# CA SiteMinder® Secure Proxy Server

관리 안내서

12.52 SP1

ca technologies

# CA Technologies 제품 참조

이 문서는 다음 CA Technologies 제품을 참조합니다 :

- CA SiteMinder for Secure Proxy Server
- CA SiteMinder®

# CA 에 문의

### 기술 지원팀에 문의

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

# 설명서 변경 사항

CA SiteMinder®의 이전 릴리스에서 발견된 문제점으로 인해 다음과 같은 내용이 12.52 설명서에서 업데이트되었습니다.

- JCE(Java Cryptographic Extension)에 대한 패치가 필요함 - 이 단원은 Java 에서 제공되는 암호화 알고리즘을 사용하기 위한 업데이트가 필요한 파일에 대해 설명합니다. CQ 174929 를 해결합니다.
- Windows 통합 인증 (페이지 261) - 이 장은 Windows 통합 인증이 지원되는 운영 체제와 Windows 인증 체계를 활성화하는 데 필요한 ACO 매개 변수에 대해 설명합니다. CQs 172603 및 172605 를 해결합니다.
- 인증 및 권한 부여 (페이지 207) - 이 장은 AgentName 형식과 인증 및 권한 부여 요청 형식에 대해 설명합니다. CQ 177424, 173173, 172762, 172758, 172760, 172764 를 해결합니다.

## 이 안내서의 두 번째 에디션에는 다음과 같은 변경된 사항이 포함되어 있습니다.

### 설명서에서만 변경된 내용

프록시 서비스 구성 (페이지 121) - -Dhttp_connection_timeout 이 구성된 경우 http_connection_timeout 의 동작에 대한 설명이 추가되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 184414, 21695829 및 STAR 21695829 를 해결합니다.

인증 REST 인터페이스 (페이지 216) - URI 형식이 업데이트되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 184721 및 STAR 21755360-1 을 해결합니다.

사전 요구 사항 (페이지 33) - ncurses 패키지의 요구 사항에 대한 설명 및 CA SiteMinder for Secure Proxy Server 가 다른 컴퓨터에 설치되어야 한다는 설명이 추가되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 184218 및 184222 를 해결합니다.

필터 구현 (페이지 317) - lib 디렉터리에 대한 경로가 수정되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 183762 및 STAR 21730064-1 을 해결합니다.

웹 에이전트 대체 역할을 하는 SPS (페이지 80) - 웹 에이전트 옵션 팩의 요구 사항이 보다 자세히 업데이트되었습니다.

CA SiteMinder for Secure Proxy Server 를 웹 에이전트 대체 서비스로 사용하기 위한 사전 요구 사항 (페이지 81) - 웹 에이전트 옵션 팩 설치를 위한 요구 사항 설명이 삭제되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 181448 및 STAR 21657979-1 을 해결합니다.

자체 서명된 인증서 생성 (페이지 255) - 자체 서명된 인증서를 생성하는 명령이 수정되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 172433 을 해결합니다.

인증 기관으로부터 인증서를 다운로드하여 설치 (페이지 256) - RootCA 또는 자체 서명된 인증서를 ca-bundle.cert 에 추가하는 방법에 대한 단계가 추가되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 172433 을 해결합니다.

Debug 특성 (페이지 171) - xmlns:nete 의 예제 값이 수정되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 184884 및 STAR 21754567-02 를 해결합니다.

CA SiteMinder for Secure Proxy Server 서버 설정 구성 (페이지 95) - enablecachepostdata, worker.ajp13.max_threads, http_connection_pool_max_size, http_connection_timeout, http_connection_stalecheck, http_connection_pool_min_size, http_connection_pool_incremental_factor 의 기본값이 수정되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 182068 및 STAR 21652689-1 을 해결합니다.

**추가 정보:**

CA SiteMinder® SPS 업그레이드 (페이지 38)
업그레이드를 위한 추가 태스크 (페이지 38)
SSL 예외 무시 (페이지 110)

# 목차

## 제 18 장: SPS API 297



## 제 19 장: 문제 해결 319

# 제 1 장: SiteMinder Secure Proxy Server 개요

이 섹션은 다음 항목을 포함하고 있습니다.



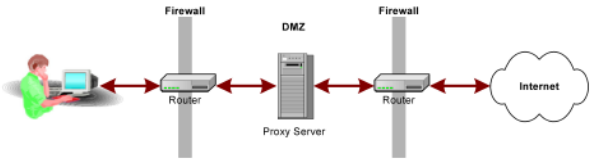
## CA SiteMinder for Secure Proxy Server 아키텍처 개요

CA SiteMinder?for Secure Proxy Server(CA SiteMinder for Secure Proxy Server)는 액세스 제어를 위한 프록시 기반 솔루션을 제공하는 독립 실행형 서버입니다. CA SiteMinder for Secure Proxy Server 는 기업을 위한 네트워크 게이트웨이를 제공하는 프록시 엔진을 사용하며 기존 쿠키 기반 기술에 의존하지 않는 다중 세션 체계를 지원합니다.

### 프록시 서버 아키텍처

기존 프록시 서버는 방화벽과 내부 네트워크 사이에 있으며 내부 네트워크의 사용자에게 리소스 캐싱 및 보안 기능을 제공합니다. 기존 프록시 서버는 인터넷 상의 모든 리소스에 대해 사용자 그룹을 대신하는 프록시 역할을 합니다.

다음 그림에서는 프록시 서버 구성을 보여 줍니다. 프록시 서버는 DMZ(Demilitarized Zone)에서 이러한 리소스에 대한 요청이 더욱 빠르게 처리되도록 액세스한 리소스를 자주 캐시합니다.



## 기존 리버스 프록시 서버 아키텍처

리버스 프록시 서버는 하나 이상의 대상 서버를 나타냅니다. 리버스 프록시 아키텍처를 사용하면 일반적으로 다음과 같은 기능이 제공됩니다.

- 캐시된 리소스
- 보안
- SSL 가속화
- 부하 분산

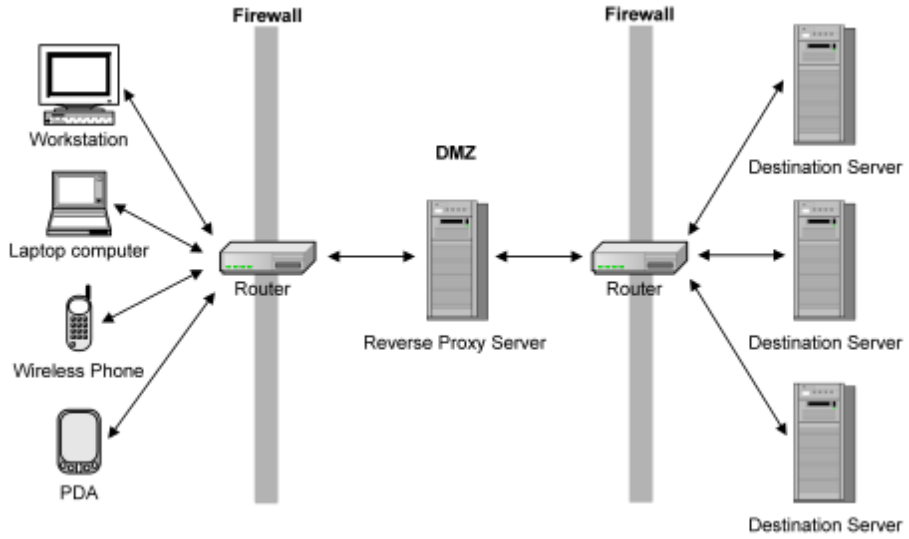
대상 서버에서 직접 리소스를 요청하는 대신 리버스 프록시 서버가 대상 서버의 콘텐츠 중 상당수를 캐시하므로 사용자 액세스가 용이해집니다. 프록시가 사용자에게 투명하게 처리되고 엔터프라이즈의 대상 서버를 대신하여 작동하므로 이 유형의 프록시 서버 배포는 리버스 프록시로 간주됩니다. 부하 분산을 위해 여러 개의 리버스 프록시 서버를 사용할 수 있으며 이 경우 HTTPS 요청에 대해 일부 SSL 가속화 기능을 제공할 수도 있습니다. 또한 리버스 프록시 서버는 DMZ 뒤에 있는 대상 서버를 격리하여 추가 보안 계층을 제공합니다.

## SPS 아키텍처

CA SiteMinder for Secure Proxy Server 는 리소스 캐싱을 제공하지 않으므로 기존 리버스 프록시 솔루션과는 다릅니다. CA SiteMinder for Secure Proxy Server 는 네트워크 액세스 방법에 상관없이 엔터프라이즈 리소스에 액세스하기 위한 단일 게이트웨이의 역할을 합니다.

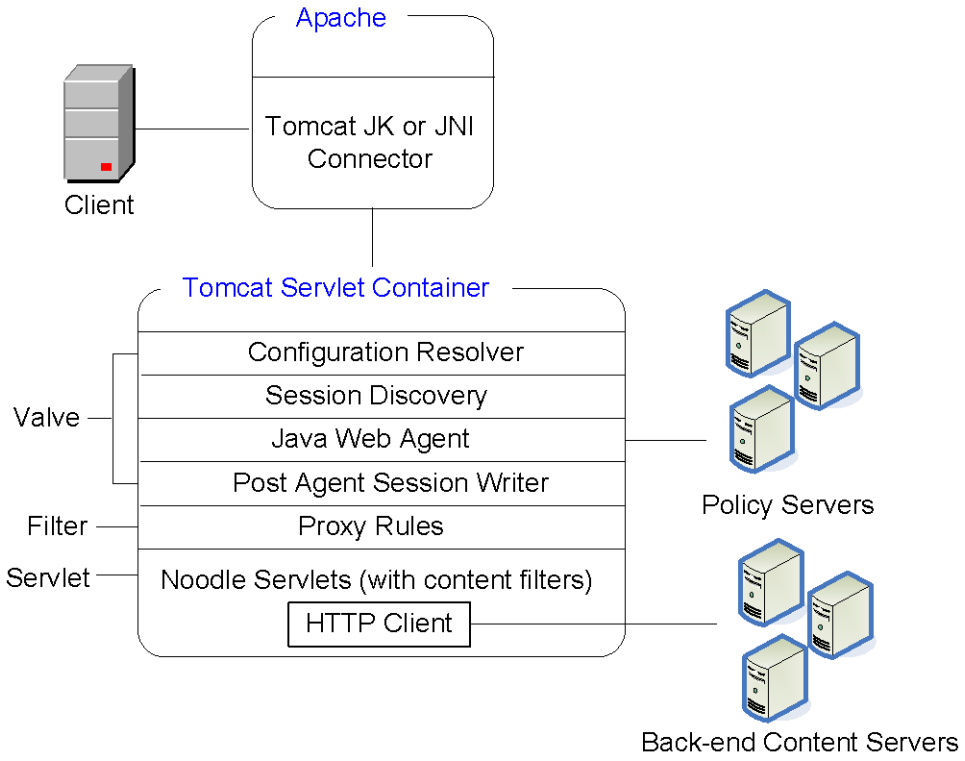
일련의 구성 가능한 프록시 규칙에 따라 CA SiteMinder for Secure Proxy Server 가 사용자 요청을 처리하는 방식이 결정됩니다. 사용자는 사용자 에이전트 유형과 가상 호스트 간의 매핑을 기반으로 여러 세션 체계를 통해 리소스에 액세스할 수 있습니다. 요청은 네트워크에 액세스하는 데 사용되는 장치의 유형에 따라 각기 다른 대상 서버로 라우팅될 수 있습니다.

다음 그림에서는 CA SiteMinder for Secure Proxy Server 의 구성을 보여 줍니다. 사용자는 다양한 장치를 사용하여 CA SiteMinder for Secure Proxy Server 에 액세스합니다. CA SiteMinder for Secure Proxy Server 는 액세스 장치를 기반으로 생성할 세션 체계를 결정한 다음 요청을 적절한 대상 서버로 전달하거나 리디렉션합니다. 사용자는 엔터프라이즈에 리버스 프록시 서버가 있다는 것을 인식하지 못합니다.

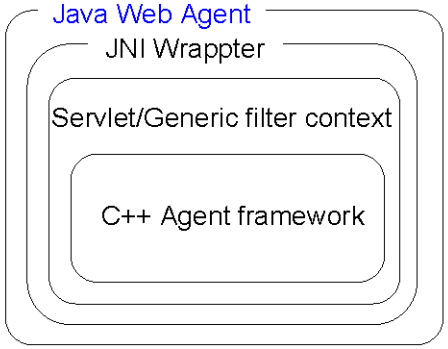


## 구성 요소

독립 실행형 CA SiteMinder for Secure Proxy Server 는 다음 그림과 같이 HTTP 수신기(Apache)와 Tomcat 서블릿 컨테이너로 구성됩니다.




**Java Web Agent Components**



CA SiteMinder for Secure Proxy Server 아키텍처에는 다음 구성 요소가 포함됩니다.

**Apache**

CA SiteMinder for Secure Proxy Server 는 오픈 소스 Apache 웹 서버를 사용하여 들어오는 요청에 대해 HTTP 수신기의 역할을 합니다. 추가 구성 요소인 mod_jk(1.2.18)는 AJP(Apache JServ Protocol)를 사용하여 Apache 웹 서버와 Tomcat 간의 통신을 가능하게 해 주는 Tomcat 커넥터의 역할을 합니다.

**Tomcat**

Tomcat 서버는 CA SiteMinder for Secure Proxy Server 에 Tomcat 서블릿 컨테이너를 제공합니다. Tomcat 초기화는 외부 응용 프로그램 또는 서블릿의 배포를 허용하지 않도록 사용자 지정됩니다. 표준 Tomcat xml(server.xml)은 초기화에 사용되지 않습니다. CA SiteMinder for Secure Proxy Server 의 Tomcat 컨테이너 내에 있는 구성 요소로는 다음이 포함됩니다.

**구성 확인자 ProxyBootstrap**

구성 확인자 proxybootstrap 은 server.conf 파일에서 CA SiteMinder for Secure Proxy Server 구성을 읽고 CA SiteMinder for Secure Proxy Server 를 초기화합니다.

**세션 검색**

세션 검색 구성 요소는 들어오는 요청을 분석하여 CA SiteMinder for Secure Proxy Server 세션 정보를 추출합니다. 사용되는 사용자 에이전트 유형 및 가상 호스트에 따라 이 구성 요소는 적절한 세션 체계를 사용하여 CA SiteMinder for Secure Proxy Server 세션 정보를 추출합니다.

요청에 기존 CA SiteMinder for Secure Proxy Server 세션이 사용되는 경우 이 구성 요소는 요청에 포함된 CA SiteMinder for Secure Proxy Server 세션 식별자를 사용하여 메모리 내 세션 저장소에서 해당 SiteMinder 세션을 추출합니다. CA SiteMinder for Secure Proxy Server 는 세션 유효성 검사를 위해 SiteMinder 세션을 Java 웹 에이전트로 전달합니다. 요청에 기존 CA SiteMinder for Secure Proxy Server 세션이 포함되어 있지 않으면 이 구성 요소는 사용자 인증을 위해 요청을 Java 웹 에이전트로 전달합니다.

**Java 웹 에이전트**

Java 웹 에이전트는 SiteMinder 정책 서버와 함께 사용자를 인증하고 권한을 부여합니다.

### Post 에이전트 세션 작성기

Post 에이전트 세션 작성기는 쿠키를 사용하지 않는 세션 체계에 대한 추가 처리를 수행합니다. Java 웹 에이전트가 사용자 인증 및 권한 부여를 수행하고 SiteMinder 세션을 생성한 후 이 구성 요소가 CA SiteMinder for Secure Proxy Server 세션 식별자를 생성합니다. 이 식별자는 Java 웹 에이전트에 의해 생성된 SiteMinder 세션에 추가됩니다.

그런 다음 이 세션 식별자가 CA SiteMinder for Secure Proxy Server 의 메모리 내 세션 저장소에서 유지 관리됩니다. 이 구성 요소는 세션 저장소에서 세션을 유지 관리할 뿐 아니라 URI 도 변환합니다. 예를 들어 Post 에이전트 세션 작성기는 simple_url 세션 체계에 대한 URI 를 조작합니다.

### 프록시 규칙 서블릿 필터

프록시 규칙 서블릿 필터는 proxyrules.xml 파일에서 프록시 규칙을 로드합니다. 들어오는 요청과 프록시 규칙에 따라 요청은 백엔드 서버로 전달되거나 리디렉션됩니다. 요청이 전달되면 오픈 소스 구성 요소인 누들이 사용됩니다.

프록시 규칙을 변경할 경우 시스템을 다시 시작하지 않아도 변경 내용이 적용됩니다. proxyrules.xml 파일에 변경 내용이 있으면 프록시 규칙이 다시 로드됩니다.

### 누들 서블릿

누들 서블릿은 요청을 백엔드 서버로 전달합니다. 또한 누들은 요청을 백엔드 서버로 전송하기 전에 수정할 수 있게 해 주는 프록시 사전 필터의 사용을 지원합니다. 마찬가지로 응답을 사용자 클라이언트로 다시 전송하기 전에 백엔드 서버에서 수신된 응답을 수정할 수 있게 해 주는 프록시 사후 필터에 대한 지원도 사용할 수 있습니다.

### HTTP 클라이언트

HTTP 클라이언트는 요청을 백엔드 서버로 전송하고 백엔드 서버로부터 응답을 수신합니다.

# 제품 기능

CA SiteMinder for Secure Proxy Server 는 다음과 같은 기능을 제공합니다.

**HTTP 및 HTTPS 요청에 대한 액세스 제어**

CA SiteMinder for Secure Proxy Server 를 사용하면 포함된 SiteMinder 웹 에이전트를 통해 HTTP 및 HTTPS 요청과 대상 서버 간의 흐름을 제어할 수 있습니다. 또한 CA SiteMinder for Secure Proxy Server 는 SiteMinder 와 완전히 통합되어 e-비즈니스 트랜잭션을 관리합니다.

**싱글 사인온**

CA SiteMinder for Secure Proxy Server 에 포함된 웹 에이전트는 엔터프라이즈 내 대상 서버에 설치할 수 있는 SiteMinder 웹 에이전트를 사용한 SSO(싱글 사인온)를 비롯하여 엔터프라이즈에서의 SSO 를 가능하게 해 줍니다.

**여러 세션 체계**

세션 체계는 인증 후 사용자의 아이덴티티를 유지 관리하기 위한 방법입니다. 핵심 SiteMinder 제품은 쿠키를 사용하여 세션을 유지 관리합니다. 그러나 CA SiteMinder for Secure Proxy Server 는 SSL ID, 미니 쿠키, 핸드헬드 장치의 장치 ID, URL 다시 쓰기, IP 주소, 세션 체계 API 를 사용하여 생성된 체계 등을 기반으로 세션을 유지 관리할 수 있습니다.

**세션 저장소**

CA SiteMinder for Secure Proxy Server 는 메모리 내 세션 저장소를 갖추고 있습니다. 세션 저장소는 세션 정보를 유지 관리합니다. CA SiteMinder for Secure Proxy Server 는 미니 쿠키 또는 SSL ID 와 같은 토큰을 사용하여 세션 저장소의 세션 정보를 참조합니다. 여러 세션 체계와 메모리 내 세션 저장소를 통해 CA SiteMinder for Secure Proxy Server 는 컴퓨터나 PAD 및 무선 전화 같은 무선 장치 이상의 e-비즈니스 관리 솔루션을 제공할 수 있습니다.

**쿠키를 사용하지 않는 싱글 사인온**

일부 엔터프라이즈는 쿠키 기술을 사용하지 않는 솔루션을 선호합니다. CA SiteMinder for Secure Proxy Server 는 세션 체계와 세션 저장소를 기본적으로 제공하므로 쿠키 기반 세션 관리를 대체하려는 엔터프라이즈에 적절한 솔루션을 제공합니다.

### 지능형 프록시 규칙

프록시 규칙을 사용하면 CA SiteMinder for Secure Proxy Server 에서 클라이언트 요청을 이행하기 위한 경로를 요청된 가상 호스트 또는 URI 문자열 등의 특성에 따라 서로 다르게 구성할 수 있습니다. 프록시 엔진은 일련의 프록시 규칙을 해석하여 사용자 요청을 처리할 방법을 결정합니다.

### 중앙 집중식 액세스 제어 관리

CA SiteMinder for Secure Proxy Server 는 네트워크 리소스에 대한 단일 게이트웨이를 제공하여 회사 네트워크를 분리하고 액세스 제어를 중앙 집중화합니다.

### 엔터프라이즈 클래스 아키텍처

CA SiteMinder for Secure Proxy Server 는 확장성과 관리 용이성을 높일 수 있도록 설계되었습니다.

# 제품 제한 사항

CA SiteMinder for Secure Proxy Server 에는 다음과 같은 제한이 있습니다.

- CA SiteMinder for Secure Proxy Server 는 다른 웹 서버에 대한 플러그인이 아닙니다. CA SiteMinder for Secure Proxy Server 는 완전히 지원되는 독립 실행형 서버입니다.
- CA SiteMinder for Secure Proxy Server 는 로컬 콘텐츠를 지원하지 않습니다. 콘텐츠를 CA SiteMinder for Secure Proxy Server 에 배치하는 기능이 노출되지 않으며 CA SiteMinder for Secure Proxy Server 는 로컬 콘텐츠에 대한 액세스를 제공하기 위한 프록시 규칙을 지원하지 않습니다.
- CA SiteMinder for Secure Proxy Server 를 사용할 경우 CA SiteMinder for Secure Proxy Server 와 동일한 시스템에 웹 서버를 배치할 수 없습니다. 같은 시스템에 둘 모두가 설치되어 있으면 웹 서버에 액세스할 때는 인터넷이 사용됩니다. 이 구성의 경우 보안이 확실하지 않습니다.
- CA SiteMinder for Secure Proxy Server 는 고유한 세션 저장소를 제공합니다. 그러나 세션 저장소에는 일반 세션 서버로 사용할 수 있는 공용 API 가 없습니다.

- CA SiteMinder for Secure Proxy Server 를 사용하는 일부 엔터프라이즈의 경우 대상 서버에 SiteMinder 웹 에이전트나 응용 프로그램 서버 에이전트를 설정했을 수도 있습니다. CA SiteMinder for Secure Proxy Server 에이전트에 대한 정책이 대상 서버에 설치된 에이전트에 대한 정책과 일치하지 않는 경우 CA SiteMinder for Secure Proxy Server 는 호출하는 클라이언트로 응답을 다시 전달할 수 있습니다. CA SiteMinder for Secure Proxy Server 는 이러한 정책을 처리할 때 불일치에 대한 경고를 제공하지 않으므로 이러한 환경에서 SiteMinder 정책을 설정할 때는 주의해야 합니다.
- CA SiteMinder for Secure Proxy Server 는 요청이 수신될 때마다 대상 서버로 새 요청을 보냅니다. 모든 캐싱 지시문은 무시됩니다.
- simple_url 세션 체계에서는 CA SiteMinder for Secure Proxy Server 가 JavaScript 에 포함된 URL 이나 JavaScript 에서 생성된 URL 을 다시 쓰지 않습니다.
- simple_url 세션 체계는 보호된 리소스에 포스트할 때 POST 데이터를 유지하지 않습니다.

# 엔터프라이즈의 CA SiteMinder® SPS

직원, 고객 및 파트너에게 네트워크 리소스에 대한 액세스를 제공하는 엔터프라이즈는 다음과 같은 여러 과제를 해결해야 합니다.

- 적절한 서비스로 요청 전달
- 사용자 아이덴티티 확인 및 권한 설정
- 권한이 있는 사용자의 세션 유지 관리
- 중앙 집중식 액세스 제어 구성 제공
- 여러 장치 유형 지원
- 유연하고 안전한 아키텍처 사용

SiteMinder 는 사용자 인증 및 권한 부여와 사용자 권한을 평가하기 위한 복잡한 엔진을 포함하여 이러한 다양한 과제에 대한 솔루션을 제공합니다. 또한 CA SiteMinder for Secure Proxy Server 는 리버스 프록시 솔루션을 제공하여 핵심 정책 서버 및 웹 에이전트 기능의 이점을 더욱 확대해 줍니다.

이 리버스 프록시 솔루션은 다음과 같은 기능을 추가합니다.

- 기존 SiteMinder 웹 에이전트와의 상호 운용성
- 쿠키를 사용하지 않는 싱글 사인온 및 세션 저장소
- 프록시 규칙을 통한 중앙 집중식 구성
- 다양한 세션 유지 관리 옵션
- 여러 장치 지원

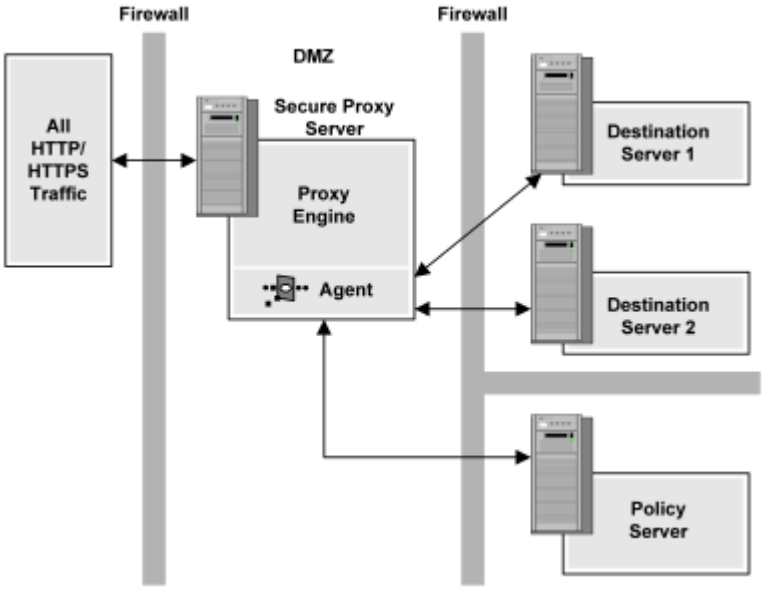
엔터프라이즈에 CA SiteMinder for Secure Proxy Server 를 배포하면 다음과 같은 기능을 제공할 수 있습니다.

- 중앙 집중식 액세스 제어 필터 역할
- 쿠키를 사용하지 않는 세션 체계 지원
- 엑스트라넷 액세스 제어 지원

## 중앙 집중식 액세스 제어 필터 역할을 하는 CA SiteMinder for Secure Proxy Server

대상 서버에 대한 액세스를 제한하고 네트워크에 대한 중앙 진입점을 제공하기 위해 CA SiteMinder for Secure Proxy Server 를 엔터프라이즈의 모든 대상 서버 앞에 배치할 수 있습니다. 엔터프라이즈로 들어오는 HTTP 또는 HTTPS 요청은 CA SiteMinder for Secure Proxy Server 를 통해 필터링된 후 이행을 위해 적절한 대상 서버로 전달될 수 있습니다.

다음 그림에서는 CA SiteMinder for Secure Proxy Server 가 모든 HTTP 및 HTTPS 요청을 처리하는 방식을 보여 줍니다.




콘텐츠가 포함된 대상 서버에는 SiteMinder 웹 에이전트가 필요하지 않습니다. 첫 번째 방화벽 뒤에 있는 네트워크 요소는 CA SiteMinder for Secure Proxy Server 뿐입니다. 모든 사용자는 두 번째 방화벽 뒤에 있는 SiteMinder 를 통해 인증되고 권한을 부여받아야 합니다. SiteMinder 와 CA SiteMinder for Secure Proxy Server 가 사용자 권한을 확인한 후에는 대상 서버가 콘텐츠를 제공합니다.

이 배포 환경의 이점은 다음과 같습니다.

- 프록시 규칙을 통해 구성 중앙 집중화

  CA SiteMinder for Secure Proxy Server 는 XML 구성 파일에 정의된 프록시 규칙을 사용하여 CA SiteMinder for Secure Proxy Server 가 요청을 처리할 방법을 설정합니다. 프록시 규칙은 다음을 기반으로 할 수 있습니다.

  - 호스트 이름
  - URI 하위 문자열
  - HTTP 헤더
  - SiteMinder 헤더
  - URI 를 기반으로 하는 정규식

  또한 프록시 규칙에 대한 조건을 중첩시켜 여러 조건을 통합한 규칙을 생성할 수도 있습니다.

- 적절한 서비스로 요청 전달

  모든 HTTP 및 HTTPS 트래픽은 CA SiteMinder for Secure Proxy Server 를 통해 전달됩니다. 요청은 CA SiteMinder for Secure Proxy Server 에 대해 설정된 프록시 규칙을 기반으로 적절한 대상 서버로 전달되어 이행됩니다.

- 사용자 아이덴티티 확인 및 권한 설정

  CA SiteMinder for Secure Proxy Server 는 기본 제공 웹 에이전트를 사용하여 SiteMinder 와 통신하고 요청에 대한 인증 및 권한 부여를 수행합니다.

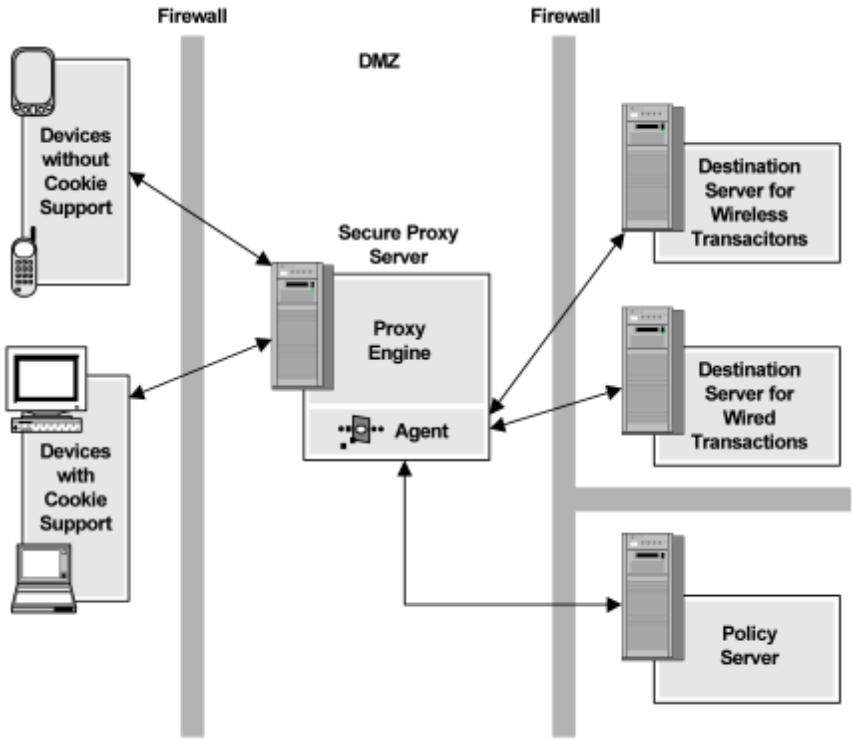
## 쿠키를 사용하지 않는 세션에 대한 CA SiteMinder® SPS 지원

대부분의 솔루션은 쿠키 기술을 사용합니다. 그러나 일부 엔터프라이즈는 HTTP 또는 HTTPS 를 통해 리소스에 액세스할 때 사용자 세션을 설정 및 유지 관리하기 위한 대체 방법을 필요로 하며 싱글 사인온에 쿠키를 사용하지 않는 솔루션을 제공합니다.

CA SiteMinder for Secure Proxy Server 는 메모리 내 세션 저장소를 제공하고 쿠키를 사용하지 않는 다음과 같은 세션 체계를 사용할 수 있게 해 줍니다.

- 미니 쿠키(표준 SiteMinder 쿠키 대신 작은 쿠키 사용)
- SSL ID
- 장치 ID
- 단순 URL 다시 쓰기
- IP 주소

다음 그림에서는 CA SiteMinder for Secure Proxy Server 가 쿠키를 사용하는 표준 세션과 쿠키를 사용하지 않는 세션을 함께 제공하는 배포 환경을 보여 줍니다.




위 그림의 배포 환경에는 다음과 같은 이점이 있습니다.

- 여러 장치 유형 지원

  CA SiteMinder for Secure Proxy Server 는 일련의 프록시 규칙을 통해 요청을 실행하는 장치의 유형에 따라 요청을 전달하거나 리디렉션합니다. 예를 들어 모든 초기 요청은 CA SiteMinder for Secure Proxy Server 에 전달된 후 장치 유형에 따라 대상 서버로 전달될 수 있습니다. 브라우저 요청은 대상 서버로 리디렉션될 수 있으며 CA SiteMinder for Secure Proxy Server 는 무선 요청을 처리합니다.

- 권한이 있는 사용자의 세션 유지 관리

  사용자 세션을 유지 관리하는 데는 표준 SiteMinder 쿠키 세션 체계와 쿠키를 사용하지 않는 세션 체계가 모두 사용됩니다. 세션 체계는 각 가상 호스트의 사용자 에이전트 유형에 따라 할당됩니다. 예를 들어 웹 브라우저를 통해 네트워크에 액세스하는 모든 사용자는 표준 쿠키 세션 체계에 할당됩니다. 무선 전화를 통해 리소스에 액세스하는 사용자는 장치 ID 세션 체계에 할당됩니다.

- 쿠키를 사용하지 않는 싱글 사인온 및 세션 저장소 제공

  CA SiteMinder for Secure Proxy Server 는 메모리 내 세션 저장소와 여러 세션 체계에 대한 지원을 통해 쿠키 기반 세션에 대한 대체 방법을 제공합니다. CA SiteMinder for Secure Proxy Server 는 세션 저장소에서 세션 정보를 유지 관리하고 토큰을 반환합니다. 이 토큰은 모든 트랜잭션과 교환되므로 CA SiteMinder for Secure Proxy Server 는 세션 저장소에서 캡처된 세션 정보와 토큰을 일치시킬 수 있습니다.

- 다양한 세션 유지 관리 옵션 제공

## 페더레이션 환경의 쿠키를 사용하지 않는 세션 체계

CA SiteMinder for Secure Proxy Server 는 쿠키를 사용하지 않는 세션 체계를 처리할 수 있는 기능이 기본적으로 포함되어 있으므로 무선 장치 등의 사용자 에이전트가 기존 SiteMinder 쿠키를 지원하지 않는 환경에도 배포될 수 있습니다.

SiteMinder FSS(Federation Security Services) 환경에 CA SiteMinder for Secure Proxy Server 를 배포할 경우 사용자 요청이 수신되면 다음 프로세스가 적용됩니다.

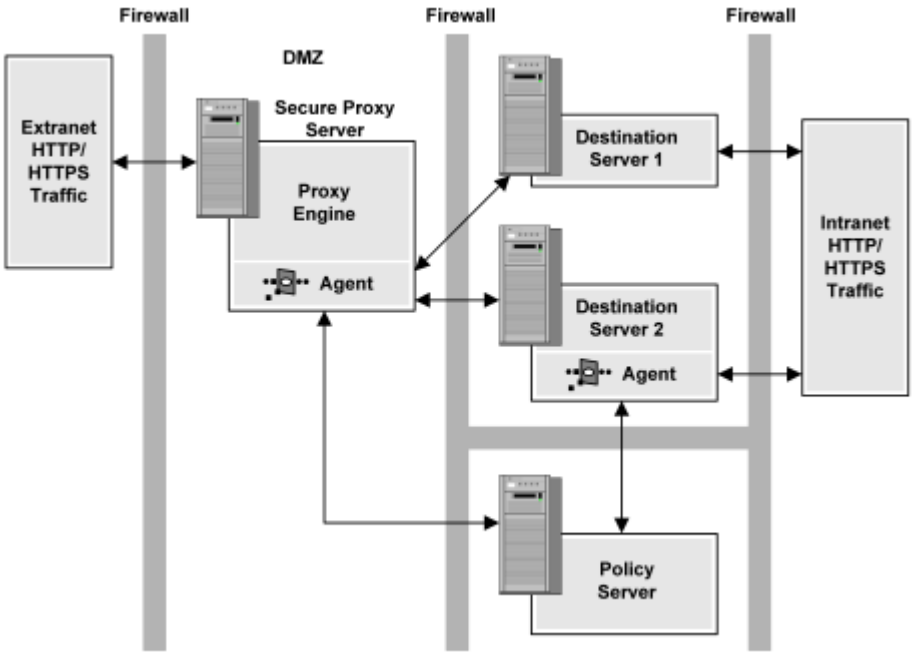
1. CA SiteMinder for Secure Proxy Server 가 페더레이션된 리소스에 대한 요청을 수신합니다. 요청이 어설션 생산 사이트의 FWS(페더레이션 웹 서비스) 응용 프로그램으로 리디렉션됩니다.
2. CA SiteMinder for Secure Proxy Server 가 리디렉션을 요청하는 가상 호스트에 쿠키를 사용하지 않는 페더레이션이 사용되는지 여부를 확인합니다.
3. 쿠키를 사용하지 않는 체계가 사용되는 경우 CA SiteMinder for Secure Proxy Server 가 현재 세션의 세션 키(SMSESSION 쿠키)를 제거합니다.
4. CA SiteMinder for Secure Proxy Server 가 FWS 리디렉션을 통해 제공된 링크로 사용자를 보냅니다.

CA SiteMinder for Secure Proxy Server 가 simple_url 세션 체계와 같이 다시 쓰기 가능한 세션 체계를 사용하는 경우 CA SiteMinder for Secure Proxy Server 가 리디렉션된 URL 에 세션 키 정보를 포함하도록 리디렉션 응답을 다시 씁니다.

# 엑스트라넷 액세스 제어에 대한 CA SiteMinder® SPS 지원

또 다른 CA SiteMinder for Secure Proxy Server 배포 방식의 경우 외부 사용자에 대해서는 액세스 제어를 제공하지만 내부 사용자에 대해서는 대상 서버에 대한 직접 액세스를 허용합니다. 대상 서버가 엔터프라이즈 내의 개인에게 보안 응용 프로그램에 대한 액세스를 제공하는 경우 액세스 제어를 제공하기 위해 대상 서버에 표준 SiteMinder 웹 에이전트를 설치할 수 있습니다. CA SiteMinder for Secure Proxy Server 를 통해 올바르게 인증된 사용자는 싱글 사인온을 사용할 수 있습니다.

다음 그림에서는 엑스트라넷 네트워크 배포의 예를 보여 줍니다.




이 배포 환경의 이점은 다음과 같습니다.

- 엑스트라넷 소스에서의 요청 전달

  사용자가 요청한 리소스에 대해 인증되고 권한이 부여된 후 모든 엑스트라넷 트래픽은 CA SiteMinder for Secure Proxy Server 를 통해 적절한 대상 서버로 전달됩니다.

- 유연한 아키텍처 사용

  모든 정보는 엑스트라넷 공격으로부터 보호하기 위해 여러 방화벽 뒤에 배치됩니다. 인트라넷 사용자에 적절한 정보는 SiteMinder 통신에 에이전트 오버헤드를 발생시키지 않습니다. 그러나 SiteMinder 는 여전히 중요한 리소스를 보호할 수 있습니다.

- 웹 에이전트 간 상호 운용성 제공

  CA SiteMinder for Secure Proxy Server 웹 에이전트와 인트라넷 웹 에이전트는 동일한 정책 서버를 사용하며 모든 대상 서버에서 권한이 있는 엑스트라넷 사용자에게 싱글 사인온을 제공합니다.

# 제 2 장: SPS 설치, 업그레이드 및 구성

이 섹션은 다음 항목을 포함하고 있습니다.

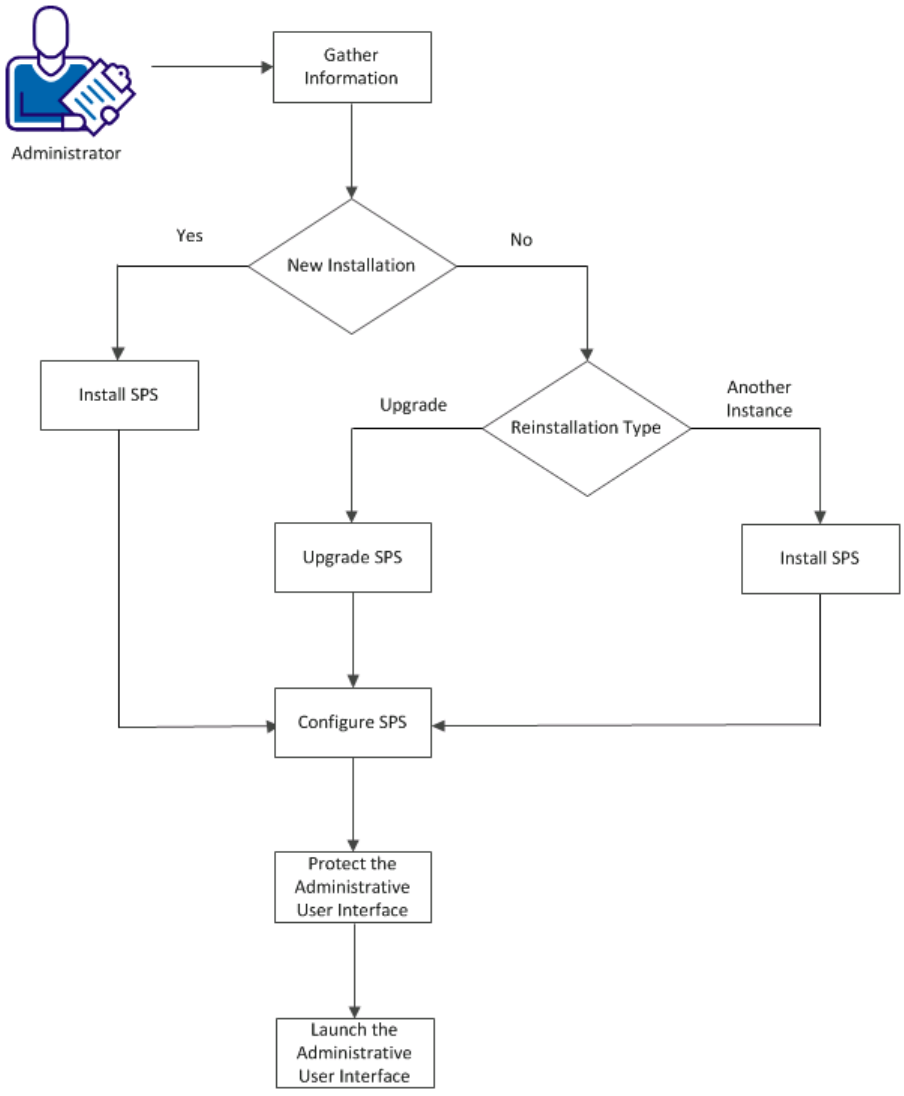


## 수동 SPS 설치, 업그레이드 및 구성

CA SiteMinder Secure Proxy Server 는 액세스 제어를 위한 프록시 기반 솔루션을 제공하는 독립 실행형 서버입니다. CA SiteMinder for Secure Proxy Server 는 엔터프라이즈용 네트워크 게이트웨이를 제공하는 프록시 엔진을 사용하며, 기존 쿠키 기반 기술을 사용하지 않는 여러 세션 체계를 지원합니다.

다음 다이어그램에서는 CA SiteMinder for Secure Proxy Server 를 설치 및 구성하는 방법에 대해 설명합니다.

**Install, Upgrade, and Configure SPS**



## 사전 요구 사항

CA SiteMinder for Secure Proxy Server 의 설치 또는 업그레이드 전에 다음 사전 요구 사항을 확인하십시오.

- CA SiteMinder for Secure Proxy Server 는 정책 서버가 설치된 컴퓨터에 설치해서는 안 됩니다.
- Linux 의 경우 /opt/etc/CA 디렉터리에 쓰기 권한이 있는지 확인하십시오.
- Linux 의 경우 CA SiteMinder for Secure Proxy Server 를 설치하는 폴더에 충분한 권한(755)이 있어야 합니다. /root 폴더는 기본 권한(750)이 충분하지 않으므로 이 폴더에는 CA SiteMinder for Secure Proxy Server 를 설치하지 마십시오.
- Solaris 의 경우 CA SiteMinder for Secure Proxy Server 는 "nobody" 사용자로 실행됩니다. CA SiteMinder for Secure Proxy Server 를 이 사용자로 실행하지 않으려면 대체 사용자를 생성하고 필요한 권한을 할당하십시오.
- RHEL 에 CA SiteMinder for Secure Proxy Server 를 설치하는 경우 ncurses 패키지를 설치했는지 확인하십시오.
- CA SiteMinder for Secure Proxy Server 를 RHEL 5 또는 RHEL 6 64 비트 컴퓨터에 설치하는 경우 컴퓨터에 다음 라이브러리가 설치되어 있는지 확인하십시오.
  - libstdc++.so.6
  - libexpat.so.0
  - libuuid.so.01
  - libkeyutils.so.1

  **참고**: 이러한 라이브러리는 64 비트 바이너리가 아니라 32 비트 바이너리여야 합니다.
- CA SiteMinder for Secure Proxy Server 를 RHEL 5.5 컴퓨터에 설치하는 경우 컴퓨터에 Legacy Software Development 패키지가 설치되어 있는지 확인하십시오.
- Linux 에서 CA SiteMinder for Secure Proxy Server 를 설치 또는 업그레이드하는 경우 keyutils-libs-1.4-4.el6.i686.rpm 패키지를 설치했는지 확인하십시오.

- **JCE** - Java 암호화 알고리즘을 사용하려면 최신 JCE(Java Cryptography Extension) Unlimited Strength Jurisdiction 패치가 필요합니다. 운영 플랫폼에 맞는 JCE 패키지를 찾으려면 Oracle 웹 사이트를 방문하십시오.

  패치를 시스템의 다음 파일에 적용하십시오.

  - local_policy.jar
  - US_export_policy.jar

  이러한 파일은 다음 디렉터리에 있습니다.

  **Windows:** *jre_home*\lib\security

  **UNIX:** *jre_home*/lib/security

  *jre_home*

  이 변수는 Java Runtime Environment 설치 위치를 지정합니다.

## 설치 워크시트

CA SiteMinder for Secure Proxy Server 구성 마법사에서 트러스트된 호스트를 등록할 때는 필요한 정보를 묻는 일련의 메시지가 표시됩니다. 트러스트된 호스트는 SiteMinder 웹 에이전트를 하나 이상 설치할 수 있는 클라이언트 컴퓨터입니다. 트러스트된 호스트와 정책 서버 간의 연결을 설정하려면 정책 서버에 호스트를 등록하십시오. 등록이 완료되면 등록 도구가 SmHost.conf 파일을 생성합니다. 이 파일이 생성되면 클라이언트 컴퓨터가 트러스트된 호스트가 됩니다.

CA SiteMinder for Secure Proxy Server 를 설치, 업그레이드 또는 구성하기 전에 호스트 등록, 포함된 Apache 웹 서버 및 Tomcat 서버에 필요한 다음 정보를 수집했는지 확인하십시오.

| 매개 변수 | 설명 |
| --- | --- |
| SiteMinder 관리자 이름 | 정책 서버에 이미 정의되어 있는 이름과 일치하는 관리자의 이름입니다. 이 관리자는 트러스트된 호스트를 만들 권한이 있어야 합니다. |
| SiteMinder 관리자 암호 | 트러스트된 호스트를 등록할 수 있는 권한이 있는 SiteMinder 관리자의 암호입니다. 정책 서버에서 사용되는 암호와 일치해야 합니다. |

| 매개 변수 | 설명 |
| --- | --- |
| 트러스트된 호스트 이름 | 설치 중 할당된 트러스트된 호스트의 이름입니다. |
| 호스트 구성 개체 | 관리 UI 에 이미 정의되어 있는 호스트 구성 개체의 이름입니다. |
| 에이전트 구성 개체 | 관리 UI 에 정의되어 있는 기존 에이전트 구성 개체의 이름입니다. |
| 호스트를 등록할 정책 서버의 IP 주소 | **이름**: SiteMinder 가 방화벽 뒤에 있는 경우 포트 번호를 포함하십시오. 예: 111.12.12.2:12 |
| 에이전트 이름 | 기본 에이전트 또는 ACO 에 정의된 에이전트의 이름입니다. |
| 마스터 키 | 고급 인증 서버를 위한 마스트 암호화 키를 식별합니다. 정책 서버에 구성한 것과 동일한 값을 입력하십시오. |
| 호스트 구성 파일 이름 및 위치 | 웹 에이전트 및 사용자 지정 에이전트가 트러스트된 호스트 대신 작업을 수행하는 데 사용하는 SmHost.conf 파일을 식별합니다. 마법사에는 기본 위치가 표시됩니다. |
| 웹 에이전트 구성 파일의 이름 및 위치 | 마법사에는 기본 위치가 표시됩니다. |
| Apache 웹 서버 관리자의 전자 메일 주소 | 관리자의 전자 메일 주소입니다. 기본값: admin@company.com |
| 서버의 정규화된 호스트 이름 | *computer_name.company.com* 형식의 정규화된 이름입니다. |
| Apache HTTP 요청에 사용되는 포트 번호 | Apache 의 HTTP 요청을 수신 대기하는 포트입니다. 기본값: 80 |
| Apache SSL 요청에 사용되는 포트 번호 | Apache 의 SSL 요청을 수신 대기하는 포트입니다.<br>기본값: 443 |
| Tomcat HTTP 요청에 사용되는 포트 번호 | Tomcat 의 HTTP 요청을 수신 대기하는 포트입니다.<br>기본값: 8080 |

| 매개 변수 | 설명 |
| --- | --- |
| Tomcat SSL 요청에 사용되는 포트 번호 | Tomcat 의 SSL 요청을 수신 대기하는 포트입니다.<br>기본값: 543 |
| Tomcat 종료 요청에 사용되는 포트 번호 | Tomcat 의 종료 요청을 수신 대기하는 포트입니다.<br>기본값: 8005 |
| AJP 의 포트 번호 | AJP 의 포트 번호입니다.<br>기본값: 8009 |

## CA SiteMinder® SPS 설치

CA SiteMinder for Secure Proxy Server 를 설치하기 전에 CA SiteMinder for Secure Proxy Server 를 설치하는 데 필요한 정보를 수집했는지 확인하십시오.

### Windows 에 CA SiteMinder® SPS 설치

**다음 단계를 수행하십시오.**

1. CA Support 사이트의 다운로드 위치에서 설치 프로그램을 복사합니다.
2. 실행 파일을 마우스 오른쪽 단추로 클릭하고 "관리자 권한으로 실행"을 선택합니다.
3. ca-proxy-<*version*>-<*operating_system*>.exe 를 두 번 클릭합니다.

   설치 프로그램이 시작됩니다.
4. 설치 마법사의 지시를 따릅니다.

   **참고**: 기본적으로 CA SiteMinder for Secure Proxy Server 는 처음 설치 시 인스턴스 이름을 **기본값**으로 설정합니다. 이 기본값을 수정할 수 없으며 다른 CA SiteMinder for Secure Proxy Server 인스턴스에 이 이름을 사용할 수도 없습니다.
5. 설치를 완료한 후 시스템을 다시 시작합니다.

### Linux 또는 Solaris 에 CA SiteMinder® SPS 설치

CA SiteMinder for Secure Proxy Server 는 Linux 및 Solaris 에 설치 가능합니다.

**다음 단계를 수행하십시오.**

1. CA Support 사이트의 다운로드 위치에서 다음 프로그램 중 하나를 임시 디렉터리에 복사합니다.

   Solaris: ca-proxy-12.5-sol.bin

   Linux: ca-proxy-12.5-rhel30.bin

2. 다음 명령 중 *하나*를 입력합니다.

   sh ./ca-proxy-12.5-sol.bin

   sh ./ca-proxy-12.5-rhel30.bin

3. 설치 마법사가 제공하는 화면 프롬프트를 따릅니다.

### CA SiteMinder® SPS 설치 확인

InstallLog 파일을 검사하여 CA SiteMinder for Secure Proxy Server 가 성공적으로 설치되었는지 확인할 수 있습니다. 모든 플랫폼에서 InstallLog 는 기본적으로 다음 위치에 설치됩니다.

*sps_home*\install_config_info\CA_SiteMinder_Secure_Proxy_Server_InstallLog.log

## 여러 CA SiteMinder® SPS 인스턴스 설치

여러 CA SiteMinder for Secure Proxy Server 인스턴스를 같은 컴퓨터에 설치할 수 있습니다. 각 CA SiteMinder for Secure Proxy Server 인스턴스는 고유한 인스턴스 이름과 통신 포트를 사용하며 개별 디렉터리 구조를 생성합니다.

**다음 단계를 수행하십시오.**

1. ca-proxy-<*version*>-<*operating_system*>.exe 를 두 번 클릭합니다.

   설치 프로그램이 시작됩니다.

2. 새 인스턴스를 설치하는 옵션을 선택합니다.
3. 설치 마법사의 지시를 따릅니다.

   **참고**: 인스턴스 이름과 통신에 사용되는 포트를 에 고유한 값으로 입력했는지 확인하십시오.

## CA SiteMinder® SPS 업그레이드

설치 프로그램을 실행하여 이전 버전의 CA SiteMinder for Secure Proxy Server 를 현재 버전으로 업그레이드할 수 있습니다.

**참고**: 필터를 구성하고 세션 체계를 사용자 지정한 경우 업그레이드 전에 Tomcat/ 경로의 lib 디렉터리를 백업하십시오.

**다음 단계를 수행하십시오.**

1. ca-proxy-<*version*>-<*operating_system*>.exe 를 두 번 클릭합니다.

   설치 프로그램이 시작됩니다.
2. "확인"을 클릭하여 CA SiteMinder for Secure Proxy Server 버전을 업그레이드합니다.
3. 설치 마법사의 지시를 따릅니다.
4. 설치를 완료한 후 시스템을 다시 시작합니다.

### 업그레이드를 위한 추가 태스크

설치 프로세스의 마지막에 몇 가지 추가 단계를 수행하여 업그레이드를 지원할 수 있습니다. CA SiteMinder for Secure Proxy Server 배포 환경에서 사용자 지정된 항목의 양에 따라 다음 태스크 중 하나 이상을 수행할 수 있습니다.

- ssl.conf 파일 및 server.conf 파일 내의 SSL 구성 경로가 사용 환경에 올바르게 지정되었는지 확인하십시오. 업그레이드 중 자동화된 단계에서는 모든 인증서가 기본 위치에 있는 것으로 가정합니다.
- Linux 에서 SSL 을 사용하도록 설정하고 CA SiteMinder for Secure Proxy Server 를 이전 릴리스에서 CA SiteMinder for Secure Proxy Server 12.5 로 업그레이드한 경우 다음 명령을 실행하여 Apache 를 SSL 모드에서 시작하십시오.

  ```
  <install-path>/secure-proxy/proxy-engine/sps-ctl startssl
  ```
- 인증서, 인증 기관 및 키가 모두 *sps_home*\secure-proxy\SSL 의 해당 폴더에 올바르게 복사되었는지 확인하십시오.
- 경로를 proxyrules.xml 파일의 프록시 규칙 DTD 파일로 수정하십시오. DTD 파일의 기본 경로는 *sps_home*\proxy-engine\conf\dtd\proxyrules.dtd 입니다.

## JVM 매개 변수 사용자 지정

다음 파일에서 JVM(Java Virtual Machine) 매개 변수를 사용자 지정할 수 있습니다.

- Windows 의 경우 *sps_home*\proxy-engine\conf 디렉터리에 있는 SmSpsProxyEngine.properties 파일을 수정하십시오.
- UNIX 의 경우 *sps_home*/proxy-engine 디렉터리에 있는 proxyserver.sh 파일을 수정하십시오.

## CA SiteMinder® SPS 구성

CA SiteMinder for Secure Proxy Server 를 설치한 후 구성 마법사를 실행하십시오. 구성 마법사를 사용하여 포함된 SiteMinder 웹 에이전트에 대한 트러스트된 호스트를 등록하고 포함된 Apache 웹 서버에 대한 일부 관리 태스크를 수행할 수 있습니다.

**중요!** 마법사를 실행하기 전에 호스트를 등록할 정책 서버에 필요한 개체를 설정했는지 확인하십시오. 이러한 개체가 구성되어 있지 않으면 트러스트된 호스트 등록이 실패합니다.

**다음 단계를 수행하십시오.**

1. 콘솔 창을 열고 sps_home/secure-proxy 디렉터리로 이동합니다.
2. 다음 명령 중 *하나*를 입력합니다.

   ```
   Windows: ca-sps-config.exe

   UNIX: ca-sps-config.sh
   ```

   구성 마법사가 시작됩니다.
3. CA SiteMinder for Secure Proxy Server 를 구성할 정책 서버의 버전을 선택합니다.
4. 호스트 등록을 즉시 수행하는 옵션을 선택합니다.
5. (선택 사항) 공유 암호 롤오버를 사용하도록 설정하는 옵션을 선택합니다.
6. 다음 단계를 수행하여 트러스트된 호스트를 등록합니다.
   a. SiteMinder 관리자의 이름과 암호를 지정합니다.

      **참고**: 입력하는 정보는 트러스트된 호스트를 등록할 정책 서버에 이미 정의되어 있어야 합니다.

   b. 트러스트된 호스트 및 호스트 구성 개체의 이름을 지정합니다.

      **참고**: 트러스트된 호스트에 대해 입력하는 이름은 고유해야 합니다. 호스트 구성 개체의 이름은 트러스트된 호스트를 등록할 정책 서버에 이미 정의되어 있어야 합니다.

   c. 트러스트된 호스트를 등록할 정책 서버의 IP 주소를 입력합니다.

   d. FIPS 모드를 선택합니다.

   e. 호스트 구성 파일 SmHost.conf 의 이름과 위치를 지정합니다. 마법사에는 기본 위치가 표시됩니다.

   f. 에이전트 구성 개체의 이름을 지정합니다.

**참고**: 입력하는 에이전트 구성 개체는 트러스트된 호스트를 등록할 정책 서버에 이미 정의되어 있어야 합니다.

7. Apache 웹 서버에 대한 다음 정보를 입력합니다.
   - 서버 이름
   - 웹 서버 관리자의 전자 메일 주소
   - HTTP 포트 번호
   - SSL 포트 번호
8. Tomcat 서버에 대한 다음 정보를 입력합니다.
   - HTTP 포트 번호
   - SSL 포트
   - 종료 포트 번호
   - AJP 포트 번호

   **참고**: Solaris 또는 Linux 를 실행하는 시스템에 설치하는 경우 Tomcat 및 Apache 를 실행할 수 있는 사용자의 이름을 묻는 추가 화면이 표시됩니다. 이 사용자는 루트일 수 없으며 해당 사용자 계정을 수동으로 생성해야 합니다. 설치 프로그램은 이 사용자 계정을 자동으로 생성하지 않습니다. Tomcat 사용자에게는 로그 디렉터리에 대한 모든 권한(rwa)이 있어야 합니다.
9. 웹 에이전트를 사용하도록 설정하려면 "예"를 선택합니다.
10. CA SiteMinder for Secure Proxy Server 가 페더레이션 게이트웨이로 작동하도록 하려면 "예"를 선택합니다.
11. 구성 요약을 검토합니다.
12. "Install"(설치)을 클릭합니다.

    CA SiteMinder for Secure Proxy Server 가 구성되고 구성 파일이 설치됩니다.
13. "Done"(완료)을 클릭하여 마법사를 종료합니다.
14. SiteMinder 보안 프록시 및 SiteMinder 프록시 엔진 서비스를 시작합니다.

**참고**: 구성 마법사를 다시 실행할 경우 SSL 을 다시 초기화해야 합니다.

## SPS 에 대한 추가 구성

CA SiteMinder for Secure Proxy Server 를 설치하고 구성 마법사를 실행한 후 사용 환경에 맞게 CA SiteMinder for Secure Proxy Server 구성을 수정할 수 있습니다. 다음 구성 파일에는 CA SiteMinder for Secure Proxy Server 에 영향을 주는 설정이 포함되어 있습니다.

**httpd.conf**

Apache 웹 서버에 대한 설정이 포함되어 있습니다.

**server.conf**

가상 호스트 및 세션 체계 매핑을 비롯한 CA SiteMinder for Secure Proxy Server 동작을 결정하는 설정이 포함되어 있습니다.

**logger.properties**

CA SiteMinder for Secure Proxy Server 로깅 동작을 결정하는 설정이 포함되어 있습니다.

**proxyrules.xml**

CA SiteMinder for Secure Proxy Server 가 들어오는 요청을 처리하는 방식을 결정하는 규칙이 포함되어 있습니다.

**추가 정보:**

프록시 규칙 구성 (페이지 165)
Apache 웹 서버 구성 (페이지 93)

## 세션 보증 관리

기본적으로 CA SiteMinder for Secure Proxy Server 는 세션 보증을 활성화합니다. 이 기능을 비활성화하려면 다음 단계를 수행하십시오.

1. server.conf 파일을 엽니다.
2. <Context name="AALoginService"> 섹션으로 이동하여 활성화 값을 no 로 설정합니다.
3. <Context name="Advanced Auth Application"> 섹션으로 이동하여 활성화 값을 no 로 설정합니다.
4. <Context name="UI Application"> 섹션으로 이동하여 활성화 값을 no 로 설정합니다.
5. 변경 내용을 저장합니다.

### SiteMinder 양식의 기본 위치 수정

CA SiteMinder for Secure Proxy Server v6.0 부터 SiteMinder 양식의 기본 위치는 더 이상 /siteminderagent/forms 가 아닙니다. 계속 이 위치를 사용하여 양식을 제공하려면 CA SiteMinder for Secure Proxy Server 양식 위치를 수정하십시오.

**다음 단계를 수행하십시오.**

1. 다음 위치에 siteminderagent 디렉터리를 생성합니다.

   *sps_home*/proxy-engine/examples/siteminderagent

2. 다음 디렉터리의 forms 폴더를 복사합니다.

   *sps_home*/proxy-engine/examples

   복사한 폴더를 다음 디렉터리에 붙여 넣습니다.

   *sps_home*/proxy-engine/examples/siteminderagent

   양식이 sps_home/proxy-engine/examples/siteminderagent/forms 에 복사됩니다.

   **참고**: 양식 폴더의 위치를 사용자 지정하는 경우에는 httpd.conf 파일을 양식 이미지의 위치로 업데이트해야 합니다.

## 관리 사용자 인터페이스 보호

기본적으로 설치 관리자는 관리 사용자 인터페이스를 보고하기 위한 보호 정책을 만듭니다. 설치 관리자는 정의된 에이전트 이름을 사용하여 다음과 같은 정보로 보호 정책을 만듭니다.

- 도메인: DOMAIN-SPSPADMINUI
- 정책: POLICY-SPSADMINUI

이 보호 정책은 사용자 디렉터리 정보를 수록하지 않습니다. 다음 단계를 수행하여 관리 사용자 인터페이스에 로그인합니다.

1. 사용자 디렉터리 정보를 사용하여 DOMAIN-SPSPADMINUI 를 업데이트합니다.
2. 사용자 정보를 사용하여 POLICY-SPSADMINUI 를 업데이트합니다.

## 관리 사용자 인터페이스 시작

프록시 엔진 서비스를 시작한 후 관리 사용자 인터페이스를 시작할 수 있습니다. URL 을 시작하려면 웹 브라우저에 다음 URL 을 입력하십시오.
`http://fullyqualifiedhostname:Tomcat_port/proxyui/`

CA SiteMinder for Secure Proxy Server 가 설치 또는 업그레이드되고 구성됩니다.

처음 설치 후 자동 설치 및 구성을 수행하려면 자동 설치 및 구성을 참조하십시오. CA SiteMinder for Secure Proxy Server 를 제거하려면 CA SiteMinder for Secure Proxy Server 제거를 참조하십시오. CA SiteMinder for Secure Proxy Server 를 다양한 모드에서 시작하려면 단일 또는 다중 프로세스 모드에서 CA SiteMinder for Secure Proxy Server 시작을 참조하십시오. SiteMinder 양식의 기본 위치를 수정하려면 SiteMinder 양식의 기본 위치 수정을 참조하십시오.

# SPS 자동 설치 및 구성

CA SiteMinder for Secure Proxy Server 를 처음 설치하고 구성한 후 나중에 무인 모드로 다시 설치하거나 저장된 구성 데이터를 사용하여 다른 CA SiteMinder for Secure Proxy Server 인스턴스를 무인 모드로 설치할 수 있습니다.

설치 후 CA SiteMinder for Secure Proxy Server 는 sps-home/install_config_info 폴더에 샘플 속성 파일을 생성합니다. 구성 후에는 동일한 속성 파일이 구성에 사용된 추가 속성으로 업데이트됩니다. 이 속성 파일은 이후에 사용자 지정된 값으로 자동 설치 및 구성을 수행하는 데 사용됩니다.

**다음 단계를 수행하십시오.**

1. 명령 창을 엽니다.
2. 속성 파일을 설치한 폴더로 이동합니다. 기본값은 sps-home/install_config_info 입니다.
3. 명령 프롬프트를 엽니다.
4. 다음 단계 중 하나 또는 둘 모두를 수행합니다.
   a. 자동 설치를 수행하려면 다음 명령을 실행합니다.

      ```
      ca-proxy-12.5-operating_system -i silent -f ca-sps-installer.properties
      ```

      **operating_system**

      운영 체제(win32, sol 또는 rhel30)를 정의합니다.

   b. 자동 구성을 수행하려면 다음 명령을 실행합니다.

      ```
      ca-sps-config -i silent -f ca-sps-installer.properties
      ```

   추가 사용자 개입 없이 설치 또는 구성이 진행됩니다.

# CA SiteMinder for Secure Proxy Server 제거

Windows 에서 제거하려면 다음 단계를 수행하십시오.

1. 명령 프롬프트를 열고 루트 설치 디렉터리로 이동합니다.
2. 제거할 각 인스턴스에 대해 다음 명령을 실행합니다.

```
ca-sps-uninstall.cmd
```

UNIX 에서 제거하려면 다음 단계를 수행하십시오.

1. 콘솔 창을 열고 루트 설치 디렉터리로 이동합니다.
2. 다음 프로그램을 실행합니다.

```
./ca-sps-uninstall.sh
```

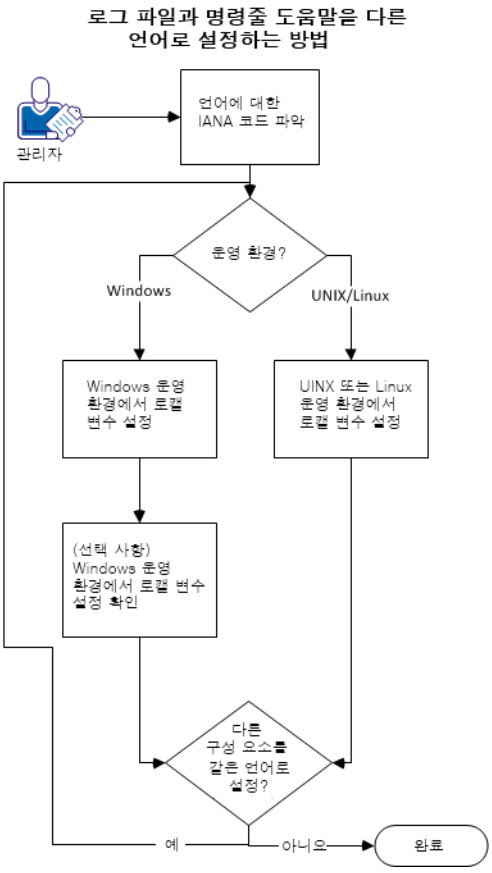
**참고**: server.conf 와 같은 임의 파일을 수정한 경우 해당 파일이나 부모 폴더는 자동으로 제거되지 않습니다. 파일 및 폴더를 수동으로 삭제해야 합니다.

# 다른 언어로 로그 파일 및 명령줄 도움말 설정

다음 구성 요소는 다른 언어로 로그 파일 및 명령줄 도움말을 설정할 수 있습니다.

- 정책 서버
- 웹 에이전트
- 보고서 서버
- CA SiteMinder Agent for SharePoint
- CA SiteMinder for Secure Proxy Server
- [set AGENT value for your book]
- CA SiteMinder® SDK 로 만든 모든 사용자 지정 소프트웨어

다음 그래프는 다른 언어로 로그 파일 및 명령줄 도움말을 설정하기 위한 워크플로를 설명합니다.




**다음 단계를 수행하십시오.**

1. 언어의 IANA 코드를 파악합니다 (페이지 48).
2. 다음 절차 중 하나를 사용하여 운영 환경에 대한 환경 변수를 만듭니다.
   - Windows 운영 환경에서 로캘 변수를 설정합니다 (페이지 50).
   - UNIX 또는 Linux 운영 환경에서 로캘 변수를 설정합니다 (페이지 52).
3. (선택 사항) Windows 운영 환경에서 로캘 변수 설정을 확인합니다 (페이지 51).
4. (선택 사항) 1 - 3 단계를 반복하여 환경의 모든 다른 구성 요소를 동일한 언어로 설정합니다.

## 언어의 IANA 코드 파악

각 언어에는 고유 코드가 있습니다. IANA(Internet Assigned Numbers Authority)는 이러한 언어 코드를 할당합니다. 언어 코드를 로캘 변수에 추가하면 소프트웨어가 표시하는 언어가 변경됩니다. 로캘 변수를 만들기 전에 원하는 언어에 대한 올바른 코드를 파악하십시오.

다음 표에는 이 소프트웨어에서 지원되는 언어에 해당되는 IANA 코드가 수록되어 있습니다.

| 언어 | IANA 코드 |
|---|---|
| 포르투갈어(브라질) | pt_BR |
| 프랑스어 | fr |
| 독일어 | de |
| 이탈리아어 | it |
| 일본어 | ja |
| 한국어 | ko |
| 중국어 간체 | zh-Hans |
| 스페인어 | es |

**참고**: IANA 언어 코드의 목록은 이 타사 웹 사이트에서 볼 수 있습니다.

## 환경 변수

환경 변수는 사용자가 자신의 필요에 맞게 컴퓨터를 사용자 지정하기 위해 사용할 수 있는 설정입니다. 환경 변수의 예로는 다음과 같은 항목이 포함됩니다.

- 다운로드된 파일을 검색 또는 저장하기 위한 기본 디렉터리
- 사용자 이름
- 실행 파일을 검색하기 위한 위치의 목록(경로)

Windows 운영 환경에서는 컴퓨터의 모든 사용자에게 적용되는 글로벌 환경 변수를 사용할 수 있습니다. UNIX 또는 Linux 운영 환경의 환경 변수는 각 사용자 또는 프로그램에 대해 설정되어야 합니다.

로캘 변수를 설정하려면 다음 목록에서 운영 환경에 대한 절차를 선택하십시오.

- Windows 운영 환경에서 로캘 변수를 설정합니다 (페이지 50).
- UNIX 또는 Linux 운영 환경에서 로캘 변수를 설정합니다 (페이지 52).

## Windows 운영 환경에서 로캘 변수 설정

다음 로캘 변수는 소프트웨어에 대한 언어 설정을 지정합니다.

SM_ADMIN_LOCALE

이 변수를 만들고 원하는 언어로 설정하십시오. 다른 언어를 사용할 *각* 구성 요소에서 이 변수를 설정하십시오. 예를 들어, 정책 서버와 에이전트를 프랑스어로 설정하려고 한다고 가정합니다. 이 경우, 이러한 두 구성 요소에서 이 변수를 프랑스어로 설정하십시오.

**참고**: 설치 관리자 또는 구성 프로그램은 이 변수를 설정하지 *않습니다.*

**다음 단계를 수행하십시오.**

1. "시작", "제어판", "시스템", "고급 시스템 설정"을 클릭합니다.

   "시스템 속성" 대화 상자가 나타납니다.

2. "고급" 탭을 클릭합니다.
3. "환경 변수"를 클릭합니다.
4. "시스템 변수" 섹션으로 이동하여 "새로 만들기"를 클릭합니다.

   "새 시스템 변수" 대화 상자가 열리고 그 안의 "변수 이름:" 필드에 커서가 위치합니다.

5. 다음 텍스트를 입력합니다.

   SM_ADMIN_LOCALE

6. "변수 이름:" 필드를 클릭한 다음 원하는 IANA 언어 코드 (페이지 48)를 입력합니다.
7. "확인"을 클릭합니다.

   "새 시스템 변수" 대화 상자가 닫히고 목록에서 SM_ADMIN_LOCALE 변수가 표시됩니다.

8. "확인"을 *두 번* 클릭합니다.

   로캘 변수가 설정되었습니다.

9. (선택 사항) 1 - 8 단계를 반복하여 동일한 언어로 다른 구성 요소를 설정합니다.

## Windows 운영 환경에서 로캘 변수 값 확인

로캘 변수가 설정된 값을 언제든 확인할 수 있습니다. 이 절차는 변수를 설치한 후 올바로 설정되었는지 확인하기 위해 수행할 수 있습니다.

**참고**: UNIX 및 Linux 에서 변수 값을 확인하는 방법은 설정 절차 (페이지 52)에 설명되어 있습니다.

**다음 단계를 수행하십시오.**

1. 다음 단계를 사용하여 명령줄 창을 엽니다.
   a. "시작", "실행"을 차례로 클릭합니다.
   b. 다음 명령을 입력합니다.

      ```
      cmd
      ```

   c. "확인"을 클릭합니다.

      명령줄 창이 열립니다.
2. 다음 명령을 입력합니다.

   ```
   echo %SM_ADMIN_LOCALE%
   ```

   다음 줄에 로캘이 표시됩니다. 예를 들어, 언어가 독일어로 설정된 경우 다음 코드가 표시됩니다.

   ```
   de
   ```

   로캘 변수의 값이 확인되었습니다.

## UNIX 또는 Linux 운영 환경에서 로캘 변수 설정

다음 로캘 변수는 소프트웨어에 대한 언어 설정을 지정합니다.

SM_ADMIN_LOCALE

이 변수를 만들고 원하는 언어로 설정하십시오. 다른 언어를 사용할 *각* 구성 요소에서 이 변수를 설정하십시오. 예를 들어, 정책 서버와 에이전트를 프랑스어로 설정하려고 한다고 가정합니다. 이 경우, 이러한 두 구성 요소에서 이 변수를 프랑스어로 설정하십시오.

**참고**: 설치 관리자 또는 구성 프로그램은 이 변수를 설정하지 *않습니다.*

**다음 단계를 수행하십시오.**

1. 원하는 구성 요소를 실행하는 컴퓨터에 로그인합니다.
2. 콘솔(명령줄) 창을 엽니다.
3. 다음 명령을 입력합니다.

   export SM_ADMIN_LOCALE=*IANA_language_code*

   다음 예의 명령은 언어를 프랑스어로 설정합니다.

   export SM_ADMIN_LOCALE=fr

   로캘 변수가 설정되었습니다.
4. (선택 사항) 다음 명령을 입력하여 로캘 변수가 올바로 설정되었는지 확인합니다.

   echo $SM_ADMIN_LOCALE

   다음 줄에 로캘이 표시됩니다. 예를 들어, 언어가 독일어로 설정된 경우 다음 코드가 표시됩니다.

   de
5. (선택 사항) 1 - 4 단계를 반복하여 동일한 언어로 다른 구성 요소를 설정합니다.

# 제 3 장: FIPS-140 지원

이 섹션은 다음 항목을 포함하고 있습니다.



## FIPS 지원 개요

보안 프록시 서버는 FIPS 140-2 표준에 지정된 암호화 모듈에 대한 요구 사항을 지원합니다. CA SiteMinder for Secure Proxy Server 를 설치하면 현재 사용 중인 구성에 필요한 FIPS 지원 수준을 선택하라는 대화 상자가 나타납니다.

새로 설치할 때는 다음 세 개의 FIPS 모드 중 하나를 선택할 수 있습니다.

- 호환성 - 설치가 FIPS 와 호환되지 않음을 나타냅니다. 이전 버전의 CA SiteMinder for Secure Proxy Server 를 실행하는 클라이언트와 상호 작용하려면 이 모드를 선택합니다.
- 마이그레이션 - 데이터가 마이그레이션되는 동안 CA SiteMinder for Secure Proxy Server 가 FIPS 호환 알고리즘과 이전 버전의 CA SiteMinder for Secure Proxy Server 에 사용되는 알고리즘을 모두 동시에 사용하여 작동하도록 지정합니다.
- 전용 - CA SiteMinder for Secure Proxy Server 에서 FIPS 호환 알고리즘만 사용되고 허용되도록 지정합니다. 이 모드에서 설치하는 경우 수동 구성이 추가로 필요합니다.

설치 시 선택하는 FIPS 모드는 대개 정책 서버에 구성된 FIPS 모드와 동일합니다. 정책 서버가 마이그레이션 모드에 있는 경우에는 모든 모드에서 CA SiteMinder for Secure Proxy Server 와 함께 작동할 수 있습니다.

COMPAT 모드를 사용하고 기존 CA SiteMinder for Secure Proxy Server 설치를 업그레이드한 경우, 새 설치는 계속 COMPAT 모드에서 작동합니다. 다음 단계에 주목하십시오.

- 이후 단원에서 설명하는 대로 smreghost 명령을 사용하여 모드를 수동으로 변경할 수 있습니다.
- JsafeJCE 보안 공급자의 최초 FIPS 모드를 설정하기 위해 다음 줄을 JVM_HOME\jre\lib\security\java.security (Windows) 또는 JVM_HOME/jre/lib/security/java.security (UNIX)에 추가하십시오.

  ```
  com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
  ```

- 모드를 변경한 후에는 웹 에이전트, CA SiteMinder® SPS 서버 및 Apache 서버가 변경 내용을 적용하도록 시스템을 다시 시작하십시오.

**추가 정보:**

FIPS 마이그레이션 모드로 마이그레이션 (페이지 55)
FIPS 전용 모드에 대한 구성 프로세스 (페이지 54)

# FIPS 전용 모드에 대한 구성 프로세스

FIPS 전용 모드에서 CA SiteMinder for Secure Proxy Server 를 설치한 후에는 다음과 같은 추가 구성 단계를 수행해야 합니다.

- CA SiteMinder for Secure Proxy Server 가 전체 SSL 모드에서 실행 중인지 확인합니다.
- SSL 모드에서 CA SiteMinder for Secure Proxy Server 를 구성하는 데 사용된 서버 키가 FIPS 호환 암호화 알고리즘을 사용하여 생성되었는지 확인합니다.
- FIPS 전용 모드에서 SSL 을 구성하는 절차를 따릅니다.

# FIPS 마이그레이션 모드로 마이그레이션

이전 버전에서 업그레이드하고 FIPS 호환 알고리즘을 사용하려면 CA SiteMinder for Secure Proxy Server 내의 웹 에이전트를 호환성 모드에서 마이그레이션 모드로 변경합니다.

**CA SiteMinder for Secure Proxy Server 를 FIPS 마이그레이션 모드로 설정하려면**

1. CA SiteMinder for Secure Proxy Server 서비스를 중지합니다.
2. 명령줄 창을 엽니다.
3. 다음 명령을 입력합니다.

   ```
   smreghost -i policy_server_ip_address -u administrator_user_name -p
   administrator_password -hn hostname_for_registration -hc host_config_object -f
   path_to_host_config_file -o -cf MIGRATE
   ```

   예:

   ```
   smreghost -i localhost -u siteminder -p firewall -hn helloworld -hc host  -f
   "C:\Program Files\CA\secure-proxy\proxy-engine\conf\defaultagent\SmHost.conf"
   -o -cf  MIGRATE
   ```

4. 컴퓨터를 다시 시작합니다(Windows 에만 해당).
5. CA SiteMinder for Secure Proxy Server 서비스를 다시 시작합니다.

CA SiteMinder for Secure Proxy Server 내의 웹 에이전트가 FIPS 호환성 모드에서 FIPS 마이그레이션 모드로 변경되었습니다.

# FIPS 전용 모드로 마이그레이션

SiteMinder 정책 서버가 FIPS 전용 또는 FIPS 호환성 모드에 있는 경우 업그레이드 후 CA SiteMinder for Secure Proxy Server 의 FIPS 모드를 FIPS 호환성에서 FIPS 전용으로 변경할 수 있습니다.

**다음 단계를 수행하십시오.**

1. CA SiteMinder for Secure Proxy Server 서비스를 중지합니다.
2. OPENSSL_FIPS 환경 변수의 값을 1 로 설정합니다.
3. 다음 단계 중 하나를 수행합니다.
    1. Windows 에서 FIPS 모드를 변경하려면 CA_SM_PS_FIPS140 환경 변수를 ONLY 로 설정합니다.
    2. UNIX 에서 FIPS 모드를 변경하려면 다음 단계를 수행합니다.
        a. proxyserver.sh 파일을 엽니다.

            **기본 경로**: sps-home/proxy-engine/proxyserver.sh

        b. CA_SM_PS_FIPS140 환경 변수의 값을 ONLY 로 설정합니다.
4. 명령 프롬프트에서 다음 명령을 실행합니다.

    ```
    smreghost -i policy_server_ip_address -u administrator_user_name -p
    administrator_password -hn hostname_for_registration -hc host_config_object -f
    path_to_host_config_file -o -cf ONLY
    ```

    예:

    ```
    smreghost -i localhost -u siteminder -p firewall -hn helloworld -hc host  -f
    "C:\Program Files\CA\secure-proxy\proxy-engine\conf\defaultagent\SmHost.conf"
    -o -cf  ONLY
    ```
5. CA SiteMinder for Secure Proxy Server 가 전체 SSL 모드에서 실행 중인지 확인합니다. CA SiteMinder for Secure Proxy Server 내의 Apache 에 SSL 이 이미 사용되도록 설정되어 있는 경우 SSL 을 사용하지 않도록 설정하고 FIPS 전용 모드용으로 다시 구성해야 합니다.
6. httpd-ssl.conf 파일을 엽니다.

    **기본 경로**: sps_home\httpd\conf\extra\httpd-ssl.conf
7. SSLPassPhraseDialog 변수의 값을 custom 으로 설정합니다.
8. 다음 행의 주석 처리를 제거합니다.

    ```
    SSLCustomPropertiesFile "<sps_home>/Tomcat/properties/spsssl.properties"
    ```

9. SSLCustomPropertiesFile 변수의 값을 <sps_home>\httpd\conf\spsapachessl.properties 로 설정합니다.
10. SSLSpsFipsMode 변수의 값을 ONLY 로 설정합니다.
11. 컴퓨터를 다시 시작합니다.
12. CA SiteMinder for Secure Proxy Server 서비스를 시작합니다.

# 제 4 장: FSS(Federation Security Services)와 함께 SPS 사용

이 섹션은 다음 항목을 포함하고 있습니다.



## FSS 소개

SiteMinder FSS(Federation Security Services)는 비즈니스 파트너 간에 보안 정보를 교환할 수 있게 해 줍니다. 이 서비스는 엔터프라이즈 간의 원활한 인증 및 세부적인 권한 부여 기능을 제공합니다.

FSS 는 다음 방법으로 CA SiteMinder for Secure Proxy Server 와 함께 구현됩니다.

- SiteMinder 웹 에이전트의 대체 서비스로
- SiteMinder 웹 에이전트 및 웹 에이전트 옵션 팩의 대체 서비스로

페더레이션 서비스를 사용하여 조직과 조직의 파트너는 다음을 수행할 수 있습니다.

- 사용자 정보를 안전하게 교환합니다.
- 한 조직의 사용자 아이덴티티를 다른 조직의 사용자 아이덴티티에 매핑합니다.
- 서로 다른 조직 간에 싱글 사인온을 제공합니다.
- 파트너에서 수신한 사용자 정보를 기반으로 리소스에 대한 액세스를 제어합니다.
- Windows, UNIX 및 다양한 웹 서버(예: IIS, Sun Java System 및 Apache) 같은 이종 환경에서 상호 운용합니다.

# SiteMinder 페더레이션 환경에서의 SPS 사용 사례

파트너 간의 비즈니스 처리 방식이 다양한 만큼 페더레이션된 네트워크의 사용 사례도 다양할 수 있습니다. 다음 사용 사례에서는 파트너 간에 싱글 사인온을 제공하기 위해 사용자 아이덴티티를 처리하는 다양한 방법을 보여 줍니다.

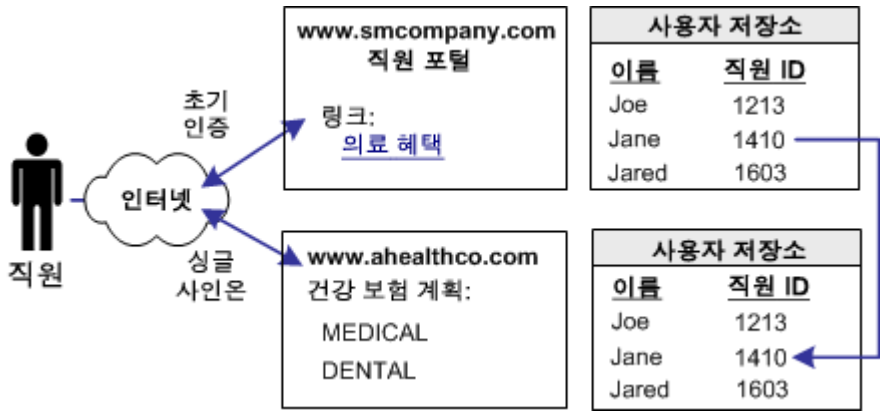
자세한 사용 사례는 *CA SiteMinder Federation Security Services Guide*(CA SiteMinder Federation Security Services 안내서)를 참조하십시오.

## 사용 사례 1: 계정 연결에 기반한 싱글 사인온

사용 사례 1 에서 smcompany.com 은 파트너 회사인 ahealthco.com 과 직원 의료 혜택 관리 계약을 맺습니다.

smcompany.com 의 직원이 이 회사의 직원 포털 사이트(www.smcompany.com)에서 인증을 받은 다음 ahealthco.com 의 의료 혜택 정보를 보기 위해 링크를 클릭합니다. 이 직원은 ahealthco.com 웹 사이트에 연결되며 웹 사이트에 사인온할 필요 없이 해당 직원의 의료 혜택 정보가 제공됩니다.

다음 그림에서는 이 사용 사례를 보여 줍니다.



ahealthco.com 이라는 회사가 smcompany.com 의 직원에 대한 모든 의료 관련 정보를 유지 관리합니다. 이를 위해 ahealthco.com 은 smcompany.com 의 모든 직원에 대한 사용자 아이덴티티를 유지 관리합니다. smcompany.com 의 직원이 ahealthco.com 에 액세스하면 해당 직원에 대한 식별자가 안전한 방식으로 smcompany.com 에서 ahealthco.com 으로 전달됩니다. 이 식별자를 통해 ahealthco.com 은 사용자를 확인하고 해당 사용자에 대해 허용할 액세스 수준을 결정할 수 있습니다.

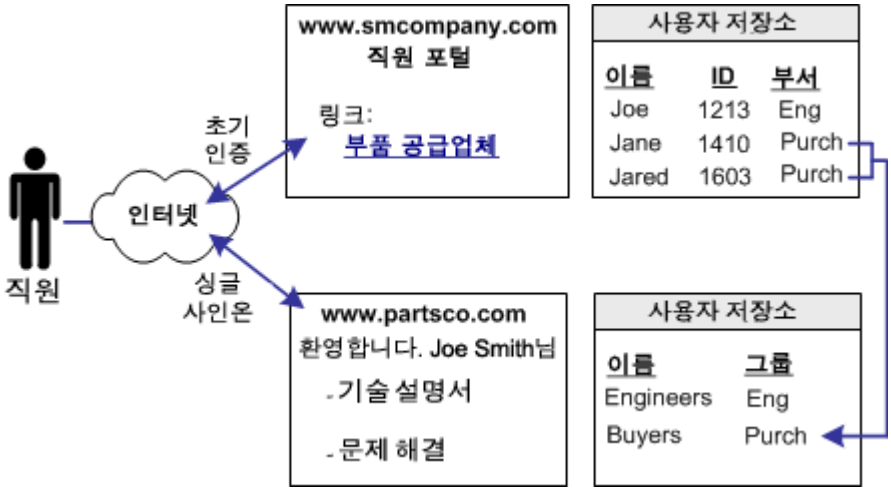
**추가 정보:**

솔루션 1: 계정 연결에 기반한 SSO (페이지 66)

## 사용 사례 2: 사용자 특성 프로필에 기반한 싱글 사인온

사용 사례 2 에서 smcompany.com 은 partsco.com 이라는 파트너로부터 부품을 구매합니다.

엔지니어가 직원 포털(smcompany.com)에서 인증을 받은 다음 partsco.com 의 정보에 액세스하기 위해 링크를 클릭합니다. smcompany.com 의 엔지니어인 사용자는 로그인할 필요 없이 partsco.com 웹 사이트의 "Specifications"(사양) 부분에 직접 연결됩니다.




smcompany.com 의 구매자가 인증을 받은 다음 partsco.com 에 대한 링크를 클릭하면 partsco.com 웹 사이트의 "Parts List"(부품 목록) 부분에 직접 연결됩니다. 구매자는 로그인할 필요가 없습니다.

개별 사용자에 대한 인터페이스를 개인화하기 위해 사용자 이름 등의 추가 특성이 smcompany.com 에서 partsco.com 으로 전달됩니다.

partsco.com 은 smcompany.com 의 일부 직원에 대한 사용자 아이덴티티만 유지 관리하면서 웹 사이트의 중요한 부분에 대한 액세스를 제어하고자 합니다. 액세스를 제어하기 위해 partsco.com 은 smcompany.com 에 있는 사용자에 대해 제한된 수의 프로필 아이덴티티를 유지 관리합니다. 엔지니어에 대한 프로필 아이덴티티와 구매자에 대한 프로필 아이덴티티가 하나씩 유지 관리됩니다.

smcompany.com 의 직원이 partsco.com 에 액세스하면 smcompany.com 이 안전한 방식으로 사용자 특성을 partsco.com 에 보냅니다. partsco.com 은 특성을 사용하여 액세스를 제어하는 프로필 아이덴티티를 결정합니다.

**추가 정보:**

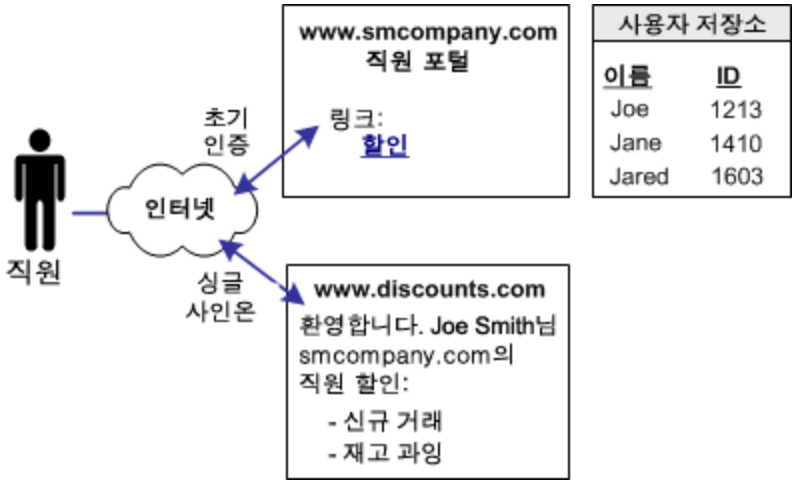
솔루션 2: 사용자 특성 프로필을 사용하는 SSO (페이지 70)

## 사용 사례 3: 로컬 사용자 계정이 없는 싱글 사인온

사용 사례 3 에서 smcompany.com 은 discounts.com 과의 파트너 관계를 설정하여 직원에게 할인을 제공합니다.

smcompany.com 의 직원은 smcompany.com 에서 인증되고 링크를 클릭하여 discounts.com 에 액세스합니다. 이 직원은 discounts.com 웹 사이트에 연결되며 discounts.com 웹 사이트에 로그인할 필요 없이 smcompany.com 의 직원이 사용할 수 있는 할인이 제공됩니다.

다음 그림에서는 이 사용 사례를 보여 줍니다.




discounts.com 은 smcompany.com 에 대한 아이덴티티를 유지 관리하지 않습니다. 이 회사는 smcompany.com 의 모든 직원이 smcompany.com 에서 인증을 받은 경우 discounts.com 에 액세스하도록 허용합니다. smcompany.com 의 직원이 discounts.com 에 액세스하면 인증 정보가 안전한 방식으로 smcompany.com 에서 discounts.com 으로 전송됩니다. 이 정보는 discounts.com 에 대한 액세스를 허용하는 데 사용됩니다.

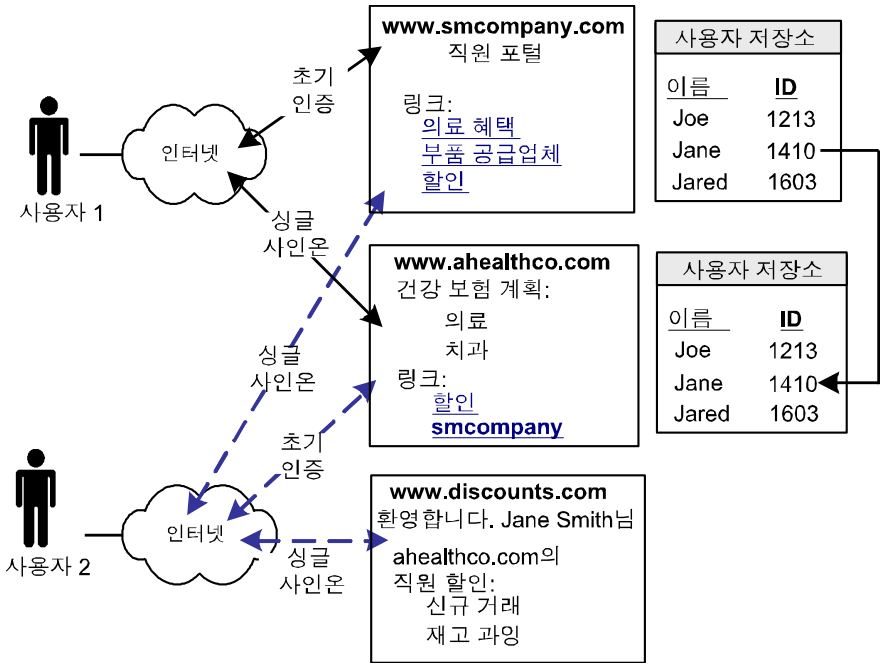
개별 사용자에 대한 인터페이스를 개인화하기 위해 사용자 이름 등의 추가 특성이 smcompany.com 에서 discounts.com 으로 전달됩니다.

**추가 정보:**

솔루션 3: 로컬 사용자 계정이 없는 SSO (페이지 72)

## 사용 사례 4: 확장 네트워크

사용 사례 4 에서 smcompany.com, ahealthco.com 및 discounts.com 은 모두 확장 페더레이션된 네트워크에 참여합니다. 이 사례는 이전 사용 사례를 확장한 것입니다.




이 네트워크에서 ahealthco.com 의 일부 고객은 smcompany.com 에서 근무하지 않습니다. ahealthco.com 은 ahealthco.com 과 discounts.com 간의 관계를 설정하여 해당 고객에게만 할인을 제공합니다. ahealthco.com 은 ahealthco.com 이 각 사용자에 대한 암호 등의 로컬 자격 증명을 관리하도록 모든 고객에 대한 사용자 아이덴티티를 유지 관리합니다. 로컬 자격 증명을 관리하여 ahealthco.com 은 사용자를 인증할 수 있고 해당 파트너에 대한 싱글 사인온 액세스를 제공할 수 있습니다.

이 확장 네트워크에서 사용자는 다음과 같이 각 웹 사이트에 다르게 액세스합니다.

- 사용자 1 은 ahealthco.com 웹 사이트의 의료 혜택 정보에 액세스합니다. 사용자 1 은 직원 포털(smcompany.com)에 있는 PartsSupplier 링크를 클릭하여 partsco.com 웹 사이트에 액세스할 수 있습니다. 사용자 1 이 직원 포털에 있는 링크를 클릭하여 discounts.com 의 할인에 액세스할 수도 있습니다.

사용자 2 는 ahealthco.com 웹 사이트에서 인증을 받은 다음 discounts.com 웹 사이트에 로그인할 필요 없이 링크를 클릭하여 discounts.com 의 할인에 액세스합니다. 이 사이트가 사용자 2 에게 제공하는 할인은 ahealthco.com 과 discounts.com 간의 비즈니스 처리 방식을 반영합니다. smcompany.com 의 직원인 사용자 2 가 ahealthco.com 에 있는 링크를 클릭하고 웹 사이트에 로그인할 필요 없이 직원 포털(smcompany.com)에 액세스할 수도 있습니다.

- 사용자 3(예에 나와 있지 않음)은 ahealthco.com 의 고객이지만 smcompany.com 의 직원이 아닙니다. 사용자 3 이 ahealthco.com 웹 사이트에서 인증을 받은 다음 discounts.com 의 할인에 액세스하기 위해 링크를 클릭합니다. 사용자 3 은 웹 사이트에 로그인하지 않습니다. 이 사이트가 사용자 3 에게 제공하는 할인은 ahealthco.com 과 discounts.com 간의 비즈니스 처리 방식을 반영합니다. 사용자 3 은 smcompany.com 의 직원이 아니기 때문에 smcompany.com 웹 사이트에 액세스할 수 없습니다.

**추가 정보:**

솔루션 4: 확장 네트워크의 SSO (페이지 74)

# SiteMinder 페더레이션 환경에서의 SPS 역할

CA SiteMinder for Secure Proxy Server 는 다음 두 가지 역할 중 하나로 페더레이션 사용 사례에 대한 솔루션을 제공할 수 있습니다.

- SiteMinder 웹 에이전트를 대체하는 표준 프록시 서버
- 페더레이션 게이트웨이

이 두 역할의 주요 차이점은 필요한 구성과 배포 작업에 있습니다. SPS 를 웹 에이전트를 대체하는 프록시 서버로 사용할 경우에는 페더레이션 웹 서비스 응용 프로그램을 실행하기 위한 별도의 서버 및 서블릿 엔진도 설정해야 합니다.

페더레이션 게이트웨이의 역할을 하는 프록시 서버에는 웹 에이전트 및 페더레이션 웹 서비스 응용 프로그램의 구성 요소가 기본적으로 제공됩니다. 전용 서버 및 서블릿 엔진이 별도로 구성되지 않으므로 페더레이션 설정이 간단합니다.

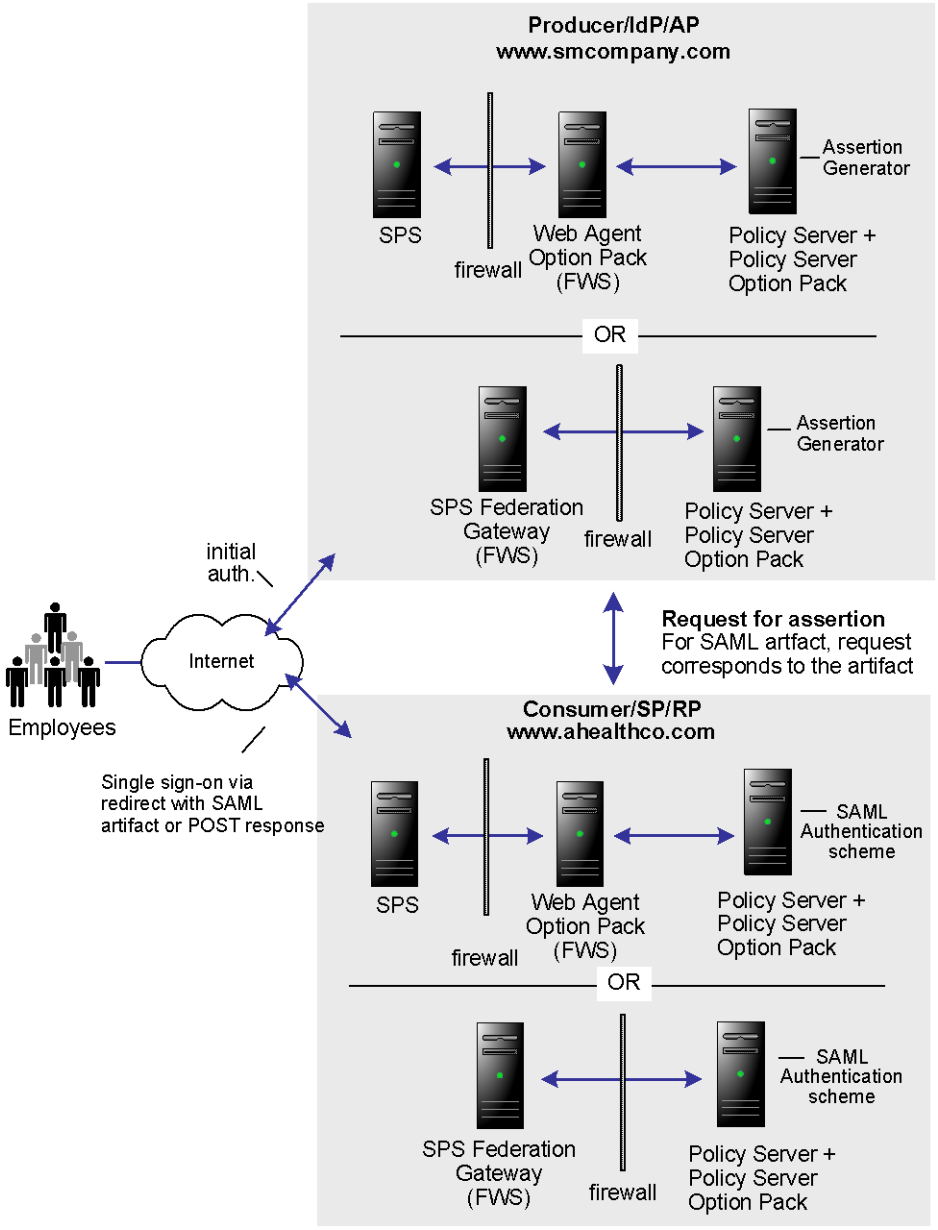
# SPS 사용 사례에 대한 솔루션

다음 단원에서는 페더레이션 사용 사례에 대한 CA SiteMinder for Secure Proxy Server 솔루션을 보여 줍니다.

## 솔루션 1: 계정 연결에 기반한 SSO

솔루션 1 에서는 smcompany.com 과 ahealthco.com 에 FSS(Federation Security Services)를 배포하여 사용 사례 1: 계정 연결에 기반한 싱글 사인온 (페이지 60)을 해결하는 방법을 보여 줍니다.

다음 그림에서는 계정 연결을 기반으로 하는 솔루션을 보여 줍니다.



CA SiteMinder 는68 두 사이트 모두에 배포되며 설치는 smcompany.com 과 ahealthco.com 모두 동일합니다. CA SiteMinder for Secure Proxy Server 와 웹 에이전트 옵션 팩, 또는 CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이는 웹 서버 시스템에 설치될 수 있으며 정책 서버 옵션 팩을 포함한 정책 서버는 다른 컴퓨터에 설치됩니다.

생산 측의 FWS 응용 프로그램은 어설션을 검색하는 서비스를 제공합니다. 소비 측의 FWS 응용 프로그램은 어설션을 소비하는 서비스를 제공합니다.

## 솔루션 1 에 SAML 1.x 아티팩트 인증 사용

다음 프로세스는 계정 연결을 사용하는 싱글 사인온을 위한 한 가지 솔루션입니다. 이 솔루션은 SAML 1.x 아티팩트 프로필을 사용합니다. 이 사용 사례에 대해 다른 프로필(SAML 1.x POST 와 SAML 2.0 아티팩트 및 POST)을 사용하는 다른 솔루션도 있습니다. 이러한 솔루션은 *CA SiteMinder Federation Security Services Guide*(CA SiteMinder Federation Security Services 안내서)를 참조하십시오.

이 솔루션에서 smcompany.com 은 생산자 사이트로 작동하고 있습니다. smcompany.com 의 직원이 직원 포털(www.smcompany.com)에 액세스하는 경우 이벤트 순서는 다음과 같습니다.

1. CA SiteMinder for Secure Proxy Server 가 초기 인증을 제공합니다.
2. 직원이 ahealthco.com 의 의료 혜택 정보를 보기 위해 smcompany.com 에 있는 링크를 클릭하면 해당 링크가 www.smcompany.com 의 사이트 간 전송 서비스에 요청합니다.
3. 사이트 간 전송 서비스가 어설션 생성기를 호출하면 어설션 생성기가 SAML 어설션을 생성하여 SiteMinder 세션 서버에 삽입하고 SAML 아티팩트를 반환합니다.
4. CA SiteMinder for Secure Proxy Server 가 SAML 브라우저 아티팩트 프로토콜에 따라 SAML 아티팩트와 함께 사용자를 www.ahealthco.com 으로 리디렉션합니다.

ahealthco.com 은 소비자 사이트로 작동하고 있습니다. SAML 아티팩트가 포함된 리디렉션 요청은 ahealthco.com 의 SAML 자격 증명 수집기 페더레이션 웹 서비스에 의해 처리됩니다.

이벤트 순서는 다음과 같습니다.

1. SAML 자격 증명 수집기가 SAML 아티팩트 인증 체계를 호출하여 smcompany.com 의 어설션 검색 서비스 위치를 파악합니다.
2. SAML 자격 증명 수집기가 www.smcompany.com 의 어설션 검색 서비스를 호출합니다.
3. www.smcompany.com 의 어설션 검색 서비스가 SiteMinder 세션 서버에서 어설션을 검색하여 ahealthco.com 의 SAML 자격 증명 수집기로 반환합니다.
4. 그런 다음 유효성 검사와 세션 생성을 위해 SAML 자격 증명 수집기가 어설션을 SAML 아티팩트 인증 체계에 전달하고 사용자의 브라우저에 SiteMinder 세션 쿠키를 발급합니다.
5. 이제 사용자는 ahealthco.com 의 정책 서버에 정의되고 ahealthco.com 의 CA SiteMinder for Secure Proxy Server 에 의해 적용된 정책을 기반으로 ahealthco.com 의 리소스에 액세스할 수 있습니다.

이 예에서 smcompany.com 의 관리자는 정책 서버 사용자 인터페이스를 사용하여 ahealthco.com 에 대해 가맹을 구성합니다. 가맹은 고유 사용자 ID 인 특성을 사용하여 구성됩니다. 이로 인해 어설션 생성기는 해당 특성을 ahealthco.com 에 대해 생성된 SAML 어설션에 사용자 프로필의 일부로 포함합니다.

ahealthco.com 의 관리자는 정책 서버 사용자 인터페이스를 사용하여 smcompany.com 에 대한 SAML 아티팩트 인증 체계를 구성합니다. 이 인증 체계는 smcompany.com 의 어설션 검색 서비스 위치, SAML 어설션에서 고유 사용자 ID 를 추출하는 방법, 어설션에서 추출된 값과 일치하는 사용자 레코드를 ahealthco.com 의 사용자 디렉터리에서 검색하는 방법 등을 지정합니다.

## 솔루션 2: 사용자 특성 프로필을 사용하는 SSO

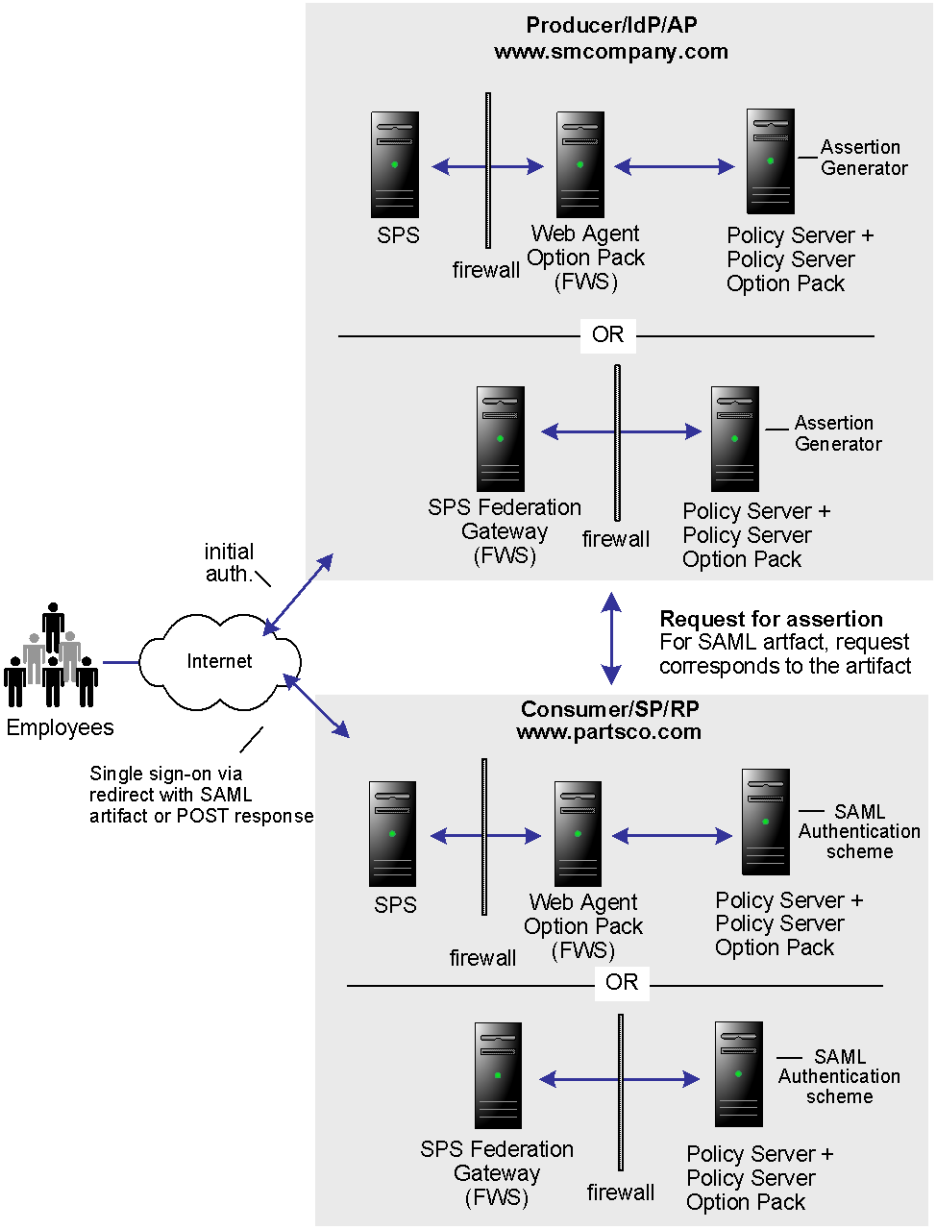
솔루션 2 에서는 smcompany.com 과 partsco.com 에 SiteMinder FSS(Federation Security Services)를 배포하여 사용 사례 2: 사용자 특성 프로필에 기반한 싱글 사인온 (페이지 61)을 해결하는 방법을 보여 줍니다.

CA SiteMinder 는70 두 사이트 모두에 배포됩니다. 사용자와 각 사이트 간의 상호 작용은 유사합니다. 이때 partsco.com 은 소비 기관으로 작동합니다. 생산 측의 FWS 응용 프로그램은 어설션을 검색하는 서비스를 제공합니다. 소비 측의 FWS 응용 프로그램은 어설션을 소비하는 서비스를 제공합니다.

다음 그림은 SAML 1.x, SAML 2.0 및 WS-페더레이션의 경우 유사하지만 다음과 같이 페더레이션 웹 서비스 구성 요소가 다릅니다.

- SAML 1.x 의 경우 어설션 검색 서비스(아티팩트 프로필만 해당)는 생산자에 있고 SAML 자격 증명 수집기는 SP 에 있습니다.
- SAML 2.0 의 경우 아티팩트 레졸루션 서비스(아티팩트 바인딩만 해당)는 IdP 에 있고 어설션 소비자 서비스는 SP 에 있습니다.
- WS-페더레이션의 경우 싱글 사인온 서비스는 AP 에 있고 보안 토큰 소비자 서비스는 RP 에 있습니다.

  **참고:** WS-페더레이션은 HTTP-POST 바인딩만 지원합니다.


Producer/IdP/AP
www.smcompany.com
SPS
firewall
Web Agent
Option Pack
(FWS)
Policy Server +
Policy Server
Option Pack
Assertion
Generator
OR
SPS Federation
Gateway
(FWS)
firewall
Policy Server +
Policy Server
Option Pack
Assertion
Generator
initial
auth.
Internet
Employees
Single sign-on via
redirect with SAML
artifact or POST response
Request for assertion
For SAML artfact, request
corresponds to the artifact
Consumer/SP/RP
www.partsco.com
SPS
firewall
Web Agent
Option Pack
(FWS)
Policy Server +
Policy Server
Option Pack
SAML
Authentication
scheme
OR
SPS Federation
Gateway
(FWS)
firewall
Policy Server +
Policy Server
Option Pack
SAML
Authentication
scheme

구성은 다음을 제외하고 솔루션 1: 계정 연결에 기반한 싱글 사인온과 유사합니다.

- smcompany.com 의 관리자는 이 회사의 사용자 부서를 지정하는 특성을 사용하여 partsco.com 에 대한 소비자/SP 를 정의합니다. 어설션 생성기는 이 특성을 partsco.com 에 대해 생성되는 SAML 어설션에 사용자 프로필의 일부로 포함합니다.
- partsco.com 의 관리자는 smcompany.com 에 대한 인증 체계(아티팩트, POST 또는 WS-페더레이션)를 정의합니다. 이 체계는 SAML 어설션에서 부서 특성을 추출하고 partsco.com 의 사용자 디렉터리에서 어설션의 부서 값과 일치하는 사용자 레코드를 검색합니다. 관리자는 partsco.com 의 웹 사이트에 액세스하도록 허용된 각 부서에 대해 사용자 프로필 레코드를 하나씩 정의합니다.

## 솔루션 3: 로컬 사용자 계정이 없는 SSO

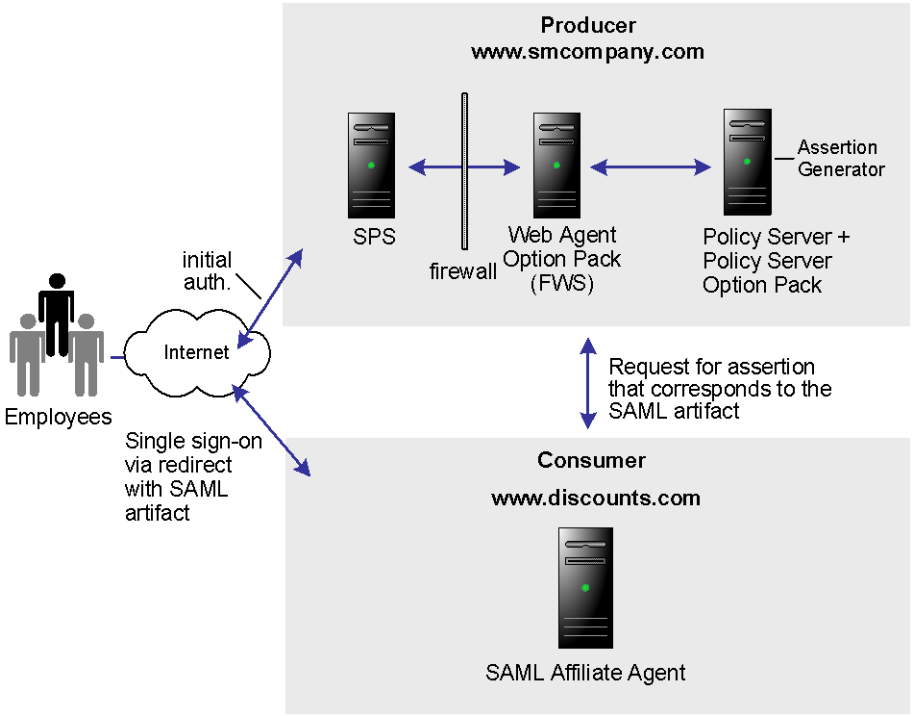
솔루션 3 에서는 smcompany.com 과 discounts.com 에 SiteMinder FSS(Federation Security Services)를 배포하여 사용 사례 3: 로컬 사용자 계정이 없는 싱글 사인온 (페이지 62)을 해결하는 방법을 보여 줍니다.

CA SiteMinder for Secure Proxy Server 와 웹 에이전트 옵션 팩을 각각 별도의 컴퓨터에 설치하고 세 번째 컴퓨터에는 정책 서버 옵션 팩을 포함한 정책 서버를 설치하여 smcompany.com 에 CA SiteMinder 가72 배포됩니다. SAML 가맹 에이전트는 discounts.com 에 설치됩니다. 이 에이전트는 SAML 1.0 만 지원합니다.

생산 측의 FWS 응용 프로그램은 어설션 검색 서비스를 제공합니다. 소비자 측의 FWS 응용 프로그램은 SAML 자격 증명 수집기를 제공합니다.

**참고:** CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이는 SAML 1.0 을 지원하지 않으므로 SAML 가맹 에이전트에 대한 생산자로 작동할 수 없습니다.

다음 그림에서는 로컬 사용자 계정이 없는 싱글 사인온을 보여 줍니다.




smcompany.com 은 SAML 1.x 생산자로 작동하고 있습니다. smcompany.com 의 직원이 직원 포털(www.smcompany.com)에 액세스하면 다음 프로세스가 발생합니다.

1. CA SiteMinder for Secure Proxy Server 가 초기 인증을 제공합니다.
2. 직원이 discounts.com 의 거래에 액세스하기 위해 www.smcompany.com 에 있는 링크를 클릭하면 해당 링크가 www.smcompany.com 의 CA SiteMinder for Secure Proxy Server 에 요청합니다.
3. www.smcompany.com 의 CA SiteMinder for Secure Proxy Server 가 어설션 생성기를 호출하면 어설션 생성기가 SAML 어설션을 생성하여 SiteMinder 세션 서버에 삽입하고 SAML 아티팩트를 반환합니다.
4. CA SiteMinder for Secure Proxy Server 가 SAML 브라우저 아티팩트 프로토콜에 따라 SAML 아티팩트와 함께 사용자를 www.discounts.com 으로 리디렉션합니다.

discounts.com 은 소비자 사이트로 작동하고 있습니다. SAML 아티팩트가 포함된 리디렉션 요청은 다음과 같이 www.discounts.com 의 SAML 가맹 에이전트에 의해 처리됩니다.

1. SAML 가맹 에이전트가 구성 파일에서 www.smcompany.com 의 어설션 검색 서비스 위치를 파악합니다.
2. SAML 가맹 에이전트가 www.smcompany.com 의 어설션 검색 서비스를 호출합니다.
3. www.smcompany.com 의 어설션 검색 서비스가 SiteMinder 세션 서버에서 어설션을 검색하여 www.discounts.com 의 SAML 가맹 에이전트로 반환합니다.
4. 그런 다음 SAML 가맹 에이전트가 SAML 어설션의 유효성을 검사하고 사용자의 브라우저에 SiteMinder 가맹 세션 쿠키를 발급합니다.
5. 이제 사용자가 discounts.com 의 리소스에 액세스할 수 있습니다.

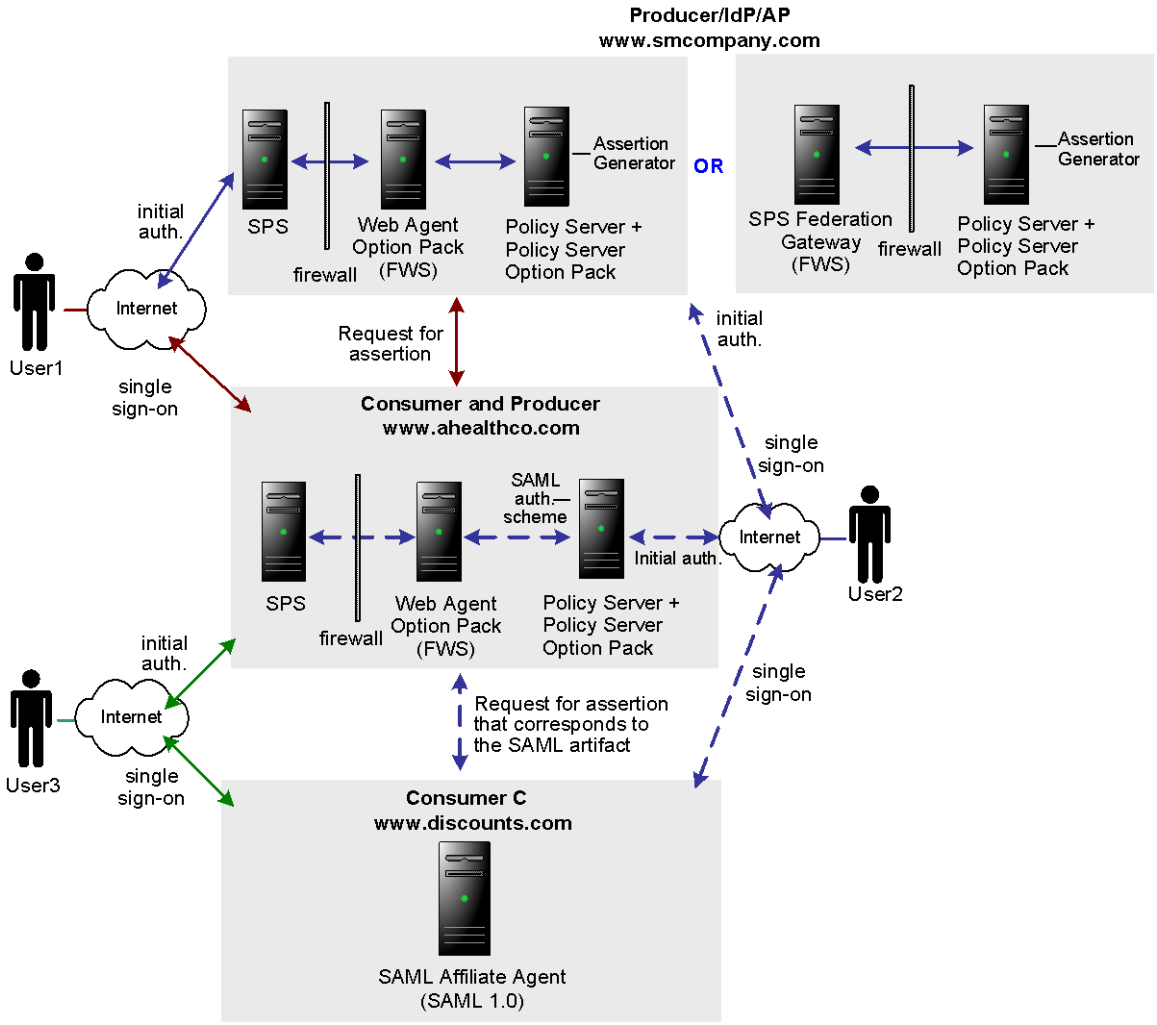
smcompany.com 의 관리자는 정책 서버 사용자 인터페이스를 사용하여 discounts.com 에 대해 가맹을 구성합니다. 가맹은 discounts.com 에 전달될 특성 정보를 사용하여 구성됩니다. 어설션 생성기는 이러한 특성을 discounts.com 에 대해 생성되는 SAML 어설션에 사용자 프로필의 일부로 포함합니다.

discounts.com 의 관리자는 discounts.com 사이트, smcompany.com 의 어설션 검색 서비스 위치, smcompany.com 에 정의된 가맹이 보호할 리소스 등에 대한 정보를 사용하여 SAML 가맹 에이전트를 구성합니다.

## 솔루션 4: 확장 네트워크의 SSO

솔루션 4 에서는 smcompany.com, ahealthco.com 및 discounts.com 에 SiteMinder FSS(Federation Security Services)를 배포하여 사용 사례 4: 확장 네트워크 (페이지 64)를 해결하는 방법을 보여 줍니다.

다음 그림에서는 확장 네트워크를 보여 줍니다. SAML 1.x 는 사용되는 프로토콜입니다.



SiteMinder 는 smcompany.com 과 ahealthco.com 에 배포됩니다. smcompany.com 에서는 CA SiteMinder for Secure Proxy Server 와 웹 에이전트 옵션 팩을 두 컴퓨터에 나눠 설치하거나 CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이를 한 컴퓨터에 설치할 수 있습니다. 정책 서버 옵션 팩을 포함한 정책 서버는 또 다른 컴퓨터에 설치합니다. ahealthco.com 에서는 CA SiteMinder for Secure Proxy Server 와 웹 에이전트 옵션 팩을 두 컴퓨터에 나눠 설치하고 정책 서버 옵션 팩을 포함한 정책 서버를 또 다른 컴퓨터에 설치합니다. discounts.com 에는 SAML 가맹 에이전트가 설치됩니다.

생산 측의 FWS 응용 프로그램은 어설션을 검색하는 서비스를 제공합니다. 소비 측의 FWS 응용 프로그램은 어설션을 소비하는 서비스를 제공합니다.

솔루션 4 에서

- smcompany.com 은 사용자 1 에 대한 생산자와 사용자 2 에 대한 소비자로 작동합니다.
- ahealthco.com 은 사용자 1 에 대한 소비자와 사용자 2 및 사용자 3 에 대한 생산자로 작동합니다.
- discounts.com 은 사용자 1, 사용자 2 및 사용자 3 에 대한 소비자로 작동합니다.

smcompany.com 의 관리자는 가맹 도메인에 ahealthco.com 과 discounts.com 을 나타내는 엔터티 두 개를 구성했습니다. 이러한 사이트는 앞서 설명한 예 1 및 예 3 과 유사한 방식으로 구성되었지만 다음과 같이 해당 구성이 확장되었습니다.

- smcompany.com 의 관리자는 SAML 인증 체계(아티팩트 또는 POST)를 구성했습니다. 사용자 2 의 경우 인증 체계를 통해 smcompany.com 이 ahealthco.com 에 대한 소비자로 작동할 수 있습니다.
- ahealthco.com 에서
  - 관리자는 사용자 2 에 대한 어설션이 생성되도록 smcompany.com 을 나타내는 가맹 개체를 구성했습니다. 이로 인해 smcompany.com 에 대한 싱글 사인온이 가능해집니다.
  - 관리자는 사용자 2 와 사용자 3 에 대한 어설션이 생성되도록 discounts.com 을 나타내는 가맹 개체를 구성했습니다. 이로 인해 discounts.com 에 대한 싱글 사인온이 가능해집니다.

- discounts.com 의 관리자는 예 3 과 마찬가지로 smcompany.com 에 대한 소비자로 작동하도록 SAML 가맹 에이전트를 구성했습니다. 두 사이트를 연결하는 화살표는 그림에 나와 있지 않습니다. 또한 discounts.com 의 관리자는 SAML 가맹 에이전트가 사용자 2 와 사용자 3 에 대해 ahealthco.com 으로부터 받은 어설션을 소비할 수 있도록 ahealthco.com 에 대한 구성 정보를 추가했습니다.

# 쿠키를 사용하지 않는 페더레이션

일부 장치 또는 환경에서는 사용자 세션을 설정하고 싱글 사인온을 제공하는 데 쿠키를 사용할 수 없습니다.

페더레이션 환경에서 사용할 수 있는 세션 체계 유형은 쿠키를 사용하지 않는 체계뿐입니다. 쿠키를 사용하지 않는 페더레이션 체계는 싱글 사인온을 설정하는 데 사용됩니다. 쿠키를 지원하지 않는 모바일 장치를 사용할 경우 FWS 가 생성한 쿠키(세션 및 특성)가 클라이언트로 다시 전송되지 않는지 확인하십시오.

### 생산 사이트에서 쿠키를 사용하지 않는 페더레이션

어설션을 생산하는 사이트에서 쿠키를 사용하지 않는 트랜잭션은 다음과 같이 처리됩니다.

1. CA SiteMinder for Secure Proxy Server 가 리디렉션을 요청하는 가상 호스트에 쿠키를 사용하지 않는 페더레이션이 사용되는지 여부를 확인합니다.
2. CA SiteMinder for Secure Proxy Server 가 세션 체계가 simple_url 체계와 같이 다시 쓰기 가능한 체계인지 확인합니다.
3. 체계가 다시 쓰기 가능한 체계이면 CA SiteMinder for Secure Proxy Server 는 해당 세션에 대한 세션 키가 생성되었는지 여부와 이 키를 사용할 수 있는지 여부를 확인합니다.
4. CA SiteMinder for Secure Proxy Server 가 HTTP 응답의 위치 헤더가 다음 조건 중 하나를 충족하는지 확인합니다.
   - 위치 헤더가 다시 써져야 합니다.
   - 위치 헤더가 요청의 호스트와 동일해야 합니다.
5. CA SiteMinder for Secure Proxy Server 가 리디렉션된 URL 에 세션 키 정보를 포함하도록 리디렉션 응답을 다시 씁니다.

### 소비 사이트에서 쿠키를 사용하지 않는 페더레이션

어설션을 소비하는 사이트에서 쿠키를 사용하지 않는 페더레이션이 사용될 경우 웹 에이전트를 대체하는 CA SiteMinder for Secure Proxy Server 는 백엔드 서버에서 SAML 인증을 사용하여 리디렉션을 처리합니다.

쿠키를 사용하지 않는 페더레이션에서 CA SiteMinder for Secure Proxy Server 가 요청을 처리하는 방식은 다음과 같습니다.

1. CA SiteMinder for Secure Proxy Server 가 휴대폰과 같이 쿠키를 사용하지 않는 장치에서 요청을 수신합니다.
2. CA SiteMinder for Secure Proxy Server 가 리디렉션을 요청하는 가상 호스트에 대해 쿠키를 사용하지 않는 페더레이션이 사용되도록 설정되어 있는지 확인합니다.
3. CA SiteMinder for Secure Proxy Server 가 다음 조건이 충족되었는지 확인합니다.
   - 백엔드 서버로부터의 응답이 리디렉션 응답이어야 합니다.
   - 응답에 SMSESSION 쿠키가 포함되어 있어야 합니다.

   이 두 조건이 동시에 충족되면 백엔드 서버의 FWS 응용 프로그램에서 SAML 인증이 수행되었음을 나타냅니다.
4. CA SiteMinder for Secure Proxy Server 가 사용 중인 세션 체계를 검색합니다.
5. CA SiteMinder for Secure Proxy Server 가 쿠키를 사용하지 않는 관련 세션을 생성하고 해당 세션 정보를 세션 저장소에 추가합니다.
6. 세션 체계가 단순 URL 세션 체계와 같이 다시 쓰기 가능한 체계일 경우 CA SiteMinder for Secure Proxy Server 가 해당 세션 키를 사용하여 위치 헤더를 다시 씁니다.
7. 쿠키를 사용하지 않는 페더레이션 세션 변환이 수행된 것으로 확인되면 CA SiteMinder for Secure Proxy Server 가 브라우저로 전송되는 응답에서 SMSESSION 쿠키를 삭제합니다.
8. CA SiteMinder for Secure Proxy Server 가 특성 쿠키도 삭제해야 하는지 확인합니다. 이 작업은 deleteallcookiesforfed 매개 변수 (페이지 142)를 확인하는 방법으로 수행됩니다. 이 매개 변수가 yes 로 설정되어 있으면 CA SiteMinder for Secure Proxy Server 가 브라우저로 전송되는 응답에서 다른 모든 쿠키를 삭제합니다.

## 소비 측에서 쿠키를 사용하지 않는 페더레이션을 사용하도록 설정

어설션 소비 측에서 CA SiteMinder for Secure Proxy Server 가 웹 에이전트를 대체하는 경우 CA SiteMinder for Secure Proxy Server 에 의해 구현되며 쿠키를 사용하지 않는 모든 세션 체계에 대해 쿠키를 사용하지 않는 페더레이션 매개 변수가 사용되도록 설정됩니다.

**소비 측에서 CA SiteMinder for Secure Proxy Server 에 대해 쿠키를 사용하지 않는 페더레이션을 사용하도록 설정하려면**

1. *sps_home*/secure-proxy/Tomcat/properties 의 noodle.properties 파일을 엽니다.
2. 다음 두 줄에서 '#'을 제거하고 파일을 저장합니다.
   - filter._cookielessfederation_.class=org.tigris.noodle.filters.CookielessFedFilter
   - filter._cookielessfederation_.order=1

   설정이 저장됩니다.
3. *sps_home*/secure-proxy/proxy-engine/conf 에 있는 server.conf 파일을 엽니다.
4. FWS 서비스를 제공하는 가상 호스트에 대한 가상 호스트 섹션에 다음 코드를 추가합니다.

   cookielessfederation="yes"
5. 파일을 저장합니다.

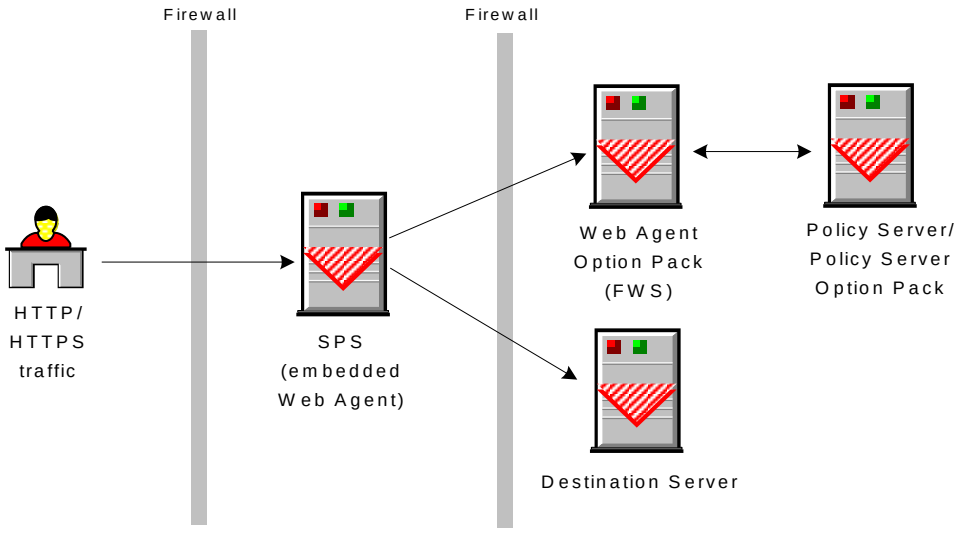
   소비하는 파트너에서 CA SiteMinder for Secure Proxy Server 가 쿠키를 사용하지 않는 페더레이션용으로 구성되었습니다.

# 웹 에이전트 대신 SPS 사용

페더레이션된 싱글 사인온을 제공하기 위해 CA SiteMinder for Secure Proxy Server 를 SiteMinder 웹 에이전트를 대체하는 역할로 사용할 수 있습니다. CA SiteMinder for Secure Proxy Server 는 웹 에이전트 옵션 팩과 결합하여 서블릿 패키지의 모음인 FWS(페더레이션 웹 서비스) 응용 프로그램을 웹 응용 프로그램으로 제공합니다. 이 응용 프로그램은 SiteMinder 페더레이션 기능의 상당수를 제공합니다.

페더레이션 환경에서 CA SiteMinder for Secure Proxy Server 를 구성하려면 SiteMinder FSS(Federation Security Services)에 대해 잘 알고 있어야 합니다. FSS 에 대한 자세한 내용은 *CA SiteMinder Federation Security Services Guide*(CA SiteMinder Federation Security Services 안내서)를 참조하십시오.

다음 그림에서는 CA SiteMinder for Secure Proxy Server 가 SiteMinder 웹 에이전트를 대체하는 환경을 보여 줍니다.




**중요!** 페더레이션 환경에 웹 에이전트 대신 CA SiteMinder for Secure Proxy Server 를 사용하려면 웹 에이전트 옵션 팩에 전용 웹 서버 및 서블릿 엔진이 필요합니다.

## SPS 를 웹 에이전트 대체 서비스로 사용하기 위한 사전 요구 사항

CA SiteMinder for Secure Proxy Server 를 SiteMinder FSS(Federation Security Services) 환경에서 사용할 수 있도록 구성하기 전에 다음 사항을 고려하십시오.

- SiteMinder 환경을 *CA SiteMinder Federation Security Services Guide*(CA SiteMinder Federation Security Services 안내서)의 정보에 따라 구성해야 합니다. FSS 가 올바르게 구성되도록 하려면 표준 웹 에이전트를 사용하여 페더레이션 환경을 구성해야 합니다.
- SiteMinder 정책 서버 사용자 인터페이스에서 어설션을 생성하는 CA SiteMinder for Secure Proxy Server 시스템의 호스트 정보(서버 및 포트 번호)를 정의하십시오. CA SiteMinder for Secure Proxy Server 호스트는 지정하려는 페더레이션 파트너에 대한 적절한 속성 대화 상자의 "서버" 필드에서 정의합니다.

## SPS 를 페더레이션용 웹 에이전트 대체 역할로 구성

페더레이션 환경에서 CA SiteMinder for Secure Proxy Server 가 작동하도록 하기 위한 구성 프로세스는 표준 CA SiteMinder for Secure Proxy Server 구성 프로세스와 유사합니다.

CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이에 대한 전체 구성 프로세스는 다음과 같습니다.

1. CA SiteMinder for Secure Proxy Server 를 설치합니다.
2. 구성 마법사를 실행합니다.
3. server.conf 파일에서 일반적인 서버 설정을 지정합니다. 대부분의 server.conf 설정에 기본값이 있지만 로깅, 세션 체계 또는 가상 호스트 설정 등의 설정은 수정할 수 있습니다.
4. 요청이 백엔드 서버로 리디렉션되도록 proxyrules.xml 파일에서 프록시 규칙을 정의합니다.

   엔터프라이즈 생산 어설션에서 FWS 를 호스트하는 백엔드 서버로 요청을 전달하는 프록시 규칙을 정의합니다. 어설션을 소비하는 측에는 대상 리소스에 대한 사용자 액세스가 허용된 후 요청을 대상 서버로 전달하는 규칙이 있어야 합니다.
5. (선택 사항) CA SiteMinder for Secure Proxy Server 에 대한 가상 호스트를 구성하려면 Apache 웹 서버 파일(httpd.conf)을 수정합니다.

**추가 정보:**

프록시 규칙 구성 (페이지 165)
Apache 웹 서버 구성 (페이지 93)
SPS 서버 설정 구성 (페이지 95)

# 페더레이션 게이트웨이로 SPS 사용

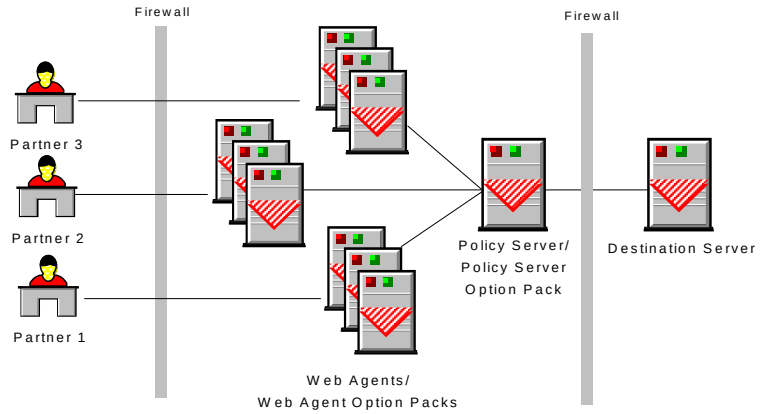
CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이를 사용하면 페더레이션 환경에 관련된 구성이 단순해집니다. 일반적인 페더레이션 환경에서는 파트너가 여러 웹 서버를 통해 통신합니다. 각 웹 서버마다 웹 에이전트와 웹 에이전트 옵션 팩을 설치하고 구성해야 합니다.

CA SiteMinder for Secure Proxy Server 를 페더레이션 게이트웨이로 사용하도록 설정하면 설치 및 설정해야 하는 구성 요소의 수가 줄어듭니다. CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이에는 CA SiteMinder for Secure Proxy Server 의 표준 구성 요소와 웹 에이전트 옵션 팩이 제공하는 페더레이션 웹 서비스 응용 프로그램이 포함되어 있습니다.

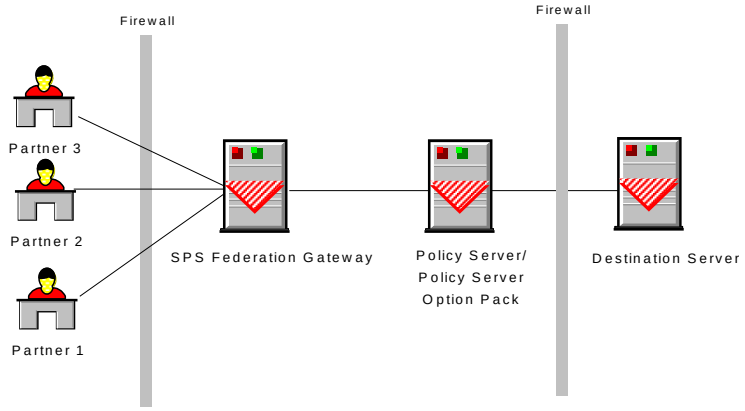
**참고:** 페더레이션 환경에서 CA SiteMinder for Secure Proxy Server 를 구성하려면 SiteMinder FSS(Federation Security Services)에 대해 잘 알고 있어야 합니다. FSS 에 대한 자세한 내용은 *CA SiteMinder Federation Security Services Guide*(CA SiteMinder Federation Security Services 안내서)를 참조하십시오.

다음 그림에서는 CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이를 사용할 때와 사용하지 않을 때의 차이를 보여 줍니다.

**Federated Environment without the SPS Federation Gateway**




**Federated Environment with the SPS Federation Gateway**



## 페더레이션 게이트웨이를 사용하기 위한 사전 요구 사항

CA SiteMinder for Secure Proxy Server 를 페더레이션 게이트웨이로 설정하려면 먼저 다음을 고려해야 합니다.

- SiteMinder 환경을 *CA SiteMinder Federation Security Services Guide*(CA SiteMinder Federation Security Services 안내서)의 정보에 따라 구성해야 합니다. 페더레이션의 정책 서버 측 구성 요소가 구성되어 있는지 확인하십시오.
- CA SiteMinder for Secure Proxy Server 를 설치하고, 관련 메시지가 표시되면 enablefederationgateway 설정을 사용하도록 설정하십시오.
- SiteMinder 정책 서버 사용자 인터페이스에서 어설션을 생성하는 CA SiteMinder for Secure Proxy Server 시스템의 호스트 정보(서버 및 포트 번호)를 정의해야 합니다. CA SiteMinder for Secure Proxy Server 호스트는 지정하려는 페더레이션 파트너에 대한 적절한 속성 대화 상자의 "서버" 필드에서 정의합니다.

## SPS 페더레이션 게이트웨이 구성

CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이는 생산자 사이트와 소비자 사이트에 있을 수 있습니다.

CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이에 대한 전체 구성 프로세스는 다음과 같습니다.

1. CA SiteMinder for Secure Proxy Server 를 설치합니다.
2. 구성 마법사를 실행합니다.
3. server.conf 파일에서 일반적인 서버 설정을 지정합니다. 대부분의 server.conf 설정에 기본값이 있지만 세션 체계 또는 가상 호스트 설정 등의 설정은 수정할 수 있습니다.

4. 요청이 백엔드 서버로 리디렉션되도록 proxyrules.xml 파일에서 프록시 규칙을 정의합니다.

   어설션을 생산하는 엔터프라이즈에서는 페더레이션 요청이 CA SiteMinder for Secure Proxy Server 에 포함된 Tomcat 서버로 전달됩니다. Tomcat 서버는 FWS 응용 프로그램을 호스트합니다. 페더레이션 요청이 처리될 때는 프록시 규칙 및 필터가 아무 영향도 주지 않습니다.

   어설션을 소비하는 엔터프라이즈에서는 대상 리소스에 대한 사용자 액세스가 허용된 후 요청을 대상 서버로 전달하는 프록시 규칙을 정의해야 합니다.

5. (선택 사항) Apache 웹 서버 파일(httpd.conf)을 수정할 수 있습니다.

## SPS 페더레이션 게이트웨이의 제한 사항

CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이를 사용하는 경우 다음 제한 사항에 유의하십시오.

- 페더레이션 리소스를 요청 중인 경우에는 사전 필터 및 사후 필터(기본 제공 필터 및 사용자 구성 필터 포함)가 실행되지 않습니다. 기본 컨텍스트에 대해 실행되는 페더레이션되지 않은 요청의 경우에는 이러한 필터가 정상적으로 실행됩니다.
- 페더레이션된 리소스를 요청 중인 경우에는 프록시 규칙이 실행되지 않습니다. 기본 컨텍스트에 대해 실행되는 페더레이션되지 않은 요청의 경우에는 이러한 규칙이 정상적으로 실행됩니다.

# 제 5 장: SPS 의 보안 영역

이 섹션은 다음 항목을 포함하고 있습니다.



## 싱글 사인온 보안 영역 개요

SSO 보안 영역은 같은 쿠키 도메인 내의 응용 프로그램 그룹 간에 구성 가능한 트러스트 관계를 제공합니다. 사용자는 같은 영역 내에서 싱글 사인온 기능을 사용하지만 다른 영역에 들어갈 때는 영역 간에 정의된 트러스트 관계에 따라 인증이 요청될 수 있습니다. 트러스트 관계에 포함된 영역은 해당 그룹의 영역에서 유효한 세션을 가진 사용자에게 인증을 요청하지 않습니다.

CA SiteMinder 웹 에이전트는 싱글 사인온 보안 영역을 구현합니다. 각 영역은 독립된 웹 에이전트 인스턴스에 있어야 하며 같은 에이전트 구성 개체를 통해 구성된 모든 웹 에이전트는 같은 싱글 사인온 영역에 속합니다.

웹 에이전트가 생성한 쿠키는 보안 영역을 식별합니다. 기본적으로 웹 에이전트는 두 개의 쿠키를 생성하는데, 하나는 SMSESSION 이라는 세션 쿠키이고 다른 하나는 SMIDENTITY 라는 아이덴티티 쿠키입니다. 보안 영역을 구성하는 경우 웹 에이전트는 쿠키 이름에 영역 가맹이 반영되도록 고유 이름을 사용하여 세션 쿠키와 아이덴티티 쿠키를 생성합니다.

**참고:** SSO 보안 영역에 대한 자세한 내용은 *CA SiteMinder Web Agent Guide*(CA SiteMinder 웹 에이전트 안내서)를 참조하십시오.

# 보안 영역의 이점

SSO 보안 영역은 CA SiteMinder 관리자가 하나의 쿠키 도메인 내에서 싱글 사인온 환경을 분할하려는 경우 사용할 수 있는 기능입니다. 예를 들어 CA.COM 도메인을 가정해 봅니다. CA SiteMinder®의 표준 SSO 기능을 사용하면 CA.COM 에서 CA SiteMinder®로 보호되는 모든 응용 프로그램은 SMSESSION 쿠키를 사용하여 싱글 사인온을 관리합니다. SSO 보안 영역이 없는 다음과 같은 시나리오를 가정해 봅니다.

1. 사용자가 APP1 응용 프로그램에 액세스합니다. 사용자는 CA SiteMinder®에서 인증 요청을 받고 CA SiteMinder®에 로그인하여 SMSESSION 쿠키를 생성합니다.
2. 사용자가 다른 응용 프로그램 APP2 에 액세스하면 CA SiteMinder®에서 다시 인증이 요청됩니다. 규칙에 따라 사용자는 APP1 사용자 자격 증명으로 APP2 에 액세스할 수 없으므로 SSO 가 실행되지 않습니다. 사용자가 로그인하여 새 SMSESSION 쿠키를 생성하면 APP2 에 대한 새 로그인 세션으로 이전 세션을 덮어쓰게 됩니다.
3. 이제 사용자가 APP1 에 반환되지만 원래 APP1 세션이 손실되었고 APP2 세션은 APP1 에 허용되지 않으므로 사용자에게 다시 인증이 요청됩니다. 즉, APP1 과 APP2 간에 SSO 가 실행되지 않으며 이는 매우 불편한 상황을 만들 수 있습니다.

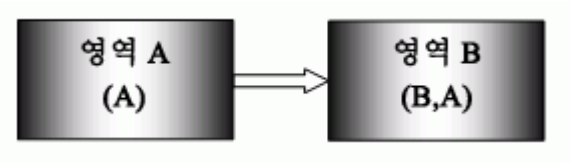
SSO 보안 영역을 사용하면 APP1 을 Z1 영역에 두고 APP2 를 Z2 영역에 둘 수 있습니다. 그런 다음 APP1 에 로그인하면 Z1SESSION 쿠키가 생성되고 APP2 에 액세스하면 Z2SESSION 쿠키가 생성됩니다. 쿠키 이름이 서로 다르므로 다른 쿠키를 덮어쓰지 않게 되고 위의 예제와 같이 사용자가 다른 응용 프로그램으로 이동할 때마다 로그인하지 않고 응용 프로그램별로 한 번만 로그인하면 됩니다.

SSO 보안 영역 기능이 나오기 전에 응용 프로그램에 대해 이와 같은 SSO 그룹화를 수행할 수 있는 방법은 네트워크 도메인, 즉 쿠키 도메인(CA1.COM, CA2.COM 등)을 여러 개 생성하고 쿠키 공급자를 통해 다중 쿠키 도메인 구성을 사용하는 것뿐이었습니다. 네트워크 도메인을 여러 개 사용하면 특정한 IT 유지 관리 및 지원이 필요하므로 이 방법은 대부분의 기업에 적합하지 않습니다.

# 보안 영역의 기본 사용 사례

싱글 사인온은 구성 가능한 트러스트 관계가 설정된 여러 보안 영역으로 분할될 수 있습니다. 예를 들어 다음과 같은 영역 A 와 영역 B 를 가정해 봅니다.

- 영역 A 에는 트러스트된 영역이 하나뿐입니다(영역 A).
- 영역 B 에는 트러스트된 영역이 두 개 있습니다(영역 A, 영역 B).




위의 그림에서 화살표는 트러스트 관계를 나타내는데, 영역 A 에서 설정된 사용자 세션을 영역 B 에서 싱글 사인온에 사용할 수 있다는 의미입니다.

이 예제의 경우 영역 A 는 관리자 전용 영역이고 영역 B 는 공용 액세스 영역으로 볼 수 있습니다. 영역 A 에서 인증된 관리자는 재인증 없이 영역 B 에 액세스할 수 있습니다. 그러나 영역 B 에서 인증된 사용자가 영역 A 에 액세스하려면 재인증이 필요합니다.

영역이 다른 사용자 세션은 서로 독립적입니다. 사용자가 처음에 영역 B 에서 인증된 후 다시 영역 B 에서 인증되는 경우를 가정해 봅니다. 그러면 서로 다른 두 세션이 생성됩니다. 실제로 두 세션에서 사용자의 아이덴티티가 다를 수 있습니다. 사용자가 영역 A 로 반환되면 해당 영역에서 설정된 세션이 사용됩니다.

사용자 세션이 없는 영역에서 싱글 사인온을 사용하여 해당 사용자의 유효성을 검사하는 경우 어떤 결과가 발생할지 생각해 보십시오. 사용자가 영역 A 에서 인증된 후 처음으로 영역 B 를 방문하면 정책 서버에 의해 업데이트된 영역 A 의 세션 정보를 기반으로 영역 B 에서 사용자 세션이 생성됩니다. 영역 A 의 사용자 세션은 사용자가 영역 A 에 반환될 때까지 업데이트되지 않습니다.

# 보안 영역에 대한 매개 변수

다음에 나열된 두 개의 싱글 사인온 매개 변수는 정책 저장소의 웹 에이전트 구성 개체에 수동으로 추가됩니다. 이러한 설정은 로컬 구성 파일에서도 사용될 수 있으며 설치 시 함께 설치되는 샘플 로컬 구성 파일에 추가되어 있습니다.

**SSOZoneName**

웹 에이전트가 지원하는 SSO(싱글 사인온) 보안 영역의 이름(대/소문자 구분)을 지정합니다. 이 매개 변수의 값은 웹 에이전트가 생성하는 쿠키 이름 앞에 추가됩니다. 이 매개 변수가 비어 있지 않으면 CA SiteMinder®는 *ZonenameCookiename* 명명 규칙을 사용하여 쿠키를 생성합니다. 기본적으로 이 매개 변수는 비어 있으며, 이 경우 SM 이 영역 이름으로 사용되며 쿠키에는 다음 기본 이름이 지정됩니다.

- SMSESSION
- SMIDENTITY
- SMDATA
- SMTRYNO
- SMCHALLENGE
- SMONDENIEDREDIR

예: 값을 Z1 로 설정하면 다음 쿠키가 생성됩니다.

- Z1SESSION
- Z1IDENTITY
- Z1DATA
- Z1TRYNO
- Z1CHALLENGE
- Z1ONDENIEDREDIR

**SSOTrustedZone**

SSO(싱글 사인온) 보안 영역에 대한 트러스트에서 트러스트된 SSOZoneName 의 정렬된 목록(대/소문자 구분)을 정의합니다. 필요한 경우 SM 을 사용하여 기본 영역을 추가하십시오. 에이전트는 항상 다른 모든 트러스트된 SSO(싱글 사인온) 영역보다 자체의 고유한 SSOZoneName 을 트러스트합니다. 기본값은 빈 값이거나 SM 또는 SSOZoneName(제공된 경우)일 수 있습니다.

# CA SiteMinder for Secure Proxy Server 보안 영역 구성

다음 방법 중 하나로 CA SiteMinder for Secure Proxy Server 의 보안 영역을 구성할 수 있습니다.

- ACO 개체를 기반으로 정책 서버로 구성된 여러 CA SiteMinder for Secure Proxy Server 서버에 보안 영역을 구성합니다.
- CA SiteMinder for Secure Proxy Server 서버 뒤에 배포된 여러 웹 서버에 보안 영역을 구성합니다.

여러 CA SiteMinder for Secure Proxy Server 서버에 보안 영역을 구성하려면 다음 단계를 수행하십시오.

1. 첫 번째 CA SiteMinder for Secure Proxy Server 서버에서 SSOZoneName 매개 변수를 구성합니다.
2. 하나의 보안 영역 또는 서로 다른 여러 보안 영역으로 그룹화할 CA SiteMinder for Secure Proxy Server 서버에서 SSOZoneName 및 SSOTrustedZone 매개 변수를 구성합니다.

CA SiteMinder for Secure Proxy Server 서버의 여러 웹 서버에 보안 영역을 구성하려면 다음 단계를 수행하십시오.

1. 보안 영역에 속해야 하는 각 웹 서버에 대해 ACO 를 생성합니다.
2. 보안 영역에 속해야 하는 단일 웹 서버 또는 웹 서버 그룹에 대해 가상 호스트를 생성합니다.
3. 각 가상 호스트가 서로 다른 보안 영역에 속하도록 고유한 ACO 가 가상 호스트를 가리키는지 확인합니다.
4. 첫 번째 웹 서버의 ACO 에서 SSOZoneName 매개 변수를 구성합니다.
5. 하나의 보안 영역이나 서로 다른 여러 보안 영역으로 그룹화할 가상 호스트의 ACO 에서 SSOZoneName 및 SSOTrustedZone 매개 변수를 구성합니다.

# 제 6 장: Apache 웹 서버 구성

이 섹션은 다음 항목을 포함하고 있습니다.



## Apache 웹 서버 구성 파일

CA SiteMinder for Secure Proxy Server 프록시 엔진은 포함된 Apache 웹 서버와 함께 작동합니다. 예를 들어 SPS 용 가상 호스트를 구성하려는 경우 Apache 웹 서버 구성을 수정할 수 있습니다.

Apache 웹 서버의 구성 파일은 다음 위치에 있는 httpd.conf 파일입니다.

*sps_home*/secure-proxy/httpd/conf/

**중요!** SPS 를 업그레이드하는 동안 또는 다른 재구성 시나리오 중에 Apache 설정을 변경할 경우에는 CA SiteMinder for Secure Proxy Server 서비스를 다시 시작해야 변경 내용이 적용됩니다. 또한 CA SiteMinder for Secure Proxy Server 도 새 설정(예: 새 포트 번호)을 사용하여 다시 시작해야 합니다.

# 제 7 장: SPS 서버 설정 구성

## SPS server.conf 파일 개요

CA SiteMinder for Secure Proxy Server 는 server.conf 파일에 포함된 설정을 통해 구성됩니다. 이 파일의 설정은 CA SiteMinder for Secure Proxy Server 가 시작 시 읽는 이름/값 쌍이나 *지시문*의 그룹입니다.

CA SiteMinder for Secure Proxy Server 는 작동 후 이 파일의 값을 확인하여 CA SiteMinder for Secure Proxy Server 웹 에이전트 로그 수준 설정이 변경되었는지 확인합니다. 변경이 감지되면 네트워크 트래픽을 중단하지 않고 CA SiteMinder for Secure Proxy Server 를 동적으로 업데이트할 수 있도록 관련 설정이 다시 로드됩니다.

server.conf 파일은 다음 디렉터리에 있습니다.

*sps_home*/secure-proxy/proxy-engine/conf

이 파일의 내용은 다음과 같은 섹션으로 나뉘어 있습니다.

- 서버 - 서버 작업, 페더레이션 게이트웨이 작업 및 SSL 에 대한 설정을 포함합니다.
- 세션 저장소 - 세션 저장소를 정의합니다.
- 서비스 디스패처 - 이 전역 서버 매개 변수에 대한 설정을 정의합니다.
- 프록시 및 리디렉션 서비스 - 프록시 서비스에 대한 연결 풀 및 필터와 리디렉션 서비스에 대한 클래스를 지정합니다.
- 세션 체계 - 세션 체계를 정의합니다.
- 사용자 에이전트 - 사용자 에이전트의 유형을 지정합니다.
- 가상 호스트 - 기본 가상 호스트 및 해당 설정을 식별합니다.

각 섹션은 XML 과 유사한 요소 태그입니다. 섹션 이름이 XML 요소의 시작 태그이고, 해당하는 끝 태그로 섹션이 끝납니다. 각 섹션에 포함된 지시문은 name=value 형식을 따릅니다.

# 기호로 시작하는 행은 주석이므로 CA SiteMinder for Secure Proxy Server 가 구성 설정을 로드할 때 이 행은 읽지 않습니다.

**참고:** Windows 시스템의 경로 이름에는 이중 백슬래시(\\)가 사용됩니다(예: \\logs\\server.log).

# server.conf 파일 수정

CA SiteMinder for Secure Proxy Server 에 대한 설정은 다음 디렉터리에 있는 server.conf 파일에서 유지 관리됩니다.

*sps_home*/secure-proxy/proxy-engine/conf

**server.conf 파일의 설정을 변경하려면**

1. 텍스트 편집기에서 파일을 엽니다.
2. 지시문을 필요한 대로 편집합니다.
3. SPS 를 다시 시작합니다.

설정이 변경됩니다.

# server.conf 파일의 일반 서버 설정

server.conf 파일의 <Server> 섹션에는 서버 커넥터, 페더레이션 및 SSL 에 대한 매개 변수가 포함되어 있습니다. 이러한 매개 변수는 다음에 나오는 단원에 설명되어 있습니다.

## HTTP 연결 매개 변수

```
#Define the listeners between #HTTP listener and proxy engine.

worker.ajp13.port=8009

worker.ajp13.host=localhost

worker.ajp13.reply_timeout=0

worker.ajp13.retries=2
```

**참고:** 커넥터 지시문의 값은 따옴표로 묶지 않습니다. 다른 유형의 지시문 값은 따옴표로 묶습니다.

이름/값 쌍은 다음과 같습니다.

**worker.ajp13.port=8009**

ajp13 커넥터의 포트를 지정합니다.

**worker.ajp13.host=localhost**

로컬 ajp13 호스트를 로컬 호스트로 지정합니다.

다음을 포함하여 HTTP 수신기와 프록시 엔진 간의 연결에 대한 추가 조정 매개 변수를 정의할 수 있습니다.

**worker.ajp13.reply_timeout**

프록시 엔진에서 받은 두 패킷 간의 최대 시간 간격(밀리초)을 지정합니다. 이 시간이 지나면 HTTP 수신기와 프록시 엔진 사이의 연결이 끊깁니다. 값을 0 으로 지정하면 응답이 수신될 때까지 무기한 기다리게 됩니다.

**기본값:** 0

**worker.ajp13.retries**

작업자가 통신 오류 시 프록시 엔진에 요청을 보내는 최대 횟수를 지정합니다.

**기본값:** 2

## server.conf 파일의 Tomcat 조정 매개 변수

SPS 에는 Tomcat 서버가 포함되어 있습니다. Tomcat 서버는 서블릿 컨테이너와 서블릿 엔진을 제공합니다.

다음은 server.conf 파일의 Tomcat 조정 섹션 중 일부분입니다.

```
#Define AJP13 tuning parameters
        #Number of request waiting in queue (queue length)
        #Number of threads created at initialization time
        #Maximum number of concurrent connections possible
        worker.ajp13.accept_count=10
        worker.ajp13.min_spare_threads=10
        worker.ajp13.max_threads=400
        worker.apj13.connection_pool_timeout=0
        worker.apj13.max_packet_size=8192
```

Tomcat 조정 지시문은 다음과 같습니다.

**worker.ajp13.accept_count**

사용 가능한 요청 처리 스레드가 모두 사용 중일 때 큐에서 대기하는 요청의 수를 정의합니다. 큐가 가득 차면 수신되는 모든 요청이 거부됩니다.

**기본값:** 10

**worker.ajp13.min_spare_threads**

언제든지 새 요청이 도착하기를 기다리는 최소 유휴 스레드 수를 정의합니다. min_spare_threads 는 0 보다 커야 합니다.

**기본값:** 10

**worker.ajp13.max_threads**

가능한 최대 동시 연결 수를 정의합니다. 풀은 이 스레드 수를 초과하여 스레드를 생성하지 않습니다.

**기본값:** 400

**worker.apj13.connection_pool_timeout**

mod-jk 를 통한 Apache 와 Tomcat 간의 유휴 연결이 시간이 만료되기 전에 연결 풀에서 유지되는 시간(초)을 정의합니다. 기본값은 연결이 시간 만료되지 않음을 나타내는 0 입니다.

**기본값**: 0

**worker.ajp13.max_packet_size**

최대 패킷 크기(바이트)를 정의합니다. 최대값은 65536 입니다.

**기본값:** 8192

## 서로 다른 Tomcat 버전의 쿠키 사양 차이 해결

Tomcat 버전 5.5 에서는 쿠키 처리 동작이 변경되었습니다. 기본적으로 Tomcat version 5.5 는 쿠키를 따옴표로 묶습니다. 이전 버전의 Tomcat 은 쿠키를 따옴표로 묶지 않습니다. Tomcat 은 쿠키를 다시 브라우저로 전송합니다. CA SiteMinder for Secure Proxy Server r12.0 SP 3 은 Tomcat 버전 5.5 를 사용합니다. 배포 환경에서 이전 버전의 SPS 에 연결해야 하는 경우 쿠키를 디코딩할 수 없습니다. 이전 버전의 CA SiteMinder for Secure Proxy Server 는 Tomcat 버전 5.0 을 사용하므로 쿠키를 따옴표로 묶지 않습니다.

서로 다른 버전의 SPS 간에 쿠키 동작이 호환되도록 하려면 server.conf 파일의 addquotestobrowsercookie 매개 변수를 "no"로 설정하십시오. 그러면 Tomcat org.apache.catalina.STRICT_SERVLET_COMPLIANCE 변수가 "TRUE"로 설정됩니다. Tomcat 은 서블릿 사양에 따라 쿠키를 구문 분석하므로 따옴표가 추가되지 않습니다. addquotestobrowsercookie 매개 변수가 "yes"로 설정된 경우 CA SiteMinder for Secure Proxy Server 는 기본값인 Tomcat 버전 5.5 쿠키 동작을 사용하도록 설정합니다.

## 쿠키 내의 등호 구문 분석

Tomcat 5.5 이상은 쿠키에 등호(=)를 추가합니다. CA SiteMinder for Secure Proxy Server 는 이러한 동작을 허용하고 등호가 포함된 쿠키 값을 구문 분석합니다. server.conf 파일의 allowequalsincookievalue 매개 변수 값은 "yes"입니다.

쿠키 값을 구문 분석하여 등호가 있을 경우 종료하려면 allowequalsincookievalue 매개 변수를 "no"로 설정하십시오.

## server.conf 파일의 페더레이션 설정

server.conf 파일의 페더레이션 설정을 사용하여 CA SiteMinder for Secure Proxy Server 가 SiteMinder 페더레이션 네트워크 내에서 페더레이션 게이트웨이의 역할을 하도록 할 수 있습니다.

다음 코드는 server.conf 파일의 <federation> 섹션 중 일부분입니다.

```
# Provide the values for the Federation related parameters here
#
# enablefederationgateway - "yes" or "no" - Enable or Disable SPS Federation Gateway
# fedrootcontext - Name of the Federation root context ("affwebservices" by default)
# authurlcontext - Path of the Authentication URL (without the jsp file name)
  (siteminderagent/redirectjsp by default)
# protectedbackchannelservices - Names of protected Backchannel services


<federation>

    enablefederationgateway="yes"
    fedrootcontext="affwebservices"
    authurlcontext="siteminderagent/redirectjsp"
        protectedbackchannelservices="saml2artifactresolution,saml2certartifactre
    solution,
    saml2attributeservice,saml2certattributeservice,assertionretriever,certassert
    ionretriever"
</federation>
```

페더레이션 매개 변수는 다음과 같습니다.

**enablefederationgateway**

CA SiteMinder for Secure Proxy Server 가 페더레이션 게이트웨이 프록시 서버의 역할을 할 수 있도록 합니다.

**제한:** yes 또는 no

이 매개 변수는 설치 중에 설정됩니다.

**fedrootcontext**

페더레이션 웹 서비스 응용 프로그램의 루트 컨텍스트를 지정합니다. 이 매개 변수는 변경하지 마십시오.

**기본값:** affwebservices

**authurlcontext**

redirect.jsp 파일의 별칭을 지정합니다. 사용자가 보호된 페더레이션 리소스를 요청할 때 어설션을 생산하는 사이트에 사용자의 SiteMinder 세션이 없으면 사용자는 redirect.jsp 파일을 가리키는 이 URL 로 보내집니다. 사용자는 생산 사이트의 웹 에이전트로 리디렉션되고 여기에서 인증 챌린지가 표시되며 성공적으로 로그인하면 세션이 설정됩니다.

**기본값:** siteminderagent/redirectjsp

**protectedbackchannelservices**

통신에 보안 백 채널이 필요한 서비스를 나열합니다.

## HttpClient 로깅

httpclientlog 매개 변수를 "yes"로 설정하여 HttpLogging 을 사용하도록 설정할 수 있습니다. 이 매개 변수는 server.conf 파일의 <Server> 섹션에 있습니다. 기본적으로 이 매개 변수는 "no"로 설정되어 있습니다.

디버깅하는 경우에만 HttpClient 로깅을 사용하도록 설정하는 것이 좋습니다. 프로덕션 환경에서 로깅을 사용하도록 설정하면 성능이 저하될 수 있습니다.

### HttpClient 로깅 구성

httpclientlogging.properties 파일에서 매개 변수 값을 설정하여 HttpClient 로깅의 다양한 측면을 구성할 수 있습니다. 이 파일은 *sps_home*\Tomcat\properties 디렉터리에 있습니다.

**중요!** 프로덕션 환경에서는 성능이 저하될 수 있으므로 HttpClient 로깅을 사용하도록 설정하지 마십시오.

httpclientlogging.properties 파일에는 다음과 같이 구성 가능한 매개 변수가 있습니다.

**java.util.logging.FileHandler.formatter**

설명: 포맷터 클래스의 이름을 지정합니다.

제한:

java.util.logging.SimpleFormatter - 로그 레코드의 간단한 요약을 작성합니다.

java.util.logging.XMLFormatter - XML 형식의 자세한 설명을 작성합니다.

기본값: java.util.logging.SimpleFormatter

**java.util.logging.FileHandler.pattern**

설명: HttpClient 로그 파일의 이름을 지정합니다.

제한:

sps_home\proxy-engine\logs\httpclient%g.log

%g 는 교환된 로그 파일의 생성 번호를 나타냅니다.

**java.util.logging.FileHandler.count**

설명: 한 주기의 출력 파일 수를 지정합니다.

기본값: 10

**java.util.logging.FileHandler.limit**

설명: 로그 파일에 작성할 수 있는 최대 바이트 수를 대략적으로 지정합니다.

제한: 0 으로 설정된 경우 제한이 없습니다.

기본값: 5,000,000

## server.conf 파일의 SSL 설정

server.conf 파일의 <sslparams> 섹션에는 CA SiteMinder for Secure Proxy Server 와 대상 서버 간에 SSL(Secure Sockets Layer) 통신이 사용되도록 설정하는 데 필요한 설정이 포함되어 있습니다.

SSL 구성 섹션은 다음과 같습니다.

```
<sslparams>
    # Set the SSL protocol version to support:SSLv3, TLSv1
    # NOTE: SSL version 2 is no longer supported versions="SSLv3"

    ciphers="-RSA_With_Null_SHA,+RSA_With_Null_MD5,-RSA_With_RC4_SHA,+RSA_With_RC
    4_MD5,+RSA_With_DES_CBC_SHA,+RSA_Export_With_RC4_40_MD5,-RSA_Export_With_DES_
    40_CBC_SHA,+RSA_Export_With_RC2_40_CBC_MD5,-DH_RSA_With_DES_CBC_SHA,-DH_RSA_W
    ith_3DES_EDE_CBC_SHA,-DH_RSA_Export_With_DES_40_CBC_SHA,-DH_DSS_With_DES_CBC_
    SHA,-DH_DSS_Export_With_DES_40_CBC_SHA,-DH_Anon_With_RC4_MD5,-DH_Anon_With_DE
    S_CBC_SHA,-DH_Anon_With_3DES_EDE_CBC_SHA,-DH_Anon_Export_With_DES_40_CBC_SHA,
    -DH_Anon_Export_With_RC4_40_MD5,-DHE_RSA_With_DES_CBC_SHA,-DHE_RSA_Export_Wit
    h_DES_40_CBC_SHA,-DHE_DSS_With_DES_CBC_SHA,-DHE_DSS_Export_With_DES_40_CBC_SH
    A"

    fipsciphers="+DHE_DSS_With_AES_256_CBC_SHA, +DHE_RSA_With_AES_256_CBC_SHA,
    +RSA_With_AES_256_CBC_SHA, +DH_DSS_With_AES_256_CBC_SHA,
    +DH_RSA_With_AES_256_CBC_SHA, +DHE_DSS_With_AES_128_CBC_SHA,
    +DHE_RSA_With_AES_128_CBC_SHA, +RSA_With_AES_128_CBC_SHA,
    +DH_DSS_With_AES_128_CBC_SHA, +DH_RSA_With_AES_128_CBC_SHA,
    +DHE_DSS_With_3DES_EDE_CBC_SHA, +DHE_RSA_With_3DES_EDE_CBC_SHA,
    +RSA_With_3DES_EDE_CBC_SHA, +DH_DSS_With_3DES_EDE_CBC_SHA"

    # Covalent SSL CA certificate bundle and certs path to be converted
    # The bundle and/or certs located at defined location will be converted
    # to binary (DER) format and loaded as SSLParams.
    # NOTE: Only put Base64 (PEM) encoded cert files/bundles in the covalent
    # certificate directory.
    cacertpath="<install-dir>\SSL\certs"
    cacertfilename="<install-dir>\SSL\certs\ca-bundle.cert"

    # This certificate configured below is used as SPS client certificate for the
    backend servers when
    # SSL client authentication is enabled.
    # Location of the Key file : <install-dir>\SSL\clientcert\key\
    # Location of public certs : <install-dir>\SSL\clientcert\certs\
    # NOTE: Only put DER encoded, password encrypted pkcs8 keyfile.
    # Client pass phrase should be encrypted using EncryptUtil tool.
    #ClientKeyFile=
    #ClientPassPhrase=

</sslparams>
```

SSL 매개 변수로는 다음이 포함됩니다.

**versions**

SPS 가 지원하는 SSL 버전을 결정합니다. 이 항목은 다음 중 하나 이상이 될 수 있습니다.

- SSLV3
- TLSV1

버전을 둘 이상 지정할 경우 값을 쉼표로 구분하십시오.

**ciphers**

사용하거나 사용하지 않도록 설정할 수 있는 암호화의 목록을 지정합니다. 사용되도록 설정된 암호화 앞에는 + 기호가 옵니다. 사용되지 않도록 설정된 암호화 앞에는 - 기호가 옵니다. 암호화를 둘 이상 지정할 경우 각 항목을 쉼표로 구분하십시오.

**cacertpath**

트러스트된 인증 기관 정보가 들어 있는 디렉터리의 경로를 지정합니다. 이 경로는 SPS 의 설치 경로에 상대적인 경로입니다. 이 값은 CA SiteMinder for Secure Proxy Server 설치 도중 구성 마법사를 실행할 때 구성되며 이 값을 변경하면 안 됩니다.

**cacertfilename**

인증 기관 인증서 번들이 들어 있는 파일의 정규화된 경로 이름을 지정합니다. 이 파일은 파일 확장명이 .cer 또는 .cert 이며 PEM 으로 인코딩되어 있어야 합니다. 또한 CA(인증 기관) 번들의 전체 경로도 포함되어야 합니다. 이 값은 CA SiteMinder for Secure Proxy Server 설치 도중 구성 마법사를 실행할 때 구성됩니다.

**ClientKeyFile**

DER 로 인코딩되고 암호로 암호화된 pkcs8 형식의 CA SiteMinder for Secure Proxy Server 클라이언트 인증서 키 파일 이름을 지정합니다. 이 파일이 다음 위치에 있는지 확인하십시오.

*<CA SiteMinder for Secure Proxy Server Installation Path>*/SSL/clientcert/key

**ClientPassPhrase**

EncryptUtil 도구를 사용하여 CA SiteMinder for Secure Proxy Server 클라이언트 인증서 키 파일에서 키를 추출하는 암호를 지정합니다.

**maxcachetime**

CA SiteMinder for Secure Proxy Server HTTPS 클라이언트가 재사용할 수 있도록 SSL 세션 ID 가 캐시되는 기간(밀리초)을 지정합니다. 사용자가 HTTPS 연결을 통해 파일을 요청하면 SSL 핸드셰이크가 수행되고 SSL 세션 ID 가 생성됩니다. 이 SSL 세션 ID 는 CA SiteMinder for Secure Proxy Server 와 백엔드 서버에서 사용자 세션을 식별하는 데 사용됩니다. 사용자에 대해 HTTPS 연결이 종료되면 CA SiteMinder for Secure Proxy Server 는 이 매개 변수에 지정된 최대 기간 동안 SSL 세션 ID 를 캐시합니다.

동일한 사용자가 백엔드 서버에 대한 새 HTTPS 연결을 요청할 때는 빠른 응답을 위해 캐시된 SSL 세션 ID 를 전송할 수 있습니다. 이 경우 사용자가 제공하는 SSL 세션 ID 와 캐시된 SSL 세션 ID 가 비교됩니다. 캐시에 사용 가능한 SSL 세션 ID 가 있으면 새 HTTPS 연결이 더 빨리 설정됩니다.

**기본값:** 120,000 밀리초

**참고**: SSL 통신을 사용하도록 설정하기 전에 SPS 를 설치하는 데 사용된 JDK 위치에 JCE(Java Cryptography Extension) Unlimited Strength Jurisdiction Policy Files 를 설치했는지 확인하십시오.

## 클라이언트 인증서 인증

CA SiteMinder for Secure Proxy Server 와 웹 서버 간의 보안 SSL 연결에 대해 클라이언트 인증서 인증을 설정할 수도 있습니다. 클라이언트 인증서 인증을 사용하도록 설정하면 CA SiteMinder for Secure Proxy Server 는 웹 서버에 리소스를 요청할 때 클라이언트 인증서를 사용하여 인증합니다.

## 사전 요구 사항

클라이언트 인증서 인증을 사용하도록 CA SiteMinder for Secure Proxy Server 를 구성하려면 먼저 다음 태스크가 완료되었는지 확인하십시오.

- 클라이언트 인증서 인증은 사용자 요청을 전달할 웹 서버에서 사용되도록 설정됩니다.
- CA SiteMinder for Secure Proxy Server 용 클라이언트 인증서는 CA SiteMinder for Secure Proxy Server 설치 시 함께 설치되는 openssl.exe 유틸리티를 사용하여 pkcs8 표준 형식으로 생성됩니다.
- 기존 클라이언트 인증서가 사용되는 경우에는 pkcs8 DER 형식으로 변환됩니다.
- 공용 인증서와 CA SiteMinder for Secure Proxy Server 클라이언트 인증서의 루트 인증서는 <*SPS_Installation_Path*>\SSL\Clientcert\certs 에 있습니다.
- 구성된 웹 서버의 공용 인증서는 <*SPS_Installation_Path*>\SSL\certs\ca-bundle.cert 에 있습니다.
- CA SiteMinder for Secure Proxy Server 클라이언트 인증서의 공용 인증서는 구성된 웹 서버의 트러스트된 저장소에 있습니다.

## 클라이언트 인증서 인증 사용

클라이언트 인증서 인증을 사용하도록 CA SiteMinder for Secure Proxy Server 를 구성하십시오.

**다음 단계를 수행하십시오.**

1. 다음 단계를 수행하여 CA SiteMinder for Secure Proxy Server 클라이언트 인증서의 개인 키 암호를 암호화합니다.
   a. 명령 프롬프트를 엽니다.
   b. <*SPS_Installation_Path*>\SSL\bin 위치로 이동합니다.
   c. 다음 명령을 실행합니다.

      **Windows**

      ```
      EncryptUtil.bat <SPSCertificatePrivateKey_Password>
      ```

      **UNIX**

      ```
      EncryptUtil.sh <SPSCertificatePrivateKey_Password>
      ```

      암호화된 암호가 표시됩니다.
2. server.conf 파일의 sslparams 섹션에서 다음 단계를 수행하여 클라이언트 인증서 인증 상세 정보를 구성합니다.
   a. ClientKeyFile 에 pkcs8 형식의 CA SiteMinder for Secure Proxy Server 클라이언트 인증서 키 파일 이름을 입력합니다.
   b. ClientPassPhrase 에 1 단계에서 생성한 암호화된 암호를 입력합니다.

   server.conf 파일에 클라이언트 인증서 인증이 구성되었습니다.
3. 클라이언트 요청을 구성된 웹 서버로 전달하도록 proxyrules.xml 파일을 구성합니다.
4. SPS 를 다시 시작합니다.

   CA SiteMinder for Secure Proxy Server 와 웹 서버 간에 클라이언트 인증서 인증이 사용되도록 설정되었습니다.

## 쿠키 내의 특수 문자에 대한 설정

server.conf 파일에는 CA SiteMinder for Secure Proxy Server 가 쿠키 매개 변수 값이 0 이 아닌 경우 해당 값을 큰따옴표로 묶는 작업을 유지하기 위한 addquotestocookie 매개 변수가 포함되어 있습니다. CA SiteMinder for Secure Proxy Server 가 쿠키 값을 백엔드로 전송하기 전에 큰따옴표로 묶지 않도록 하려면 addquotestocookie 값을 "no"로 변경하면 됩니다.

server.conf 파일에서 이 항목은 다음과 같이 나타납니다.

```
<Server>
.
.
<sslparams>
.
.
</sslparams>
#This parameter is applicable to the cookie added by backend.
#"yes"--- Default Value. Quotes will be added to the cookie parameter value
#which contains special characters if the cookie version is other than "0"
#"no" --- Quotes will not be added to the cookie.
addquotestocookie="yes"

</Server>
```

## 코드 헤더의 문자 집합 선택

requestheadercharset 매개 변수의 값을 설정하여 적절한 로캘의 문자 집합을 지정할 수 있습니다. HttpClient 응용 프로그램은 이 값을 읽고 백엔드 서버로 전송할 헤더의 인코딩 방법을 결정합니다. 가능한 값은 다음과 같습니다.

- US-ASCII - 로캘에 미국 영어가 사용되도록 지정합니다.
- Shift_JIS - 일본어 사용자 이름에 대한 지원을 포함하여 로캘에 일본어가 사용되도록 지정합니다.

기본값은 requestheadercharset="US-ASCII"입니다.

## POST 데이터 캐싱

server.conf 에는 POST 데이터 캐싱을 사용하도록 설정하고 캐시되는 데이터의 크기를 설정하기 위한 다음 두 개의 매개 변수가 포함되어 있습니다.

**enablecachepostdata**

설명: CA SiteMinder for Secure Proxy Server 가 POST 데이터를 캐시할지 여부를 지정합니다.

제한: Yes, No

기본값: Yes

**maxcachedpostdata**

설명: 캐시할 POST 데이터의 크기(KB)를 지정합니다.

제한: 없음

기본값: 1024KB

## SSL 예외 무시

백엔드 웹 서버가 CA SiteMinder for Secure Proxy Server 에 연결 끊기 알림을 반환할 때 심각하지 않은 SSL 예외를 무시하도록 CA SiteMinder for Secure Proxy Server 를 구성할 수 있습니다. 심각하지 않은 SSL 예외를 무시하려면 ignoresslbackendexception 매개 변수를 yes 로 설정하십시오.

# 사용자 지정 오류 페이지 매개 변수

CA SiteMinder for Secure Proxy Server 는 사용자 지정 오류 페이지 기능을 지원하므로 클라이언트 요청이 실패할 경우 웹 서버에 표시되는 오류 페이지를 사용자 지정할 수 있습니다.

사용자 지정 오류 페이지 섹션의 형식은 다음과 같습니다.

```
<customerrorpages>
#possible values are: "yes","no"
#default value is "no"
enable="no|yes"
#custom error pages implementation class
class="com.netegrity.proxy.errorpages.ErrorPageImpl"
#defines type of locale.
#possible values are: "0" (for Server specific), "1" (for Browser specific)
#default value is "0"
locale_type="0|1"
#this value should be the language code that will be understood by the java
#locale object, say "zh" for Chinese, "fr" for French, "es" for Spanish, "en" for
#english, etc.
#default value is "en"
locale_language="en"
#this value should be the country/region code that will be understood by the
#java locale object, say "CN" for China, "CH" for Switzerland, "AR" for
#Argentina, "US" for United States.
#default value is "US"
locale_country="US"
#display the complete call stack for exceptions
          #possible values: "true" or "false".
          #default value: "false"
          showcallstack="true"
</customerrorpages>
```

**no|yes**

CA SiteMinder for Secure Proxy Server 가 웹 서버 오류 페이지를 관리하는 방식을 지정합니다. 이 값이 yes 로 설정되어 있으면 웹 서버 오류 페이지를 사용자 지정하여 클라이언트 요청이 실패할 경우 웹 서버에 사용자 지정된 오류 페이지가 표시되도록 할 수 있습니다. 이 값이 no 로 설정되어 있으면 클라이언트 요청이 실패할 경우 웹 서버에 기본 HTTP 1.1 오류 페이지가 표시됩니다. CA SiteMinder for Secure Proxy Server 는 시작 시 이 값을 읽습니다.

**기본값**: no

**0|1**

사용자 지정 오류 페이지의 로캘을 지정합니다. 이 값이 0 으로 설정되어 있으면 CA SiteMinder for Secure Proxy Server 는 사용자 지정 오류 메시지를 표시하는 데 CA SiteMinder for Secure Proxy Server 서버의 로캘 설정을 사용합니다. 이 값이 1 로 설정되어 있으면 CA SiteMinder for Secure Proxy Server 는 사용자 지정 오류 메시지를 표시하는 데 브라우저의 로캘 설정을 사용합니다.

**기본값**: 0

**showcallstack**

디버깅할 예외를 추적하기 위해 호출 스택을 가져와야 하는지 여부를 지정합니다. 이 값이 true 로 설정되어 있으면 CA SiteMinder for Secure Proxy Server 는 호출 스택을 가져와 사용자 지정 오류 페이지에 표시합니다. 이 값이 false 로 설정되어 있으면 호출 스택이 숨겨집니다.

**기본값**: false

**중요!** 디버깅할 예외의 전체 상세 정보를 추적하려는 경우가 아니면 보안을 위해 showcallstack 옵션을 사용하도록 설정하지 않는 것이 좋습니다. 디버깅을 완료한 후에는 showcallstack 을 사용하지 않도록 설정하십시오.

## 사용자 지정 오류 메시지 사용

클라이언트 요청이 실패할 경우 웹 서버에 사용자 지정된 오류 페이지를 표시하는 기능을 사용하도록 설정하고 오류 메시지 또는 코드를 사용자 지정하십시오. 기본적으로 CA SiteMinder for Secure Proxy Server 는 모든 HTTP 1.1 오류 코드를 지원합니다.

**다음 단계를 수행하십시오.**

1. server.conf 파일을 엽니다.
2. customerrorpages 섹션으로 이동합니다.
3. 다음 명령을 설정합니다.

   ```
   enable="yes"
   ```
4. 변경 내용을 저장합니다.
5. CA SiteMinder for Secure Proxy Server 서버를 다시 시작합니다.

# 기본 사용자 지정 오류 페이지

기본적으로 CA SiteMinder for Secure Proxy Server 는 CA SiteMinder for Secure Proxy Server 와 웹 서버에서 수신될 수 있는 요청 오류의 목록을 제공합니다. SPSErrorMessages.properties 및 WebServerErrorMessages.properties 파일에 있는 오류 메시지의 형식은 다음과 같습니다.

`exception|error code=error message|URL`

설명

**exception|error code**

사용자 요청이 실패할 경우 CA SiteMinder for Secure Proxy Server 또는 웹 서버가 반환하는 예외 또는 오류 코드를 정의합니다.

**제한**: 100 자

***error message|URL***

예외 또는 오류 코드가 반환될 때 CA SiteMinder for Secure Proxy Server 또는 웹 서버가 표시하는 오류 메시지를 정의합니다. 오류 메시지는 일반 텍스트나 사용자가 사용해야 하는 대상 URL 로 지정할 수 있습니다.

**제한**: 4,096 자

## 기본 CA SiteMinder for Secure Proxy Server 오류 페이지

기본적으로 CA SiteMinder for Secure Proxy Server 서버의 구성 설정이 적절하지 않아 사용자 요청이 실패하면 CA SiteMinder for Secure Proxy Server 는 오류 페이지를 표시합니다. 각 오류 페이지는 오류 코드에 매핑됩니다. 사용자 요청이 실패하면 CA SiteMinder for Secure Proxy Server 는 오류 코드를 확인하고 해당 오류 페이지를 표시합니다.

다음 표에는 CA SiteMinder for Secure Proxy Server 서버의 구성 설정이 적절하지 않아 CA SiteMinder for Secure Proxy Server 가 표시하는 기본 오류가 나와 있습니다.

| 오류 코드 | 오류 메시지 |
| --- | --- |
| VirtualHostNotFound | "Virtual host might not have been configured properly. Verify the virtual host settings in the server.conf file" (가상 호스트가 올바르게 구성되어 있지 않을 수 있습니다. server.conf 파일에서 가상 호스트 설정을 확인하십시오.) |
| SessionSchemeNotFound | "The defined session scheme is not found. Verify the session scheme settings in the server.conf file" (정의된 세션 체계를 찾을 수 없습니다. server.conf 파일에서 세션 체계 설정을 확인하십시오.) |
| SessionCreateError | "The session store might have been corrupted during creation. Restart the SPS" (세션 저장소가 생성 중에 손상되었을 수 있습니다. SPS 를 다시 시작하십시오.) |
| SessionUpdateError | "The session store might have been corrupted during update. Restart the SPS" (세션 저장소가 업데이트 중에 손상되었을 수 있습니다. SPS 를 다시 시작하십시오.) |

| 오류 코드 | 오류 메시지 |
| --- | --- |
| WebAgentException | "CA SiteMinder for Secure Proxy Server might not be communicating with the Web agent. For more information about the error, see CA SiteMinder for Secure Proxy Server logs" (CA SiteMinder for Secure Proxy Server 가 웹 에이전트와 통신하고 있지 않을 수 있습니다. 오류에 대한 자세한 내용은 CA SiteMinder for Secure Proxy Server 로그를 참조하십시오." |
| Noodle_GenericException | "There might be an error during processing a request at noodle stage. For more information about the error, see CA SiteMinder for Secure Proxy Server logs" (요청을 처리하는 중 누들 단계에서 오류가 발생한 것 같습니다. 오류에 대한 자세한 내용은 CA SiteMinder for Secure Proxy Server 로그를 참조하십시오." |
| Noodle_FilterException | "There might be an error during pre/post processing of filters at noodle." (누들에서 필터 전처리/후처리 중 오류가 발생한 것 같습니다.) |
| Noodle_UnknownHostException | "The IP address of the targeted host could not be determined" (대상 호스트의 IP 주소를 확인하지 못했습니다.) |
| Noodle_ConnectException | "An error occurred while attempting to connect a socket to a remote address and port" (소켓을 원격 주소 및 포트에 연결하는 중 오류가 발생했습니다.) |

| 오류 코드 | 오류 메시지 |
| --- | --- |
| Noodle_NoRouteHostException | "An error occurred while attempting to connect a socket to a remote address and port because of an intervening firewall or unavailability of an intermediate router" (소켓을 원격 주소 및 포트에 연결하는 도중 중간에 방화벽이 있거나 중간 라우터를 사용할 수 없어서 오류가 발생했습니다.) |
| Noodle_InteruptedIOException | "An input or output transfer is terminated because the thread that is performing the transfer was interrupted" (전송을 수행하고 있는 스레드가 중단되어 입력 또는 출력 전송이 종료되었습니다.) |
| Noodle_SocketException | "An error occurred in the underlying protocol" (기본 프로토콜에서 오류가 발생했습니다.) |
| Noodle_NoSuchMethodException | "CA SiteMinder for Secure Proxy Server does not support the method" (CA SiteMinder for Secure Proxy Server 는 이 메서드를 지원하지 않습니다.) |
| Noodle_ProxytoProxyNotSupported | "CA SiteMinder for Secure Proxy Server does not support proxy Web Server" (CA SiteMinder for Secure Proxy Server 는 프록시 웹 서버를 지원하지 않습니다.) |
| Noodle_CouldNotConnectToBackEndServer | "Could not connect to backend web server due to lack of processing threads at the SPS" (SPS 의 처리 스레드가 부족하여 백엔드 웹 서버에 연결할 수 없습니다.) |
| Noodle_NoAvailableConnections | "There are no connections available to serve the request" (요청을 처리하는 데 사용할 수 있는 연결이 없습니다.) |

| 오류 코드 | 오류 메시지 |
| --- | --- |
| Redirect_NoTargetURLParameter | "No URL is specified during Redirect" (리디렉션 중에 지정된 URL 이 없습니다.) |
| FedRequest_Redirect_ImproperURL | "The redirect URL is not specified or is malformed. Verify the federation settings or the RelayState in the client request" (리디렉션 URL 이 지정되지 않았거나 잘못된 형식입니다. 페더레이션 설정이나 클라이언트 요청의 RelayState 를 확인하십시오.) |
| FedRequest_Redirect_RewriteLocationHeaderFailure | "Unable to create a redirect session because the memory holding the session keys might have crashed while processing the federation request" (페더레이션 요청을 처리하는 도중 세션 키를 보관하는 메모리가 손상되어 리디렉션 세션을 생성하지 못했습니다.) |
| FedRequest_CookieProcessingError | "Unable to create a cookie while processing the federation request" (페더레이션 요청을 처리하는 도중 쿠키를 생성하지 못했습니다.) |
| FedRequest_ResponseProcess_AddSessionError | "An error occurred while adding the session during the federation request processing, the memory holding the session keys might have crashed" (페더레이션 요청을 처리하는 도중 세션 추가 단계에서 오류가 발생했습니다. 세션 키를 보관하는 메모리가 손상된 것 같습니다.) |

| 오류 코드 | 오류 메시지 |
| --- | --- |
| FedRequest_ResponseProcess_LocHeaderModifyError | "An error occurred while updating the location header during federation request processing, the memory holding the session keys might have crashed" (페더레이션 요청을 처리하는 도중 위치 헤더 업데이트 단계에서 오류가 발생했습니다. 세션 키를 보관하는 메모리가 손상된 것 같습니다.) |
| SPSException | "An error occurred while processing the request. For more information about the error, see CA SiteMinder for Secure Proxy Server logs" (요청을 처리하는 중 오류가 발생했습니다. 오류에 대한 자세한 내용은 CA SiteMinder for Secure Proxy Server 로그를 참조하십시오." |

## CA SiteMinder for Secure Proxy Server 오류 페이지 수정

CA SiteMinder for Secure Proxy Server 오류 페이지의 오류 메시지를 수정할 수 있습니다.

**다음 단계를 수행하십시오.**

1. SPSErrorMessages.properties 파일을 엽니다.

   **기본 경로**: <*SPS_installation_folder*>\Tomcat\properties\

2. 수정할 오류 레코드로 이동합니다.
3. 필요한 대로 변경합니다.
4. 변경 내용을 저장합니다.

### 기본 웹 서버 오류 페이지

기본적으로 CA SiteMinder for Secure Proxy Server 는 모든 HTTP 1.1 오류 코드를 지원하며, 클라이언트 요청이 실패할 경우 웹 서버에는 기본 HTTP 1.1 오류 페이지가 표시됩니다. 사용자 지정 오류 페이지 기능을 사용하도록 설정하면 오류 페이지를 사용자 지정할 수 있으며 이 경우 클라이언트 요청이 실패하면 웹 서버에 사용자 지정 오류 상세 정보가 표시됩니다. 각 오류 페이지는 오류 코드에 매핑됩니다.

이 기능을 사용하도록 설정하고 오류 페이지를 사용자 지정하면 오류가 발생할 경우 웹 서버에 사용자 지정된 오류 페이지가 표시됩니다. 이 기능을 사용하도록 설정하고 오류 페이지를 사용자 지정하지 않으면 오류가 발생할 경우 웹 서버가 반환하는 기본 오류 페이지가 웹 서버에 표시됩니다.

### 웹 서버 오류 페이지 수정

웹 서버 오류 페이지의 오류 코드나 오류 메시지를 추가, 수정 또는 삭제할 수 있습니다. 오류 코드의 오류 메시지를 삭제하면 CA SiteMinder for Secure Proxy Server 는 다음 메시지가 포함된 페이지를 표시합니다.
"No custom message to display. Find more details in logs." (표시할 사용자 지정 메시지가 없습니다. 자세한 내용은 로그에서 확인하십시오.)

**다음 단계를 수행하십시오.**

1. WebServerErrorMessages.properties 파일을 엽니다.

   **기본 경로**: <*SPS_installation_folder*>\Tomcat\properties\

2. 수정할 오류 레코드로 이동합니다.
3. 필요한 대로 변경합니다.
4. 변경 내용을 저장합니다.

## server.conf 파일의 세션 저장소 설정

server.conf 파일의 <SessionStore> 섹션은 사용자 세션을 저장하기 위한 설정을 지정합니다. 세션 저장소 구성 형식은 다음과 같습니다.

```
<SessionStore>
    # Session Store Information
    class="com.netegrity.proxy.session.SimpleSessionStore"
    max_size="10000"
    clean_up_frequency="60"
</SessionStore>
```

SessionStore 매개 변수는 다음과 같습니다.

**class**

사용자 세션을 유지 관리하는 데 사용되는 구현을 나타냅니다. 이 값은 수정하지 마십시오.

**기본값:** com.netegrity.proxy.session.SimpleSessionStore

**max_size**

세션 저장소의 최대 크기를 지정합니다. 지정된 숫자는 메모리 내 세션 저장소의 최대 동시 세션 수입니다.

**기본값:** 10000

**clean_up_frequency**

CA SiteMinder for Secure Proxy Server 가 세션 저장소 캐시에 있는 만료된 세션을 지우기 전에 대기하는 간격(초)을 설정합니다.

**참고:** 세션 만료 시간이 길면 서버가 암호화 및 암호 해독해야 하는 세션 쿠키의 수가 줄지만 캐시에 유지되는 총 세션 수는 늘어날 수 있습니다. 사용자 연결 횟수가 적은 경우 캐시 시간은 짧게, 캐시 크기는 작게 지정하십시오. 그러나 사이트에 자주 오는 사용자가 많은 경우에는 캐시 시간을 길게, 캐시 크기를 크게 지정하십시오.

# server.conf 파일의 서비스 디스패처 설정

<ServiceDispatcher> 섹션은 CA SiteMinder for Secure Proxy Server 가 프록시 서비스를 제공하는 방식을 결정합니다. 또한 프록시 규칙 XML 구성 파일의 위치를 지정합니다.

**참고:** 이 매개 변수는 전역 서버 구성 매개 변수로, 각 개별 가상 호스트에 대해 구성되지 않습니다.

<ServiceDispatcher> 섹션은 다음과 같습니다.

```
# Service Dispatcher
# This is new since proxy 6.0
# Service Dispatcher is now a global server configuration parameter and is no longer
# configured on a per virtual host basis.
<ServiceDispatcher>
    class="com.netegrity.proxy.service.SmProxyRules" rules_file=
    "C:\Program Files\CA\secure-proxy\proxy-engine\conf\proxyrules.xml"
</ServiceDispatcher>
```

이 섹션의 매개 변수는 다음과 같습니다.

**class**

CA SiteMinder for Secure Proxy Server 가 사용자 요청을 라우팅하는 데 사용하는 서비스 디스패처를 지정합니다. 기본값을 변경하지 마십시오.

**기본값**: com.netegrity.proxy.service.SmProxyRules

**rules_file**

proxyrules.xml 의 위치를 지정합니다. file

**기본값:** *sps_home*/secure-proxy/proxy-engine/conf/proxyrules.xml

# server.conf 파일의 프록시 및 리디렉션 설정

server.conf 파일의 <Service> 섹션은 프록시 서비스와 리디렉션 서비스로 구성되어 있습니다.

CA SiteMinder for Secure Proxy Server 에 대해 미리 정의된 두 개의 프록시 서비스는 다음과 같습니다.

- 전달
- 리디렉션

파일에는 이러한 각 서비스마다 <Service name> 요소로 정의된 섹션이 있습니다. 사용자 지정 서비스는 관리자가 설정한 매개 변수를 포함하여 server.conf 파일에 유사하게 정의됩니다.

## 프록시 서비스 구성

CA SiteMinder for Secure Proxy Server 의 전달 서비스는 프록시 규칙 XML 구성 파일의 조건 및 사례에 따라 요청을 적절한 대상 서버로 전달합니다. 이 서비스의 매개 변수는 server.conf 파일의 <Service name="forward"> 섹션에 정의되어 있습니다.

대부분의 지시문은 SPS 에 의해 유지 관리되는 연결 풀을 관리합니다. 이러한 지시문은 연결을 유지 관리하고 대상 서버로 들어오는 각 요청마다 새 연결을 설정하는 오버헤드를 줄여 서버 성능을 향상시키는 데 유용합니다.

추가 지시문은 프록시 필터를 정의합니다. 프록시 필터를 추가 지시문에 정의하여 요청이 대상 서버로 전달되기 전이나 대상 서버가 SPS 에 데이터를 반환한 후에 처리 태스크를 수행할 수 있습니다. 필터 이름은 고유합니다.

다음은 <Service name="forward"> 섹션 중 일부분입니다.

**참고:** 여기에는 실제 server.conf 파일에 표시되는 대부분의 주석이 포함되어 있지 않습니다.

```
# Proxy Service
<Service name="forward">
    class="org.tigris.noodle.Noodle"
    protocol.multiple="true"
    http_connection_pool_min_size"2"
    http_connection_pool_max_size="420"
    http_connection_pool_incremental_factor="2"
    http_connection_pool_connection_timeout="1"
    http_connection_pool_wait_timeout="0"
    http_connection_pool_max_attempts="3"
    http_connection_timeout="3 minutes"
    http_connection_stalecheck="true"

    # Proxy filters may be defined here to perform pre/post processing tasks.
    # The following format must be used to configure filters:
    # filter.<filter name>.class=<fully qualified filter class name> (required)
    # filter.<filter name>.init-param.<param name1>=<param value1> (optional)
    # filter.<filter name>.init-param.<param name2>=<param value2>
    # filter.<filter name>.init-param.<param name3>=<param value3>

    # The following example illustrates the use of custom filters in a group
    # Defines filter groups with valid Custom filter names.
    #groupfilter.group1="filter1,filter2"
</Service>
```

전달 섹션의 매개 변수는 다음과 같습니다.

**class**

SPS 에 대한 전달 서비스를 제공하는 구현을 지정합니다. 이 값을 변경하지 마십시오. 이 값은 드물기는 하지만 사용자 지정 서비스가 프록시 규칙 XML 구성 파일에 지정된 요청을 전달할 수 있는 경우를 위해서만 제공됩니다.

**기본값:** org.tigris.noodle.Noodle

**protocol.multiple**

CA SiteMinder for Secure Proxy Server 가 HTTP 이외의 프로토콜을 지원하는지 여부를 나타냅니다. 다음 값 중 하나를 지정합니다.

**true**

HTTP 이외의 프로토콜이 지원됨을 나타냅니다. 현재 SPS 에서는 HTTPS 만 추가 프로토콜로 지원됩니다. true 가 이 지시문의 기본값입니다.

**false**

HTTP 프로토콜만 지원됨을 나타냅니다.

**http_connection_pool_min_size**

사용자 요청을 처리할 수 있는 단일 대상 서버에 대한 최소 연결 수를 설정합니다.

**기본값:** 2

**http_connection_pool_max_size**

CA SiteMinder for Secure Proxy Server 와 대상 서버 간의 최대 연결 수를 설정합니다.

**기본값:** 420

**중요!** CA SiteMinder for Secure Proxy Server 가 설정한 각 연결마다 소켓이 생성됩니다. UNIX 운영 체제의 경우 연결 풀의 최대 크기가 크면 많은 수의 소켓을 수용할 수 있도록 파일 설명자에 대한 제한을 늘릴 수 있습니다.

**http_connection_pool_incremental_factor**

요청을 처리하는 데 사용 가능한 모든 연결이 사용 중인 경우 CA SiteMinder for Secure Proxy Server 가 여는 대상 서버에 대한 연결 수를 설정합니다.

**기본값:** 2

**http_connection_pool_connection_timeout_unit**

만료 시간 단위를 초 또는 분으로 설정합니다.

**기본값:** Minutes

**http_connection_pool_connection_timeout**

시스템이 연결 풀의 유휴 연결을 닫기 전에 대기하는 시간(분)을 정의합니다.

**기본값:** 1

**http_connection_pool_wait_timeout**

CA SiteMinder for Secure Proxy Server 가 사용 가능한 연결을 기다리는 시간(밀리초)을 정의합니다.

**기본값:** 0

기본값 0 은 CA SiteMinder for Secure Proxy Server 가 알림을 받고 http_connection_pool_max_attempts 의 사용을 무효화할 때까지 연결을 기다리도록 지정합니다.

**http_connection_pool_max_attempts**

시스템이 연결을 가져오기 위해 시도하는 횟수를 나타냅니다. 이 지시문은 대기 만료 시간이 0 이 아닌 경우에만 사용할 수 있습니다.

**기본값:** 3

다음 값 중 하나를 지정합니다.

**0**

CA SiteMinder for Secure Proxy Server 가 무한대로 시도함을 나타냅니다.

**3**

CA SiteMinder for Secure Proxy Server 가 세 번 시도함을 나타냅니다.

**http_connection_timeout**

소켓을 생성할 때 호스트 이름 변환과 백엔드 서버와의 연결을 생성하는 데 소요되는 시간(밀리초)을 정의합니다.

**기본값:** 3 분

**참고**

- 이 시간 만료는 명시적으로 HTTP 연결을 의미하며 연결 풀을 의미하지 않습니다.
- CA SiteMinder for Secure Proxy Server 시작 중 SmSpsProxyEngine.properties 파일에 -Dhttp_connection_timeout 매개 변수를 구성한 경우 -Dhttp_connection_timeout 의 값이 http_connection_timeout 의 값보다 우선합니다.

**http_connection_stalecheck**

오래된 연결 검사를 수행해야 하는지 여부를 지정합니다. 이 값을 true 로 설정할 경우 각 요청을 실행하기 전에 오래된 연결 검사가 수행됩니다. 이 값을 false 로 설정할 경우 백엔드 웹 서버에서 닫힌 연결을 통해 요청을 실행하면 I/O 오류가 발생할 수 있습니다.

**기본값**: true

**filter.*filter name*.class=*fully qualified filter class name***

프록시 규칙에서 호출된 각 고유 필터에 server.conf 파일에 구성된 필터를 지정합니다.

**예:** filter.PreProcess.class=SampleFilter

**filter.*filter name*.init-param.*param name1=param value1***

필터가 필터 API 를 사용하여 정의되는 방식에 따라 필터에 대한 초기화 매개 변수를 지정합니다. server.conf 파일을 구성하여 각 필터에 대한 매개 변수를 정의합니다.

**예:** filter.PreProcess.init-param.param1=value1

**groupfilter.<groupname> = "*filtername1,filtername2,…….filtername*"**

지정된 프록시 규칙에 대해 하나 이상의 필터를 구현할 필터 그룹을 지정합니다. CA SiteMinder for Secure Proxy Server 는 그룹 필터에 선언된 필터 이름을 읽고 해당 필터를 한 체인에서 처리합니다. 그룹 필터 이름은 proxyrules.xml 의 필터 이름과 유사하게 사용될 수 있습니다. CA SiteMinder for Secure Proxy Server 가 그룹 필터를 처리할 때는 사후 필터보다 사전 필터가 먼저 처리되며 이는 그룹 필터에 필터가 정의된 순서가 그 반대인 경우에도 해당됩니다.

다음과 같은 제한 사항이 적용됩니다.

- 필터 이름은 유효하고 고유해야 합니다.
- 그룹 필터 이름은 고유해야 합니다. 여러 그룹에 동일한 그룹 이름을 지정할 경우 마지막 그룹만 유지됩니다.
- 그룹 필터 이름과 필터 이름은 달라야 합니다.

**예**:

groupfilter.BatchProcess="SampleFilter1, SampleFilter2, SampleFilter3"

## 연결 풀링 권장 사항

연결 풀링은 CA SiteMinder for Secure Proxy Server 성능을 관리하는 데 중요한 부분입니다. 엔터프라이즈에서 CA SiteMinder for Secure Proxy Server 가 가능한 최상의 서비스를 제공하도록 하려면 연결에 연결 유지 메시지가 사용되도록 설정된 상태로 대상 서버를 구성해야 합니다. 대상 서버에 대한 연결 유지 메시지를 사용하도록 설정하면 CA SiteMinder for Secure Proxy Server 가 연결 풀링 기능을 사용할 수 있습니다.

연결 유지 메시지는 웹 서버 유형마다 다르게 관리됩니다.

연결 유지 메시지를 사용하도록 설정할 뿐 아니라 대상 서버와 SPS 에 다음 설정도 사용하는 것이 좋습니다. 다음 표에는 시간 만료 및 연결 풀 권장 사항이 나와 있습니다.

| 설정 | HTTP | HTTPS |
|---|---|---|
| 대상 서버 연결 유지 최대 요청 수 (http_connection_pool_max_attempts) | 제한 없음 | 제한 없음 |
| 대상 서버 만료 시간 | 시간 만료되지 않음 | HTTP 연결 풀 만료 시간보다 크거나 같음 |

| 설정 | HTTP | HTTPS |
|---|---|---|
| 보안 프록시 서버 HTTP 연결 풀 만료 시간 단위<br>(http_connection_pool_connection_timeout_unit) | 초 또는 분으로 설정(기본값: 분) | 초 또는 분으로 설정(기본값: 분) |
| 보안 프록시 서버 HTTP 연결 풀 만료 시간<br>(http_connection_pool_connection_timeout) | 1 분 | 1 분 |
| 보안 프록시 서버 HTTP 연결 풀 대기 만료 시간<br>(http_connection_pool_wait_timeout) | 0<br>알림이 있을 때까지 대기 | 0<br>알림이 있을 때까지 대기 |
| 보안 프록시 서버 HTTP 연결 풀 최대 시도 횟수<br>(http_connection_pool_max_attempts) | 3<br>이 값은 HTTP 연결 풀 만료 시간이 0 보다 큰 경우에만 유용합니다. | 3<br>이 값은 HTTP 연결 풀 만료 시간이 0 보다 큰 경우에만 유용합니다. |
| 보안 프록시 서버 HTTP 연결 만료 시간<br>(http_connection_timeout) | 3 분 | 3 분 |

## 리디렉션 서비스 구성

CA SiteMinder for Secure Proxy Server 의 리디렉션 서비스는 대상 서버에 요청을 전송합니다. 전달 서비스와 달리 이후 요청은 SPS 가 아니라 대상 서버가 처리합니다.

리디렉션 서비스의 형식은 다음과 같습니다.

```
<Service name="redirect">
        class=com.netegrity.proxy.service.RedirectService
</Service>
```

지시문은 다음과 같습니다.

**class**

리디렉션된 요청을 처리하는 구현을 나타냅니다. 이 지시문은 수정하면 안 됩니다.

**기본값:** com.netegrity.proxy.service.RedirectService

## 연결 지향 연결 풀 구성

웹 서버가 연결 지향 인증 체계를 사용하는 경우 안전한 요청 전달 처리를 위해 연결 지향 연결 풀을 구성하십시오.

**중요!** 연결 지향 연결 풀은 가급적 구성하지 않는 것이 좋습니다.

**다음 단계를 수행하십시오.**

1. httpd.conf 파일에서 JK 환경 변수 REMOTE_PORT 의 값이 설정되어 있는지 확인합니다.
2. server.conf 를 열고 <Service name="forward"> 섹션에 다음 행을 추가합니다.

```
# Pool configuraiton for connection oriented authentication backend
    # connections eg: NTLM.
    <connection-pool name="connection oriented authentication">
    connection-timeout="connection_timeout_value"
    max-size="maximum_connections"
    enabled="yes|no"
</connection-pool>
```

***connection_timeout_value***

연결 만료 시간(초)을 정의합니다. 될수록 낮은 값을 설정하는 것이 좋습니다.

**기본값**: 5

***maximum_connections***

연결 풀의 연결 수를 정의합니다.

**기본값**: 50

***yes|no***

연결 지향 연결 풀의 상태를 지정합니다. 연결 지향 연결 풀을 사용하도록 설정하려면 값을 yes 로 설정합니다.

**기본값**: yes

3. proxyrules.xml 을 열고 전달 규칙에 connection-auth 특성을 추가합니다.

```
예: <nete:forward connection-auth="yes">hostname:port$1</nete:forward>
```

# server.conf 파일의 세션 체계 설정

세션 체계는 사용자의 아이덴티티가 유지 관리되는 방식을 결정하여 세션 과정 중에 싱글 사인온을 제공합니다. 각각의 잠재적 세션 체계가 server.conf 파일의 SessionScheme 섹션에 포함되어야 합니다. 세션 체계는 세션의 동작을 정의하는 Java 클래스 파일과 연결되어야 합니다. 특정 유형의 사용자 에이전트에 대해 세션 체계가 지정되지 않은 경우 기본 세션 체계가 사용됩니다.

엔터프라이즈 트랜잭션에서 중요한 한 가지 과제는 사용자 세션을 유지 관리하는 것입니다. SiteMinder 는 쿠키를 사용하여 세션 정보를 캡슐화합니다. SiteMinder 와 달리 CA SiteMinder for Secure Proxy Server 는 여러 방법을 사용하며, 쿠키를 사용하지 않고 세션을 유지 관리하기 위한 대체 방법을 지원하는 일련의 API 를 제공합니다. 쿠키를 사용하지 않는 세션 체계에는 CA SiteMinder for Secure Proxy Server 메모리 내 세션 저장소에서 유지 관리되는 세션 정보를 참조하는 일종의 토큰이 사용됩니다. 세션 저장소는 CA SiteMinder for Secure Proxy Server 서버의 메모리에 상주하며 서버를 다시 시작하면 지울 수 있습니다.

CA SiteMinder for Secure Proxy Server 는 server.conf 파일에서 구성할 수 있는 다음과 같은 세션 체계를 기본적으로 제공합니다. 이러한 체계는 server.conf 파일에 정의된 각 가상 호스트의 사용자 에이전트 유형에 연결되어 있을 수 있습니다. 세션 체계와 사용자 에이전트 유형의 연결을 세션 체계 매핑이라고 합니다.

CA SiteMinder for Secure Proxy Server 에는 다음과 같은 체계가 포함되어 있습니다.

- 기본 체계
- SSL ID
- IP 주소

- 미니 쿠키
- 단순 URL 다시 쓰기
- 장치 ID

**참고**

- 추가 사용자 지정 세션 체계를 생성하려면 세션 체계 API 를 사용하면 됩니다. 세션 체계 API 를 사용하여 고유한 세션 체계를 생성하는 경우 사용자 지정 세션 체계와 연결된 이름 및 Java 클래스에 대한 특정 정보를 포함하여 <SessionScheme> 섹션을 server.conf 파일에 추가해야 합니다.
- makefile.cygwin 을 사용하여 사용자 지정 세션 체계를 생성하려면 cygwin 지침에 따라 JAVA_HOME 및 SPS_HOME 의 값을 설정하십시오. 예를 들어 Windows 2008 R2 컴퓨터에서 makefile.cygwin 을 사용하려면 다음 값을 설정하면 됩니다.

  ```
  JAVA_HOME=C:/Progra~2/Java/jdk1.7.0_03

  SPS_HOME=C:/Progra~2/CA/secure-proxy
  ```

## 사용자 세션 설정

사용자 세션을 설정하려면 다음과 같이 고유한 단계를 거칩니다.

1. 검색 단계

   이 세션 단계 중에 CA SiteMinder for Secure Proxy Server 는 사용자 에이전트 유형을 기반으로 적절한 세션 키를 찾습니다. 세션 키는 SiteMinder 쿠키이거나 CA SiteMinder for Secure Proxy Server 메모리 내 세션 저장소의 적절한 정보를 가리키는 토큰입니다. 이전에 설명한 것처럼 토큰은 미니 쿠키, SSL ID, 장치 ID 또는 다른 토큰 형식일 수 있습니다. 세션 키를 식별할 수 없으면 CA SiteMinder for Secure Proxy Server 의 웹 에이전트는 인증 및 권한 부여에 대한 요청을 승계하여 전달하고 사용자의 아이덴티티 및 권한을 설정합니다.

2. 에이전트 처리 단계

   CA SiteMinder for Secure Proxy Server 에는 SiteMinder 와 통신하는 웹 에이전트가 포함되어 있습니다. 이 웹 에이전트는 SiteMinder 세션 정보를 암호 해독하고 SiteMinder 를 사용하여 세션의 유효성을 검사하는 작업을 수행합니다. 사용자 요청이 SMSESSION 쿠키와 함께 수신되거나 CA SiteMinder for Secure Proxy Server 의 세션 저장소에 사용자 세션이 있는 경우 웹 에이전트는 SiteMinder 를 사용하여 사용자 요청의 유효성을 검사합니다.

3. 리버스 프록시 단계

   사용자 세션의 유효성이 검사된 후 이 단계에서는 CA SiteMinder for Secure Proxy Server 가 정의된 서비스(전달, 리디렉션 또는 기타 서비스) 중 하나를 사용하여 사용자의 요청을 처리합니다. 이 단계에서 CA SiteMinder for Secure Proxy Server 의 작업은 프록시 규칙 XML 구성 파일에 포함된 프록시 규칙에 따라 결정됩니다.

   **참고:** URL 다시 쓰기 세션 체계의 경우 콘텐츠가 사용자에게 다시 전송되기 전에 이 단계에서 다시 쓰기 메커니즘으로 전달됩니다.

## 기본 세션 체계

기본 세션 체계는 사용자 에이전트 유형에 지정된 다른 체계가 없는 경우 CA SiteMinder for Secure Proxy Server 가 사용자 세션을 설정 및 유지 관리하는 데 사용하는 체계입니다. <SessionScheme> 요소에는 사용자 에이전트 유형에 체계를 할당할 때 세션 체계를 식별하는 데 사용되는 name 특성이 포함되어 있습니다. server.conf 파일에는 기본 세션 체계 구성이 포함되어야 합니다.

사용 가능한 세션 체계를 사용하도록 기본 세션 체계를 구성할 수 있습니다.

기본 세션 체계 섹션의 형식은 다음과 같습니다.

```
#Session Schemes
<SessionScheme name="default">
    class="com.netegrity.proxy.session.SessionCookieScheme"
    accepts_smsession_cookies="true"
</SessionScheme>
```

<SessionScheme> 요소에는 다음과 같은 지시문이 있습니다.

**class**

기본 세션 체계가 포함된 Java 클래스를 나타냅니다.

**기본값:** com.netegrity.proxy.session.SSLIdSessionScheme

**accepts_smsession_cookies**

사용자 에이전트 유형이 SiteMinder 쿠키 세션 체계와 연결된 경우 해당 사용자 에이전트 유형을 통해 리소스에 액세스하는 사용자가 기존 SiteMinder 쿠키를 사용하여 세션을 유지 관리함을 나타냅니다.

SiteMinder 는 쿠키를 사용하여 세션을 추적하므로 SPS 는 쿠키 체계를 지원합니다. SMSESSION 쿠키가 허용되는지 여부를 나타냅니다.

다음 값 중 하나를 지정합니다.

**true**

SMSESSION 쿠키가 허용되며 세션 체계에서 사용됨을 나타냅니다.

**false**

세션 체계에서 SMSESSION 쿠키가 지원되지 않음을 나타냅니다.

## 기본 세션 체계 지정

기본 세션 체계는 사용자 에이전트 유형에 대해 다른 세션 체계가 지정되지 않은 경우에 사용됩니다.

기본 세션 체계 지시문은 다음과 같습니다.

**defaultsessionscheme**

기본 체계로 SiteMinder 쿠키 세션 체계가 아닌 다른 세션 체계를 지정합니다. 이 항목을 수정하여 사용자가 원하는 세션 체계를 기본 세션 체계로 포함할 수 있습니다.

**기본값:** default

**enablewritecookiepath**

CA SiteMinder for Secure Proxy Server 가 프록시 뒤에 있는 서버가 설정한 URI 에서 초기 요청의 URI 로 쿠키 경로를 다시 쓰도록 합니다.

**기본값:** no

**enablewritecookiedomain**

CA SiteMinder for Secure Proxy Server 가 프록시 뒤에 있는 서버가 설정한 도메인에서 초기 요청의 도메인으로 쿠키 도메인을 다시 쓰도록 합니다.

**기본값:** no

## SSL ID 세션 체계

SSL(Secure Sockets Layer) 연결에는 SSL 연결이 시작될 때 생성된 고유 식별자가 포함되어 있습니다. CA SiteMinder for Secure Proxy Server 는 CA SiteMinder for Secure Proxy Server 메모리 내 세션 저장소에서 유지 관리되는 사용자에 대한 세션 정보를 참조할 때 이 고유 ID 를 토큰으로 사용합니다. 이 체계는 사용자의 세션을 유지 관리하기 위한 메커니즘으로 쿠키를 제거합니다.

SSL ID 세션 체계의 제한 사항은 CA SiteMinder for Secure Proxy Server 와의 초기 연결 시 SSL 세션 ID 가 설정된다는 점입니다. 사용자의 SSL 세션이 중단되고 새 SSL 연결이 설정될 경우 새 SSL 연결에서는 새 서버에 연결하게 되므로 해당 서버가 동일한 시스템에 있는 가상 서버이더라도 사용자 인증 및 권한 부여를 다시 수행해야 합니다. 이는 또한 HTML 양식 인증 체계에 사용되는 양식을 보호된 리소스와 동일한 호스트 이름에서 제공해야 함을 의미합니다.

## SSL ID 세션 체계 구성

SSL ID 섹션에는 SSL ID 를 사용하는 세션 체계가 나열됩니다.

SSL ID 세션 체계는 SPS 와 함께 패키지로 제공되는 Java 클래스를 사용하여 사용자 지정 작업 없이 지원될 수 있습니다.

**중요!** SSL ID 인증 체계를 사용하려면 Apache 웹 서버의 httpd.conf 파일에 있는 설정도 사용하도록 설정해야 합니다.

SSL ID 세션 체계의 형식은 다음과 같습니다.

```
<SessionScheme name="ssl_id">
    class="com.netegrity.proxy.session.SSLIdSessionScheme"
    accepts_smsession_cookies="false"
</SessionScheme>
```

ssl_id 에 대한 지시문은 다음과 같습니다.

**class**

SSL ID 세션 체계를 처리하는 Java 클래스를 지정합니다.

**기본값:** com.netegrity.proxy.session.SSLIdSessionScheme

**accepts_smsession_cookies**

SMSESSION 쿠키가 허용되는지 여부를 나타냅니다. 다음 값 중 하나를 지정합니다.

**true**

SMSESSION 쿠키가 허용되며 세션 체계에서 사용됨을 나타냅니다.

**false**

세션 체계에서 SMSESSION 쿠키가 지원되지 않음을 나타냅니다.

## SSL ID 체계의 httpd.conf 파일 수정

server.conf 파일에서 SSL ID 세션 체계를 구성할 뿐 아니라, SSL 을 사용하도록 Apache 웹 서버 httpd.conf 파일도 수정해야 합니다.

**SSL ID 체계의 httpd.conf 파일을 수정하려면**

1. *sps_home*/secure-proxy/httpd/conf 디렉터리에 있는 httpd.conf 파일을 엽니다.
2. 이 파일에서 다음 행을 찾습니다.

   #SSLOptions +StdEnvVars +ExportCertData +CompatEnvVars

3. 행의 처음 부분에서 # 기호를 삭제합니다.

   **참고**: CA SiteMinder for Secure Proxy Server r6.0 SP 3 이상의 경우 행이 다음과 같이 되도록 +CompateEnvVars 도 제거하십시오.

   SSLOptions +StdEnvVars +ExportCertData

4. httpd.conf 파일을 저장합니다.
5. SPS 를 다시 시작합니다.

## IP 주소 세션 체계

IP 주소가 고정된 환경에서 CA SiteMinder for Secure Proxy Server 는 IP 주소를 사용하여 세션 저장소에 있는 사용자의 세션 정보를 참조할 수 있습니다. 이 체계는 쿠키를 제거하지만, 사용자에게 고정 IP 주소가 할당되는 환경에서만 이 체계를 사용할 수 있다는 단점이 있습니다.

## 미니 쿠키 세션 체계

기존 SiteMinder 쿠키 기반 세션 체계의 단점 중 하나는 쿠키 크기에 있습니다. 각 요청과 함께 전송되는 데이터의 양이 증가하면 무선 전화와 같은 일부 장치 유형의 경우 액세스 비용도 증가합니다.

미니 쿠키는 SiteMinder 메모리 내 저장소의 세션 정보를 참조하는 데 사용할 수 있는 토큰이 포함된 약 10 바이트 크기의 작은 쿠키입니다. 미니 쿠키는 표준 SiteMinder 쿠키의 크기에 비해 매우 작으며 표준 SiteMinder 쿠키 대신 사용할 수 있습니다.

### 미니 쿠키 세션 체계 구성

미니 쿠키 세션 체계는 세션 정보를 CA SiteMinder for Secure Proxy Server 메모리 내 세션 저장소에 저장하고 CA SiteMinder for Secure Proxy Server 가 사용자에게 반환하는 암호화된 토큰이 포함된 쿠키를 생성합니다.

이 섹션의 형식은 다음과 같습니다.

```
<SessionScheme name="minicookie">
    class="com.netegrity.proxy.session.MiniCookieSessionScheme"
    accepts_smsession_cookies="false"
    # The name of the small cookie to be stored in the client.
    cookie_name="SMID"
</SessionScheme>
```

미니 쿠키 세션 체계의 지시문은 다음과 같습니다.

**class**

세션 체계를 정의하는 Java 클래스를 지정합니다. SPS 와 함께 제공되는 미니 쿠키 세션 체계를 사용하려는 경우에는 이 지시문을 수정하지 않습니다.

**기본값:** com.netegrity.proxy.session.MiniCookieSessionScheme

**accepts_smsession_cookies**

SMSESSION 쿠키가 허용되는지 여부를 나타냅니다. 다음 값 중 하나를 지정합니다.

**true**

SMSESSION 쿠키가 허용되며 세션 체계에서 사용됨을 나타냅니다.

**false**

세션 체계에서 SMSESSION 쿠키가 지원되지 않음을 나타냅니다. 세션 체계에 미니 쿠키 세션만 사용되는지 확인하려면 이 설정을 사용하십시오.

**cookie_name**

사용자 세션에 대한 토큰이 포함된 미니 쿠키의 이름을 나타냅니다.

**참고:** 이 이름은 싱글 사인온을 제공하는 모든 CA SiteMinder for Secure Proxy Server 에 대해 동일한 값을 사용하여 구성되지 않습니다.

## 단순 URL 다시 쓰기 세션 체계

단순 URL 다시 쓰기는 요청된 URL 에 토큰을 추가하여 사용자 세션을 추적하기 위한 방법입니다. 이 토큰은 메모리 내 세션 저장소에서 세션 정보를 검색하는 데 사용됩니다.

### 단순 URL 다시 쓰기 구성

simple_url 체계는 사용자 지정 작업 없이 수행할 수 있는 단순 URL 다시 쓰기를 지원합니다.

**참고:** CGI 기반 및 FCC 기반 암호 체계는 simple_url 세션 체계와 함께 지원됩니다.

### 예

사용자가 호스트에 액세스하여 단순 URL 다시 쓰기 세션 체계를 통해 사용자 세션이 설정됩니다. 초기 요청은 다음 예와 같을 수 있습니다.

http://banking.company.com/index.html

사용자가 적절한 자격 증명을 제공하여 인증을 받고 권한이 부여되면 사용자가 요청한 URL 이 다음과 유사한 형식으로 다시 써져서 사용자에게 반환됩니다.

http://banking.company.com/SMID=*nnnnnnnnnn*/index.html

**nnnnnnnnnn**

CA SiteMinder for Secure Proxy Server 가 사용자 세션을 식별하는 데 사용하는 무작위로 생성된 해시 토큰을 나타냅니다.

**중요!** 단순 URL 다시 쓰기 세션 체계가 작동하기 위해서는 엔터프라이즈에 정의된 모든 링크가 상대 링크여야 합니다. 링크가 절대 링크인 경우에는 단순 URL 다시 쓰기 체계가 실패합니다. 또한 요청이 전달될 때 CA SiteMinder for Secure Proxy Server 가 URL 에 추가하는 토큰이 제거됩니다. 이 토큰은 백엔드 서버 처리에 방해가 되지 않도록 CA SiteMinder for Secure Proxy Server 상호 작용 수준에서만 추가됩니다.

SimpleURL 체계의 형식은 다음과 같습니다.

```
<SessionScheme name="simple_url">
    class="com.netegrity.proxy.session.SimpleURLSessionScheme"
    accepts_smsession_cookies="false"
    session_key_name="SMID"
</SessionScheme>
```

SimpleURL 체계의 지시문은 다음과 같습니다.

**class**

세션 체계를 정의하는 Java 클래스를 지정합니다. 쿠키를 사용하지 않는 다시 쓰기 세션 체계를 사용하려는 경우에는 이 지시문을 수정하지 않습니다.

**기본값:** com.netegrity.proxy.session.SimpleURLSessionScheme

**accepts_smsession_cookies**

SMSESSION 쿠키가 허용되는지 여부를 나타냅니다. 다음 값 중 하나를 지정합니다.

**true**

SMSESSION 쿠키가 허용되며 세션 체계에서 사용됨을 나타냅니다.

**false**

세션 체계에서 SMSESSION 쿠키가 지원되지 않음을 나타냅니다. 세션 체계에 쿠키를 사용하지 않는 다시 쓰기 쿠키 세션만 사용되는지 확인하려면 이 설정을 사용하십시오.

**session_key_name**

SMID(SiteMinder ID) 세션 식별자를 지정합니다

**참고:** 쿠키를 사용하지 않는 페더레이션 트랜잭션이 CA SiteMinder for Secure Proxy Server 페더레이션 게이트웨이에 의해 처리되고 simple_url 세션 체계가 사용될 때는 SMID 가 URI 에 추가되지 않고 요청에 쿼리 매개 변수로 추가됩니다.

## 다시 쓰기 가능한 세션 체계에 대해 쿠키를 사용하지 않는 페더레이션을 사용하도록 설정

페더레이션 환경에서 CA SiteMinder for Secure Proxy Server 가 단순 URL 세션 체계와 같이 다시 쓰기 가능한 세션 체계를 사용할 수 있도록 하려면 쿠키를 사용하지 않는 페더레이션을 구성하십시오.

### 쿠키를 사용하지 않는 페더레이션을 구성하려면

1. 텍스트 편집기에서 server.conf 파일을 엽니다. 이 파일은 *sps_home*/secure-proxy/proxy-engine/conf 디렉터리에 있습니다.
2. FWS 서비스를 제공하는 가상 호스트에 대한 가상 호스트 섹션에 다음 코드를 추가합니다.

   cookielessfederation="yes"
3. 파일을 저장합니다.
4. SPS 를 다시 시작합니다.

**참고:** SPS 페더레이션 게이트웨이에는 CookielessFedFilter 와 같은 별도의 사후 필터를 사용하도록 설정할 필요가 없습니다. 이 기능은 페더레이션 게이트웨이 기능을 사용하도록 설정하면 기본적으로 제공됩니다. SPS 가 페더레이션 게이트웨이로 작동하지 않는 경우에는 이 사후 필터를 사용하도록 설정해야 합니다.

### 단순 URL 세션 체계의 FWS 리디렉션 다시 쓰기

페더레이션 환경에 CA SiteMinder for Secure Proxy Server 를 배포할 경우 사이트 제작 어설션 측에서 사용할 수 있는 세션 체계 중 하나는 단순 URL 세션 체계입니다. 이 체계를 사용할 경우 세션 키가 링크에 추가되도록 사용자를 적절한 사이트로 연결하는 링크를 다시 써야 할 수 있습니다. SiteMinder 설명서에서는 이러한 SAML 1.x 용 링크를 사이트 간 전송 URL 이라고 합니다. SAML 2.0 의 경우에는 이러한 링크를 원치 않는 응답 또는 AuthnRequest 링크라고 합니다.

URL 기준에 세션 키 정보가 추가되도록 링크를 다시 쓰기 위해 샘플 사후 필터인 RewriteLinksPostFilter 가 CA SiteMinder for Secure Proxy Server 필터 예제와 함께 제공됩니다. 이 필터를 컴파일하여 사이트 간 전송 URL, 원치 않는 응답 또는 AuthnRequest 로의 전달을 처리하는 적절한 프록시 규칙에 추가할 수 있습니다.

CA SiteMinder for Secure Proxy Server 와 함께 제공되는 RewriteLinksPostFilter 는 샘플 필터입니다. 이 필터를 사용하려면 요구 사항에 맞게 구성해야 합니다.

**참고:** SPS 페더레이션 게이트웨이를 사용하는 트랜잭션에 simple_url 세션 체계를 사용하는 경우 세션 키(SMID)는 URI 에 추가되는 대신 요청에 쿼리 매개 변수로 추가됩니다. 그러나 백엔드 서버에서 최종 대상 리소스에 액세스하는 경우에는 SMID 가 URI 에 추가됩니다.

## 무선 장치 ID 세션 체계

일부 무선 장치에는 고유한 장치 식별 번호가 있습니다. 이 번호는 리소스에 대한 요청과 함께 헤더 변수로 전송됩니다. CA SiteMinder for Secure Proxy Server 는 이 장치 ID 를 토큰으로 사용하여 세션 저장소의 세션 정보를 참조할 수 있습니다.

장치 ID 는 무선 장치 공급업체에 따라 다르므로 server.conf 파일에서 장치 ID 세션 체계를 정의하십시오. 이 체계에는 클래스 정보와 공급업체 관련 장치 ID 로 설정된 device_id_header_name 지시문을 포함해야 합니다.

장치 ID 체계의 형식은 다음과 같습니다.

```
<SessionScheme name="device_id">
    class="com.netegrity.proxy.session.DeviceIdSessionScheme"
    accepts_smsession_cookies="false"
    device_id_header_name="vendor_device_id_header_name"
</SessionScheme>
```

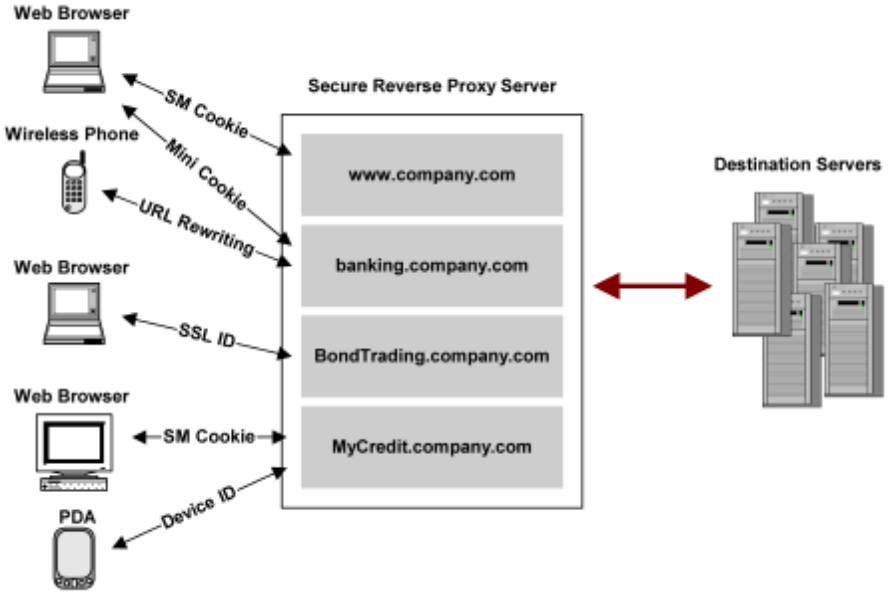
## 각 세션 체계의 사용 사례

다음 표에서는 각 세션 체계가 사용되는 시나리오를 보여 줍니다. 세션 체계는 가상 호스트에 있는 리소스의 중요도를 기반으로 합니다.

| 세션 체계 | 보안 수준 | 권장 사항 |
|---|---|---|
| SSL 세션 ID | 높음 | 이 체계는 사용자 세션을 유지하기 위한 완전하고 높은 보안 수단을 제공합니다. 이 체계는 사용 가능한 체계 중 가장 안전하지만 확장성에 제한이 있습니다. 모든 콘텐츠는 SSL 을 통해 제공되어야 하고, 사용자가 계속 동일한 CA SiteMinder for Secure Proxy Server 서버에 액세스해야 세션이 유지됩니다. 또한 일부 브라우저(일부 IE 버전)는 기본적으로 2 분 후에 SSL 세션을 종료합니다. 이 체계는 보안 요구 사항이 높은 인트라넷 및 엑스트라넷 응용 프로그램에 적합합니다. |
| SiteMinder 쿠키 | 중간 또는 높음 | 이 체계는 기존 SiteMinder 세션 메커니즘으로, 많은 엔터프라이즈 배포 환경에서 보안 수준이 높은 것으로 입증되었습니다.<br>보안을 최대화하기 위해 WebAgent SecureCookie 설정은 "Yes"로 설정되어 있습니다. |
| IP 주소 | 낮음 | 이 체계는 사용자가 HTTP GET 을 사용하여 보호된 리소스에서 정보를 검색하며 HTTP POST 를 사용하여 보안 응용 프로그램으로 정보를 전송하지 않는 응용 프로그램에만 사용됩니다. 적절한 응용 프로그램의 예로는 온라인 라이브러리를 들 수 있습니다. 적절하지 않은 응용 프로그램의 예로는 채권 거래 응용 프로그램을 들 수 있습니다. |
| 미니 쿠키 | 중간 또는 높음 | 이 체계는 사용자 클라이언트가 쿠키를 허용하지만 속도 및 대역폭이 제한된 연결을 통해 응용 프로그램에 액세스하는 응용 프로그램에 적합합니다.<br>보안을 최대화하기 위해 WebAgent SecureCookie 설정은 "Yes"로 설정되어 있습니다. |
| 단순 URL 다시 쓰기 | 중간 | 이 체계는 쿠키를 지원하지 않거나 쿠키를 사용하지 않으려는 환경에 적합합니다. |
| 장치 ID | 중간 | 이 체계는 사용자를 식별하기 위해 모든 클라이언트 요청과 함께 장치 ID 가 전송되는 무선 환경에 맞게 설계되었습니다. |

## 가상 호스트에 대한 여러 세션 체계

CA SiteMinder for Secure Proxy Server 는 엔터프라이즈의 각 가상 호스트에 대해 여러 세션 체계를 지원합니다. 가상 호스트에 액세스할 수 있는 각 사용자 에이전트 유형에 세션 체계를 할당할 수 있습니다. 다음 그림에서는 네 개의 가상 호스트에 대해 구성된 CA SiteMinder for Secure Proxy Server 를 보여 줍니다.




위 그림에서는 사용자가 액세스한 가상 호스트에 따라 사용자 에이전트에 대한 세션 체계가 달라짐을 보여 줍니다. 예를 들어 브라우저에서 www.company.com 에 액세스하는 경우 CA SiteMinder for Secure Proxy Server 는 SiteMinder 쿠키를 사용하여 사용자 세션을 유지 관리합니다. 그러나 BondTrading.company.com 에 액세스할 경우에는 브라우저 세션이 사용자 HTTPS 연결의 SSL ID 를 사용하여 유지 관리됩니다. PDA 및 무선 사용자의 세션은 쿠키를 사용하지 않는 세션 체계를 통해 유지 관리되는 반면에 브라우저 사용자 세션은 쿠키 또는 미니 쿠키를 통해 유지 관리됩니다.

## 쿠키를 사용하지 않는 페더레이션을 위해 특성 쿠키 삭제

쿠키를 교환하지 않으려는 환경을 지원하기 위해 CA SiteMinder for Secure Proxy Server 는 쿠키를 사용하지 않는 세션 체계에 SiteMinder 세션 쿠키를 매핑하고 응답에서 쿠키를 삭제합니다. 하지만 특성 쿠키는 매핑된 상태로 응답에 유지됩니다.

FWS 응용 프로그램은 페더레이션 요청을 처리하고 특성 쿠키와 SiteMinder 가 생성한 쿠키를 응답에 삽입합니다. 이 응답은 SPS 로 전달됩니다.

FWS 응용 프로그램에 의해 삽입된 특성 쿠키를 삭제하도록 CA SiteMinder for Secure Proxy Server 를 구성할 수 있습니다.

**세션 및 특성 쿠키를 삭제하도록 CA SiteMinder for Secure Proxy Server 를 구성하려면**

1. 텍스트 편집기에서 server.conf 파일을 엽니다.

   이 파일은 *sps_home*/secure-proxy/proxy-engine/conf 디렉터리에 있습니다.

2. FWS 서비스를 제공하는 가상 호스트에 대한 가상 호스트 섹션에 다음 코드를 추가합니다.

   deleteallcookiesforfed="yes"

3. 파일을 저장합니다.
4. SPS 를 다시 시작합니다.

# Server.conf 의 사용자 에이전트 설정

사용자 에이전트는 사용자가 네트워크 리소스에 액세스하는 데 사용할 수 있는 웹 브라우저, 무선 전화, PDA 등의 장치 유형을 정의합니다. 모든 사용자 에이전트는 <UserAgent> 요소에서 정의해야 합니다. 각 <UserAgent> 요소에는 사용자 에이전트 유형을 식별하는 이름 특성이 포함됩니다. 기본적으로 server.conf 파일에 미리 정의된 사용자 에이전트 유형은 없습니다.

사용자 에이전트 구성 섹션의 형식은 다음과 같습니다.

```
#<UserAgent name="user_agent_name_1">
#  header_name_1=some regular expression
# </UserAgent>
```

UserAgent 섹션의 지시문은 다음과 같습니다.

**header_name**

이 지시문에는 HTTP 요청의 사용자-에이전트 헤더가 포함됩니다. 이 헤더는 요청을 하는 장치의 유형을 나타냅니다. 정규식을 사용할 수 있으며 식의 일부로 이름의 일부를 제공할 수 있습니다. 이를 통해 사용자-에이전트 헤더에 포함된 버전 번호 등이 약간 다른 사용자 에이전트 유형도 지정할 수 있습니다.

<SessionSchemeMapping> 요소에서 장치 유형을 세션 체계와 연결하려면 먼저 <UserAgent> 요소에서 장치 유형을 정의해야 합니다.

## Nokia 사용자 에이전트 설정

Nokia 무선 전화를 위한 Nokia 사용자 에이전트 유형을 지정할 수 있습니다. <UserAgent> 섹션의 name 특성은 세션 체계 매핑을 지정할 때 사용자 에이전트 유형을 식별하는 데 사용되는 이름입니다.

Nokia 사용자 에이전트 항목의 형식은 다음과 같습니다.

```
# Nokia
<UserAgent name="Nokia">
    User-Agent="Nokia."
        attribute_name="value"
</UserAgent>
```

Nokia 사용자 에이전트에 대한 지시문은 다음과 같습니다.

**User-Agent**

이 지시문에는 HTTP 요청의 사용자 에이전트 헤더 내용이 포함됩니다. 이 헤더는 요청을 하는 장치의 유형을 나타냅니다. 정규식을 사용할 수 있으며 식의 일부로 이름의 일부를 제공할 수 있습니다. 이 지시문을 통해 사용자-에이전트 헤더에 포함된 버전 번호 등이 약간 다른 사용자 에이전트 유형도 지정할 수 있습니다.

**기본값:** Nokia

**attribute_name**

server.conf 에서 무선 장치 및 다른 사용자 에이전트 유형에 대한 섹션은 추가 특성과 이러한 특성의 값을 포함할 수 있습니다. 특성은 반드시 필요한 것은 아니지만 일부 응용 프로그램에는 사용하는 것이 좋을 수 있습니다.

# server.conf 파일의 가상 호스트 설정

server.conf 파일의 <VirtualHostDefaults> 요소는 가상 호스트에 대한 기본 설정을 지정합니다. 이러한 설정은 SPS 에 추가하는 각 가상 호스트에 사용됩니다.

가상 호스트에 대해 기본값이 아닌 값을 지정하려면 <VirtualHostDefaults> 요소 뒤에 <VirtualHost> 요소를 추가하십시오. <VirtualHost> 요소에는 기본 가상 호스트와는 다른 지시문 및 값을 포함해야 합니다.

기본 가상 호스트 설정은 다음과 같은 섹션으로 나뉩니다.

- 기본 세션 체계
- 세션 체계 매핑
- WebAgent.conf 설정
- 기본 가상 호스트 이름

<VirtualHostDefaults> 섹션의 형식은 다음과 같습니다.

```
<VirtualHostDefaults>
    # default session scheme
    defaultsessionscheme="default"
    enablerewritecookiepath="no"
    enablerewritecookiedomain="no"
    enableproxypreservehost="no"
    filteroverridepreservehost="no"

    # specify the block size for request and response in KBs
    requestblocksize="4"
    responseblocksize="4"

    #TO-DO: Define any session scheme mappings
    #<SessionSchemeMappings>
    #    user_agent_name=session_scheme_name
    #</SessionSchemeMappings>



    # Web Agent.conf
    <WebAgent>
    sminitfile="C:\Program Files\netegrity\secure-proxy\proxy-engine\
    conf\defaultagent\WebAgent.conf"
    </WebAgent>
</VirtualHostDefaults>
```

## 가상 호스트 쿠키 경로 및 도메인을 올바른 URI 로 설정

server.conf 파일의 가상 호스트 구성에는 enablerewritecookiepath 및 enablerewritecookiedomain 매개 변수가 포함되어 있으며 이 매개 변수를 사용하여 SPS 뒤에 있는 대상 서버가 생성한 쿠키를 관리할 수 있습니다. CA SiteMinder for Secure Proxy Server 는 클라이언트로부터 요청을 수신하면 사용자를 인증하고 클라이언트를 요청된 대상 서버에 연결합니다. 대상 서버가 쿠키를 생성하여 브라우저에 배치하면 CA SiteMinder for Secure Proxy Server 는 이 쿠키를 사용하여 사용자에게 클라이언트 응답을 다시 전송합니다. 클라이언트는 SPS 로부터 응답을 받은 후 쿠키를 저장합니다.

클라이언트가 후속 요청을 전송하면 브라우저는 저장된 쿠키에서 해당 URL 과 연결된 쿠키를 검색합니다. 일부 경우에는 대상 서버가 쿠키 경로를 초기 요청의 URI 가 아니라 자체 리소스 URI 로 설정했을 수 있습니다. 따라서 클라이언트가 후속 요청을 전송할 때 브라우저에는 잘못된 쿠키가 포함되어 있거나 쿠키가 아예 없습니다. 요청은 대상 서버에서 잘못된 쿠키를 포함하거나 쿠키를 포함하지 않은 상태로 수신됩니다.

브라우저에서 올바른 쿠키가 설정되도록 하려면 쿠키 경로와 쿠키 도메인을 다시 쓰도록 CA SiteMinder for Secure Proxy Server 를 구성하면 됩니다. 대상 서버는 쿠키 경로와 쿠키 도메인을 CA SiteMinder for Secure Proxy Server 서버에 있는 리소스의 URI 로 설정합니다. 클라이언트는 후속 요청에서 올바른 쿠키를 SPS 로 다시 전송할 수 있습니다.

다음 두 매개 변수가 사용됩니다.

**enablerewritecookiepath**

CA SiteMinder for Secure Proxy Server 가 프록시 뒤에 있는 서버가 설정한 URI 에서 초기 요청의 URI 로 쿠키 경로를 다시 쓰도록 합니다.

**기본값:** no

**enablerewritecookiedomain**

CA SiteMinder for Secure Proxy Server 가 프록시 뒤에 있는 서버가 설정한 도메인에서 초기 요청의 도메인으로 쿠키 도메인을 다시 쓰도록 합니다.

**기본값:** no

### 예

클라이언트가 CA SiteMinder for Secure Proxy Server 리소스 http://mysps.ca.com/basic/test/page0.html 을 요청합니다. enablerewritecookiepath 가 yes 로 설정되어 있으면 브라우저가 클라이언트로 다시 보내지기 전에 쿠키 경로가 /basic/test 로 다시 써집니다. 이 쿠키는 원래 CA SiteMinder for Secure Proxy Server 가 대상 서버에서 수신한 쿠키에 있던 쿠키 경로와 상관없이 다시 써집니다.

### 백엔드 쿠키 경로 및 도메인을 다시 쓰려면

1. 텍스트 편집기에서 server.conf 파일을 엽니다.
2. 다음 매개 변수 중 하나 또는 둘 모두를 yes 로 설정합니다.
   - enablerewritecookiepath
   - enablerewritecookiedomain
3. 파일을 저장합니다.
4. SPS 를 다시 시작합니다.

## HOST 헤더 파일 유지

HTTP HOST 헤더 파일을 유지하고 enableproxypreservehost 매개 변수를 사용하여 이 파일을 백엔드 서버로 보낼 수 있습니다. enableproxypreservehost 매개 변수를 사용하려면 다음 단계를 수행하십시오.

1. server.conf 파일을 엽니다.
2. 구성할 가상 호스트의 Virtual Host 섹션에 다음 매개 변수를 추가합니다.

   `enableproxypreservehost`
3. enableproxypreservehost 의 값을 yes 로 설정합니다.

enableproxypreservehost 를 활성화하면 이 매개 변수는 HTTP HOST 헤더를 제어하도록 구성된 필터보다 우선적으로 적용됩니다. enableproxypreservehost 를 비활성화하고 필터가 이 매개 변수보다 우선적으로 적용되도록 하려면 다음 단계를 수행하십시오.

1. server.conf 파일을 엽니다.
2. 구성할 가상 호스트의 Virtual Host 섹션에 다음 매개 변수를 추가합니다.

   `filteroverridepreservehost`
3. filteroverridepreservehost 의 값을 yes 로 설정합니다.

HTTP HOST 헤더를 제어하는 필터가 있는 경우에만 filteroverridepreservehost 를 활성화할 수 있습니다.

## 데이터 블록을 사용하여 대용량 파일 처리

CA SiteMinder for Secure Proxy Server 는 CA SiteMinder for Secure Proxy Server 와 백엔드 서버 사이에 전송되는 데이터를 블록으로 나누는 방법으로 사용자에게 보내는 대용량 파일의 전송을 처리합니다. CA SiteMinder for Secure Proxy Server 가 읽는 블록 크기는 server.conf 파일의 가상 호스트 섹션에 있는 다음 두 매개 변수를 사용하여 제어합니다.

- requestblocksize
- responseblocksize

사용자가 파일을 백엔드 서버로 전송하면 CA SiteMinder for Secure Proxy Server 는 해당 가상 호스트에 대해 지정된 requestblocksize 를 확인합니다. 이 requestblocksize 의 값에 따라 CA SiteMinder for Secure Proxy Server 는 데이터를 여러 블록으로 나눈 다음 블록을 백엔드 서버에 전달합니다.

마찬가지로 백엔드 서버가 데이터를 사용자에게 전송하면 CA SiteMinder for Secure Proxy Server 는 해당 가상 호스트에 대해 지정된 responseblocksize 를 확인합니다. 이 responseblocksize 의 값에 따라 CA SiteMinder for Secure Proxy Server 는 블록 처리를 계속하기 전에 백엔드 서버에서 블록 내의 데이터를 읽습니다. 이 경우 사용자는 이러한 파일 전송을 위한 읽기-쓰기 작업의 수를 제어할 수 있습니다. 대용량 파일 전송을 처리하려면 큰 블록 크기를 사용하십시오.

**참고:** requestblocksize 및 responseblocksize 매개 변수는 CA SiteMinder for Secure Proxy Server Java 프로세스의 사용 가능한 JVM 힙 크기 및 할당된 JVM 힙 크기에 비례하여 정의되어야 합니다.

## 대용량 파일 처리를 위한 파일 데이터 블록 크기 정의

가상 호스트에 대한 블록 크기를 구성할 때 대용량 파일을 처리하기 위한 블록 크기를 정의하려면 각 가상 호스트의 요청 및 응답 블록 크기를 수정하십시오. 이러한 매개 변수는 해당 가상 호스트에만 유효합니다. 데이터 블록 크기는 가상 호스트마다 다를 수 있으며, 해당 설정은 구성하는 관련 가상 호스트에만 적용됩니다.

**다음 단계를 수행하십시오.**

1. 텍스트 편집기에서 server.conf 파일을 엽니다.
2. 가상 호스트 구성에서 다음 매개 변수를 편집합니다.

   **requestblocksize**

   데이터 블록을 백엔드 서버로 전송하기 전에 한 번에 읽을 요청 데이터의 블록 크기를 정의합니다. 블록 크기의 단위는 KB 입니다.

   **제한:** 1 KB ~ 약 352,000 KB. 값이 8 KB 보다 크거나 같은 경우 8 KB 의 청크가 생성됩니다. 값이 1 KB 에서 8 KB 사이인 경우에는 해당하는 크기의 청크가 생성됩니다.

   **기본값:** 4

   **responseblocksize**

   백엔드 서버에서 사용자에게 데이터 블록을 전달하기 전에 한 번에 읽을 응답 데이터의 블록 크기를 정의합니다. 블록 크기의 단위는 KB 입니다.

   **제한:** 1 KB ~ 약 352,000 KB.

   **기본값:** 8
3. server.conf 파일을 저장합니다.
4. SPS 를 다시 시작합니다.

## 데이터 블록의 JVM 힙 크기 조정

requestblocksize 및 responseblocksize 매개 변수는 CA SiteMinder for Secure Proxy Server Java 프로세스의 사용 가능한 JVM 힙 크기 및 할당된 JVM 힙 크기에 비례하여 정의됩니다.

**CA SiteMinder for Secure Proxy Server JVM 힙 크기를 정의하려면**

1. 적절한 디렉터리로 이동합니다.
   - *Windows: sps_home*/secure-proxy/proxy-engine/conf
   - UNIX: sps_home/secure-proxy/proxy-engine
2. 다음 파일 중 하나를 엽니다.
   - Windows 시스템: SmSpsProxyEngine.properties 파일
   - UNIX 시스템: proxyserver.sh 파일

3. 파일의 Java 섹션에 다음 매개 변수를 추가합니다.
   - -Xms256m
   - -Xmx512m
4. 파일을 저장합니다.

## 기본 가상 호스트에 대한 세션 체계 매핑

세션 체계 매핑은 세션 체계를 사용자 에이전트 유형과 연결합니다. server.conf 파일의 <UserAgent> 요소에 정의된 사용자 에이전트 유형은 <SessionScheme> 요소에 정의된 세션 체계에 매핑되어야 합니다.

사용자 에이전트와 연결된 세션 체계 매핑의 형식은 다음과 같습니다.

```
#<SessionSchemeMappings>
    #     user_agent_name=session_scheme_name
#</SessionSchemeMappings>
```

이 섹션의 지시문은 다음과 같습니다.

**user_agent_name**

사용자 에이전트를 세션 체계와 연결합니다. 다음과 같이 값을 설정합니다.

**user_agent_name**

파일의 <UserAgent> 섹션에 정의된 이름을 지정합니다.

**session_scheme_name**

SessionScheme 요소에 정의된 체계를 지정합니다.

**예:**

browser, phone1 및 phone2 라는 사용자 에이전트가 정의되어 파일에 정의된 세션 체계에 매핑되었습니다. 이 예의 경우 browser 는 default 세션 체계에 매핑되고, phone1 은 simple_url 체계에 매핑되고, phone2 는 minicookie 세션 체계에 매핑되었습니다.

결과 <SessionSchemeMappings> 요소는 다음과 같이 나타납니다.

```
# Session Scheme Maps
<SessionSchemeMappings>
  browser="default"
  phone1="simple_url"
  phone2="minicookie"
</SessionSchemeMappings>
```

## 기본 가상 호스트에 대한 웹 에이전트 설정

server.conf 파일에는 <VirtuallHostDefaults>에 대한 <WebAgent> 섹션이 포함되어 있습니다. sminitfile 지시문은 기본 웹 에이전트에 대한 구성 파일인 WebAgent.conf 를 지정합니다. 로컬 구성이 허용되는 경우 WebAgent.conf 파일은 로컬 구성 파일 LocalConfig.config 를 가리킵니다.

가상 호스트를 여러 개 생성하는 경우 웹 에이전트 구성 파일의 대체 설정을 사용하지 않으려면 기본 웹 에이전트를 사용합니다. 예를 들어 다른 로그 수준을 지정하기 위해 지시문을 다르게 설정하려면 새 가상 호스트에 대해 다른 웹 에이전트를 사용하십시오.

**새 가상 호스트에 대한 새 웹 에이전트를 구성하려면**

1. serverb 와 같은 새 가상 호스트 이름으로 디렉터리를 생성합니다.
2. 기본 가상 호스트의 디렉터리 내용을 새 디렉터리에 복사합니다.

   새 웹 에이전트가 설치된 다른 SiteMinder 를 가리키는 경우 smreghost 를 실행합니다.
   **참고:** 두 가상 호스트의 웹 에이전트 구성 개체가 모두 동일한 SiteMinder 설치를 가리키는 경우에는 smreghost 를 실행할 필요가 없습니다. 두 웹 에이전트 모두에 동일한 smhost 파일을 사용하면 됩니다.
3. 텍스트 편집기를 사용하여 새 에이전트 구성 개체를 반영하도록 WebAgent.conf 를 수정합니다. 웹 에이전트에 다른 로그 파일이 있는지 확인합니다.

4. WebAgent.conf 파일을 열고 다음의 필수 지시문을 고유 값과 함께 추가합니다.

   ServerPath="*path*"

   **path**

   편집 중인 WebAgent.conf 파일의 정규화된 경로를 지정합니다.

   - Windows 의 경우 이 값은 고유한 영숫자 문자열이어야 합니다. 이 문자열에는 백슬래시('\') 문자가 허용되지 않습니다.
   - UNIX 의 경우 이 값은 편집 중인 WebAgent.conf 파일의 정규화된 경로여야 합니다.

5. 정책 서버에서 server.conf 파일의 첫 번째 호스트 구성 개체에 해당하는 에이전트 구성 개체에 액세스합니다. MaxResoureceCacheSize 및 MaxSessionCacheSize 의 에이전트 캐시 설정을 확인하고 캐시 제한이 모든 에이전트 구성 개체를 고려한 것인지 확인합니다.

**참고:** 웹 에이전트 설정에 대한 자세한 내용은 *CA SiteMinder Web Agent Guide*(CA SiteMinder 웹 에이전트 안내서)를 참조하십시오.

## requirecookies 매개 변수 설정

server.conf 파일의 requirecookies 설정은 정책 서버 구성 중에 기본 인증이 설정된 경우에만 유용한 특수 웹 에이전트 설정입니다. 이 설정을 사용할 경우 에이전트는 SMSESSION 또는 SMCHALLENGE 쿠키가 있어야 기본 인증 헤더를 포함하여 HTTP 요청을 성공적으로 처리할 수 있습니다.

포함된 웹 에이전트를 쿠키가 필요하도록 구성할 경우 브라우저가 HTTP 쿠키를 허용해야 합니다. 브라우저가 HTTP 쿠키를 허용하지 않으면 에이전트에서 보호된 모든 리소스에 대한 액세스를 거부하는 오류 메시지가 표시됩니다.

연결된 가상 서버의 모든 사용자 에이전트 유형이 기본 세션 체계를 사용하는 경우 requirecookies 설정을 yes 로 설정하십시오. 에이전트 유형이 쿠키를 사용하지 않는 세션 체계를 사용하는 경우에는 requirecookies 매개 변수를 no 로 설정하십시오.

## 대상 서버에 의한 리디렉션 처리

일부 대상 서버는 CA SiteMinder for Secure Proxy Server 의 요청에 대해 리디렉션으로 응답할 수 있습니다.

**참고:** CA SiteMinder for Secure Proxy Server 에 대한 요청의 결과로 발생하는 리디렉션은 프록시 규칙에서 발생하는 리디렉션과 같지 않습니다. 프록시 규칙의 리디렉션에 대한 자세한 내용은 nete:redirect 를 참조하십시오.

대상 서버가 시작하는 리디렉션의 대상은 DMZ 뒤의 서버일 수 있으며 이 경우 리디렉션에 지정된 URL 로 인해 오류가 발생합니다. 그러나 대상 서버에서의 리디렉션 대신 가상 호스트 서버 이름 및 포트 번호로 대체하는 매개 변수를 가상 호스트 구성에 포함할 수 있습니다.

리디렉션 작성 대신 가상 호스트 서버 및 포트로 대체하려면 다음을 구성하십시오.

**enableredirectrewrite**

리디렉션 다시 쓰기를 사용하거나 사용하지 않도록 설정합니다. 이 지시문의 값이 yes 로 설정되어 있으면 대상 서버가 시작하는 리디렉션의 URL 이 SPS 에 의해 검사됩니다. 리디렉션 URL 에 관련 redirectrewritablehostnames 매개 변수에 지정된 문자열 목록의 문자열이 포함되어 있는 경우, 리디렉션의 서버 이름 및 포트 번호가 가상 호스트의 서버 이름 및 포트 번호로 대체됩니다.

이 매개 변수의 값이 no 로 설정되어 있으면 대상 서버가 시작하는 모든 리디렉션이 요청하는 사용자에게 다시 전달됩니다.

**redirectrewritablehostnames**

대상 서버에 의해 리디렉션이 시작될 때 CA SiteMinder for Secure Proxy Server 가 검색하는 문자열의 쉼표로 구분된 목록을 포함합니다. 지정된 문자열이 리디렉션 URL 의 서버 또는 포트 부분에 있는 경우 CA SiteMinder for Secure Proxy Server 는 리디렉션 URL 의 서버 이름 및 포트 부분을 가상 호스트의 이름 및 포트 번호로 대체합니다.

이 매개 변수의 값을 "ALL"로 지정하면 CA SiteMinder for Secure Proxy Server 는 대상 서버가 시작하는 모든 리디렉션을 가상 호스트의 서버 이름 및 포트 번호로 대체합니다.

예를 들어 server.conf 파일의 가상 호스트 구성에 다음 매개 변수가 포함되어 있다고 가정하십시오.

```
<VirtualHost name="sales">
    hostnames="sales, sales.company.com"
    enableredirectrewrite="yes"
    redirectrewritablehostnames="server1.company.com,domain1.com"
</VirtualHost>
```

사용자가 http://sales.company.com:80 에서 요청하면 CA SiteMinder for Secure Proxy Server 는 해당 요청을 프록시 규칙에 따라 대상 서버에 전달합니다. 대상 서버가 server1.internal.company.com 에 대한 리디렉션으로 응답하는 경우에는 리디렉션이 사용자에게 sales.company.com:80 으로 전달되기 전에 다시 써집니다.

**참고:** 리디렉션된 요청을 처리하도록 CA SiteMinder for Secure Proxy Server 에 대한 프록시 규칙을 구성해야 합니다.

## 가상 호스트 이름 구성

CA SiteMinder for Secure Proxy Server 가 하나 이상의 호스트 이름에 대해 가상 호스트 역할을 하도록 하려면 관련 호스트 이름 및 IP 주소에 대한 <VirtualHost> 요소를 구성하십시오. 각 server.conf 파일에는 기본 가상 호스트에 대한 <VirtualHost> 요소 하나와 다른 IP 주소에 있는 호스트 이름에 대한 추가 요소가 포함되어야 합니다.

다음은 server.conf 파일의 기본 가상 호스트에 대한 <VirtualHost> 요소의 예입니다.

```
# Default Virtual Host

<VirtualHost name="default">
hostnames="home, home.company.com"
addresses="123.123.12.12"
</VirtualHost>
```

위의 예에서 기본 가상 호스트는 IP 주소 123.123.12.12 에 있는 home.company.com 이라는 호스트입니다. 호스트 이름을 쉼표로 구분된 값 목록으로 추가하여 기본 가상 호스트로 확인되는 호스트 이름을 추가할 수 있습니다. 프록시 구성에 다른 가상 호스트를 추가하려면 추가 가상 호스트에 대한 호스트 이름 지시문을 포함하는 다른 <VirtualHost> 요소를 추가하십시오.

**예:**

서버에 대한 영업 가상 호스트(sales.company.com 가상 호스트)를 추가하려면 다음 요소를 추가하십시오.

```
<VirtualHost name="sales">
hostnames="sales, sales.company.com"
</VirtualHost>
```

## 가상 호스트의 기본값 재정의

특정 가상 호스트에 대해 명시적으로 설정을 입력하지 않은 한 server.conf 파일에 정의된 모든 가상 호스트에는 <VirtualHostDefaults> 설정이 사용됩니다.

따라서 단일 가상 호스트에 대해 모든 가상 설정을 다시 구성할 필요가 없습니다. <VirtualHost> 요소에서 다시 정의되지 않은 모든 설정은 <VirtualHostDefaults> 설정에서 적용됩니다.

### 가상 호스트 기본값을 재정의하려면

1. 수정하려는 <VirtualHost> 요소에 기본 가상 호스트 구성의 지시문을 추가합니다.
2. <VirtualHost> 요소에서 지시문 값을 새로 지정합니다.
3. 파일을 저장합니다.
4. SPS 를 다시 시작합니다.

### 예

"sales"라는 가상 호스트에 기본 가상 호스트에 대해 구성된 기본 세션 체계가 필요한 경우 <VirtualHost> 요소를 다음과 같이 수정할 수 있습니다.

```
<VirtualHost name="sales">
    hostnames="sales, sales.company.com"
    addresses="123.123.22.22"
    defaultsessionscheme="minicookie"
</VirtualHost>
```

# 제 8 장: SPS 로그 설정 구성

## SPS logger.properties 파일 개요

CA SiteMinder for Secure Proxy Server 로그 설정은 logger.properties 파일을 통해 구성됩니다. 이 파일의 설정은 CA SiteMinder for Secure Proxy Server 가 런타임에 읽는 이름/값 쌍이나 지시문의 그룹입니다. SPS 를 다시 시작하지 않고도 logger.properties 파일을 업데이트할 수 있습니다.

logger.properties 파일의 기본 위치는 다음과 같습니다.

*sps_home*/Tomcat/properties

## logger.properties 파일 수정

CA SiteMinder for Secure Proxy Server 에 대한 로그 설정은 다음 디렉터리에 있는 logger.properties 파일에서 유지 관리됩니다.

*sps_home*/Tomcat/properties

**다음 단계를 수행하십시오.**

1. 텍스트 편집기에서 파일을 엽니다.
2. 지시문을 필요한 대로 편집합니다.
3. 파일을 저장합니다.

로그 설정이 변경되었습니다.

# 로깅 설정

logger.properties 파일의 내용은 다음과 같은 섹션으로 나뉘어 있습니다.

- SvrConsoleAppender 설정
- SvrFileAppender 설정
- 서버 로그 설정
- 서버 로그 롤링 설정

이 파일에 포함된 지시문은 name=value 형식을 따릅니다. # 기호로 시작하는 행은 주석이므로 CA SiteMinder for Secure Proxy Server 가 구성 설정을 로드할 때 이 행은 읽지 않습니다.

**참고**: Windows 시스템의 경로 이름에는 이중 백슬래시(\\)가 사용됩니다.

## SvrConsoleAppender 설정

SvrConsoleAppender 설정 섹션에는 콘솔에 이벤트를 로깅하기 위한 설정이 포함되어 있습니다. 이 섹션의 형식은 다음과 같습니다.

```
# SvrConsoleAppender is set to be a ConsoleAppender.
log4j.appender.SvrConsoleAppender=org.apache.log4j.ConsoleAppender
log4j.appender.SvrConsoleAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.SvrConsoleAppender.layout.ConversionPattern=<log_message_display_format_on_console>
```

***log_message_display_format_on_console***

콘솔에 표시할 로그 메시지의 형식을 지정합니다. 이 제품은 모든 log4j 날짜 패턴 문자열을 지원합니다.

**기본값**: [%d{dd/MMM/yyyy:HH:mm:ss-SSS}] [%p] - %m%n

## SvrFileAppender 설정

SvrFileAppender 설정 섹션에는 파일에 이벤트를 로깅하기 위한 설정이 포함되어 있습니다. 이 섹션의 형식은 다음과 같습니다.

```
# SvrFileAppender is set to be a FileAppender.
log4j.appender.SvrFileAppender=org.apache.log4j.FileAppender
log4j.appender.SvrFileAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.SvrFileAppender.layout.ConversionPattern=<log_message_display_format_in_file>
```

***log_message_display_format_in_file***

파일에 표시할 로그 메시지의 형식을 지정합니다. 이 제품은 모든 log4j 날짜 패턴 문자열을 지원합니다.

**기본값**: [%d{dd/MMM/yyyy:HH:mm:ss-SSS}] [%p] - %m%n

# 로그 설정

서버 로그 설정 섹션에서는 로깅을 사용하거나 사용하지 않도록 설정하고, 로깅 수준을 설정하고, 로그 메시지의 출력 형식을 설정할 수 있습니다. 이 섹션의 형식은 다음과 같습니다.

```
# Server.conf settings:
# details of setting "log4j.rootCategory":
# For First attribute:
# Depending on the logging level needed, set the appropriate level
# Possible values : OFF, FATAL, ERROR, WARN, INFO, DEBUG, ALL
# For Second attribute:
# if you want to enable log console, then add SvrConsoleAppender, else don't  add this.
# For Third attribute:
# if you want to enable logging into file, theb add SvrFileAppender, else don't add
this.
log4j.rootCategory=<log_level>,<output_format>
```

***log_level***

메시지의 로그 수준을 지정합니다. 다음 목록에는 가능한 값이 우선 순위 기준 오름차순으로 나와 있습니다.

- OFF
- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- ALL

값이 OFF 로 설정되어 있으면 로깅이 사용되지 않습니다. 값이 OFF 가 아닌 다른 값으로 설정되어 있으면 로깅이 사용됩니다.

**기본값**: INFO

***output_format***

로그 메시지가 표시되는 방식을 지정합니다. 로그 메시지를 콘솔에 표시하거나, 파일에 저장하거나, 둘 모두를 수행할 수 있습니다.

**기본값**: SvrFileAppender

예를 들어 로그 수준이 INFO 인 경우 로그 메시지를 콘솔에 표시하고 파일에 저장하려면 다음 명령을 사용하십시오.

```
log4j.rootCategory=INFO,SvrConsoleAppender,SvrFileAppender
```

## 로그 롤링 설정

서버 로그 롤링 설정 섹션에서는 로그 롤링을 사용하도록 설정할 수 있습니다. 다음 메커니즘 중 *하나*를 기반으로 로그 롤링을 사용하도록 설정할 수 있습니다.

- 파일 크기를 기반으로 한 로그 롤링
- 파일 사용 기간을 기반으로 한 로그 롤링

이 섹션의 형식은 다음과 같습니다.

```
# Enable the below setting only if file logging is enabled above. if not make it as
an comment by adding "#" at the begging of the line.
log4j.appender.SvrFileAppender.File=<logfile_path>
# Enable this only if file logging is enabled above.
# set vale to "true" if messages are to be appended to the existing file. else set
to "false"
log4j.appender.SvrFileAppender.Append=true|false
#Configurations to rollover server log file based on file size
log4j.appender.SvrFileAppender=org.apache.log4j.RollingFileAppender
log4j.appender.SvrFileAppender.MaxFileSize=<maximum_logfile_size>
log4j.appender.SvrFileAppender.MaxBackupIndex=<maximum_number_of_logfile>
```

파일 크기를 기반으로 한 로그 롤링 섹션에는 파일 크기를 기반으로 로그 롤링을 사용하도록 설정하기 위한 다음 설정이 포함되어 있습니다.

***logfile_path***

로그 파일의 이름과 경로를 지정합니다.

**기본 이름**: server.log

**기본 경로**: *install_dir_home*/secure-proxy/proxy-engine/logs/

**true|false**

시스템이 로그 파일을 관리하는 방식을 지정합니다. 이 값이 true 로 설정되어 있으면 시스템은 시작 시 새 로그 메시지를 기존 로그 파일에 추가합니다. 이 값이 false 로 설정되어 있으면 시스템은 시작 시 기존 로그 파일을 롤오버하고 새 로그 메시지를 위한 로그 파일을 생성합니다.

**기본값**: true

***MaxFileSize***

로그 파일의 최대 크기를 지정합니다. 이 크기를 넘으면 시스템이 새 로그 파일을 만들어야 합니다.

**기본값:** 1 MB

***MaxBackupIndex***

시스템이 생성하는 최대 로그 파일 수를 지정합니다. 로그 파일의 수가 지정된 최대 수를 초과하면 시스템은 가장 오래된 로그 파일을 삭제하고 새 로그 파일을 만듭니다.

**기본값:** 10

파일 사용 기간을 기반으로 한 로그 롤링 섹션에는 파일 사용 기간을 기반으로 로그 롤링을 사용하도록 설정하기 위한 다음 설정이 포함되어 있습니다.

***date_pattern***

시스템이 새 로그 파일을 생성해야 하는 날짜를 지정합니다.

**기본값**: yyyy-MM-dd

다음 형식으로 새 로그 파일이 생성되었습니다.

*<logfile_name>.<date_format>*

***logfile_name***

로그 파일의 이름을 지정합니다.

**기본값**: server.log

***date_format***

로그 파일이 생성된 날짜를 지정합니다. 이 파일은 모든 log4j 날짜 패턴 문자열을 지원합니다.

**기본값**: yyyy-MM-dd

# 로깅을 위해 WebAgent.conf 의 ServerPath 수정

가상 호스트에 대한 웹 에이전트를 구성하는 경우 각 호스트마다 고유한 웹 에이전트 캐시, 로그 파일 및 건전성 모니터링 리소스가 있어야 합니다. 리소스가 고유하도록 하려면 ServerPath 매개 변수를 구성하십시오.

ServerPath 매개 변수는 캐시, 로깅 및 건전성 모니터링의 웹 에이전트 리소스에 대한 고유 식별자를 지정합니다. 각 서버 인스턴스가 이러한 에이전트 리소스의 고유 집합을 갖도록 하려면 ServerPath 매개 변수의 값이 고유해야 합니다.

예를 들어 ServerPath 매개 변수를 웹 서버 로그 파일이 저장된 디렉터리(예: *server_instance_root*/logs)로 설정할 수 있습니다.

사용 환경에 가상 호스트가 있는 경우 각 WebAgent.conf 파일에 ServerPath 매개 변수가 있는지 확인하십시오.

**각 WebAgent.conf 파일에 ServerPath 매개 변수가 있는지 확인하려면**

1. *sps_home*\secure-proxy\proxy-engine\conf\defaultagent 디렉터리에 있는 WebAgent.conf 파일로 이동합니다.
2. 파일을 엽니다.
3. ServerPath 설정이 고유 문자열 또는 경로로 구성되어 있는지 확인합니다.

   Windows 의 경우 고유 문자열을 지정할 수 있습니다. UNIX 의 경우에는 고유 시스템 경로를 지정하십시오.
4. WebAgent.conf 파일을 저장합니다.

# 제 9 장: 프록시 규칙 구성

이 섹션은 다음 항목을 포함하고 있습니다.



## 프록시 규칙 개요

CA SiteMinder for Secure Proxy Server 의 기본 용도는 요청을 엔터프라이즈의 적절한 대상 서버로 라우팅하는 것입니다. CA SiteMinder for Secure Proxy Server 는 서버의 기본 제공 프록시 엔진을 사용하여 요청을 라우팅합니다. 프록시 엔진은 프록시 규칙을 해석하고, 백엔드 리소스에 대한 모든 사용자 요청의 배치를 처리하기 위한 전달 서비스와 리디렉션 서비스를 제공합니다.

프록시 규칙은 CA SiteMinder for Secure Proxy Server 가 엔터프라이즈 내 대상 서버에 있는 리소스에 대한 요청을 전달하거나 리디렉션하는 방식을 세부적으로 정의합니다. SPS 와 함께 설치된 프록시 규칙 DTD 에 따라 일련의 프록시 규칙이 XML 구성 파일에 정의되어 있습니다.

**참고:** proxyrules.xml 파일에는 요청을 http://www.ca.com$0 으로 전달하는 기본 규칙이 포함되어 있습니다. 여기서 $0 은 원래 요청의 전체 URI 가 대상(이 경우 www.ca.com)에 추가됨을 나타냅니다. 사용 환경에 맞게 이 규칙을 수정해야 합니다.

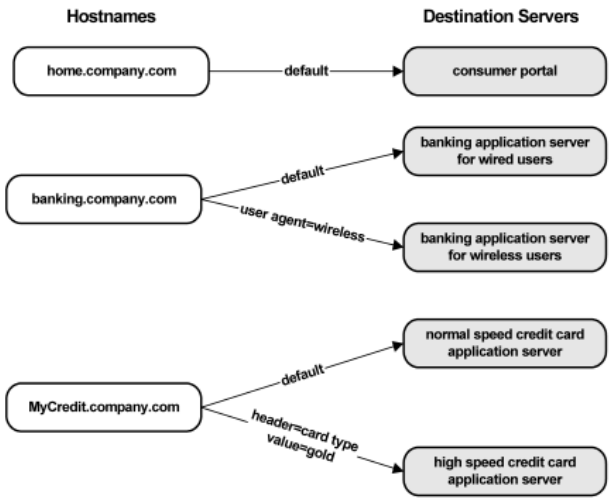
**추가 정보:**

프록시 규칙 DTD (페이지 170)

## 들어오는 요청에 대한 라우팅 계획

proxyrules.xml 파일을 설정하기 전에 들어오는 요청에 대한 라우팅을 자세히 계획해야 합니다. 요청된 리소스가 포함된 가상 호스트에 따라 프록시 규칙은 기본 대상을 사용하거나, 사용자 에이전트 유형을 기준으로 요청을 전달하거나, HTTP 헤더 값을 사용하여 요청을 이행할 대상 서버를 결정할 수 있습니다. CA SiteMinder for Secure Proxy Server 는 여러 가상 호스트에 대한 액세스를 제공할 수 있습니다.

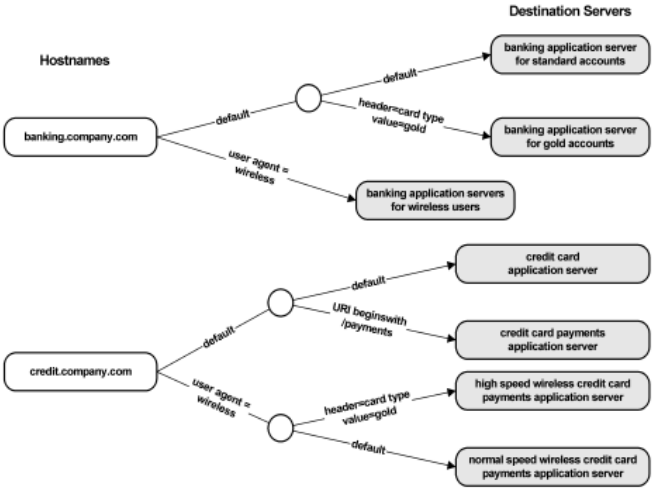
다음 그림에서는 프록시 규칙을 사용하여 요청을 적절한 대상 서버로 라우팅하는 방법을 보여 줍니다.




**참고:** 프록시 규칙 구성 파일은 CA SiteMinder for Secure Proxy Server 에 의해 하향식으로 처리됩니다. 즉, 들어오는 요청이 충족하는 첫 번째 조건에 따라 프록시 엔진의 동작이 결정됩니다. 예를 들어 프록시 규칙 집합에 요청의 URI 에 포함된 문자열을 기준으로 하는 조건이 두 개 있고 들어오는 요청의 URI 중 일부가 두 문자열 모두와 일치하는 경우 해당 요청을 라우팅하는 데는 프록시 규칙 파일에 나열된 첫 번째 조건이 사용됩니다.

프록시 규칙을 사용하여 요청을 대상 서버로 전달하기 위한 더 복잡한 조건을 제어할 수도 있습니다.

다음 그림에서는 부모 조건 내에 중첩된 두 번째 수준의 조건과 함께 프록시 규칙을 사용하는 방법을 보여 줍니다.



# 프록시 규칙 용어

프록시 규칙 구성 파일에서는 CA SiteMinder for Secure Proxy Server 가 사용자 요청을 라우팅할 때 사용하는 XML 요소를 설명합니다. 프록시 규칙을 설명하는 데 사용되는 용어는 다음과 같습니다.

**대상**

대상은 요청을 이행하는 URL 입니다. CA SiteMinder for Secure Proxy Server 는 요청을 대상에 전달하거나 대상을 지정하는 사용자에게 리디렉션 응답을 전송합니다. 프록시 규칙 집합에는 프록시 규칙에 정의된 조건 및 사례에 따라 연결할 수 있는 대상이 포함되어야 합니다.

**조건**

조건은 CA SiteMinder for Secure Proxy Server 가 요청의 대상을 확인할 수 있게 해 주는 요청 특성입니다. 각 조건에는 CA SiteMinder for Secure Proxy Server 가 요청을 적절한 대상으로 리디렉션하기 위해 평가하는 여러 사례가 있을 수 있습니다. 각 조건에는 요청이 조건에 정의된 사례와 일치하지 않을 경우의 동작을 정의하는 기본 요소가 포함되어야 합니다.

조건에는 다음 중 *하나* 또는 그 *이상*이 포함될 수 있습니다.

**URI**

CA SiteMinder for Secure Proxy Server 는 요청된 URL 중 호스트 이름 다음에 나오는 부분을 사용하여 요청을 라우팅할 방법을 결정합니다. DTD 에 설명된 조건을 사용하면 요청된 리소스의 파일 확장명과 같은 URI 의 일부분을 사용하여 요청을 라우팅할 수 있습니다.

**쿼리 문자열**

CA SiteMinder for Secure Proxy Server 는 URI 의 쿼리 문자열 부분을 사용하여 요청을 라우팅할 방법을 결정합니다. 쿼리 문자열에는 요청에서 '?' 뒤에 나오는 모든 문자가 포함됩니다.

**호스트**

CA SiteMinder for Secure Proxy Server 는 요청된 서버 호스트 이름을 사용하여 요청을 라우팅할 방법을 결정합니다. 호스트 이름의 포트 번호를 요청을 라우팅하기 위한 조건으로 사용할 수도 있습니다. 이 조건은 프록시에 가상 서버가 여러 개 있는 경우에 사용됩니다.

**헤더**

CA SiteMinder for Secure Proxy Server 는 HTTP 헤더의 값을 사용하여 요청을 라우팅할 방법을 결정합니다. 리소스에 액세스하는 데 사용되는 장치 유형을 기준으로 요청을 라우팅하기 위해 USER_AGENT HTTP 헤더에 따라 요청을 라우팅할 수 있습니다.

**참고:** SiteMinder 응답에서 파생된 HTTP 헤더를 사용하여 요청을 라우팅할 방법을 결정할 수도 있습니다.

**쿠키**

CA SiteMinder for Secure Proxy Server 는 쿠키가 있는지 여부와 쿠키 값을 사용하여 요청을 라우팅할 방법을 결정합니다. 쿠키 값이 인코딩된 경우 인코딩 매개 변수에서 인코딩 체계를 지정하십시오. CA SiteMinder for Secure Proxy Server 는 base64 인코딩 체계만 지원합니다.

**사례**

사례는 CA SiteMinder for Secure Proxy Server 가 요청의 최종 대상을 결정하는 데 필요한 정보를 제공하는 조건에 대한 일련의 특정 값입니다. 예를 들어 일련의 프록시 규칙이 호스트 조건을 사용하는 경우 사례에는 home.company.com 및 banking.company.com 과 같이 시스템에 대해 구성된 가상 호스트가 포함됩니다.

# 프록시 규칙 구성 파일 설정

프록시 규칙 구성 파일은 server.conf 파일에서 <ServiceDispatcher> 요소의 rules_file 지시문으로 식별되는 XML 구성입니다. rules_file 지시문은 설치 디렉터리에서 프록시 규칙 구성 파일까지의 상대 경로를 나타냅니다. 설치 시 기본 프록시 규칙 구성 파일의 상대 경로가 자동으로 생성되어 기본 가상 호스트에 대한 rules_file 지시문에 삽입됩니다.

생성되는 경로와 프록시 규칙 파일 이름은 다음과 같습니다.

*sps_home*/secure-proxy/proxy-engine/conf/proxyrules.xml

proxyrules.xml 파일의 모든 항목은 올바르게 구성되고 XML 사양에 따른 구문 규칙을 충족해야 합니다. 프록시 규칙 구성 파일을 변경할 경우 서버를 다시 시작하지 않아도 변경 내용이 적용됩니다. 파일이 변경되면 CA SiteMinder for Secure Proxy Server 는 이를 감지하고 새 프록시 규칙 파일을 로드합니다.

**참고**: SPS 는 요청을 충족시키는 백엔드 서버의 MBCS(멀티바이트 문자 집합) URL 로 전달된 요청을 수신할 수 있습니다.

프록시 규칙을 구문 분석하는 동안 규칙에서 오류를 발견하면 CA SiteMinder for Secure Proxy Server 는 서버 로그에 오류를 기록하며 변경 내용을 무시하고 기존 프록시 규칙을 사용합니다. 서버 로그 파일 위치는 logger.properties 파일에 지정되어 있습니다.

**추가 정보:**

server.conf 파일의 서비스 디스패처 설정 (페이지 120)

# 프록시 규칙 DTD

proxyrules.xml 파일은 프록시 규칙 DTD 를 사용하여 생성해야 합니다. 프록시 규칙 DTD 를 보려면 다음 디렉터리로 이동하십시오.

*sps_home*\secure-proxy\proxy-engine\conf\dtd

DTD 에서 구성할 수 있는 요소는 다음과 같습니다.

- nete:proxyrules
- nete:description
- nete:case
- nete:cond
- nete:default
- nete:forward
- nete:redirect
- nete:local
- nete:xprcond

## nete:proxyrules

nete:proxyrules 요소에 대한 전체 정의는 다음과 같습니다.

```
<!ELEMENT nete:proxyrules (nete:description?,(nete:cond | nete:forward |
nete:redirect | nete:local)) >
```

이 요소는 프록시 규칙 XML 구성 파일의 루트 요소입니다. 이 요소에는 선택적 nete:description 요소와 다음 요소 중 하나가 포함되어 있습니다.

- nete:cond
- nete:xprcond
- nete:forward
- nete:redirect
- nete:local

nete:proxyrules 요소는 프록시 규칙 구성 파일에 반드시 있어야 합니다.

### debug 특성

nete:proxyrules 요소에는 다음 특성이 있습니다.

```
<ATTLIST nete:proxyrules
debug (yes|no) "no"
```

이 특성은 프록시 규칙을 디버깅하는 데 도움이 되는 로깅을 사용하거나 사용하지 않도록 설정합니다. 이 특성의 기본값은 no 입니다. 로깅을 사용하도록 설정하려면 이 특성을 yes 로 설정하십시오.

예를 들면 다음과 같습니다.

```
<nete:proxyrules xmlns:nete="http://www.ca.com/" debug="yes">
```

로깅이 사용되도록 설정되면 CA SiteMinder for Secure Proxy Server 에 대한 추적 로깅 정보가 추적 로그에 포함됩니다. 로그 파일의 위치는 CA SiteMinder for Secure Proxy Server 설치 프로세스 중에 지정한 에이전트 구성 개체의 TraceFileName 매개 변수에 따라 결정됩니다. 동일한 에이전트 구성 개체의 TraceConfigFile 매개 변수는 보안 프록시 관련 추적 로깅 구성 파일을 가리켜야 합니다.

이 파일은 기본적으로 다음 위치에 있습니다.
<*install-dir*>\proxy-engine\conf\defaultagent\SecureProxyTrace.conf

**참고:** 이 변경 내용은 CA SiteMinder for Secure Proxy Server 를 다시 시작할 필요 없이 바로 적용됩니다.

## nete:description

nete:description 요소에 대한 전체 정의는 다음과 같습니다.

```
<!ELEMENT nete:description (#PCDATA)>
```

이 요소는 다른 요소에 대한 PCDATA(구문 분석된 문자 데이터) 설명을 포함하는 선택적 요소입니다.

**참고:** PCDATA 는 태그 텍스트가 아닌 텍스트입니다.

nete:description 요소는 nete:proxyrules 요소의 자식 요소로, 선택 항목에 대한 설명을 포함할 수 있습니다.

## nete:case

nete:case 요소는 nete:cond 부모 요소에 정의된 특정 조건 값과 연결되는 대상을 제공합니다. SPS 는 nete:case 요소의 값을 사용하여 사례를 평가합니다.

nete:case 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:case (nete:cond | nete:xprcond | nete:forward | nete:redirect |
nete:local)>
```

모든 nete:case 요소에 다음 자식 요소 중 하나가 포함되어야 합니다.

**nete:cond**

nete:case 요소는 추가 nete:cond 요소를 포함할 수 있습니다. 그러면 프록시 규칙 집합에 여러 조건을 중첩시킬 수 있습니다.

**nete:xprcond**

nete:case 요소는 URI 의 정규식 일치를 위한 추가 nete:xprcond 요소를 포함할 수 있습니다. 그러면 다른 조건 집합에 정규식 조건을 중첩시킬 수 있습니다.

**nete:forward**

nete:case 비교를 수행하는 요청이 전달될 대상을 제공합니다.

**nete:redirect**

nete:case 비교를 수행하는 요청이 리디렉션될 대상을 제공합니다. 리디렉션된 요청은 SPS 를 통하지 않고 대상 서버에 의해 직접 이행됩니다.

다음 예에서 nete:cond 요소는 URI 일치를 지정하며, nete:case 요소는 비교 유형 특성의 가능한 사용 사례를 보여 줍니다.

```
<nete:cond type="uri" criteria="beginswith">
    <nete:case value="/hr">
      <nete:forward>http://hr.company.com$0</nete:forward>
        </nete:case>
        <nete:case value="/employee">
           <nete:forward>http://employees.company.com$1
                        </nete:forward>
        </nete:case>
    </nete:cond>
```

**참고:** <nete:forward>*URL*</nete:forward> 요소는 같은 줄에 있어야 합니다. 이 예에서는 닫는 태그 </nete:forward>가 공간 제약으로 인해 별도의 줄에 표시되기도 하지만, 실제 프록시 규칙 파일에 줄 바꿈이 있으면 오류가 발생합니다. SPS 는 닫는 태그 </nete:forward> 앞의 줄 바꿈을 nete:forward 요소에 포함된 URL 의 일부 문자로 해석합니다.

## 전달 및 리디렉션 구문

요청을 전달하거나 리디렉션할 때 SPS 는 사용자가 지정한 URI(Universal Resource Indicator)의 일부 또는 모두를 유지 관리하기 위한 시스템을 사용합니다. 이 URI 는 대상 서버에 있는 리소스를 가리키며, 요청을 이행하려면 SPS 가 이 URI 를 해석해야 합니다.

전달 또는 리디렉션 대상에서 지정된 URL 에는 다음 중 하나가 추가될 수 있습니다.

**$0**

사용자 요청의 전체 URI 문자열을 프록시 규칙에 지정된 대상에 추가합니다.

예를 들어 프록시 규칙이 www.company.com 에 대한 모든 사용자 요청을 proxy.company.com$0 으로 전달하는 경우 사용자가 proxy.company.com/employees/hr/index.html 을 요청하면 해당 요청은 www.company.com/employees/hr/index.html 로 전달됩니다.

**$1**

nete:case 요소에서만 사용될 수 있습니다. 이때 부모 nete:cond 요소는 "다음으로 시작" 비교를 사용하여 URI 하위 문자열 일치를 지정합니다. $1 은 일치하는 텍스트의 오른쪽에 있는 모든 문자열이 전달 또는 리디렉션되는 요청에 추가됨을 나타냅니다.

예를 들어 프록시 규칙 구성 파일에 다음과 같은 nete:cond 요소가 있다고 가정하십시오.

<nete:cond type="uri" criteria="beginswith">

또한 이 조건은 www.company.com 의 호스트 이름과 다음과 같은 nete:case 요소에 대해 URI 를 평가하는 조건의 자식 조건이라고 가정하십시오.

```
<nete:case value="/hr">
    <nete:forward>http://hr.company.com$1</nete:forward>
    </nete:case>
```

이때 사용자가 다음을 요청합니다.

http://www.company.com/hr/employees/index.html

그러면 이 요청은 다음으로 전달됩니다.

http://hr.company.com/employees/index.html

**참고:** 이 예에서는 $1 매개 변수를 지정하므로 요청이 hr.company.com 으로 전달될 때 URI 의 /hr 부분은 생략됩니다.

## nete:cond

nete:cond 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:cond (nete:case+,nete:default)>
```

또한 nete:cond 요소에는 다음과 같은 특성이 있습니다.

```
<!ATTLIST nete:cond type (header | host | uri | query | cookie) #REQUIRED
criteria (equals | beginswith | endswith | contains | exists) "equals"
headername CDATA #IMPLIED>
```

nete:cond 요소는 CA SiteMinder for Secure Proxy Server 가 들어오는 요청을 처리할 방법을 결정하기 위해 평가되는 조건을 지정합니다. 이 요소에는 SPS 가 평가할 특성이 포함되어야 합니다.

가능한 특성으로는 ATTLIST 요소에 정의된 대로 다음이 포함됩니다.

**header**

HTTP 헤더를 지정합니다. HTTP 헤더는 SiteMinder 가 사용자 디렉터리에서 검색할 수 있는 이름-값 쌍입니다. type 으로 header 를 선택할 경우에는 헤더 이름도 지정해야 합니다. 다음은 type 으로 header 를 사용하는 nete:cond 요소의 예입니다.

```
<nete:cond type="header" headername="USER_AGENT">
```

이 요소는 요청의 대상을 결정하는 데 헤더가 사용되며 CA SiteMinder for Secure Proxy Server 에 의해 평가되는 헤더가 USER_AGENT 임을 나타냅니다. 요청의 실제 대상은 다음 단원에서 설명하는 대로 nete:cond 요소의 자식인 nete:case 요소에 의해 결정됩니다.

비교 유형으로 헤더를 사용하는 조건에는 추가 인수 headername="*value*"가 필요합니다. 여기서 *value* 는 HTTP 또는 SiteMinder 헤더의 이름입니다.

**참고:** SiteMinder 응답을 통해 생성된 HTTP 헤더는 nete:cond 요소에 지정될 수 있습니다.

**host**

배포 환경에서 CA SiteMinder for Secure Proxy Server 가 여러 가상 호스트에 대해 프록시하는 호스트 이름을 지정합니다.

포트 번호는 호스트의 일부로 간주되므로 호스트 조건과 함께 뒷부분에 설명된 endswith 조건을 사용하면 포트 번호를 기준으로 요청을 라우팅할 수 있습니다.

### 쿼리

URI 중 '?' 문자 다음에 나오는 쿼리 문자열 부분을 지정합니다. 이는 다음과 같이 URI 를 사용하는 nete:cond 와 유사합니다.

**URI**

요청된 URL 중 서버 이름 뒤에 나오는 부분인 URI(Universal Resource Indicator)를 지정합니다.

URI 조건과 함께 endswith 조건을 사용하면 파일 확장명을 기준으로 요청을 라우팅할 수 있습니다.

**cookie**

요청을 라우팅할 방법을 결정하려면 cookie 특성을 지정합니다. 쿠키 값이 인코딩된 경우 인코딩 매개 변수에서 인코딩 체계를 지정하십시오. CA SiteMinder for Secure Proxy Server 는 base64 인코딩 체계만 지원하고 쿠키 생성은 지원하지 않습니다. 다음은 인코딩된 쿠키의 가능한 사례입니다.

- 쿠키가 base64 를 사용하여 인코딩된 경우 value 특성에서 쿠키 값을 지정하고 nete:case 요소에서 인코딩 매개 변수로 base64 를 지정하십시오. CA SiteMinder for Secure Proxy Server 는 base64 인코딩 알고리즘을 사용하여 httprequest 에서 수신된 쿠키 값을 디코딩하고 디코딩된 값의 결과를 value 특성에 지정된 값과 비교합니다.
- 쿠키가 인코딩되지 않은 경우 value 특성에 쿠키 값을 일반 텍스트로 입력하고 nete:case 요소에서 인코딩 매개 변수를 빈 값으로 지정하십시오. CA SiteMinder for Secure Proxy Server 는 지정된 일반 텍스트를 쿠키 값으로 허용하고 nete:cond 를 확인합니다.

쿠키가 다른 인코딩 체계를 사용하여 인코딩된 경우 value 특성에 인코딩된 쿠키 값을 입력하고 nete:case 요소에서 인코딩 매개 변수를 빈 값으로 지정하십시오. CA SiteMinder for Secure Proxy Server 는 지정된 인코딩된 쿠키 값을 일반 텍스트로 허용하고 일반 텍스트 쿠키 값을 사용하여 nete:cond 를 확인합니다.

위에 설명된 type 특성 중 하나가 nete:cond 요소에 포함되어야 합니다. 또한 nete:cond 요소는 프록시 엔진이 들어오는 요청에 대해 조건 값을 대상으로 실행할 비교를 정의하는 조건을 지정해야 합니다. 가능한 조건은 다음과 같습니다.

**equals**

nete:cond 부모 요소의 type 특성 값이 nete:case 요소의 value 특성 내용과 같아야 요청이 처리됨을 나타냅니다.

**beginswith**

nete:cond 부모 요소의 type 특성 값이 nete:case 요소의 value 특성 내용으로 시작해야 요청이 처리됨을 나타냅니다.

**endswith**

nete:cond 부모 요소의 type 특성 값이 nete:case 요소의 value 특성 내용으로 끝나야 요청이 처리됨을 나타냅니다.

**contains**

nete:cond 부모 요소의 type 특성 값이 nete:case 요소의 value 특성 내용을 포함해야 요청이 처리됨을 나타냅니다.

**exists**

nete:cond 부모 요소가 있고 nete:case 요소의 value 특성이 true 여야 요청이 처리됨을 나타냅니다. exists 조건은 header 및 cookie 특성과 함께만 사용할 수 있습니다.

**참고:** 조건을 지정하지 않으면 CA SiteMinder for Secure Proxy Server 는 기본 조건인 equals 가 지정된 것으로 간주합니다.

각 nete:cond 요소에는 nete:case 자식 요소가 하나 이상 있어야 합니다. nete:case 자식 요소는 CA SiteMinder for Secure Proxy Server 가 적절한 대상으로 요청을 라우팅하는 데 사용하는 고유한 값을 제공합니다.

## nete:default

nete:default 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:default (nete:cond | nete:xprcond | nete:forward | nete:redirect |
nete:local)>
```

이 요소는 필수 요소이며 각 nete:cond 요소의 자식 요소여야 합니다. 요청이 nete:cond 요소에 포함된 nete:case 요소의 요구 사항을 충족하지 않는 경우 nete:default 요소가 해당 요청의 처리 방법을 결정합니다.

nete:default 요소와 연결될 수 있는 자식 요소는 nete:case 요소에 사용할 수 있는 요소와 동일합니다. nete:cond 요소를 nete:default 의 자식 요소로 생성하는 경우 SPS 가 가능한 모든 클라이언트 요청을 처리할 수 있도록 기본 사례를 주의 깊게 고려해야 합니다.

다음 예에서 nete:default 요소는 다른 모든 프록시 규칙의 조건을 충족하지 않는 요청을 모두 일반 정보가 포함된 홈 페이지로 전달합니다.

```
<nete:default>
    <nete:forward>http://home.company.com/index.html
    </nete:forward>
</nete:default>
```

<nete:forward>*URL*</nete:forward> 요소는 여는 태그와 닫는 태그를 포함하여 모두 같은 줄에 있어야 합니다. 이 예에서는 닫는 태그 </nete:forward>가 공간 제약으로 인해 별도의 줄에 표시되기도 하지만, 실제 프록시 규칙 파일에 줄 바꿈이 있으면 오류가 발생합니다. CA SiteMinder for Secure Proxy Server 는 닫는 태그 </nete:forward> 앞의 줄 바꿈을 nete:forward 요소에 포함된 URL 의 일부 문자로 해석합니다.

## nete:forward

nete:forward 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:forward (#PCDATA)>
```

nete:forward 요소는 요청을 지정된 URL 에 전달합니다.

**참고:** <nete:forward> 및 </nete:forward> 태그는 프록시 규칙 파일에서 한 줄에 있어야 합니다. 두 태그가 같은 줄에 있지 않으면 CA SiteMinder for Secure Proxy Server 는 줄 바꿈을 요소에 포함된 URL 의 일부로 해석합니다. 이 때문에 전달 서비스가 실패하게 됩니다.

다음 예에서 nete:forward 요소는 사용자가 요청한 URI 를 유지하면서 요청을 전달합니다.

```
<nete:forward>http://home.company.com$0</nete:forward>
```

사용자의 요청이 nete:case 부모 요소의 조건을 충족하는 경우 해당 요청은 home.company.com 으로 전달됩니다. 따라서 이전 nete:forward 요소가 전달한 http://server.company.com/hr/benefits/index.html 에 대한 요청은 요청을 http://home.company.com/hr/benefits/index.html 로 전달하는 방법으로 이행됩니다.

SSL 을 통해 요청을 전달하려면 <nete:forward> 요소에 포함된 대상을 정의할 때 http 대신 https 를 사용해야 합니다.

nete:forward 요소에는 다음 특성이 포함되어 있습니다.

```
<!ATTLIST nete:forward filter CDATA #IMPLIED>
```

이 특성을 사용하여 CA SiteMinder for Secure Proxy Server 에서 대상 서버로 전달하는 동안 호출할 수 있는 Java 필터 클래스의 이름을 지정할 수 있습니다. 필터는 필터 API 를 사용하여 작성할 수 있습니다.

**추가 정보:**

전달 및 리디렉션 구문 (페이지 174)
필터 API 개요 (페이지 305)

## 필터 특성

nete:forward 요소에는 다음 특성이 포함되어 있습니다.

```
<!ATTLIST nete:forward filter CDATA #IMPLIED>
```

이 특성을 사용하여 CA SiteMinder for Secure Proxy Server 에서 대상 서버로 전달하는 동안 호출할 수 있는 Java 필터 클래스의 이름을 지정할 수 있습니다. 필터는 필터 API 에 따라 작성할 수 있습니다.

## nete:redirect

nete:redirect 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:redirect (#PCDATA)>
```

nete:redirect 요소는 사용자에게 반환되며 사용자 요청을 적절한 대상 서버로 리디렉션하는 응답을 지정합니다. PCDATA 는 표준 전달 및 리디렉션 구문을 따릅니다. 리디렉션 후에는 CA SiteMinder for Secure Proxy Server 가 요청의 완료를 처리하지 않습니다. 대신 리디렉션 대상인 서버가 요청을 처리합니다.

<nete:redirect> 및 </nete:redirect> 태그는 프록시 규칙 파일에서 한 줄에 있어야 합니다. 두 태그가 같은 줄에 있지 않으면 CA SiteMinder for Secure Proxy Server 는 줄 바꿈을 요소에 포함된 URL 의 일부로 해석합니다. 이 때문에 리디렉션 서비스가 실패하게 됩니다.

다음 예에서 nete:redirect 요소는 사용자가 요청한 URI 를 유지하면서 요청을 리디렉션합니다. nete:forward 요소와 달리 요청이 리디렉션된 후에는 트랜잭션에서 CA SiteMinder for Secure Proxy Server 가 제외되고 대상 서버가 사용자에게 직접 리소스를 제공합니다.

```
<nete:redirect>http://home.company.com$0</nete:redirect>
```

http://www.company.com/hr/index.html 에 대한 사용자의 요청이 부모 nete:case 요소의 조건을 충족하고 위의 예에 의해 리디렉션되면 사용자가 리디렉션되고 사용자의 브라우저에는 다음과 같이 요청을 이행하는 대상 서버의 URL 이 표시됩니다.

http://home.company.com/hr/index.html

**참고:** SSL 을 통해 요청을 리디렉션하려면 <nete:redirect> 요소에 포함된 대상을 정의할 때 http 대신 https 를 사용해야 합니다.

**추가 정보:**

전달 및 리디렉션 구문 (페이지 174)

## nete:local

nete:local 요소는 향후 기능을 지원하기 위해 포함된 것으로, 프록시 규칙 구성 파일에 이 요소를 포함하면 안 됩니다.

## nete:xprcond

프록시 규칙에서 정규식을 조건으로 적용하려는 경우 nete:xprcond 요소를 nete:cond 요소처럼 사용할 수 있습니다. 정규식은 프록시 규칙에서 URI 문자열 및 연결된 쿼리 문자열을 평가하는 데 사용될 수 있습니다.

nete:xprcond 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:xprcond (nete:xpr+, nete:xpr-default)>
```

이 요소에는 nete:xpr 요소 하나 이상과 nete:xpr-default 요소 하나를 포함해야 합니다.

## nete:xpr 및 nete:xpr-default

nete:xpr 요소는 nete:cond 요소와 유사하며, 정규식과 CA SiteMinder for Secure Proxy Server 가 식의 일치 항목을 찾은 경우의 결과 동작을 기반으로 하는 규칙에 대한 다른 요소도 포함합니다. nete:xpr-default 요소에는 URI 또는 쿼리 문자열 조합이 nete:xprcond 요소 내의 nete:xpr 요소에 포함된 어떤 정규식과도 일치하지 않을 때의 기본 동작이 포함됩니다.

nete:xpr 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:xpr (nete:rule, nete:result)>
```

nete:xpr 요소에는 정규식을 정의하는 nete:rule 요소와, 정규식과 일치하는 문자열에 대한 동작을 지정하는 nete:result 요소가 포함됩니다.

nete:xpr-default 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:xpr-default (nete:forward|nete:redirect|nete:local)>
```

nete:xpr-default 요소는 CA SiteMinder for Secure Proxy Server 가 평가하는 URI 또는 쿼리 문자열이 부모 nete:xprcond 요소 내의 모든 nete:xpr 요소에 명시된 조건과 일치하지 않을 경우 수행해야 하는 전달 또는 리디렉션을 지정합니다.

## nete:rule 및 nete:result

nete:rule 및 nete:result 요소는 nete:xpr 요소에 포함되어야 합니다. nete:rule 요소는 SPS 가 들어오는 요청에 대해 평가하는 정규식을 지정합니다. nete:result 요소는 일치하는 요청에 대한 동작을 결정합니다.

nete:rule 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:rule (#PCDATA)>
```

이 요소는 정규식 문자열을 포함합니다. 요청이 nete:xpr 조건을 충족하는지 확인하기 위해 요청의 URI 및 쿼리 문자열이 이 정규식과 일치하는지 여부가 확인됩니다.

nete:result 요소는 다음과 같이 정의됩니다.

```
<!ELEMENT nete:result (#PCDATA)>
```

nete:result 요소에는 다음과 같은 특성이 있을 수 있습니다.

```
<!ATTLIST nete:result service (forward|redirect) "forward">
```

이 요소는 SPS 가 서비스(전달 또는 리디렉션 서비스)에 전달할 URL 을 생성하는 데 사용하는 대체 문자열(URL)로 구성된 문자열을 포함합니다. service 특성은 URL 을 수신할 적절한 서비스를 지정하는 데 사용됩니다. 기본 서비스는 server.conf 구성 파일에 정의된 전달 서비스입니다.

nete:result 요소의 대체 URL 은 $#을 선택적으로 포함하는 URL 이어야 합니다. 여기서 #은 0, 1, 2 등입니다.

- $0 은 평가되는 전체 URI 및 쿼리 문자열입니다.
- $n 은 요청된 URI 중 연결된 nete:rule 요소에 설명된 정규식의 n 번째 괄호 집합과 일치하는 부분입니다.

예를 들어 프록시 규칙 집합에는 다음과 같은 내용이 포함될 수 있습니다.

```
<nete:xprcond>
    <nete:xpr>
        <nete:rule>^/realma(.*)</nete:rule>
        <nete:result>http://server1.company.com$1</nete:result>
    </nete:xpr>


    <nete:xpr-default>
        <nete:forward>http://www.company.com$0</nete:forward>
    </nete:xpr-default>
</nete:xprcond>
```

<nete:result>URL</nete:result>태그는 프록시 규칙 파일에서 한 줄에 있어야 합니다. 두 태그가 같은 줄에 있지 않으면 CA SiteMinder for Secure Proxy Server 는 줄 바꿈을 요소에 포함된 URL 의 일부로 해석합니다. 이 때문에 결과 서비스가 실패하게 됩니다.

앞의 nete:xprcond 프록시 규칙 예에서 다음 리소스가 요청된다고 가정하십시오.

http://www.company.com/realma/index.html

이 요청은 다음으로 전달됩니다.

http://server1.company.com/index.html

## nete:xprcond 요소의 작동 방식

nete:xprcond 요소의 처리 방식은 다른 모든 nete:cond 요소의 처리 방식과 유사합니다. CA SiteMinder for Secure Proxy Server 가 요청을 처리할 때 프록시 규칙 구성 파일에 nete:xprcond 요소가 있으면 다음 작업이 수행됩니다.

1. CA SiteMinder for Secure Proxy Server 가 nete:xprcond 요소에서 첫 번째 nete:xpr 요소를 검사합니다.
2. 프록시 엔진이 요청의 URI 및 쿼리 문자열을 기준으로 nete:rule 요소에 설명된 정규식을 평가합니다.
3. 요청된 URI 와 쿼리 문자열이 nete:rule 요소에 지정된 정규식과 일치하면 CA SiteMinder for Secure Proxy Server 는 일치 결과를 사용하여 결과 문자열을 확인하며, 요청은 nete:result 요소에서 파생된 URL 을 사용하여 지정된 서비스로 전달됩니다.
4. 요청된 URI 와 쿼리 문자열이 첫 번째 nete:xpr 요소의 정규식과 일치하지 않으면 프록시 규칙 엔진은 2 단계와 3 단계를 반복하여 다음 nete:xpr 요소를 평가합니다. 이 프로세스는 규칙 엔진이 일치 항목을 찾거나 nete:xpr-default 요소에 도달할 때까지 계속됩니다.
5. nete:xpr-default 요소에 도달하기 전에 일치 항목이 발견되지 않으면 nete:xpr-default 요소의 내용에 따라 요청을 라우팅할 방법이 결정됩니다.

# 정규식 구문

이 단원에서는 nete:rule 요소에 대한 정규식을 구성하는 데 사용되는 구문에 대해 설명합니다. nete:xprcond 요소의 형식은 다음과 같습니다.

```
<nete:xprcond>
    <nete:xpr>
        <nete:rule>regular_expression</nete:rule>
        <nete:result>result</nete:result>
    </nete:xpr>
    <nete:xpr-default>forward_destination</nete:xpr-default>
</nete:xprcond>
```

nete:xpr 요소에서 nete:rule 요소는 다음 표에 설명된 구문을 사용하는 정규식으로 구성해야 합니다. 이 구문은 Apache 가 지원하는 정규식 구문(http://www.apache.org 참조)과 일치합니다.

| 문자 | 결과 |
|---|---|
| 유니코드 문자 | 동일한 유니코드 문자와 일치 |
| \ | 메타 문자(예: '*')를 나타내는 데 사용됨 |
| \\ | 단일 '\' 문자와 일치 |
| \0nnn | 지정된 8 진수 문자와 일치 |
| \xhh | 지정된 8 비트 16 진수 문자와 일치 |
| \\uhhhh | 지정된 16 비트 16 진수 문자와 일치 |
| \t | ASCII 탭 문자와 일치 |
| \n | ASCII 줄 바꿈 문자와 일치 |
| \r | ASCII 리턴 문자와 일치 |
| \f | ASCII form feed 문자와 일치 |
| [abc] | 단순 문자 클래스 |
| [a-zA-Z] | 범위가 있는 문자 클래스 |
| [^abc] | 부정 문자 클래스 |
| [:alnum:] | 영숫자 |
| [:alpha:] | 영문자 |
| [:blank:] | 공백 및 탭 문자 |

| 문자 | 결과 |
|---|---|
| [:cntrl:] | 제어 문자 |
| [:digit:] | 숫자 |
| [:graph:] | 인쇄 및 표시가 가능문자(공백은 인쇄 가능하지만 표시되지는 않는 반면 'a'는 인쇄 가능하고 표시됨) |
| [:lower:] | 소문자 영문자 |
| [:print:] | 인쇄 가능한 문자(제어 문자가 아닌 문자) |
| [:punct:] | 문장 부호(문자, 숫자, 제어 문자 또는 공백이 아닌 문자) |
| [:space:] | 공백 문자(예: 공백, 탭 및 formfeed) |
| [:upper:] | 대문자 영문자 |
| [:xdigit:] | 16 진수 문자 |
| [:javastart:] | Java 식별자의 처음 |
| [:javapart:] | Java 식별자의 일부 |
| . | 새 줄이 아닌 모든 문자와 일치 |
| \w | "단어" 문자(영숫자 + "_")와 일치 |
| \W | 단어가 아닌 문자와 일치 |
| \s | 공백 문자와 일치 |
| \S | 공백이 아닌 문자와 일치 |
| \d | 숫자와 일치 |
| \D | 숫자가 아닌 문자와 일치 |
| ^ | 줄의 시작 부분에서만 일치 |
| $ | 줄의 끝 부분에서만 일치 |
| \b | 단어 경계에서만 일치 |
| \B | 비 단어 경계에서만 일치 |
| A* | A 와 0 번 이상 일치(확장) |
| A+ | A 와 1 번 이상 일치(확장) |
| A? | A 와 1 번 또는 0 번 일치(확장) |

| 문자 | 결과 |
| --- | --- |
| A{n} | A 와 정확히 n 번 일치(확장) |
| A{n,} | A 와 n 번 이상 일치(확장) |
| A{n,m} | A 와 n 번 이상 m 번 이하 일치(확장) |
| A*? | A 와 0 번 이상 일치(축소) |
| A+? | A 와 1 번 이상 일치(축소) |
| A?? | A 와 0 번 또는 1 번 일치(축소) |
| AB | 순서대로 A 및 B 와 일치 |
| A\|B | A 또는 B 와 일치 |
| (A) | 하위 식 그룹화에 사용됨 |
| \1 | 첫 번째 괄호 안의 하위 식에 대한 후참조 |
| \n | n 번째 괄호 안의 하위 식에 대한 후참조 |

모든 종료 연산자(+, *, ?, {m,n})는 기본적으로 확장이므로 전체 일치가 실패하지 않고 가능한 한 많은 문자열 요소와 일치합니다. 종료를 축소(확장 아님)로 설정하려는 경우 뒤에 '?'만 추가하면 됩니다. 축소 종료는 일치 항목을 찾을 때 가능한 한 적은 문자열 요소와 일치합니다. {m,n} 종료는 현재 축소를 지원하지 않습니다.

## nete:rule 및 nete:result 의 정규식 예

정규식은 CA SiteMinder for Secure Proxy Server 프록시 규칙에 사용할 수 있는 매우 유연하고 강력한 도구입니다. 이 단원에서는 프록시 규칙에 nete:rule 요소를 사용하는 몇 가지 예를 제공합니다. 또한 이러한 예에는 nete:rule 에서 그룹화를 사용하여 nete:xprcond 요소의 자식 요소에 의해 생성된 대상에 영향을 주는 방법을 보여 주기 위한 nete:result 요소의 다양한 사용 사례도 포함되어 있습니다.

### 여러 대상 서버에 단일 규칙 매핑

다음 예에서 nete:rule 요소에는 요청을 서로 다른 여러 대상으로 전달하는 데 사용할 수 있는 정규식이 포함되어 있습니다. 이 예에서는 CA SiteMinder for Secure Proxy Server 가 다음과 같은 형식의 URI 를 수신한다고 가정합니다.

/GOTO=*some path and or filename*

nete:xprcond 요소에는 다음과 같은 자식 요소가 포함됩니다.

```
<nete:xpr>
    <nete:rule>/GOTO=(.*)/(.*)</nete:rule>
    <nete:result>http://$1/$2</nete:result>
</nete:xpr>
```

nete:rule 요소의 정규식과 관련 nete:result 요소는 /GOTO=string 을 생성하는 URI 와 일치합니다. 일치 항목을 찾으면 CA SiteMinder for Secure Proxy Server 는 URI 에서 = 기호 다음의 첫 번째 문자열을 결과의 $1 값으로 사용하고, = 기호 다음에 나타나는 첫 번째 / 기호 다음의 값을 $2 결과로 사용합니다. nete:result 요소는 이러한 요소를 조합하여 URL 을 생성합니다. 기본적으로 nete:result 요소는 CA SiteMinder for Secure Proxy Server 전달 서비스를 사용합니다.

예를 들어 위에서 설명한 nete:xpr 요소에 의해 평가되는 요청의 URI 가 다음과 같다고 가정하십시오.

/GOTO=server1.company.com/index.html

그러면 nete:rule 요소의 정규식은 일치 항목을 찾은 후 $1 의 값을 server1.company.com 으로, $2 의 값을 index.html 로 할당합니다. nete:result 요소는 이러한 값을 다음 URL 로 결합합니다.

http://server1.company.com/index.html

이 URL 은 CA SiteMinder for Secure Proxy Server 가 요청을 확인하는 데 사용하는 대상입니다.

### 사용자를 리디렉션하는 정규식

nete:result 요소를 사용하여 리소스를 요청하는 사용자에게 반환되는 리디렉션 응답을 생성할 수도 있습니다. 그러면 인증 및 권한 부여 후에 CA SiteMinder for Secure Proxy Server 가 아니라 서버에서 요청 이행이 처리됩니다. 다음은 nete:result 자식 요소에서 리디렉션을 지정하는 nete:xpr 요소의 예입니다.

```
<nete:xpr>
    <nete:rule>/REDIR=(.*)/(.*)</nete:rule>
    <nete:result service="redirect">http://$1/$2</nete:result>
</nete:xpr>
```

**참고:** service 특성은 CA SiteMinder for Secure Proxy Server 가 기본 전달 서비스 대신 리디렉션 서비스를 사용하도록 합니다.

## 전달, 리디렉션 및 결과 필터의 헤더 값

HTTP 헤더 또는 SiteMinder 응답 헤더의 값은 nete:forward, nete:redirect 또는 nete:result 요소로 직접 대체될 수 있습니다. 전달 또는 리디렉션 요소의 URI 나 결과 필터 요소의 규칙에 {{*HEADER_NAME*}}이 포함되어 있는 경우 프록시 엔진은 전달, 리디렉션 또는 결과를 확인하기 전에 요청에서 지정된 헤더와 대체 헤더 값이 일치하는 헤더를 검색합니다. 요청에 일치하는 헤더가 없으면 프록시 엔진은 헤더 값 대신 빈 문자열로 대체합니다.

**참고:** 헤더 이름은 대/소문자를 구분합니다.

## nete:forward 의 동적 헤더 값

nete:forward 요소의 일부로 동적 헤더 값을 사용하려면 전달의 URL 부분에 *{{HEADER_NAME}}*을 삽입하기만 하면 됩니다. 예를 들면 다음과 같습니다.

```
<nete:forward>http://www.company.com/{{RESPONSE1}}$1</nete:forward>
```

단일 nete:forward 요소에 여러 헤더를 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
<nete:forward>http://www.company.com/{{RESPONSE1}}/{{RESPONSE2}}$1
</nete:forward>
```

## nete:redirect 의 동적 헤더 값

nete:redirect 요소의 일부로 동적 헤더 값을 사용하려면 리디렉션의 URL 부분에 *{{HEADER_NAME}}*을 삽입하기만 하면 됩니다. 예를 들면 다음과 같습니다.

```
<nete:redirect>http://www.company.com/{{RESPONSE1}}$1</nete:redirect>
```

단일 nete:redirect 요소에 여러 헤더를 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
<nete:redirect>http://www.company.com/{{RESPONSE1}}/{{RESPONSE2}}$1
</nete:redirect>
```

## nete:result 의 동적 헤더 값

nete:result 요소의 일부로 동적 헤더 값을 사용하려면 결과의 URL 부분에 *{{HEADER_NAME}}*을 삽입하기만 하면 됩니다. 예를 들면 다음과 같습니다.

```
<nete:result>http://www.company.com/{{HEADER_NAME}}$1</nete:result>
```

필터와 같은 프록시 규칙의 다른 기능을 동적 헤더 값과 함께 사용할 수 있습니다. 예를 들면 다음과 같습니다.

```
<nete:result filter="filter1">http://$1/$2?{{HEADER_NAME}}</nete:result>
```

# 응답 처리

CA SiteMinder for Secure Proxy Server 는 SiteMinder 응답을 사용하여 요청의 대상을 결정합니다. CA SiteMinder for Secure Proxy Server 를 통해 라우팅되는 트랜잭션에는 CA SiteMinder for Secure Proxy Server 웹 에이전트와 SiteMinder 간의 상호 작용이 포함되므로 CA SiteMinder for Secure Proxy Server 는 인증 및 권한 부여 프로세스 중에 수집되는 모든 SiteMinder 응답을 사용하여 요청의 대상을 결정할 수 있습니다.

예를 들어 사용자 디렉터리에 뱅킹 웹 사이트의 계정 유형에 대한 정보가 포함되어 있는 경우 CA SiteMinder for Secure Proxy Server 는 사용자를 계정 유형별로 서로 다른 대상으로 프록시할 수 있습니다. 이를 통해 기업은 우량 고객에게 더 높은 품질의 서비스를 제공할 수 있습니다. 표준 계정을 가진 고객은 대상 서버 집합 하나에서 처리하고 프리미엄 계정을 가진 고객은 별도의 고성능 대상 서버에서 처리할 수 있습니다.

# 프록시 규칙 수정

프록시 규칙을 수정하려면 텍스트 편집기를 사용하여 프록시 규칙 XML 구성 파일을 편집해야 합니다. 프록시 규칙은 XML 파일이므로 프록시 규칙 구성 파일은 올바르게 구성되고 유효해야 합니다. 올바르게 구성된 XML 파일의 태그는 모두 여는 태그와 닫는 태그로 구성되어야 합니다. 또한 유효한 파일은 proxyrules.dtd 의 지침을 따라야 합니다.

프록시 규칙 XML 구성 파일의 변경 내용은 SPS 에 의해 자동으로 적용됩니다. CA SiteMinder for Secure Proxy Server 는 요청을 수신하면 프록시 규칙이 변경되었는지 여부를 확인합니다. 파일이 변경되었으면 요청을 이행하기 전에 규칙이 다시 로드됩니다.

**참고:** server.conf 파일의 <ServiceDispatcher> 요소에 있는 rules_file 지시문에서 프록시 규칙 XML 구성 파일의 이름을 변경할 경우 CA SiteMinder for Secure Proxy Server 를 다시 시작해야 합니다.

# 샘플 프록시 규칙 구성 파일

CA SiteMinder for Secure Proxy Server 는 몇 가지 프록시 규칙 구성 파일 예제를 설치합니다. 프록시 규칙 파일을 생성할 때 이러한 예제 XML 파일을 기초로 사용할 수 있습니다.

이러한 예제 파일은 *sps_home*\secure-proxy\proxy-engine\examples\proxyrules 디렉터리에 있습니다. 이 안내서의 설명을 읽을 때 예제 파일을 함께 보는 것이 좋습니다.

파일을 복사하여 필요에 맞게 사용자 지정할 수도 있습니다.

**프록시 규칙 파일을 사용자 지정 및 배포하려면**

1. *sps_home*\secure-proxy\proxy-engine\examples\proxyrules 디렉터리로 이동합니다.
2. 사용할 예제 파일의 복사본을 만듭니다.
3. 내용을 사용자 지정한 후 새 파일을 고유한 이름으로 저장합니다.
4. 수정한 파일을 *sps_home*/secure-proxy/proxy-engine/conf 디렉터리에 복사합니다.
5. server.conf 파일을 열고 파일의 프록시 규칙 섹션을 수정하여 사용자 지정한 파일을 가리키도록 합니다.

## 프록시 규칙 예 - 가상 호스트를 기준으로 요청 라우팅

예제 파일인 proxyrules_example1.xml 파일은 요청에 지정된 호스트 이름을 기준으로 요청을 라우팅합니다. 또한 예제 파일인 proxyrules_example10.xml 파일은 요청에 지정된 호스트 이름을 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅하고, CA SiteMinder for Secure Proxy Server 는 프록시 규칙의 PID 를 사용하여 프록시 규칙이 트리거된 횟수를 계산합니다. CA SiteMinder for Secure Proxy Server 를 모니터링하도록 CA Wily Introscope 를 구성한 경우 이 횟수는 CA Wily Introscope 데이터 메트릭에 표시됩니다.

이 파일에서는 간단한 프록시 규칙 집합이 요청된 리소스에 지정된 가상 호스트를 기준으로 사용자 요청을 라우팅합니다. bondtrading.company.com 서버에 대한 모든 요청은 server2 로 전달되고, banking.company.com 에 대한 모든 요청은 server1 로 전달되며, 다른 모든 요청은 회사 홈 서버로 전달됩니다. 홈 서버는 다른 모든 nete:cond 요소의 조건과 일치하지 않는 요청이 기본적으로 전달되는 위치입니다.

**참고:** 포트 번호는 사용자가 요청한 가상 호스트의 일부로 간주되므로 nete:case 요소에서 포트를 지정해야 합니다. 포트 번호가 필요하지 않도록 하려면 beginswith 조건을 사용하십시오.

다음 표에서는 가상 호스트를 기준으로 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| **요청된 URL** | **전달된 URL** |
|---|---|
| http://banking.company.com/index.html | http://server1.company.com/index.html |
| http://bondtrading.company.com/index.html | http://server2.company.com/index.html |
| http://www.company.com/index.html | http://home.company.com/index.html |

## 프록시 규칙 예 - 헤더 값을 기준으로 요청 라우팅

예제 파일인 proxyrules_example2.xml 파일은 HTTP 헤더의 값을 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다. HTTP 헤더는 표준 헤더이거나 SiteMinder 응답을 사용하여 생성된 헤더일 수 있습니다.

**참고:** SiteMinder 응답에 대한 자세한 내용은 *CA SiteMinder Policy Design*(CA SiteMinder 정책 설계)을 참조하십시오.

이 예에서는 CA SiteMinder for Secure Proxy Server 가 www.company.com 의 기본 가상 호스트에 대한 요청을 라우팅하는 것으로 가정합니다.

이 파일에서 HTTP 헤더 변수 "HEADER"의 값은 요청의 대상을 결정합니다.

다음 표에서는 HTTP 헤더를 기준으로 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| 요청된 URL | 전달된 URL |
|---|---|
| http://www.company.com/index.html<br>HTTP_HEADER 가 다음 값을 갖습니다.<br>HTTP_HEADER="value1" | http://server1.company.com/index.html |
| http://www.company.com/index.html<br>HTTP_HEADER 가 다음 값을 갖습니다.<br>HTTP_HEADER="value2" | http://server2.company.com/index.html |
| http://www.company.com/index.html<br>HTTP_HEADER 가 value1 또는 value2 가 아닌 다른 값을 갖습니다. | http://home.company.com/index.html |

**참고:** nete:cond 요소에는 헤더 변수 이름의 HTTP_를 포함할 필요가 없습니다. CA SiteMinder for Secure Proxy Server 는 헤더 변수 이름에 대해 HTTP_를 가정합니다.

헤더 값을 사용하는 프록시 규칙은 원하는 서비스 수준을 기준으로 요청을 전달할 수 있는 뛰어난 방법입니다. 예를 들어 사용자 계정 유형을 포함하는 HTTP 헤더 변수의 값을 사용하면 프리미엄 계정을 가진 고객을 위해 요청을 고성능 서버로 분산할 수 있습니다.

## 프록시 규칙 예 - 장치 유형을 기준으로 요청 라우팅

예제 파일인 proxyrules_example3.xml 파일은 리소스에 액세스하는 데 사용된 장치의 유형을 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다.

**참고:** 사용자 에이전트 HTTP 헤더 값은 요청을 라우팅할 방법을 결정하는 데 사용됩니다.

이 파일에서 브라우저(사용자 에이전트에 웹 브라우저용 Mozilla 가 포함됨)를 사용하여 리소스에 액세스하는 사용자는 웹 서버로 전달되고 다른 모든 사용자는 무선 서버로 전달됩니다.

다음 표에서는 장치 유형을 기준으로 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| 요청된 URL | 전달된 URL |
|---|---|
| http://www.company.com/index.html<br>사용자가 웹 브라우저를 통해 리소스에 액세스합니다. | http://home.company.com/index.html |
| http://www.company.com/index.wml<br>사용자가 무선 장치를 통해 리소스에 액세스합니다. | http://wireless.company.com/index.wml |

## 프록시 규칙 예 - URI 를 사용하여 요청 라우팅

예제 파일인 proxyrules_example4.xml 파일은 사용자 요청에 지정된 URI 를 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다.

다음 표에서는 URI 를 기준으로 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| 요청된 URL | 전달된 URL |
|---|---|
| http://www.company.com/dir1/index.html | http://server1.company.com/index.html |
| http://www.company.com/dir2/index.html | http://server2.company.com/index.html |

| 요청된 URL | 전달된 URL |
| --- | --- |
| http://www.company.com/index.html | http://home.company.com/index.html |

## 프록시 규칙 예 - 파일 확장명을 기준으로 요청 라우팅

예제 파일인 proxyrules_example5.xml 파일은 사용자가 요청한 파일 확장명을 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다. 이를 위해 URI 조건과 endswith 조건이 사용됩니다.

파일에서 <nete:forward> 및 </nete:forward> 태그는 공간 제약으로 인해 별도의 줄에 표시됩니다. 그러나 프록시 규칙 구성 파일에서는 <nete:forward> 요소의 여는 태그와 닫는 태그가 같은 줄에 표시되어야 합니다. 그렇지 않으면 CA SiteMinder for Secure Proxy Server 는 줄 바꿈을 전달 URL 의 일부로 해석하므로 요청이 올바르게 전달되지 않습니다.

앞의 예에서 .jsp 리소스에 액세스하는 사용자는 응용 프로그램 서버로 전달되고 무선 사용자는 무선 서버로 전달됩니다. 다른 모든 사용자는 홈 서버로 전달됩니다.

다음 표에서는 파일 확장명을 기준으로 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| 요청된 URL | 전달된 URL |
| --- | --- |
| http://www.company.com/app.jsp | http://application.company.com/app.jsp |
| http://www.company.com/index.wml | http://wireless.company.com/index.wml |
| http://www.company.com/index.html | http://home.company.com/index.html |

## 프록시 규칙 예 - 중첩된 조건을 사용하여 요청 라우팅

예제 파일인 proxyrules_example6.xml 파일은 호스트 이름, 특정 헤더 및 장치 유형을 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다. 이 파일은 CA SiteMinder for Secure Proxy Server 가 단일 구성 파일에서 복잡한 관계를 처리하는 방식을 보여 줍니다.

이 파일에서 <nete:forward>*URL*</nete:forward> 요소는 같은 줄에 있어야 합니다. 이 예에서는 닫는 태그 </nete:forward>가 공간 제약으로 인해 별도의 줄에 표시되기도 하지만, 실제 프록시 규칙 파일에 줄 바꿈이 있으면 오류가 발생합니다. CA SiteMinder for Secure Proxy Server 는 닫는 태그 </nete:forward> 앞의 줄 바꿈을 nete:forward 요소에 포함된 URL 의 일부 문자로 해석합니다.

다음 표에서는 중첩된 조건과 함께 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| **요청된 URL** | **전달된 URL** |
|---|---|
| http://banking.company.com/index.wml | http://wireless.company.com/banking/index.wml |
| http://banking.company.com/index.html | http://server1.company.com/banking/index.html |
| http://bondtrading.company.com/index.html<br>헤더 값 GOLD_USER="yes" 사용 | http://fast.company.com/bondtrading/index.html |
| http://bondtrading.company.com/index.html<br>헤더 값 GOLD_USER="no" 사용 | http://server2.company.com/bondtrading/index.html |
| http://www.company.com/index.wml<br>무선 장치 이름을 포함하는 USER_AGENT 헤더 값 사용 | http://home.company.com/<br>wireless/index.wml |
| http://www.company.com/index.html<br>무선 장치 이름을 포함하지 않는 USER_AGENT 헤더 값 사용 | http://home.company.com/index.html |

## 프록시 규칙 예 - 프록시 규칙에 정규식 사용

예제 파일인 proxyrules_example7.xml 파일은 정규식이 포함된 nete:xprcond 요소를 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다. 정규식은 요청의 URI 와 쿼리 문자열에 따라 평가됩니다.

이 파일에서 요청의 URI 및 쿼리 문자열은 nete:xpr 요소에 정의된 세 개의 정규식을 기준으로 평가됩니다. 첫 번째 nete:xpr 요소에 대해 일치 항목을 찾을 수 없으면 CA SiteMinder for Secure Proxy Server 는 두 번째 정규식과 마지막으로 세 번째 정규식에 대해 일치 항목을 찾습니다. 일치 항목을 찾을 수 없으면 요청을 처리하는 데 nete:xpr-default 조건이 사용됩니다.

다음 표에서는 정규식 프록시 규칙을 사용한 요청 결과를 보여 줍니다.

| 요청된 URL | 전달된 URL |
|---|---|
| http://server.company.com/realma/hr/index.html | http://server1.company.com/hr/index.html |
| http://server.company.com/GOTO=server2.company.com/index.html | http://server2.company.com/index.html |
| http://server.company.com/REDIR=server2.company.com/index.html | http://server2.company.com/index.html<br>사용자가 리디렉션되므로 server2.company.com 이 직접 사용자의 요청을 이행해야 합니다. |
| http://server.company.com/index.html | http://www.company.com/index.html |

## 프록시 규칙 예 - 쿠키 존재 여부를 기준으로 요청 라우팅

예제 파일인 proxyrules_example8.xml 파일은 쿠키가 있는지 여부를 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다.

이 예에서 CA SiteMinder for Secure Proxy Server 는 요청에 쿠키 헤더 "mycookie"가 포함되어 있는 경우 요청을 www.company.com 으로 라우팅합니다.

## 프록시 규칙 예 - 쿠키 값을 기준으로 요청 라우팅

예제 파일인 proxyrules_example9.xml 파일은 쿠키의 값을 기준으로 CA SiteMinder for Secure Proxy Server 요청을 라우팅합니다.

이 예에서 CA SiteMinder for Secure Proxy Server 는 요청에 쿠키 헤더 "mycookie"가 포함되어 있고 요청에 인코딩 메커니즘이 지정되지 않은 경우 요청을 www.abcd.com 으로 라우팅합니다. 요청에 쿠키 헤더 "mycookie"와 base64 인코딩 메커니즘이 포함되어 있는 경우 CA SiteMinder for Secure Proxy Server 는 요청을 www.xyz.com 으로 라우팅합니다.

# 제 10 장: SPS 배포

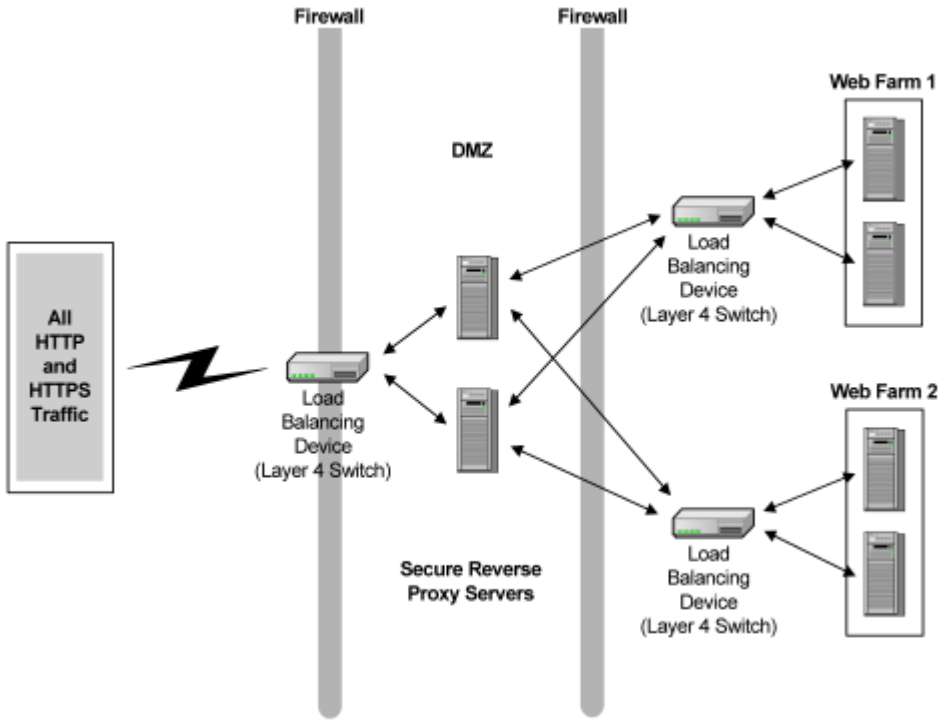
이 섹션은 다음 항목을 포함하고 있습니다.



## 엔터프라이즈의 CA SiteMinder for Secure Proxy Server

CA SiteMinder for Secure Proxy Server 는 리버스 프록시 아키텍처를 사용하여 액세스 제어, 싱글 사인온 및 SSL 가속화가 가능하게 합니다. 그러나 콘텐츠 캐싱 기능과 기존 리버스 프록시 서버가 제공하는 몇 가지 다른 기능은 제공하지 않습니다. CA SiteMinder for Secure Proxy Server 는 다른 프록시 기술을 대체하기 위한 것이 아니라 엔터프라이즈 아키텍처를 보강하기 위한 것입니다. 따라서 CA SiteMinder for Secure Proxy Server 는 양쪽에 부하 분산 장치와 캐싱 장치를 포함하고 있는 클러스터에 배치할 수 있습니다.

다음 그림에서는 CA SiteMinder for Secure Proxy Server 를 네트워크에 추가하여 부하 분산 장치와 함께 작동하도록 하는 방법을 보여 줍니다.



**참고:** 부하 분산 장치 외에 캐싱 장치도 CA SiteMinder for Secure Proxy Server 클러스터의 양쪽에 배치할 수 있습니다.

## 고정 비트 부하 분산

CA SiteMinder for Secure Proxy Server 에서 지원되는 쿠키를 사용하지 않는 세션 체계를 사용하는 경우 CA SiteMinder for Secure Proxy Server 를 통해 리소스에 액세스하는 사용자의 세션 정보는 메모리 내 세션 저장소에서 유지 관리됩니다. 세션 정보는 사용자가 처음 인증된 CA SiteMinder for Secure Proxy Server 에서 유지 관리되므로 단일 세션의 모든 사용자 요청에 동일한 CA SiteMinder for Secure Proxy Server 가 사용됩니다. 클러스터에 구현된 경우 CA SiteMinder for Secure Proxy Server 를 고정 비트 부하 분산과 함께 사용하여 동일한 CA SiteMinder for Secure Proxy Server 에 대해 일관된 연결을 제공하고, 기존 SiteMinder 쿠키 세션 체계가 아닌 다른 세션 체계를 사용할 때 싱글 사인온이 가능하도록 해야 합니다.

쿠키를 사용하지 않는 세션 체계를 사용하여 CA SiteMinder for Secure Proxy Server 를 배포하려면 다음을 고려해야 합니다.

- 대부분의 배포 환경에서 CA SiteMinder for Secure Proxy Server 는 들어오는 요청의 부하가 여러 서버로 분산되는 클러스터에 배포됩니다. 부하 분산은 부하 분산 장치에 의해 처리됩니다. 이러한 장치는 고정 비트 기능이 있어야 싱글 사인온을 유지 관리할 수 있습니다.

  고정 비트 부하 분산은 클러스터의 특정 CA SiteMinder for Secure Proxy Server 를 사용하여 사용자의 세션이 설정되고 나면 해당 CA SiteMinder for Secure Proxy Server 가 사용자의 요청을 모두 처리할 수 있도록 합니다. CA SiteMinder for Secure Proxy Server 는 쿠키를 사용하지 않는 세션에 대한 세션 정보를 활성 메모리에서 유지 관리하므로 이 기능이 필요합니다. 사용자 요청이 고정 비트 기술을 사용하여 처리되지 않는 경우에는 서버 클러스터에 있는 다른 CA SiteMinder for Secure Proxy Server 가 요청을 이행할 때마다 사용자에게 새 자격 증명이 요청됩니다.
- CA SiteMinder for Secure Proxy Server 에 대한 설정을 구성할 때는 CA SiteMinder for Secure Proxy Server 의 server.conf 파일에 정의된 기본 가상 호스트를 부하 분산 장치의 이름 및 IP 주소를 사용하여 정의해야 합니다.
- 부하 분산 장치는 CA SiteMinder for Secure Proxy Server 에 대한 진입점으로 구성해야 합니다.
- 부하 분산 장치는 CA SiteMinder for Secure Proxy Server 클러스터를 가리켜야 합니다.

- *sps_home*/secure-proxy/httpd/conf 디렉터리에 있는 httpd.conf 파일을 수정하여 ServerName 지시문의 값을 CA SiteMinder for Secure Proxy Server 가 설치된 시스템이 아니라 부하 분산 장치의 이름으로 설정해야 합니다.
- SSL 을 사용하는 경우 인증서는 CA SiteMinder for Secure Proxy Server 가 아니라 부하 분산 장치에 발급해야 합니다.
- CA SiteMinder for Secure Proxy Server 를 설치한 시스템에는 메모리 내 세션 저장소에서 유지 관리할 동시 사용자 세션을 위한 메모리가 세션당 약 1 KB 씩 있어야 합니다. 예를 들어 단일 시스템이 1,000 개의 동시 세션을 유지 관리해야 한다면 이 시스템에는 이 용도로 사용할 수 있는 1 MB 의 RAM 이 있어야 합니다.

## 신뢰할 수 있는 사이트 및 신뢰할 수 없는 사이트로의 프록시

CA SiteMinder for Secure Proxy Server 는 엔터프라이즈 내 신뢰할 수 있는 사이트를 프록시합니다. 프록시 트랜잭션을 처리하는 과정에서 SiteMinder 가 생성한 HTTP 헤더 변수와 SiteMinder 응답에 의해 생성된 모든 변수는 각 HTTP 및 HTTPS 요청과 함께 전달됩니다. 이러한 응답은 다른 엔터프라이즈 응용 프로그램에서 사용될 수 있습니다.

**중요!** 신뢰할 수 없는 사이트의 콘텐츠를 프록시하는 트랜잭션에서 CA SiteMinder for Secure Proxy Server 를 사용할 경우 트랜잭션에 사용되는 헤더도 신뢰할 수 없는 사이트로 전달됩니다. 따라서 CA SiteMinder for Secure Proxy Server 는 엔터프라이즈에서 신뢰하는 대상을 프록시하는 데 사용하는 것이 좋습니다.

## 가상 호스트 구성

CA SiteMinder for Secure Proxy Server 를 여러 호스트로 구성하고 하나 이상의 호스트 이름에 대해 가상 호스트의 역할을 하도록 할 수 있습니다.

**다음 단계를 수행하십시오.**

1. server.conf 파일의 <VirtualHost> 매개 변수를 편집하여 CA SiteMinder for Secure Proxy Server 가 하나 이상의 호스트 이름에 대해 가상 호스트의 역할을 하도록 구성합니다.
2. 포함된 Apache 웹 서버의 구성 파일을 편집합니다.

**추가 정보:**

가상 호스트 이름 구성 (페이지 154)

## 여러 가상 호스트를 처리하도록 Apache 구성 파일 편집

여러 가상 호스트를 CA SiteMinder for Secure Proxy Server 와 함께 동일한 운영 환경에서 실행 중이고 이 환경에서 트랜잭션이 실행되는 경우 Apache 구성 파일(httpd.conf)을 업데이트하십시오. 이 파일은 *sps_home*\secure-proxy\httpd\conf 디렉터리에 있습니다. 웹 서버에 SSL 이 사용되도록 설정된 경우에는 httpd-ssl.conf 파일에 대해서도 동일한 업데이트를 수행하십시오. 이 파일은 *sps_home*\secure-proxy\httpd\conf\extra 디렉터리에 있습니다. 업데이트는 운영 환경이 IPv4 를 기반으로 하는지 IPv6 을 기반으로 하는지에 따라 다릅니다.

**httpd.conf 파일과 선택적으로 httpd-ssl.conf 파일을 업데이트하여 여러 가상 호스트를 처리하려면**

- IPv4 환경의 경우 다음 LISTEN 지시문을 추가합니다.

  LISTEN 127.0.0.1:<*_port*>

- IPv6 환경의 경우 다음 LISTEN 지시문을 추가합니다.

  LISTEN [::1]:<*_port*>

- IPv4 및 IPv6 을 지원하는 이중 스택 환경의 경우 다음 LISTEN 지시문을 추가합니다.

  LISTEN 127.0.0.1:<*_port*>

  LISTEN [::1]:<*_port*>

또한 새 호스트 이름이 추가되도록 다음과 같이 호스트 파일의 루프백 주소 항목을 업데이트하십시오.

- IPV4: 127.0.0.1
- IPV6: [::1]

Windows 의 경우 호스트 파일은 대개 C:\WINDOWS\system32\drivers\hosts 에 있습니다. UNIX 의 경우 호스트 파일은 대개 /etc/hosts 에 있습니다.

## 여러 가상 호스트에 대한 세션 체계 매핑 구현

여러 사용자 에이전트 유형을 인식하여 가상 호스트를 기준으로 각 사용자 에이전트에 대해 서로 다른 세션 체계 매핑을 할당하도록 CA SiteMinder for Secure Proxy Server 를 구성하려면 다음 단계를 수행해야 합니다.

1. 세션 체계를 구성하거나, CA SiteMinder for Secure Proxy Server 와 함께 제공된 체계의 구성을 확인합니다.
2. server.conf 파일에서 사용자 에이전트 유형을 정의합니다.
3. server.conf 파일에서 각 가상 호스트에 대해 기본 설정과는 다른 지시문을 정의하는 섹션을 생성합니다(가상 호스트에 대한 기본값 재정의).
4. 각 가상 호스트에 대한 세션 체계 매핑을 정의합니다.

다음은 server.conf 파일의 일부분으로, IE(Internet Explorer) 브라우저 사용자에 대해 사용자 에이전트 유형이 정의된 예를 제공합니다. IE 사용자는 가상 호스트에 대해 정의된 기본 세션 체계가 아닌 다른 세션 체계를 사용하도록 매핑됩니다. 다음 예에서는 server.conf 파일에 정의된 세션 체계를 보여 줍니다.

```
#Session Schemes
<SessionScheme name="default">
        class="com.netegrity.proxy.session.SessionCookieScheme"
        accepts_smsession_cookies="true"
</SessionScheme>
<SessionScheme name="ssl_id">
        class="com.netegrity.proxy.session.SSLIdSessionScheme"
        accepts_smsession_cookies="false"
</SessionScheme>
<SessionScheme name="simple_url">
        class="com.netegrity.proxy.session.SimpleURLSessionScheme"
        accepts_smsession_cookies="false"
</SessionScheme>
<SessionScheme name="minicookie">
        class="com.netegrity.proxy.session.MiniCookieSessionScheme"
        accepts_smsession_cookies="false"
        cookie_name="MiniMe"
</SessionScheme>
```

다음 예에서는 IE 사용자 에이전트 유형의 정의를 보여 줍니다. 이 사용자 에이전트 유형은 나중에 server.conf 파일에서 세션 체계 매핑을 정의할 때 참조됩니다.

```
# TO-DO: Define Any User Agents, if you want to
# use a different session scheme based on
# the type of client accessing the server.
#
# NOTE:  UserAgent matching is done in the order
# in which the user agents are defined in this file.
 <UserAgent name="IE">
     User-Agent="MSIE"
 </UserAgent>
# <UserAgent name="NS">
#     User-Agent=some other regular expression
# </UserAgent>
```

앞의 예에서는 defaultsessionscheme 지시문에 지정된 기본 세션 체계가 미니 쿠키임을 보여 줍니다. 이 세션 체계는 세션 체계 매핑에 다른 세션 체계가 명시적으로 포함되었거나 다른 체계가 가상 호스트 정의의 기본 세션 체계를 재정의하지 않는 한 모든 트랜잭션에 사용됩니다.

<VirtualHostDefaults> 지시문은 <UserAgent name="IE">에 정의된 IE 사용자 에이전트 유형에 대한 세션 체계 매핑을 보여 줍니다. 이 매핑은 기본 세션 체계 매핑을 사용하는 모든 가상 호스트에 대해 IE 브라우저 사용자의 세션이 단순 URL 다시 쓰기 세션 체계를 사용하여 유지 관리됨을 나타냅니다.

```
<VirtualHostDefaults>
        # Service Dispatcher
        <ServiceDispatcher>
                class="com.netegrity.proxy.service.SmProxyRules"
                rules_file="conf\proxyrules.xml"
        </ServiceDispatcher>
        # default session scheme
        defaultsessionscheme="minicookie"
        #TO-DO:  Define any session scheme mappings
        <SessionSchemeMappings>
       #    user_agent_name=session_scheme_name
           IE="simple_url"
       #    NS=simple_url
        </SessionSchemeMappings>
```

가상 호스트 지시문은 CA SiteMinder for Secure Proxy Server 에 대해 구성된 기본 가상 호스트의 서버 이름과 IP 주소를 보여 줍니다.

```
# Default Virtual Host
<VirtualHost name="default">
        hostnames="server1, server1.company.com"
        addresses="192.168.1.10"

        #The defaults can be overriden
        #not only for the Virtual Host
        #but for the WebAgent for that
        #virtual host as well
        #<WebAgent>
        #</WebAgent>

</VirtualHost>
```

추가 가상 호스트에 대한 가상 호스트 지시문은 server2 가상 호스트에 대해 재정의되는 특정 기본 가상 호스트 설정을 보여 줍니다. 이러한 재정의에는 새 세션 체계 매핑이 포함되어 있습니다. server2 의 기본 체계는 "default"입니다. 세션 체계 지시문에서 "default"는 기존 SiteMinder 쿠키 세션 체계로 정의되어 있습니다. 또한 가상 호스트 지시문에서 IE 사용자에 대한 세션 체계 매핑은 "default" 체계에도 매핑됩니다. 따라서 CA SiteMinder for Secure Proxy Server 는 SiteMinder 쿠키 세션 체계를 사용하여 server2 에 액세스하는 모든 사용자에 대한 세션을 유지 관리합니다.

```
# Additional Virtual Host
<VirtualHost name="host2">
       requestblocksize="4"
        responseblocksize="4"
        hostnames="server2, server2.company.com"
        #addresses="192.168.1.15"
        # default session scheme
        defaultsessionscheme="default"

        #TO-DO:  Define any session scheme mappings
        <SessionSchemeMappings>
       #user_agent_name=session_scheme_name
           IE="default"
        </SessionSchemeMappings>

        #<WebAgent>
        #</WebAgent>
</VirtualHost>
```

# 제 11 장: 웹 서비스 구성

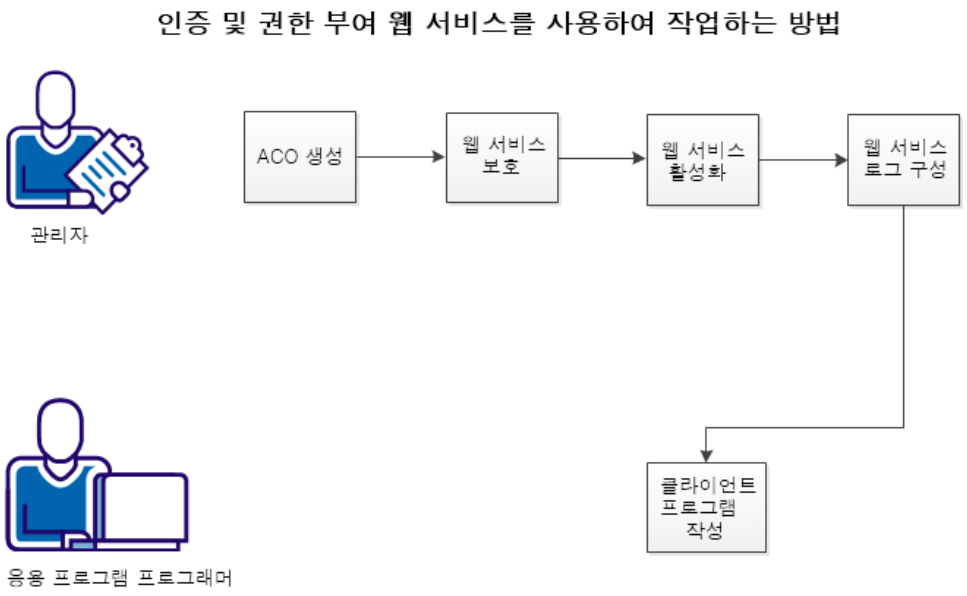
이 섹션은 다음 항목을 포함하고 있습니다.



## 인증 및 권한 부여

### 인증 및 권한 부여 웹 서비스로 작업하는 방법

CA SiteMinder®에서는 현재 인증 웹 서비스와 권한 부여 웹 서비스를 제공하고 있습니다. CA SiteMinder® 인증 및 권한 부여 웹 서비스를 사용하는 프로세스에는 다음 다이어그램의 절차가 포함됩니다.



인증 및 권한 부여 웹 서비스에 대한 작업을 하려면 다음 단계를 수행하십시오.

1. ACO 를 생성합니다 (페이지 210).
2. 웹 서비스를 보호합니다 (페이지 211).
3. 웹 서비스가 사용되도록 설정합니다 (페이지 211).
4. 웹 서비스 로그를 구성합니다 (페이지 212).
5. 클라이언트 프로그램을 생성합니다 (페이지 213).

## 인증 및 권한 부여 웹 서비스 개요

CA SiteMinder 인증 및 권한 부여 웹 서비스는 CA SiteMinder for Secure Proxy Server 설치에 포함됩니다. 각 구성 요소의 사용 여부를 개별적으로 설정할 수 있습니다.

웹 서비스 구성 프로세스에서는 다음과 같은 CA SiteMinder 개체가 구성되어 있다고 가정합니다.

- 호출자가 인증하는 대상 응용 프로그램을 보호하기 위한 에이전트 하나 이상
- 인증 및 권한 부여에 필요한 영역, 사용자 디렉터리, 정책 및 응답

인증 및 권한 부여 웹 서비스를 사용하면 다른 방법으로 보호되지 않는 응용 프로그램을 지원할 수 있습니다. 예를 들어 휴대폰의 독립형 응용 프로그램은 적절한 CA SiteMinder 개체가 있으면 사용자를 인증할 수 있습니다.

이러한 웹 서비스는 SOAP 1.2 프로토콜과 HTTP 기반 RESTful 아키텍처를 지원합니다. 인증 및 권한 부여 웹 서비스에서는 다음과 같은 기능을 제공합니다.

- 로그인(login) - 인증을 시도하고 인증이 성공하면 세션 토큰을 반환합니다.

  **참고**: 사용자 추적 사용 옵션이 사용되도록 설정되면 응답에 아이덴티티 토큰도 포함됩니다.
- 부울 로그인(blogin) - 인증을 시도하고 로그인 성공 여부를 확인합니다. 세션 토큰은 반환하지 않습니다.

- 로그아웃(logout) - 사용자 또는 사용자 그룹을 로그아웃합니다.
- 권한 부여(authorize) - 권한 부여 상태 메시지와 새로 고친 세션 토큰을 반환합니다.

작업 요청에 대한 응답은 SiteMinder 에서 생성한 헤더에 따라 다릅니다. 리소스가 익명 인증 체계로 보호되는 경우 응답에 세션 토큰이 포함되지 않고 아이덴티티 토큰이 포함됩니다. 이 아이덴티티 토큰을 이후 권한 부여 요청에 세션 토큰 대신 사용할 수 있습니다.

인증 요청에는 다음 매개 변수가 포함됩니다.

- 응용 프로그램 ID(appId)
- 리소스 문자열(resource)
- 작업(action)
- 사용자 자격 증명

응용 프로그램 ID 는 CA SiteMinder 응용 프로그램 개체가 아니라 리소스 계층의 위치에 대해 사용자가 정의한 논리적 이름을 나타냅니다. 응용 프로그램 ID 는 내부적으로 에이전트에 매핑됩니다. CA SiteMinder 는209 에이전트 이름을 사용하여 영역을 확인합니다. 영역, 리소스 문자열 및 사용자 자격 증명으로 사용자를 인증할 수 있습니다.

권한 부여 요청은 인증 요청보다 단순합니다. 권한 부여 요청에는 로그인 응답에서 가져온 응용 프로그램 ID(appId), 리소스 경로, 작업 및 세션 토큰이 포함됩니다. 웹 서비스는 토큰의 유효성을 검사한 후 지정된 리소스에 대한 액세스 권한을 부여할지 여부를 결정합니다.

## 웹 서비스 구성

기본적으로 CA SiteMinder for Secure Proxy Server 12.51 을 설치하거나 SPS 12.51 로 업그레이드하면 웹 서비스 기능이 설치됩니다.

웹 서비스를 구성하려면 다음 단계를 수행하십시오.

1. WAMUI 를 통해 웹 서비스에 대한 ACO 를 생성합니다.
2. 웹 서비스를 보호합니다.
3. 관리 UI 를 통해 웹 서비스가 사용되도록 설정합니다.
4. (선택 사항) 웹 서비스 로그를 구성합니다.

## 웹 서비스에 대한 ACO 생성

ACO 를 통해 웹 서비스를 관리할 수 있습니다. ACO 는 또한 리소스 액세스 보호를 위해서도 사용되며, AgentName 에 반드시 정의되어야 합니다. 웹 서비스를 사용하려면 enableauth 매개 변수나 enableaz 매개 변수, 또는 둘 모두를 활성화해야 합니다.

**다음 단계를 수행하십시오.**

1. WAMUI 에서 AuthAzServiceDefaultSettings 템플릿을 기반으로 하는 ACO 를 생성합니다.
2. 웹 서비스를 서비스로서 사용하도록 다음 매개 변수를 구성합니다.

**AgentName**

리소스를 보호하는 웹 에이전트의 이름, defaultagentname *또는* 웹 서비스를 보호하는 ACO 의 에이전트 이름을 정의합니다. 이러한 값을 AgentName 에 추가해야 합니다.

응용 프로그램을 보호하는 여러 웹 에이전트를 정의하려면 다음 형식으로 여러 값 쌍을 입력하십시오.

```
agent_name1,appID1
agent_name2,appID2
agent_namen,appIDn
```

***agent_name***

리소스를 보호하는 웹 에이전트의 이름을 정의합니다.

***appID***

agent_name에 지정된 웹 에이전트의 참조 이름 또는 웹 에이전트에 의해 보호되는 응용 프로그램의 참조 이름을 정의합니다. CA SiteMinder®는 웹 서비스 요청에서 이 값을 사용하므로 사용자로부터 에이전트 이름을 보호합니다.

지정된 AgentName 에 defaultagentname 을 추가하거나 *또는* 웹 서비스를 보호하는 ACO 의 에이전트 이름을 추가하십시오.

웹 서비스를 보호하는 ACO 의 에이전트 이름을 사용하려면 다음 형식으로 에이전트 이름을 정의하십시오.

```
agent_name,hostname
```

웹 서비스를 보호하는 ACO 의 defaultagentname 을 사용하려면 다음 형식으로 에이전트 이름을 정의하십시오.

```
agent_name
```

**enableauth**

인증 웹 서비스의 상태를 지정합니다. 인증 웹 서비스를 사용하려면 이 값을 yes 로 설정하십시오.

**enableaz**

권한 부여 웹 서비스의 상태를 지정합니다. 권한 부여 웹 서비스를 사용하려면 이 값을 yes 로 설정하십시오.

**RequireAgentEnforcement**

CA SiteMinder® 에이전트가 웹 서비스를 보호해야 하는지 여부를 지정합니다. 프로덕션 환경에서는 CA SiteMinder® 에이전트가 웹 서비스를 보호하도록 이 값을 yes 로 설정할 것을 강력히 권장합니다. 이 값을 yes 로 설정한 경우 웹 서비스가 보호되지 않으면 웹 서비스에 대한 요청이 실패합니다.

**참고**: RequireAgentEnforcement 의 값은 테스트 환경인 경우 또는 CA SiteMinder® 이외의 다른 메커니즘을 사용하여 웹 서비스를 보호하는 경우 'no'로 설정할 수 있습니다.

3. 변경 내용을 저장합니다.

## 웹 서비스 보호

프로덕션 환경에서는 웹 서비스를 보호할 것을 권장합니다. 웹 서비스의 웹 에이전트를 보호하면 사용자 요청이 처리되기 전에 CA SiteMinder 가211 웹 서비스 클라이언트를 인증 및 권한 부여할 수 있습니다. 프로덕션 환경에서 웹 서비스를 보호하면 CA SiteMinder for Secure Proxy Server 가 사용자 요청에 SMSESSION 쿠키를 포함합니다. RequestSmSessionCookie ACO 매개변수가 활성화된 경우 CA SiteMinder 는211 웹 서비스가 사용자 요청을 처리하기 전에 SMSESSION 쿠키에 대한 사용자 요청을 확인하도록 합니다.

웹 서비스를 보호하기 위해 X.509 클라이언트 인증서 인증 체계를 사용하여 웹 서비스 루트 URL 을 보호하도록 CA SiteMinder for Secure Proxy Server 를 구성할 것을 권장합니다.

## 웹 서비스가 사용되도록 설정

관리 UI 를 통해 이전 절차에서 생성한 ACO 를 사용하여 웹 서비스가 사용되도록 설정하십시오.

**참고**: enableauth 및 enableaz 의 값이 no 로 설정되어 있으면 CA SiteMinder for Secure Proxy Server 관리 UI 를 통해 해당 지원 기능이 사용되도록 설정해도 웹 서비스가 작동하지 않습니다.

**다음 단계를 수행하십시오.**

1. "프록시 구성", "인증 및 권한 부여 웹 서비스"로 이동합니다.
2. "호스트 이름"에 웹 서비스 가상 호스트의 고유 호스트 이름을 입력합니다.
3. "에이전트 구성 개체"에 웹 서비스에 대해 생성한 ACO 의 이름을 입력합니다.
4. "저장"을 클릭합니다.

   웹 서비스가 사용되도록 설정됩니다.

## 웹 서비스 로그 구성

웹 서비스가 사용되도록 설정하면 CA SiteMinder for Secure Proxy Server 에서 해당 웹 서비스의 로그를 server.log 파일에 저장합니다. 로그 위치를 server.log 에서 authazws.log 로 변경할 수도 있습니다.

로그 위치를 변경하려면 다음 단계를 수행하십시오.

1. *sps_home*/proxy-engine/conf/webservicesagent 로 이동합니다.
2. authaz-log4j.xml 파일을 백업합니다.
3. 원본 authaz-log4j.xml 파일을 열고 다음 단계를 수행합니다.
   a. 다음 AuthAZ_ROLLING appender 태그의 주석 처리를 제거합니다.

```
<appender name="AuthAZ_ROLLING"
class="org.apache.log4j.DailyRollingFileAppender">
<param name="File" value="logs/authazws.log"/>
        <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern" value="%d %-5p [%c] - %m%n"/>
</layout>
</appender>
```

   b. AuthAZ_ROLLING 태그에 대한 다음 appender-ref 를 모두 찾아 해당 주석 처리를 제거합니다.

```
<appender-ref ref="AuthAZ_ROLLING"/>
```

4. 변경 내용을 저장하고 CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

   로그 위치가 *sps_home*/proxy-engine/logs/에 있는 authazws.log 파일로 변경됩니다.

# 클라이언트 프로그램 생성

클라이언트 프로그램은 다른 응용 프로그램 대신 웹 서비스에 대한 인증 및 권한 부여 요청을 발행하는 역할을 합니다. 클라이언트 프로그램에는 클라이언트 스텁에 대한 코드가 필요합니다. 스텁은 웹 서비스와의 통신을 위해 메시지를 관리하고 서로 주고받습니다. 웹 서비스는 WSDL 파일(SOAP 프로토콜의 경우)과 WADL 파일(REST 아키텍처의 경우)을 지원합니다. 웹 브라우저를 사용하여 WSDL 또는 WADL 파일에 액세스한 후 이를 XML 파일로 저장할 수 있습니다.

**다음 단계를 수행하십시오.**

1. 필요한 자격 증명을 수집하는 응용 프로그램용 비즈니스 논리를 작성합니다.
2. 클라이언트 스텁을 생성합니다. 필요한 경우 타사 도구와 함께 WSDL 또는 WADL 파일을 사용하여 클라이언트 스텁을 생성할 수 있습니다.
   - WSDL 을 로드하려면 다음 URL 을 사용하십시오:

     `http://hostname:port/authazws/auth?wsdl`
   - WADL 을 로드하려면 다음 URL 을 사용하십시오:

     `http://hostname:port/authazws/AuthRestService/application.wadl`

   참고: 이러한 위치에서 메타데이터를 가져오려면 ACO 의 DefaultAgentName 매개 변수를 에이전트 중 하나로 설정하십시오.
3. 클라이언트 스텁을 가져오고, 웹 서비스를 호출하는 스텁 개체를 인스턴스화합니다.

다음 단원에서는 참조를 위해 단순화된 샘플 SOAP 및 REST 메시지를 보여 줍니다.

## 인증 SOAP 인터페이스

이 단순화된 샘플에서는 SOAP 프로토콜을 사용한 인증 작업을 보여 줍니다. username, password 및 binaryCredentials 라는 세 개의 필드만으로 구성된 IdentityContext 로 대부분의 인증 체계를 지원할 수 있습니다. 다른 필드가 더 필요한 체계의 경우 자격 증명 유형에 맞게 입력이 조정된 추가 작업에서 지원됩니다.

다음 예제는 인증 웹 서비스의 일반적인 사용자 로그인 요청입니다.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
            xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
  <s:Header/>
  <s:Body>
    <aut:login>
      <identityContext>
        <binaryCreds>
        </binaryCreds>
        <password>user1</password>
        <userName>user1</userName>
      </identityContext>
      <appId>app1</appId >
      <action>GET</action>
      <resource>/*</resource >
    </aut:login>
  </s:Body>
</s:Envelope>
```

부울 로그인(blogin) 작업은 로그인(login) 작업과 유사하지만 다음 예제와 같이 blogin 은 응답에 SMSESSION 값을 반환하지 않는다는 차이점이 있습니다.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
            xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
  <s:Header/>
  <s:Body>
    <aut:blogin>
      <identityContext>
        <binaryCreds>
        </binaryCreds>
        <password>user1</password>
        <userName>user1</userName>
      </identityContext>
      <appId>app1</appId >
      <action>GET</action>
      <resource>/*</resource >
    </aut:blogin>
  </s:Body>
</s:Envelope>
```

다음 예제에서는 성공적인 로그인 응답을 나타냅니다.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header/>
  <s:Body>
    <aut:loginResponse
xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
      <return>
        <message>Authentication successful.</message>
        <resultCode>LOGIN_SUCCESS</resultCode>
        <sessionToken>session</sessionToken>
    <responses>
      <response/>
      <response/>
    </responses>
      </return>
    </aut:loginResponse>
  </s:Body>
</s:Envelope>
```

다음 예제에서는 실패한 로그인 시도를 나타냅니다.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header/>
  <s:Body>
    <ns2:loginResponse xmlns:ns2="http://webservice.sm.services.soa.ca.com/">
      <return>
        <message>Authentication failured</message>
        <resultCode>LOGIN_FAILED</resultCode>
        <smSessionCookieValue/>
      </return>
    </ns2:loginResponse>
  </s:Body>
</s:Envelope>
```

다음 예제에서는 인증 웹 서비스 사용자 로그아웃 요청을 나타냅니다.

**참고**: 사용자가 성공적으로 로그아웃했더라도 해당 SessionToken 은 유효한 사용자 자격 증명으로 간주되므로 에이전트가 권한 부여에 이 SessionToken 을 계속 사용할 수 있습니다.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
            xmlns:aut="http://ca.com/2010/04/15/authentication.xsd">
  <s:Header/>
  <s:Body>
    <aut:logout>
      <smSessionCookieValue>session</smSessionCookieValue>
    </aut:logout>
  </s:Body>
</s:Envelope>
```

다음 예제에서는 성공적인 인증 웹 서비스 로그아웃 응답을 나타냅니다.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header/>
  <s:Body>
    <ns2:logoutResponse
xmlns:ns2="http://ca.com/2010/04/15/authentication.xsd">
      <return>
        <message>Logout successful.</message>
        <resultCode>SUCCESS</resultCode>
      </return>
    </ns2:logoutResponse>
  </s:Body>
</s:Envelope>
```

## 인증 REST 인터페이스

REST 는 REpresentational State Transfer 를 의미합니다. REST 에서는 서비스 요청이 개체의 상태를 URI 를 통해 액세스할 수 있도록 변환합니다. HTTP 의 경우 생성, 읽기, 업데이트 및 삭제 같은 작업을 통해 상태가 변경됩니다.

인증 및 권한 부여를 위한 URI 매핑은 appId 와 resourcePath 로 구성됩니다. 리소스 상태란 인증 또는 권한 부여된 사용자와 리소스를 연결한 것을 총칭합니다. 인증에 사용되는 서비스 이름은 login, blogin 및 logout 입니다.

http://hostname:port/authazws/AuthRestService/login/*appID*/*Resource* 형식의 URI 는 다음과 같은 요청을 게시합니다.

```
<loginRequest>
      <binaryCreds></binaryCreds>
       <password>user1</password>
       <userName>user1</userName>
       <action>GET</action>
</loginRequest>
```

로그인 응답:

HTTP 반환 코드 200

```
<loginResponse>
<message>Authentication successful</message>
<resultCode>LOGIN_SUCCESS</resultCode>
<sessionToken>session</sessionToken>
<authenticationResponses>
    <response>
            <name>SM_SESSIONDRIFT</name>
            <value>0</value>
    </response>
</authenticationResponses>
</loginResponse>
```

HTTP 반환 코드 400

```
<loginResponse>
<message>Bad Request</message>
<resultCode>LOGIN_ERROR</resultCode>
</loginResponse>
```

HTTP 반환 코드 200

```
<loginResponse>
<message>Authentication Failed</message>
<resultCode>LOGIN_FAILED</resultCode>
<authenticationResponses>
    <response><name>SM_AUTHREASON</name>
    <value>0</value>
    </response>
</authenticationResponses>
</loginResponse>
```

HTTP 반환 코드 500

```
<loginResponse>
<message>System</message>
<resultCode>Server Error</resultCode>
</loginResponse>
```

부울 로그인(blogin) 작업은 로그인(login)과 유사합니다. http://host:port#/blogin/appId/resourcePath 형식의 URI 는 로그인 요청에 표시된 것과 같이 포스트하며, 응답 메시지에서 yes 또는 no 를 반환합니다.

http://host:port#/authazws/AuthRestService/logout/ 형식의 URI 는 다음과 같은 로그아웃 요청을 포스트합니다.

```
<logoutRequest>
<sessionToken>session</sessionToken>
</logoutRequest>
```

인증 웹 서비스 로그아웃 응답:

```
<logoutResponse>
<message>로그아웃 성공</message>
<resultCode>LOGOUT_SUCCESS</resultCode>
<smSessionCookieValue>yyy</smessionCookieValue>
</logoutResponse>

<logoutResponse>
<message>로그아웃 실패</message>
<resultCode>LOGOUT_FAILURE</resultCode>
<smSessionCookieValue>yyy</smessionCookieValue>
</logoutResponse>
```

## 권한 부여 SOAP 서비스

다음 XML 은 웹 서비스에 대한 권한 부여 요청을 대략적으로 나타낸 것입니다.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:aut="http://ca.com/2010/04/15/authorization.xsd">
   <soapenv:Header/>
   <soapenv:Body>
      <aut:authorize>
    <sessionToken>session</sessionToken>
          <appId></appId>
          <action>GET,POST</action>
     <resource>/domainAdmin/a.jsp</resource>
      </aut:authorize>
   </soapenv:Body>
</soapenv:Envelope>
```

다음 예제에서는 권한 부여 웹 서비스 AUTHORIZED 응답을 나타냅니다.

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <env:Body>
      <ns2:authorizeResponse
xmlns:ns2="http://ca.com/2010/04/15/authorization.xsd">
         <return>
            <message>Authorization Successful</message>
            <resultCode>AUTHORIZED</resultCode>
    <sessionToken>aklaks</sessionToken>
    <authorizationResponses>
    <response/>
    </authorizationResponses>
         </return>
      </ns2:authorizeResponse>
   </env:Body>
</env:Envelope>
```

다음 예제에서는 권한 부여 웹 서비스 UN AUTORIZED 응답을 나타냅니다.

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <env:Body>
      <ns2:authorizeResponse
xmlns:ns2="http://ca.com/2010/04/15/authorization.xsd">
         <return>
            <message> Authorization Failed</message>
            <resultCode>NOTAUTHORIZED</resultCode>
         </return>
      </ns2:authorizeResponse>
   </env:Body>
</env:Envelope>
```

**참고**: 유효한 세션 토큰을 포함한 권한 부여 웹 서비스 요청의 경우 NOTAUTHORIZED 권한 부여 응답에는 다음과 같은 제약 조건이 있습니다.

1. WAMUI 에서 다음 특성으로만 응답을 구성할 수 있습니다.
   - SM_ONREJECTTEXT
   - SMREDIRECTURL 또는 SM_REDIRECTURL
   - SMERROR
2. 응답에는 세션 토큰이 포함되지 않습니다.

### 권한 부여 REST 인터페이스

권한 부여를 위한 REST 인터페이스는 http://hostname:port/authazws/AuthRestService/authz/*appID*/*Resource* 입니다.

```
<authorizationRequest>
<action>POST</action>
<resource>RealmA/index.html</resource>
<sessionToken>affl;;alkf;l;fd</sessionToken>
</authorizationRequest>
```

HTTP 반환 코드 200:

```
<authorizationResult >
<message>The user is authorized.</message>
<resultCode>AUTHORIZED</resultCode>
</authorizationResult >
```

# 보안 토큰 서비스

CA SiteMinder for Secure Proxy Server 는 토큰 발급 및 트랜잭션에 WS-Trust 기반 메커니즘을 제공할 수 있도록 Office 365 용 STS(보안 토큰 서비스)를 지원합니다. 하나의 CA SiteMinder for Secure Proxy Server 컴퓨터에 하나 또는 여러 개의 STS 인스턴스를 배포할 수 있습니다.

## 여러 CA SiteMinder for Secure Proxy Server 인스턴스 배포

여러 STS 인스턴스를 배포하려면 각 STS 인스턴스가 개별 로그 파일에 로깅하도록 모든 STS 인스턴스가 동일한 log4j 구성을 사용해야 합니다.

**다음 단계를 수행하십시오.**

1. 다음 작업 중 *하나*를 수행합니다.
   - Windows 에서 다음 단계를 수행하십시오.
     a. *installation_home*/proxy-engine/conf 로 이동합니다.
     b. SmSpsProxyEngine.properties 파일을 열고 파일에서 STS_AGENT_LOG_CONFIG_FILE 변수의 주석 처리를 해제합니다.
     c. 변경 내용을 저장합니다.
   - UNIX 에서 다음 단계를 수행하십시오.
     a. *installation_home/proxy-engine* 으로 이동합니다.
     b. proxyserver.sh 파일을 열고 파일에서 STS_AGENT_LOG_CONFIG_FILE 변수의 주석 처리를 해제합니다.
     c. 변경 내용을 저장합니다.
2. installation_home/proxy-engine/conf/sts-config/globalconfig 로 이동합니다.
3. agent-multiinstance-log4j.xml 파일을 엽니다.
4. 각 STS 인스턴스에 대해 다음 단계를 수행합니다.
   a. STS 인스턴스에 대한 어펜더를 만듭니다.

      **참고**: 기본적으로 이 파일은 하나의 STS 인스턴스에 대한 하나의 어펜더를 포함하고 있습니다.
   b. 어펜더에서 [SPS ROOT FOLDER]를 CA SiteMinder for Secure Proxy Server 루트 폴더로 변경합니다.
   c. 어펜더에서 [STS Service Name]을 STS 인스턴스의 서비스 이름을 변경합니다.
5. 변경 내용을 저장합니다.
6. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

   각 STS 인스턴스의 로그 파일이 다음 형식으로 *installation_home*/proxy-engine/logs 에 만들어집니다.

   *STS_service_name*.log

7. 각 STS 인스턴스가 개별 로그 파일에 로깅하는지 확인합니다.

# 제 12 장: SiteMinder 와 SPS 통합

이 섹션은 다음 항목을 포함하고 있습니다.



## SPS 가 SiteMinder 와 상호 작용하는 방식

SiteMinder 는 e-비즈니스를 안전하게 관리하기 위한 솔루션입니다. SiteMinder 는 엔터프라이즈의 정책을 지정할 수 있는 정책 서버와 웹 서버에 설치되는 웹 에이전트로 구성되어 있습니다. 웹 에이전트는 정책 서버와 통신하여 인증, 권한 부여 및 기타 기능을 제공합니다.

CA SiteMinder for Secure Proxy Server 에는 SiteMinder 웹 에이전트 및 정책 서버 기술과 호환되는 웹 에이전트가 포함되어 있습니다. 모든 SiteMinder 웹 에이전트와 마찬가지로 CA SiteMinder for Secure Proxy Server 도 SiteMinder 의 개체로 구성해야 합니다. 또한 대상 서버에 액세스하기 위한 인증 및 권한 부여 요구 사항을 결정하는 정책을 생성해야 합니다.

SiteMinder 개체는 SiteMinder <adminui>를 사용하여 구성됩니다. 다음 개체를 구성할 수 있습니다.

**에이전트**

SPS 에 포함된 웹 에이전트에 대한 설정을 사용하여 에이전트 개체를 구성하십시오. 영역을 생성할 때 이 웹 에이전트를 지정하십시오.

**사용자 디렉터리**

사용자를 인증하고 권한을 부여하는 사용자 디렉터리에 대한 연결을 구성하십시오.

**정책 도메인**

영역, 규칙 및 정책을 포함하는 정책 도메인을 구성하십시오.

**영역**

SiteMinder 로 보호할 리소스를 포함하는 영역을 구성하십시오.

**규칙**

SiteMinder 로 보호할 특정 리소스 및 작업을 식별하는 규칙을 구성하십시오.

**응답**

응용 프로그램 또는 SPS 에 정보를 반환할 수 있는 응답을 구성하십시오. CA SiteMinder for Secure Proxy Server 에 반환되는 정보는 사용자 요청의 라우팅 방식을 결정할 수 있습니다.

**정책**

사용자 및 그룹을 규칙 및 응답에 바인딩하는 정책을 구성하십시오.

**참고:** SiteMinder 개체를 구성하는 방법에 대한 자세한 내용은 *CA SiteMinder 정책 서버 구성 안내서*를 참조하십시오.

## 인증 체계 고려 사항

SiteMinder 는 리소스를 보호하기 위해 인증 체계를 적용합니다. 사용자가 SiteMinder 웹 에이전트나 SPS 를 통해 보호된 리소스에 액세스하려고 하면 SiteMinder 는 해당 리소스를 보호하는 인증 체계에 따라 자격 증명을 요구합니다.

또한 SiteMinder 는 각 인증 체계에 보호 수준을 제공합니다. 보호 수준은 사용자가 다른 인증 체계로 보호되는 리소스에 액세스하려고 할 때 싱글 사인온 중에 적용됩니다. 이러한 경우 각 인증 체계에 대한 보호 수준이 동일하거나 더 낮으면 사용자는 재인증 없이도 다른 인증 체계로 보호되는 리소스에 액세스할 수 있습니다. 낮은 보호 수준에서 높은 보호 수준으로 이동할 때는 사용자에게 인증이 요청됩니다. 높은 보호 수준에서 낮은 보호 수준으로 이동할 때는 사용자에게 재인증이 요청되지 않습니다.

CA SiteMinder for Secure Proxy Server 가 SiteMinder 와 통합된 경우 CA SiteMinder for Secure Proxy Server 는 SiteMinder 웹 에이전트와 유사하게 작동합니다. 하지만 기본 인증을 사용하는 CA SiteMinder for Secure Proxy Server 는 CA SiteMinder for Secure Proxy Server 가 기본 SessionCookieScheme 체계를 사용하여 사용자 세션을 추적하도록 구성된 경우에만 웹 에이전트와 유사하게 작동합니다. CA SiteMinder for Secure Proxy Server 가 다른 고급 세션 체계나 쿠키를 사용하지 않는 세션 체계를 사용하도록 구성되어 있으면 사용자가 다시 인증해야 합니다. 싱글 사인온은 작동하지 않습니다.

예를 들어 보호 수준이 5 인 기본 인증 체계가 resource1 및 resource2 라는 두 리소스를 보호하며 CA SiteMinder for Secure Proxy Server 가 미니 쿠키 세션 체계 또는 쿠키를 사용하지 않는 다른 세션 체계를 사용하여 사용자 세션을 추적하도록 구성되어 있는 경우, 사용자가 resource1 에 액세스하려고 하면 CA SiteMinder for Secure Proxy Server 는 요청을 SiteMinder 로 전달합니다. SiteMinder 는 resource1 에 대한 인증 체계를 확인하고 사용자에게 자격 증명을 요청합니다.

CA SiteMinder for Secure Proxy Server 는 사용자로부터 자격 증명을 수집하고, SiteMinder 에 의해 인증에 성공하면 해당 사용자가 resource1 에 액세스할 수 있도록 합니다. 그런 다음 사용자가 resource2 에 액세스하려고 하면 CA SiteMinder for Secure Proxy Server 는 요청을 SiteMinder 로 전달합니다. SiteMinder 는 resource2 에 대한 인증 체계를 확인하고 사용자에게 자격 증명을 요청합니다. 이때 CA SiteMinder for Secure Proxy Server 는 미니 쿠키 세션 체계를 사용하도록 구성되어 있으므로 사용자에게 다시 인증할 것을 요청합니다. CA SiteMinder for Secure Proxy Server 가 기본 SiteMinder 쿠키 세션 체계를 사용하도록 구성되어 있으면 사용자는 다시 인증할 필요 없이 resource2 에 액세스할 수 있습니다.

**참고:** 인증 체계 및 해당 보호 수준에 대한 자세한 내용은 *CA SiteMinder Policy Configuration Guide*(CA SiteMinder 정책 구성 안내서)를 참조하십시오.

## 프록시 관련 WebAgent.conf 설정

엔터프라이즈의 DMZ 뒤에 설치된 웹 에이전트에 대한 WebAgent.conf 구성 파일에는 SPS 에 특정 영향을 주는 여러 설정이 있습니다.

대상 서버의 WebAgent.conf 파일에서 수정해야 하는 설정은 다음과 같습니다.

**proxytrust**

최적화 방법으로 SPS 뒤에 있는 대상 서버 웹 에이전트에 대해 proxytrust 지시문을 설정할 수 있습니다. 다음 설정 중 하나를 입력하십시오.

**yes**

대상 서버 웹 에이전트가 SPS 에 의한 권한 부여를 자동으로 트러스트합니다.

**no**

대상 서버 웹 에이전트가 항상 인증을 요청합니다. 기본값입니다.

**proxytimeout**

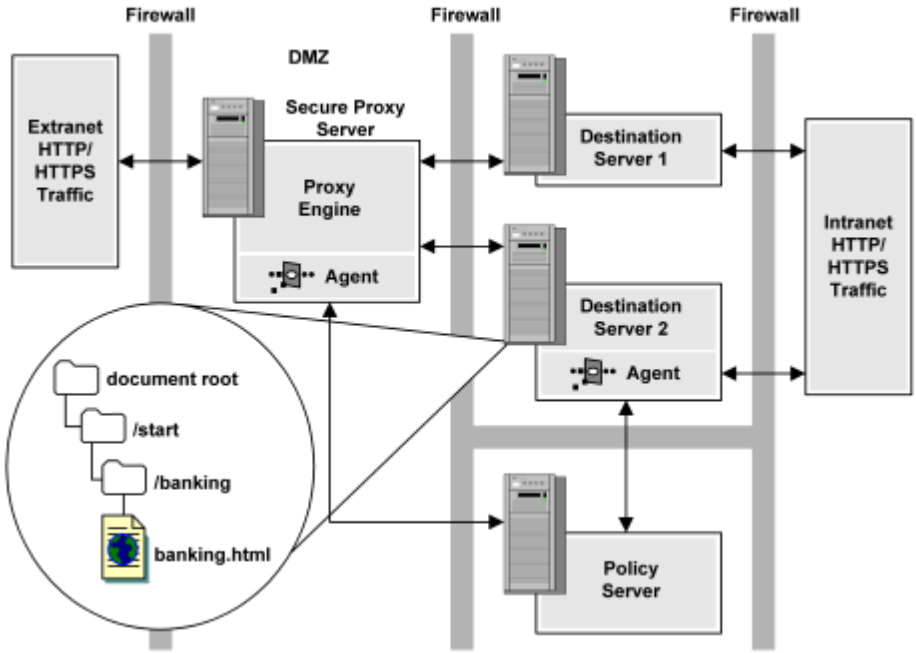
대상 서버의 웹 에이전트가 SPS 의 요청에 사용된 싱글 사인온 토큰을 일정 시간 후 만료시키도록 합니다. 값을 초 단위로 입력하십시오.

**기본값:** 120 초

## 대상 서버 웹 에이전트와의 정책 충돌 방지

일부 배포 환경에서 CA SiteMinder for Secure Proxy Server 가 프록시 트러스트 모드에서 실행 중인 경우 CA SiteMinder for Secure Proxy Server 는 리소스를 한 사용자 집합으로부터 보호하고 대상 서버의 웹 에이전트는 동일한 리소스를 다른 사용자 집합으로부터 보호합니다.

다음 그림에서는 대상 서버 2 에 고유한 웹 에이전트가 있습니다. 엑스트라넷 사용자는 SPS 에서 인증되고 권한이 부여되는 반면에 인트라넷 사용자는 대상 서버의 웹 에이전트를 통해 인증되고 권한이 부여됩니다. 이 경우 포함된 CA SiteMinder for Secure Proxy Server 웹 에이전트와 대상 서버의 웹 에이전트에 대한 정책이 SiteMinder 정책 저장소에 있어야 합니다.

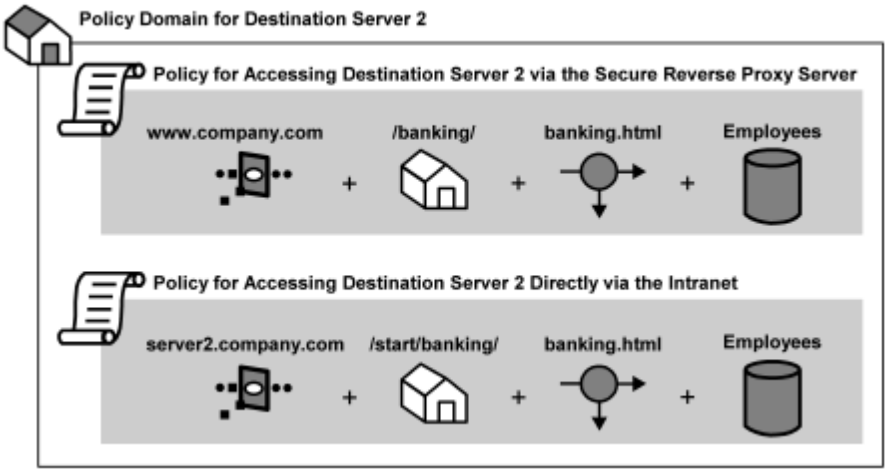



**참고:** 정책을 생성할 때 관리자는 정책이 서로 충돌하지 않도록 해야 합니다. 정책이 서로 충돌하면 SiteMinder 가 원하지 않거나 예기치 않은 동작을 허용하게 될 수 있습니다.

대상 서버 2 에 포함된 리소스에 대해 정책과 그 밖의 필요한 SiteMinder 개체를 올바르게 생성하려면 SiteMinder 에 다음 개체를 생성합니다.

- CA SiteMinder for Secure Proxy Server 웹 에이전트
- 대상 서버 2 웹 에이전트
- CA SiteMinder for Secure Proxy Server 웹 에이전트를 사용하는 영역
- 대상 서버 2 웹 에이전트를 사용하는 영역
- CA SiteMinder for Secure Proxy Server 웹 에이전트를 통해 액세스하는 리소스에 대한 규칙
- 대상 서버 2 웹 에이전트를 통해 액세스하는 리소스에 대한 규칙
- CA SiteMinder for Secure Proxy Server 웹 에이전트 리소스에 대한 정책
- 대상 서버 2 웹 에이전트 리소스에 대한 정책

다음 그림에서는 CA SiteMinder for Secure Proxy Server 와 웹 에이전트가 모두 포함된 환경에서 호환성 모드가 사용되는 경우 단일 리소스를 보호하려면 두 정책을 어떻게 생성해야 하는지 보여 줍니다.



위 그림에서처럼 동일한 리소스에 대한 규칙 및 영역이 대상 서버의 리소스 위치와 요청을 전달하는 데 사용되는 프록시 규칙에 따라 경로가 달라질 수 있습니다.

예를 들어 위 그림의 샘플 구성을 사용할 경우 banking.html 이라는 리소스가 대상 서버 2 의 server2.company.com/start/banking/ 디렉터리에 있지만 CA SiteMinder for Secure Proxy Server 에는 www.company.com/banking/banking.html 에 대한 모든 요청을 서버 2 의 동일한 대상으로 전달하는 프록시 규칙이 있을 수 있습니다. 따라서 동일한 리소스에 동일한 리소스를 나타내는 서로 다른 SiteMinder 규칙이 두 개 있을 수 있습니다. 한 규칙은 인트라넷에서 직원에 대해 직접 리소스 액세스 권한을 부여하고, 다른 규칙은 출장 중 엑스트라넷에서 동일한 리소스에 액세스하려고 하는 직원에게 권한을 부여합니다.

## 사용자를 리디렉션하는 SiteMinder 규칙 구성

SiteMinder 는 특정 상황에서 요청을 리디렉션하는 응답 개체를 생성하는 기능을 제공합니다. 예를 들어 인증 실패(OnRejectRedirect) 후 요청을 사용자 지정 오류 페이지로 리디렉션하는 응답을 생성할 수 있습니다. 기본적으로 쿠키를 사용하지 않으며 요청된 URL 을 다시 쓰는(단순 URL 다시 쓰기)하는 세션 체계의 경우 CA SiteMinder for Secure Proxy Server 는 리디렉션 후 사용자 세션 정보를 인식합니다.

리디렉션 후 사용자 세션을 종료하려면 SiteMinder 에서 SiteMinder SM_REWRITE_URL 헤더의 값을 수정하는 관련 정책에 대한 응답 특성을 생성하십시오. 리디렉션 후 강제로 새 세션을 적용하려면 이 HTTP 헤더를 NO 로 설정해야 합니다.

예를 들어 보호 수준이 5 인 인증 체계로 보호되는 영역 A 에 리소스가 있고 보호 수준이 10 인 인증 체계로 보호되는 영역 B 에 두 번째 리소스가 있는 경우, 영역 A 에서 성공적으로 인증된 사용자라도 보호 수준이 더 높은 영역 B 로 이동할 때는 자격 증명이 요청됩니다.

영역 B 에 OnRejectRedirect 응답이 연결되어 있는 경우 영역 B 에서 자격 증명이 요청될 때 사용자가 인증에 실패하면 CA SiteMinder for Secure Proxy Server 의 기본 동작에 따라 사용자의 원래 세션 정보가 유지됩니다. 이는 사용자가 사용자 지정 오류 페이지로 리디렉션된 후에도 해당됩니다.

리디렉션 후 사용자의 세션을 종료하고 다음 로그인 시도 시 완전히 새로운 세션을 강제로 적용하려면 SM_REWRITE_URL=NO 를 설정하는 응답 특성을 생성하고 이 응답을 적절한 정책과 연결해야 합니다.

# SPS 및 SharePoint 리소스

CA SiteMinder for Secure Proxy Server 를 사용하여 Microsoft SharePoint 에 의해 관리되는 리소스를 보호하려면 다음과 같이 구성을 변경하십시오.

- CA SiteMinder for Secure Proxy Server 에이전트 구성 개체에서 다음 매개 변수를 설정하십시오.
  - SPClientIntegration = *server_name:port*

    서버 이름은 httpd.conf 파일의 ServerName 필드에 설정된 값과 일치해야 합니다. 대부분의 경우 ServerName 은 정규화된 호스트 이름이지만 이 값이 IP 주소일 수도 있습니다.
  - ProxyAgent = Yes

  **참고**: 이러한 매개 변수는 CA SiteMinder for Secure Proxy Server LocalConfig.conf 파일에도 추가할 수 있습니다.
- CA SiteMinder for Secure Proxy Server WebAgent.conf 파일에서 SharePoint 플러그인(Windows 의 경우 SPPlugin.dll, Solaris 의 경우 libSPPlugin.so) 위치를 가리키는 LoadPlugin 매개 변수를 추가하십시오.
- server.conf 파일에서 다음 매개 변수를 사용하여 VirtualHost 요소를 추가하십시오.

```
<VirtualHost name="VHName">
hostnames="host_name, host_name"
enableredirectwrite="yes"
redirectwritablehostnames="server1.company.com, domain1.com"
</VirtualHost>
```

**참고**: 자세한 내용은 *CA SiteMinder Agent for SharePoint Guide*(CA SiteMinder Agent for SharePoint 안내서)를 참조하십시오.

# SPS 및 ERP 리소스

CA SiteMinder for Secure Proxy Server 를 사용하여 ERP 시스템에 의해 관리되는 리소스를 보호할 수 있습니다. CA SiteMinder for Secure Proxy Server 는 다음 ERP 시스템을 보호하는 ERP 에이전트 앞에서 프록시로 작동할 수 있습니다.

- Siebel 응용 프로그램 서버
- PeopleSoft 응용 프로그램 서버

- SAP AS 웹 응용 프로그램 서버
- SAP ITS 응용 프로그램 서버

ERP 에이전트는 ERP 서버에 설치되어야 하는 반면에 CA SiteMinder for Secure Proxy Server 는 정책 서버에서 ERP 리소스를 보호합니다.

**참고**: ERP 서버를 지원하는 데 필요한 정책 서버 설정에 대한 자세한 내용은 적절한 CA ERP 에이전트 안내서를 참조하십시오.

### CA SiteMinder for Secure Proxy Server 를 ERP 에이전트에 대한 리버스 프록시로 구성하려면

1. proxyRules.xml 파일에서 <_nete:forward> 요소에 ERP 서버 및 적절한 포트 번호를 지정합니다.
2. server.conf 파일에서 다음 값을 지정합니다.
   - enableredirectrewrite 매개 변수의 값을 "yes"로 설정합니다.
   - redirectrewritablehostnames 매개 변수의 값을 ERP 서버가 실행 중인 시스템의 호스트 이름으로 설정합니다. 예를 들면 다음과 같습니다.

```
<VirtualHost name="sales">
    hostnames="sales, sales.company.com"
    enableredirectrewrite="yes"
    redirectrewritablehostnames="server1.company.com,domain1.com"
</VirtualHost>
```

   - server.conf 의 <_Sever> 섹션에서 addquotestocookie 매개 변수의 값을 "no"로 설정합니다. 예를 들면 다음과 같습니다.

```
addquotestocookie="no"
```

   **참고**: ERP 서버 측의 필요한 설정에 대한 자세한 내용은 ERP 서버의 설명서를 참조하십시오.

CA SiteMinder for Secure Proxy Server 가 ERP 에이전트에 대한 프록시로 구성되었습니다.

# SPS 에 대한 암호 서비스

암호 서비스는 SiteMinder 관리자가 사용자 암호를 관리할 수 있도록 하여 보호된 리소스에 대한 보안 수준을 강화하는 SiteMinder 기능입니다. 관리자는 암호 서비스를 사용하여 암호 만료, 조합 및 사용에 대한 규칙과 제한을 정의하는 암호 정책을 생성할 수 있습니다.

SiteMinder 에서 암호 서비스를 구성할 경우 암호 정책이 디렉터리에 연결됩니다. 디렉터리에 포함된 모든 사용자나 LDAP 검색 식으로 식별된 디렉터리의 일부는 암호 정책을 따라야 합니다. 암호 서비스는 에이전트를 호스트하는 백엔드 웹 서버가 아니라 Apache 웹 서버 내부에서 처리됩니다.

**참고:** 암호 서비스에 대한 자세한 내용은 *Policy Design Guide*(정책 설계 안내서)를 참조하십시오.

## SPS 에 대한 암호 정책 구성

CA SiteMinder for Secure Proxy Server 배포 환경에서 SiteMinder 가 암호 서비스를 구현하기 위해서는 정책 서버 사용자 인터페이스에 지정된 리디렉션 URL 에 특정 가상 디렉터리 경로를 추가하여 해당 URL 이 CA SiteMinder for Secure Proxy Server 서버를 나타내도록 해야 합니다.

**SPS 에 대한 암호 정책을 구성하려면**

1. 정책 서버 사용자 인터페이스에 로그인합니다.
2. 정책 서버 사용자 인터페이스에서 "시스템" 탭을 선택합니다.
3. '사용자 디렉터리" 개체를 클릭합니다.
4. "사용자 디렉터리 목록"에서 암호 서비스로 보호할 사용자 디렉터리를 선택합니다.
5. "속성 = 사용자 디렉터리"를 마우스 오른쪽 단추로 클릭하여 선택합니다.

   "사용자 디렉터리 속성" 대화 상자가 나타납니다.
6. "자격 증명 및 연결" 탭에서 "자격 증명 필요"를 선택합니다.
7. 사용자 이름 및 암호를 포함하여 관리자의 자격 증명을 입력합니다.

8. 같은 대화 상자의 "사용자 특성" 탭에서 다음 디렉터리 사용자 프로필 특성의 이름을 입력합니다.
   - 유니버설 ID(예: uid)
   - 비활성화된 플래그(예: carLicense)
   - Password Attribute(암호 특성)(예: userPassword)
   - 암호 데이터(예: audio)

   **참고:** "사용자 디렉터리 속성" 대화 상자에 대한 자세한 내용은 정책 서버 사용자 인터페이스 도움말을 참조하십시오.
9. "확인"을 클릭합니다.
10. "시스템" 탭에서 "암호 정책" 개체를 선택합니다.
11. "암호 정책" 개체를 마우스 오른쪽 단추로 클릭하고 "암호 정책 만들기"를 선택합니다.

    "암호 정책 속성" 대화 상자가 나타납니다.
12. "일반" 탭에서 암호 서비스를 설정한 사용자 디렉터리의 이름을 선택합니다.
13. "일반" 탭에서 "리디렉션 URL"을 다음과 같이 지정합니다.

    ```
    /siteminderagent/pw/smpwservicescgi.exe
    ```
14. "확인"을 클릭합니다.

    구성이 완료되었습니다.

## SPS 에 대한 암호 서비스 확인

SPS 에 대한 암호 서비스를 구성한 후 간단한 테스트를 수행하여 암호 서비스가 적용되는지 확인할 수 있습니다.

### 암호 서비스가 작동하는지 확인하려면

1. "사용자 디렉터리" 목록에서 암호로 보호된 디렉터리를 선택합니다.
2. "도구" 메뉴에서 "사용자 계정 관리"를 선택합니다.

   "사용자 관리" 대화 상자가 나타납니다.
3. 사용자를 선택합니다.

4. "다음에 로그인할 때 암호 변경"을 선택합니다.
5. "확인"을 클릭합니다.

다음에 CA SiteMinder for Secure Proxy Server 의 보호된 페이지를 요청할 때 인증이 요청되면 지정된 사용자의 자격 증명을 입력하십시오. "암호 변경" 화면이 나타나면 암호 서비스가 작동 중인 것입니다.

# 방화벽 고려 사항

SPS 가 포함될 DMZ 에 대한 방화벽을 구성할 때 CA SiteMinder for Secure Proxy Server 는 내부 통신에 포트 8007 및 8009 를 사용합니다. 이러한 포트는 DMZ 외부에 있는 엔터티가 액세스할 수 없도록 보호되어야 합니다.

**참고:** server.conf 파일에서 적절한 지시문을 변경하여 CA SiteMinder for Secure Proxy Server 가 사용하는 포트를 변경할 수 있습니다.

# 연결 유지 및 연결 풀링

CA SiteMinder for Secure Proxy Server 는 연결 풀을 통해 초기 서버 연결에서 생성된 워크로드를 분산하여 성능을 향상시킬 수 있도록 설계되었습니다. 성능상 대상 서버에 대해 KEEP ALIVE 설정을 지정하는 것이 좋습니다.

모든 대상 서버 제품에는 연결 유지 설정을 관리하기 위한 개별 메서드 및 특성이 있습니다. SPS 를 구성할 때 이러한 설정을 검토하고 이해해야 합니다.

# Sun Java 웹 서버용 HTTP 헤더 구성

기본적으로 Sun Java 웹 서버와 같은 일부 웹 서버는 요청에 사용할 수 있는 헤더 변수의 수를 제한합니다. 사용자 지정 헤더가 많이 포함된 트랜잭션을 처리할 수 있도록 이 상한값을 늘려야 하는 경우가 있습니다.

헤더 수가 허용 가능한 최대값보다 많을 경우 서버는 일반적으로 "요청 엔터티가 너무 큽니다."라는 413 오류를 반환합니다. 자세한 내용은 대상 서버의 관리 안내서를 참조하십시오.

**최대 헤더 수를 변경하려면**

1. 백엔드 Sun Java 웹 서버의 magnus.conf 파일을 찾아 텍스트 편집기에서 엽니다.
2. magnus.conf 에서 다음 항목을 추가하거나 수정합니다.

   *MaxRqHeaders 50*

   최대값을 CA SiteMinder for Secure Proxy Server 트랜잭션이 생성하는 헤더 수보다 높은 수준으로 설정해야 합니다.
3. 변경 내용이 적용되도록 Sun Java 웹 서버를 다시 시작합니다.

# SPS 를 사용한 SiteMinder 처리를 위한 HTTP 헤더

CA SiteMinder for Secure Proxy Server 는 기존 SiteMinder 아키텍처에 추가 계층을 도입합니다. 이 계층은 모든 요청을 엔터프라이즈 내의 대상 서버에 전달하거나 리디렉션합니다. CA SiteMinder for Secure Proxy Server 가 요청을 처리할 때 사용자가 요청한 URL 은 SM_PROXYREQUEST 라는 HTTP 헤더 변수에 유지됩니다. 이 헤더는 CA SiteMinder for Secure Proxy Server 가 요청을 프록시하기 전에 사용자가 요청한 원래 URL 을 필요로 하는 다른 응용 프로그램에 사용될 수 있습니다.

SM_PROXYREQUEST HTTP 헤더를 백엔드에 보낼 수 있도록 하려면 에이전트 구성 개체의 ProxyAgent 매개 변수 값을 YES 로 설정해야 합니다.

**참고:** 이 헤더는 요청 대상이 보호된 리소스일 때만 추가됩니다.

# 인코딩된 URL 처리

웹 서버는 정규화되거나 이스케이프된 인코딩 및 디코딩 URL 을 모두 처리할 수 있습니다. 웹 서버가 인코딩된 URL 을 처리하는 방식은 서버 유형에 따라 다릅니다. 보안상의 이유와 일관된 동작을 제공할 목적으로 CA SiteMinder for Secure Proxy Server 는 항상 URI 를 처리하기 전에 디코딩하거나 정규화합니다. 이를 통해 단일 URL 을 일관되게 표현할 수 있으며, 인코딩된 문자열을 사용한 CA SiteMinder for Secure Proxy Server 악용을 방지할 수 있습니다.

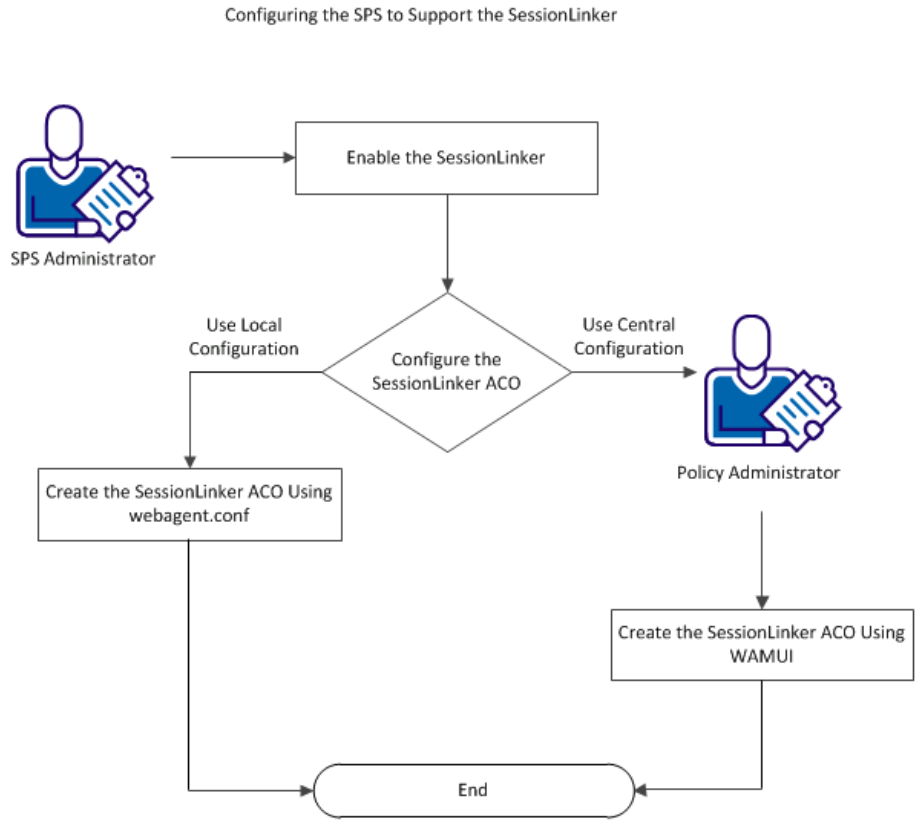
# 제 13 장: 세션 링커를 지원하도록 CA SiteMinder® SPS 구성

이 섹션은 다음 항목을 포함하고 있습니다.

# 세션 링커를 지원하도록 SPS 구성

세션 링커는 보안 향상을 위해 CA SiteMinder 세션을 타사 응용 프로그램 세션과 동기화합니다. 기본적으로 CA SiteMinder for Secure Proxy Server 는 세션 링커를 "사용 안 함" 모드로 설치합니다.

다음 다이어그램에서는 CA SiteMinder for Secure Proxy Server 관리자와 정책 관리자가 세션 링커를 지원하도록 CA SiteMinder for Secure Proxy Server 를 구성하는 방법에 대해 설명합니다.



## 세션 링커의 작동 방식

세션 링커는 보안 향상을 위해 SiteMinder 세션을 타사 응용 프로그램 세션과 동기화합니다. 사용자가 SiteMinder 에서 로그아웃하면 세션 링커는 타사 응용 프로그램의 관련 세션을 무효화합니다.

사용자가 인증되면 SiteMinder 는 해당 사용자 세션에 고유 세션 식별자를 할당합니다. SiteMinder 세션 ID 라고도 하는 이 세션 식별자는 사용자 세션의 수명 주기 동안 해당 사용자에 대해 일관되게 유지됩니다. 사용자가 로그아웃 URL 을 통해 SiteMinder 에서 로그아웃하면 SiteMinder 는 SiteMinder 세션 ID 를 추적하는 데 사용하는 SMSESSION 쿠키를 삭제합니다.

세션 링커 모듈은 응용 프로그램 세션 쿠키를 가져와 SiteMinder 세션과 하나씩 연결합니다. 연결된 응용 프로그램 쿠키(여기서는 외부 쿠키라고 함)는 해당하는 특정 SiteMinder 세션과 함께만 사용될 수 있습니다. 세션 링커는 다른 SiteMinder 세션이 동일한 외부 세션을 사용하지 못하도록 합니다.

세션 링커의 작동 방식을 이해하려면 SiteMinder 세션과 SiteMinder 가 추적하는 해당 외부 쿠키를 다음 예와 같이 표로 연결하십시오.

| SiteMinder 세션 ID | 외부 쿠키 |
|---|---|
| ONE | ABCD |
| TWO | LMNO |
| THREE | PQRST |
| FOUR | VWXY |

세션 링커는 다음 프로세스를 사용합니다.

1. 세션 링커가 웹 서버로부터 요청을 수신합니다.
2. 세션 링커가 HTTP 헤더에서 SiteMinder 세션 ID 를 추출하고 들어오는 모든 HTTP 쿠키에서 외부 쿠키를 추출합니다.
3. 세션 링커가 다음 예와 같이 웹 서버에서 제공된 값을 표의 내용과 비교하여 요청을 허용해야 하는지 여부를 결정합니다.
   a. 세션 ID 가 FIVE 이고 외부 쿠키가 RSTU 인 경우 세션 링커는 해당 값을 표에 삽입합니다.
   b. 세션 ID 가 SIX 이고 외부 쿠키가 ABCD 인 경우 외부 쿠키 ABCD 는 이미 세션 ONE 과 연결되어 있으므로 세션 링커는 요청을 차단합니다.
   c. 세션 ID 가 ONE 이고 외부 쿠키가 HIJK 인 경우 이전 세션은 고아가 되며 세션 링커는 표를 업데이트하여 세션 ID ONE 을 HIJK 와 연결합니다. 세션이 고아가 되면 해당 외부 쿠키는 더 이상 어떤 세션에서도 제공될 수 없습니다. 이 기능을 통해 세션 링커는 사용자 세션 도중 쿠키를 업데이트하는 응용 프로그램을 지원할 수 있습니다.

각 외부 쿠키에 대해 전체 프로세스가 반복됩니다. 결과 표는 다음과 같이 나타납니다.

| SiteMinder 세션 ID | 외부 쿠키 |
|---|---|
| ***Orphaned*** | ABCD |
| ONE | HIJK |
| TWO | LMNO |
| THREE | PQRST |
| FOUR | VWXY |
| FIVE | RSTU |

## 세션 링커가 지원하지 않는 기능

세션 링커는 다음과 같은 태스크를 수행하지 *않습니다*.

- CA SiteMinder® 환경 전체에서 사용자에게 발급된 쿠키 추적. 이 태스크를 수행하려면 세션 링커를 사용하는 모든 웹 서버가 읽거나 쓸 수 있는 영구 데이터 저장소가 필요합니다. 이 추적 기능을 지원하는 데 필요한 읽기 및 쓰기 작업의 수가 매우 많으면 상당한 처리 능력과 대역폭이 필요하므로 관리가 용이하지 않습니다.
- 사용자가 CA SiteMinder®에서 로그아웃할 때 기존 사용자의 쿠키 삭제. 쿠키는 중앙에서 추적되지 않으므로 삭제할 쿠키를 판별할 메커니즘이 없습니다. 또한 웹 브라우저마다 쿠키를 처리하는 방식이 다르므로 로그아웃 페이지에서 사용자가 수신한 쿠키를 확인할 수 없는 경우도 있습니다. 마지막으로, 세션 링커는 실제로 CA SiteMinder® 로그아웃 프로세스와 통합되지 않습니다.
- 기본 응용 프로그램의 세션 종료. 이 기능을 지원하려면 세션 링커가 각 응용 프로그램의 세션 종료 방법을 알고 있어야 하지만 대부분의 응용 프로그램에는 세션을 관리하기 위해 노출되는 API 가 없습니다. 응용 프로그램은 유휴 상태에서 시간이 지나면 세션을 종료하도록 구성될 수 있고 세션을 활성 상태로 유지하기 위한 오버헤드도 거의 없으므로 이 기능이 구현되지 않았습니다.

세션 링커는 연결을 수행할 때 사용자가 유효하지 않은 외부 세션 쿠키를 제공하지 못하도록 합니다.

## 세션 링커 사용

CA SiteMinder 세션을 타사 응용 프로그램 세션과 동기화하려면 세션 링커를 사용하도록 설정하십시오. 이 기능을 사용하도록 설정한 후 세션 링커 ACO 를 구성할 수 있습니다.

**다음 단계를 수행하십시오.**

1. "가상 호스트", "가상 호스트", "가상 호스트 추가/편집", "웹 에이전트 구성"으로 이동합니다.
2. "세션 링커 사용" 옵션을 선택합니다.
3. "저장"을 클릭합니다.
4. CA SiteMinder for Secure Proxy Server 서버를 다시 시작합니다.

## NPS_Session_Linker ACO 생성

CA SiteMinder for Secure Proxy Server 는 ACO 를 통해 세션 링커 구성을 관리합니다. 세션 링커 ACO 의 구문은 다음과 같습니다.

```
SessionLinker=Cookie=cookie_value;BLOT|NOBLOT;Orphantimeout=timeout_value;
```

설명

**COOKIE=*cookie name***

외부 세션을 보관하는 쿠키의 이름을 지정합니다. 쿠키 이름이 변경될 수 있는 경우 와일드카드 문자로 별표를 사용하십시오.

**BLOT|NOBLOT**

(선택 사항) 세션 링커가 유효하지 않은 세션에 응답하는 방식을 지정합니다. 이 매개 변수의 값을 BLOT 로 설정하면 사용자에게 액세스 권한이 부여되지만 외부 세션 쿠키가 웹 서버를 통해 대상 페이지에 전달되지는 않습니다. 이 매개 변수의 값을 NOBLOT 로 설정하면 요청에서 외부 쿠키가 삭제되며 사용자는 URL 매개 변수에 지정된 URL 로 리디렉션됩니다. URL 매개 변수에 URL 을 지정하지 않은 경우에는 내부 서버 오류가 표시됩니다.

**기본값:** BLOT

**ORPHANTIMEOUT=*seconds***

세션 링커가 고아 세션을 유지하는 시간(초)을 지정합니다.

**기본값**: 86400(24 기간의 초)

**제한**: 타사(외부) 응용 프로그램의 쿠키가 허용되는 *최대 시간(초)*보다 *작을* 수 없습니다.

CA SiteMinder for Secure Proxy Server 가 세션 링커를 지원하도록 구성하려면 다음 *중 한* 방법으로 SessionLinker 라는 ACO 를 생성하십시오.

- webagent.conf 파일 사용
- WAMU 사용

## webagent.conf 를 사용하여 NPS_Session_Linker ACO 생성

로컬 구성을 사용하여 세션 링커 ACO 를 생성하려면 webagent.conf 파일을 사용하여 ACO 를 생성하십시오.

**다음 단계를 수행하십시오.**

1. webagent.conf 파일에 지정된 localconfigfile 을 엽니다.
2. 파일에 다음 명령을 추가합니다.

   SessionLinker= Cookie=*cookie_value;*BLOT|NOBLOT;Orphantimeout=*timeout_value*;
3. 변경 내용을 저장합니다.

세션 링커 ACO 가 생성되었습니다. 또한 CA SiteMinder for Secure Proxy Server 가 세션 링커를 지원하도록 구성되었습니다.

세션 링커가 쿠키와 함께 작동하도록 구성하려면 쿠키 작업을 참조하십시오. 세션 링커로 인해 발생하는 오류를 해결하려면 문제 해결을 참조하십시오.

## 관리 UI 를 사용하여 NPS_Session_Linker ACO 생성

중앙 구성을 사용하여 세션 링커 ACO 를 생성하려면 정책 관리자가 관리 UI 를 통해 ACO 를 생성해야 합니다.

**다음 단계를 수행하십시오.**

1. WAMUI 를 엽니다.
2. 다음 상세 정보와 함께 ACO 를 추가합니다.

   **이름**: SessionLinker

   **값**: Cookie=*cookie_name;*BLOT|NOBLOT;Orphantimeout=*timeout_value*;
3. "저장"을 클릭합니다.

세션 링커 ACO 가 생성되었습니다. 또한 CA SiteMinder for Secure Proxy Server 가 세션 링커를 지원하도록 구성되었습니다.

세션 링커가 쿠키와 함께 작동하도록 구성하려면 쿠키 작업을 참조하십시오. 세션 링커로 인해 발생하는 오류를 해결하려면 문제 해결을 참조하십시오.

# 쿠키 작업

## 단일 세션 쿠키 적용

대부분의 경우 응용 프로그램에는 연결된 세션 쿠키에 대해 항상 사용되는 특정 이름이 있습니다. 그 외의 경우 쿠키 이름은 ASPSESSIONID 또는 MYAPPSESSION 와 같은 알려진 문자열로 시작하고 임의의 또는 예측할 수 없는 접미사로 끝납니다. 이러한 경우 세션 링커는 사용자가 이러한 쿠키를 둘 이상 제공하지 못하도록 하고 예상된 세션 연결을 적용합니다.

잠재적 세션 쿠키를 여러 개 발견할 경우 세션 링커는 다음 단계를 수행합니다.

1. 세션에 대한 액세스를 차단합니다.
2. 모든 쿠키를 삭제합니다.
3. 사용자를 지정된 URL 로 리디렉션합니다. URL 을 지정하지 않은 경우에는 내부 서버 오류가 표시됩니다.

## 와일드카드 쿠키 이름 사용

정책 서버에 구성된 ACO 의 다음 매개 변수를 이미 선택된 구성 설정에 추가할 수 있습니다.

**COOKIE**

지정된 이름으로 시작하는 쿠키를 잠재적 외부 세션 쿠키로 간주해야 하는지를 지정합니다. 쿠키 값은 별표(*)로 끝날 수 있습니다. 와일드카드 구문이 아닌 쿠키 값을 지정할 경우 들어오는 쿠키의 삭제 방법을 결정하는 COOKIEPATH 및 COOKIEDOMAIN 값을 지정해야 합니다.

**COOKIEPATH**

쿠키 경로를 지정합니다. COOKIE 매개 변수에 와일드카드 구문을 지정한 경우에는 이 매개 변수를 지정하지 마십시오. COOKIEPATH 값은 세션 쿠키에 따라 다르며 다음 형식으로 지정됩니다.

COOKIEPATH=<*쿠키 또는 아웃바운드 쿠키의 경로*>

**기본값**: /

**예**: COOKIEPATH=/

**COOKIEDOMAIN**

쿠키 도메인을 지정합니다. COOKIE 매개 변수에 와일드카드 구문을 지정한 경우 이 값을 다음 형식으로 지정할 수 있습니다.

COOKIEDOMAIN=<*쿠키 또는 아웃바운드 쿠키의 도메인 이름*>

**기본값**: 공백

**예**: COOKIEDOMAIN=.ca.com

## 쿠키 설정 확인

쿠키 이름 설정을 확인하려면 다음 단계를 수행하십시오.

1. 응용 프로그램에 여러 번 액세스합니다.
2. 응용 프로그램이 전송한 쿠키를 적어 둡니다.
3. 웹 브라우저에서 쿠키에 대한 경고 메시지가 표시되도록 합니다.
4. 나타나는 경고를 확인합니다.

## COOKIE 설정

응용 프로그램의 세션 쿠키 이름이 동일한 텍스트 문자열로 시작하지만 다르게 끝나는 경우 다음 형식으로 쿠키 이름을 지정하십시오.

COOKIE=*cookiename**

## COOKIEDOMAIN 설정

쿠키의 도메인 이름은 다음 항목으로 구성됩니다.

- 선행 마침표(.)가 접두사로 붙은 웹 서버의 도메인 이름
- 웹 서버의 정규화된 컴퓨터 이름(myserver.example.com)
- 공백

정규화된 컴퓨터 이름이나 빈 이름은 동일합니다.

**참고**: Internet Explorer 는 도메인을 표시하기 전에 선행 마침표를 삭제합니다. 따라서 스테이징 환경에서 다양한 구성을 테스트하여 올바른 설정을 결정하는 것이 좋습니다.

## COOKIEPATH 설정

쿠키와 연결된 경로는 일반적으로 디렉터리이지만 파일이나 다른 문자열일 수도 있습니다. 웹 브라우저의 쿠키 경고 대화 상자에 표시된 경로를 확인하십시오. 표시된 경로가 슬래시(/)가 *아니면* COOKIEPATH 설정에 올바른 값을 입력하십시오.

## 여러 쿠키에 대한 연결 유지

일부 웹 응용 프로그램은 사이트의 동일한 영역 내에서 동시에 여러 개의 쿠키를 사용합니다. 단일 CA SiteMinder 세션과 여러 쿠키의 연결을 유지하도록 세션 링커를 구성할 수 있습니다. 단일 CA SiteMinder 세션에 최대 10 개의 외부 세션 쿠키가 연결될 수 있습니다.

**다음 단계를 수행하십시오.**

1. 각 쿠키에 올바른 구성 설정을 결정합니다.

   **참고**: 각 구성 문자열에는 COOKIE 지시문이 하나 이상 필요하지만 어떤 지시문이든 함께 사용할 수 있습니다.

2. 각 쿠키에 0 에서 9 사이의 정수를 할당합니다.
3. 지시문 이름에 선택한 숫자를 추가합니다.

   **참고**: 각 지시문 집합에 어떤 숫자든 사용할 수 있지만 단일 쿠키에 대한 설정에는 동일한 숫자를 사용해야 합니다.

4. 개별 구성 문자열을 단일 문자열로 연결합니다.

## SessionLinker 문제 해결

오류가 발생하면 다음 사항을 고려하여 오류를 해결하십시오.

- 유효한 SMSESSION 쿠키 및 FOREIGN SESSION 쿠키가 사용자에게 설정되어 있고 CA SiteMinder for Secure Proxy Server 에 전달되었는지 확인하십시오.
- webagent.conf 파일을 사용하여 세션 링커가 사용되도록 설정한 경우 웹 에이전트가 사용되는지 확인하십시오.
- 세션 링커 ACO 구문이 올바른지 확인하십시오.
- 세션 링커 ACO 에서 에이전트 추적이 사용되도록 설정된 경우 에이전트 로그 및 추적 내역에서 로그 및 추적 메시지를 확인하십시오.
- CA SiteMinder for Secure Proxy Server 가 세션 링커 플러그인 바이너리를 올바르게 로드했는지 확인하십시오. agents.log 파일의 로그 메시지를 확인하십시오. 오류가 있으면 CA SiteMinder for Secure Proxy Server 에 세션 링커 플러그인 라이브러리에 대한 종속 라이브러리가 있는지 확인하십시오.
- 요청이 거부되는 경우 CA SiteMinder® 정책 서버의 세션 식별자(SMSESSION)와 응용 프로그램 웹 서버의 세션 식별자(FOREIGN SESSION)가 동일한 사용자에 연결되어 있는지 확인하십시오.

# 제 14 장: SSL 및 보안 프록시 서버

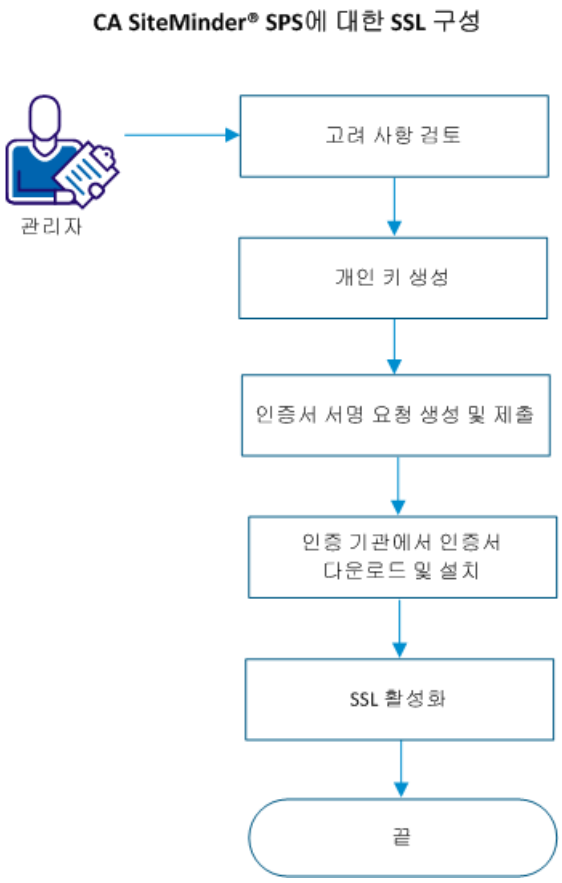
이 섹션은 다음 항목을 포함하고 있습니다.



## SPS 에 대해 SSL 구성

CA SiteMinder for Secure Proxy Server 는 클라이언트와 서버 사이에 보안 통신을 제공하기 위해 SSL 을 지원합니다. CA SiteMinder for Secure Proxy Server 는 SSL v2/v3 및 TLS(Transport Layer Security) v1 네트워크 프로토콜과 이러한 프로토콜에 필요한 관련 암호화 표준을 구현하는 OpenSSL 암호화 도구 키트를 사용합니다. OpenSSL 도구 키트에는 키 및 인증서 생성을 위한 openssl 명령줄 도구가 포함되어 있습니다. openssl 실행 파일 이미지와 지원 라이브러리는 *installation_home*\SSL\bin 폴더나 해당하는 UNIX 디렉터리에 있습니다.

다음 순서도는 CA SiteMinder for Secure Proxy Server 에 대해 SSL 을 구성하는 방법을 설명합니다.



**다음 단계를 수행하십시오.**

1. 고려 사항을 검토합니다.
2. 다음 단계 중 *하나*를 사용하여 개인 키를 만듭니다.
   - 개인 암호화된 RSA 서버 키를 만듭니다.
   - 개인 암호화되지 않은 RSA 서버 키를 만듭니다.
3. 다음 단계 중 *하나*를 사용하여 인증 서명 요청을 생성하여 제출합니다.
   - 인증 서명 요청을 생성하여 인증 기관에 제출합니다.
   - 인증 서명 요청을 생성하여 자체 서명합니다.
4. 인증 기관으로부터 인증서를 다운로드하여 설치합니다.
5. 다음 단계 중 *하나*를 사용하여 SSL 을 활성화합니다.
   - 암호화된 개인 키에 대해 SSL 을 활성화합니다.
   - Windows 에서 암호화되지 않은 개인 키에 대해 SSL 을 활성화합니다.
   - UNIX 에서 암호화되지 않은 개인 키에 대해 SSL 을 활성화합니다.

SSL 이 구성되었습니다.

## 고려 사항 검토

SSL 을 구성하기 전에 개인 키와 서버 인증서에 대한 다음 정보를 검토하십시오.

- 서버 인증서 및 개인 키는 함께 작동하므로 서버 인증서를 해당 개인 키와 함께 사용하십시오.
- 서버 인증서는 인증 기관(CA)에 의해 디지털 서명되어야 합니다. 내부 데모를 위해 SSL 을 활성화하려는 경우 서버 인증서는 자체 개인 키를 사용하여 자체 서명될 수 있습니다.
- SSL.conf 파일의 SSLCertificateFile 및 SSLCertificateKeyFile 지시문은 해당하는 인증서 및 키 파일을 가리켜야 합니다.
- Apache 의 가상 호스트 기능을 사용하는 경우 보호하려는 각 가상 호스트마다 별도의 개인 키와 서버 인증서가 있어야 합니다.

## 개인 키 생성

SSL 은 키를 사용하여 메시지를 암호화 및 암호 해독합니다. 키는 공개 키 하나와 개인 키 하나의 쌍으로 이루어집니다.

키에는 다양한 암호화 알고리즘과 키 교환 방법이 사용됩니다. 인증서와 함께 사용하기 위해 개인 키를 생성할 때는 DES(Date Encryption Standard) 암호화 알고리즘을 사용한 RSA 키 교환 방식이 일반적으로 사용됩니다. 키 출력 파일은 암호화된 ASCII PEM 형식을 사용합니다.

## 개인 암호화된 RSA 서버 키 생성

서버 키를 보호하려면 개인 암호화된 RSA 서버 키를 만드십시오.

**다음 단계를 수행하십시오.**

1. 명령줄 창을 엽니다.
2. 다음 디렉터리로 이동합니다.

   *installation_home*\SSL\bin

   ***installation_home***

   CA SiteMinder for Secure Proxy Server 가 설치된 디렉터리를 정의합니다.

   **기본값**: (Windows) [32 비트] C:\Program Files\CA\secure-proxy

   **기본값**: (Windows) [64 비트] C:\CA\secure-proxy

   **기본값**: (UNIX/Linux) /opt/CA/secure-proxy
3. 다음 명령을 실행합니다.

   .\openssl genrsa -des3 -out ..\keys*server*.key [*numbits*]

   ***server***

   서버의 정규화된 도메인 이름을 지정합니다.

   **(선택 사항) *numbits***

   반드시 생성되어야 하는 개인 키의 비트 크기를 지정합니다.

   **기본값**: 1024

   **범위**: 1024 - 2048

   개인 암호화된 서버 키가 생성되고 지정된 키 출력 파일에 작성됩니다. 키 출력 파일은 암호화된 ASCII PEM 형식을 사용합니다. 이 파일은 암호화되므로 파일을 보호하고 나중에 필요할 경우 암호 해독하는 데 사용할 암호를 묻는 메시지가 표시됩니다.

## 개인 암호화되지 않은 RSA 서버 키 생성

서버 키를 보호하지 않으려면 개인 암호화되지 않은 RSA 서버 키를 만드십시오.

**다음 단계를 수행하십시오.**

1. 명령줄 창을 엽니다.
2. 다음 디렉터리로 이동합니다.

   *installation_home*\SSL\bin

   ***installation_home***

   CA SiteMinder for Secure Proxy Server 가 설치된 디렉터리를 정의합니다.

   **기본값**: (Windows) [32 비트] C:\Program Files\CA\secure-proxy

   **기본값**: (Windows) [64 비트] C:\CA\secure-proxy

   **기본값**: (UNIX/Linux) /opt/CA/secure-proxy
3. 다음 명령을 실행합니다.

   .\openssl genrsa -out ..\keys*server*.key [*numbits*]

   ***server***

   서버의 정규화된 도메인 이름을 지정합니다.

   **(선택 사항) *numbits***

   반드시 생성되어야 하는 개인 키의 비트 크기를 지정합니다.

   **기본값**: 1024

   **범위**: 1024 - 2048

   개인 암호화되지 않은 서버 키가 생성됩니다.

# 인증서 서명 요청 생성 및 제출

개인 키를 사용하여 인증서 요청 또는 인증서 서명 요청을 만드십시오. 인증서 서명을 위해 인증서 서명 요청을 인증 기관에 보내거나 내부 데모를 위해 자체 서명한 인증서를 만들 수 있습니다.

## 인증서 서명 요청을 생성하여 인증 기관에 제출

인증서 서명 요청을 생성하여 조직이 사용하는 인증 기관에 제출하십시오.

**다음 단계를 수행하십시오.**

1. 명령줄 창을 엽니다.
2. 다음 명령을 실행합니다.

   ```
   .\openssl req -config .\openssl.cnf -new -key ..\keys\server.key
   -out ..\keys\server.csr
   ```

3. 값을 요청하면 값을 입력합니다.

   시스템이 인증서 파일 이름과 요청 번호를 사용하여 인증서 요청을 생성합니다.

4. (선택 사항) 이후 참조할 수 있도록 파일 이름과 인증서 서명 요청을 기록해 둡니다.
5. 인증 기관으로 인증서 서명 요청을 제출합니다.

## 자체 서명된 인증서 생성

내부 데모를 위해 SSL 을 활성화하려면 자체 서명된 인증서를 만드십시오.

**다음 단계를 수행하십시오.**

1. 명령줄 창을 엽니다.
2. 다음 명령을 실행합니다.

   ```
   .\openssl req -config .\openssl.cnf -new -x509 -key ..\keys\server.key
   -out ..\certs\cert_name.crt
   ```

3. 출력을 다음 위치에 넣으십시오.

   *sps_home*\SSL\certs

   자체 서명된 인증서가 생성되었습니다.

## 인증 기관으로부터 인증서를 다운로드하여 설치

인증 기관으로부터 자체 서명된 인증서를 다운로드하십시오.

**다음 단계를 수행하십시오.**

1. 인증서 요청을 생성한 CA SiteMinder for Secure Proxy Server 호스트에 로그인합니다.
2. httpd-ssl conf 파일을 엽니다.

   **기본 경로**: *installation_home*\httpd\conf\extra\httpd-ssl.conf
3. 서버 키 및 인증서의 지시문이 올바른지 확인합니다.
4. SSLPassPhraseDialog 변수의 값을 custom 으로 설정합니다.
5. SSLCustomPropertiesFile 변수의 값을 <*installation_home*>\httpd\conf\spsapachessl.properties 로 설정합니다.
6. 루트 CA 에 대한 참조가 설정되었는지 확인합니다.
7. 다음 단계를 수행하여 RootCA 또는 자체 서명된 인증서를 ca-bundle.cert 에 추가합니다.
   a. 메모장에서 인증서를 열고 BEGIN 부터 END 까지의 행을 복사합니다.
   b. 메모장에서 ca-bundle.cert 를 열고 끝에 인증서에서 복사한 줄을 붙여넣습니다.
   c. 변경 내용을 저장합니다.

## SSL 사용

암호화되거나 암호화되지 않은 개인 키에 대해 SSL 을 활성화할 수 있습니다.

## Windows 에서 암호화되지 않은 개인 키에 대해 SSL 활성화

Windows 에서 암호화되지 않은 개인 키에 대해 SSL 을 활성화하려면 spsapachessl.properties 파일을 생성하십시오.

**다음 단계를 수행하십시오.**

1. 관리자 권한으로 명령줄 창을 엽니다.
2. 다음 디렉터리로 이동합니다.

   ```
   installation_home\httpd\bin
   ```

3. 다음 스크립트 파일을 실행합니다.

   ```
   configssl.bat -enable
   ```

   **참고**: 덮어쓰기 경고가 표시되면 기존 spsapachessl.properties 파일의 덮어쓰기를 승인하십시오.

   SSL 이 구성되었습니다.

## UNIX 에서 암호화되지 않은 개인 키에 대해 SSL 활성화

UNIX 에서 암호화되지 않은 개인 키를 활성화하려면 다음 위치에 있는 spsapachessl.properties 를 편집하십시오.

*installation_home*/httpd/conf/spsapachessl.properties

**다음 단계를 수행하십시오.**

1. 텍스트 편집기에서 spsapachessl.properties 파일을 엽니다.
2. 다음 줄을 추가 또는 편집합니다.
3. 다음 작업 중 하나를 수행합니다.
   - apache.ssl.enabled=가 파일에 있으면 다음 값으로 이 줄을 설정하십시오.

     ```
     apache.ssl.enabled=Y
     ```

   - apache.ssl.enabled=가 파일에 없으면 다음 형식으로 행을 추가하십시오.

     ```
     apache.ssl.enabled=Y
     ```

4. 변경 내용을 저장합니다.

   SSL 이 구성되었습니다.

### 암호화된 개인 키에 대해 SSL 을 활성화

암호화된 개인 키에 대해 SSL 을 활성화하려면 spsapachessl.properties 파일을 생성하십시오.

**다음 단계를 수행하십시오.**

1. 관리자 권한으로 명령줄 창을 엽니다.
2. 다음 디렉터리로 이동합니다.

   **Windows**

   ```
   installation_home\httpd\bin
   ```

   **UNIX**

   ```
   installation_home/httpd/bin
   ```

3. 다음 스크립트를 실행합니다.

   **Windows**

   ```
   configssl.bat -enable passphrase
   ```

   **UNIX**

   ```
   configssl.sh passphrase
   ```

   **참고**: passphrase 값은 서버 키의 passphrase 값과 일치해야 합니다. 덮어쓰기 경고가 표시되면 기존 spsapachessl.properties 파일의 덮어쓰기를 승인하십시오.

4. 암호는 암호화되어 spsapachessl.properties 파일에 저장됩니다.
5. 보안 프록시 서비스를 다시 시작합니다.

   SSL 이 구성되었습니다.

# 가상 호스트에 SSL 사용

Apache 서버는 가상 호스트, 즉 단일 Apache 바이너리에서 실행되는 여러 웹 호스트를 지원합니다. Apache 가상 호스트는 이름 기반이거나 IP 기반일 수 있습니다. 이름 기반 가상 호스트는 단일 IP 주소를 공유할 수 있는 반면에 IP 기반 가상 호스트의 경우에는 각 가상 호스트마다 서로 다른 IP 주소가 필요합니다.

SSL 프로토콜을 사용하는 Apache 가상 호스트는 다음 조건을 충족해야 합니다.

- 프로토콜의 제한 사항으로 인해 IP 기반이어야 합니다.
- Apache 구성 파일에 보안(HTTPS) 및 비보안(HTTP) 요청 모두에 대한 가상 호스트 컨테이너가 있어야 합니다.

다음은 보안(HTTPS) 가상 호스트의 예입니다.

```
<VirtualHost 10.0.0.1:443>
DocumentRoot ".../htdocs/site1"
ServerName www.site1.net
ServerAdmin webmaster@site1.net
ErrorLog logs/covalent_error_log_site1
TransferLog logs/...
SSLEngine on
SSLCertificateFile /www.site1.net.cert
SSLCertificateKeyFile /www.site1.net.key
CustomLog logs/cipher_log_site1 \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
```

# 제 15 장: Windows 통합 인증을 지원하도록 CA SiteMinder® SPS 구성

이 섹션은 다음 항목을 포함하고 있습니다.

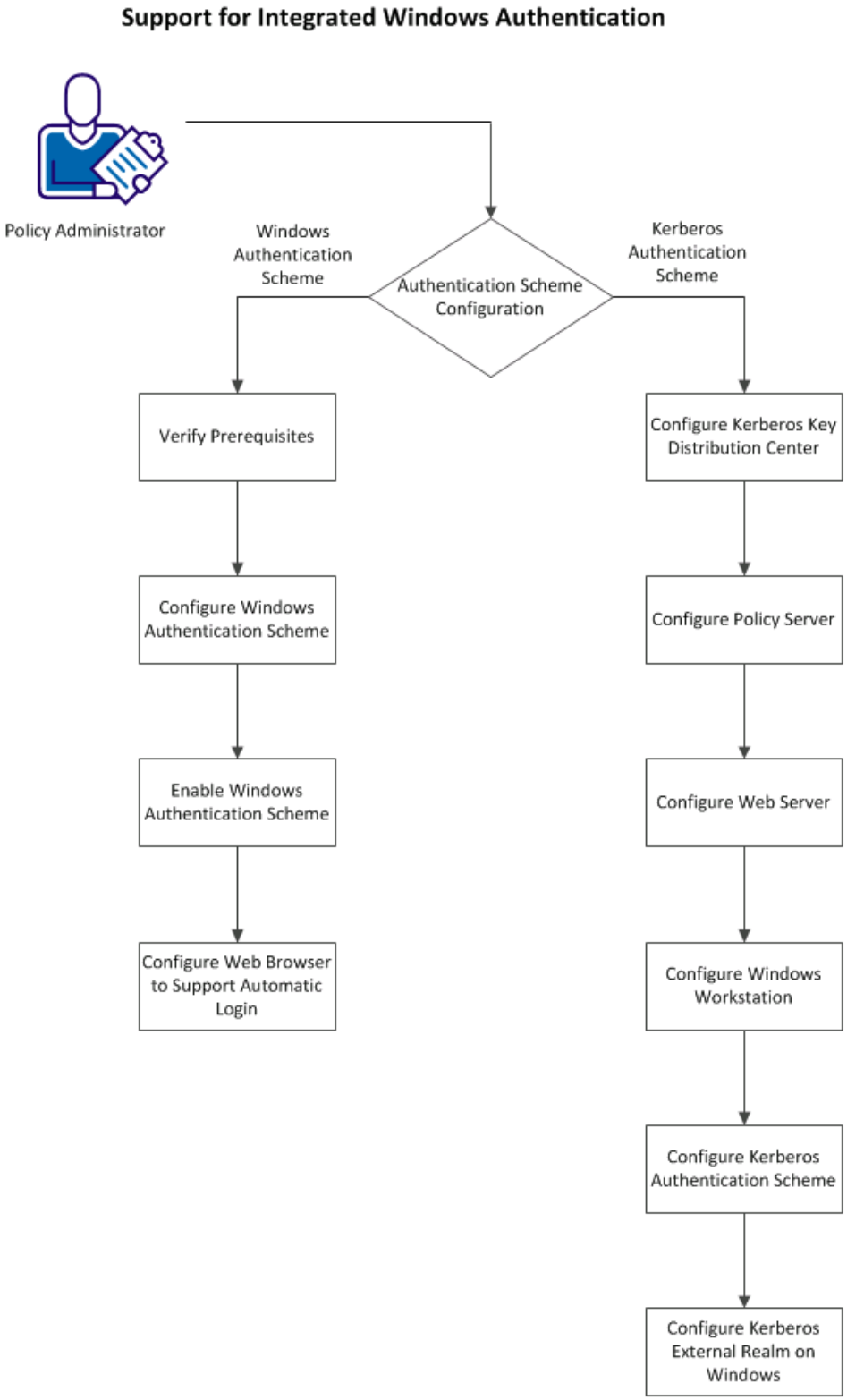


## Windows 통합 인증을 지원하도록 SPS 구성

IWA(Windows 통합 인증)는 Windows 클라이언트 및 서버의 보안 기능을 사용합니다. IWA 를 사용할 경우 초기 대화형 데스크톱 로그인 프로세스 중에 Windows 가 사용자 자격 증명을 수집할 수 있으며 그런 다음 이 정보가 보안 계층에 전송됨으로써 싱글 사인온이 적용됩니다.

Windows 인증 체계를 사용하는 SiteMinder 는 Microsoft Windows 통합 인증 인프라를 통해 얻은 사용자 자격 증명을 처리하여 리소스를 보호합니다.

다음 다이어그램에서는 IWA 를 지원하도록 CA SiteMinder for Secure Proxy Server 를 구성하는 방법에 대해 설명합니다.


Support for Integrated Windows Authentication
Policy Administrator
Windows Authentication Scheme
Authentication Scheme Configuration
Kerberos Authentication Scheme
Verify Prerequisites
Configure Windows Authentication Scheme
Enable Windows Authentication Scheme
Configure Web Browser to Support Automatic Login
Configure Kerberos Key Distribution Center
Configure Policy Server
Configure Web Server
Configure Windows Workstation
Configure Kerberos Authentication Scheme
Configure Kerberos External Realm on Windows

CA SiteMinder for Secure Proxy Server 에는 다음 인증 체계 중 *하나*를 구성할 수 있습니다.

- Windows 서버의 Windows 인증 체계
- Windows 및 UNIX 서버의 Kerberos 인증 체계

## Windows 인증 체계

Windows 인증 체계를 사용하면 Active Directory 가 네이티브 모드로 실행 중이고 NTLM 인증을 지원하도록 구성된 배포 환경에서 SiteMinder 를 통해 액세스를 제어할 수 있습니다. Windows 인증 체계는 SPNEGO 를 사용하여 이니시에이터 및 승인자가 Kerberos 또는 NTLMSSP 와 협상할 수 있도록 합니다.

### Windows 인증 구성

Windows 인증을 지원하려면 다음 단계를 수행하십시오.

1. 사전 요구 사항을 확인합니다.
2. Windows 인증 체계를 구성합니다.
3. Windows 인증 체계를 사용하도록 설정합니다.
4. 자동 로그인을 지원하도록 웹 브라우저를 구성합니다.

### 사전 요구 사항 확인

IWA 를 지원하도록 CA SiteMinder for Secure Proxy Server 를 구성하기 전에 다음 태스크를 수행해야 합니다.

1. Windows 도메인 컨트롤러를 구성합니다.
2. Windows 도메인 컨트롤러에 대한 도메인 호스트의 구성원으로 CA SiteMinder for Secure Proxy Server 호스트를 추가합니다.
3. 혼합 모드의 레거시 WinNT 디렉터리 또는 Active Directory 가 다음 조건을 충족하는지 확인합니다.
   - 관리 UI 에서 생성한 사용자 디렉터리 연결에 WinNT 네임스페이스가 지정되어 있습니다.
   - 요청된 리소스를 모든 유형의 웹 서버에 배치할 수 있습니다.

4. 네이티브 모드에서 실행되는 Active Directory 가 다음 조건을 충족하는지 확인합니다.
   - 사용자 데이터가 Active Directory 에 있습니다.
   - 사용자 디렉터리 연결에 LDAP 또는 AD 네임스페이스가 지정되어 있어야 합니다.
   - 요청된 리소스를 모든 유형의 웹 서버에 배치할 수 있습니다.
   - 클라이언트 및 서버 계정을 위임할 수 있습니다.

## Windows 인증 체계 구성

Windows 인증 체계를 사용하여 Windows 환경에서 사용자를 인증할 수 있습니다.

**참고**: 다음 절차에서는 개체를 생성하고 있다고 가정합니다. 기존 개체의 속성을 복사하여 개체를 생성할 수도 있습니다. 자세한 내용은 "정책 서버 개체 복제"를 참조하십시오.

**다음 단계를 수행하십시오.**

1. "인프라", "인증"을 차례로 클릭합니다.
2. "인증 체계"를 클릭합니다.
3. "인증 체계 만들기"를 클릭합니다.

   "인증 체계 유형의 새 개체 만들기"가 선택되어 있는지 확인합니다.

   "확인"을 클릭합니다.
4. 이름 및 보호 수준을 입력합니다.
5. "인증 체계 유형" 목록에서 "Windows 인증 템플릿"을 선택합니다.

   체계 특정 설정이 표시됩니다.

6. "서버 이름", "대상" 및 "사용자 DN" 정보를 입력합니다. 사용자 환경에 NT 챌린지/응답 인증이 필요한 경우 에이전트 소유자에게서 다음의 값을 가져옵니다.

   **서버 이름**

   CA SiteMinder for Secure Proxy Server 의 정규화된 도메인 이름입니다. 예를 들면 다음과 같습니다.

   server1.myorg.com

   **대상**

   /siteminderagent/ntlm/smntlm.ntc

   **참고:** 디렉터리는 설치 환경에서 이미 구성된 가상 디렉터리와 일치해야 합니다. 대상 파일인 smntlm.ntc 는 없어도 되며 .ntc 로 끝나는 모든 이름이나 기본값 대신 사용하는 사용자 지정 MIME 유형일 수 있습니다.

   **라이브러리**

   smauthntlm

7. 제출을 클릭합니다.

   인증 체계가 저장되고 이를 영역에 할당할 수 있습니다.

**참고**: NTLM 인증에서 사용자 인증이 실패하면 브라우저가 중지할 때까지 인증 프로세스가 계속됩니다. 이 문제를 해결하려면 인증이 실패할 경우 사용자를 사용자 지정 페이지로 리디렉션하는 다음과 같은 리디렉션 응답을 만드십시오.

- onauthreject 및 onauthusernotfound 를 사용한 규칙
- Webagent-onreject-redirect 를 사용한 응답

## Windows 인증 체계 사용

이제 CA SiteMinder for Secure Proxy Server 는 정책 서버에 구성된 Windows 인증 체계를 지원합니다. CA SiteMinder for Secure Proxy Server 가 Windows 인증 체계를 지원하도록 하려면 다음 단계를 수행하십시오.

1. 정책 서버에서 WindowsNativeAuthentication ACO 매개 변수를 생성합니다.
2. WindowsNativeAuthentication 의 값을 no 로 설정합니다.

**참고**: WindowsNativeAuthentication ACO 매개 변수의 값을 정의 또는 설정하지 않으면 CA SiteMinder for Secure Proxy Server 는 Windows 인증을 지원하지 않습니다.

## 자동 로그인을 지원하도록 웹 브라우저 구성

Internet Explorer 5.x 및 6.x 브라우저의 자동 로그온을 구성하려면 다음 단계를 수행하십시오.

1. Internet Explorer 의 메뉴 표시줄에서 "도구", "인터넷 옵션"을 차례로 선택합니다.

   "인터넷 옵션" 대화 상자가 열립니다.

2. "보안" 탭을 클릭하여 표시합니다.
3. "인터넷 영역"을 선택하고 "사용자 지정 수준"을 클릭합니다.

   "보안 설정" 대화 상자가 열립니다.

4. 아래로 스크롤하여 "사용자 인증", "로그온"을 찾습니다.
5. "현재 사용자 이름 및 암호를 사용하여 자동으로 로그온" 라디오 단추를 선택합니다.
6. "확인"을 클릭합니다.

Internet Explorer 4.x 브라우저의 자동 로그온을 구성하려면 다음 단계를 수행하십시오.

1. Internet Explorer 의 메뉴 표시줄에서 "보기", "인터넷 옵션"을 차례로 선택합니다.

   "인터넷 옵션" 대화 상자가 열립니다.

2. "보안" 탭을 클릭하여 표시합니다.
3. 드롭다운 목록에서 "인터넷 영역"을 선택합니다.
4. "인터넷 영역" 그룹 상자에서 "사용자 지정" 라디오 단추를 선택하여 클릭하고 "설정"을 클릭합니다.

   "보안 설정" 대화 상자가 열립니다.

5. 아래로 스크롤하여 "사용자 인증", "로그온"을 찾습니다.
6. "현재 사용자 이름 및 암호를 사용하여 자동으로 로그온" 라디오 단추를 선택합니다.
7. "확인"을 클릭합니다.

CA SiteMinder for Secure Proxy Server 가 Windows 인증을 지원하도록 구성되었습니다.

## Kerberos 인증 체계

Kerberos 는 개방형 네트워크에서 클라이언트와 서버 간의 인증 수단을 제공하기 위해 MIT 에서 고안한 표준 프로토콜입니다. Kerberos 프로토콜은 메시지를 도청 및 재생 공격으로부터 보호합니다. Kerberos 는 공유 암호, 대칭 키 및 Kerberos 서비스를 사용합니다. Microsoft Windows 운영 환경에서는 Kerberos V5 를 기본 인증 패키지로 사용합니다. Solaris 10 에도 Kerberos V5 가 포함되어 있습니다.

Kerberos 환경에서는 사용자 계정과 서비스 계정을 프린서펄이라고 합니다. Kerberos 는 트러스트된 타사, 즉 KDC(Key Distribution Center)를 사용하여 프린서펄 간의 메시지 교환을 중개합니다. Key Distribution Center 의 용도는 키 교환 시 내재된 위험을 줄이는 것입니다.

Kerberos 인증은 티켓을 요청하고 전달하는 메시지를 기반으로 합니다. Key Distribution Center 는 다음 두 가지 유형의 티켓을 처리합니다.

- TGT(Ticket-Granting Ticket) - KDC 가 요청자의 자격 증명을 TGS(Ticket-Granting Service)에 전송할 때 내부적으로 사용됩니다.
- 세션 티켓 - TGS(Ticket-Granting Service)가 요청자의 자격 증명을 대상 서버 또는 서비스에 전송하는 데 사용됩니다.

Kerberos 는 keytab 파일을 사용하여 KDC 에 로그인합니다. Keytab 파일은 Kerberos 프린서펄과 Kerberos 암호에서 파생된 암호화된 키의 쌍으로 구성됩니다.

Kerberos 프로토콜 메시지 교환 과정을 간단히 요약하면 다음과 같습니다.

1. 사용자가 로그인하면 클라이언트는 KDC 인증 서비스에 연결하여 사용자 아이덴티티 정보가 포함된 일시적인 메시지(Ticket-Granting Ticket)를 요청합니다.
2. KDC 인증 서비스는 TGT 를 생성하고 클라이언트가 Ticket-Granting Service 와의 통신을 암호화하는 데 사용할 수 있는 세션 키를 생성합니다.
3. 사용자가 로컬 또는 네트워크 리소스에 대한 액세스를 요청하면 클라이언트는 KDC 에 TGT(Ticket-Granting Ticket), 인증자 및 대상 서버의 SPN(서비스 프린서펄 이름)을 제공합니다.

4. Ticket-Granting Service 가 Ticket-Granting Ticket 과 인증자를 검사합니다. 이러한 자격 증명이 허용되는 경우 Ticket-Granting Service 는 TGT 에서 복사된 사용자 아이덴티티 정보를 포함하는 서비스 티켓을 생성합니다. 이 서비스 티켓은 다시 클라이언트로 전송됩니다.

   **참고**: Ticket-Granting Service 는 사용자에게 대상 리소스에 대한 액세스 권한이 부여되었는지 여부를 확인할 수 없습니다. Ticket-Granting Service 는 단지 사용자를 인증하고 세션 티켓을 반환합니다.

5. 클라이언트는 세션 티켓을 받은 후 대상 서버에 세션 티켓과 새 인증자를 보내 리소스에 대한 액세스를 요청합니다.
6. 서버는 티켓의 암호를 해독하고 인증자의 유효성을 검사한 다음 사용자에게 리소스에 대한 액세스 권한을 부여합니다.

## Kerberos 인증 구성

Kerberos 인증을 구성하려면 다음 단계를 수행하십시오.

1. Kerberos KDC(Key Distribution Center)를 구성합니다.
2. 정책 서버를 구성합니다.
3. 웹 서버를 구성합니다.
4. Kerberos 인증 체계를 구성합니다.
5. Windows 에서 Kerberos 외부 영역을 구성합니다.

## Kerberos Key Distribution Center 구성

Kerberos 를 사용할 경우 도메인 컨트롤러가 Kerberos 영역의 KDC(Key Distribution Center)입니다. 순수한 Windows 환경에서는 Kerberos 영역이 Windows 도메인에 해당합니다. 도메인 컨트롤러 호스트는 사용자, 서비스 계정, 자격 증명, Kerberos 티켓 서비스 및 Windows 도메인 서비스에 대한 저장소를 제공합니다.

Kerberos 인증에는 사용자에게 암호를 요구하지 않고 KDC 로 사용자를 인증할 수 있게 해 주는 keytab 파일이 필요합니다. Windows 의 경우 ktpass 명령 도구 유틸리티를 사용하여 keytab 파일을 생성하고, UNIX 의 경우 ktadd 유틸리티를 사용하여 keytab 파일을 생성하십시오.

KDC 를 구성하려면 다음 단계를 수행하십시오.

1. 워크스테이션에 로그인하는 데 사용할 사용자 계정을 생성합니다.
2. 웹 서버 호스트에 로그인하는 데 사용할 웹 서버의 서비스 계정을 생성합니다.
3. 정책 서버 호스트에 로그인하는 데 사용할 정책 서버의 서비스 계정을 생성합니다.
4. 웹 서버 계정을 웹 서버 프린서펄 이름과 연결합니다.
5. 웹 서버 호스트로 전송되는 keytab 파일을 생성합니다.
6. 정책 서버 계정을 정책 서버 프린서펄 이름과 연결합니다.
7. 또 다른 keytab 파일을 생성하고 새 keytab 파일을 정책 서버 호스트로 전송합니다.
8. 웹 서버 및 정책 서버 계정이 위임에 대해 트러스트되었는지 확인합니다.

**중요!** Kerberos 프로토콜을 사용하는 서비스의 경우 SPN(서비스 프린서펄 이름)을 service/fqdn_host@REALM_NAME 형식으로 생성해야 합니다.

## 정책 서버 구성

표준 정책 서버 구성 외에도 다음 단계를 수행하십시오.

1. 구성할 에이전트의 ACO 를 열고 다음 단계를 수행합니다.
   a. KCCExt 매개 변수에 .kcc 값을 추가합니다.
   b. HttpServicePrincipal 매개 변수에 웹 서버 프린서펄 이름 값을 추가합니다.

      예: HTTP/win2k8sps.test.com@TEST.COM
   c. SmpsServicePrincipal 매개 변수에 정책 서버 프린서펄 이름을 추가합니다.

      예: smps@winps.test.com
2. Kerberos 구성 파일 krb5.ini 를 구성하고 다음 단계 중 하나를 수행합니다.
   - Windows 의 경우 krb5.ini 파일을 시스템 루트 경로에 넣습니다.
   - UNIX 의 경우 krb5.ini 파일을 /etc/krb5/ 경로에 넣습니다.
3. KDC 에 생성된 keytab 파일, 즉 정책 서버의 보안 위치에 대한 정책 서버 프린서펄 자격 증명이 포함된 파일을 배포합니다.

**중요!** 정책 서버는 Windows 에 설치되어 있고 KDC 는 UNIX 에 배포된 경우 Ksetup 유틸리티를 사용하여 정책 서버 호스트에서 추가 구성 작업을 수행해야 합니다.

## 웹 서버 구성

웹 서버를 구성하려면 다음 단계를 수행하십시오.

1. SiteMinder Kerberos 인증 체계를 지원하는 SiteMinder 웹 에이전트를 설치합니다.
2. 트러스트된 호스트를 정책 서버에 등록하고 웹 에이전트를 구성합니다.
3. Kerberos 구성 파일 krb5.ini 를 구성하고 다음 단계를 수행합니다.
   a. Kerberos 영역(도메인)에 대한 KDC 를 구성합니다.
   b. 웹 서버 프린서펄의 자격 증명이 포함된 keytab 파일을 사용하도록 krb5.ini 를 구성합니다.
   c. krb5.ini 를 시스템 루트 경로(Windows 의 경우) 또는 /etc/krb5/(UNIX 의 경우)에 저장합니다.
4. KDC 에 생성된 keytab 파일, 즉 웹 서버의 보안 위치에 대한 웹 서버 자격 증명이 포함된 파일을 배포합니다.

**중요!** 웹 서버는 Windows 에 설치되어 있고 KDC 는 UNIX 에 배포된 경우 Ksetup 유틸리티를 사용하여 웹 서버에서 추가 구성 작업을 수행해야 합니다.

## Windows 워크스테이션 구성

Windows 워크스테이션을 구성하려면 다음 단계를 수행하십시오.

**중요!** KDC 가 UNIX 에 배포된 경우 Ksetup 유틸리티를 사용하여 워크스테이션에서 필요한 추가 구성 작업을 수행해야 합니다.

1. KDC 도메인에 Windows 워크스테이션에 대한 호스트를 추가합니다.
2. KDC 에 생성된 사용자 계정을 사용하여 호스트에 로그인합니다.
3. 자격 증명을 자동으로 전달하도록 Internet Explorer 를 구성합니다.
   a. IE 웹 브라우저의 인스턴스를 시작합니다.
   b. "인터넷 옵션" 메뉴를 선택합니다.
   c. "보안" 탭을 선택합니다.
   d. "로컬 인트라넷" 탭을 선택합니다.
   e. "사이트"를 클릭하고 세 개의 확인란을 모두 선택합니다.
   f. "고급" 탭을 선택하고 로컬 인트라넷 영역에 http://*.*domain.com* 을 추가합니다.
   g. "보안" 설정 아래에서 "사용자 지정 수준" 탭을 선택하고 "사용자 인증" 탭 아래에서 "인트라넷 영역에서만 자동으로 로그온"을 선택합니다.
   h. "인터넷 옵션"에서 "고급" 탭을 선택하고 "통합된 Windows 인증 사용(다시 시작해야 함)" 옵션을 선택합니다.
   i. 브라우저를 닫습니다.

## Kerberos 인증 체계 구성

SIteMinder 환경에서 Kerberos 인증을 지원하려면 사용자 지정 인증 체계가 필요합니다. 이 인증 체계를 Kerberos 인증으로 리소스를 보호하는 영역에 연결하십시오.

**다음 단계를 수행하십시오.**

1. 관리 UI 에 로그인합니다.

   **참고:** [set the ufi variable for your book]에서 정책 서버 개체를 생성하거나 수정할 때는 ASCII 문자를 사용하십시오. ASCII 가 아닌 문자로는 개체를 생성하거나 수정할 수 없습니다.

2. "인프라", "인증", "인증 체계"를 차례로 선택합니다.
3. "인증 체계 만들기"를 클릭합니다.
4. "인증 체계 유형" 목록에서 "사용자 지정 템플릿"을 선택합니다.

   "사용자 지정 템플릿" 설정이 표시됩니다.
5. "라이브러리" 필드에 **smauthkerberos** 를 입력합니다.
6. "매개 변수" 필드에 다음 값을 입력합니다. 다음 목록의 순서를 따라 세미콜론으로 구분하여 값을 입력합니다.
   a. 호스트 웹 서버의 이름 및 대상 필드
   b. Kerberos 도메인의 정책 서버 프린서펄 이름
   c. 사용자 프린서펄과 사용자 저장소 검색 필터 간의 매핑

   **LDAP 예 1:**
   http://win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TEST.COM;(uid=%{UID})

   **LDAP 예 2:**
   http:/win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TEST.COM;(uid=%{UID})

   **AD 예 1:**
   http://win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TEST.COM;(cn=%{UID})

   **AD 예 2:**
   http://win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TEST.COM;(cn=%{UID})

   **ODBC 예 1:**
   http://win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TEST.COM;%{UID}

   **ODBC 예 2**:
   http://win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TEST.COM;%{UID}
7. "확인"을 클릭합니다.

   Kerberos 인증 체계가 저장되고 "인증 체계" 목록에 나타납니다.

## Windows 에서 Kerberos 외부 영역 구성

Windows 워크스테이션이 UNIX 에 배포된 Kerberos KDC 를 사용하도록 하려면 Kerberos KDC 서버와 워크스테이션을 모두 구성하십시오.

Kerberos 영역에서 Windows 호스트의 호스트 프린서펄을 생성합니다. 다음 명령을 사용합니다.

```
kadmin.local: addprinc host/machine-name.dns-domain_name.
```

예를 들어 Windows 워크스테이션 이름이 W2KW 이고 Kerberos 영역 이름이 EXAMPLE.COM 인 경우 프린서펄 이름은 host/w2kw.example.com 입니다.

Kerberos 영역은 Windows 도메인이 아니므로 다음 절차를 수행하여 KDC 운영 환경을 작업 그룹의 구성원으로 구성하십시오.

1. Windows 도메인에서 호스트를 제거합니다.
2. 로컬 사용자 데이터베이스에 테스트 사용자(예: testkrb)를 추가합니다.
3. Kerberos 영역을 추가합니다.

   ```
   ksetup /SetRealm EXAMPLE.COM
   ```
4. 호스트를 다시 시작합니다.
5. KDC 를 추가합니다.

   ```
   ksetup /addkdc EXAMPLE.COM rhasmit
   ```
6. 새 암호를 설정합니다.

   ```
   ksetup /setmachpassword password
   ```

   **참고**: 여기에 사용되는 암호는 MIT KDC 에서 호스트 프린서펄 계정을 생성하는 동안 사용된 암호와 동일합니다.
7. 호스트를 다시 시작합니다.

   **참고**: 외부 KDC 및 영역 구성을 변경할 때마다 호스트를 다시 시작해야 합니다.
8. 영역 플래그를 설정합니다.

   ```
   ksetup /SetRealmFlags EXAMPLE.COM delegate
   ```
9. AddKpasswd 를 실행합니다.

   ```
   ksetup /AddKpasswd EXAMPLE.COM rhasmit
   ```

10. Ksetup 을 통해 Kerberos 프린서펄에 대한 Windows 호스트 계정 간의 계정 매핑을 정의하여 로컬 워크스테이션 계정의 싱글 사인온을 구성합니다. 예를 들면 다음과 같습니다.

    ```
    ksetup /mapuser testkrb@EXAMPLE.COM testkrb
    ksetup /mapuser * *
    ```

    두 번째 명령은 클라이언트를 동일한 이름의 로컬 계정에 매핑합니다. 인수 없이 Ksetup 을 사용하여 현재 설정을 확인합니다.

CA SiteMinder for Secure Proxy Server 가 Kerberos 인증을 지원하도록 구성되었습니다.

## Kerberos 구성 예

다음 구성에는 SiteMinder 환경에서 Kerberos 인증을 구현하는 데 필요한 keytab 파일 생성 등의 구체적인 예가 포함되어 있습니다. KDC 가 UNIX 운영 환경에 배포되어 있고 정책 서버, 웹 서버 또는 워크스테이션은 Windows 운영 환경에 있는 경우에는 추가 구성이 필요합니다.

## Windows 2008 에서 KDC 구성 예

다음에 나열된 단계에서는 SiteMinder Kerberos 인증을 지원하도록 Windows 도메인 컨트롤러를 구성하는 방법을 예로 보여 줍니다.

1. Windows dcpromo 유틸리티를 사용하여 Windows Server 를 도메인 컨트롤러(이 예에서는 test.com)로 승격합니다.
2. 도메인 기능 수준을 올립니다.
    1. "관리 도구"에서 "Active Directory 사용자 및 컴퓨터" 대화 상자를 엽니다.
    2. 대화 상자 왼쪽의 test.com 드롭다운을 마우스 오른쪽 단추로 클릭합니다.
    3. "도메인 기능 수준 올리기"를 클릭합니다.
    4. Active Directory 의 도메인 기능 수준을 올립니다.

       **중요!** 이 단계는 되돌릴 수 없습니다.
3. 사용자 계정(예: testkrb)을 생성합니다. 이 계정에 대한 암호를 제공합니다. 옵션의 선택을 취소하면 사용자가 다음에 로그온할 때 암호를 변경해야 합니다. 사용자가 로그인 권한을 갖도록 이 계정을 도메인 관리자 그룹에 추가합니다. Windows 워크스테이션은 이 계정을 사용하여 test.com 에 로그인합니다.

4. 웹 서버의 서비스 계정(예: wasrvwin2k8sps)을 생성합니다. 이 계정에 대한 암호를 생성합니다. 옵션의 선택을 취소하면 사용자가 다음에 로그온할 때 암호를 변경해야 합니다. 사용자가 로그인 권한을 갖도록 이 계정을 도메인 관리자 그룹에 추가합니다. CA SiteMinder for Secure Proxy Server 는 이 계정을 사용하여 test.com 에 로그인합니다.
5. 정책 서버의 서비스 계정(예: polsrvwinps)을 생성합니다. 이 계정에 대한 암호를 제공합니다. 옵션의 선택을 취소하면 사용자가 다음에 로그온할 때 암호를 변경해야 합니다. 사용자가 로그인 권한을 갖도록 이 계정을 도메인 관리자 그룹에 추가합니다. 정책 서버 호스트(winps)는 이 계정을 사용하여 test.com 에 로그인합니다.
6. 4~5 단계에서 생성한 서비스 계정을 사용하여 웹 서버(win2k8sps) 및 정책 서버(winps) 호스트를 test.com 도메인에 가입시킵니다.
7. Ktpass 유틸리티를 사용하여 웹 서버 계정(wasrvwin2k8sps)을 웹 서버 프린서펄 이름(HTTP/win2k8sps.test.com@TEST.COM)과 연결하고 keytab 파일을 생성합니다. 구문은 정책 서버가 Windows 에 있는지 UNIX 에 있는지에 따라 다릅니다.

   **참고**: Ktpass 명령 도구 유틸리티는 Windows 지원 도구입니다. 이 유틸리티는 MSDN 다운로드나 설치 CD 에서 설치할 수 있습니다. 항상 지원 도구의 버전을 확인하십시오. 기본 암호화 유형은 항상 RC4-HMAC 여야 합니다. 암호화 유형은 명령 프롬프트에서 kpass /?를 사용하여 확인할 수 있습니다.

정책 서버가 Windows 에 있는 경우

```
ktpass -out c:\wasrvwin2k8sps.keytab -princ HTTP/win2k8sps.test.com@TEST.COM
-ptype KRB5_NT_PRINCIPAL -mapuser wasrvwin2k8sps -pass <<password>>

Targeting domain controller: winkdc.Test.com
Using legacy password setting method
Successfully mapped HTTP/win2k8sps.test.com to wasrvwin2k8sps.
Key created.
Output keytab to c:\wasrvwin2k8sps.keytab:
Keytab version: 0x502
keysize 67 HTTP/win2k8sps.test.com@TEST.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 2
etype 0x17 (RC4-HMAC) keylength 16 (0xfd77a26f1f5d61d1fafd67a2d88784c7)
```

암호는 웹 서버의 서비스 계정을 생성하는 데 사용한 암호와 동일합니다.

정책 서버가 UNIX 에 있는 경우

```
ktpass -out d:\sol10sunone_host.keytab -princ
host/sol10sunone.test.com@TEST.COM -pass <<password>> -mapuser sol10sunone
-crypto DES-CBC-MD5 +DesOnly -ptype KRB5_NT_PRINCIPAL -kvno 3

Targeting domain controller: winkdc.test.com
Successfully mapped host/sol10sunone.test.com to sol10sunone.
Key created.
Output keytab to d:\sol10sunone_host.keytab:
Keytab version: 0x502
keysize 52 host/sol10sunone.test.com@TEST ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype
0x3 (DES-CBC-MD5) keylength 8 (0xb5a87ab5070e7f4a)
Account sol10sunone has been set for DES-only encryption.
```

8. 정책 서버 계정(polsrvwinps)을 정책 서버 프린서펄 이름(smps/winps.test.com@TEST.COM)과 연결하고 정책 서버 호스트(winps)로 보낼 다른 keytab 파일을 생성합니다.

   정책 서버가 Windows 에 있는 경우

   ```
   Ktpass -out c:\polsrvwinps.keytab -princ smps/winps.test.com@TEST.COM -ptype
   KRB5_NT_PRINCIPAL -mapuser polsrvwinps -pass <<password>>
   Targeting domain controller: winkdc.Test.com
   Using legacy password setting method
   Successfully mapped smps/winps.test.com to polsrvwinps.
   Key created.
   Output keytab to c:\polsrvwinps.keytab:
   Keytab version: 0x502
   keysize 72 smps/winps.test.com@TEST.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 2 etype
   0x17 (RC4-HMAC) keylength 16 (0xfd77a26f1f5d61d1fafd67a2d88784c7)
   ```

   암호는 정책 서버의 서비스 계정을 생성하는 데 사용한 암호와 동일합니다.

   정책 서버가 UNIX 에 있는 경우

   ```
   ktpass -out d:\sol10polsrv.keytab -princ host/sol10sunone.test.com@TEST.COM
   -pass <<password>> -mapuser sol10sunone -crypto DES-CBC-MD5 +DesOnly -ptype
   KRB5_NT_PRINCIPAL -kvno 3

   Targeting domain controller: winkdc.test.com
   Successfully mapped host/sol10sunone.test.com to sol10sunone.
   Key created.
   Output keytab to d:\sol10polserv.keytab:
   Keytab version: 0x502
   keysize 52 host/sol10sunone.test.com@TEST ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype
   0x3 (DES-CBC-MD5) keylength 8 (0xb5a87ab5070e7f4a)
   Account sol10sunone has been set for DES-only encryption.
   ```

9. 다음과 같이 웹 서버 및 정책 서버 서비스 계정이 위임에 대해 트러스트되도록 지정합니다.

   1. 서비스 계정(polsrvwinps/wasrvwin2k8sps) 속성을 마우스 오른쪽 단추로 클릭합니다.
   2. "위임" 탭을 선택합니다.
   3. 두 번째 옵션인 "모든 서비스에 대한 위임용으로 이 사용자 트러스트(Kerberos 만)"를 선택합니다.

      또는 세 번째 옵션인 "지정한 서비스에 대한 위임용으로만 이 사용자 트러스트"를 선택합니다. "Kerberos 만 사용" 옵션 단추를 선택하고 해당하는 서비스 프린서펄 이름을 추가합니다.

이제 이 도메인 컨트롤러를 SiteMinder Kerberos 인증에 사용할 수 있습니다.

## UNIX 에서 KDC 구성 예

다음 절차에서는 SiteMinder Kerberos 인증을 지원하도록 UNIX 호스트의 KDC Kerberos 영역을 구성하는 방법을 예로 보여 줍니다.

1. 필요한 경우 MIT Kerberos 를 설치합니다.
2. kdb5_util 명령을 사용하여 Kerberos 데이터베이스와 선택적 stash 파일을 생성합니다. stash 파일은 호스트 자동 부팅 시퀀스의 일부로 kadmind 및 krb5kdc 데몬을 시작하기 전에 KDC 를 자동으로 자체 인증하는 데 사용됩니다.

   stash 파일과 keytab 파일은 모두 침입에 악용될 수 있는 진입점입니다. stash 파일을 설치할 경우 이 파일은 루트에서만 읽을 수 있고 백업되지 않아야 하며 KDC 로컬 디스크에만 있어야 합니다. stash 파일이 필요하지 않으면 -s 옵션 없이 kdb5_util 을 실행하십시오.

   이 예에서는 kdc.conf 파일에 지정된 디렉터리에 다음과 같은 다섯 개의 데이터베이스 파일을 생성합니다.

   - Kerberos 데이터베이스 파일 두 개: principal.db 및 principal.ok
   - Kerberos 관리 데이터베이스 파일 한 개: principal.kadm5
   - 관리 데이터베이스 잠금 파일 한 개: principal.kadm5.lock
   - stash 파일 한 개: .k5stash

   ```
   [root@rhasmit init.d]# kdb5_util create -r EXAMPLE.COM -s
   Initializing database '/var/kerberos/krb5kdc/principal' for realm
   'EXAMPLE.COM',
   master key name 'K/M@EXAMPLE.COM'
   You will be prompted for the database Master Password.
   It is important that you NOT FORGET this password.
   Enter KDC database master key:
   Re-enter KDC database master key to verify:
   ```

3. 프린서펄(testkrb)을 생성합니다.
4. 웹 서버 호스트의 사용자 프린서펄(예: testwakrb), 호스트 프린서펄(host/win2k8sps.example.com@EXAMPLE.COM) 및 서비스 프린서펄(HTTP/win2k8sps.example.com@EXAMPLE.COM)을 생성합니다. 호스트 계정을 생성하는 데 사용된 암호는 웹 서버 호스트에서 ksetup 유틸리티를 사용할 때 지정한 암호와 동일해야 합니다.
5. 정책 서버 호스트의 사용자 프린서펄(testpskrb), 호스트 프린서펄(host/winps.example.com@EXAMPLE.COM) 및 서비스 프린서펄(smps/winps.example.com@EXAMPLE.COM)을 생성합니다. 호스트 계정을 생성하는 데 사용된 암호는 정책 서버 호스트에서 ksetup 유틸리티를 사용할 때 지정한 암호화 동일해야 합니다.

6. 다음과 같이 웹 서버 서비스 프린서펄에 대한 keytab 파일을 생성합니다.

   ```
   ktadd -k /tmp/win2k8sps.keytab HTTP/win2k8sps.example.com
   ```

7. 다음과 같이 정책 서버 서비스 프린서펄에 대한 keytab 을 생성합니다.

   ```
   ktadd -k /tmp/winps.keytab smps/winps.example.com
   ```

UNIX 호스트에 SiteMInder 에 대한 Kerberos 영역이 구성됩니다.

## UNIX 의 정책 서버에서 Kerberos 구성 예

다음 절차에서는 CA SiteMinder?Kerberos 인증을 지원하도록 UNIX 호스트의 정책 서버를 구성하는 방법을 예로 보여 줍니다.

**다음 단계를 수행하십시오.**

1. Active Directory 의 정책 서버 호스트(sol10ps)에 대한 서비스 계정을 생성하는 데 사용된 것과 동일한 암호를 사용하여 사용자(예: sol10psuser)를 생성합니다.
2. 호스트를 test.com 도메인에 추가하고 사용자 sol10psuser 로 호스트에 로그인합니다.
3. CA SiteMinder® 정책 서버를 설치하고 구성합니다.
4. 정책 저장소 디렉터리 서비스를 설치하고 구성합니다.
5. Solaris 정책 서버를 참조하는 호스트 구성 개체를 추가합니다.
6. 에이전트 구성 개체를 추가하고 다음 세 개의 새 매개 변수를 추가합니다.

| 매개 변수 | 값 |
|---|---|
| KCCExt | .kcc |
| HttpServicePrincipal | 웹 서버 프린서펄 이름을 지정합니다.<br>**예:** HTTP/win2k8sps.test.com@TEST.COM |
| SmpsServicePrincipal | 정책 서버 프린서펄 이름을 지정합니다.<br>**예:** smps@winps.test.com |

7. 사용자 디렉터리를 생성합니다.
8. 사용자 디렉터리에 사용자(예: testkrb)를 생성합니다.

9. CA SiteMinder® 관리 UI 를 사용하여 새 인증 체계를 구성합니다.
   1. 사용자 지정 템플릿을 사용하여 체계를 생성합니다.
   2. CA SiteMinder® Kerberos 인증 체계 라이브러리를 지정합니다.
   3. 매개 변수 필드를 선택하고 다음 세 값을 아래에 나온 순서대로 세미콜론으로 구분하여 지정합니다.
      - 서버 이름 및 대상 필드
      - Windows 2003 Kerberos 영역의 정책 서버 프린서펄 이름
      - 사용자 프린서펄과 LDAP 검색 필터 간의 매핑

      샘플 매개 변수 필드:

      ```
      http://sol10sunone.test.com/siteminderagent/Kerberos/creds.kcc;smps/sol10ps.test.com@TEST.COM;(uid=%{UID})
      ```
10. 정책 도메인을 구성합니다.
11. 인증 체계를 사용하여 리소스를 보호할 영역을 추가합니다.
12. 사용자 testkrb 에 대해 액세스를 허용하는 규칙 및 정책을 추가합니다.

13. Kerberos 구성 파일(krb5.ini)을 구성하고 krb5.ini 를 /etc/krb5 시스템 경로에 넣습니다.

    - Windows 2003 도메인 컨트롤러를 사용하도록 Windows 2003 Kerberos 영역(도메인)에 대한 KDC 를 구성합니다.
    - 정책 서버 프린서펄 자격 증명이 포함된 Windows 2003 KDC keytab 파일을 사용하도록 krb5.ini 를 구성합니다.

      다음의 샘플 krb5.ini 를 참조하십시오.

```
[libdefaults]
 ticket_lifetime = 24000
 default_realm=TEST.COM
 default_tgs_enctypes = des-cbc-md5
 default_tkt_enctypes = des-cbc-md5
 default_keytab_name = FILE:/etc/krb5.keytab
 dns_lookup_realm = false
 dns_lookup_kdc = false
 forwardable = true
 proxiable = true
[realms]
 TEST.COM = {
  kdc = winkdc.test.com:88
  admin_server = winkdc.test.com:749
  default_domain = test.com
 }
[domain_realm]
 .test.com=TEST.COM
  test.com=TEST.COM
```

14. ktutil 유틸리티를 사용하여 정책 서버 호스트의 호스트 프린서펄 및 서비스 프린서펄 이름이 포함된 keytab 파일(sol10ps_smps.keytab 및 sol10ps_host.keytab)을 /etc/krb5.keytab 파일에 병합합니다.

```
ktutil: rkt sol10ps_host.keytab
ktutil: wkt /etc/krb5.keytab
ktutil: q
ktutil: rkt sol10ps_smps.keytab
ktutil: wkt /etc/krb5.keytab
ktutil: q
```

15. 생성된 krb5.keytab 이 다음과 같은지 확인합니다.

```
klist -k
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
----
--------------------------------------------------------------------------
   3 host/sol10ps.test.com@TEST.COM
   3 smps/sol10ps.test.com@TEST.COM
```

16. 정책 서버의 보안 위치에 대한 호스트 및 정책 서버 프린서펄 자격 증명이 포함된 Windows 2003 KDC keytab 파일을 배포합니다.
17. 정책 서버를 시작하기 전에 다음 환경 변수가 설정되었는지 확인합니다.

```
KRB5_CONFIG=/etc/krb5/krb5.conf
```

UNIX 호스트의 정책 서버에 대해 Kerberos 인증이 구성되었습니다.

## Windows 의 정책 서버에서 Kerberos 구성 예

다음 절차에서는 CA SiteMinder?Kerberos 인증을 지원하도록 Windows 의 정책 서버를 구성하는 방법을 예로 보여 줍니다.

**참고**: 정책 서버는 Windows 에 설치되어 있고 KDC 는 UNIX 에 배포된 경우 Ksetup 유틸리티를 사용하여 정책 서버 호스트에서 필요한 추가 구성 작업을 수행해야 합니다.

**다음 단계를 수행하십시오.**

1. CA SiteMinder® 정책 서버를 설치하고 구성합니다.
2. 정책 저장소 디렉터리 서비스를 설치하고 구성합니다.
3. Windows 도메인 컨트롤러의 Active Directory 에 생성된 서비스 계정(예: polsrvwinps)을 사용하여 정책 서버 호스트에 로그인합니다.
4. 정책 서버를 참조하는 호스트 구성 개체를 추가합니다.
5. 에이전트 구성 개체를 생성하고 다음 세 개의 새 매개 변수를 추가합니다.

| 매개 변수 | 값 |
|---|---|
| KCCExt | .kcc |

| 매개 변수 | 값 |
| --- | --- |
| HttpServicePrincipal | 웹 서버 프린서펄 이름을 지정합니다.<br>**예:** HTTP/win2k8sps.test.com@TEST.COM |
| SmpsServicePrincipal | 정책 서버 프린서펄 이름을 지정합니다.<br>**예:** smps@winps.test.com |

6. 사용자 디렉터리를 생성합니다.
7. 사용자 디렉터리에 사용자(예: testkrb)를 생성합니다.
8. CA SiteMinder® 관리 UI 를 사용하여 새 인증 체계를 구성합니다.
    1. 사용자 지정 템플릿을 사용하여 체계를 생성합니다.
    2. CA SiteMinder® Kerberos 인증 체계 라이브러리를 지정합니다.
    3. 매개 변수 필드를 선택하고 다음 세 값을 아래에 나온 순서대로 세미콜론으로 구분하여 지정합니다.
        - 서버 이름 및 대상 필드
        - Windows 2003 Kerberos 영역의 정책 서버 프린서펄 이름
        - 사용자 프린서펄과 LDAP 검색 필터 간의 매핑

        샘플 매개 변수 필드:

        ```
        http://win2k8sps.test.com/siteminderagent/Kerberos/creds.kcc;smps/winps.test.com@TES.COM;(uid=%{UID})
        ```
9. 정책 도메인을 구성합니다.
10. 인증 체계를 사용하여 리소스를 보호할 영역을 추가합니다.
11. 사용자 testkrb 에 대해 액세스를 허용하는 규칙 및 정책을 추가합니다.
12. Kerberos 구성 파일(krb5.ini)을 구성하고 krb5.ini 를 Windows 시스템 루트 경로에 넣습니다.
    - Windows 2003 도메인 컨트롤러를 사용하도록 Windows 2003 Kerberos 영역(도메인)에 대한 KDC 를 구성합니다.
    - 정책 서버 프린서펄 자격 증명이 포함된 Windows 2003 KDC keytab 파일을 사용하도록 krb5.ini 를 구성합니다.

다음의 샘플 krb5.ini 를 참조하십시오.

```
[libdefaults]
default_realm = TEST.COM
default_keytab_name = C:\WINDOWS\krb5.keytab
default_tkt_enctypes = rc4-hmac des-cbc-md5
default_tgs_enctypes = rc4-hmac des-cbc-md5
[realms]
TEST.COM = {
kdc = winkdc.test.com:88
default_domain = test.com
}
[domain_realm]
.test.com = TEST.COM
```

13. 정책 서버의 보안 위치에 대한 정책 서버 프린서펄 자격 증명이 포함된 Windows KDC keytab 파일을 배포합니다.

Windows 호스트의 정책 서버에 대해 Kerberos 인증이 구성되었습니다.

# 제 16 장: CA Wily Introscope 를 사용한 데이터 모니터링

이 섹션은 다음 항목을 포함하고 있습니다.



## CA Wily Introscope 를 사용한 데이터 모니터링

조직에서 이미 CA Wily Introscope 를 사용하고 있는 경우 SPS 의 건전성을 모니터링할 수 있습니다. Wily EPAgent 를 사용하여 Tomcat 서버의 다음 통계를 모니터링할 수 있습니다.

- 다음 CA SiteMinder for Secure Proxy Server 구성 요소 각각의 평균 응답 시간
  - 세션 검색
  - Java 웹 에이전트
  - Post 에이전트 세션 작성기
  - 프록시 규칙 필터
  - 누들 서블릿
  - HTTP 클라이언트
- 각 백엔드 서버의 평균 응답 시간
- CA SiteMinder for Secure Proxy Server 요청 대기 시간
- 각 프록시 규칙의 적중 횟수
- CA SiteMinder for Secure Proxy Server 에이전트 프레임워크 인스턴스의 건전성 데이터

데이터 모니터링 섹션의 형식은 다음과 같습니다.

```
<Server>
.
.
monitor_data_buffer_size="1000"
.
.
</Server>
    <metric-reporter name="Wily Metric Reporter">
    enabled="yes"
    class="com.ca.proxy.monitor.wily.WilyMetricReporter"
    endpoint="tcp://localhost:8886"
</metric-reporter>
```

***monitor_data_buffer_size***

수집된 통계를 Wily 로 전송하기 전에 저장하기 위해 CA SiteMinder for Secure Proxy Server 가 유지 관리하는 버퍼의 크기를 지정합니다.

**기본값**: 1000

***metric-reporter name***

메트릭 리포터의 이름을 지정합니다. 이 이름을 사용하여 메트릭 오류를 디버깅할 수 있습니다.

***enabled***

데이터 모니터링 기능의 상태를 지정합니다. SPS 의 건전성을 모니터링하려면 이 값을 yes 로 설정하십시오. SPS 의 건전성을 모니터링하지 않으려면 이 값을 no 로 설정하십시오.

**기본값:** no

***endpoint***

EPAgent 의 구성 상세 정보를 지정합니다. SPS 와 통신하기 전에 EPAgent 를 시작하십시오. endpoint 매개 변수의 형식은 다음과 같습니다.

*protocol*://*hostname_of_EPAgent*:*port*

***protocol***

EPAgent 가 사용하는 통신 프로토콜을 지정합니다.

***hostname_EPAgent***

EPAgent 가 설치된 컴퓨터의 호스트 이름을 지정합니다.

**기본값**: localhost

***port***

EPAgent 가 SPS 와 통신하는 데 사용하는 포트 번호를 지정합니다. TCP 프로토콜을 사용할 경우 EPAgent 에 구성된 네트워크 데이터 포트를 입력하십시오. HTTP 프로토콜을 사용할 경우에는 EPAgent 에 구성된 HTTP 포트 번호를 입력하십시오.

## 데이터 모니터링 사용

데이터 모니터링을 사용하도록 설정하려면 다음 단계를 수행하십시오.

**참고**: CA Wily EPAgent 구성에 대한 자세한 내용은 *CA Wily Introscope Environment Performance Agent Guide*(CA Wily Introscope 환경 성능 에이전트 안내서)를 참조하십시오.

1. SPS 에서 메트릭을 수집하도록 CA Wily EPAgent 를 구성합니다.
2. 다음 단계를 수행하여 server.conf 파일을 구성합니다.
   a. server.conf 파일을 열고 metric-reporter 섹션으로 이동합니다.
   b. 다음 값을 설정합니다.

      ```
      enabled=yes
      ```

   c. (선택 사항) CA SiteMinder for Secure Proxy Server 에 구성된 에이전트 인스턴스 메트릭을 모니터링하려면 다음 값을 설정합니다.

      ```
      enablemonitoring=yes
      ```

   d. 변경 내용을 저장합니다.
3. SPS 에 구성된 각 웹 에이전트에 대해 ACO 를 구성합니다.
4. CA Wily EPAgent 를 시작합니다.
5. SPS 를 다시 시작합니다.

# OneView 모니터를 사용하여 웹 에이전트 모니터링

CA SiteMinder?OneView 모니터는 관리자가 웹 에이전트를 분석하고 세부적으로 조정하는 데 사용할 수 있도록 캐시 통계 및 기타 정보를 정책 서버에 보고합니다. 다음 매개 변수를 사용하여 CA SiteMinder?OneView 모니터를 제어합니다.

**EnableMonitoring**

CA SiteMinder® 웹 에이전트가 정책 서버에 모니터링 정보를 보낼지 여부를 지정합니다.

**기본값:** No

웹 에이전트에서 CA SiteMinder® OneView 모니터를 사용하게 하려면 EnableMonitoring 매개 변수를 yes 로 설정하십시오.

**참고:** 자세한 내용은 정책 서버 설명서를 참조하십시오.

# 제 17 장: 에이전트를 위한 운영 체제 조정

이 섹션은 다음 항목을 포함하고 있습니다.

# 공유 메모리 세그먼트 조정

Solaris 시스템에 Apache 기반 에이전트를 설치하는 경우 에이전트가 올바로 작동하려면 운영 환경의 공유 메모리 설정을 조정해야 합니다. 운영 환경의 공유 메모리 세그먼트를 늘리면 에이전트 성능이 향상됩니다. 공유 메모리 세그먼트를 제어하는 변수는 운영 환경의 사양 파일에 정의됩니다.

**참고:** Linux 운영 환경에서 공유 메모리 세그먼트를 조정해야 할 경우가 있습니다. 공유 메모리 세그먼트에 대한 내용 및 이러한 세그먼트를 조정하는 방법을 보려면 해당 운영 환경의 설명서를 참조하십시오.

**다음 단계를 수행하십시오.**

1. 운영 환경에 맞는 절차를 따릅니다.
   - **Solaris:** 선택한 편집기를 사용하여 /etc/system 파일을 엽니다.
2. 다음 방법 중 *하나*를 사용하여 공유 메모리 변수를 수정합니다.
   - Solaris: 아래 목록에 나오는 변수를 추가하고 예제의 권장 설정을 사용하여 변수를 구성합니다. 다음 구문을 사용하십시오.

     ```
     set shmsys:shminfo_shmmax=33554432
     ```

     **shmsys:shminfo_shmmax**

     최대 공유 메모리 세그먼트 크기를 지정합니다. 이 값은 에이전트 리소스 및 세션 캐시의 최대 크기를 제어합니다.

     **참고:** 필요한 메모리 세그먼트의 크기를 계산하려면 각 캐시의 항목당 4 KB 를 할당하거나 OneView 모니터에서 캐시 사용 통계를 확인해야 합니다. OneView 모니터 사용에 대한 자세한 내용은 *웹 에이전트 구성* 안내서를 참조하십시오.

     **예:** 33554432(32 MB) - 사용량이 많아 큰 캐시가 필요한 사이트의 경우

     **shmsys:shminfo_shmmin**

     (Solaris 의 경우 필요하지 않음) 최소 공유 메모리 세그먼트 크기입니다. 에이전트 리소스 및 세션 캐시의 최소 크기를 제어합니다.

**shmsys:shminfo_shmmni**

시스템 전체에서 동시에 존재할 수 있는 최대 공유 메모리 세그먼트의 수를 지정합니다.

**예:** (Solaris 9 제외) 해당 없음

**예:** (Solaris 9) 200

**shmsys:shminfo_shmseg**

(Solaris 9 의 경우 필요하지 않음) 각 프로세스에 대해 최대 공유 메모리 세그먼트 수를 지정합니다.

**예:** 24

**semsys:seminfo_semmni**

세마포 식별자 수를 지정합니다. 시스템에서 실행하는 에이전트의 모든 인스턴스에 대해 11 을 사용합니다.

**예:** (Solaris 9 제외) 100

**예:** (Solaris 9) 200

**semsys:seminfo_semmns**

시스템의 세마포 수를 지정합니다. 시스템에서 실행하는 에이전트의 모든 인스턴스에 대해 10 을 사용합니다.

**예:** (Solaris 9) 100

**예:** (Solaris 9) 400

**semsys:seminfo_semmnu**

undo 기능을 사용하는 프로세스의 수를 지정합니다. 최적의 성능을 위해서는 semmnu 값을 시스템에서 한 번에 실행되는 Apache 자식 프로세스의 수보다 큰 값으로 설정합니다. Apache 기반 서버의 경우 maxclients 설정보다 200 이상 큰 값을 사용하십시오.

**예:** (Solaris 9) 200

3. 변경 내용을 저장하고 파일 또는 유틸리티를 종료합니다.
4. 시스템을 재부팅합니다.
5. 다음 명령을 실행하여 변경 내용을 확인합니다.

```
$ sysdef -i
```

# Solaris 10 리소스 컨트롤을 조정하는 방법

프로젝트 수준에서 리소스 컨트롤을 조정하여 에이전트 성능을 향상시킬 수 있습니다.

**참고:** 자세한 내용은 Solaris 설명서를 참조하십시오.

Solaris 10 에서 리소스 컨트롤을 조정하려면 다음을 수행하십시오.

1. 웹 에이전트를 실행하는 사용자 계정과 연결된 프로젝트를 확인합니다.
2. 해당 프로젝트에 대해 다음 리소스 컨트롤의 설정을 늘립니다.

**project.max-shm-ids**

프로젝트의 최대 공유 메모리 ID 수를 지정합니다.

**project.max-sem-ids**

프로젝트의 최대 세마포 ID 수를 지정합니다.

**project.max-msg-ids**

프로젝트의 최대 메시지 큐 ID 수를 지정합니다.

**project.max-shm-memory**

프로젝트에 허용되는 전체 공유 메모리의 크기를 지정합니다.

**process.max-sem-nsems**

각 세마포 집합에 허용되는 최대 세마포 수를 지정합니다.

**process.max-sem-ops**

각 semop 에 허용되는 최대 세마포 작업 수를 지정합니다.

**process.max-msg-messages**

메시지 큐의 최대 메시지 수를 지정합니다.

**process.max-msg-qbytes**

메시지 큐의 최대 메시지 바이트 수를 지정합니다.

# 제 18 장: SPS API

이 섹션은 다음 항목을 포함하고 있습니다.



## 세션 체계 API

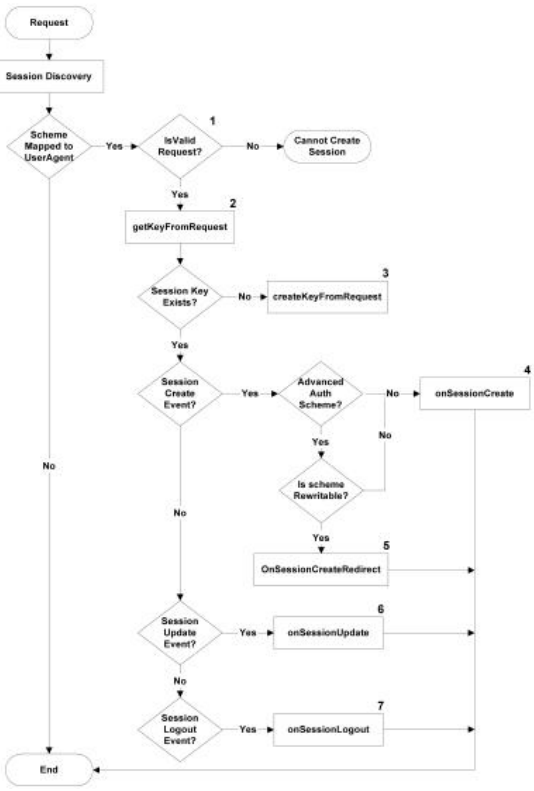
CA SiteMinder for Secure Proxy Server 는 사용자 지정 세션 체계를 개발할 수 있게 해 주는 Java API 를 지원합니다. 이러한 체계는 CA SiteMinder for Secure Proxy Server 에 구성된 각 가상 호스트의 사용자 에이전트 유형에 할당될 수 있습니다.

### 세션 체계 API 처리 개요

CA SiteMinder for Secure Proxy Server 는 여러 메서드를 처리하여 일반적인 사용자 세션을 설정, 유지 관리 및 종료합니다. 세션을 처리하는 단계 중에는 체계가 다시 쓰기 가능한지 여부를 확인하는 단계가 포함됩니다. 다시 쓰기 가능한 체계에서는 URL 을 수정할 수 있습니다. 다시 쓰기 가능한 체계의 예로는 요청된 URL 을 토큰이 포함되도록 다시 쓰는 단계가 요청 처리 과정에 포함된 단순 URL 다시 쓰기 세션 체계를 들 수 있습니다.

다시 쓰기 가능한 세션 체계를 구현하려면 다시 쓰기 가능한 세션 체계 (페이지 302)에 설명된 대로 다시 쓰기 가능한 인터페이스를 구현해야 합니다.

다음 그림에서는 세션 체계 API 메서드의 프로세스 흐름을 보여 줍니다.




그림에 나타난 메서드는 다음과 같습니다.

1. **isValidRequest** - 유효한 요청을 구성하는 조건을 결정하고 확인하려면 사용자 지정 세션 체계에서 이 메서드를 구현해야 합니다.
2. **getKeyFromRequest** - 유효한 요청에서 키를 추출하려면 이 메서드를 구현해야 합니다. 키가 없으면 createKeyFromRequest 메서드가 호출됩니다.
3. **createKeyFromRequest** - 새 세션에 대한 키 생성을 트리거하려면 이 메서드를 구현해야 합니다.
4. **onSessionCreate** - 세션 생성 시 사용 중인 세션 체계가 다시 쓰기 가능하지 않으면 이 메서드가 호출됩니다. 이 메서드는 새 세션이 시작될 때 트리거되는 코드를 사용하여 구현할 수 있습니다.
5. **onSessionCreateRedirect** - 세션 생성 시 체계가 다시 쓰기 가능하면 이 메서드가 호출됩니다. 다시 쓰기 가능한 세션 체계의 경우 새 세션이 시작될 때 호출되는 코드를 사용하여 이 메서드를 구현할 수 있습니다.

6. **onSessionUpdate** - 세션은 세션 도중 새 요청이 있을 때마다 업데이트됩니다. 이 메서드는 각 세션 업데이트 중에 호출됩니다. 세션 업데이트 중 트리거되는 코드를 추가하여 이 메서드를 구현할 수 있습니다.
7. **onSessionLogout** - 이 메서드는 세션이 종료될 때 호출됩니다. 사용자 세션이 종료될 때 실행되는 코드를 사용하여 이 메서드를 구현할 수 있습니다.

## 세션 체계 API 클래스 파일

CA SiteMinder for Secure Proxy Server 세션 체계 API 는 *sps_home*/Tomcat/server/lib/proxycore.jar 에 포함된 세션 체계 추상 클래스를 사용합니다.

## 세션 체계 API 의 생성자

사용자 지정 세션 체계의 생성자는 다음과 같이 구성되어야 합니다.

```
public IPAddrSessionScheme(String name, boolean
                                acceptFlag, Hashtable props) {
               // Always call the parent constructor for proper
               // initialization of the scheme

               super(name,acceptFlag,props);
```

설정은 다음과 같습니다.

**name**

사용자 지정 세션 체계 클래스와 연결된 server.conf 파일 내의 이름으로 채워지는 문자열입니다.

**acceptFlag**

사용자 지정 세션 체계가 SiteMinder 의 SMSESSION 쿠키를 허용할지 여부를 결정하는 부울 값입니다.

**props**

server.conf 파일에 지정된 세션 체계가 필요로 하는 다른 모든 속성의 이름/값 쌍으로 구성된 목록입니다.

## 세션 체계 API 메서드

세션 체계 API 클래스는 다음 메서드로 구성되어 있습니다.

| 반환 값 | 메서드 | 참고 |
|---|---|---|
| 부울 | acceptsCookies() | server.conf 파일에 있는 세션 체계의 accepts_smsession_cookies 매개 변수에서 acceptFlag 의 값을 검색하고, 이 체계가 SiteMinder SMSESSION 쿠키를 지원하는지 여부를 나타내는 값을 반환합니다. |
| abstract java.lang.String | createKeyFromRequest(HttpServletRequest req) | 코드를 실행하여 요청에서 새 세션 키를 생성하는 데 필요한 값을 검색합니다. |
| abstract java.lang.String | getKeyFromRequest(HttpServletRequest req) | 요청에서 세션 키를 검색합니다. |
| java.lang.String | getName() | server.conf 파일에 정의된 사용자 지정 세션 체계의 이름을 검색합니다. |
| abstract Boolean | isValidRequest(HttpServletRequest req) | 이 세션 체계에 대한 요청이 유효한지 평가합니다. |
| abstract int | onSessionCreate(java.lang.String id, HttpServletRequest req, HttpServletResponse resp) | 세션 생성 시 사용할 수 있는 후크입니다. |
| java.lang.String | onSessionCreateRedirect(java.lang.String id, java.lang.String url, HttpServletRequest req, HttpServletResponse resp) | 다시 쓰기 가능한 체계의 경우 세션 생성 시 사용할 수 있는 후크입니다. |
| abstract void | onSessionLogOut(HttpServletRequest req, HttpServletResponse resp) | 세션 종료(로그아웃) 시 사용할 수 있는 후크입니다. |
| abstract void | onSessionUpdate(HttpServletRequest req, HttpServletResponse resp) | 세션 업데이트 시 사용할 수 있는 후크입니다. 이 메서드는 내부 전용입니다. |
| 정적 부울 | usingDefaultSessionScheme(HttpServlet Request req) | 요청이 기본 세션 체계를 사용하고 있음을 인식하기 위한 도우미 메서드입니다. |

# 사용자 지정 세션 체계 구현

사용자 지정 세션 체계를 구현하려면 다음 절차를 따르십시오.

**다음 단계를 수행하십시오.**

1. IP 주소 세션 체계에서 IP 주소 세션 체계에 대한 샘플 코드를 검토합니다.
2. 세션 체계에 대한 소스 코드를 작성합니다.
3. 다시 쓰기 가능한 세션 체계를 생성하려면 다시 쓰기 가능한 인터페이스를 구현합니다.
4. 시스템 CLASSPATH 에 다음 내용이 포함되어 있는지 확인합니다.
   - 세션 체계 API 가 포함된 proxycore.jar
   - JDK 버전 1.6.0_30 이상의 jar 파일
   - *sps_home*/Tomcat/lib jar 파일
5. 세션 체계를 컴파일합니다.
6. 다음 단계 중 *하나*를 수행합니다.
   - 사용자 지정 세션 체계를 포함하는 .jar 파일을 생성한 다음 이 .jar 파일을 *sps_home*/Tomcat/lib 디렉터리에 복사합니다.
   - *sps_home*/Tomcat/lib 디렉터리의 jar 에 사용자 지정 세션 체계에 대한 클래스 파일을 추가하고 CA SiteMinder for Secure Proxy Server server.conf 파일에서 체계를 구성합니다.
7. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

### server.conf 파일에서 사용자 지정 세션 체계 구성

사용자 지정 세션 체계에 대한 코드를 컴파일할 때는 CA SiteMinder for Secure Proxy Server server.conf 파일에서 세션 체계를 구성해야 합니다. 세션 체계를 구성하려면 이 파일에 SessionScheme 요소를 추가하십시오. 예를 들면 다음과 같습니다.

```
<SessionScheme name="custom_scheme">
                class="com.netegrity.proxy.session.CustomScheme"
                accepts_smsession_cookies="false"
                property1="value1"
                property2="value2"
</SessionScheme>
```

또한 사용자 에이전트 유형을 구성한 경우 세션 체계를 적절한 사용자 에이전트 유형에 매핑할 수 있습니다.

**추가 정보:**

기본 가상 호스트에 대한 세션 체계 매핑 (페이지 150)

## 다시 쓰기 가능한 세션 체계 구성

세션 체계에 사용자가 요청한 URL 을 수정할 수 있는 기능이 있어야 하는 경우 다시 쓰기 가능한 인터페이스를 구현해야 합니다. 예를 들어 이 인터페이스는 단순 URL 다시 쓰기 체계에서 세션 체계가 URL 요청의 끝에 토큰을 추가할 수 있도록 하는 데 사용됩니다.

### 다시 쓰기 가능한 인터페이스 구현

다시 쓰기 가능한 인터페이스를 구현하려는 경우 다음 메서드를 사용할 수 있습니다.

| 반환 값 | 메서드 | 설명 |
|---|---|---|
| public string | rewrite(String url, String id, HttpServletRequest req) | 세션 식별자를 포함하도록 요청된 URL 을 다시 씁니다. |

| 반환 값 | 메서드 | 설명 |
| --- | --- | --- |
| public string | onSessionCreateRedirect(String id, String url, HttpServletRequest req, HttpServletResponse resp) | 세션 생성 시 리디렉션을 통해 콜백을 제공합니다. 일반적으로 대상 URL 과 요청된 URL 이 다른 양식 기반 인증과 함께 사용됩니다. 예를 들어 인증 체계가 URL 을 수정하거나 쿠키를 추가할 수 있습니다. |

다시 쓰기 가능한 인터페이스뿐 아니라 사용자 지정 세션 체계에서도 다음 메서드를 구현해야 합니다.

| 반환 값 | 메서드 | 설명 |
| --- | --- | --- |
| protected void | setRequestURI(HttpServletRequest req, String uri) | 체계가 요청 URI 를 수정할 수 있습니다. |
| protected void | setRequestPathInfo(HttpServletRequest req, String pathInfo) | 체계가 요청의 경로 정보를 수정할 수 있습니다. |

## IP 주소 세션 체계 사용

기본 CA SiteMinder for Secure Proxy Server 설치에는 IP 주소 세션 체계가 포함되어 있습니다. 이 체계는 클라이언트의 IP 주소를 사용하여 세션을 매핑합니다. 사용자가 요청을 하면 CA SiteMinder for Secure Proxy Server 는 HTTP 헤더에서 클라이언트의 IP 주소를 검색하고 이 IP 주소를 사용하여 클라이언트 세션에 맞는 세션 키를 생성합니다.

IP 주소 세션 체계는 세션 체계 API 를 사용하여 생성되었습니다. 이 체계의 소스 코드는 *sps_home*\secure-proxy\proxy-engine\examples\sessionschemes 디렉터리에서 찾을 수 있습니다.

**참고:** 샘플 세션 체계 파일에서 백슬래시(\) 문자는 해당 행이 계속되어야 하지만 문서의 공간 제약으로 인해 불가피하게 중단되었음을 나타냅니다.

### IP 주소 세션 체계를 구현하려면

1. 다음과 같이 server.conf 파일에 <SessionScheme> 섹션을 추가합니다.

```
<SessionScheme name="ip_address">
    class="com.netegrity.proxy.session.IPAddrSessionScheme"
    accepts_smsession_cookies="false"
    allowed_proxied_addresses="true"
</SessionScheme>
```

   지시문은 다음과 같습니다.

   **class**

   이 지시문은 IP 주소 세션 체계를 처리하는 Java 클래스를 지정합니다. CA SiteMinder for Secure Proxy Server 와 함께 설치된 기본 IP 주소 세션 체계를 사용하려면 이 값을 수정하면 안 됩니다.

   **기본값:** com.netegrity.proxy.session.IPAddrSessionScheme

   **accepts_smsession_cookies**

   이 세션 체계에서 SiteMinder smsession 쿠키가 지원되지 않음을 나타냅니다. IP 주소 체계를 사용하며 쿠키를 사용하지 않는 세션을 확인하려면 이 지시문의 값을 변경하면 안 됩니다.

   **기본값:** false

   **allowed_proxied_addresses**

   SessionScheme.isValidRequest 호출을 사용하여 요청의 유효성을 검사할지 여부를 나타냅니다. 프록시 주소를 사용할 수 있도록 하려면 이 값을 true 로 설정하십시오. VIA HTTP 헤더 변수가 있는지 여부를 확인하는 데 isValidRequest 메서드를 사용하려면 기본값인 false 를 그대로 사용하십시오. 이 변수가 있으면 CA SiteMinder for Secure Proxy Server 는 주소가 프록시된 주소임을 확인하고 요청을 차단합니다.

   **기본값:** true

2. 세션 체계를 server.conf 파일에 있는 가상 호스트용 사용자 에이전트 하나 이상에 매핑합니다.
3. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

## 세션 저장소 API

CA SiteMinder for Secure Proxy Server 는 세션 토큰에서 SiteMinder 세션으로의 매핑을 저장합니다. 이 정보에 액세스하려면 SessionStorageAPI 를 사용합니다.

SessionStorageAPI 는 다음 기능을 제공합니다.

**세션 생성**

새 세션을 생성할 수 있습니다.

**세션 업데이트 또는 동기화**

SiteMinder 세션 정보를 업데이트할 수 있습니다.

**세션 검색**

올바른 세션 키가 함께 제공될 경우 세션 정보를 검색할 수 있습니다.

**명시적 세션 제거**

특정 세션 키를 사용하여 세션을 제거할 수 있습니다.

**세션 만료**

만료된 모든 세션을 제거할 수 있습니다.

# 필터 API 개요

사용자 지정 필터는 고객의 요구에 의해 정의된 필터입니다. CA SiteMinder for Secure Proxy Server 는 요청을 백엔드 서버로 전달하기 전에 사용자 지정 필터를 사용하여 요청을 조작할 뿐 아니라 백엔드 서버가 사용자 클라이언트로 전송하는 응답도 조작합니다.

CA SiteMinder for Secure Proxy Server 는 각 요청에 대해 단일 사용자 지정 필터나 사용자 지정 필터 그룹을 처리할 수 있습니다. 사용자 지정 필터 그룹을 만들면 CA SiteMinder for Secure Proxy Server 는 해당 사용자 지정 필터 그룹에 속한 모든 필터를 한 체인에서 처리합니다.

필터 API 를 사용하여 생성된 전처리 필터와 후처리 필터의 소스 코드를 볼 수 있습니다. 이러한 샘플은 다음 디렉터리에서 찾을 수 있습니다.

*sps_home*/proxy-engine/examples/filters

**참고:** 코드 샘플에서 백슬래시(\) 문자는 해당 행이 계속되어야 하지만 문서의 공간 제약으로 인해 불가피하게 중단되었음을 나타냅니다.

**추가 정보**

프록시 규칙에 사용자 지정 필터 연결 (페이지 307)

## SPS 가 사용자 지정 필터를 처리하는 방식

CA SiteMinder for Secure Proxy Server 에는 요청의 프록시 단계에 전처리 및 후처리를 삽입하기 위한 API 가 포함되어 있습니다.

표준 CA SiteMinder for Secure Proxy Server 트랜잭션에서의 프로세스는 다음과 같습니다.

1. 사용자가 리소스를 요청합니다.
2. CA SiteMinder for Secure Proxy Server 가 해당 프록시 규칙을 검사하고 요청을 전달할 위치를 결정합니다(인증 및 권한 부여 성공 후).
3. 대상 서버가 요청된 리소스를 CA SiteMinder for Secure Proxy Server 로 전송하고 CA SiteMinder for Secure Proxy Server 는 사용자에게 리소스를 전달합니다.

개발자는 필터 API 를 사용하여 요청이 대상 서버로 전달되기 전에(앞의 프로세스 중 2 단계) 처리를 삽입하거나, 대상 서버에서 CA SiteMinder for Secure Proxy Server 로 응답이 반환된 후(앞의 프로세스 중 3 단계) 리소스가 사용자에게 전달되기 전에 처리를 삽입할 수 있습니다.

## 프록시 규칙에 사용자 지정 필터 연결

요청 또는 응답을 수신하면 CA SiteMinder for Secure Proxy Server 는 프록시 규칙을 읽고 연결된 필터를 처리합니다. server.conf 파일에 선언된 사용자 지정 필터 또는 사용자 지정 그룹 필터는 프록시 규칙과 연결되어야 합니다. 프록시 규칙에 사용자 지정 필터 또는 사용자 지정 그룹 필터를 연결하려면 <*install dir*>/secure-proxy/proxy-engine/conf 에 있는 proxyrules.xml 을 연 다음 해당 필터를 실행할 규칙에 맞게 proxyrules.xml 파일을 편집하십시오.

예를 들면 다음과 같습니다.

<nete:forward filter="*your filter name or your groupfiltername*">http://FQDN$0</nete:forward>

## 필터 API 클래스 파일

CA SiteMinder for Secure Proxy Server 필터 API 는 *sps_home*/Tomcat/server/lib/proxyrt.jar 에 포함된 프록시 필터 클래스를 사용합니다.

## ProxyFilter 인터페이스

ProxyFilter 인터페이스는 프록시 필터에 의해 구현되는 인터페이스를 정의합니다. 그러나 ProxyFilter 인터페이스를 구현하는 대신 BaseProxyFilter 추상 구현을 확장하는 것이 좋습니다.

ProxyFilter 인터페이스는 다음 메서드로 구성되어 있습니다.

| 반환 값 | 메서드 |
|---|---|
| void | doFilter(ProxyRequest prequest, ProxyResponse presponse)<br>필터링을 수행합니다.<br>매개 변수:<br>request - 프록시 요청 데이터입니다.<br>response - 프록시 응답 데이터입니다.<br>throw:<br>ProxyFilterException - 필터링 처리 중 문제가 발생할 경우에 throw 됩니다. |

| 반환 값 | 메서드 |
| --- | --- |
| ProxyFilterConfig | getFilterConfig()<br>이 필터의 ProxyFilterConfig 개체를 반환합니다. ProxyFilterConfig 개체는 이 필터를 초기화한 개체입니다. |
| void | init(ProxyFilterConfig config)<br>필요한 초기화를 수행하기 위해 필터가 생성될 때 호출됩니다.<br>매개 변수:<br>config - 필터의 구성 및 초기화 매개 변수를 포함하는 ProxyFilterConfig 개체입니다.<br>throw:<br>ProxyFilterException - 이 필터의 초기화 중 문제가 발생할 경우에 throw 됩니다. |

## BaseProxyFilter 추상 구현

필터 API 에는 ProxyFilter 를 생성하기 위한 하위 클래스로 구현될 수 있는 프록시 필터의 추상 구현인 BaseProxyFilter 가 포함되어 있습니다.

**참고:** ProxyFilter 인터페이스를 구현하는 대신 BaseProxyFilter 추상 구현을 확장하는 것이 좋습니다.

BaseProxyFilter 의 하위 클래스는 다음 메서드 중 하나 이상을 재정의해야 합니다.

- doPreFilter
- doPostFilter
- doFilter(권장되지 않음)

다음 표에 나열된 대로 BaseProxyFilter 는 필터 초기화를 포함하며 CA SiteMinder for Secure Proxy Server 트랜잭션에 사용자의 필터를 삽입하기 위한 전처리 후크와 후처리 후크를 구분합니다.

| 반환 값 | 메서드 |
| --- | --- |
| Void | doFilter(ProxyRequest prequest, ProxyResponse presponse) throws ProxyFilterException<br>이 구현은 요청 처리 상태를 확인하여 처리가 인바운드 상태에 있으면 doPreFilter 를 호출하고 아웃바운드 상태에 있으면 doPostFilter 를 호출합니다. 필터가 호출될 때 처리 상태는 이 두 상태 중 하나일 수만 있습니다.<br>지정 방법:<br>ProxyFilter 인터페이스에서 doFilter 실행<br>매개 변수:<br>request - 프록시 요청 데이터입니다.<br>response - 프록시 응답 데이터입니다.<br>throw:<br>ProxyFilterException - 필터링 처리 중 문제가 발생할 경우에 throw 됩니다. |
| Void | doPreFilter(ProxyRequest prequest, ProxyResponse presponse) throws ProxyFilterException<br>사전 필터링을 수행합니다. 요청이 대상 서버로 전송되기 전에 필터링 태스크를 수행하려면 이 메서드를 재정의하십시오.<br>매개 변수:<br>request - 프록시 요청 데이터입니다.<br>response - 프록시 응답 데이터입니다.<br>throw:<br>ProxyFilterException - 필터링 처리 중 문제가 발생할 경우에 throw 됩니다. |

| 반환 값 | 메서드 |
| --- | --- |
| Void | doPostFilter(ProxyRequest prequest, ProxyResponse presponse) throws ProxyFilterException<br>사후 필터링을 수행합니다. 대상 서버로부터 응답이 수신된 후 필터링 태스크를 수행하려면 이 메서드를 재정의하십시오.<br>매개 변수:<br>request - 프록시 요청 데이터입니다.<br>response - 프록시 응답 데이터입니다.<br>throw:<br>ProxyFilterException - 필터링 처리 중 문제가 발생할 경우에 throw 됩니다. |
| ProxyFilterConfig | getFilterConfig()<br>이 필터의 ProxyFilterConfig 개체를 반환합니다.<br>지정 방법:<br>ProxyFilter 인터페이스에서 getFilterConfig 실행<br>반환:<br>ProxyFilterConfig - 이 필터를 초기화한 ProxyFilterConfig 개체입니다. |

| 반환 값 | 메서드 |
| --- | --- |
| Void | init(ProxyFilterConfig config) throws ProxyFilterException<br>필요한 초기화를 수행하기 위해 필터가 생성될 때 호출됩니다.<br>**참고:** 이 메서드를 재정의하려면 첫 번째 문에서 부모 init 메서드 "super.init(config);"를 호출해야 합니다.<br>지정 방법:<br>ProxyFilter 인터페이스에서 init 실행<br>매개 변수:<br>config - 필터의 구성 및 초기화 매개 변수를 포함하는 ProxyFilterConfig 개체입니다.<br>throw:<br>ProxyFilterException - 이 필터의 초기화 중 문제가 발생할 경우에 throw 됩니다. |

## ProxyFilterConfig 인터페이스

필터에 사용할 수 있는 구성 데이터에 대한 인터페이스를 정의합니다. 이 인터페이스는 다음 메서드로 구성되어 있습니다.

| 반환 값 | 메서드 |
| --- | --- |
| java.lang.String | getFilterName()<br>이 필터의 이름을 반환합니다. |
| java.lang.String | getInitParameter(java.lang.String name)<br>명명된 초기화 매개 변수의 값을 포함하는 문자열을 반환하거나, 매개 변수가 없는 경우 null 을 반환합니다.<br>매개 변수:<br>name - 초기화 매개 변수의 이름을 지정하는 문자열입니다. |

| 반환 값 | 메서드 |
| --- | --- |
| java.util.Enumeration | getInitParameterNames()<br>필터의 초기화 매개 변수 이름을 문자열 개체의 열거형으로 반환하거나, 필터에 초기화 매개 변수가 없는 경우 빈 열거형을 반환합니다. |

## ProxyResponse 인터페이스

프록시 클라이언트에 반환되는 HTTP 응답 정보에 액세스할 수 있게 해 주는 인터페이스를 정의합니다. 이 인터페이스는 다음 메서드로 구성되어 있습니다.

| 반환 값 | 메서드 |
| --- | --- |
| void | addHeader(java.lang.String name, java.lang.String value)<br>지정된 이름 및 값을 갖는 헤더를 추가합니다. 이 메서드를 사용할 경우 응답 헤더에 여러 값을 포함할 수 있습니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다.<br>value - 헤더 값을 지정하는 문자열입니다. |
| byte[] | getContent()<br>프록시 요청에 대한 응답 내용의 바이트 배열을 반환합니다. 이 내용은 프록시 클라이언트에 반환되는 내용입니다. |
| java.lang.String | getHeader(java.lang.String name)<br>지정된 헤더의 값을 문자열로 반환합니다. 헤더가 없으면 이 메서드는 null 을 반환합니다. 헤더 이름은 대/소문자를 구분하지 않습니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다. |

| 반환 값 | 메서드 |
| --- | --- |
| java.util.Enumeration | getHeaderNames()<br>모든 헤더 이름의 열거형을 반환합니다. 헤더가 없으면 이 메서드는 빈 열거형을 반환합니다. |
| int | getStatusCode()<br>프록시 요청에 대한 응답의 HTTP 응답 상태 코드를 반환합니다. |
| java.lang.String | removeHeader(java.lang.String name)<br>지정된 헤더를 제거하고, 제거된 헤더의 값을 문자열로 반환합니다. 헤더가 없으면 이 메서드는 null 을 반환합니다. 헤더 이름은 대/소문자를 구분하지 않습니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다. |
| void | setContent(byte[] content)<br>프록시 요청에 대한 응답 내용을 설정합니다. 이 내용은 프록시 클라이언트에 반환되는 내용을 덮어씁니다.<br>매개 변수:<br>content - 내용을 포함하는 바이트 배열입니다. |
| void | setHeader(java.lang.String name, java.lang.String value)<br>지정된 이름 및 값을 갖는 헤더를 설정합니다. 같은 이름의 헤더가 이미 있으면 기존 헤더를 덮어씁니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다.<br>value - 헤더 값을 지정하는 문자열입니다. |

## ProxyFilterException 클래스

ProxyFilterException 클래스는 필터 작업에 어려움이 있을 경우 필터가 throw 할 수 있는 일반적인 예외를 정의합니다.

| 생성자 서명 | 설명 |
| --- | --- |
| ProxyFilterException() | 새 ProxyFilterException 을 구성합니다. |
| ProxyFilterException(java.lang.String message) | 지정된 메시지를 포함하여 새 ProxyFilterException 을 구성합니다.<br>매개 변수:<br>message - 예외 메시지입니다. |
| ProxyFilterException(java.lang.String message, java.lang.Throwable rootCause) | 지정된 메시지와 근본 원인을 포함하여 새 ProxyFilterException 을 구성합니다.<br>매개 변수:<br>message - 예외 메시지입니다.<br>rootCause - 이 예외의 발생 원인이 되는 예외입니다. |
| ProxyFilterException(java.lang.Throwable rootCause) | 지정된 메시지와 근본 원인을 포함하여 새 ProxyFilterException 을 구성합니다.<br>매개 변수:<br>rootCause - 이 예외의 발생 원인이 되는 예외입니다. |

## ProxyRequest 인터페이스

프록시가 보내는 HTTP 요청 정보에 액세스할 수 있게 해 주는 인터페이스를 정의합니다. 이 인터페이스는 다음 메서드로 구성되어 있습니다.

| 반환 값 | 메서드 |
|---|---|
| java.lang.String | getHeader(java.lang.String name)<br>지정된 헤더의 값을 문자열로 반환합니다. 헤더가 없으면 이 메서드는 null 을 반환합니다. 헤더 이름은 대/소문자를 구분하지 않습니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다. |
| java.util.Enumeration | getHeaderNames()<br>모든 헤더 이름의 열거형을 반환합니다. 헤더가 없으면 이 메서드는 빈 열거형을 반환합니다. |
| javax.servlet.http.HttpServletRequest | getOriginalRequest()<br>프록시에 대한 원래 HttpServletRequest 를 반환합니다. |
| java.lang.String | getSessionKey()<br>세션 키의 값을 문자열로 반환합니다. 세션 키를 사용할 수 없으면 이 메서드는 null 을 반환합니다. 세션 키는 쿠키를 사용하지 않는 체계를 사용할 때 콘텐츠의 URL 을 다시 쓰는 데 사용할 수 있습니다.<br>**참고:** SessionScheme 은 키를 생성하고 이를 SessionScheme.DEFAULT_SESSION_KEY_NAME 이라는 특성에 저장하는 작업을 수행합니다. |
| java.lang.String | getTargetQueryString()<br>프록시가 대상 URL 과 함께 사용할 쿼리 문자열을 반환합니다. 쿼리 문자열은 원래 요청이나 프록시 규칙을 통해 정의된 새 요청에서 가져올 수 있습니다. URL 에 쿼리 문자열이 없는 경우 이 메서드는 null 을 반환합니다. |

| 반환 값 | 메서드 |
| --- | --- |
| java.lang.String | getTargetURL()<br>프록시가 프록시 규칙에 정의된 대로 요청을 하는 데 사용할 URL 을 반환합니다. 이 URL 에는 쿼리 문자열 매개 변수가 포함되지 않습니다 |
| 부울 | isInbound()<br>요청 처리 상태를 나타내는 부울 값을 반환합니다. 요청이 대상 서버로 전달되지 않은 경우에는 true 가 반환됩니다. 요청이 전송되었고 응답이 수신되었으면 false 가 반환됩니다. |
| 부울 | isOutbound()<br>요청 처리 상태를 나타내는 부울 값을 반환합니다. 요청이 대상 서버로 전달되지 않은 경우에는 false 가 반환됩니다. 요청이 전송되었고 응답이 수신되었으면 true 가 반환됩니다. |
| java.lang.String | removeHeader(java.lang.String name)<br>지정된 헤더의 값을 제거한 후 문자열로 반환합니다. 헤더가 없으면 이 메서드는 null 을 반환합니다. 헤더 이름은 대/소문자를 구분하지 않습니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다. |
| void | setHeader(java.lang.String name, java.lang.String value)<br>지정된 이름 및 값을 갖는 헤더를 설정합니다. 같은 이름의 헤더가 이미 있으면 기존 헤더를 덮어씁니다.<br>매개 변수:<br>name - 헤더 이름을 지정하는 문자열입니다.<br>value - 헤더 값을 지정하는 문자열입니다. |
| byte[] | getContent()<br>요청 POST 데이터의 내용을 포함하는 바이트 배열을 반환합니다. 이 내용은 백엔드 서버로 전송되는 내용입니다. |

| 반환 값 | 메서드 |
| --- | --- |
| void | setContent(byte[] content)<br>프록시 요청에 대한 POST 데이터의 내용을 설정합니다.<br>이 내용은 백엔드 서버로 전송되는 내용을 덮어씁니다.<br>매개 변수:<br>content - 요청 POST 데이터를 포함하는 바이트 배열입니다. |

## 필터 구현

세션 키를 사용하는 필터는 세션 체계에 따라 키를 정의합니다. 필터에 세션 키를 사용할 수 있도록 하려면 SessionScheme.DEFAULT_SESSION_KEY_NAME 으로 키가 지정된 특성을 설정하여 키가 createKeyFromRequest(..) 콜백에 의해 생성되고 후속 요청에서 세션 체계 API 의 getKeyFromRequest(..) 콜백에 의해 수신될 때 키 값을 유지하도록 해야 합니다.

세션 키를 생성하는 기본 제공 세션 체계는 다음과 같습니다.

- 미니 쿠키
- 단순 URL 다시 쓰기

**다음 단계를 수행하십시오.**

1. 필터 예제에서 필터에 대한 샘플 코드를 검토합니다.
2. 필터에 대한 소스 코드를 작성합니다.
3. 시스템 CLASSPATH 에 다음 내용이 포함되어 있는지 확인합니다.
   - 필터 API 가 포함된 proxyrt.jar
   - JDK 버전 1.6.0_30 이상의 jar 파일
   - *sps_home*/Tomcat/lib jar 파일
4. 필터를 컴파일합니다.

5. 다음 단계 중 *하나*를 수행합니다.
   - 필터를 포함하는 .jar 파일을 생성한 다음 이 .jar 파일을 *sps_home*/Tomcat/lib 디렉터리에 복사합니다.
   - 필터에 대한 클래스 파일을 *sps_home*/Tomcat/lib 디렉터리의 jar 에 추가합니다.
6. CA SiteMinder for Secure Proxy Server server.conf 파일을 구성합니다.
7. 필터를 구현하는 데 필요한 규칙에 맞게 proxyrules.xml 파일을 편집합니다. 예를 들면 다음과 같습니다.
   <nete:forward filter="your filter name">http://FQDN$0</nete:forward>
8. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

## 필터 API 예제

CA SiteMinder for Secure Proxy Server 설치에는 전처리 필터와 후처리 필터용 샘플 소스 파일이 포함되어 있습니다. 이러한 샘플은 모두 BaseProxyFilter 추상 구현을 사용합니다. 예제 필터에 대한 전체 설명은 필터 예제를 참조하십시오.

## 필터를 사용하여 요청된 페이지에서 절대 링크 다시 쓰기

필터 API 의 가장 일반적인 용도 중 하나는 사용자가 CA SiteMinder for Secure Proxy Server 를 통해 요청한 페이지에서 절대 링크 다시 쓰기를 지원하는 것입니다. 절대 링크가 CA SiteMinder for Secure Proxy Server 에서 올바르게 처리되도록 하려면 필터 API 를 사용하여 CA SiteMinder for Secure Proxy Server 요청을 기반으로 사용자에게 반환되는 리소스에 포함된 절대 링크를 적절하게 대체해야 합니다.

# 제 19 장: 문제 해결

이 섹션은 다음 항목을 포함하고 있습니다.

# SSL 구성 후 브라우저에 팝업 창이 표시됨

### 증상

SSL 구성 후 인터넷 브라우저를 사용하여 CA SiteMinder® SPS 에 액세스하면 팝업 창이 표시됩니다.

### 해결 방법

SSL 구성 후 약한 암호화를 사용하는 인터넷 브라우저를 사용하여 처음 CA SiteMinder® SPS 에 액세스하면 팝업 창이 표시됩니다. 팝업을 허용하고 CA SiteMinder® SPS 에 계속 액세스할 수 있습니다. 팝업이 표시되지 않도록 하려면 다음 단계 중 하나를 수행하십시오.

- 강력한 암호화 알고리즘을 지원하는 Internet Explorer 8.0, Chrome, Mozilla Firefox 등의 인터넷 브라우저를 사용하십시오.
- X509 기반 인증 체계를 사용하지 않는 경우 다음 단계를 수행하십시오.
  a. <install_path>/httpd\conf\extra\로 이동합니다.
  b. httpd-ssl.conf 파일을 엽니다.
  c. SSLVerifyClient 로 이동하여 값을 none 으로 바꿉니다.
  d. 변경 내용을 저장합니다.
  e. CA SiteMinder® SPS 를 다시 시작합니다.

# UNIX 시스템에서 Apache 를 시작할 수 없음

### 증상

UNIX 시스템에서 CA SiteMinder for Secure Proxy Server 가 실행 중일 때 Apache 서버를 시작할 수 없습니다. Apache 로그 파일에 다음과 같은 오류 메시지가 나타납니다.

```
"Invalid argument: setgid: unable to set group id to ..." (잘못된 인수: setgid: 그룹 ID 를 설정할 수 없습니다...)
```

### 해결 방법

UNIX 시스템의 Run-As-User 그룹이 Apache 구성 파일(httpd.conf)에 지정된 그룹에 해당하지 않으면 이 오류가 발생합니다. 이 오류가 발생하면 Apache httpd.conf 파일에서 Group 지시문을 편집하십시오.

### Group 지시문을 편집하려면

1. Group 지시문 앞의 주석 기호(#)를 제거합니다.
2. Run-As-User 가 속하는 그룹을 지정합니다.
3. CA SiteMinder for Secure Proxy Server 시작 명령(sps-ctl start 또는 startssl)을 다시 실행합니다.

# 영어가 아닌 입력 문자에 정크 문자가 포함됨

**UNIX/Linux 에 해당**

### 증상

영어가 아닌 일부 입력 문자가 콘솔 창에 올바로 표시되지 않습니다.

### 해결 방법

콘솔 창의 터미널 설정을 확인하십시오. 콘솔이 입력 문자의 높은(8) 비트를 삭제하지 않는지 확인하십시오. 다음 명령을 실행합니다.

```
stty –istrip
```

# 페더레이션 웹 서비스 오류를 로깅할 수 없음

### 증상

페더레이션 웹 서비스 오류가 로깅되지 않습니다.

### 해결 방법

페더레이션 웹 서비스의 오류를 로깅하려면 LoggerConfig.properties 파일에서 AffWebServices 및 FWSTrace 로그 매개 변수를 사용하도록 설정하십시오.

### 다음 단계를 수행하십시오.

1. LoggerConfig.properties 파일을 엽니다.

   **기본 경로**:
   *sps_home*/secure-proxy/Tomcat/webapps/affwebservices/WEB-INF/classes/LoggerConfig.properties

2. 다음 매개 변수를 구성합니다.

   ```
   LoggingOn=Y

   TracingOn=Y
   ```

3. 변경 내용을 저장합니다.

# 모든 요청에 대해 DNS 가 캐시됨

### 증상

CA SiteMinder for Secure Proxy Server 가 서버의 DNS 이름 조회 설정을 캐시하는 것을 원하지 않습니다.

### 해결 방법

CA SiteMinder for Secure Proxy Server 는 기본적으로 서버의 DNS 설정을 캐시하도록 구성되어 있습니다. 이 기본 동작을 변경하려면 java.security 파일의 networkaddress.ttl 설정을 조정하십시오.

### 다음 단계를 수행하십시오.

1. NETE_SPS_JAVA_HOME\jre\lib\security 디렉터리로 이동합니다.
2. java.security 파일을 엽니다.
3. networkaddress.cache.ttl 매개 변수를 양의 정수로 설정합니다. 예: networkaddress.cache.ttl=2

   **networkaddress.ttl**

   CA SiteMinder for Secure Proxy Server 가 성공한 DNS 이름 조회를 캐시하는 기간(초)을 지정합니다. 양수를 입력합니다. 음수 값을 입력하면 CA SiteMinder for Secure Proxy Server 가 DNS 설정을 캐시합니다.

   **기본값:** -1

# 리소스 요청 실패

### 증상

CA SiteMinder for Secure Proxy Server 가 리소스 요청을 처리하지 못했습니다.

### 해결 방법

이 오류를 해결하려면 다음 로그 파일에서 오류 상세 정보를 확인하십시오.

- spsagent 및 spsagenttrace 로그
- Apache 액세스 및 오류 로그
- httpclient.log
- server.log
- mod_jk.log

로그 파일에 로그가 없는 경우 로그 파일에 로깅하는 기능이 사용되도록 설정되었는지 확인하십시오.

### 추가 정보:

spsagent 로그 구성 (페이지 325)
SPSAgentTrace 로그 구성 (페이지 326)
mod_jk.log 파일 구성 (페이지 327)
httpclient.log 파일 구성 (페이지 327)

## spsagent 로그 구성

CA SiteMinder for Secure Proxy Server 는 spsagent 로그에 프록시 엔진과 관련된 오류를 로깅합니다. 정책 서버의 로컬 구성 파일 또는 ACO 에는 오류 로깅을 사용하도록 설정하고 로깅 옵션을 결정하는 매개 변수가 포함되어 있습니다.

**다음 단계를 수행하십시오.**

1. 정책 서버에서 CA SiteMinder for Secure Proxy Server 의 ACO 를 엽니다.
2. LogFile 매개 변수의 값을 yes 로 설정합니다.

   **참고**: 로컬 구성 파일에서 이 매개 변수의 값을 yes 로 설정하면 정책 서버에 정의된 로깅 설정이 무시됩니다.
3. 다음 매개 변수를 완성합니다.

   **LogFileName**

   로그 파일의 이름을 포함한 전체 경로를 지정합니다.

   **LogAppend**

   새 로그 정보를 기존 로그 파일의 끝에 추가합니다. 이 매개 변수가 no 로 설정되면 로깅을 호출할 때마다 전체 로그 파일이 다시 작성됩니다.

   **LogFileSize**

   로그 파일 크기 제한(MB)을 지정합니다. 현재 로그 파일의 크기가 이 값이 되면 새 로그 파일이 생성됩니다.

   **LogLocalTime**

   로그에 GMT(그리니치 표준시)를 사용할지 아니면 로컬 시간을 사용할지를 지정합니다. GMT 를 사용하려면 이 설정을 no 로 변경하십시오. 이 매개 변수가 없으면 기본 설정이 사용됩니다.
4. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

오류 로그가 구성되었습니다.

## SPSAgentTrace 로그 구성

파일의 크기 및 형식을 제어하도록 추적 로그를 구성할 수 있습니다. 추적 로깅이 구성되면 추적 로그 파일의 내용을 별도로 결정합니다. 이렇게 하면 추적 로그 파일 자체의 매개 변수를 변경하지 않고 추적 로그에 포함되는 정보 유형을 언제든지 변경할 수 있습니다.

**다음 단계를 수행하십시오.**

1. SecureProxyTrace.conf 파일을 찾아서 복제합니다.
2. 에이전트 구성 개체 또는 로컬 구성 파일을 엽니다.
3. TraceFile 매개 변수를 yes 로 설정합니다.

   **참고**: 로컬 구성 파일에서 이 매개 변수의 값을 yes 로 설정하면 정책 서버에 정의된 로깅 설정이 무시됩니다.
4. 다음 매개 변수를 구성합니다.

   **TraceFileName**

   추적 로그 파일의 전체 경로를 지정합니다.

   **TraceConfigFile**

   모니터링할 구성 요소 및 이벤트를 결정하는 SecureProxyTrace.conf 구성 파일의 위치를 지정합니다.

   **TraceAppend**

   로깅이 호출될 때마다 전체 파일을 다시 쓰지 않고 기존 로그 파일의 끝에 새 로깅 정보를 추가해야 하는지 여부를 지정합니다.

   **TraceFormat**

   추적 파일에서 메시지를 표시하는 방식을 지정합니다.

   **TraceDelimiter**

   추적 파일에서 필드를 구분하는 사용자 지정 문자를 지정합니다.

   **TraceFileSize**

   추적 파일의 최대 크기(MB)를 지정합니다. 이 제한에 도달하면 CA SiteMinder for Secure Proxy Server 가 새 파일을 생성합니다.

   **LogLocalTime**

   로그에 GMT(그리니치 표준시)를 사용할지 아니면 로컬 시간을 사용할지를 지정합니다. GMT 를 사용하려면 이 설정을 no 로 변경하십시오. 이 매개 변수가 없으면 기본 설정이 사용됩니다.
5. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

## mod_jk.log 파일 구성

CA SiteMinder for Secure Proxy Server 는 Apache 와 프록시 엔진 간의 모든 통신 메시지를 mod_jk.log 파일에 로깅합니다. 기본적으로 이 파일에서 로깅이 사용되도록 설정되며, 이 로그 파일은 *sps_home*\secure-proxy\httpd\logs\mod_jk.log 에 있습니다.

**다음 단계를 수행하십시오.**

1. httpd.conf 파일을 엽니다.

   **기본 경로**: *sps_home*\secure-proxy\httpd\conf

2. 사용 가능한 매개 변수를 필요한 대로 수정합니다.

   **참고**: httpd.conf 파일 및 mod_jk.log 파일 구성에 대한 자세한 내용은 Apache 설명서를 참조하십시오.

3. JkRequestLogFormat 이 %w %V %T %m %h %p %U %s 형식으로 설정되어 있는지 확인합니다.
4. 변경 내용을 저장합니다.
5. CA SiteMinder for Secure Proxy Server 를 다시 시작합니다.

## httpclient.log 파일 구성

디버깅 용도로만 httpclient.log 가 사용되도록 설정할 수 있습니다. 기본적으로 httpclient.log 파일은 *sps_home*\secure-proxy\proxy-engine\logs 에 있습니다.

**다음 단계를 수행하십시오.**

1. server.conf 파일을 엽니다.
2. httpclientlog 가 yes 로 설정되어 있는지 확인합니다.
3. httpclientlogging.properties 파일을 엽니다.

   **기본 경로**: *sps_home*\Tomcat\properties 디렉터리

4. 사용 가능한 매개 변수를 필요한 대로 수정합니다.

   **참고**: httpclientlogging.properties 파일 구성에 대한 자세한 내용은 Apache 설명서를 참조하십시오.

5. 변경 내용을 저장합니다.

# 설치 프로그램에 경고가 표시됨

**UNIX 에 해당**

**증상**

CA SiteMinder for Secure Proxy Server 설치 시 일부 파일을 수동으로 구성해야 한다는 경고가 설치 마법사에 표시됩니다.

**해결 방법**

루트 권한이 없는 경우 CA SiteMinder for Secure Proxy Server 를 설치할 수는 있지만 일부 설치 단계가 자동 설치 프로세스로 완료되지 않습니다. 수동으로 구성해야 할 파일을 확인할 수 있도록 설치 마법사에 경고가 표시됩니다.

**참고:** SSL 사용 서버에는 비 루트 설치가 권장되지 않습니다. 비 루트 설치의 경우 루트 권한이 있는 다른 사용자가 키 및 인증서에 액세스할 수 있으므로 보안 수준이 낮습니다.

# SPS 서버를 시작할 수 없음

**증상**

CA SiteMinder for Secure Proxy Server 서버를 시작할 수 없습니다.

**해결 방법**

서버를 시작할 수 없는 경우 다음 정보를 사용하십시오.

- *sps_home*/secure-proxy/httpd/conf/httpd.conf 의 ServerName 지시문이 서버 이름에 해당하는지 확인하십시오.
- 다음 명령 중 하나를 실행하여 서버가 이미 실행 중이 아닌지 확인하십시오.
  - ps -ax|grep http on BSD compatible systems
  - ps -elf|grep http on System V release 4 compatible systems

  실행 결과 프로세스 목록이 나타나면 실행 중인 서버를 중지한 후에 새 서버를 시작하십시오.
- *sps_home*/secure-proxy/httpd/logs 디렉터리에 있는 로그 파일을 확인하십시오.

- httpd.conf 파일의 SSLCertificateFile 및 SSLCertificateKeyFile 지시문이 해당 인증서 및 키 파일을 가리키는지 확인하십시오. httpd.conf 파일은 *sps_home*/secure-proxy/httpd/conf 디렉터리에 있습니다.
- IP 기반이 아닌 가상 호스트를 사용하고 있는지 여부를 확인하십시오. SSL 을 사용하려면 IP 기반 가상 호스트가 필요합니다.
- SPS 의 기본 포트에서 실행 중인 다른 서버가 없는지 확인하십시오. 기본 포트는 httpd.conf 파일에 지정되어 있습니다.
- SSL 을 사용하는 경우 서버를 시작하기 전에 키와 인증서를 생성했어야 합니다. 그렇지 않으면 오류가 발생합니다.

# 브라우저를 사용하여 SPS 에 액세스할 수 없음

### 증상

브라우저를 사용하여 CA SiteMinder for Secure Proxy Server 에 액세스하는 데 문제가 있습니다.

### 해결 방법

브라우저를 사용하여 CA SiteMinder for Secure Proxy Server 에 액세스하려면

- nslookup <servername> 명령을 사용하여 DNS 가 서버 이름을 인식하는지 확인하거나, ping <servername> 명령을 사용하여 서버를 'ping'해 보십시오.
- SSL 을 사용하지 않고 서버를 실행한 후 웹 사이트에 액세스하여 문제가 키 또는 인증서 파일과 관련된 것인지 확인하십시오. SSL 을 사용하지 않고 서버를 시작하려면 *sps_home*\secure-proxy\proxy engine 디렉터리에서 ./sps-ctl start 를 실행하십시오.
- 웹 서버의 포트 80 및 443(또는 사용자가 지정한 기본 포트 외의 포트)에 텔넷으로 연결해 보십시오. 루트가 아닌 사용자로 설치한 경우에는 포트 8080 및 8443 에 연결해 보십시오.

# 가상 호스트 구성 문제

**증상**

가상 호스트를 구성하는 데 문제가 있습니다.

**해결 방법**

다음 위치에서 가상 호스트 구성에 대한 정보를 참조하십시오.

http://httpd.apache.org/docs-2.0/vhosts/

# 가상 호스트 구성 실패

**증상**

가상 호스트의 구성에 실패합니다.

**해결 방법**

가상 호스트를 구성하는 방법에 대한 자세한 내용은 www.apache.org 를 참조하십시오.

# SPS 가 요청을 전달하지 않음

**증상**

리소스 액세스 시 404 파일을 찾을 수 없음 오류가 브라우저에 표시되지만 웹 에이전트 로그에는 해당 작업이 기록되지 않습니다.

**해결 방법**

server.conf 파일에서 가상 호스트의 이름과 IP 주소를 확인하십시오.

# SharePoint 페이지 액세스 오류

### 증상

SPS 를 통해 SharePoint 페이지에 액세스할 경우 proxyrule.xml 파일에 설정된 모드(전달 또는 리디렉션)에 관계없이 CA SiteMinder for Secure Proxy Server 가 항상 대체 액세스 매핑 연결 매개 변수를 표시합니다.

### 해결 방법

이 문제를 해결하려면 다음 단계를 수행하십시오.

1. SharePoint 서버에서 "중앙 관리", "작업", "대체 액세스 매핑"으로 이동합니다. "대체 액세스 매핑"에 "기본 영역의 내부 URL"과 "공용 URL"이 포함되어 있는지 확인합니다.
2. http://<SPS Host>:port 로 설정된 "공용 URL"과 "기본 영역"을 사용하여 내부 URL 을 하나 추가합니다.
3. http://<SharePoint Host>:port 로 설정된 "공용 URL"과 "기본 영역"을 사용하여 내부 URL 을 하나 더 추가합니다.
4. 3 단계에서 생성한 인트라넷 영역의 항목을 편집하고 "공용 URL"을 http://<SPS Host>:port 로 지정합니다.
5. CA SiteMinder for Secure Proxy Server proxyrule.xml 파일에서 백엔드는 CA SiteMinder for Secure Proxy Server 호스트를 가리키는 공용 URL 이 포함된 내부 URL 입니다. 예를 들면 다음과 같습니다.

```
<!--Proxy Rules-->
<nete:proxyrules xmlns:nete="http://ww.ca.com/">
      <nete:forward>http://SharePointServer with public URL as SPS
           host:port$0</nete:forward>
</nete:proxyrules>
```