
./amadmin -u amadmin -w administrator_password -v -t name_of_XML_file.
```

이 2차 LDAP 구성은 콘솔에서 보거나 수정할 수 없습니다.

정보 – 다중 LDAP 구성에 사용할 수 있는 샘플이 있습니다. `/AccessManager-base /SUNWam/samples/admin/cli/bulk-ops/`의 `serviceAddMultipleLDAPConfigurationRequests .xml` 명령줄 템플리트를 참조하십시오. `/AccesManager-base/SUNWam/samples/admin/cli/`의 `Readme.html`에 있는 지침을 참조할 수 있습니다.

세션 업그레이드

인증 서비스를 사용하면 한 영역에서 동일한 사용자가 수행한 2차 인증 성공을 기반으로 유효한 세션 토큰을 업그레이드할 수 있습니다. 유효한 세션 토큰을 가진 사용자가 현재 영역에서 보호한 자원에 인증을 시도하고 이 2차 인증 요청이 성공하면 해당 세션은 새 인증을 기반으로 한 새 등록 정보로 업데이트됩니다. 인증에 실패하면 사용자의 현재 세션이 업그레이드되지 않고 반환됩니다. 유효한 세션을 가진 사용자가 다른 영역에서 보호한 자원에 인증을 시도하는 경우 새 영역에 인증할 것인지 묻는 메시지를 받게 됩니다. 이때 사용자는 현재 세션을 유지하거나 새 영역에 인증을 시도할 수도 있습니다. 인증이 성공하면 이전 세션이 삭제되고 새 세션이 만들어집니다.

세션 업그레이드 중 로그인 페이지가 시간 초과되면 원래의 성공 URL로 리디렉션됩니다. 시간 초과 값은 다음에 따라 결정됩니다.

- 각 모듈에 설정한 페이지 시간 초과 값(기본값: 1분)
- `AMConfig.properties`의 `com.iplanet.am.invalidMaxSessionTime` 등록 정보(기본값: 10분)
- `iplanet-am-max-session-time`(기본값: 120분)

com.iplanet.am.invalidMaxSessionTimeout 값 및 iplanet-am-max-session-time 값은 페이지 시간 초과 값보다 커야 합니다. 그렇지 않으면 세션 업그레이드 중 유효한 세션 정보가 손실되고 이전의 성공 URL에 대한 리디렉션이 실패하게 됩니다.

플러그인 인터페이스 검증

관리자는 영역에 적합한 사용자 이름 또는 비밀번호 검증 논리를 만들고 이를 인증 서비스에 플러그인으로 추가할 수 있습니다. (이 기능은 LDAP 및 구성원 인증 모듈에서만 지원됩니다.)사용자를 인증하거나 비밀번호를 변경하기 전에 Access Manager에서는 이 플러그인을 호출합니다. 검증이 성공하면 인증이 계속되지만, 실패하면 인증 실패 페이지가 나타납니다. 이 플러그인은 서비스 관리 SDK의 일부인 com.iplanet.am.sdk.AMUserPasswordValidation 클래스를 확장합니다. 이 SDK에 대한 자세한 내용은 Access Manager Javadocs의 com.iplanet.am.sdk 패키지를 참조하십시오.

▼ 검증 플러그인을 작성 및 구성하려면

1. **새 플러그인 클래스는 com.iplanet.am.sdk. AMUserPasswordValidation 클래스를 확장하고 validateUserID() 및 validatePassword() 메소드를 구현합니다. AMException은 검증이 실패할 때 나타납니다.**
2. **플러그인 클래스를 컴파일하고 원하는 위치에 .class 파일을 놓습니다. 런타임 동안 Access Manager에서 액세스할 수 있도록 클래스 경로를 업데이트합니다.**
3. **Access Manager 콘솔에 최상위 관리자로 로그인합니다. [서비스 관리] 탭을 누르고 관리 서비스에 대한 속성을 확인합니다. 사용자 아이디 및 비밀번호 검증 플러그 인 클래스 필드에 플러그 인 클래스의 이름(패키지 이름 포함)을 입력합니다.**
4. **로그아웃한 후 다시 로그인합니다.**

JAAS 공유 상태

JAAS 공유 상태는 인증 모듈들이 사용자 아이디와 비밀번호를 공유하게 합니다. 다음 인증 모듈에 옵션이 정의되어 있습니다.

- 영역(또는 조직)
- 사용자
- 서비스
- 역할

실패할 때 모듈에는 필수 자격 증명에 대한 메시지가 나타납니다. 인증이 실패하고 나면 모듈의 실행이 중지되거나 로그아웃 공유 상태가 지워집니다.

JAAS 공유 상태 활성화

JAAS 공유 상태를 구성하려면 다음을 수행합니다.

- `iplanet-am-auth-shared-state-enabled` 옵션을 사용합니다.
- 공유 상태 옵션은 다음의 경우에 사용됩니다.
  `iplanet-am-auth-shared-state-enabled=true`
- 이 옵션의 기본값은 true입니다.
- 이 변수는 인증 체이닝 구성의 [옵션] 열에서 지정됩니다.

실패하면 인증 모듈에는 필수 자격 증명 프롬프트가 JAAS 사양에 제시된 `tryFirstPass` 옵션 동작에 따라 나타납니다.

JAAS 공유 상태 저장소 옵션

JAAS 공유 상태 저장소 옵션을 구성하려면 다음을 수행합니다.

- `iplanet-amauth-store-shared-state-enabled` 옵션을 사용합니다.
- 저장소 공유 상태 옵션은 다음과 같은 경우에 사용됩니다.
  `iplanet-am-auth-store-shared-state-enabled=true`
- 이 옵션의 기본값은 false입니다.
- 이 변수는 인증 체이닝 구성의 [옵션] 열에서 지정됩니다.

완결, 중단 또는 로그아웃한 후에는 공유 상태가 지워집니다.

◆ ◆ ◆ 5 장

5 정책 관리

이 장에서는 Sun Java™ System Access Manager의 정책 관리 기능에 대해 설명합니다. Access Manager의 정책 관리 기능을 사용하면 최상위 수준 관리자 또는 최상위 수준 정책 관리자가 모든 영역에서 사용할 수 있는 특정 서비스의 정책을 보고, 만들고, 삭제하고, 수정할 수 있습니다. 또한 영역이나 하위 영역 관리자 또는 정책 관리자가 영역 수준에서 정책을 보고, 만들고, 삭제하고, 수정할 수 있는 방법을 제공합니다.

이번 장은 다음 절로 구성됩니다.

개요

정책은 조직의 보호 대상 자원에 대한 액세스 권한을 지정하는 규칙을 정의합니다. 보호하고 관리하고 모니터링해야 하는 자원, 응용 프로그램 및 서비스가 있습니다. 정책은 주어진 자원에 대한 작업을 사용자가 언제 어떤 방법으로 수행할 수 있는지 정의하여 이러한 자원에 대한 액세스 권한과 용도를 제어합니다. 정책은 특정 기본에 대해 자원을 정의합니다.

주 – 기본은 아이디를 가질 수 있는 개인, 회사, 역할 또는 그룹이 될 수 있습니다. 자세한 내용은 Java™ 2 Platform Standard Edition Javadoc (`http://java.sun.com/j2se/1.4.2/docs/api/java/security/Principal.html`)을 참조하십시오.

단일 정책은 이진 또는 비 이진 결정 중 하나를 정의할 수 있습니다. 이진 결정은 *yes*/*no*, *true*/*false* 또는 *allow*/*deny* 중에서 정의할 수 있습니다. 비 이진 결정은 속성의 값을 나타냅니다. 예를 들어, 메일 서비스에는 각 사용자에 대한 최대 저장 값이 설정된 `mailboxQuota` 속성이 포함될 수 있습니다. 일반적으로 정책은 한 기본이 어떤 자원에 대해 어떤 조건 하에서 어떤 작업을 수행할 수 있는지 정의하도록 구성됩니다.

정책 관리 기능

정책 관리 기능은 정책을 만들고 관리하기 위한 **정책 서비스**를 제공합니다. 정책 서비스는 관리자가 Access Manager 배포 내에서 자원을 보호하기 위해 권한을 정의, 수정, 부여, 철회 및 삭제할 수 있도록 합니다. 일반적으로 정책 서비스에는 데이터 저장소, 생성을 허용하는 인터페이스 라이브러리, 정책 관리 및 평가, 정책 집행자 또는 **정책 에이전트**가 포함됩니다. 기본적으로 Sun Java Enterprise System Directory Server를 사용하여 Access Manager는 데이터를 저장하며, 정책 평가 및 정책 서비스 사용자 정의를 위해 Java와 C API를 제공합니다. 자세한 내용은 **Sun Java System Access Manager 7.1 Developer's Guide**를 참조하십시오. 또한 관리자가 Access Manager 콘솔을 사용하여 정책을 관리할 수 있게 해줍니다. Access Manager는 다운로드할 수 있는 정책 에이전트를 사용하여 정책을 집행하는 정책 가능 서비스인 URL 정책 에이전트 서비스를 제공합니다.

URL 정책 에이전트 서비스

Access Manager를 설치하면 HTTP URL 보호를 위한 정책을 정의하는 URL 정책 에이전트 서비스가 제공됩니다. 이 서비스를 사용하여 관리자는 정책 집행자 또는 **정책 에이전트**를 통해 정책을 만들고 관리할 수 있습니다.

정책 에이전트

정책 에이전트는 회사의 자원이 저장된 서버에 대한 정책 적용 지점(PEP)입니다. 정책 에이전트는 웹 서버에 Access Manager와 별도로 설치되며 사용자가 보호를 받는 웹 서버에 있는 웹 자원에 대한 요청을 보낼 때 추가 인증 단계 역할을 합니다. 이 인증 단계는 자원에서 수행하는 사용자 인증 요청에 추가로 이루어집니다. 에이전트는 웹 서버를 보호하고 자원은 인증 플러그 인에 의해 보호됩니다.

예를 들어, 원격 설치된 Access Manager에 의해 보호되는 인적 자원 웹 서버에는 에이전트가 설치되어 있을 수 있습니다. 이 에이전트는 제대로된 정책 없이 기밀 정보인 봉급 정보나 기타 민감한 데이터를 보지 못하도록 방지합니다. 정책은 Access Manager 관리자가 정의하여 Access Manager 배포 내에 저장하며 정책 에이전트가 원격 웹 서버의 내용에 대한 사용자 액세스를 허용 또는 거부하는 데 사용됩니다.

최신 Access Manager 정책 에이전트는 Sun Microsystems 다운로드 센터에서 다운로드할 수 있습니다.

정책 에이전트 설치 및 관리에 대한 자세한 내용은 **Sun Java System Access Manager Policy Agent 2.2 User's Guide**를 참조하십시오.

주 – 정책은 특별한 순서로 평가되지 않습니다. 그러나 정책을 평가할 때 한 가지 작업 값이 **거부**로 평가되는 경우 정책 구성 서비스에서 거부 결정에 대한 평가 계속 속성이 활성화되지 않으면 후속 정책은 평가되지 않습니다.

Access Manager 정책 에이전트는 웹 URL(`http://...` 또는 `https//...`)에서만 결정을 실행합니다. 그러나 Java 및 C 정책 평가 API를 사용하여 다른 자원에서 정책을 실행하는 에이전트를 작성할 수 있습니다.

또한 정책 구성 서비스의 자원 비교기 속성도 기본 구성에서 다음과 같은 구성으로 변경해야 합니다.

```
serviceType=Name_of_LDAPService
|class=com.sun.identity.policy.plugins.SuffixResourceName|wildcard=*

|delimiter=,|caseSensitive=false
```

또는 `LDAPResourceName`과 같은 구현을 제공하여 `com.sun.identity.policy.interfaces.ResourceName`을 구현하고 자원 비교기를 구성하는 방법도 사용할 수 있습니다.

정책 에이전트 프로세스

보호 대상 웹 자원을 위한 프로세스는 정책 에이전트에 의해 보호를 받는 서버에 상주하는 URL을 웹 브라우저에서 요청할 때 시작됩니다. 서버의 설치된 정책 에이전트는 요청을 인터셉트하여 기존 인증 자격 증명(세션 토큰)을 확인합니다.

에이전트가 요청을 인터셉트하고 기존 세션 토큰을 확인하면 다음 프로세스가 이어집니다.

1. 세션 토큰이 유효하면 사용자에게 권한이 부여 또는 거부됩니다. 유효한 토큰이 아닐 경우 사용자는 다음과 같은 단계를 거쳐 인증 서비스로 리디렉션됩니다.

   에이전트가 기존 세션 토큰이 없는 요청을 인터셉트했다면 다른 인증 방법을 사용하여 자원을 보호하더라도 사용자를 로그인 페이지로 리디렉션합니다.
2. 사용자의 자격 증명이 인증되면 에이전트가 Access Manager의 내부 서비스 연결에 사용되는 URL을 정의하는 이름 지정 서비스에 요청을 발행합니다.
3. 자원이 에이전트에서 구성된 비강제 목록과 일치하면 액세스가 허용됩니다.
4. 이름 지정 서비스는 정책 서비스에 대한 로케이터, 세션 서비스 및 로깅 서비스를 반환합니다.
5. 에이전트는 사용자에게 적용할 수 있는 정책 결정을 얻기 위해 정책 서비스에 요청을 보냅니다.

6. 액세스 대상 자원에 대한 정책 결정에 따라 사용자는 액세스 권한이 부여되거나 거부됩니다. 정책 결정에 대한 조언에 다른 인증 수준 또는 방법이 제시되면 에이전트는 모든 검색 조건이 확인될 때까지 요청을 인증 서비스로 다시 보냅니다.

정책 유형

Access Manager를 사용하여 다음 두 가지 유형의 정책을 구성할 수 있습니다.

- 94 페이지 "일반 정책"
- 99 페이지 "참조 정책"

일반 정책

Access Manager에서 액세스 권한을 정의하는 정책을 **일반** 정책이라고 합니다. 일반 정책은 **규칙**, **주제**, **조건** 및 **응답 공급자**로 구성됩니다.

규칙

규칙에는 서비스 유형, 하나 이상의 작업 및 값이 포함되어 있습니다. 기본적으로 규칙이 정책을 정의합니다.

- 서비스 유형은 보호하고 있는 자원 유형을 정의합니다.
- **작업**은 자원에 대해 수행될 수 있는 작업의 이름입니다. 예를 들어, 웹 서버 작업으로는 POST 또는 GET 등이 있습니다. 인적 자원 서비스에는 집 전화 번호 변경 작업 등이 허용될 수 있습니다.
- **값**은 작업에 대한 권한(예: 허용 또는 거부)을 정의합니다.

주 – 일부 서비스에 대해 자원 없이 작업을 정의할 수 있습니다.

주제

주제는 정책의 영향을 받는 사용자 또는 사용자 모음(예: 특정 역할을 가진 그룹 또는 사용자)을 정의합니다. 사용자가 적어도 정책의 한 주제의 구성원일 경우에만 정책이 적용되는 것이 일반적인 규칙입니다. 기본 주제는 다음과 같습니다.

Access Manager Identity 주제	이 주제는 사용자가 영역 주제 탭에서 만들고 관리하는 Identity를 해당 주제의 구성원으로 추가할 수 있다는 것을 나타냅니다.
인증된 사용자	이 주제 유형은 유효한 SSO 토큰을 가진 사용자가 이 주제의 구성원이라는 것을 나타냅니다. 인증된 사용자는 정책이 정의된 조직과 다른 영역에 인증한 경우에도 모두 이 주제의 구성원이 됩니다. 이는

자원 소유자가 관리되는 자원에 대한 액세스 권한을 다른 조직의 사용자에게 제공하는 경우에 유용합니다. 보호되는 자원에 대한 액세스 권한을 특정 조직의 구성원에게만 제한하려면 조직 주제를 사용하십시오.

웹 서비스 클라이언트

이 주제 유형은 SSO 토큰에 포함된 기본의 DN이 이 주제의 선택된 임의 값과 일치할 경우 SSO 토큰으로 식별된 웹 서비스 클라이언트(WSC)가 이 주제의 구성원이라는 것을 나타냅니다. 유효한 값은 로컬 JKS 키 저장소에 있는 신뢰할 수 있는 인증서(신뢰할 수 있는 WSC의 인증서에 해당)의 DN입니다. 이 주제는 리버티 웹 서비스 프레임워크에 대해 종속성을 가지며 리버티 서비스 공급자가 WSC를 인증하기 위해서만 사용해야 합니다.

이 주제를 정책에 추가하기 전에 키 저장소를 만들어야 합니다. 키 저장소 설정에 대한 내용은 다음 사이트를 참조하십시오.

AccessManager-base
`/SUNWam/samples/saml/xmlsig/keytool.html`

해당 영역의 정책 구성 서비스에서 주제를 선택하여 다음과 같은 추가 주제를 사용할 수 있습니다.

Access Manager 역할

이 주제 유형은 Access Manager 역할의 구성원이 이 주제의 구성원이라는 것을 나타냅니다. Access Manager를 레거시 모드로 실행하거나 6.3 기반 콘솔을 사용하여 Aeecss Manager 역할을 만듭니다. 이러한 역할은 Access Manager에 의해 위임되는 객체 클래스를 가집니다. Access Manager 역할은 Access Manager 정책 서비스를 호스트하는 방법으로만 액세스할 수 있습니다.

LDAP 그룹

이 주제 유형은 LDAP 그룹의 구성원이 이 주제의 구성원이라는 것을 나타냅니다.

LDAP 역할

이 주제 유형은 LDAP 역할의 구성원이 이 주제의 구성원이라는 것을 나타냅니다. LDAP 역할은 Directory Server 역할 기능을 사용하는 임의의 역할 정의입니다. 이러한 역할은 Directory Server 역할 정의에 의해 위임되는 객체 클래스를 가집니다. 정책 구성 서비스에서 LDAP 역할 검색 필터를 수정하여 범위를 좁히고 성능을 향상시킬 수 있습니다.

LDAP 사용자

이 주제 유형은 LDAP 사용자가 이 주제의 구성원이라는 것을 나타냅니다.

조직	이 주제 유형은 영역이 이 주제의 구성원이라는 것을 나타냅니다.

Access Manager 역할 대 LDAP 역할

Access Manager 역할은 Access Manager를 사용하여 작성됩니다. 이러한 역할은 Access Manager에 의해 위임되는 객체 클래스를 가집니다. LDAP 역할은 Directory Server 역할 기능을 사용하는 임의의 역할 정의입니다. 이러한 역할은 Directory Server 역할 정의에 의해 위임되는 객체 클래스를 가집니다. 모든 Access Manager 역할은 Directory Server 역할로 사용될 수 있습니다. 그러나 모든 Directory Server 역할이 Access Manager 역할은 아닙니다. 119 페이지 "정책 구성 서비스"를 구성하여 기존 디렉토리에서 LDAP 역할을 활용할 수 있습니다. Access Manager 역할은 Access Manager 정책 서비스를 호스트하는 방법으로만 액세스할 수 있습니다. 정책 구성 서비스에서 LDAP 역할 검색 필터를 수정하여 범위를 좁히고 성능을 향상시킬 수 있습니다.

중첩된 역할

중첩된 역할은 정책 정의의 주제에서 LDAP 역할로 올바르게 평가될 수 있습니다.

조건

조건을 사용하여 정책에 대한 제약 조건을 정의할 수 있습니다. 예를 들어, 급여 응용 프로그램에 대한 정책을 정의할 경우 지정된 시간 동안만 응용 프로그램에 대한 액세스를 제한하는 조건을 현재 작업에서 정의할 수 있습니다. 또는 주어진 IP 주소 집합이나 회사 인트라넷에서 요청을 보낸 경우에만 작업을 허가하는 조건을 정의할 수 있습니다.

조건을 추가로 사용하여 동일한 도메인에서 다른 URL에 대한 다른 정책을 구성할 수 있습니다. 예를 들어 오전 9시에서 오후 5시까지만 `org.example.net`에서 `http://org.example.com/hr/*jsp`에 액세스할 수 있습니다. 이 작업은 시간 조건과 함께 IP 조건을 사용하여 수행할 수 있습니다. 규칙 자원을 `http://org.example.com/hr/*.jsp`로 지정할 경우 `http://org.example.com/hr` 및 하위 디렉토리에 있는 모든 JSP 정책이 적용됩니다.

주 – 참조, 규칙, 자원, 주제, 조건, 작업 및 값 등의 용어는 `policy.dtd`의 *Referral*, *Rule*, *ResourceName*, *Subject*, *Condition*, *Attribute* 및 *Value* 요소에 해당합니다.

추가할 수 있는 기본 조건은 다음과 같습니다.

활성 세션 시간

사용자 세션 데이터를 기반으로 조건을 설정합니다. 수정할 수 있는 필드는 다음과 같습니다.

최대 세션 시간	세션이 시작할 때부터 정책을 적용할 수 있는 최대 기간을 지정합니다.
세션 종료	선택된 경우 세션 시간이 최대 세션 시간 필드에 정의된 허용되는 최대 시간을 초과하면 사용자 세션이 종료됩니다.

인증 체인

지정된 영역에서 사용자가 인증 체인에 대해 성공적으로 인증된 경우 해당 정책이 적용됩니다. 영역을 지정하지 않으면 인증 체인의 모든 영역에 대한 인증이 조건을 만족하게 됩니다.

인증 수준(보다 크거나 같음)

사용자의 인증 수준이 조건에 설정된 인증 수준보다 높거나 같은 경우에 정책이 적용됩니다. 이 속성은 지정된 영역에서 인증에 대한 신뢰 수준을 나타냅니다.

인증 수준(보다 낮거나 같음)

사용자의 인증 수준이 조건에 설정된 인증 수준보다 낮거나 같은 경우에 정책이 적용됩니다. 이 속성은 지정된 영역에서 인증에 대한 신뢰 수준을 나타냅니다.

인증 모듈 인스턴스

지정된 영역에서 사용자가 인증 모듈에 대해 성공적으로 인증된 경우 해당 정책이 적용됩니다. 영역을 지정하지 않으면 인증 모듈의 모든 영역에 대한 인증이 조건을 만족하게 됩니다.

현재 세션 등록 정보

사용자의 Access Manager 세션에 설정된 등록 정보의 값을 기반으로 정책을 요청에 적용할 수 있는지 여부를 결정합니다. 정책 평가 중 조건에 정의된 모든 등록 정보 값이 사용자의 세션에 있는 경우에만 true를 반환합니다. 조건에 여러 값으로 정의된 등록 정보의 경우 조건의 등록 정보에 대해 나열된 값이 토큰에 하나 이상 있으면 충분합니다.

IP 주소/DNS 이름

IP 주소 범위에 따라 조건을 설정합니다. 정의할 수 있는 필드는 다음과 같습니다.

보내는/받는 IP 주소	IP 주소의 범위를 지정합니다.
DNS 이름	DNS 이름을 지정합니다. 이 필드는 정규화된 호스트 이름이나 다음 형식의 문자열이 될 수 있습니다. *domainname*

**.domainname*

LDAP 필터 조건

정의된 LDAP 필터가 정책 구성 서비스에 지정된 LDAP 디렉토리에서 사용자 항목을 찾은 경우 해당 정책이 적용됩니다. 해당 정책이 정의된 영역에서만 적용할 수 있습니다.

영역 인증

지정된 영역에 대해 사용자가 인증된 경우 정책이 적용됩니다.

시간(요일, 날짜, 시간 및 표준 시간대)

시간 제약 조건을 기반으로 조건을 설정합니다. 필드는 다음과 같습니다.

시작/끝 날짜	날짜의 범위를 지정합니다.
시간	하루 중 시간의 범위를 지정합니다.
요일	요일의 범위를 지정합니다.
표준 시간대	표준 또는 사용자 정의 표준 시간대를 지정합니다. 사용자 정의 표준 시간대는 Java에서 구성한 표준 시간대 아이디(예: PST)만 될 수 있습니다. 지정된 값이 없을 경우 기본값은 Access Manager JVM에 설정된 표준 시간대입니다.

응답 공급자

응답 공급자는 정책 기반 응답 속성을 제공하는 플러그인입니다. 응답 공급자 속성은 정책 결정과 함께 PEP로 전송됩니다. Access Manager에는 하나의 구현인 `IDResponseProvider`가 포함되어 있습니다. 사용자 정의 응답 공급자는 이 버전의 Access Manager에서 지원되지 않습니다. 에이전트, PEP는 보통 이러한 응답 속성을 헤더로 응용 프로그램에 전달합니다. 응용 프로그램은 일반적으로 이러한 속성을 사용하여 포털 페이지와 같은 응용 프로그램 페이지를 사용자 설정합니다.

정책 권고

조건에 따라 결정한 대로 정책을 적용할 수 없을 때는 그 조건에서 해당 정책을 요청에 적용할 수 없는 이유를 나타내는 권고 메시지를 만듭니다. 이러한 권고 메시지는 정책 적용 지점에 대한 정책 결정에 전달됩니다. 정책 적용 지점은 이 권고를 검색하고 더 높은 수준으로 인증하는 인증 메커니즘으로 사용자를 리디렉션하는 등의 적당한 조치를 취하게 됩니다. 적합한 조치가 취해진 후 정책이 적용 가능하게 되면 사용자에게 더 높은 수준의 인증에 관한 프롬프트가 나타나 자원에 액세스할 수 있게 됩니다.

자세한 내용은 다음 클래스를 참조하십시오.

`com.sun.identity.policy.ConditionDecision.getAdvices()`

해당 조건이 충족되지 않으면 `AuthLevelCondiiton` 및 `AuthSchemeCondition` 에서 권고를 제공합니다.

`AuthLevelCondition` 권고는 다음 키와 관련되어 있습니다.

`com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_LEVEL_CONDITION_ADVICE`

`AuthSchemeCondition` 권고는 다음 키와 관련되어 있습니다.

`com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_SCHEME_CONDITION_ADVICE`

사용자 정의한 조건도 권고를 만들 수 있습니다. 그러나 Access Manager 정책 에이전트는 인증 수준 권고와 인증 스키마 권고에만 응답합니다. 사용자 정의 에이전트를 작성하고 기존 Access Manager 에이전트를 확장하여 더 많은 권고를 이해하고 응답할 수 있습니다. 자세한 내용은 **Sun Java System Access Manager Policy Agent 2.2 User's Guide**를 참조하십시오.

참조 정책

관리자는 한 영역의 정책 정의와 결정을 다른 영역에 위임해야 할 수 있습니다. 또는 자원에 대한 정책 결정을 다른 정책 제품에 위임할 수 있습니다. **참조** 정책은 정책 작성과 평가를 위해 이 정책 위임을 제어합니다. 이 정책은 하나 이상의 **규칙**과 하나 이상의 **참조**로 구성됩니다.

정책 구성 서비스에는 조직 별칭 참조라고 하는 전역 속성이 포함되어 있습니다. 이 속성을 사용하면 최상위 수준 또는 상위 영역에서 참조 정책을 만들지 않고도 하위 영역에서 정책을 만들 수 있습니다. 정규화된 호스트 이름이 해당 영역의 영역/DNS 별칭과 일치하는 HTTP 또는 HTTPS 자원을 보호하는 정책만 만들 수 있습니다. 기본적으로 이 속성은 [아니오]로 정의됩니다.

규칙

규칙은 정책 정의와 평가가 참조되는 자원을 정의합니다.

참조

참조는 정책 평가가 참조되는 조직을 정의합니다. 기본적으로 참조에는피어 영역과 하위 영역이 있습니다. 이러한 참조는 각각 동일한 수준의 영역과 하위 수준의 영역에 위임됩니다. 자세한 내용은 111 페이지 "피어 영역 및 하위 영역에 대한 정책 만들기"를 참조하십시오.

주 – 참조 대상 영역은 참조된 자원 또는 그 하위 자원에 대해서만 정책을 정의하거나 평가할 수 있습니다. 그러나 이 제한은 최상위 영역에는 적용되지 않습니다.

정책 정의 유형 문서

일단 작성하여 구성한 정책은 Directory Server에 XML 파일로 저장됩니다. Directory Server에서 XML로 인코딩된 데이터는 한 장소에 저장됩니다. `amAdmin.dtd`(또는 콘솔)를 사용하여 정책을 정의하고 구성하지만 실제로 Directory Server에는 `policy.dtd`를 기반으로 한 XML로 저장됩니다. `policy.dtd`에는 정책 작성 태그가 없고 `amAdmin.dtd`에서 추출한 정책 요소 태그가 포함됩니다. 그러므로 정책 서비스는 Directory Server에서 정책을 로드할 때 `policy.dtd`를 기반으로 XML의 구문을 분석합니다. `amAdmin.dtd`는 명령줄을 사용하여 정책을 만들 때만 사용됩니다. 이 절에서는 `policy.dtd`의 구조에 대해 설명합니다. `policy.dtd`는 다음 위치에 있습니다.

```
AccessManager-base/SUNWam/dtd(Solairs)
AccessManager-base/identity/dtd(Linux)
AccessManager-base/identity/dtd(HP-UX)
AccessManager-base\identity\dtd(Windows)
```

주 – 이 장에서는 Solaris 디렉토리에 대한 내용만 설명합니다. Linux, HP-UX 및 Windows의 디렉토리 구조는 서로 다르므로 주의해 주십시오.

Policy 요소

Policy 요소는 정책의 권한 또는 **규칙**과 규칙 적용 대상 또는 **주제**를 정의하는 루트 요소입니다. 또한 정책이 **참조**(위임) 정책인지 아닌지 여부와 제한(또는 **조건**)이 있는지 여부도 정의합니다. Policy 요소에는 **규칙**, **조건**, **주제**, **참조** 또는 **응답 공급자**와 같은 하위 요소가 하나 이상 포함될 수 있습니다. 필수 XML 속성은 정책의 이름을 지정하는 `name` 속성입니다. `referralPolicy` 속성은 정책이 참조 정책인지 여부를 나타내며 정의하지 않을 경우 기본값은 일반 정책입니다. 선택 XML 속성은 *name* 속성과 *description* 속성입니다.

주 – 정책에 **참조**라는 태그를 붙이면 정책 평가 시 주제와 조건은 무시됩니다. 반대로 **일반**이라는 태그를 붙이면 정책을 평가할 때 참조가 무시됩니다.

Rule 요소

Rule 요소는 정책에 대한 구체적인 사항을 정의하며 *ServiceName*, *ResourceName* 또는 *AttributeValuePair*의 3가지 하위 요소를 취할 수 있습니다. Rule 요소는 정책이 만들어졌던 서비스 또는 응용 프로그램의 유형과 자원 이름, 수행되는 작업을 정의합니다. 규칙은 작업 없이 정의될 수 있습니다. 예를 들어, 참조 정책 규칙에는 작업이 없습니다.

주 – *ResourceName* 요소가 정의되지 않은 정책을 정의할 수도 있습니다.

ServiceName 요소

ServiceName 요소는 정책이 적용되는 서비스의 이름을 정의합니다. 이 요소는 서비스 유형을 나타내며 이 요소에는 다른 요소가 포함되지 않습니다. 이 요소의 값은 `sms.dtd`를 기반으로 서비스의 XML 파일에 정의된 값과 같습니다. *ServiceName* 요소의 XML 서비스 속성은 문자열 값을 취하는 서비스의 이름입니다.

ResourceName 요소

ResourceName 요소는 작업 수행 대상인 객체를 정의합니다. 정책은 이 객체를 보호하도록 특별히 구성되었습니다. 이 요소에는 다른 요소가 포함되지 않습니다. *ResourceName* 요소의 XML 서비스 속성은 객체의 이름입니다. *ResourceName*의 예로는 웹 서버의 `http://www.sunone.com:8080/images` 또는 디렉토리 서버의 `ldap://sunone.com:389/dc=example,dc=com` 등이 있을 수 있습니다. 보다 구체적인 예를 들면 `salary://uid=jsmith,ou=people,dc=example,dc=com` 자원이 있을 수 있습니다. 이 예에서 작업이 수행되는 객체는 John Smith의 급여 정보입니다.

AttributeValuePair 요소

AttributeValuePair 요소는 작업과 그 작업의 값을 정의합니다. 이 요소는 102 페이지 “Subject 요소”, 103 페이지 “Referral 요소” 및 103 페이지 “Condition 요소”의 하위 요소로 사용됩니다. *Attribute* 요소와 *Value* 요소가 모두 포함되며 XML 서비스 속성은 포함되지 않습니다.

Attribute 요소

Attribute 요소는 작업의 이름을 정의합니다. 작업은 자원에 대해 수행되는 작업 또는 이벤트입니다. POST 또는 GET는 웹 서버 자원에 대해 수행되는 작업이며 READ 또는 SEARCH는 디렉토리 서버 자원에 대해 수행되는 작업입니다. *Attribute* 요소는 *Value* 요소와 함께 사용되어야 합니다. *Attribute* 요소 자체는 다른 요소를 포함하지 않습니다. *Attribute* 요소의 XML 서비스 속성은 작업의 이름입니다.

Value 요소

Value 요소는 작업 값을 정의합니다. 작업 값으로는 허용/거부 또는 예/아니요 등이 있습니다. 그 밖의 작업 값은 부울, 숫자 또는 문자열일 수 있습니다. 작업 값은 `sms.dtd`를 기반으로 서비스의 XML 파일에 정의됩니다. *Value* 요소는 다른 요소를 포함하지 않으며 XML 서비스 속성도 포함하지 않습니다.

주 – 거부 규칙은 허용 규칙보다 항상 우선됩니다. 예를 들어 한 정책이 액세스를 거부하고 다른 정책은 허용할 경우, 두 정책에 대한 다른 모든 조건은 충족된다고 가정할 때 잠재적인 충돌이 일어날 수 있으므로 거부 정책을 사용할 때는 매우 주의해야 합니다. 명시적인 거부 규칙이 사용될 경우 역할이나 그룹 구성원 처럼 다른 주제를 통해 사용자에게 할당된 정책 때문에 액세스가 거부될 수 있습니다. 일반적으로 정책 정의 프로세스에서는 허용 규칙만 사용해야 합니다. 기본 거부는 다른 정책이 적용되지 않을 때 사용될 수 있습니다.

Subjects 요소

Subjects 하위 요소는 정책이 적용되는 객체의 집합을 식별합니다. 이 집합은 그룹의 구성원, 역할의 소유자 또는 개인 사용자에 따라 선택됩니다. 이 요소의 하위 요소는 *Subject*입니다. 정의할 수 있는 XML 속성은 다음과 같습니다.

name. 이 속성은 컬렉션의 이름입니다.

description. 이 속성은 주제에 대한 설명입니다.

includeType. 이 속성은 현재 사용되지 않습니다.

Subject 요소

Subject 하위 요소는 정책이 적용되는 기본의 집합을 식별합니다. 이 집합은 Subjects 요소에 의해 정의되는 집합에서 보다 구체적인 객체들의 집합을 가려낸 것입니다. 이 집합의 구성원은 역할, 그룹 구성원 또는 개별 사용자를 기반으로 할 수 있습니다. 이 요소에는 하위 요소인 101 페이지 “AttributeValuePair 요소”가 포함됩니다. 필수 XML 속성은 `type`입니다. 이 속성은 정의된 주제가 취해지는 객체의 집합을 식별합니다. 다른 XML 속성으로는 집합의 이름을 정의하는 `name` 속성과 집합이 정의된 대로인지 정책이 Subject의 구성원이 아닌 사용자에게 적용되는지 여부를 정의하는 `includeType` 속성이 있습니다.

주 – 다수의 Subjects를 정의할 때는 최소한 그 중 하나가 정책이 적용될 사용자에게 적용되어야 합니다. false로 설정된 `includeType`으로 Subject를 정의한 경우 사용자는 적용할 정책에 대한 해당 Subject의 구성원이 아니어야 합니다.

Referrals 요소

Referrals 하위 요소는 정책 참조 집합을 식별합니다. 이 요소는 *Referral* 하위 요소를 취합니다. 정의될 수 있는 XML 속성은 집합의 이름을 정의하는 `name` 속성과 설명을 취하는 `description` 속성입니다.

Referral 요소

Referral 하위 요소는 특정 정책 참조를 식별합니다. 이 요소는 101 페이지 "AttributeValuePair 요소"를 하위 요소로 취합니다. 필수 XML 속성은 구체적으로 정의된 참조를 취하는 할당의 집합을 식별하는 `type` 속성입니다. 집합의 이름을 정의하는 `name` 속성도 포함될 수 있습니다.

Conditions 요소

Conditions 하위 요소는 정책 제한 사항(시간 범위, 인증 수준 등)의 집합을 식별합니다. 이 요소는 하나 이상의 *Condition* 하위 요소를 포함해야 합니다. 정의될 수 있는 XML 속성은 집합의 이름을 정의하는 `name` 속성과 설명을 취하는 `description` 속성입니다.

주 – Conditions 요소는 정책의 선택 요소입니다.

Condition 요소

Condition 하위 요소는 특정 정책 제한 사항(시간 범위, 인증 수준 등)을 식별합니다. 이 요소는 101 페이지 "AttributeValuePair 요소"를 하위 요소로 취합니다. 필수 XML 속성은 구체적으로 정의된 조건을 취하는 제한 사항의 집합을 식별하는 `type` 속성입니다. 집합의 이름을 정의하는 `name` 속성도 포함될 수 있습니다.

정책 가능 서비스 추가

지정된 서비스의 자원에 대한 정책은 서비스 방식에 sms.dtd 에 구성되는 <Policy> 요소가 있는 경우에만 정의할 수 있습니다.

기본적으로 Access Manager는 URL 정책 에이전트 서비스(iPlanetAMWebAgentService)를 제공합니다. 이 서비스는 다음 디렉토리에 있는 XML 파일에 정의됩니다.

```
/etc/opt/SUNWam/config/xml/
```

그러나 Access Manager에 정책 서비스를 추가할 수 있습니다. 일단 정책 서비스가 만들어지면 amadmin 명령줄 유틸리티를 통해 Access Manager에 추가합니다.

▼ 새 정책 사용 가능 서비스를 추가하려면

1 **sms.dtd에 따라 XML 파일 형식의 새 정책 서비스를 개발합니다. Access Manager는 두 가지 정책 서비스 XML 파일을 제공하며 사용자는 다음과 같은 새 정책 서비스 파일을 기준으로 사용하게 됩니다.**

amWebAgent.xml - 기본 URL 정책 에이전트 서비스를 위한 XML 파일로/etc/opt/SUNWam/config/xml/에 있습니다.

SampleWebService.xml- AccessManager-base/samples/policy에 있는 샘플 정책 서비스 파일입니다.

2 **새 정책 서비스를 로드할 디렉토리에 XML 파일을 저장합니다. 예를 들면 다음과 같습니다.**

```
/config/xml/newPolicyService.xml
```

3 **amadmin 명령줄 유틸리티를 사용하여 새 정책 서비스를 로드합니다. 예를 들면 다음과 같습니다.**

```
AccessManager-base/SUNWam/bin/amadmin
    --runasdn “uid=amAdmin,ou=People,default_org,
root_suffix
    --password password
    --schema /config/xml/newPolicyService.xml
```

4 **새 정책 서비스를 로드한 후 amadmin을 통해 새 정책을 로드하거나 Access Manager 콘솔을 통해 정책 정의 규칙을 정의할 수 있습니다.**

정책 만들기

정책 API와 Access Manager 콘솔을 통해 정책을 만들고 수정하고 삭제할 수 있으며 amadmin 명령줄 도구를 통해 정책을 만들고 삭제할 수 있습니다. amadmin 유틸리티를 사용하여 XML의 정책을 가져오고 나열할 수도 있습니다. 이 절에서는 amadmin 명령줄 유틸리티와 Access Manager 콘솔을 통해 정책을 만드는 방법에 대해 설명합니다. 정책 API에 대한 자세한 내용은 **Sun Java System Access Manager 7.1 Developer's Guide**를 참조하십시오.

정책은 일반적으로 XML 파일을 사용하여 만들어지며 amadmin 명령줄 유틸리티를 통해 Access Manager에 추가된 후 Access Manager 콘솔을 사용하여 관리됩니다(콘솔을 사용하여 정책을 만들 수도 있음). amadmin을 사용하여 직접 정책을 수정할 수 없기 때문입니다. 정책을 수정하려면 Access Manager에서 정책을 삭제한 다음 amadmin을 사용하여 수정된 정책을 추가해야 합니다.

일반적으로 정책은 영역(또는 하위 영역) 수준에서 만들어져 영역 트리 전체에 사용됩니다.

▼ amadmin을 사용하여 정책을 만들려면

1 amadmin.dtd를 기반으로 정책 XML 파일을 만듭니다. 이 파일은 다음 디렉토리에 있습니다.

AccessManager-base/SUNWam/dtd

다음은 정책 XML 파일의 한 예입니다. 이 예에는 기본 주제와 조건 값이 모두 포함되어 있습니다. 이러한 값에 대한 정의는 94 페이지 "정책 유형"을 참조하십시오.

```
<Policy name="bigpolicy" referralPolicy="false" active="true" >
<Rule name="rule1">
<ServiceName name="iPlanetAMWebAgentService" />
<ResourceName name="http://thehost.thedomain.com:80/*.html" />
<AttributeValuePair>
<Attribute name="POST" />
<Value>allow</Value>
</AttributeValuePair>
<AttributeValuePair>
<Attribute name="GET" />
<Value>allow</Value>
</AttributeValuePair>
</Rule>
<Subjects name="subjects" description="desccription">
<Subject name="webservicescleint" type="WebServicesClients" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/><Value>CN=sun-unix,
```

```
OU=SUN Java System Access Manager, O=Sun, C=US</Value>
</AttributeValuePair>
</Subject>
<Subject name="amrole" type="IdentityServerRoles" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/><Value>
cn=organization admin role,o=realm1,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="au" type="AuthenticatedUsers" includeType="inclusive">
</Subject>
<Subject name="ldaporganization" type="Organization" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldapuser" type="LDAPUsers" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldaprole" type="LDAPRoles" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>cn=Organization Admin Role,o=realm1,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldapgroup" type="LDAPGroups" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>cn=g1,ou=Groups,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="amidentitysubject" type="AMIdentitySubject" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>id=amAdmin,ou=user,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
</Subjects>
<Conditions name="conditions" description="description">
<Condition name="ldapfilter" type="LDAPFilterCondition">
<AttributeValuePair><Attribute name="ldapFilter"/>
<Value>dept=finance</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelge-nonrealmqualified" type="AuthLevelCondition">
```

```
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>1</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelle-realmqaulfied" type="LEAuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>/:2</Value>
</AttributeValuePair>
</Condition>
<Condition name="sessionproperties" type="SessionPropertyCondition">
<AttributeValuePair><Attribute name="valueCaseInsensitive"/>
<Value>true</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="a"/><Value>10</Value>
<Value>20</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="b"/><Value>15</Value>
<Value>25</Value>
</AttributeValuePair>
</Condition>
<Condition name="activesessiontime" type="SessionCondition">
<AttributeValuePair><Attribute name="TerminateSession"/>
<Value>session_condition_false_value</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="MaxSessionTime"/>
<Value>30</Value>
</AttributeValuePair>
</Condition>
<Condition name="authelevelle-nonrealmqualfied"
        type="LEAuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>2</Value>
</AttributeValuePair>
</Condition>
<Condition name="ipcondition" type="IPCondition">
<AttributeValuePair><Attribute name="DnsName"/>
<Value>*.iplanet.com</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndIp"/>
<Value>145.15.15.15</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartIp"/>
<Value>120.10.10.10</Value>
```

```
</AttributeValuePair>
</Condition>
<Condition name="authchain-realmqualfied"
        type="AuthenticateToServiceCondition">
<AttributeValuePair><Attribute name="AuthenticateToService"/>
<Value>/:ldapService</Value>
</AttributeValuePair>
</Condition>
<Condition name="auth to realm"
     type="AuthenticateToRealmCondition">
<AttributeValuePair><Attribute name="AuthenticateToRealm"/>
<Value>/</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelge-realmqualified"
     type="AuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>/:2</Value>
</AttributeValuePair>
</Condition>
<Condition name="authchain-nonrealmqualfied"
    type="AuthenticateToServiceCondition">
<AttributeValuePair><Attribute name="AuthenticateToService"/>
<Value>ldapService</Value>
</AttributeValuePair>
</Condition>
<Condition name="timecondition" type="SimpleTimeCondition">
<AttributeValuePair><Attribute name="EndTime"/>
<Value>17:00</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartTime"/>
<Value>08:00</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndDate"/>
<Value>2006:07:28</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EnforcementTimeZone"/>
<Value>America/Los_Angeles</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartDay"/>
<Value>mon</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartDate"/>
```

```
<Value>2006:01:02</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndDay"/>
<Value>fri</Value>
</AttributeValuePair>
</Condition>
</Conditions>
<ResponseProviders name="responseproviders"
     description="description">
<ResponseProvider name="idresponseprovidere"
   type="IDRepoResponseProvider">
<AttributeValuePair>
<Attribute name="DynamicAttribute"/>
</AttributeValuePair>
<AttributeValuePair>
<Attribute name="StaticAttribute"/>
<Value>m=10</Value>
<Value>n=30</Value>
</AttributeValuePair>
</ResponseProvider>
</ResponseProviders>
</Policy>
```

2 일단 정책 XML 파일이 만들어지면 다음 명령을 사용하여 로드할 수 있습니다.

```
AccessManager-base/SUNWam/bin/amadmin
--runasdn "uid=amAdmin,ou=People,default_org,
root_suffix"
--password password
--data policy.xml
```

여러 정책을 동시에 추가하려면 각 XML 파일에 정책을 하나씩 사용하는 대신 XML 파일 하나에 여러 정책을 입력합니다. 여러 XML 파일을 사용하여 정책을 빠르게 연속으로 로드하면 내부 정책 색인이 손상되어 일부 정책이 정책 평가에 포함되지 않을 수 있습니다.

`amadmin`을 통해 정책을 만들 경우, 인증 스키마 조건을 만드는 동안 인증 모듈이 영역에 등록되고 영역, LDAP 그룹, LDAP 역할 및 LDAP 사용자 주제를 만드는 동안 해당 LDAP 객체(영역, 그룹, 역할 및 사용자)가 존재하며 `IdentityServerRoles` 주제를 만드는 동안 Access Manager 역할이 존재하고 하위 영역 또는 피어 영역 참조를 만드는 동안 관련 영역이 존재하는지 확인합니다.

`SubrealmReferral`, `PeerRealmReferral`, `Realm` 주제, `IdentityServerRoles` 주제, `LDAPGroups` 주제, `LDAPRoles` 주제 및 `LDAPUsers`의 값 요소 텍스트에서 주제는 전체 DN이어야 합니다.

▼ Access Manager 콘솔을 사용하여 일반 정책을 만들려면

1 정책을 만들려는 영역을 선택합니다.

2 [정책] 탭을 누릅니다.

3 정책 목록에서 [새 정책]을 누릅니다.

4 정책에 대한 이름 및 설명을 추가합니다.

5 정책을 활성화하려면 활성 속성에서 [예]를 선택합니다.

6 이 시점에서 일반 정책에 대한 모든 필드를 정의할 필요는 없습니다. 정책을 만든 다음 나중에 규칙, 주제, 조건 및 응답 공급자를 추가할 수 있습니다. 자세한 내용은 113 페이지 "정책 관리"를 참조하십시오.

7 [확인]을 누릅니다.

▼ Access Manager 콘솔을 사용하여 참조 정책을 만들려면

1 정책을 만들려는 영역을 선택합니다.

2 [정책] 탭에서 [새 참조]를 누릅니다.

3 정책에 대한 이름 및 설명을 추가합니다.

4 정책을 활성화하려면 활성 속성에서 [예]를 선택합니다.

5 이 시점에서 참조 정책에 대한 모든 필드를 정의할 필요는 없습니다. 정책을 만든 다음 나중에 규칙 및 참조를 추가할 수 있습니다. 자세한 내용은 113 페이지 "정책 관리"를 참조하십시오.

6 [확인]을 누릅니다.

피어 영역 및 하위 영역에 대한 정책 만들기

피어 및 하위 영역에 대해 정책을 만들려면 먼저 상위 또는 다른 피어 영역에 참조 정책을 만들어야 합니다. 참조 정책은 해당 규칙 정의에 하위 영역에서 관리될 자원 접두어를 포함해야 합니다. 상위 영역(또는 다른 피어 영역)에 참조 정책이 만들어지면 하위 영역(또는 피어 영역)에 일반 정책을 만들 수 있습니다.

이 예에서 `o=isp`는 상위 영역이고 `o=example.com`은 하위 영역으로, `http://www.example.com`의 자원과 하위 자원을 관리합니다.

▼ 하위 영역에 대한 정책을 만들려면

1 **`o=isp`에 참조 정책을 만듭니다. 참조 정책에 대한 내용은 117 페이지 "참조 정책 수정" 절차를 참조하십시오.**

참조 정책은 `http://www.example.com`을 규칙의 자원으로 정의하고, `example.com`을 갖는 `SubRealmReferral`을 참조 값으로 포함해야 합니다.

2 **`example.com` 하위 영역으로 이동합니다.**

3 **이제 `isp`에서는 `example.com`으로 자원을 참조하며 `http://www.example.com` 자원 또는 `http://www.example.com`으로 시작하는 모든 자원에 대한 일반 정책을 만들 수 있습니다.**

`example.com`에 의해 관리되는 다른 자원에 대한 정책을 정의하려면 `o=isp`에 추가 참조 정책을 만들어야 합니다.

다른 Access Manager 인스턴스로 정책 내보내기

Access Manager를 사용하면 `amadmin` 명령줄 도구를 사용하여 정책을 내보낼 수 있습니다. 이 도구는 기존의 많은 정책을 다른 Access Manager 인스턴스로 옮기거나 일괄 처리 모드로 기존 정책에 대해 변경한 사항을 검사할 경우에 유용합니다. 정책을 내보내려면 `amadmin` 명령줄 유틸리티를 사용하여 지정된 정책을 파일로 내보냅니다. 구문은 다음과 같습니다.

`amamdin - u` *username* `–w` *password* `–ofilename` *output_file.xml* `–t` *policy_data_file.xml*

정책 이름에 와일드카드(*)를 사용하면 모든 문자열과 일치시킬 수 있습니다.

다음은 *policy_data_file.xml* 파일의 한 예입니다.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--
    Copyright (c) 2005 Sun Microsystems, Inc. 모든 권리는 저작권자의 소유입니다.
    본 제품의 사용은 사용권 조항의 적용을 받습니다.
```

```
-->

<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun Java System Access Manager 6.2 Admin CLI DTD//EN"
    "/opt/SUNWam/dtd/amAdmin.dtd"
>>

<!--  CREATE REQUESTS -->

<!-- to export to file use option -ofilename fileName -->

<Requests>

<RealmRequests >
<RealmGetPolicies realm="/" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>p*</Value>
</AttributeValuePair>
</RealmGetPolicies>
</RealmRequests>

<RealmRequests >
<RealmGetPolicies realm="/" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>g10</Value>
<Value>g11</Value>
</AttributeValuePair>
</RealmGetPolicies>

</RealmRequests>
<RealmRequests >
<RealmGetPolicies realm="/realm1" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>*</Value>
</AttributeValuePair>
</RealmGetPolicies>
</RealmRequests>

</Requests>
```

정책은 *Output_file.xml* 파일로 내보내게 되며,이제 이 파일에 포함된 정책 정의를 변경할 수 있습니다. 다른 Access Manager 인스턴스로 정책을 가져오기 전에 `amadmin` 명령

유틸리티에서 사용할 수 있도록 출력 파일을 변경해야 합니다. amadmin 호환 정책 데이터 파일 예를 포함하여 정책을 내보내는 방법에 대한 자세한 지침은 amadmin을 사용하여 정책을 만들려면을 참조하십시오.

정책 관리

일단 일반 또는 참조 정책을 만들어 Access Manager에 추가하면 Access Manager 콘솔을 통해 규칙, 주제, 조건 및 참조를 수정하여 정책을 관리할 수 있습니다.

일반 정책 수정

정책 탭을 통해 액세스 권한을 정의하는 일반 정책을 수정할 수 있습니다. 여러 규칙, 주제, 조건 및 자원 비교기를 정의 및 구성할 수 있습니다. 이 절에서는 이를 수행하는 단계를 나열하고 설명합니다.

▼ 규칙을 일반 정책에 추가하거나 수정하려면

1. **이미 정책을 만든 경우 규칙을 추가하려는 정책의 이름을 누릅니다. 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 일반 정책을 만들려면 "을 참조하십시오.**

2. **[규칙] 메뉴에서 [새로 만들기]를 누릅니다.**

3. **규칙에 대해 다음 기본 서비스 유형 중 하나를 선택합니다. 정책에 대해 사용 가능한 서비스가 많은 경우 목록이 더 클 수도 있습니다.**

검색 서비스	검색 서비스 쿼리에 대한 인증 작업을 정의하고 지정된 자원에 대한 웹 서비스 클라이언트의 프로토콜 호출을 수정합니다.
리버티 개인 프로필 서비스	리버티 개인 프로필 서비스 쿼리에 대한 인증 작업을 정의하고 지정된 자원에 대한 웹 서비스 클라이언트의 프로토콜 호출을 수정합니다.
URL 정책 에이전트	URL 정책 에이전트 서비스에 대한 인증 작업을 정의합니다. HTTP 및 HTTPS URL을 보호하는 정책을 정의하는 데 사용됩니다. Access Manager 정책의 가장 일반적인 사용 예입니다.

4. **[다음]을 누릅니다.**

5 규칙에 대한 이름 및 자원 이름을 입력합니다.

현재 Access Manager 정책 에이전트는 `http://` 및 `https://` 자원만 지원하고 호스트 이름 대신 IP 주소는 지원하지 않습니다.

프로토콜, 호스트, 포트 및 자원 이름에 대해 와일드카드 문자가 지원됩니다. 예를 들면 다음과 같습니다.

```
http*://*:*/*.html
```

URL 정책 에이전트 서비스의 경우 포트 번호를 입력하지 않으면 기본 포트 번호는 `http://`의 경우 80이고 `https://`의 경우 443입니다.

6 규칙의 작업을 선택합니다. 서비스 유형에 따라 다음을 선택할 수 있습니다.

- LOOKUP(검색 서비스)
- UPDATE(검색 서비스)
- MODIFY(리버티 개인 프로필 서비스)
- QUERY(리버티 개인 프로필 서비스)
- GET(URL 정책 에이전트)
- POST(URL 정책 에이전트)

7 작업 값 선택

- 동의 상호 작용 — 자원에 대한 동의를 위해 리버티 상호 작용 프로토콜을 호출합니다. 리버티 개인 프로필 서비스 유형에만 해당합니다.
- 값 상호 작용 — 자원에 대한 값에 대해 리버티 상호 작용 프로토콜을 호출합니다. 리버티 개인 프로필 서비스 유형에만 해당합니다.
- 허용 — 규칙에 정의된 자원과 일치하는 자원에 액세스할 수 있게 합니다.
- 거부 — 규칙에 정의된 자원과 일치하는 자원에 대한 액세스를 거부합니다.

  거부 규칙은 항상 정책의 허용 규칙보다 우선합니다. 예를 들어, 주어진 자원에 대해 두 개의 정책, 즉 액세스를 거부하는 정책과 액세스를 허용하는 정책이 있을 경우 결과적으로 액세스가 거부됩니다(두 정책에 대한 조건이 충족될 경우). 정책 간에 잠재적인 충돌이 일어날 수 있으므로 거부 정책을 사용할 때는 매우 주의해야 합니다. 일반적으로 정책 정의 프로세스는 허용 규칙만 사용해야 하며 거부를 수행하기 위해 적용되는 정책이 없을 경우 기본 거부를 사용해야 합니다.

  명시적 거부 규칙이 사용될 경우 다른 주제(예: 역할 및/또는 그룹 구성원)를 통해 주어진 사용자에 할당되는 정책은 하나 이상의 정책에 액세스를 허용할 경우 자원에 대한 액세스가 거부될 수 있습니다. 예를 들어, 사원 역할에 적용할 수 있는 자원에 대한 거부 정책이 있고 관리자 역할에 적용할 수 있는 동일한 자원에 대한 허용 정책이 있는 경우 사원 역할과 관리자 역할이 모두 할당된 사용자에 대한 정책 결정이 거부됩니다.

  이러한 문제를 해결하는 한 가지 방법은 조건 플러그인을 사용하여 정책을 설계하는 것입니다. 위의 경우에 사원 역할에 인증된 사용자에게 거부 정책을 적용하고 관리자 역할에 인증된 사용자에게 허용 정책을 적용하는 "역할 조건"을 지정하여 두 정책을

차별화할 수 있습니다. 다른 방법은 **인증 수준** 조건을 사용하는 것입니다. 이 조건에서는 관리자 역할이 더 높은 인증 수준으로 인증됩니다.

8 **[마침]을 누릅니다.**

▼ 주제를 일반 정책에 추가하거나 수정하려면

1 **이미 정책을 만든 경우 주제를 추가하려는 정책의 이름을 누릅니다. 아직 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 일반 정책을 만들려면 "을 참조하십시오.**

2 **주제 목록에서 [새로 만들기]를 누릅니다.**

3 **다음 기본 주제 유형 중 하나를 선택합니다. 주제 유형에 대한 설명은 94 페이지 "주제"를 참조하십시오.**

4 **[다음]을 누릅니다.**

5 **주제의 이름을 입력합니다.**

6 **[단독] 필드를 선택하거나 선택 취소합니다.**

이 필드를 선택하지 않을 경우(기본값) 주제의 구성원인 Identity에 정책이 적용됩니다. 이 필드를 선택할 경우 정책은 주제의 구성원이 **아닌** Identity에 적용됩니다.

정책에 여러 주제가 존재하는 경우, 최소한 하나 이상의 주제에서 정책이 주어진 Identity에 적용된다는 것을 나타내면 정책이 Identity에 적용됩니다.

7 **주제에 추가할 Identity를 표시하기 위해 검색을 수행합니다. 이 단계는 인증된 사용자 주제 또는 웹 서비스 클라이언트 주제에는 적용되지 않습니다.**

기본(*) 검색 패턴은 모든 정규화된 항목을 표시합니다.

8 **주제에 대해 추가할 개별 Identity를 선택하거나 [모두 추가]를 눌러 모든 Identity를 한 번에 추가합니다. [추가]를 눌러 Identity를 선택 목록으로 이동합니다. 인증된 사용자 주제에 대해서는 이 단계가 해당되지 않습니다.**

9 **[마침]을 누릅니다.**

10 **정책에서 주제를 제거하려면 해당 주제를 선택하고 [삭제]를 누릅니다. 주제 이름을 눌러 주제 정의를 편집할 수 있습니다.**

▼ 일반 정책에 조건을 추가하려면

1 이미 정책을 만든 경우 조건을 추가하려는 정책의 이름을 누릅니다. 아직 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 일반 정책을 만들려면 "을 참조하십시오.

2 조건 목록에서 [새로 만들기]를 누릅니다.

3 조건 유형을 선택하고 [다음]을 누릅니다.

4 조건 유형의 필드를 정의합니다.

5 [마침]을 누릅니다.

▼ 일반 정책에 응답 공급자를 추가하려면

1 이미 정책을 만든 경우 응답 공급자를 추가하려는 정책의 이름을 누릅니다. 아직 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 일반 정책을 만들려면 "을 참조하십시오.

2 [응답 공급자] 목록에서 [새로 만들기]를 누릅니다.

3 응답 공급자의 이름을 입력합니다.

4 다음 값을 정의합니다.

StaticAttribute	정책에 저장된 `IDResponseProvider`의 인스턴스에서 정의된 속성 값 형식의 정적 속성입니다
DynamicAttribute	여기에서 선택한 응답 속성은 먼저 해당 영역의 정책 구성 서비스에 정의되어야 합니다. 정의된 속성 이름은 구성된 데이터 저장소(IDRepository)에 있는 이름의 하위 집합이어야 합니다. 속성을 정의하는 방법에 대한 자세한 내용은 정책 구성 속성 정의를 참조하십시오. 특정 속성 또는 여러 속성을 선택하려면 Ctrl 키를 누른 상태에서 마우스 왼쪽 버튼을 누릅니다.

5 [마침]을 누릅니다.

6 정책에서 응답 공급자를 제거하려면 해당 주제를 선택하고 [삭제]를 누릅니다. 이름을 눌러 응답 공급자 정의를 편집할 수 있습니다.

참조 정책 수정

참조 정책을 사용하여 정책 정의와 영역 결정을 다른 영역으로 위임할 수 있습니다. 사용자 정의 참조는 정책 대상 지점에서 정책 결정을 가져오는 데 사용됩니다. 참조 정책을 만들면 관련된 규칙, 참조 및 자원 공급자를 추가 또는 수정할 수 있습니다.

▼ 규칙을 참조 정책에 추가하거나 수정하려면

1 **이미 정책을 만든 경우 규칙을 추가하려는 정책의 이름을 누릅니다. 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 참조 정책을 만들려면 "을 참조하십시오.**

2 **[규칙] 메뉴에서 [새로 만들기]를 누릅니다.**

3 **규칙에 대해 다음 기본 서비스 유형 중 하나를 선택합니다. 정책에 대해 사용 가능한 서비스가 많은 경우 목록이 더 클 수도 있습니다.**

검색 서비스	검색 서비스 쿼리에 대한 인증 작업을 정의하고 지정된 자원에 대한 웹 서비스 클라이언트의 프로토콜 호출을 수정합니다.
리버티 개인 프로필 서비스	리버티 개인 프로필 서비스 쿼리에 대한 인증 작업을 정의하고 지정된 자원에 대한 웹 서비스 클라이언트의 프로토콜 호출을 수정합니다.
URL 정책 에이전트	URL 정책 에이전트 서비스에 대한 인증 작업을 정의합니다. HTTP 및 HTTPS URL을 보호하는 정책을 정의하는 데 사용됩니다. Access Manager 정책의 가장 일반적인 사용 예입니다.

4 **[다음]을 누릅니다.**

5 **규칙에 대한 이름 및 자원 이름을 입력합니다.**

현재 Access Manager 정책 에이전트는 `http://` 및 `https://` 자원만 지원하고 호스트 이름 대신 IP 주소는 지원하지 않습니다.

프로토콜, 호스트, 포트 및 자원 이름에 대해 와일드카드 문자가 지원됩니다. 예를 들면 다음과 같습니다.

```
http*://*:*/*.html
```

URL 정책 에이전트 서비스의 경우 포트 번호를 입력하지 않으면 기본 포트 번호는 `http://`의 경우 80이고 `https://`의 경우 443입니다.

주 – 6단계 및 7단계는 참조 정책에 해당되지 않습니다.

6 [마침]을 누릅니다.

▼ 참조를 정책에 추가 또는 수정하려면

1 이미 정책을 만든 경우 응답 공급자를 추가하려는 정책의 이름을 누릅니다. 아직 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 참조 정책을 만들려면 "을 참조하십시오.

2 참조 목록에서 [새로 만들기]를 누릅니다.

3 규칙 필드에서 자원을 정의합니다. 필드는 다음과 같습니다.

참조— 현재 참조 유형을 표시합니다.

이름— 참조 이름을 입력합니다.

자원 이름— 자원 이름을 입력합니다.

필터— [값] 필드에 표시할 영역 이름에 대해 필터를 지정합니다. 기본적으로 이 필드에는 모든 영역 이름이 표시됩니다.

값 — 참조의 영역 이름을 선택합니다.

4 [마침]을 누릅니다.

정책에서 참조를 제거하려면 참조를 선택하고 [삭제]를 누릅니다.

참조 이름 옆에 있는 [편집] 링크를 눌러 모든 참조 정의를 편집할 수 있습니다.

▼ 참조 정책에 응답 공급자를 추가하려면

1 이미 정책을 만든 경우 응답 공급자를 추가하려는 정책의 이름을 누릅니다. 아직 정책을 만들지 않은 경우 110 페이지 "Access Manager 콘솔을 사용하여 참조 정책을 만들려면 "을 참조하십시오.

2 [응답 공급자] 목록에서 [새로 만들기]를 누릅니다.

3 응답 공급자의 이름을 입력합니다.

4 다음 값을 정의합니다.

StaticAttribute	정책에 저장된 `IDResponseProvider`의 인스턴스에서 정의된 속성 값 형식의 정적 속성입니다
DynamicAttribute	여기에서 선택한 응답 속성은 먼저 해당 영역의 정책 구성 서비스에 정의되어야 합니다. 정의된 속성 이름은 구성된 데이터 저장소(IDRepository)에 있는 이름의 하위 집합이어야 합니다. 속성을 정의하는 방법에 대한 자세한 내용은 정책 구성 속성

정의를 참조하십시오. 특정 속성 또는 여러 속성을 선택하려면 Ctrl 키를 누른 상태에서 마우스 왼쪽 버튼을 누릅니다.

5 **[마침]을 누릅니다.**

6 **정책에서 응답 공급자를 제거하려면 해당 주제를 선택하고 [삭제]를 누릅니다. 이름을 눌러 응답 공급자 정의를 편집할 수 있습니다.**

정책 구성 서비스

정책 구성 서비스는 Access Manager 콘솔을 통해 각 조직에 대한 정책 관련 속성을 구성하는 데 사용됩니다. Access Manager 정책 프레임워크에 사용되는 자원 이름 구현 및 Directory Server 데이터 저장소를 정의할 수도 있습니다. 정책 구성 서비스에 지정된 Directory Server는 LDAP 사용자, LDAP 그룹, LDAP 역할 및 조직 정책 주제의 구성원 평가에 사용됩니다.

주제 결과 수명

정책 평가 성능을 향상시키려면 정책 구성 서비스의 주제 결과 수명 속성에 정의된 시간 동안 구성원 평가를 캐시에 저장합니다. 이렇게 캐시에 저장된 구성원 결정은 주제 결과 수명 속성에 정의된 시간이 다 지날 때까지 사용됩니다. 이후의 구성원 평가는 디렉토리 내 사용자의 현재 상태를 반영하는 데 사용됩니다.

동적 속성

목록에 표시되고 정책 응답 공급자 동적 속성을 정의하기 위해 선택된 허용된 동적 속성 이름입니다. 정의된 이름은 데이터 저장소에 정의된 속성 이름과 같아야 합니다.

amldapuser 정의

`amldapuser`는 기본으로 사용되는 설치 중에 정책 구성 서비스에서 지정된 Directory Server에 생성된 사용자입니다. 이는 필요에 따라 관리자 또는 해당 영역의 정책 관리자에 의해 변경될 수 있습니다.

정책 구성 서비스 추가

영역이 생성될 때 정책 구성 서비스 속성이 자동으로 이 영역에 대해 설정됩니다. 하지만 필요한 경우 속성을 수정할 수 있습니다.

자원 기반 인증

일부 조직에서는 사용자가 액세스를 시도하는 자원에 따라 특정 모듈에 대해 인증하는 고급 인증 시나리오를 요구합니다. 자원 기반 인증은 사용자가 기본 인증 모듈이 아니라 자원을 보호하는 특정 인증 모듈에 인증해야 하는 Access Manager의 기능입니다. 이 기능은 처음으로 사용자를 인증하는 경우에만 사용할 수 있습니다.

주 – 이 기능은 88 페이지 "세션 업그레이드"에 설명된 자원 기반 인증과는 다른 기능입니다. 해당 특정 기능에는 제한 사항이 없습니다.

제한 사항

자원 기반 인증에는 다음과 같은 제한 사항이 포함됩니다.

- 자원에 적용할 수 있는 정책에 여러 인증 모듈이 있는 경우 해당 시스템에서 임의로 하나의 인증 모듈을 선택합니다.
- 이 정책에 대해 정의될 수 있는 조건은 수준과 방법뿐입니다.
- 이 기능은 서로 다른 DNS 도메인 사이에서는 사용할 수 없습니다.

▼ 자원 기반 인증을 구성하려면

Access Manager와 정책 에이전트가 모두 설치되면 자원 기반 인증을 구성할 수 있습니다. 자원 기반 인증을 구성하려면 Access Manager가 게이트웨이 서블릿을 가리켜야 합니다.

1 **`AMAgent.properties`를 엽니다.**

`AMAgent.properties`는 Solaris 환경의 경우 `/etc/opt//SUNWam/agents/config/`에 있습니다.

2 **다음 행을 주석으로 처리합니다.**

```
#com.sun.am.policy.am.loginURL = http://Access
Manager_server_host.domain_name:port/amserver/UI/Login.
```

3 **다음 행을 추가합니다.**

```
com.sun.am.policy.am.loginURL =
http://AccessManager_host.domain_name:port/amserver/gateway
```

주 – 게이트웨이 서블릿은 Policy Evaluation API를 사용하여 개발하며 자원 기반 인증을 수행하는 사용자 정의 기법을 작성하는 데 사용됩니다. **Sun Java System Access Manager 7.1 Developer's Guide**의 3 장, "Using the Policy APIs"를 참조하십시오.

---

4 에이전트를 다시 시작합니다.

◆ ◆ ◆ 6 장

6 주제 관리

주제 인터페이스를 사용하면 영역에서 기본 아이디를 관리할 수 있습니다. 주제 인터페이스에서 만든 모든 아이디는 Access Manager 아이디 주제 유형으로 만든 정책의 주제 정의에서 사용할 수 있습니다.

만들고 수정할 수 있는 아이디는 다음과 같습니다.

사용자

사용자는 개인의 아이디를 나타냅니다. 그룹의 사용자를 만들고 제거할 수 있으며 역할 및 그룹에 사용자를 추가하거나 제거할 수 있습니다. 또한 서비스를 사용자에게 할당할 수도 있습니다.

▼ 사용자를 만들거나 수정하려면

1. **[사용자] 탭을 누릅니다.**
2. **[새로 만들기]를 누릅니다.**
3. **다음 필드에 데이터를 입력합니다.**

   사용자 아이디. 이 필드에는 사용자가 Access Manager에 로그인할 때 사용하는 이름을 입력합니다. 이 등록 정보는 DN이 아닌 값일 수 있습니다.

   이름. 이 필드는 사용자의 이름을 가집니다.

성. 이 필드는 사용자의 성을 가집니다.

전체 이름. 이 필드는 사용자의 전체 이름을 가집니다.

비밀번호.이 필드에는 사용자 아이디 필드에 지정된 이름의 비밀번호를 입력합니다.

비밀번호(확인). 비밀번호를 확인합니다.

사용자 상태. 이 옵션은 사용자에게 Access Manager를 통한 인증이 허용되었는지 여부를 나타냅니다.

4 **[만들기]를 누릅니다.**

5 **사용자가 생성되면 사용자의 이름을 눌러 사용자 정보를 편집할 수 있습니다. 사용자 속성에 대한 자세한 내용은 사용자 속성을 참조하십시오. 다음을 수행할 수 있습니다.**

- 123 페이지 “사용자를 만들거나 수정하려면”
- 124 페이지 “역할 및 그룹에 사용자를 추가하려면”
- 124 페이지 “Identity에 서비스를 추가하려면”

▼ 역할 및 그룹에 사용자를 추가하려면

1 **수정할 사용자의 이름을 누릅니다.**

2 **[역할] 또는 [그룹]을 선택합니다. 이미 사용자에게 할당된 역할과 그룹만 표시됩니다.**

3 **[사용 가능] 목록에서 역할 또는 그룹을 선택하고 [추가]를 누릅니다.**

4 **[선택] 목록에 역할 또는 그룹이 표시되면 [저장]을 누릅니다.**

▼ Identity에 서비스를 추가하려면

1 **서비스를 추가할 아이디를 선택합니다.**

2 **[서비스] 탭을 누릅니다.**

3 **[추가]를 누릅니다.**

4 **선택한 아이디 유형에 따라 다음과 같은 서비스 목록이 표시됩니다.**

- 인증 구성
- 검색 서비스
- 리버티 개인 프로필 서비스
- 세션

- 사용자

5 **추가할 서비스를 선택하고 [다음]을 누릅니다.**

6 **서비스에 대한 속성을 편집합니다. 서비스에 대한 설명을 참조하려면 4단계에서 서비스 이름을 누릅니다.**

7 **[마침]을 누릅니다.**

에이전트 프로필

Access Manager 정책 에이전트는 웹 서버 및 웹 프록시 서버의 내용을 권한이 없는 침입으로부터 보호합니다. 정책 에이전트는 관리자가 구성한 정책에 기초하여 서비스 및 웹 자원에 대한 액세스를 제어합니다.

에이전트 객체는 정책 에이전트 프로필을 정의하고 Access Manager 자원을 보호하고 있는 특정 에이전트에 대한 인증 및 기타 프로필 정보를 Access Manager에서 저장할 수 있게 합니다. 관리자는 Access Manager 콘솔을 통해 에이전트 프로필을 확인, 작성, 수정 및 삭제할 수 있습니다.

에이전트 객체 만들기 페이지는 에이전트가 Access Manager에 대한 인증에 사용할 UID/비밀번호를 정의할 수 있는 위치입니다. 같은 Access Manager를 사용하는 웹 컨테이너 설정이 여러 개 있는 경우 다른 에이전트에 대해 여러 아이디를 활성화하고 이들을 Access Manager와 별개로 활성화 및 비활성화하는 옵션을 제공합니다. 또한 각 컴퓨터에서 `AMAgent.properties`를 편집하지 않고 에이전트에 대한 일부 기본 설정 값을 중앙에서 관리할 수 있습니다.

▼ 에이전트를 만들거나 수정하려면

1 **[에이전트] 탭을 누릅니다.**

2 **[새로 만들기]를 누릅니다.**

3 **다음과 같은 필드에 값을 입력합니다.**

이름. 에이전트의 이름 또는 아이디를 입력합니다. 에이전트는 Access Manager에 로그인할 때 이 이름을 사용합니다. 멀티바이트 이름은 사용할 수 없습니다.

비밀번호. 에이전트의 비밀번호를 입력합니다. 이 비밀번호는 LDAP 인증 도중에 에이전트가 사용하는 비밀번호와는 달라야 합니다.

비밀번호 확인. 비밀번호를 확인합니다.

장치 상태. 에이전트의 장치 상태를 입력합니다. 활성으로 설정된 경우 에이전트는 Access Manager에 대해 인증되어 Access Manager와 통신할 수 있습니다. 비활성으로 설정된 경우 에이전트는 Access Manager에 대해 인증될 수 없습니다.

4 **[만들기]를 누릅니다.**

5 **에이전트를 만든 후에는 다음과 같은 필드를 추가로 편집할 수 있습니다.**

설명. 에이전트에 대한 간단한 설명을 입력합니다. 예를 들어, 에이전트 인스턴스 이름이나 에이전트가 보호하는 응용 프로그램의 이름을 입력할 수 있습니다.

에이전트 키 값. 키/값 쌍을 사용하여 에이전트 등록 정보를 설정합니다. Access Manager는 이 등록 정보를 사용하여 사용자의 자격 증명에 대한 에이전트 요청을 받습니다. 현재는 하나의 등록 정보만 유효하며 다른 등록 정보는 모두 무시됩니다. 다음 형식을 사용합니다.

agentRootURL=protocol://*hostname:port/*

입력 항목이 정확해야 하며 agentRootURL은 대소문자를 구분합니다.

protocol	사용되는 프로토콜(HTTP 또는 HTTPS)을 나타냅니다.
hostname	에이전트가 상주하는시스템의 호스트 이름을 나타냅니다. 또한 이 시스템은 해당 에이전트에서 보호하는 자원도 호스팅합니다.
port	에이전트가 설치된 포트 번호를 나타냅니다. 에이전트는 이 포트에서 트래픽을 수신하고 호스트의 자원에 대한 액세스 요청을 모두 가로챕니다.

쿠키 하이재킹을 차단하도록 Access Manager 구성

쿠키 하이재킹은 신뢰할 수 없는 응용 프로그램을 사용하는 해커(강탈자)가 쿠키에 무단 액세스하려고 시도하는 상황을 말합니다. 하이재킹되는 쿠키가 세션 쿠키인 경우 시스템 구성 방식에 따라 쿠키 하이재킹으로 인해 보호되는 웹 자원에 대한 무단 액세스 위협이 잠재적으로 높아질 수 있습니다.

Sun에서는 "Precautions Against Session-Cookie Hijacking in an Access Management Deployment"라는 기술 자료를 통해 세션 쿠키 하이재킹과 관련된 특정 보안 위협에 대해 취할 수 있는 사전 예방 조치를 설명하고 있습니다. 다음 문서를 참조하십시오.

Technical Note: Precautions Against Cookie Hijacking in an Access Manager Deployment

필터링된 역할

필터링된 역할은 LDAP 필터를 사용하여 만든 동적 역할입니다. 모든 사용자가 필터를 통해 걸러져 역할을 만들 때 해당 역할에 할당됩니다. 필터는 항목의 임의 속성 값 쌍(예: `ca=user*`)을 찾아 해당 속성을 포함하는 사용자를 역할에 자동으로 할당합니다.

▼ 필터링된 역할을 만들려면

1. **[이동] 창에서 역할을 만들 조직으로 이동합니다.**
2. **[새로 만들기]를 누릅니다.**
3. **필터링된 역할의 이름을 입력합니다.**
4. **검색 조건에 대한 정보를 입력합니다.**

   예:

   ```
   (&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
   ```

   필터를 비워두면 기본적으로 다음 역할이 만들어집니다.

   ```
   (objectclass = inetorgperson)
   ```

5. **[만들기]를 눌러 필터 조건을 바탕으로 검색을 시작합니다. 필터 조건에서 정의한 아이디가 자동으로 역할에 할당됩니다.**
6. **필터링된 역할이 만들어지면 역할 이름을 눌러 해당 역할에 속한 사용자를 확인합니다. 또한 [서비스] 탭을 눌러 역할에 서비스를 추가할 수도 있습니다.**

역할

역할의 구성원은 역할을 소유하는 LDAP 항목입니다. 역할의 기준 자체는 속성과 함께 LDAP 항목으로 정의되며 항목의 고유 이름(DN) 속성으로 식별됩니다. 역할을 만들면 서비스와 사용자를 직접 추가할 수 있습니다.

▼ 역할을 만들거나 수정하려면

1. **[역할] 탭을 누릅니다.**
2. **역할 목록에서 [새로 만들기]를 누릅니다.**

3 **역할의 이름을 입력합니다.**

4 **[만들기]를 누릅니다.**

▼ 역할 또는 그룹에 사용자를 추가하려면

1 **사용자를 추가할 역할 또는 그룹의 이름을 누릅니다.**

2 **[사용자] 탭을 누릅니다.**

3 **[사용 가능] 목록에서 추가할 사용자를 선택한 다음 [추가]를 누릅니다.**

4 **[선택] 목록에 사용자가 표시되면 [저장]을 누릅니다.**

그룹

그룹은 공통적인 기능, 특징 또는 관심을 가지는 사용자의 집합을 나타냅니다. 일반적으로 이 그룹에는 연관된 권한이 없습니다. 그룹은 두 가지 수준 즉, 조직 내에서와 다른 관리 대상 그룹 내에서 존재할 수 있습니다.

▼ 그룹을 만들거나 수정하려면

1 **[그룹] 탭을 누릅니다.**

2 **그룹 목록에서 [새로 만들기]를 누릅니다.**

3 **그룹의 이름을 입력합니다.**

4 **[만들기]를 누릅니다.**
그룹을 만든 다음에는 그룹 이름과 [사용자] 탭을 차례로 눌러 그룹에 사용자를 추가할 수 있습니다.

2

디렉토리 관리 및 기본 서비스

Sun Java System Access Manager 7.1 관리 설명서의 제2부입니다. 디렉토리 관리 장에서는 Access Manager를 레거시 모드로 배포할 때 디렉토리 객체를 관리하는 방법에 대해 설명합니다. 다른 장에서는 Access Manager의 일부 기본 서비스를 구성하고 사용하는 방법에 대해 설명합니다. 다음 내용으로 구성되어 있습니다.

◆ ◆ ◆ 7 장

7 디렉토리 관리

디렉토리 관리 탭은 Access Manager를 레거시 모드로 설치할 경우에만 표시됩니다. 디렉토리 관리 기능은 Sun Java System Directory Server를 사용하는 Access Manager 배포를 위한 Identity 관리 솔루션을 제공합니다.

레거시 모드 설치 옵션에 대한 자세한 내용은 **Sun Java Enterprise System 5 Installation Guide for UNIX**를 참조하십시오.

디렉토리 객체 관리

디렉토리 관리 탭에는 Directory Server 객체를 보고 관리하는 데 필요한 모든 구성 요소가 포함되어 있습니다. 이 절에서는 객체 유형과 객체 유형을 구성하는 방법에 대해 설명합니다. Access Manager 콘솔 또는 명령줄 인터페이스를 사용하여 사용자, 역할, 그룹, 조직, 하위 조직 및 컨테이너 객체를 정의, 수정 또는 삭제할 수 있습니다. 콘솔에는 다양한 권한으로 디렉토리 객체를 생성하고 관리하는 기본 관리자가 있습니다. (역할을 기반으로 추가 관리자를 만들 수 있습니다.) 관리자는 Access Manager 설치 시 Directory Server 내에 정의됩니다. 다음은 사용자가 관리할 수 있는 Directory Server 객체입니다.

조직

조직은 기업에서 부서와 자원을 관리하는 데 사용되는 최상위 수준의 계층 구조를 나타냅니다. 설치 시 Access Manager는 Access Manager 엔터프라이즈 구성을 관리하기

위해 최상위 수준 조직(설치하는 동안 정의됨)을 동적으로 만듭니다. 설치 후에 추가 조직을 만들어 별도 엔터프라이즈를 관리할 수 있습니다. 생성되는 모든 조직은 최상위 조직 아래에 놓입니다.

▼ 조직 만들기

1 [디렉토리] 관리 탭을 누릅니다.

2 [조직] 목록에서 [새로 만들기]를 누릅니다.

3 필드에 대한 값을 입력합니다. [이름] 필드만 필수입니다. 필드는 다음과 같습니다.

이름 — 조직의 이름 값을 입력합니다.

도메인 이름 — 조직의 완전한 DNS(Domain Name System) 이름을 입력합니다(있을 경우).

조직 상태 — `active` 또는 `inactive` 상태를 선택합니다. 기본값은 `active`입니다. 이 값은 조직의 수명 동안 [등록 정보] 아이콘을 선택하여 언제든지 변경할 수 있습니다. `inactive`를 선택하면 조직에 로그인할 때 사용자 액세스가 사용 불가능하게 됩니다.

조직 별칭 — 이 필드는 URL 로그인에서 별칭을 사용하여 인증할 수 있도록 조직에 대한 별칭 이름을 정의합니다. 예를 들어, 조직 이름이 `exampleorg`이고 `123` 및 `abc`를 별칭으로 정의하는 경우 다음 URL 중 하나를 사용하여 조직에 로그인할 수 있습니다.

```
http://machine.example.com/amserver/UI/Login?org=exampleorg

http://machine.example.com/amserver/UI/Login?org=abc

http://machine.example.com/amserver/UI/Login?org=123
```

조직 별칭 이름은 조직 전체에서 고유해야 합니다. 고유 속성 목록을 사용하여 고유성을 강제로 적용할 수 있습니다.

DNS 별칭 이름 — 조직의 DNS 이름에 대해 별칭 이름을 추가할 수 있습니다. 이 속성은 "실제" 도메인 별칭(임의의 문자열은 허용 안 됨)만 수락합니다. 예를 들어, DNS 이름이 `example.com`이고 `example1.com` 및 `example2.com`을 `exampleorg` 조직에 대한 별칭으로 정의하는 경우 다음 URL 중 하나를 사용하여 조직에 로그인할 수 있습니다.

```
http://machine.example.com/amserver/UI/

Login?org=exampleorg

http://machine.example1.com/amserver/
```

```
UI/Login?org=exampleorg

http://machine.example2.com/amserver/

UI/Login?org=exampleorg
```

고유 속성 목록 조직의 사용자에 대한 고유 속성 이름 목록을 추가할 수 있습니다. 예를 들어, 전자 메일 주소를 지정하는 고유한 속성 이름을 추가할 경우 동일한 전자 메일 주소를 가지는 두 명의 사용자를 만들 수 없습니다. 또한, 이 필드에서는 쉼표로 구분된 목록을 허용합니다. 목록에 있는 속성 이름 중 하나가 고유성을 정의합니다. 예를 들어, 필드에 다음과 같은 속성 이름 목록이 있고

```
PreferredDomain, AssociatedDomain
```

`PreferredDomain`이 특정 사용자에 대한 `http://www.example.com`으로 정의되는 경우 전체 쉼표로 구분된 목록이 해당 URL에 대한 고유성으로 정의됩니다. 고유 속성 목록에 이름 지정 속성인 'ou'를 추가하면 기본 그룹, 사용자 컨테이너의 속성에 고유성이 강제 적용되지 않습니다. (ou=Groups,ou=People)

모든 하위 조직에 고유성이 강제 적용됩니다.

주 – 고유 속성은 영역 모드에서 설정할 수 없습니다. 또한 레거시 모드의 경우 7.0 또는 7.1 기반 콘솔에서 설정할 수도 없습니다. 고유 속성 목록을 만들려면 6.3 기반 콘솔에 로그인해야 합니다. 자세한 내용은 21 페이지 “레거시 모드 6.3 콘솔”을 참조하십시오.

---

4 **[확인]을 누릅니다.**

새 조직이 조직 목록에 표시됩니다. 조직을 만드는 동안 정의한 등록 정보를 편집하려면 편집할 조직의 이름을 누르고 등록 정보를 변경한 다음 저장을 누릅니다.

▼ 조직 삭제

1 **삭제할 조직의 이름 옆에 있는 확인란을 선택합니다.**

2 **[삭제]를 누릅니다.**

주 – 삭제를 수행할 때 경고 메시지가 나타나지 않습니다. 조직 내의 모든 항목이 삭제되고 실행 취소를 수행할 수 없습니다.

정책에 조직 추가

Access Manager 객체는 정책의 주제 정의를 통해 정책에 추가됩니다. 정책을 작성하거나 수정할 때 조직, 역할, 그룹 및 사용자를 주제로 정의할 수 있습니다. 주제가 정의되고 나면 정책이 객체에 적용됩니다. 자세한 내용은 113 페이지 "정책 관리"를 참조하십시오.

컨테이너

객체 클래스와 속성의 차이로 인해 조직 항목을 사용할 수 없는 경우 **컨테이너** 항목을 사용합니다. Access Manager 컨테이너 항목과 Access Manager 조직 항목이 LDAP 객체 클래스 `organizationalUnit` 및 `organization`과 반드시 같을 필요가 없다는 것이 중요합니다. 추상적인 identity 항목입니다. 이상적인 경우라면 컨테이너 항목 대신 조직 항목이 사용됩니다.

주 – 컨테이너 표시는 선택 사항입니다. 컨테이너를 보려면 [구성] > [콘솔 등록 정보] 아래의 관리 서비스에서 [컨테이너 표시]를 선택해야 합니다.

▼ 컨테이너 만들기

1. **새 컨테이너가 생성될 조직의 위치 링크 또는 컨테이너를 선택합니다.**
2. **[컨테이너] 탭을 누릅니다.**
3. **[컨테이너] 목록에서 [새로 만들기]를 누릅니다.**
4. **만들려는 컨테이너의 이름을 입력합니다.**
5. **[확인]을 누릅니다.**

▼ 컨테이너 삭제

1. **[컨테이너] 탭을 누릅니다.**
2. **삭제할 컨테이너의 이름 옆에 있는 확인란을 선택합니다.**
3. **[삭제]를 누릅니다.**

주 – 컨테이너를 삭제하면 해당 컨테이너에 존재하는 모든 객체가 삭제됩니다. 여기에는 모든 객체와 하위 컨테이너가 포함됩니다.

그룹 컨테이너

그룹 컨테이너는 그룹을 관리하는 데 사용됩니다. 그룹 컨테이너는 그룹과 다른 그룹 컨테이너만 포함할 수 있습니다. 그룹 컨테이너 그룹은 모든 관리 대상 그룹에 대한 부모 항목으로 동적으로 할당됩니다. 원하는 경우 추가 그룹 컨테이너를 추가할 수 있습니다.

주 – 그룹 컨테이너의 표시는 선택 사항입니다. 그룹 컨테이너를 보려면 [구성] > [콘솔 등록 정보]의 관리 서비스에서 [그룹 컨테이너 사용 가능]을 선택해야 합니다.

▼ 그룹 컨테이너 만들기

1. **새 그룹 컨테이너를 포함할 조직의 위치 링크 또는 그룹 컨테이너를 선택합니다.**
2. **[그룹 컨테이너] 탭을 선택합니다.**
3. **[그룹 컨테이너] 목록에서 [새로 만들기]를 누릅니다.**
4. **이름 필드에 값을 입력하고 [확인]을 누릅니다. [그룹 컨테이너] 목록에 새 그룹 컨테이너가 표시됩니다.**

▼ 그룹 컨테이너 삭제

1. **삭제할 그룹 컨테이너가 포함된 조직으로 이동합니다.**
2. **[그룹 컨테이너] 탭을 선택합니다.**
3. **삭제할 그룹 컨테이너 옆의 확인란을 선택합니다.**
4. **[삭제]를 누릅니다.**

그룹

그룹은 공통된 기능, 특징 또는 관심사를 가진 사용자 모음을 나타냅니다. 일반적으로 이 그룹에는 연관된 권한이 없습니다. 그룹은 두 가지 수준 즉, 조직 내에서와 다른 관리 대상 그룹 내에서 존재할 수 있습니다. 다른 그룹 내에서 존재하는 그룹을 **하위 그룹**이라고 부릅니다. 하위 그룹은 상위 그룹 내에서 "물리적으로" 존재하는 하위 노드입니다.

Access Manager는 또한 단일 그룹에 포함된 기존 그룹의 "표현"인 **중첩 그룹**을 지원합니다. 하위 그룹과 달리 중첩 그룹은 DIT 내의 어디에나 존재할 수 있습니다. 중첩 그룹은 다수의 사용자에 대한 액세스 권한을 신속하게 설정할 수 있게 합니다.

정적 그룹과 동적 그룹의 두 가지 유형의 그룹을 만들 수 있습니다. 사용자는 정적 그룹에만 수동으로 추가할 수 있습니다. 동적 그룹은 필터를 통해 사용자의 추가를 제어합니다. 중첩 또는 하위 그룹은 두 유형 모두에 추가될 수 있습니다.

정적 그룹

정적 그룹은 사용자가 지정한 관리 대상 그룹 유형을 기준으로 만들어집니다. `groupOfNames` 또는 `groupOfUniqueNames` 객체 클래스를 사용하여 그룹 항목에 그룹 구성원을 추가합니다.

주 – 기본적으로 관리 대상 그룹 유형은 동적입니다. 관리 서비스 구성에서 이 기본값을 변경할 수 있습니다.

동적 그룹

LDAP 필터를 사용하여 동적 그룹을 만듭니다. 모든 항목이 필터를 통해 걸러져 그룹에 동적으로 할당됩니다. 필터는 항목에서 속성을 검색하여 속성이 포함된 항목을 반환합니다. 예를 들어, 건물 번호를 기반으로 그룹을 만들 경우 필터를 사용하여 해당 건물 번호 속성을 포함하는 모든 사용자 목록을 반환할 수 있습니다.

주 – 참조 무결성 플러그 인을 사용하려면 Access Manager를 Directory Server와 함께 구성해야 합니다. 참조 무결성 플러그 인은 사용 가능하게 될 경우 삭제 또는 이름 바꾸기 작업 직후에 지정된 속성에 대한 무결성 업데이트를 수행합니다. 따라서 관련된 항목 간의 관계가 데이터베이스 전체에서 유지됩니다. 데이터베이스 색인은 Directory Server에서 검색 성능을 향상시킵니다. 플러그인 사용에 대한 자세한 내용은 Access Manager 6 Migration Guide를 참조하십시오.

▼ 정적 그룹을 만들려면

1. **새 그룹을 만들 조직, 그룹 또는 그룹 컨테이너로 이동합니다.**
2. **[그룹] 목록에서 [새 정적]을 누릅니다.**
3. **[이름] 필드에 그룹의 이름을 입력합니다. [다음]을 누릅니다.**
4. **[사용자가 이 그룹에 가입할 수 있음] 속성을 선택하여 사용자가 그룹에 직접 가입할 수 있게 합니다.**
5. **[확인]을 누릅니다.**
   그룹이 만들어지면 그룹 이름을 선택하고 [일반] 탭을 눌러서 [사용자가 이 그룹에 가입할 수 있음] 속성을 편집할 수 있습니다.

▼ 정적 그룹에서 구성원 추가 또는 제거

1 [그룹] 목록에서 구성원을 추가할 그룹을 선택합니다.

2 [작업 선택] 메뉴에서 수행할 작업을 선택합니다. 수행할 수 있는 작업은 다음과 같습니다.

새 사용자 — 이 작업은 새 사용자를 만들며 사용자 정보를 저장할 때 사용자를 그룹에 추가합니다.

사용자 추가 — 이 작업은 기존 사용자를 그룹에 추가합니다. 이 작업을 선택하면 추가할 사용자를 지정할 검색 기준을 만들 수 있습니다. 기준을 만드는 데 사용되는 필드는 ANY 또는 ALL 연산자를 사용합니다. ALL은 지정된 모든 필드에 해당하는 사용자를 반환합니다. ANY는 지정된 필드 중 하나 이상에 해당하는 사용자를 반환합니다. 필드를 비워두면 해당 특정 속성과 일치하는 가능한 모든 항목을 반환합니다.

검색 기준을 작성하고 나서 [다음]을 누릅니다. 반환된 사용자 목록에서 추가할 사용자를 선택하고 [마침]을 누릅니다.

그룹 추가 — 이 작업은 중첩 그룹을 현재 그룹에 추가합니다. 이 작업을 선택할 경우 검색 범위와 그룹 이름("*" 와일드카드 사용 가능)을 포함하는 검색 조건을 만들며 사용자가 그룹에 직접 가입할 수 있는지 여부를 지정할 수 있습니다. 정보를 입력하고 [다음]을 누릅니다. 반환된 그룹 목록에서 추가할 그룹을 선택하고 [마침]을 누릅니다.

구성원 제거 — 이 작업은 그룹에서 구성원(사용자 및 그룹 포함)을 제거하지만 삭제하지는 않습니다. 제거할 구성원을 선택하고 [작업 선택] 메뉴에서 [구성원 제거]를 선택합니다.

구성원 삭제 — 이 작업은 선택한 구성원을 영구적으로 삭제합니다. 삭제할 구성원을 선택한 다음 [구성원 삭제]를 선택합니다.

▼ 동적 그룹을 만들려면

1 새 그룹을 만들 조직 또는 그룹으로 이동합니다.

2 [그룹] 탭을 누릅니다.

3 [새 동적]을 누릅니다.

4 [이름] 필드에 그룹의 이름을 입력합니다.

5 LDAP 검색 필터를 생성합니다.

기본적으로 Access Manager는 기본 검색 필터 인터페이스를 표시합니다. 필터를 생성하는 데 사용되는 기본 필드는 ANY 또는 ALL 연산자를 사용합니다. ALL은 지정된 모든

필드에 해당하는 사용자를 반환합니다. ANY는 지정된 필드 중 하나 이상에 해당하는 사용자를 반환합니다. 필드를 비워두면 해당 특정 속성과 일치하는 가능한 모든 항목을 반환합니다.

6 **[확인]을 누르면 검색 조건과 일치하는 모든 사용자가 자동으로 그룹에 추가됩니다.**

▼ 동적 그룹에서 구성원을 추가 또는 제거하려면

1 **[그룹] 목록에서 구성원을 추가할 그룹의 이름을 누릅니다.**

2 **[작업 선택] 메뉴에서 수행할 작업을 선택합니다. 수행할 수 있는 작업은 다음과 같습니다.**

그룹 추가 — 이 작업은 중첩 그룹을 현재 그룹에 추가합니다. 이 작업을 선택할 경우 검색 범위와 그룹 이름("*" 와일드카드 사용 가능)을 포함하는 검색 조건을 만들며 사용자가 그룹에 직접 가입할 수 있는지 여부를 지정할 수 있습니다. 정보를 입력하고 [다음]을 누릅니다. 반환된 그룹 목록에서 추가할 그룹을 선택하고 [마침]을 누릅니다.

구성원 제거 — 이 작업은 그룹에서 구성원(그룹 포함)을 제거하지만 삭제하지는 않습니다. 제거할 구성원을 선택한 다음 [구성원 제거]를 선택합니다.

구성원 삭제 — 이 작업은 선택한 구성원을 영구적으로 삭제합니다. 삭제할 구성원을 선택한 다음 [구성원 삭제]를 선택합니다.

정책에 그룹 추가

Access Manager 객체는 정책의 주제 정의를 통해 정책에 추가됩니다. 정책을 작성하거나 수정할 때 정책의 주제 페이지에서 조직, 역할, 그룹 및 사용자를 주제로 정의할 수 있습니다. 주제가 정의되고 나면 정책이 객체에 적용됩니다. 자세한 내용은 113 페이지 "정책 관리"를 참조하십시오.

사용자 컨테이너

사용자 컨테이너는 조직 내에서 사용자가 만들어질 때 모든 사용자가 할당되는 기본 LDAP 조직 구성 단위입니다. 사용자 컨테이너는 조직 수준에서 표시되거나 사용자 컨테이너 수준에서 하위 사용자 컨테이너로 표시될 수 있습니다. 사용자 컨테이너는 다른 사용자 컨테이너와 사용자만 포함할 수 있습니다. 원하는 경우 추가 사용자 컨테이너를 조직에 추가할 수 있습니다.

주 – 사용자 컨테이너의 표시는 선택 사항입니다. 사용자 컨테이너를 보려면 관리 서비스에서 [사용자 컨테이너 사용 가능]을 선택해야 합니다.

▼ 사용자 컨테이너 만들기

1 새 사용자 컨테이너를 만들려는 조직이나 사용자 컨테이너로 이동합니다.

2 [사용자 컨테이너] 목록에서 [새로 만들기]를 누릅니다.

3 만들려는 사용자 컨테이너의 이름을 입력합니다.

4 [확인]을 누릅니다.

▼ 사용자 컨테이너를 삭제하려면

1 삭제할 사용자 컨테이너를 포함하는 조직이나 사용자 컨테이너로 이동합니다.

2 삭제할 사용자 컨테이너의 이름 옆에 있는 확인란을 선택합니다.

3 [삭제]를 누릅니다.

주 – 사용자 컨테이너를 삭제하면 해당 사용자 컨테이너에 존재하는 모든 객체가 삭제됩니다. 여기에는 모든 사용자와 하위 사용자 컨테이너가 포함됩니다.

사용자

사용자는 개인의 아이디를 나타냅니다. Access Manager Identity 관리 모듈을 통해 사용자를 조직, 컨테이너 및 그룹에서 만들고 삭제할 수 있으며 역할 및/또는 그룹에서 추가 또는 제거할 수 있습니다. 또한 서비스를 사용자에게 할당할 수도 있습니다.

주 – `amadmin`과 동일한 사용자 아이디를 사용하여 하위 조직의 사용자를 만들 경우 `amadmin`에 대한 로그인이 실패하게 됩니다. 이런 문제가 발생할 경우 관리자는 Directory Server 콘솔을 통해 사용자의 아이디를 변경해야 합니다. 이렇게 하면 관리자는 기본 조직에 로그인할 수 있습니다. 또한 인증 서비스에서 사용자 검색을 시작할 DN을 사용자 컨테이너 DN으로 설정하여 로그인 프로세스 도중 고유한 일치가 반환되도록 할 수 있습니다.

▼ 사용자 만들기

1 사용자를 만들 조직, 컨테이너 또는 사용자 컨테이너로 이동합니다.

2 해당 사용자 탭을 누릅니다.

3 **사용자 목록에서 [새로 만들기]를 누릅니다.**

4 **다음 값의 데이터를 입력합니다.**

사용자 아이디	이 필드에는 사용자가 Access Manager에 로그인할 때 사용하는 이름을 입력합니다. 이 등록 정보는 DN이 아닌 값일 수 있습니다.
이름	이 필드에는 사용자의 이름을 입력합니다. 이름 값 및 성 값은 현재 로그인된 사용자 필드에서 사용자를 식별합니다. 이 값은 필수 값이 아닙니다.
성	이 필드에는 사용자의 성을 입력합니다. 이름 값 및 성 값은 사용자를 식별합니다.
전체 이름	이 필드에는 사용자의 전체 이름을 입력합니다.
비밀번호	이 필드에는 사용자 아이디 필드에 지정된 이름의 비밀번호를 입력합니다.
비밀번호(확인)	비밀번호를 확인합니다.
사용자 상태	이 옵션은 사용자에게 Access Manager를 통한 인증이 허용되었는지 여부를 나타냅니다. 활성 사용자만 인증될 수 있습니다. 기본값은 **활성**입니다.

5 **[확인]을 누릅니다.**

▼ 사용자 프로필을 편집하려면

관리 역할이 할당되지 않은 사용자가 Access Manager에 대해 인증될 경우 기본 보기는 해당 사용자 프로필입니다. 또한 적절한 권한이 있는 관리자가 사용자 프로필을 편집할 수 있습니다. 이 보기에서 사용자는 개인 프로필 특정의 속성 값을 수정할 수 있습니다. 사용자 프로필 보기에 표시되는 속성은 확장할 수 있습니다. 객체 및 Identity에 대한 사용자 정의 속성 추가에 대한 자세한 내용은 Access Manager Developer's Guide를 참조하십시오.

1 **프로필을 편집할 사용자를 선택합니다. 기본적으로 일반 보기가 표시됩니다.**

2 **다음 필드를 편집합니다.**

이름	이 필드에는 사용자의 이름을 입력합니다.
성	이 필드에는 사용자의 성을 입력합니다.
전체 이름	이 필드에는 사용자의 전체 이름을 입력합니다.
비밀번호	사용자 비밀번호를 추가 및 확인하려면 편집 링크를 누릅니다.
전자 메일 주소	이 필드에는 사용자의 전자 메일 주소를 입력합니다.

사원 번호 — 이 필드에는 사용자의 사원 번호를 입력합니다.

전화 번호 — 이 필드에는 사용자의 전화 번호를 입력합니다.

집 주소 — 이 필드에는 사용자의 집 주소를 입력합니다.

사용자 상태 — 이 옵션은 사용자에게 Access Manager를 통한 인증이 허용되었는지 여부를 나타냅니다. 활성 사용자만 Access Manager를 통해 인증될 수 있습니다. 기본값은 활성입니다. 다음 중 하나를 풀다운 메뉴에서 선택할 수 있습니다.

- 활성 — Access Manager를 통해 사용자를 인증할 수 있습니다.
- 비활성 — Access Manager를 통해 사용자를 인증할 수 없지만 사용자 프로필은 디렉토리에 저장된 채로 남습니다.

주 – 사용자 상태를 비활성으로 변경하는 것은 Access Manager를 통한 인증에만 영향을 줍니다. Directory Server는 *nsAccountLock* 속성을 사용하여 사용자 계정 상태를 결정합니다. Access Manager 인증에 대해 비활성화된 사용자 계정은 여전히 Access Manager가 필요하지 않은 작업을 수행할 수 있습니다. 단순히 Access Manager 인증에 대해서가 아니라 디렉토리에서 사용자 계정을 비활성화하려면 *nsAccountLock* 값을 false로 설정합니다. 사이트의 위임된 관리자가 정기적으로 사용자를 비활성화할 경우 *nsAccountLock* 속성을 Access Manager 사용자 프로필 페이지에 추가하는 방법을 고려하십시오. 자세한 내용은 **Sun Java System Access Manager 7.1 Developer's Guide**를 참조하십시오.

계정 만료 날짜 — 이 속성이 있으면 현재 날짜와 시간이 지정된 계정 만료일을 지난 경우 인증 서비스는 로그인을 허용하지 않습니다. 이 속성의 형식은 *mm/dd/yyyy hh:mm*입니다.

사용자 인증 구성 — 이 속성은 사용자의 인증 체인을 설정합니다.

사용자 별칭 목록 — 이 필드는 사용자에게 적용될 수 있는 별칭 목록을 정의합니다. 이 속성에 구성된 별칭을 사용하려면 LDAP 서비스의 사용자 항목 검색 속성 필드에 `iplanet-am-user-alias-list` 속성을 추가하여 LDAP 서비스를 수정해야 합니다.

기본 로켈 — 이 필드는 사용자의 로켈을 지정합니다.

성공 URL	이 속성은 인증 성공 시 사용자가 리디렉션되는 URL을 지정합니다.
실패 URL	이 속성은 인증 실패 시 사용자가 리디렉션되는 URL을 지정합니다.
비밀번호 재설정 옵션	이 옵션은 잊어버린 비밀번호를 복구하는 데 사용되는 비밀번호 분실 페이지에서 질문을 선택하는 데 사용됩니다.
사용자 검색 자원 오퍼링	사용자에 대한 사용자 검색 서비스의 자원 오퍼링을 설정합니다.
MSIDSN 번호	MSISDN 인증을 사용 중인 경우 사용자의 MSISDN 번호를 정의합니다.

▼ 역할 및 그룹에 사용자 추가

1 **[사용자] 탭을 누릅니다.**

2 **수정할 사용자의 이름을 누릅니다.**

3 **[역할] 또는 [그룹] 탭을 선택합니다.**

4 **사용자를 추가할 역할이나 그룹을 선택하고 [추가]를 누릅니다.**

5 **[저장]을 누릅니다.**

주 – 역할이나 그룹에서 사용자를 제거하려면 역할 또는 그룹을 선택하고 [제거]를 누른 다음 [저장]을 누릅니다.

정책에 사용자 추가

Access Manager 객체는 정책의 주제 정의를 통해 정책에 추가됩니다. 정책을 작성하거나 수정할 때 정책의 주제 페이지에서 조직, 역할, 그룹 및 사용자를 주제로 정의할 수 있습니다. 주제가 정의되고 나면 정책이 객체에 적용됩니다. 자세한 내용은 113 페이지 "정책 관리"를 참조하십시오.

역할

역할은 **그룹**의 개념과 비슷한 Directory Server 항목 체계입니다. 그룹이 구성원을 가지므로 역할도 구성원을 가집니다. 역할의 구성원은 역할을 소유하는 LDAP 항목입니다. 역할의 기준 자체는 속성과 함께 LDAP 항목으로 정의되며 항목의 고유

이름(DN) 속성에 의해 식별됩니다. Directory Server에는 여러 가지 유형의 역할이 있지만 Access Manager는 그 중에서(관리 대상 역할)만 관리할 수 있습니다.

주 – 다른 Directory Server 역할 유형은 Access Manager 콘솔에서 관리할 수는 없지만 디렉토리를 배포하는 데 사용할 수 있습니다. 정책의 주제 정의에 다른 Directory Server 유형을 사용할 수 있습니다. 정책 주제에 대한 자세한 내용은 105 페이지 "정책 만들기"를 참조하십시오.

사용자는 하나 이상의 역할을 소유할 수 있습니다. 예를 들어, 세션 서비스 및 비밀번호 재설정 서비스의 속성을 갖는 계약자 역할을 만들 수 있습니다. 새 계약직 직원이 회사에 합류하면 관리자는 계약자 항목에 개별 속성을 설정하는 대신 이 역할을 할당할 수 있습니다. 계약자가 엔지니어링 부서에서 일하며, 엔지니어링 직원이 사용할 수 있는 서비스와 액세스 권한을 요구하는 경우, 관리자는 계약자를 계약자 역할 외에 엔지니어링 역할에도 지정할 수 있습니다.

Access Manager는 역할을 사용하여 액세스 제어 명령을 적용합니다. 처음 설치되면 Access Manager는 관리자 사용 권한을 정의하는 액세스 제어 명령(ACI)을 구성합니다. 그런 다음 이러한 ACI는 사용자에게 할당될 때 사용자의 액세스 권한을 정의하는 역할(예: 조직 관리자 역할 및 조직 도움말 데스크 관리자 역할)에 지정됩니다.

사용자는 관리 서비스에서 사용자 프로필 페이지에 역할 표시 속성이 사용 가능하게 된 경우에만 할당된 역할을 볼 수 있습니다.

주 – 참조 무결성 플러그 인을 사용하려면 Access Manager를 Directory Server와 함께 구성해야 합니다. 참조 무결성 플러그 인은 사용 가능하게 될 경우 삭제 또는 이름 바꾸기 작업 직후에 지정된 속성에 대한 무결성 업데이트를 수행합니다. 따라서 관련된 항목 간의 관계가 데이터베이스 전체에서 유지됩니다. 데이터베이스 색인은 Directory Server에서 검색 성능을 향상시킵니다.

다음과 같은 두 가지 역할 유형이 있습니다.

- 정적 — 정적 역할은 역할을 만들 때 사용자 추가 없이 만듭니다. 역할이 만들어진 다음 해당 역할에 특정 사용자를 추가할 수 있습니다. 따라서 주어진 역할에 사용자를 추가할 때 더 많은 것을 제어할 수 있습니다.
- 동적 – 동적 역할은 LDAP 필터를 사용하여 만듭니다. 모든 사용자가 필터를 통해 걸러져 역할 작성 시 역할에 할당됩니다. 필터는 항목의 임의 속성 값 쌍(예: `ca=user*`)을 찾아 해당 속성을 포함하는 사용자를 역할에 자동으로 할당합니다.

▼ 정적 역할 만들기

1 역할을 만들 조직으로 이동합니다.

2 [역할] 탭을 누릅니다.

기본 역할 세트는 조직이 구성될 때 만들어지며 역할 목록에 표시됩니다. 기본 역할은 다음과 같습니다.

컨테이너 도움말 데스크 관리자.컨테이너 도움말 데스크 관리자 역할은 조직 구성 단위의 모든 항목에 대한 읽기 권한과 이 컨테이너 단위에 한하여 사용자 항목의 `userPassword` 속성에 대한 쓰기 권한을 가집니다.

조직 도움말 데스크 관리자.조직의 도움말 데스크 관리자는 조직의 모든 항목에 대한 읽기 권한과 `userPassword` 속성에 대한 쓰기 권한을 가집니다.

주 – 하위 조직을 만들 때 관리 역할이 부모 조직이 아닌 하위 조직에서 만들어진다는 점에 주의하십시오.

컨테이너 관리자.컨테이너 관리자 역할은 LDAP 조직 구성 단위의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다. Access Manager에서 LDAP 조직 구성 단위를 흔히 컨테이너라고 부릅니다.

조직 정책 관리자.조직 정책 관리자는 모든 정책에 대한 읽기 및 쓰기 권한을 가지며 해당 조직 내의 모든 정책을 작성, 할당, 수정 및 삭제할 수 있습니다.

사용자 관리자.기본적으로 새로 만든 조직의 모든 사용자 항목은 해당 조직에 속한 구성원입니다. 사용자 관리자는 조직의 모든 사용자 항목에 대한 읽기 및 쓰기 권한을 가집니다. 이 역할은 역할 및 그룹 DN을 포함하는 속성에 대한 읽기 및 쓰기 권한을 갖지 않으므로 역할 또는 그룹의 속성을 수정하거나 역할 또는 그룹에서 사용자를 제거할 수 없다는 점에 주의하십시오.

주 – Access Manager에서 다른 컨테이너를 구성하여 사용자 항목, 그룹 항목 또는 다른 컨테이너를 포함할 수 있습니다. 조직이 이미 구성된 후에 만든 컨테이너에 관리자 역할을 할당하면 컨테이너 관리자 역할 또는 컨테이너 도움말 데스크 관리자 기본값이 사용됩니다.

그룹 관리자.그룹을 만들 때 생성되는 그룹 관리자는 특정 그룹의 모든 구성원에 대한 읽기 및 쓰기 권한을 가지며, 새 사용자를 만들고 관리하는 그룹에 사용자를 할당하고 만든 사용자를 삭제하는 등의 작업을 수행할 수 있습니다.

그룹이 만들어지면 해당 그룹을 관리하는 데 필요한 권한과 함께 그룹 관리자 역할이 자동으로 생성됩니다. 이 역할은 그룹 구성원에 자동으로 할당되지 않습니다. 따라서 그룹 작성자나 그룹 관리자 역할에 대한 액세스 권한을 가진 누군가가 이 역할을 할당해야 합니다.

최상위 수준 관리자.최상위 수준 관리자는 최상위 수준 조직의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다. 즉, 이 최상위 수준 관리자 역할은 Access Manager 응용 프로그램 내의 모든 구성 기본에 대한 권한을 가집니다.

조직 관리자.조직 관리자는 조직의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다. 조직이 만들어지면 해당 조직을 관리하는 데 필요한 권한과 함께 조직 관리자 역할이 자동으로 생성됩니다.

3 **[새 정적] 버튼을 누릅니다.**

4 **역할의 이름을 입력합니다.**

5 **역할에 대한 설명을 입력합니다.**

6 **[유형] 메뉴에서 역할 유형을 선택합니다.**

역할은 관리 역할 또는 서비스 역할이 될 수 있습니다. 역할 유형은 Access Manager 콘솔에서 사용자를 시작할 위치를 파악하기 위해 사용됩니다. 관리 역할은 역할 소유자가 관리 권한을 갖고 있다는 것을 콘솔에 알리고 서비스 역할은 역할 소유자가 최종 사용자라는 것을 콘솔에 알립니다.

7 **[액세스 권한] 메뉴에서 역할에 적용할 기본 사용 권한 집합을 선택합니다. 이러한 사용 권한은 조직 내의 항목에 대한 액세스를 제공합니다. 기본 사용 권한은 특별한 순서 없이 표시됩니다. 다음과 같은 권한이 있습니다.**

사용 권한 없음	역할에 사용 권한이 설정되지 않습니다.
조직 관리자	조직 관리자는 구성된 조직의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다.
조직 도움말 데스크 관리자	조직 도움말 데스크 관리자는 구성된 조직의 모든 항목에 대한 읽기 권한과 `userPassword` 속성에 대한 쓰기 권한을 가집니다.
조직 정책 관리자	조직 정책 관리자는 조직의 모든 정책에 대한 읽기 및 쓰기 권한을 가집니다. 조직 정책 관리자는 피어 조직에 대한 참조 정책을 만들 수 없습니다. 일반적으로 서비스 역할에는 사용 권한 없음 ACI가 할당되고 관리 역할에는 임의의 기본 ACI가 할당됩니다.

▼ 정적 역할에 사용자 추가

1 **사용자를 추가할 역할의 이름을 누릅니다.**

2 **[구성원] 목록의 [작업 선택] 메뉴에서 [사용자 추가]를 선택합니다.**

3 **검색 조건에 대한 정보를 입력합니다. 하나 이상의 표시된 필드에 기초하여 사용자를 검색할 수 있습니다. 이러한 필드는 다음과 같습니다.**

일치	필터에 포함할 필드를 선택할 수 있습니다. ALL은 지정된 모든 필드에 해당하는 사용자를 반환합니다. ANY는 지정된 필드 중 하나 이상에 해당하는 사용자를 반환합니다.
이름	이름을 기준으로 사용자를 검색합니다.
사용자 아이디	사용자 아이디를 기준으로 사용자를 검색합니다.
성	성을 기준으로 사용자를 검색합니다.
전체 이름	성명을 기준으로 사용자를 검색합니다.
사용자 상태	상태(활성 또는 비활성)를 기준으로 사용자를 검색합니다.

4 **[다음]을 눌러 검색을 시작합니다. 검색 결과가 표시됩니다.**

5 **아이디 옆에 있는 확인란을 선택하여 반환된 이름에서 사용자를 선택합니다.**

6 **[마침]을 누릅니다.**
사용자가 이제 역할에 할당됩니다.

▼ 동적 역할을 만들려면

1 **역할을 만들 조직으로 이동합니다.**

2 **[역할] 탭을 누릅니다.**
기본 역할 세트는 조직이 구성될 때 만들어지며 역할 목록에 표시됩니다. 기본 역할은 다음과 같습니다.

컨테이너 도움말 데스크 관리자. 컨테이너 도움말 데스크 관리자 역할은 조직 구성 단위의 모든 항목에 대한 읽기 권한과 이 컨테이너 단위에 한하여 사용자 항목의 userPassword 속성에 대한 쓰기 권한을 가집니다.

조직 도움말 데스크 관리자. 조직의 도움말 데스크 관리자는 조직의 모든 항목에 대한 읽기 권한과 userPassword 속성에 대한 쓰기 권한을 가집니다.

주 – 하위 조직을 만들 때 관리 역할이 부모 조직이 아닌 하위 조직에서 만들어진다는 점에 주의하십시오.

컨테이너 관리자. 컨테이너 관리자 역할은 LDAP 조직 구성 단위의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다. Access Manager에서 LDAP 조직 구성 단위를 흔히 컨테이너라고 부릅니다.

조직 정책 관리자.조직 정책 관리자는 모든 정책에 대한 읽기 및 쓰기 권한을 가지며 해당 조직 내의 모든 정책을 작성, 할당, 수정 및 삭제할 수 있습니다.

사용자 관리자.기본적으로 새로 만든 조직의 모든 사용자 항목은 해당 조직에 속한 구성원입니다. 사용자 관리자는 조직의 모든 사용자 항목에 대한 읽기 및 쓰기 권한을 가집니다. 이 역할은 역할 및 그룹 DN을 포함하는 속성에 대한 읽기 및 쓰기 권한을 갖지 않으므로 역할 또는 그룹의 속성을 수정하거나 역할 또는 그룹에서 사용자를 제거할 수 없다는 점에 주의하십시오.

주 – Access Manager에서 다른 컨테이너를 구성하여 사용자 항목, 그룹 항목 또는 다른 컨테이너를 포함할 수 있습니다. 조직이 이미 구성된 후에 만든 컨테이너에 관리자 역할을 할당하면 컨테이너 관리자 역할 또는 컨테이너 도움말 데스크 관리자 기본값이 사용됩니다.

그룹 관리자.그룹을 만들 때 생성되는 그룹 관리자는 특정 그룹의 모든 구성원에 대한 읽기 및 쓰기 권한을 가지며, 새 사용자를 만들고 관리하는 그룹에 사용자를 할당하고 만든 사용자를 삭제하는 등의 작업을 수행할 수 있습니다.

그룹이 만들어지면 해당 그룹을 관리하는 데 필요한 권한과 함께 그룹 관리자 역할이 자동으로 생성됩니다. 이 역할은 그룹 구성원에 자동으로 할당되지 않습니다. 따라서 그룹 작성자나 그룹 관리자 역할에 대한 액세스 권한을 가진 누군가가 이 역할을 할당해야 합니다.

최상위 수준 관리자.최상위 수준 관리자는 최상위 수준 조직의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다. 즉, 이 최상위 수준 관리자 역할은 Access Manager 응용 프로그램 내의 모든 구성 기본에 대한 권한을 가집니다.

조직 관리자.조직 관리자는 조직의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다. 조직이 만들어지면 해당 조직을 관리하는 데 필요한 권한과 함께 조직 관리자 역할이 자동으로 생성됩니다.

3 **[새 동적] 버튼을 누릅니다.**

4 **역할의 이름을 입력합니다.**

5 **역할에 대한 설명을 입력합니다.**

6 **[유형] 메뉴에서 역할 유형을 선택합니다.**

역할은 관리 역할 또는 서비스 역할이 될 수 있습니다. 역할 유형은 Access Manager 콘솔에서 사용자를 시작할 위치를 파악하기 위해 사용됩니다. 관리 역할은 역할 소유자가 관리 권한을 갖고 있다는 것을 콘솔에 알리고 서비스 역할은 역할 소유자가 최종 사용자라는 것을 콘솔에 알립니다.

7 [액세스 권한] 메뉴에서 역할에 적용할 기본 사용 권한 집합을 선택합니다. 이러한 사용 권한은 조직 내의 항목에 대한 액세스를 제공합니다. 기본 사용 권한은 특별한 순서 없이 표시됩니다. 다음과 같은 권한이 있습니다.

사용 권한 없음	역할에 사용 권한이 설정되지 않습니다.
조직 관리자	조직 관리자는 구성된 조직의 모든 항목에 대한 읽기 및 쓰기 권한을 가집니다.
조직 도움말 데스크 관리자	조직 도움말 데스크 관리자는 구성된 조직의 모든 항목에 대한 읽기 권한과 userPassword 속성에 대한 쓰기 권한을 가집니다.
조직 정책 관리자	조직 정책 관리자는 조직의 모든 정책에 대한 읽기 및 쓰기 권한을 가집니다. 조직 정책 관리자는 피어 조직에 대한 참조 정책을 만들 수 없습니다. 일반적으로 서비스 역할에는 사용 권한 없음 ACI가 할당되고 관리 역할에는 임의의 기본 ACI가 할당됩니다.

8 검색 조건에 대한 정보를 입력합니다. 필드는 다음과 같습니다.

일치	필터에 포함할 임의의 필드에 대한 연산자를 포함할 수 있습니다. ALL은 지정된 모든 필드에 해당하는 사용자를 반환합니다. ANY는 지정된 필드 중 하나 이상에 해당하는 사용자를 반환합니다.
이름	이름을 기준으로 사용자를 검색합니다.
사용자 아이디	사용자 아이디를 기준으로 사용자를 검색합니다.
성	성을 기준으로 사용자를 검색합니다.
전체 이름	성명을 기준으로 사용자를 검색합니다.
사용자 상태	상태(활성 또는 비활성)를 기준으로 사용자를 검색합니다.

9 [확인]을 눌러 필터 조건에 기초한 검색을 시작합니다. 필터 조건에서 정의한 사용자가 자동으로 역할에 할당됩니다.

▼ 역할에서 사용자 제거

1 수정할 역할을 포함하는 조직으로 이동합니다.

Identity 관리 모듈의 [보기] 메뉴에서 [조직]을 선택하고 [역할] 탭을 선택합니다.

2 수정할 역할을 선택합니다.

3 [보기] 메뉴에서 [사용자]를 선택합니다.

4 **제거할 각 사용자 옆에 있는 확인란을 선택합니다.**

5 **[작업 선택] 메뉴에서 [사용자 제거]를 누릅니다.**
사용자가 이제 역할에서 제거됩니다.

정책에 역할 추가

Access Manager 객체는 정책의 주제 정의를 통해 정책에 추가됩니다. 정책을 작성하거나 수정할 때 정책의 주제 페이지에서 조직, 역할, 그룹 및 사용자를 주제로 정의할 수 있습니다. 주제가 정의되고 나면 정책이 객체에 적용됩니다. 자세한 내용은 113 페이지 “정책 관리”를 참조하십시오.

◆ ◆ ◆ 8 장

8 현재 세션

이 장에서는 Access Manager의 세션 관리 기능에 대해 설명합니다. 세션 관리 모듈은 사용자 세션 정보를 확인하고 사용자 세션을 관리하기 위한 솔루션을 제공합니다. 세션 관리 모듈을 사용하면 다양한 세션 시간을 추적할 수 있으며 관리자가 세션을 종료할 수 있습니다. 시스템 관리자는 플랫폼 서버 목록에 있는 로드 밸런서 서버를 무시해야 합니다.

현재 세션 인터페이스

현재 세션 모듈 인터페이스를 사용하면 적절한 사용 권한이 있는 관리자가 현재 Access Manager에 로그인한 사용자의 세션 정보를 볼 수 있습니다.

세션 관리

세션 관리 프레임은 현재 관리되고 있는 Access Manager의 이름을 표시합니다.

세션 정보

세션 정보 창은 현재 Access Manager에 로그인한 모든 사용자 및 각 사용자의 세션 시간을 표시합니다. 표시 필드는 다음과 같습니다.

사용자 아이디.현재 로그인한 사용자의 사용자 아이디를 표시합니다.

남은 시간.사용자가 다시 인증하기 전에 해당 세션에 대해 남은 시간(분)을 표시합니다.

최대 세션 시간.세션이 만료되고 액세스 권한을 다시 얻기 위해 다시 인증하기 전까지 사용자가 로그인할 수 있는 최대 시간(분)을 표시합니다.

유휴 시간. 사용자가 유휴 상태인 시간(분)을 표시합니다.

최대 유휴 시간.사용자가 다시 인증하기 전까지 유휴 상태로 있을 수 있는 최대 시간(분)을 표시합니다.

시간 제한은 관리자가 세션 관리 서비스에서 정의합니다.

[사용자 아이디] 필드에 문자열을 입력하고 [필터]를 눌러 특정 사용자 세션이나 사용자 세션의 특정 범위를 표시할 수 있습니다. 와일드카드를 사용할 수 있습니다.

[새로 고침] 버튼을 누르면 사용자 세션 표시가 업데이트됩니다.

세션 종료

적절한 사용 권한을 가진 관리자가 언제든지 사용자 세션을 종료할 수 있습니다.

▼ 세션을 종료하려면

1. **종료하려는 사용자 세션을 선택합니다.**
2. **[종료]를 누릅니다.**

◆ ◆ ◆ 9 장

9 비밀번호 재설정 서비스

Access Manager는 사용자가 Access Manager로 보호되는 지정된 서비스 또는 응용 프로그램에 액세스하기 위한 비밀번호를 재설정할 수 있게 해주는 비밀번호 재설정 서비스를 제공합니다. 비밀번호 재설정 서비스 속성은 최상위 관리자가 정의하고, 비밀번호 질문 형태로 사용자 검증 자격 증명을 제어하며 새로운 또는 기존 비밀번호 알림 기법을 제어합니다. 그리고 잘못된 사용자 검증에 대한 가능한 잠금 간격을 설정합니다.

이번 장은 다음 절로 구성됩니다.

비밀번호 재설정 서비스 등록

사용자가 소속된 영역에 대해서는 비밀번호 재설정 서비스를 등록할 필요가 없습니다. 사용자가 위치한 조직에 비밀번호 재설정 서비스가 없는 경우 서비스 구성에서 해당 서비스에 대해 정의된 값을 상속합니다.

▼ 다른 영역의 사용자에 대해 비밀번호 재설정을 등록하려면

1 **사용자에 대한 비밀번호를 등록하려는 영역으로 이동합니다.**

2 **영역 이름을 누르고 [서비스] 탭을 누릅니다.**
서비스가 아직 영역에 추가되지 않은 경우 [추가] 버튼을 누릅니다.

3 **[비밀번호 재설정]을 선택하고 [다음]을 누릅니다.**
비밀번호 재설정 서비스 속성이 표시됩니다. 속성 정의는 온라인 도움말을 참조하십시오.

4 **[마침]을 누릅니다.**

비밀번호 재설정 서비스 구성

비밀번호 재설정 서비스가 등록되어 있는 경우 관리자 권한이 있는 사용자가 서비스를 구성해야 합니다.

▼ 서비스를 구성하려면

1 **비밀번호 재설정 서비스를 등록할 영역을 선택합니다.**

2 **[서비스] 탭을 누릅니다.**

3 **서비스 목록에서 [비밀번호 재설정]을 누릅니다.**

4 **비밀번호 재설정 속성이 표시되고 사용자는 이 속성을 사용하여 비밀번호 재설정 서비스에 대한 요구 사항을 정의할 수 있습니다. 비밀번호 재설정 서비스가 사용 가능(기본값)한지 확인합니다. 최소한 다음 속성을 정의해야 합니다.**

- 사용자 검증
  - 비밀 문제
  - 바인드 DN
  - 바인드 비밀번호

    바인드 DN 속성은 비밀번호 재설정 권한이 있는 사용자(예: 도움말 데스크 관리자)를 포함해야 합니다. Directory Server의 제한 때문에 바인드 DN이 `cn=Directory Manager`인 경우에는 비밀번호 재설정이 실행되지 않습니다.

    나머지 속성은 선택 사항입니다. 서비스 속성에 대한 설명은 온라인 도움말을 참조하십시오.

주 – Access Manager는 임의의 비밀번호 생성을 위한 비밀번호 재설정 웹 응용 프로그램을 자동으로 설치합니다. 그러나 비밀번호 생성 및 비밀번호 알림을 위한 사용자 플러그 인 클래스를 작성할 수 있습니다. 이러한 플러그 인 클래스에 대해서는 다음 위치에 있는 다음 Readme.html 파일을 참조하십시오.

PasswordGenerator:

```
AccessManager-base/SUNWam/samples/console/PasswordGenerator
```

NotifyPassword:

```
AccessManager-base/SUNWam/samples/console/NotifyPassword
```

5 **사용자가 고유 개인 문제를 직접 정의해야 하는 경우 개인 문제 사용 가능 속성을 선택합니다. 속성을 정의한 다음 [저장]을 누릅니다.**

▼ 비밀 문제를 현지화하려면

현지화된 Access Manager 버전을 실행하는 경우 사용자의 로켈에 해당하는 문자 집합으로 비밀 문제를 표시하려면 다음을 수행하십시오.

1 **비밀번호 재설정 서비스에서 [비밀 문제] 속성 아래의 [현재 값] 목록에 비밀 문제 키를 추가합니다. 예를 들면** `favorite-color`**와 같습니다.**

2 `amPasswordReset.properties` **파일에 키 값을 표시할 문제와 함께 해당 키를 추가합니다. 예를 들면 다음과 같습니다.**

`favorite-color=`**가장 좋아하는 색상은?**

3 `/opt/SUNWam/locale`**의** `AMPasswordReset_`*locale*`.properties` **파일에 현지화된 질문과 함께 해당 키를 추가합니다. 사용자가 비밀번호를 변경하려는 경우 현지화된 문제가 표시됩니다.**

비밀번호 재설정 잠금

비밀번호 재설정 서비스에는 사용자가 비밀 문제에 올바르게 응답하기 위해 시도할 수 있는 횟수를 제한하는 잠금 기능이 포함됩니다. 잠금 기능은 비밀번호 재설정 서비스 속성을 통해 구성됩니다. 서비스 속성에 대한 설명은 온라인 도움말을 참조하십시오. 비밀번호 재설정은 메모리 잠금과 물리적 잠금이라는 두 가지 유형의 잠금을 지원합니다.

메모리 잠금

이 잠금은 임시 잠금이며 비밀번호 재설정 실패 잠금 기간 속성의 값이 0보다 크고 비밀번호 재설정 실패 잠금 사용 가능 속성이 활성화된 경우에만 유효합니다. 이 잠금은 사용자가 비밀번호 재설정 웹 응용 프로그램을 통해 비밀번호를 재설정하지 못하게 합니다. 잠금은 비밀번호 재설정 실패 잠금 기간에 지정된 기간 동안 지속되거나 서버가 다시 시작될 때까지 지속됩니다. 서비스 속성에 대한 설명은 온라인 도움말을 참조하십시오.

물리적 잠금

보다 영구적인 잠금입니다. 비밀번호 재설정 실패 잠금 횟수 속성 값을 0으로 설정하고 비밀번호 재설정 실패 잠금 사용 가능 속성을 활성화하면 사용자가 비밀번호 질문에 잘못 대답할 경우 해당 사용자의 계정 상태가 비활성 상태로 변경됩니다. 서비스 속성에 대한 설명은 온라인 도움말을 참조하십시오.

최종 사용자에 대한 비밀번호 재설정

다음 절에서는 비밀번호 재설정 서비스에 대한 사용자 경험을 설명합니다.

비밀번호 재설정 사용자 정의

비밀번호 재설정 서비스가 사용 가능하고 관리자가 속성을 정의한 경우 사용자는 Access Manager 콘솔에 로그인하여 비밀 문제를 사용자 정의할 수 있습니다.

▼ 비밀번호 재설정을 사용자 정의하려면

1. **사용자가 아이디와 비밀번호를 제공하여 Access Manager 콘솔에 로그인하면 성공적으로 인증됩니다.**

2. **사용자 프로필 페이지에서 비밀번호 재설정 옵션을 선택합니다. 사용 가능한 문제 응답 화면이 표시됩니다.**

3. **관리자가 해당 서비스에 대해 정의한 사용 가능한 문제가 표시됩니다. 예를 들면 다음과 같습니다.**

   - 애완동물 이름은?
     - 가장 좋아하는 TV 쇼는?
     - 어머니의 성함은?
     - 자주 가는 식당은?

4. **비밀번호 질문을 선택합니다. 비밀번호 질문은 관리자가 영역에 대해 정의한 최대 문제 수 이하로 선택할 수 있습니다(최대 양은 비밀번호 재설정 서비스를 통해 정의됨). 그런**

다음 선택한 문제에 대한 대답을 입력합니다. 이러한 문제와 대답은 사용자의 비밀번호 재설정을 위한 기초가 됩니다(다음 절 참조). 관리자가 개인 문제 사용 가능 속성을 선택한 경우 사용자가 고유한 비밀 문제를 입력하고 대답을 제공할 수 있는 텍스트 필드가 제공됩니다.

5 **[저장]을 누릅니다.**

잊어버린 비밀번호 재설정

사용자가 비밀번호를 잊어버린 경우 Access Manager는 비밀번호 재설정 웹 응용 프로그램을 사용하여 새 비밀번호를 임의로 생성하여 사용자에게 새 비밀번호를 알려 줍니다. 다음은 일반적인 잊어버린 비밀번호 시나리오입니다.

▼ 잊어버린 비밀번호를 재설정하려면

1 **관리자가 지정해 준 URL에서 비밀번호 재설정 웹 응용 프로그램에 로그인합니다. 예를 들면 다음과 같습니다.**

`http://hostname:port/ampassword`(기본 영역용)

또는

`http://hostname: port/`*deploy_uri* `/UI/PWResetUserValidation?realm=realmname`(여기서 realmname은 영역 이름임)

주 – 비밀번호 재설정 서비스가 상위 영역에 대해서는 사용 가능으로 설정되어 있지 않고 하위 영역에 대해서는 사용 가능으로 설정되어 있는 경우 사용자가 서비스에 액세스하려면 다음 구문을 사용해야 합니다.

`http://hostname: port/`*deploy_uri*`/UI/PWResetUserValidation?realm=realmname`

---

2 **사용자 아이디를 입력합니다.**

3 **비밀번호 재설정 서비스에서 정의하고 사용자 정의 과정에서 사용자가 선택한 개인 문제가 표시됩니다. 사용자 프로필 페이지에 로그인하지 않고 개인 문제를 사용자 정의한 경우 비밀번호가 생성되지 않습니다.**

사용자가 문제에 올바르게 대답하면 새 비밀번호를 생성하여 전자 메일로 사용자에게 알려 줍니다. 문제에 올바르게 대답했는지 여부에 관계 없이 사용자에게 시도 알림을 보냅니다. 새 비밀번호와 시도 알림을 받으려면 사용자 프로필 페이지에 전자 메일 주소를 입력해야 합니다.

비밀번호 정책

비밀번호 정책은 지정된 디렉토리에서 비밀번호가 사용되는 방법을 제어하는 일련의 규칙이며, 대개 Directory Server 콘솔을 통해 Directory Server에 정의됩니다. 보안 비밀번호 정책은 다음을 적용하여 비밀번호를 쉽게 추측할 수 있는 위험을 최소화합니다.

- 일정에 따라 비밀번호를 변경해야 합니다.
- 쉽게 추정할 수 없는 비밀번호를 지정해야 합니다.
- 잘못된 비밀번호로 여러 번 바인드하면 계정이 잠길 수 있습니다.

Directory Server에서는 트리의 노드에서 여러 가지 방법으로 비밀번호 정책을 설정할 수 있으며 여러 가지 정책 설정 방법을 제공합니다. 자세한 내용은

Directory Server Enterprise Edition 6.0 관리 설명서의 Directory Server 비밀번호 정책을 참조하십시오.

주 – Directory Server의 비밀번호 정책에는 지정된 시간(초)이 경과하면 사용자 비밀번호가 만료되는지 여부를 정의하는 `passwordExp` 속성이 포함됩니다. `passwordExp` 속성을 `on`으로 설정하면 Access Manager의 관리 계정(예: `amldap`, `dsame` 및 `puser`)과 함께 최종 사용자의 비밀번호에 대한 만료 여부가 설정됩니다. Access Manager 관리자의 계정 비밀번호가 만료된 경우 최종 사용자가 로그인하면 해당 사용자에게 비밀번호 변경 화면이 표시됩니다. 그러나 Access Manager는 이 비밀번호 변경 화면과 관련된 사용자를 지정하지 않습니다. 이 경우 관리자를 위해 제공되는 화면이므로 최종 사용자는 비밀번호를 변경할 수 없습니다.

이 문제를 해결하려면 관리자가 Directory Server에 로그인하여 `amldap`, `dsame` 및 `puser` 비밀번호를 변경하거나 `passwordExpirationTime` 속성을 변경해야 합니다.

◆ ◆ ◆ 10장

10 로깅 서비스

Sun Java™ System Access Manager는 사용자 작업, 트래픽 패턴 및 인증 위반과 같은 정보를 기록하기 위한 로깅 서비스를 제공합니다. 또한 관리자는 디버그 파일을 사용하여 설치 문제를 해결할 수 있습니다.

로그 파일

로그 파일은 모니터링하는 각 서비스에 대한 여러 가지 이벤트를 기록합니다. 관리자는 이 파일을 정기적으로 확인해야 합니다. 로그 파일의 기본 디렉토리는 SPARC 시스템의 경우 `/var/opt/SUNWam/logs`, Linux 시스템의 경우 `/var/opt/sun/identity`, HP-UX 시스템의 경우 `/var/opt/sun/identity`, Windows 시스템의 경우 *jes-install-dir*`\identity`입니다. 로그 파일 디렉토리는 Access Manager 콘솔을 사용하여 로깅 서비스에서 구성할 수 있습니다.

기본 로그 파일 유형, 로그에 기록되는 정보 및 로그 파일 형식에 대한 자세한 내용은 **Sun Java System Access Manager 7.1 Technical Overview**의 "Logging Overview"를 참조하십시오.

로깅 서비스에 대한 속성 정의는 Access Manager 콘솔에 있는 도움말 버튼을 눌러 온라인 도움말을 참조하십시오.

Access Manager 서비스 로그

서비스 로그 파일에는 액세스 로그 파일과 오류 로그 파일의 두 가지유형이 있습니다. 액세스 로그 파일에는 작업 시도와 성공적인 결과에 대한 기록이 포함됩니다. 오류 로그 파일은 Access Manager 서비스 내에서 발생한 오류를 기록합니다. 플랫 로그 파일에는 `.error` 또는 `.access` 확장자가 붙습니다. 데이터베이스 열 이름은 Oracle 데이터베이스의 경우 `_ERROR` 또는 `_ACCESS`로 끝나고 MySQL 데이터베이스는 `_error` 또는 `_access`로 끝납니다. 예를 들어 콘솔 이벤트를 기록하는 플랫 파일의 이름은 `amConsole.access`로, 같은 이벤트를 기록하는 데이터베이스 열의 이름은 `AMCONSOLE_ACCESS`로 지정됩니다. 다음 절에서는 로깅 서비스에서 기록하는 로그 파일에 대해 설명합니다.

세션 로그

로깅 서비스는 세션 서비스에 대해 다음 이벤트를 기록합니다.

- 로그인
- 로그아웃
- 세션 유휴 시간 초과
- 세션 최대 시간 초과
- 로그인 실패
- 세션 재활성화
- 세션 삭제

세션 로그에는 `amSSO` 접두어가 붙습니다.

콘솔 로그

Access Manager 콘솔 로그는 조직, 조직 구성 단위, 사용자, 역할, 정책 및 그룹 등을 포함한 Identity 관련 객체, 정책 및 서비스의 생성, 삭제 및 수정을 기록합니다. 또한 비밀번호를 포함한 사용자 속성 수정, 역할 및 그룹에서의 사용자 추가 및 제거를 기록합니다. 이외에도 콘솔 로그는 위임 및 데이터 저장소 작업을 기록합니다. 콘솔 로그에는 `amConsole` 접두어가 붙습니다.

인증 로그

인증 구성 요소는 사용자 로그인과 로그아웃을 기록합니다. 인증 로그에는 `amAuthentication` 접두어가 붙습니다.

연합 로그

연합 구성 요소는 인증 도메인 생성 및 호스트 공급자 생성을 포함하나 이에 제한되지 않은 연합 관련 이벤트를 기록합니다. 연합 로그에는 `amFederation` 접두어가 붙습니다.

정책 로그

정책 구성 요소는 정책 관리(정책 생성, 삭제 및 수정) 및 정책 평가를 포함하나 이에 제한되지 않은 정책 관련 이벤트를 기록합니다. 정책 로그에는 `amPolicy` 접두어가 붙습니다.

에이전트 로그

정책 에이전트 로그는 사용자에게 허용 또는 거부된 로그 자원에 관한 로깅 예외 기록을 담당합니다. 에이전트 로그에는 amAgent 접두어가 붙습니다. amAgent 로그는 에이전트 서버에만 있습니다. 에이전트 이벤트는 Access Manager 서버에서 인증 로그에 기록됩니다. 이 기능에 대한 자세한 내용은 대상 정책 에이전트에 대한 설명서를 참조하십시오.

SAML 로그

SAML 구성 요소는 명제 및 아티팩트 생성 또는 제거, 응답 및 요청 정보, SOAP 오류를 포함하나 이에 제한되지 않은 SAML 관련 이벤트를 기록합니다. 세션 로그에는 amSAML 접두어가 붙습니다.

amadmin 로그

명령줄 로그는 명령줄 도구를 사용한 작업 중에 발생한 이벤트 오류를 기록합니다. 이러한 이벤트에는 서비스 스키마 로드, 정책 생성 및 사용자 삭제 등이 포함됩니다(이에 제한되지 않음). 명령줄 로그에는 amAdmin 접두어가 붙으며, amadmin.access 및 amadmin.error 로그 파일은 주 로깅 디렉토리의 하위 디렉토리에 있습니다. 기본적으로 amadmin 명령줄 도구 로그 파일은 /var/opt/SUNWam/logs에 있습니다.

로깅 기능

로깅 서비스에는 추가 기능을 사용할 수 있도록 해주는 여러 가지의 특수 기능이 있습니다. 이러한 기능에는 보안 로깅 사용, 명령줄 로깅 및 원격 로깅이 포함됩니다.

보안 로깅

로깅 기능에 추가 보안 수단을 적용합니다(선택 사항). 보안 로깅을 통해 보안 로그의 인증되지 않은 변경이나 손상을 감지할 수 있습니다. 이 기능을 사용하기 위해 특별한 코딩이 필요하지는 않습니다. 보안 로깅은 시스템 관리자가 구성한 미리 등록된 인증서를 사용하여 수행됩니다. 이러한 MAC(Manifest Analysis and Certification)는 모든 로그 레코드에 대해 생성 및 저장됩니다. 특수 '서명' 로그 레코드가 정기적으로 삽입되어 해당 지점에 기록된 로그의 내용에 대한 서명을 나타냅니다. 두 레코드의 조합으로 로그가 손상되지 않았음을 확인할 수 있습니다. 보안 로깅을 활성화하는 방법은 JSS(Java Security Server) 공급자를 사용하는 방법과 JCE(Java Cryptography Extension) 공급자를 사용하는 방법이 있습니다.

▼ JSS 공급자를 통해 보안 로깅을 활성화하려면

1 이름이 `Logger`인 인증서를 만들어 Access Manager를 실행 중인 배포 컨테이너에 설치합니다.

Application Server에 대한 자세한 지침은 **Sun Java System Application Server Enterprise Edition 8.2 Administration Guide**의 "Working with Certificates and SSL"을 참조하십시오.

Web Server에 대한 자세한 지침은 **Sun Java System Web Server 7.0 Administrator's Guide**의 "Managing Certificates"를 참조하십시오.

2 Access Manager 콘솔을 사용하여 로깅 서비스 구성에서 보안 로깅을 활성화하고 변경 내용을 저장합니다. 관리자는 로깅 서비스의 다른 속성에 대한 기본값도 수정할 수 있습니다.

로깅 디렉토리가 기본 디렉토리(`/var/opt/SUMWam/logs`)에서 변경된 경우 권한이 0700으로 설정되었는지 확인하십시오. 로깅 서비스는 디렉토리가 없으면 만들지만 권한이 0755로 설정된 디렉토리를 생성하게 됩니다.

또한 기본값에서 다른 디렉토리를 지정하는 경우 웹 컨테이너의 `server.policy` 파일에 있는 다음 매개 변수를 새 디렉토리로 변경해야 합니다.

```
permission java.io.FilePermission "/var/opt/SUNWam/logs/*","delete,write"
```

3 *AccessManager-base*`/SUNWam/config` 디렉토리에 인증서 데이터베이스 비밀번호를 포함한 파일을 만들고 이름을 `.wtpass`로 지정합니다.

주 – 파일 이름 및 이 파일에 대한 경로는 `AMConfig.properties` 파일에서 구성할 수 있습니다. 자세한 내용은 **Access Manager Administration Reference**에 있는 AMConfig.properties 파일 참조 장의 "Certificate Database"를 참조하십시오.

보안을 위해 배포 컨테이너 사용자가 이 파일에 대한 읽기 권한을 가진 유일한 관리자임을 확인합니다.

---

4 서버를 다시 시작합니다.

보안 로깅 시작 시에 `/var/opt/SUNWam/debug/amLog` 파일에 잘못된 확인 오류가 기록될 수 있으므로 보안 로그 디렉토리를 지워야 합니다.

보안 로그에 허용되지 않은 변경 또는 손상이 있는지 알아보려면 확인 프로세스에서 `/var/opt/SUNWam/debug/amLog`에 기록한 오류 메시지를 확인합니다. 손상을 수동으로 확인하려면 `VerifyArchive` 유틸리티를 실행합니다. 자세한 내용은 **Access Manager Administration Reference**의 VerifyArchive 명령줄 장을 참조하십시오.

▼ JCE 공급자를 통해 보안 로깅을 활성화하려면

1 Java keytool 명령으로 Logger라는 인증서를 만들고 JKS 키 저장소에 설치합니다. 예를 들면 다음과 같습니다.

```
JAVA-HOME/jre/lib/security/Logger.jks
```

Application Server에 대한 자세한 지침은 **Sun Java System Application Server Enterprise Edition 8.2 Administration Guide**의 "Working with Certificates and SSL"을 참조하십시오.

Web Server에 대한 자세한 지침은 **Sun Java System Web Server 7.0 Administrator's Guide**의 "Managing Certificates"를 참조하십시오.

2 Access Manager 콘솔을 사용하여 로깅 서비스 구성에서 보안 로깅을 활성화하고 변경 내용을 저장합니다. 관리자는 로깅 서비스의 다른 속성에 대한 기본값도 수정할 수 있습니다.

로깅 디렉토리가 기본 디렉토리(/var/opt/SUMWam/logs)에서 변경된 경우 권한이 0700으로 설정되었는지 확인하십시오. 로깅 서비스는 디렉토리가 없으면 만들지만 권한이 0755로 설정된 디렉토리를 생성하게 됩니다.

또한 기본값에서 다른 디렉토리를 지정하는 경우 웹 컨테이너의 server.policy 파일에 있는 다음 매개 변수를 새 디렉토리로 변경해야 합니다.

```
permission java.io.FilePermission "/var/opt/SUNWam/logs/*","delete,write"
```

3 *AccessManager-base*/SUNWam/config 디렉토리에 JKS 키 저장소 비밀번호가 포함된 파일을 만들고 이름을 .wtpass로 지정합니다.

주 – 파일 이름 및 이 파일에 대한 경로는 AMConfig.properties 파일에서 구성할 수 있습니다. 자세한 내용은 **Access Manager Administration Reference**의 AMConfig.properties 파일 참조 장에 있는 "Certificate Database"를 참조하십시오.

보안을 위해 배포 컨테이너 사용자가 이 파일에 대한 읽기 권한을 가진 유일한 관리자임을 확인합니다.

---

4 *AccessManager-base*/config/xml 디렉토리에 있는 amLogging.xml 파일에서 다음 항목을 편집합니다:

```
sun-am-logging-secure-log-helper

                <AttributeSchema name="iplanet-am-logging-secure-log-helper"
                    type="single"
                    syntax="string"
                    i18nKey="">
                    <DefaultValues>
                        <Value>com.sun.identity.log.secure.impl.SecureLogHelperJCEImpl</Value>
                    </DefaultValues>
                </AttributeSchema>
```

sun-am-logging-secure-certificate-store

```
<AttributeSchema name="iplanet-am-logging-secure-certificate-store"
    type="single"
    syntax="string"
    i18nKey="">
    <DefaultValues>
        <Value>/dir-to-signing-cert-store/Logger.jks</Value>
    </DefaultValues>
</AttributeSchema>
```

5 **기존 서비스 스키마인 iPlanetAMLoggingService를 삭제합니다. 예를 들면 다음과 같습니다.**

```
./amadmin -u amadmin -w netscape -r iPlanetAMLoggingService
```

6 **amadmin 명령줄 도구를 사용하여 편집된 amLoging.xml 파일을 Access Manager로 로드합니다. 예를 들면 다음과 같습니다.**

```
./amadmin -u amadmin -w netscape -s /etc/opt/SUNWam/config/xml/amLogging.xml
```

7 **서버를 다시 시작합니다.**

보안 로그에 허용되지 않은 변경 또는 손상이 있는지 알아보려면 확인 프로세스에서 /var/opt/SUNWam/debug/amLog에 기록한 오류 메시지를 확인합니다. 손상을 수동으로 확인하려면 VerifyArchive 유틸리티를 실행합니다. 자세한 내용은 **Access Manager Administration Reference**의 VerifyArchive 명령줄 장을 참조하십시오.

명령줄 로깅

amadmin 명령줄 도구를 사용해 Directory Server에서 Identity 객체(예: 조직, 사용자 및 역할)를 생성, 수정 및 삭제할 수 있습니다. 이 도구는 또한 서비스 템플리트를 로드, 생성 및 등록할 수 있습니다. 로깅 서비스는 -t 옵션을 호출하여 이러한 작업을 기록할 수 있습니다. AMConfig.properties의 com.iplanet.am.logstatus 등록 정보가 활성(ACTIVE) 상태이면 로그 레코드가 생성됩니다. 이 등록 정보는 기본적으로 사용 가능합니다. 명령줄 로그에는 amAdmin 접두어가 붙습니다. 자세한 내용은 **Access Manager Administration Reference**의 "The amadmin Command Line Tool"을 참조하십시오.

로깅 등록 정보

AMConfig.properties 파일에는 로깅 출력에 영향을 주는 다음과 같은 등록 정보가 있습니다.

com.iplanet.am.logstatus=ACTIVE	이 등록 정보는 로깅을 활성 또는 비활성화합니다. 기본값은 ACTIVE입니다.

`iplanet-am-logging.`*service*`.level=` *level*	*service*는 서비스의 일반 로그 파일 이름입니다. 예를 들어 `amSAML.access`의 로깅 수준을 지정하려면 `iplanet-am-logging.amSAML.access.level` 등록 정보를 사용합니다.*level*은 `java.util.logging.Level` 값 중 하나이며 로그 파일에 기록되는 세부 정보의 수준을 나타냅니다. 수준은 OFF, SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST 및 ALL이 있습니다. 대부분의 서비스는 INFO 이하의 정보 수준으로 로그를 기록합니다.

원격 로깅

Access Manager는 원격 로깅을 지원합니다. 따라서 클라이언트 응용 프로그램이 Access Manager 서버가 설치된 호스트를 사용하여 원격 시스템에 배포된 Access Manager 인스턴스에 로그 레코드를 만들 수 있습니다. 원격 로깅은 다음 중 하나의 시나리오에 의해 시작됩니다.

1. Access Manager 인스턴스의 이름 지정 서비스에 있는 로깅 URL이 원격 인스턴스를 가리키고 이 둘 사이에 신뢰 관계가 구성되어 있는 경우 원격 Access Manager 인스턴스에 로그가 기록됩니다.
2. Access Manager SDK가 원격 Access Manager 인스턴스에 대해 설치되어 있고 클라이언트(또는 단순 Java 클래스)가 로깅 API를 사용하는 SDK 서버에서 실행 중이면 원격 Access Manager 시스템에 로그가 기록됩니다.
3. Access Manager 에이전트가 로깅 API를 사용하는 경우.

▼ 웹 컨테이너를 사용하여 원격 로깅을 활성화하려면

1 Application Server 또는 Web Server의 관리 콘솔에 로그인하고 다음 JVM 옵션을 추가합니다.

- `java.util.logging.manager=com.sun.identity.log.LogManager`
- `java.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`

  Application Server 관리 콘솔에 대한 자세한 내용은 **Sun Java System Application Server Enterprise Edition 8.2 Administration Guide**를 참조하십시오.

  Web Server 관리 콘솔에 대한 자세한 내용은 **Sun Java System Web Server 7.0 Administrator's Guide**를 참조하십시오.

- 사용 중인 Java™ 2 Platform, Standard Edition이 1.4 이상이면 명령줄에서 다음을 호출하여 수행합니다.

  ```
  java -cp /AccessManager-base /SUNWam/lib/am_logging.jar:/ AccessManager-base
  /SUNWam/lib/xercesImpl.jar:/ AccessManager-base
  /SUNWam/lib/xmlParserAPIs.jar:/ AccessManager-base /SUNWam/lib/jaas.jar:/
  AccessManager-base /SUNWam/lib/xmlParserAPIs.jar:/ AccessManager-base
  /SUNWam/lib/servlet.jar:/ AccessManager-base /SUNWam/locale:/
  AccessManager-base/SUNWam/lib/am_services.jar:/
  AccessManager-base/SUNWam/lib/am_sdk.jar:/
  AccessManager-base/SUNWam/lib/jss311.jar:/ AccessManager-base/SUNWam/lib:.

  -Djava.util.logging.manager=com.sun.identity.log.LogManager

  -Djava.util.logging.config.file=/ AccessManager-base
  /SUNwam/lib/LogConfig.properties
  ```

- 사용 중인 Java 2 Platform, Standard Edition이 1.4 이전 버전이면 명령줄에서 다음을 호출하여 수행합니다.

  ```
  java -Xbootclasspath/a:/ AccessManager-base /SUNWam/lib/jdk_logging.jar -cp
  /AccessManager-base /SUNWam/lib/am_logging.jar:/ AccessManager-base
  /SUNWam/lib/xercesImpl.jar:/ AccessManager-base
  /SUNWam/lib/xmlParserAPIs.jar:/ AccessManager-base /SUNWam/lib/jaas.jar:/
  AccessManager-base /SUNWam/lib/xmlParserAPIs.jar:/ AccessManager-base
  /SUNWam/lib/servlet.jar:/ AccessManager-base /SUNWam/locale:/
  AccessManager-base/SUNWam/lib/am_services.jar:/
  AccessManager-base/SUNWam/lib/am_sdk.jar:/
  AccessManager-base/SUNWam/lib/jss311.jar:/ AccessManager-base/SUNWam/lib:.

  -Djava.util.logging.manager=com.sun.identity.log.LogManager

  -Djava.util.logging.config.file=/ AccessManager-base
  /SUNwam/lib/LogConfig.properties
  ```

2 *AccessManager-base*/SUNWam/lib**에 있는** LogConfig.properties**에 다음 매개 변수가 구성되어 있는지 확인합니다.**

- `iplanet-am-logging-remote-handler=com.sun.identity.`

  `log.handlers.RemoteHandler`

- `iplanet-am-logging-remote-formatter=com.sun.`

  `identity.log.handlers.RemoteFormatter`

- `iplanet-am-logging-remote-buffer-size=1`

  원격 로깅은 로그 레코드 수를 기반으로 버퍼링을 지원합니다. 이 값은 레코드의 수에 따라 로그 버퍼 크기를 정의합니다. 버퍼가 꽉 차면 버퍼링된 레코드는 모두 서버로 플러시됩니다.

- `iplanet-am-logging-buffer-time-in-seconds=3600`

이 값은 로그 버퍼 클리너 스레드를 호출하는 시간 제한 기간을 정의합니다.

- `iplanet-am-logging-time-buffering-status=OFF`

  이 값은 로그 버퍼링 및 버퍼 클리너 스레드의 사용 가능 여부를 정의합니다. 기본적으로 이 기능은 비활성화되어 있습니다.

  타이머 기반 버퍼링이 활성화(`iplanet-am-logging-time-buffering-status=ON`)된 경우 로그 레코드의 수가 `iplanet-am-logging-remote-buffer-size`에 지정된 값에 도달하거나 타이머가 `iplanet-am-logging-buffer-time-in-seconds`에 지정된 시간 제한 값이 만료되면 로그 레코드의 버퍼가 로깅 서비스를 제공하는 AM 서버로 플러시됩니다. 버퍼 크기에 도달하기 전에 타이머가 만료되면 버퍼에 들어있는 레코드가 전송됩니다. 원격 로깅의 타이머 기반 버퍼링을 비활성화하면 버퍼 크기에 따라 버퍼를 플러시하는 시기가 결정됩니다. 예를 들어 버퍼 크기가 10이고 응용 프로그램에서 7개 레코드만 보내는 경우 버퍼는 플러시되지 않으며 로그 레코드도 기록되지 않습니다. 응용 프로그램이 종료되면 버퍼의 레코드가 플러시됩니다.

주 – 로그 파일이 비어 있으면 보안 로깅에 "확인 실패" 메시지가 표시될 수 있습니다.이는 생성된 파일의 수가 아카이브 크기와 같기 때문이며, 이 경우 보안 로깅은 이 세트부터 아카이브한 다음 다시 시작합니다. 대부분의 인스턴스에서는 이 오류를 무시해도 됩니다. 레코드 수가 아카이브 크기와 같으면 오류가 표시되지 않습니다.

---

3 클라이언트 SDK가 있는 프로그램을 사용하는 경우 AMConfig.properties 파일의 다음 등록 정보를 적절히 설정해야 합니다.

- com.iplanet.am.naming.url
- com.sun.identityagents.app.username
- com.iplanet.am.service.password
- com.iplanet.am.server.protocol
- com.iplanet.am.server.host
- com.iplanet.am.server.port

`/opt/SUNWam/war` 디렉토리에 있는 클라이언트 SDK 샘플(`README.clientsdk`)을 참조하십시오. 이 샘플에서는 `/opt/SUNWam/war/clientsdk-samples` 디렉토리에 대해 `AMConfig.properties` 및 make 파일을 생성하는 방법에 대해 설명하며,이러한 파일은 샘플의 'makefiles' 컴파일 및 실행 항목에서 사용됩니다.

오류 및 액세스 로그

Access Manager 로그 파일에는액세스 로그 파일 및 오류 로그 파일의 두 가지 유형이 있습니다.

액세스 로그 파일에는 Access Manager 배포와 관련된 일반 감사 정보가 기록됩니다. 로그에는 인증 성공과 같은 이벤트에 대한 단일 레코드가 포함될 수 있습니다. 로그에는 동일한 이벤트에 대해 여러 레코드가 포함될 수 있습니다. 예를 들어 관리자가 콘솔을

사용하여 속성 값을 변경하면 로깅 서비스에서 하나의 레코드에 변경 시도를 기록합니다. 또한 로깅 서비스는 두 번째 레코드에 변경의 실행 결과를 기록합니다.

오류 로그 파일에는 응용 프로그램에서 발생한 오류가 기록됩니다. 작업 오류는 오류 로그에 기록되고, 작업 시도는 액세스 로그 파일에 기록됩니다.

플랫 로그 파일에는 .error 또는 .access 확장자가 붙습니다. 데이터베이스 테이블 이름은 _ERROR 또는 _ACCESS로 끝납니다. 예를 들어 플랫 파일 로깅 콘솔 이벤트의 이름은 amConsole.access이지만 동일한 이벤트를 기록하는 데이터베이스 테이블의 이름은 AMCONSOLE_ACCESS 또는 amConsole_access입니다.

다음 표에는 Access Manager의 각 구성 요소에서 생성되는 로그 파일에 대해 간략히 정리되어 있습니다.

표 10–1 Access Manager 구성 요소 로그

구성 요소	로그 파일 이름 접두어	기록된 정보
세션	amSSO	로그인 시간, 로그아웃 시간, 시간 초과 제한과 같은 세션 관리 속성 값.
관리 콘솔	amConsole	Identity 관련 객체, 영역, 정책의 생성, 삭제, 수정과 같이 관리 콘솔을 통해 수행된 사용자 작업.
인증	amAuthentication	사용자 로그인 및 로그아웃.
아이디 연합	amFederation	인증 도메인 생성 및 호스트 공급자 생성 등의 연합 관련 이벤트. 연합 로그에는amFederation 접두어가 붙습니다.
인증(정책)	amPolicy	정책 생성, 삭제 또는 수정 및 정책 평가와 같은 정책 관련 이벤트.
정책 에이전트	amAgent	사용자가 액세스했거나 사용자에 대한 액세스가 거부된 자원 관련 예외. amAgent 로그는 정책 에이전트가 설치된 서버에 상주합니다. 에이전트 이벤트는 Access Manager 시스템에서 인증 로그에 기록됩니다.
SAML	amSAML	명제, 아티팩트 생성 또는 삭제, 응답 및 요청 세부 정보, SOAP 오류와 같은 SAML 관련 이벤트.
명령줄	amAdmin	amadmin 명령줄 도구를 사용한 작업 도중 발생한 이벤트 오류. 플랫 파일 로깅을 지정하면 amAdmin 로그 파일이 주 로깅 디렉토리(기본적으로 /var/opt/SUNWam/logs)의 amadmincli 하위 디렉토리에 저장됩니다. 예: 서비스 스키마 로딩, 정책 생성 및 사용자 삭제.

Access Manager 로그 파일의 목록과 설명에 대한 자세한 내용은 **Access Manager Administration Reference**의 Access Manager Log File Reference를 참조하십시오.

디버그 파일

디버그 파일은 로깅 서비스의 기능이 아닙니다. 디버그 파일은 로깅 API와는 독립적인 다른 API를 사용하여 작성됩니다. 디버그 파일은 `/var/opt/SUNWam/debug`에 저장됩니다. 이 위치는 디버그 정보의 수준과 함께 *AccessManager-base*`/SUNWam/lib/` 디렉토리의 `AMConfig.properties` 파일에서 구성할 수 있습니다. 디버그 등록 정보에 대한 자세한 내용은 **Access Manager Administration Reference**의 AMConfig.properties 파일 참조 장을 참조하십시오.

디버그 수준

디버그 파일에 기록할 수 있는 정보의 수준에는 여러 가지가 있습니다. 디버그 수준은 `AMConfig.properties`에 있는 `com.iplanet.services.debug.level` 등록 정보를 사용하여 설정합니다.

1. Off—디버그 정보를 기록하지 않습니다.
2. Error—이 수준은 프로덕션에 사용됩니다. 프로덕션 중에는 디버그 파일에 오류가 있으면 안됩니다.
3. Warning—현재 이 수준은 사용하지 않는 것이 좋습니다.
4. Message—이 수준은 코드 추적을 사용하여 가능한 문제를 경고합니다. 대부분의 Access Manager 모듈은 이 수준을 사용하여 디버그 메시지를 보냅니다.

주 – Warning 및 Message 수준은 프로덕션에서는 사용하면 안됩니다. 이 두 수준은 많은 디버그 메시지와 함께 심각한 성능 저하를 일으킵니다.

디버그 출력 파일

디버그 파일은 모듈에서 기록해야 생성됩니다. 따라서 기본 `error` 모드에서는 디버그 파일에 생성되지 않습니다. 기본 로그인 시에 디버그 수준이 `message`로 설정되어 생성되는 디버그 파일은 다음과 같습니다.

- amAuth
- amAuthConfig
- amAuthContextLocal
- amAuthLDAP
- amCallback
- amClientDetection

- amConsole
- amFileLookup
- amJSS
- amLog
- amLoginModule
- amLoginViewBean
- amNaming
- amProfile
- amSDK
- amSSOProvider
- amSessionEncodeURL
- amThreadManager

가장 자주 사용되는 파일은 `amSDK`, `amProfile` 및 인증과 관련된 모든 파일입니다. 캡처된 정보에는 날짜, 시간 및 메시지 유형(Error, Warning, Message)이 포함됩니다.

디버그 파일 사용

디버그 수준은 기본적으로 `error`로 설정됩니다. 디버그 파일은 관리자가 다음과 같은 작업을 수행하는 경우 유용합니다.

- 사용자 정의 인증 모듈 작성.
- Access Manager SDK를 사용하여 사용자 정의 응용 프로그램 작성. `amProfile` 및 `amSDK` 디버그 파일은 이 정보를 캡처합니다.
- 콘솔 또는 SDK 사용 중에 액세스 권한 문제 해결. `amProfile` 및 `amSDK` 디버그 파일은 이 정보도 캡처합니다.
- SSL 문제 해결
- LDAP 인증 모듈 문제 해결. `amAuthLDAP` 디버그 파일은 이 정보를 캡처합니다.

디버그 파일은 향후 제공될 수 있는 모든 문제 해결 설명서와 함께 사용되어야 합니다. 예를 들어 SSL이 실패하는 경우, 디버그를 message로 활성화하고 `amJSS` 디버그 파일을 확인하여 특정 인증서 오류를 찾을 수 있습니다.

◆ ◆ ◆ 11 장

11 알림 서비스

Sun Java System Access Manager 7.1 알림 서비스를 사용하면 세션 알림을 원격 웹 컨테이너로 보낼 수 있습니다. Access Manager 서버 자체에서 원격으로 실행되는 SDK 응용 프로그램에서 이 서비스를 활성화하여 사용해야 합니다. 이 장에서는 알림을 수신하도록 원격 웹 컨테이너를 활성화하는 방법을 설명하며, 다음 내용으로 구성되어 있습니다.

개요

알림 서비스를 사용하면 Access Manager SDK를 원격으로 실행하는 웹 컨테이너로 세션 알림을 보낼 수 있습니다. 알림은 세션, 정책 및 이름 지정 서비스에만 적용됩니다. 또한 원격 응용 프로그램이 웹 컨테이너에서 실행되고 있어야 합니다. 알림의 목적은 다음과 같습니다.

- 개별 서비스의 클라이언트측 캐시 동기화
- 클라이언트에 대한 더욱 효율적인 실시간 업데이트(알림이 없는 경우 폴링 사용)
- 어떤 클라이언트 응용 프로그램도 변경하지 않고 알림 지원

원격 SDK가 웹 컨테이너에 설치되어 있는 경우에만 알림을 수신할 수 있습니다.

알림 서비스 활성화

다음 단계에 따라 세션 알림을 수신하도록 원격 SSO SDK를 구성할 수 있습니다.

▼ 세션 알림을 수신하려면

1 **시스템 1에 Access Manager를 설치합니다.**

2 **시스템 2에 Sun Java System Web Server를 설치합니다.**

3 **같은 시스템에** `SUNWamsdk`**를 Web Server로 설치합니다.**

Access Manager SDK를 원격으로 설치하기 위한 지침은 **Sun Java Enterprise System 5 설치 설명서**를 참조하십시오.

4 **SDK가 설치된 시스템과 관련하여 다음 사항을 확인합니다.**

a. **SDK가 설치된 서버에서** */remote_SDK_server*`/SUNWam/lib` **및** */remote_SDK_server*`/SUNWam/locale` **디렉토리에 대한 액세스 권한이 올바르게 설정되었는지 확인합니다.**

이들 디렉토리에는 원격 서버의 파일과 jar 파일이 있습니다.

b. **Web Server에 있는** `server.policy` **파일의 Grant 섹션에 다음 권한이 설정되었는지 확인합니다.**

`server.policy`는 설치된 Web Server의 `config` 디렉토리에 있습니다. 필요한 경우 다음 권한을 복사하여 붙여넣습니다.

```
permission java.security.SecurityPermission
"putProviderProperty.Mozilla-JSS"

permission java.security.SecurityPermission "insertProvider.Mozilla-JSS";
```

c. `server.xml`**에 클래스 경로가 올바르게 설정되었는지 확인합니다.**

`server.xml` 역시 설치된 Web Server의 `config` 디렉토리에 있습니다. 일반적으로 클래스 경로는 다음과 같습니다.

```
<JAVA javahome="/export/home/ws61/bin/https/jdk"
serverclasspath="/export/home/ws61/bin/https/jar/webserv-rt.jar:
${java.home}/lib/tools.jar:/export/home/ws61/bin/https/jar/webserv-ext.jar:
/export/home/ws61/bin/https/jar/webserv-jstl.jar:/export/home/ws61/
    bin/https/jar/nova.jar"
classpathsuffix="::/IS_CLASSPATH_BEGIN_DELIM:
                //usr/share/lib/xalan.jar:
                //export/SUNWam/lib/xmlsec.jar:
                //usr/share/lib/xercesImpl.jar:
                //usr/share/lib/sax.jar:
                //usr/share/lib/dom.jar:
                //export/SUNWam/lib/dom4j.jar:
                //export/SUNWam/lib/jakarta-log4j1.2.6.jar:
                //usr/share/lib/jaxm-api.jar:
```

```
                        //usr/share/lib/saaj-api.jar:
                        //usr/share/lib/jaxrpc-api.jar:
                        //usr/share/lib/jaxrpc-impl.jar:
                        //export/SUNWam/lib/jaxm-runtime.jar:
                        //usr/share/lib/saaj-impl.jar:/export/SUNWam
                        //lib:/export/SUNWam/locale:
                        //usr/share/lib/mps/jss3.jar:
                        //export/SUNWam/lib/    am_sdk.jar:
                        //export/SUNWam/lib/am_services.jar:
                        //export/SUNWam/lib/am_sso_provider.jar:
                        //export/SUNWam/lib/swec.jar:
                        //export/SUNWam/lib/acmecrypt.jar:
                        //export/SUNWam/lib/iaik_ssl.jar:
                        //usr/share/lib/jaxp-api.jar:
                        //usr/share/lib/mail.jar:
                        //usr/share/lib/activation.jar:
                        //export/SUNWam/lib/servlet.jar:
                        //export/SUNWam/lib/am_logging.jar:
                        //usr/share/lib/commons-logging.jar:
                        //IS_CLASSPATH_END_DELIM:"
    envclasspathignored="true" debug="false"
    debugoptions="-Xdebug -Xrunjdwp:transport=dt_socket,
    server=y,suspend=n"
    javacoptions="-g"
    dynamicreloadinterval="2">
```

5 **구성 용도로는 원격 SDK 서버에 설치된 SSO 샘플을 사용합니다.**

   a. */remote_SDK_server*/SUNWam/samples/sso **디렉토리로 변경합니다.**

   b. gmake**를 실행합니다.**

   c. */remote_SDK_server*/SUNWam/samples/sso**에서 생성된 클래스 파일을** */remote_SDK_server*/SUNWam/lib/**로 복사합니다.**

6 **Access Manager와 함께 설치된** AMConfig.properties **파일의** am.encryption.pwd **암호화 값을 SDK가 설치된 원격 서버의** AMConfig.properties **파일에 복사합니다.**

   am.encryption.pwd 값은 비밀번호를 암호화하고 해독하는 데 사용됩니다.

7 amadmin**으로 Access Manager에 로그인합니다.**

   http://*AcceessManager-HostName* :3000/amconsole

8 **브라우저 위치 필드에** `http://`*remote_SDK_host*`:58080/servlet/SSOTokenSampleServlet`**을 입력하고** `SSOToken`**을 확인하여 서블릿을 실행합니다.**

`SSOTokenSampleServlet`은 세션 토큰을 확인하고 수신기를 추가하는 데 사용됩니다. 서블릿를 실행하면 다음과 같은 메시지가 출력됩니다.

```
SSOToken host name: 192.18.149.33 SSOToken Principal name:
uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com Authentication type used: LDAP
IPAddress of the host: 192.18.149.33 The token id is
AQIC5wM2LY4SfcyURnObg7vEgdkb+32T43+RZN30Req/BGE= Property: Company is - Sun
Microsystems Property: Country is - USA SSO Token Validation test Succeeded
```

9 **클라이언트 SDK가 설치된 컴퓨터의** `AMConfig.properties`**에서** `com.iplanet.am.notification.url=` **등록 정보를 다음과 같이 설정합니다.**

```
com.iplanet.am.notification.url=http://clientSDK_host.domain:port
/servlet
    com.iplanet.services.comm.client.PLLNotificationServlet
```

10 **Web Server를 다시 시작합니다.**

11 `amadmin`**으로 Access Manager에 로그인합니다.**

`http://`*AcceessManager-HostName* `:3000/amconsole`

12 **브라우저 위치 필드에** `http://`*remote_SDK_host*`:58080/servlet/SSOTokenSampleServlet`**을 입력하고** `SSOToken`**을 확인하여 서블릿을 다시 실행합니다.**

원격 SDK가 실행되고 있는 컴퓨터에서 알림을 수신하는 경우 세션 상태가 변경되면 각각의 수신기가 호출됩니다. 원격 SDK가 웹 컨테이너에 설치되어 있는 경우에만 알림을 수신할 수 있습니다.

▼ 포털 전용 설치에서 알림 서비스를 활성화하려면

이 절에서는 기본적으로 **폴링** 모드에서 실행되는 포털 전용 설치 시 WebLogic 8.1에서 알림을 활성화하는 단계를 설명합니다. 또한 `amserver` 구성 요소가 포함된 포털 인스턴스에는 이 절차가 필요하지 않습니다. `amserver` 구성 요소의 경우 알림을 수행하도록 자동으로 구성되기 때문입니다.

1 **WebLogic에** `PLLNotificationServlet`**을 등록합니다.**

WebLogic 8.1을 사용하려면 웹 응용 프로그램을 배포해야 합니다. 또한 브라우저에서 액세스하는 경우 다음 메시지가 반환되도록 하려면 서블릿 URL이 유효해야 합니다.

`Webtop 2.5 Platform Low Level` **알림 서블릿**

2 다음과 같이 등록된 URL을 AMConfig.properties에 입력합니다.

```
com.iplanet.am.notifaction.url=http://
weblogic_instance-host.domain:port/notification/PLLNotificationServlet
```

3 AMConfig.properties에서 폴링을 비활성화합니다. 이렇게 하면 알림이 자동으로 활성화됩니다.

```
com.iplanet.am.session.client.polling.enable=false
```

4 WebLogic을 다시 시작하고 구성을 테스트합니다.

디버그 모드를 message로 설정한 경우 트리거될 때 포털에 도착하는 세션 알림이 표시됩니다. 예를 들어 Access Manager 콘솔에서 사용자가 종료되면 알림 이벤트가 발생합니다.

색인