SiteMinder

웹 에이전트 구성 안내서

12.52 SP1

ca technologies

도움말 시스템 및 전자적으로 배포된 매체를 포함하는 본 문서(이하 "문서")는 최종 사용자에게 정보를 제공하기 위한 것이며, CA 는 언제든지 본 문서를 변경 또는 철회할 수 있습니다. 본 문서는 CA 의 재산적 정보이며 CA 의 사전 서면 동의 없이 본 문서의 전체 혹은 일부를 복사, 전송, 재생, 공개, 수정 또는 복제할 수 없습니다.

CA 소프트웨어의 라이선스를 허여받은 사용자들은 본인 및 그 직원들의 해당 소프트웨어와 관련된 내부적인 사용을 위해 1 부의 문서 사본을 만들 수 있습니다. 단, 이 경우 복사본에는 CA 저작권 표시 및 문구 일체가 기재되어야 합니다.

본건 문서의 사본 인쇄 또는 제작 권한은 해당 소프트웨어의 라이선스가 전체 효력을 가지고 유효한 상태를 유지하는 기간으로 제한됩니다. 어떤 사유로 인해 라이선스가 종료되는 경우, 귀하는 서면으로 문서의 전체 또는 일부 복사본이 CA 에 반환되거나 파기되었음을 입증할 책임이 있습니다.

CA 는 관련법의 허용 범위 내에서, 상품성에 대한 묵시적 보증, 특정 목적에 대한 적합성 또는 권리 위반 보호를 비롯하여(이에 제한되지 않음) 어떤 종류의 보증 없이 본 문서를 "있는 그대로" 제공합니다. CA 는 본 시스템의 사용으로 인해 발생되는 직, 간접 손실이나 손해(수익의 손실, 사업 중단, 영업권 또는 데이터 손실 포함)에 대해서는 (상기 손실이나 손해에 대해 사전에 명시적으로 통지를 받은 경우라 하더라도) 귀하나 제 3 자에게 책임을 지지 않습니다.

본건 문서에 언급된 모든 소프트웨어 제품의 사용 조건은 해당 라이선스 계약을 따르며 어떠한 경우에도 이 문서에서 언급된 조건에 의해 라이선스 계약이 수정되지 않습니다.

본 문서는 CA 에서 제작되었습니다.

본 시스템은 "제한적 권리"와 함께 제공됩니다. 미합중국 정부에 의한 사용, 복제 또는 공개는 연방조달규정(FAR) 제 12.212 조, 제 52.227-14 조, 제 52.227-19(c)(1)호 - 제(2)호 및 국방연방구매규정(DFARS) 제 252.227-7014(b)(3)호 또는 해당하는 경우 후속 조항에 명시된 제한 사항을 따릅니다.

Copyright © 2014 CA. All rights reserved. 이 문서에서 언급된 모든 상표, 상호, 서비스 표시 및 로고는 각 해당 회사의 소유입니다.

CA Technologies 제품 참조

이 문서는 다음 CA Technologies 제품을 참조합니다 :

- SiteMinder
- CA Introscope®(이전의 CA Wily Introscope)
- CA IdentityMinder™(이전의 CA Identity Manager)
- eTrust SOA Security Manager(이전의 CA SOA Security Manager)

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

SiteMinder 의 이전 릴리스에서 발견된 문제점으로 인해 다음과 같은 내용이 12.52 SP1 설명서에서 업데이트되었습니다.

- 유효한 세션이 여러 개 있는 경우 영역 시간 만료 후 재인증되지 않도록 방지 (페이지 133) - 새 매개 변수 compatRealmtimeouts 추가됨 (CQ160965, 158664, STAR 문제 # 21025754:01.)
- 잘못된 URL 문자 지정 (페이지 90) - BadUrlChars 의 기본값이 "기본값: 비활성화됨(모든 문자가 허용됨)"으로 수정됨(CQ: 155294 및 STAR 문제 20933042-1)
- 기본 CSS 문자 집합 재정의 (페이지 86) - 명확성을 높이기 위해 재작성됨 (CQ: 155294 및 STAR 문제 20933042-1)
- 웹 응용 프로그램 클라이언트 응답 (페이지 116) - 웹 응용 프로그램 클라이언트 응답 기능이 기본 인증 체계에서 작동하지 않음을 알려 주는 참고가 추가되었습니다. CQ 177736 및 STAR 문제 21467829-1 을 해결합니다.
- 오류 처리를 설정하는 방법 (페이지 165) - CQ170498, STAR Issue # 21389742-01 을 해결하기 위해 URL 예제가 수정되었습니다.
- 세션 쿠키 생성 또는 업데이트 방지 (페이지 125) - 새 ACO 매개 변수가 추가되었습니다.
- 쿠키를 포함하는 서버 응답의 캐싱 방지 (페이지 333) - CQ171158, CQ171396, STAR Issue # 21407131:01 을 해결하기 위해 새 ACO 매개 변수가 추가되었습니다.
- 유효한 세션이 여러 개 있는 경우 영역 시간 만료 후 재인증되지 않도록 방지 (페이지 133) - 새 매개 변수 compatRealmtimeouts 추가됨 (CQ160965, 158664, STAR 문제 # 21025754:01)
- HTTP 헤더에 레거시 변수 사용 (페이지 161) - Apache 2.4.x 웹 서버에 대한 참고가 추가됨 (CQ178440 및 STAR 문제 # 21545697)

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

설명서에서만 변경된 내용

- 교차 사이트 스크립팅 공격에 대해 J2EE 응용 프로그램 보호 (페이지 85)
  - DisallowUTF8NonCanonical 의 기본값이 수정되었습니다. 설명서는 12.52 SP1 에서 업데이트되었습니다. 이 변경 사항은 CQ 183701 및 STAR 21715698-1 을 해결합니다.

목차

제 1 장: 웹 에이전트

이 섹션은 다음 항목을 포함하고 있습니다.

웹 에이전트의 리소스 보호 방식

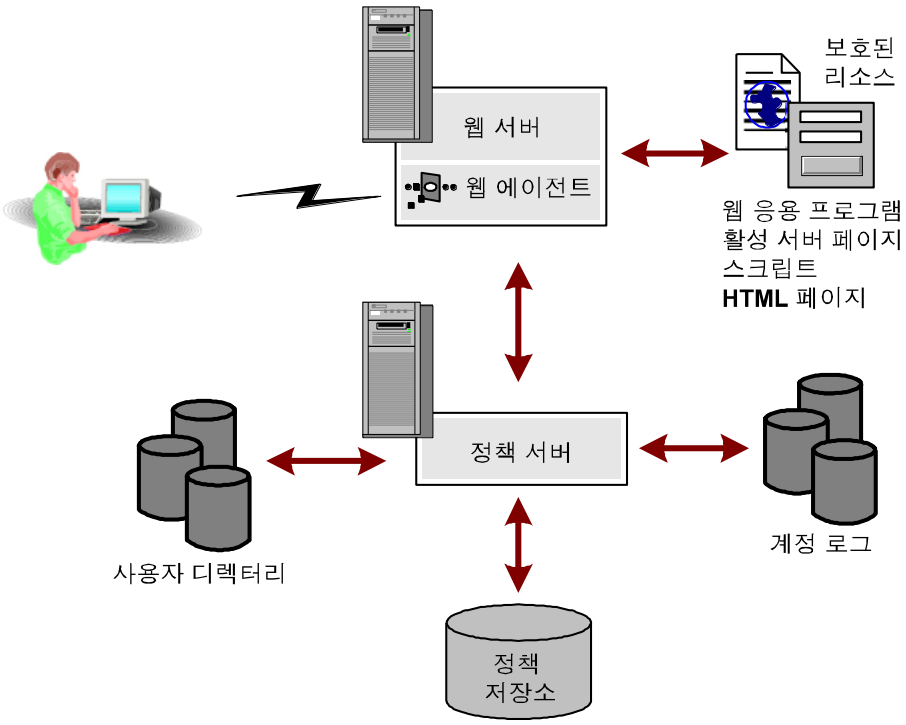
SiteMinder 웹 에이전트는 URL 로 식별할 수 있는 리소스에 대한 액세스를 제어하는 소프트웨어 구성 요소입니다. 웹 에이전트는 웹 서버에 있으며 리소스에 대한 요청을 가로챈 후 해당 리소스가 SiteMinder 에서 보호되는지 여부를 확인합니다. 그런 다음 웹 에이전트는 정책 서버와 상호 작용하여 보호된 웹 서버 리소스에 대한 액세스를 요청하는 사용자를 인증하고 권한을 부여합니다.

웹 에이전트는 다음 태스크를 수행합니다.

- 보호된 리소스에 대한 액세스 요청을 가로채고 정책 서버와 연동하여 사용자에게 액세스 권한을 부여할지 여부를 결정합니다.
- 콘텐츠를 사용자에게 제공할 방법(정책 기반 맞춤 설정)과 액세스 권한을 제공할 방법을 나타내는 정보를 웹 응용 프로그램에 제공합니다.
- 사용자가 빠르고 안전하게 정보에 액세스할 수 있도록 보장합니다. 웹 에이전트는 사용자 액세스 권한에 대한 상황별 정보를 세션 캐시에 저장합니다. 캐시 설정을 수정하여 성능을 최적화할 수 있습니다.
- 단일 쿠키 도메인 또는 여러 쿠키 도메인의 모든 웹 서버에서 사용자가 다시 인증할 필요가 없도록 SSO(싱글 사인온)를 지원합니다.

SiteMinder 웹 에이전트 및 지원되는 웹 서버 플랫폼의 목록은 기술 지원에서 "SiteMinder Support Matrix"(SiteMinder 지원표)를 참조하십시오.

웹 에이전트는 다음 다이어그램에 나와 있는 것처럼 웹 서버에 상주합니다.


보호된
리소스
웹 서버
웹 에이전트
웹 응용 프로그램
활성 서버 페이지
스크립트
HTML 페이지
정책 서버
사용자 디렉터리
계정 로그
정책
저장소

웹 에이전트와 정책 서버의 상호 작용 방식

웹 에이전트는 모든 인증 및 권한 부여 요청을 판단하고 결정하는 정책 서버와 상호 작용하여 액세스 제어를 적용합니다.

웹 에이전트는 리소스에 대한 사용자 요청을 가로챈 후 요청된 리소스가 보호되는지 여부를 정책 서버에 확인합니다. 해당 리소스가 보호되지 않는 경우에는 액세스 요청이 웹 서버에 직접 전달되고 리소스가 보호되는 경우에는 다음과 같은 순서로 진행됩니다.

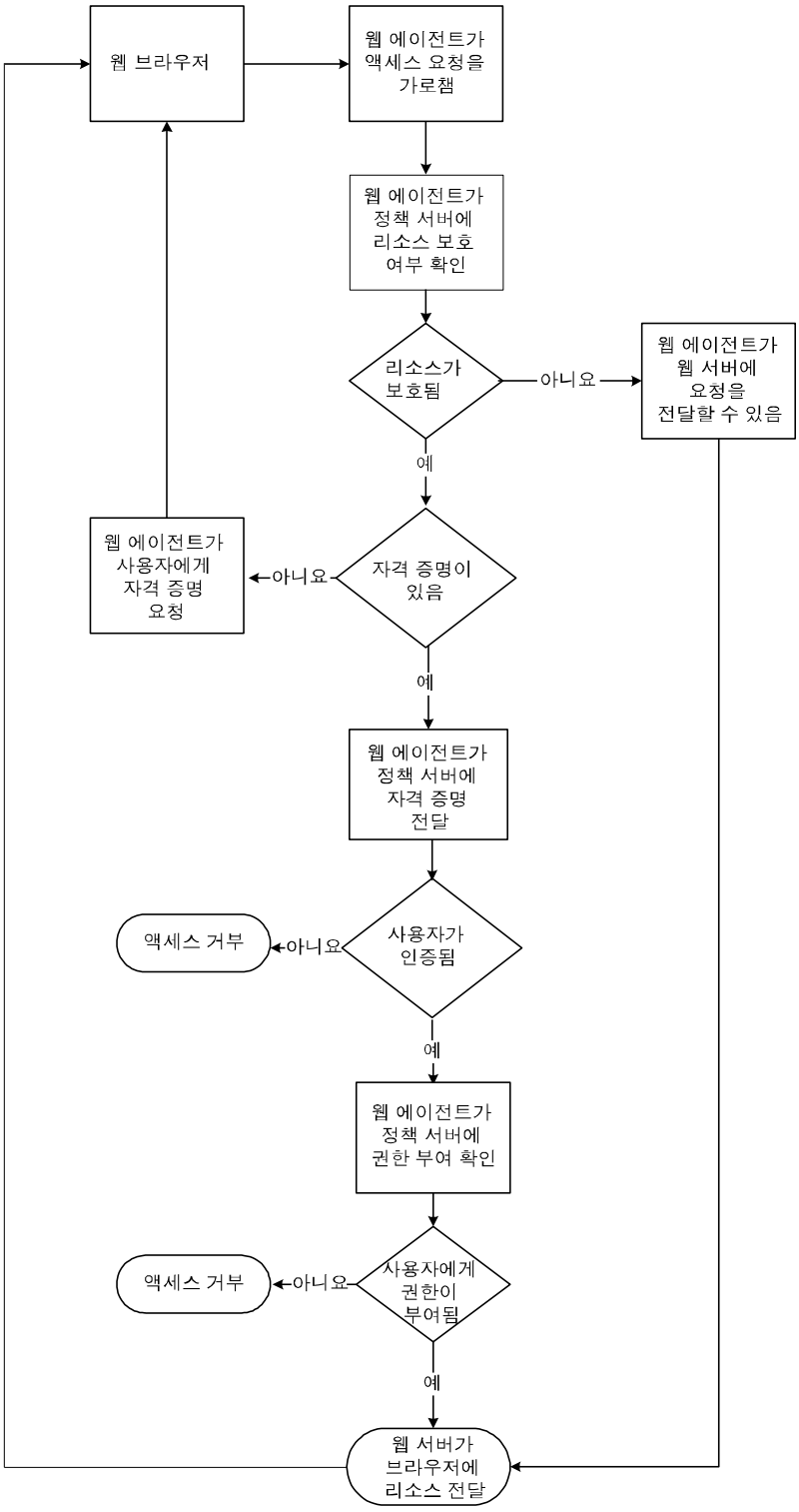
1. 웹 에이전트가 해당 리소스에 필요한 인증 방법을 확인합니다. 일반적으로 사용되는 자격 증명은 이름과 암호이지만 인증서 또는 토큰 카드 PIN 같은 다른 자격 증명이 필요할 수 있습니다.
2. 웹 에이전트가 사용자에게 자격 증명을 요청합니다.

   사용자가 적절한 자격 증명으로 응답합니다.
3. 웹 에이전트가 자격 증명을 정책 서버에 전달하여 해당 자격 증명이 올바른지 여부를 확인합니다.
4. 사용자가 인증 단계를 통과하면 정책 서버는 해당 사용자가 리소스에 액세스할 수 있는지 여부를 확인합니다. 정책 서버가 액세스 권한을 부여하면 웹 에이전트가 요청을 웹 서버에 전달할 수 있습니다.

또한 웹 에이전트는 웹 콘텐츠 개인 설정 및 세션 관리를 활성화하기 위해 사용자 고유 특성을 *응답* 형태로 받습니다. 응답은 사용자에게 권한을 부여한 후 정책 서버에서 웹 에이전트에 반환하는 개인 설정 메시지 또는 기타 사용자 고유 정보입니다. 이러한 응답은 웹 응용 프로그램에 사용하기 위해 웹 에이전트에서 HTTP 헤더에 추가하는 이름-값 특성 쌍으로 구성됩니다. 다음은 응답에 대한 예입니다.

- 사용자에게 웹 응용 프로그램에 대한 액세스 권한을 부여한 후 웹 에이전트가 사용자 세션 지속 가능 시간을 나타내는 정보를 웹 응용 프로그램에 보낼 수 있습니다.
- 사이트에 이미 등록된 사용자가 반환되는 경우 웹 에이전트가 해당 사용자의 구매 선호도 정보를 반환할 수 있습니다.

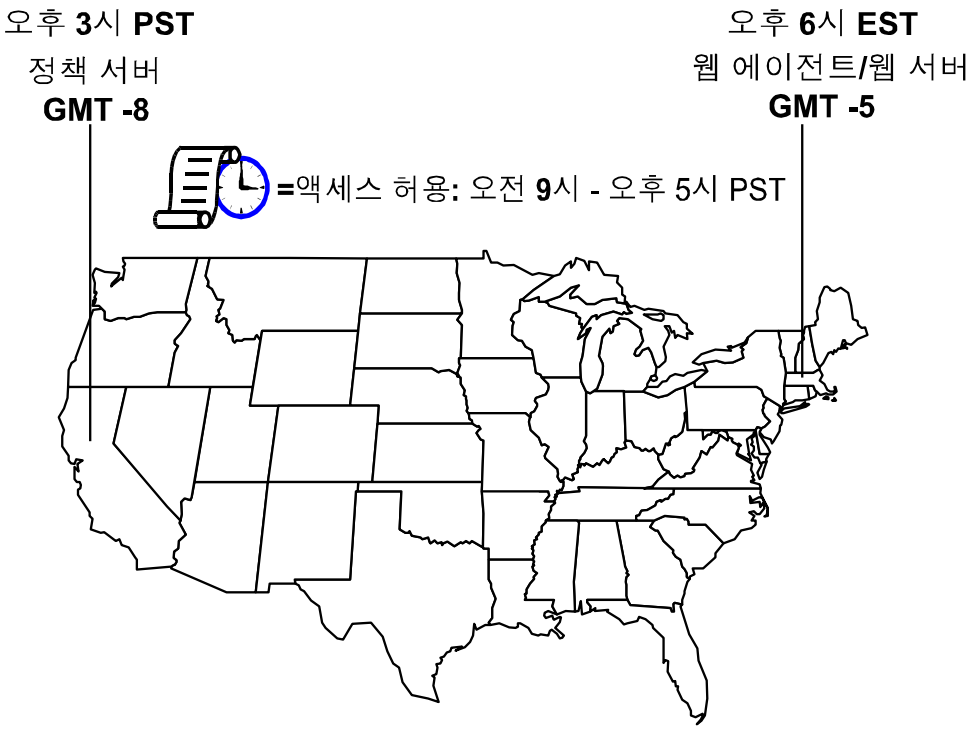
다음 다이어그램에서는 웹 에이전트와 정책 서버 간의 통신을 보여 줍니다.


웹 브라우저
웹 에이전트가 액세스 요청을 가로챔
웹 에이진트가 정책 서버에 리소스 보호 여부 확인
리소스가 보호됨
아니요
웹 에이전트가 웹 서버에 요청을 전달할 수 있음
예
웹 에이전트가 사용자에게 자격 증명 요청
아니요
자격 증명이 있음
예
웹 에이전트가 정책 서버에 자격 증명 전달
액세스 거부
아니요
사용자가 인증됨
예
웹 에이전트가 정책 서버에 권한 부여 확인
액세스 거부
아니요
사용자에게 권한이 부여됨
예
웹 서버가 브라우저에 리소스 전달

웹 에이전트와 정책 서버의 표준 시간대가 다른 경우 고려할 사항

기본적으로 정책 서버와 웹 에이전트는 GMT(그리니치 표준시)를 기준으로 시간을 계산합니다. 따라서 정책 서버 또는 웹 에이전트가 설치된 각 시스템의 경우 해당 시스템의 지리적 위치에 맞는 표준 시간대로 시스템 클록을 설정해야 합니다.

다음 그림에서는 정책 서버가 시간에 따라 정책을 실행하는 방식을 보여 줍니다. 리소스는 메사추세츠의 웹 서버에 저장되어 있으며 캘리포니아의 정책 서버에서 보호됩니다. 정책에 따라 오전 9 시~오후 5 시 사이에 리소스에 액세스할 수 있습니다. 그러나 이 정책은 웹 에이전트의 표준 시간대인 EST(동부 표준시)와 3 시간 차이가 나는 정책 서버의 시간대, 즉 PST(태평양 표준시)를 기준으로 하기 때문에 메사추세츠의 사용자는 오후 6 시에도 리소스에 액세스할 수 있습니다.



정책 서버에 따르면 오후 **6**시는 메사츄세츠에서 오후 **3**시이므로 메사츄세츠 사용자가 리소스에 액세스할 수 있음

참고: Windows 시스템의 경우 날짜/시간 제어판에 설정되는 시간과 표준 시간대 설정이 일치해야 합니다. 예를 들어 미국의 시스템을 동부 표준시에서 태평양 표준시로 재설정하려면 다음 태스크를 순서대로 수행하십시오.

1. 표준 시간대를 태평양 표준시로 설정합니다.
2. 시스템 클록에 동부 표준시보다 3 시간 빠른 시간이 표시되는지 확인합니다.

참고: 이러한 설정이 다르면 여러 도메인 사이의 싱글 사인온 및 에이전트 키 관리가 올바로 작동하지 않습니다.

에이전트가 SiteMinder 쿠키를 읽는 방법

웹 에이전트는 에이전트 키를 사용하여 SiteMinder 쿠키를 암호화하거나 암호를 해독하여 쿠키에 포함된 데이터를 읽을 수 있도록 합니다. 에이전트는 사용자 브라우저에 쿠키를 보내기 전에 키를 사용하여 쿠키를 암호화하고 다른 웹 에이전트에서 받은 쿠키의 암호를 해독할 때도 키를 사용합니다.

모든 웹 에이전트는 같은 키를 인식해야 하고 정책 서버와 통신하는 모든 에이전트에 대해 키를 같은 값으로 설정해야 합니다. 이 규칙은 싱글 사인온 환경에 포함된 에이전트의 경우 특히 중요합니다. 키 보안을 유지하기 위해 정책 서버는 키 *롤오버*를 수행합니다. 키 롤오버는 새 키를 생성하여 암호화한 후 SiteMinder 환경의 모든 웹 에이전트에 키를 배포하는 일련의 프로세스입니다.

웹 에이전트가 시작되고 관리 호출을 요청하면 정책 서버에서 현재 키 집합을 제공합니다. 웹 에이전트는 정책 서버를 폴링할 때마다 다시 관리 호출을 요청합니다. 이때 웹 에이전트는 업데이트된 키를 받습니다.

정책 서버는 다음과 같은 유형의 키를 제공합니다.

동적 키

정책 서버 알고리즘에 의해 생성되고 연결된 다른 정책 서버 및 관련 웹 에이전트에 배포되는 키를 말합니다. 동적 키는 일정한 간격에 따라 자동으로 롤오버되거나 관리 UI 를 사용하여 수동으로 변경할 수 있습니다.

정적 키

같은 값이 무기한 유지되는 키를 말하며 정책 서버 알고리즘에 의해 생성되거나 수동으로 구성할 수 있습니다. SiteMinder 에서는 정보를 오랫동안 쿠키에 저장해야 하는 기능의 하위 집합에 대해 이 유형의 키를 사용합니다.

키 변경을 자동화하면 단일 *키 저장소*를 공유하는 큰 규모의 SiteMinder 설치 환경에서 에이전트 키를 쉽게 관리할 수 있습니다. 키 저장소는 모든 키 정보가 저장되는 위치입니다. 정책 서버는 키 저장소에 액세스하여 현재 키를 가져온 후 웹 에이전트에 전달합니다. 싱글 사인온을 위해 구성된 에이전트의 경우 키 저장소를 복제하여 싱글 사인온 환경의 모든 정책 서버에서 공유해야 합니다. 또한 키 변경을 자동화하면 키의 무결성을 유지할 수 있습니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

웹 에이전트 및 동적 키 롤오버

관리 UI 를 사용하여 동적 에이전트 키 롤오버를 구성할 수 있습니다. 웹 에이전트는 정책 서버를 정기적으로 폴링하여 키 업데이트를 확인합니다. 키가 업데이트된 경우 웹 에이전트는 폴링 중에 변경 내용을 가져옵니다. 기본 폴링 시간은 30 초이지만 웹 에이전트의 PSPollInterval 매개 변수 값을 변경하여 시간을 사용자 지정할 수 있습니다.

웹 에이전트는 키 롤오버가 발생했음을 감지하면 다음 에이전트 키에 대한 새 값을 검색합니다.

이전 키

현재 값 이전에 동적 에이전트 키에 사용된 마지막 값을 포함합니다.

현재 키

현재 동적 에이전트 키의 값을 포함합니다.

이후 키

동적 에이전트 키 롤오버에서 *현재 키*로 사용될 다음 값을 포함합니다.

정적 키

사용자를 식별하고 이 정보를 오랫동안 유지해야 하는 SiteMinder 기능을 위해 에이전트에서 사용할 수 있는 장기 키를 포함합니다. 또한 정적 키는 동적 키를 사용할 수 없는 경우 싱글 사인온에 대해 쿠키 암호화를 지원합니다.

웹 에이전트에는 쿠키 데이터를 유지하고 이전 키와 새 키 사이의 원활한 전환을 위해 키가 여러 개 필요합니다.

추가 정보:

에이전트에서 정책 또는 키 업데이트 검사 간격 변경 (페이지 78)

키 저장소

정책 서버는 동적 키를 생성한 후 키 저장소에 키를 저장하고 유지 관리합니다. 키 저장소는 모든 정책 서버가 최신 키를 검색하는 리포지토리입니다. 웹 에이전트는 정책 서버에서 현재 키를 가져옵니다. 키 저장소는 SiteMinder 정책 저장소에 포함되거나 독립형 키 저장소로 유지 관리될 수 있습니다.

참고: 관리자가 에이전트 키 롤오버를 연속적으로 여러 개 발급하는 경우 싱글 사인온에 대해 발급된 모든 쿠키가 무효화될 수 있으며 현재 로그인된 모든 사용자에 대해 싱글 사인온이 중단될 수 있습니다. 싱글 사인온이 정상적으로 작동하려면 이러한 사용자를 다시 인증해야 합니다.

프레임워크 에이전트 및 기존 에이전트 아키텍처

모든 SiteMinder 에이전트는 다음 아키텍처 중 *하나*를 기반으로 합니다.

- 기존 에이전트는 처음의 SiteMinder 에이전트 아키텍처를 기반으로 합니다.
- 프레임워크 에이전트는 SiteMinder 버전 5.x QMR 6 에서 처음 도입되었습니다.

기본적으로 에이전트 기능은 아키텍처에 관계없이 동일하지만 약간의 차이가 있습니다. 예를 들어 프레임워크 에이전트는 다양한 WebAgent.conf 파일 및 LocalConfig.conf 파일을 사용하지만 기존 에이전트는 이러한 파일을 사용하지 *않습니다*.

기존 에이전트는 다음 웹 서버에 설치됩니다.

- Domino(지원되는 모든 버전)

프레임워크 에이전트는 다음 웹 서버에 설치됩니다.

- Microsoft IIS(인터넷 정보 서비스) 7.0, 7.5
- Apache 2.0.54, 2.2.x 및 기타 Apache 2.0 기반 서버(예: IBM HTTP Server 및 HP Apache 서버)
- Oracle iPlanet 웹 서버 버전 6.1 이상

참고: Oracle iPlanet 웹 서버의 이전 이름은 "Sun Java Systems" 또는 "SunONE"입니다.

추가 정보:

프레임워크 에이전트와 기존 에이전트 간의 POST 보존 사용 (페이지 186)

변경되면 서버를 다시 시작해야 하는 매개 변수

일부 에이전트 매개 변수는 동적으로 업데이트됩니다. 다음 매개 변수에 대한 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다.

AgentConfigObject

정책 서버에 저장된 에이전트 구성 개체의 이름을 로컬 에이전트 구성 파일에 정의합니다. 이 매개 변수는 에이전트 구성 개체에서 사용되지 *않습니다*.

기본값: 기본값 없음

CacheAnonymous

웹 에이전트가 익명 사용자 정보를 캐시할지 여부를 지정합니다. 다음과 같은 경우 이 매개 변수를 설정할 수 있습니다.

- 웹 사이트의 방문자가 대부분 익명 사용자이고 이러한 사용자의 세션 정보를 저장하려는 경우
- 웹 사이트의 방문자가 등록된 사용자와 익명 사용자가 섞여 있는 경우

  익명 사용자 정보가 캐시를 채워 등록된 사용자에 대한 공간이 부족해지지 않도록 이 매개 변수를 해제할 수도 있습니다.

기본값: No

HostConfigFile

트러스트된 호스트 컴퓨터가 정책 서버에 성공적으로 등록된 후 생성되는 SMHost.conf 파일(IIS 6.0 또는 Apache 에이전트)의 경로를 지정합니다. 컴퓨터의 모든 웹 에이전트가 SMHost.conf 파일을 공유합니다.

기본값: 기본값 없음

MaxResourceCacheSize

웹 에이전트가 리소스 캐시에 유지하는 최대 항목 수를 지정합니다. 항목에는 다음 정보가 포함됩니다.

- 리소스가 보호되는지 여부에 대한 정책 서버 응답
- 응답과 함께 반환된 추가 특성

최대 수에 도달하면 새 리소스 레코드가 가장 오래된 리소스 레코드를 대체합니다.

이 값을 높은 수로 설정하는 경우 충분한 시스템 메모리를 사용할 수 있어야 합니다.

OneView 모니터를 사용하여 웹 에이전트 통계를 보면 ResourceCacheCount 에 대해 표시되는 값이 MaxResourceCacheSize 매개 변수에 지정한 값보다 크다는 것을 알 수 있습니다. 이는 오류가 아닙니다. 웹 에이전트는 MaxResourceCacheSize 매개 변수를 기준으로 사용하며 MaxResourceCacheSize 매개 변수는 리소스 캐시에서 평균 크기 항목의 최대 수를 나타내기 때문에 때로는 값이 다를 수 있습니다. 실제 캐시 항목은 미리 결정된 평균 크기보다 크거나 작을 수 있으므로 실제 최대 항목 수는 지정된 값보다 크거나 작을 수 있습니다.

참고: 프레임워크 에이전트처럼 공유 메모리를 사용하는 웹 에이전트의 경우 캐시가 MaxResourceCacheSize 값에 따라 상수 크기로 사전 할당되므로 커지지 않습니다.

기본값: (Domino 웹 서버) 1000

기본값: (IIS 및 Sun Java System 웹 서버) 700

기본값: (Apache 웹 서버) 750

MaxSessionCacheSize

에이전트가 세션 캐시에 유지하는 최대 사용자 수를 지정합니다. 세션 캐시에는 성공적으로 인증한 사용자의 세션 ID 가 저장됩니다. 세션 중에 해당 영역 내의 다른 리소스에 액세스하는 인증된 사용자는 정책 서버 대신 세션 캐시를 사용하여 인증됩니다. 최대 수에 도달하면 에이전트는 가장 오래된 사용자 레코드를 새로운 사용자 레코드로 대체합니다.

지속된 기간 동안 리소스에 액세스하여 리소스를 사용할 것으로 예상되는 사용자의 수를 기반으로 이 매개 변수의 값을 지정하십시오. 이 값을 높은 수로 설정하는 경우 충분한 시스템 메모리를 사용할 수 있어야 합니다.

참고: 캐시 크기에 관계없이 웹 에이전트의 세션 캐시에 저장된 모든 항목은 1 시간 후 *자동으로 만료됩니다.*

기본값: (Domino 웹 서버) 1000

기본값: (IIS 및 Oracle iPlanet 웹 서버) 700

기본값: (Apache 웹 서버) 750

PostPreservationFile

다음 POST 보존 템플릿 파일 중 *하나*에 대한 경로를 지정하여 기존 에이전트와 프레임워크 에이전트 간에 POST 보존 데이터를 전송하도록 설정합니다.

- tr2fw.pptemplate - 기존 에이전트를 실행하는 서버에서 호스트되는 리소스가 프레임워크 에이전트에서 실행되는 FCC 로 보호되도록 지정합니다.
- fw2tr.pptemplate - 프레임워크 에이전트를 실행하는 서버에서 호스트되는 리소스가 기존 에이전트에서 실행되는 FCC 로 보호되도록 지정합니다.

기본값: 기본값 없음

예: *web_agent_home*/samples/forms/fw2tr.pptemplate

ResourceCacheTimeout

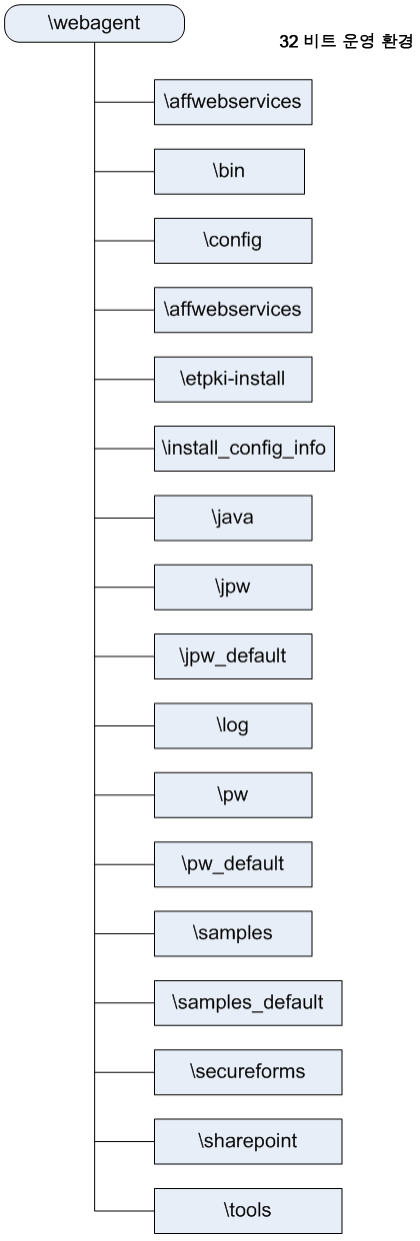
리소스 항목이 캐시에서 유지되는 시간(초)을 지정합니다. 이 시간 간격이 초과된 후 사용자가 보호된 리소스에 액세스하려고 하면 웹 에이전트는 캐시된 항목을 제거하고 정책 서버에 연결합니다.

기본값: 600(10 분)

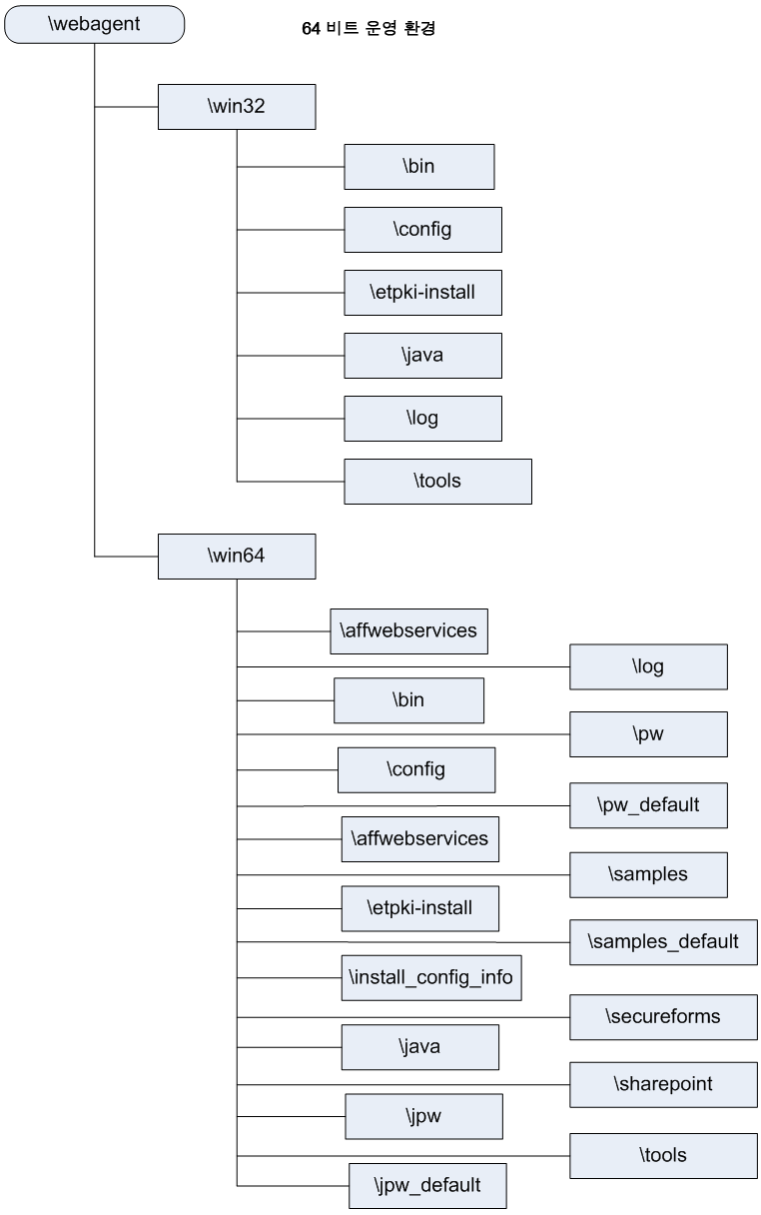
IIS 용 에이전트의 여러 디렉터리 구조

IIS 웹 서버에 추가되는 에이전트 파일 디렉터리 구조는 IIS 웹 서버의 운영 환경에 따라 다릅니다. 예를 들면 다음과 같은 디렉터리 구조를 사용합니다.

- IIS 용 SiteMinder 에이전트는 다음 그림과 같은 디렉터리 구조를 사용합니다.



- 64 비트 운영 환경에 설치된 IIS 용 SiteMinder 에이전트는 다음 그림과 같은 디렉터리 구조를 사용합니다.



제 2 장: 에이전트 구성 방법

이 섹션은 다음 항목을 포함하고 있습니다.

중앙 구성

중앙 에이전트 구성은 정책 서버의 에이전트 구성 개체에서 하나 이상의 웹 에이전트를 관리합니다. 에이전트 구성 개체는 정책 서버에 있으며 웹 에이전트에 사용되는 매개 변수를 포함합니다. 중앙 구성을 사용하면 여러 에이전트의 매개 변수 설정을 동시에 업데이트할 수 있다는 이점이 있습니다. 대부분의 매개 변수 변경은 동적으로 수행되지만 일부 프레임워크 매개 변수의 경우 변경된 후 웹 서버를 다시 시작해야 합니다.

관리 UI 를 사용하여 에이전트 구성 개체를 생성하고 편집합니다. 정책 서버와 통신하는 각 웹 에이전트는 에이전트 구성 개체와 연결되어야 하지만 대부분의 웹 에이전트는 단일 에이전트 구성 개체를 사용할 수 있습니다.

참고: 에이전트 구성 개체를 생성하는 방법에 대한 자세한 내용은 정책 서버 설명서를 참조하십시오.

추가 정보

변경되면 서버를 다시 시작해야 하는 매개 변수 (페이지 28)

중앙 구성 구현

중앙 구성은 기본적으로 사용되도록 설정됩니다. 에이전트는 구성 마법사를 사용하여 에이전트를 구성할 때 지정한 기존 에이전트 구성 개체의 구성 설정을 사용합니다. 필요에 따라 언제든지 매개 변수의 설정을 변경할 수 있습니다.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그인합니다.

   "시작" 화면이 표시됩니다.

2. "인프라", "에이전트 구성 개체"를 차례로 클릭합니다.

   에이전트 구성 개체 목록이 나타납니다.

3. 원하는 에이전트 구성 개체 행에서 수정 아이콘을 클릭합니다.

   "에이전트 구성 수정" 창이 나타납니다.

4. AllowLocalConfig 매개 변수의 값이 no 로 설정되어 있는지 확인합니다.

5. 관리 UI 를 사용하여 다른 매개 변수의 설정을 필요에 따라 수정합니다.

6. "제출"을 클릭합니다.

   "에이전트 구성 수정" 창이 닫히고 확인 메시지가 표시됩니다.

7. (선택 사항) 나중에 참조할 수 있도록 "설명" 필드에 변경 내용에 대해 간단히 입력합니다.

8. "예"를 클릭합니다.

   확인 메시지가 표시됩니다. 중앙 구성이 구현되었습니다. 대부분의 매개 변수 변경은 동적으로 실행되지만 일부 매개 변수의 경우 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다.

추가 정보:

변경되면 서버를 다시 시작해야 하는 매개 변수 (페이지 28)

로컬 에이전트 구성

로컬 구성

로컬 에이전트 구성은 웹 서버를 호스트하는 시스템에 설치된 로컬 파일을 사용하여 웹 에이전트를 관리합니다. 로컬 파일의 매개 변수 설정은 정책 서버의 에이전트 구성 개체에 저장된 모든 설정을 무시합니다. 에이전트 구성 개체의 설정은 변경되지 않습니다. 로컬 에이전트 구성을 고려해야 하는 상황은 다음과 같습니다.

- Apache 웹 에이전트가 3 개 있고 처음 두 에이전트(A 와 B)가 동일한 매개 변수 설정을 사용하지만 세 번째 Apache 에이전트(C)는 A 와 B 의 설정의 대부분을 사용하는 한편 리버스 프록시로 작동하도록 하려는 경우. 이를 위해 Apache 에이전트 A 와 B 에는 중앙 에이전트 구성을 사용하고 Apache 에이전트 C 에는 로컬 구성을 사용하면 됩니다.

- 정책 서버 관리자가 에이전트를 구성하는 사용자 또는 그룹과 다를 경우. 예를 들어 회사의 정보 기술 부서에서 정책 서버를 유지 관리하지만 재무 부서에서 에이전트를 사용하여 회계 응용 프로그램에 대한 액세스 권한을 제어할 수 있습니다. 정보 기술 부서의 누군가가 정책 서버에서 에이전트에 대한 로컬 구성을 설정하지만 재무 부서의 다른 누군가가 회계 응용 프로그램을 보호하는 에이전트의 특정 구성 설정을 제어합니다.

프레임워크 웹 에이전트는 다음과 같은 로컬 구성 파일을 사용합니다.

WebAgent.conf

프레임워크 웹 에이전트가 시작하고 정책 서버에 연결할 때 사용하는 핵심 설정을 포함합니다.

LocalConfig.conf

프레임워크 웹 에이전트에 대한 구성 설정을 포함합니다.

기존 웹 에이전트는 다음과 같은 로컬 구성 파일을 사용합니다.

WebAgent.conf

기존 웹 에이전트에 대한 모든 구성 설정을 포함합니다.

WebAgent.conf 파일 위치

다음 표에서는 여러 웹 서버의 WebAgent.conf 파일 위치를 보여 줍니다.

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

웹 서버	파일 위치
IIS	*web_agent_home*\bin\IIS
Oracle iPlanet(iPlanet/SunOne)	*Oracle_iPlanet_server_home*/https-*hostname*/config 여기서 *Oracle_iPlanet_home* 은 Oracle iPlanet 웹 서버가 설치된 위치이고 *hostname* 은 서버 이름입니다.
Apache, IBM HTTP Server Oracle HTTP Server	*web_server_home*/conf 여기서 *web_server_home* 은 웹 서버가 설치된 위치입니다.
Domino	Windows: c:\lotus\domino UNIX: $HOME/notesdata

추가 정보:

웹 에이전트 사용 (페이지 73)
웹 에이전트 사용 안 함 (페이지 74)

프레임워크 에이전트의 WebAgent.conf 파일

AgentConfigObject, HostConfigFile 및 EnableWebAgent 매개 변수 외에 다음 매개 변수도 프레임워크 에이전트의 WebAgent.conf 파일에 추가됩니다.

중요! 웹 에이전트 외에 다른 SiteMinder 제품을 참조하는 파일 섹션을 수정하면 안 됩니다. 그러나 파일에서 웹 에이전트 매개 변수의 값은 변경할 수 있습니다.

LocalConfigFile

대부분의 에이전트 구성 설정이 포함된 LocalConfig.conf 파일의 위치를 지정합니다.

ServerPath

에이전트에 대한 웹 서버 디렉터리(Apache 2.0 및 Oracle iPlanet 웹 서버)를 식별합니다.

LoadPlugin

프레임워크 에이전트에 대해 로드되는 플러그인을 지정합니다. 플러그인은 다양한 유형의 에이전트 기능을 지원합니다. 다음과 같은 플러그인을 사용할 수 있습니다.

HttpPlugin

웹 에이전트가 HTTP 에이전트로 작동하는지 여부를 지정합니다.

기본값: 사용

SAMLAffiliatePlugin

Federation Security Services 를 구입한 경우 웹 에이전트와 SAML 가맹 에이전트 간의 통신을 허용합니다.

기본값: 사용 안 함

Affiliate10Plugin

웹 에이전트와 4.x 가맹 에이전트 간의 통신을 허용합니다.

기본값: 사용 안 함

제한: SAML 가맹 에이전트는 이 플러그인을 사용하지 않습니다.

OpenIDPlugin

웹 에이전트에서 OIAS(OpenID 인증 체계)를 사용할 수 있도록 합니다.

기본값: 사용 안 함

다른 LoadPlugin 항목을 사용하려면 행의 맨 앞에 있는 파운드 기호(#)를 제거하십시오.

AgentIdFile

에이전트의 고유 ID 문자열을 저장하는 AgentId 파일의 경로를 지정합니다. 에이전트는 AgentId 파일을 자동으로 생성하며 이 파일은 수정하면 안 됩니다. Windows 와 UNIX 둘 다에서 에이전트는 AgentId 파일을 업데이트할 수 있는 쓰기 권한이 있어야 합니다. Windows 의 경우 웹 에이전트 구성 마법사에서 자동으로 쓰기 권한을 부여합니다.

기본 이름: Agentid.dat

경로: WebAgent.conf 디렉터리/AgentId.dat

추가 정보

웹 서버 인스턴스가 여러 개인 경우 웹 에이전트 관리 (페이지 63)

LocalConfig.conf 파일 위치(프레임워크 에이전트)

프레임워크 웹 에이전트를 설치하는 경우 SiteMinder 설치 프로그램에서 다음 디렉터리에 LocalConfig.conf 파일을 생성합니다.

Windows

web _agent_home\config

UNIX

web _agent_home/config

중요! 이 파일은 모든 기본 설정을 포함합니다. 이 파일을 수정하지 마십시오. 나중에 참조하거나 복구에 사용할 수 있도록 이 파일의 백업 복사본을 만들어 놓는 것이 좋습니다.

웹 에이전트를 구성하는 경우 구성 마법사가 LocalConfig.conf 파일을 다음 디렉터리에 복사합니다.

IIS 웹 서버

web_agent_home\bin\IIS

Oracle iPlanet 웹 서버

Oracle_iPlanet_home/https-*hostname*/config

Apache 웹 서버

Apache_home/conf

웹 에이전트는 LocalConfig.conf 파일 복사본에서 구성 설정을 검색합니다.

로컬 구성 파일에만 있는 매개 변수

중앙 에이전트 구성의 경우 로컬 구성 파일에 있는 대부분의 매개 변수가 에이전트 구성 개체에도 있습니다. 다음 매개 변수는 로컬 구성 파일에만 사용되며 에이전트 구성 개체에 *없습니다*.

AgentConfigObject

정책 서버에 저장된 에이전트 구성 개체의 이름을 로컬 에이전트 구성 파일에 정의합니다. 이 매개 변수는 에이전트 구성 개체에서 사용되지 *않습니다*.

기본값: 기본값 없음

EnableWebAgent

웹 에이전트를 활성화하여 정책 서버와 통신할 수 있도록 합니다. 구성 매개 변수를 모두 변경한 후에만 이 매개 변수를 yes 로 설정하십시오.

기본값: No

HostConfigFile

트러스트된 호스트 컴퓨터가 정책 서버에 성공적으로 등록된 후 생성되는 SMHost.conf 파일(IIS 6.0 또는 Apache 에이전트)의 경로를 지정합니다. 컴퓨터의 모든 웹 에이전트가 SMHost.conf 파일을 공유합니다.

기본값: 기본값 없음

로컬 구성 구현

다음 매개 변수를 사용하여 로컬 구성 허용 여부를 제어할 수 있습니다.

AllowLocalConfig

로컬 구성 파일을 읽고 에이전트에 대한 구성 매개 변수를 가져오도록 정책 서버의 에이전트 구성 개체에 지시합니다. 이 매개 변수는 에이전트 구성 개체에서만 사용됩니다.

로컬 구성 파일에서 변경할 수 있는 매개 변수를 제어하려면 에이전트 구성 개체에서 이 매개 변수에 대해 값을 여러 개 추가하십시오. 이 매개 변수에 여러 개의 값이 설정된 경우에는 다음과 같은 순서로 처리됩니다.

- yes 가 사용되면 *모든* 매개 변수를 로컬로 설정할 수 있습니다.
- No 는 매개 변수 목록보다 우선 순위가 높습니다. 또한 두 값이 함께 설정된 경우 No 가 yes 보다 우선 적용됩니다. 이 옵션을 사용하면 에이전트 구성 개체에서 다른 구성 매개 변수를 제거할 필요 없이 로컬 구성 전체를 빠르게 비활성화할 수 있습니다.

기본값: No(로컬 구성 허용 안 함)

예: No, EnableAuditing, EnableMonitoring(모든 로컬 구성 허용 안 함)

예: No, Yes(모든 로컬 구성 허용 안 함)

예: EnableAuditing, EnableMonitoring(지정된 두 매개 변수만 로컬 제어 허용)

다음 단계를 수행하십시오.

1. 관리 UI 에 로그인합니다.
2. "인프라", "에이전트 구성 개체"를 차례로 클릭합니다.
3. 원하는 에이전트 구성 개체 행에서 수정 아이콘을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 나타납니다.
4. AllowLocalConfig 매개 변수 왼쪽에 있는 편집 아이콘을 클릭합니다.

   "매개 변수 편집" 대화 상자가 나타납니다.
5. "값" 필드의 텍스트를 "yes"로 변경하고 "확인"을 클릭합니다.

   "매개 변수 편집" 대화 상자가 닫힙니다.
6. 제출을 클릭합니다.
7. (선택 사항) 나중에 참조할 수 있도록 "설명" 필드에 변경 내용에 대해 간단히 입력합니다.

8. "예"를 클릭합니다.

   로컬 구성을 사용할 수 있습니다.

9. 웹 서버에서 해당하는 로컬 구성 파일을 열고 원하는 매개 변수 설정을 변경합니다.
10. (기존 에이전트만 해당) EnableWebAgent 매개 변수의 값을 yes 로 설정합니다.
11. 로컬 구성 파일을 저장한 후 닫습니다.
12. (프레임워크 에이전트만 해당) 다음 단계를 수행하십시오.
    a. WebAgent.conf 파일을 엽니다.
    b. EnableWebAgent 매개 변수의 값을 yes 로 설정합니다.
    c. WebAgent.conf 파일을 저장한 후 닫습니다.
13. 웹 서버를 다시 시작합니다.

    로컬 구성을 사용할 수 있고 업데이트된 매개 변수가 변경되었습니다.

추가 정보:

웹 에이전트 사용 (페이지 73)
변경되면 서버를 다시 시작해야 하는 매개 변수 (페이지 28)

에이전트 구성 파일을 편집하는 방법

에이전트 구성 파일은 로컬로 구성된 웹 에이전트의 설정을 제어합니다. 이러한 설정을 변경하려면 다음 과정을 수행하십시오.

1. WebAgent.conf(기존 에이전트의 경우) 또는 LocalConfig.conf(프레임워크 에이전트의 경우) 파일의 백업 복사본을 생성합니다.
2. 텍스트 편집기에서 에이전트 구성 파일 원본을 엽니다.
3. 다음 태스크를 수행하여 매개 변수를 사용하거나 사용하지 않도록 설정합니다.
   - 매개 변수를 사용하려면 행의 맨 앞에 있는 파운드 기호(#)를 제거합니다.
   - 매개 변수를 사용하지 않으려면 행의 맨 앞에 파운드 기호(#)를 추가합니다.
4. 다음 지침을 사용하여 매개 변수의 값을 변경합니다.
   - 매개 변수 이름, = 기호 및 매개 변수 값 사이에는 공백을 추가하면 안 됩니다.
   - 매개 변수 값은 따옴표로 묶습니다.
   - WebAgent.conf 및 LocalConfig.conf 파일은 대/소문자를 구분하지 않습니다. 따라서 에이전트와 함께 설치된 샘플 파일에 표시되는 대/소문자를 일치시킬 필요가 없습니다.
   - <AgentName>, <IPAddress>와 같이 파일의 여러 값은 설명이 포함된 변수로 나타납니다. 꺾쇠 괄호와 텍스트를 원하는 값으로 바꾸십시오.
   - 값이 비어 있는 경우 공백은 기본값으로 유효합니다. 기본값은 매개 변수 앞에 파운드 기호(#)가 없을 때만 적용됩니다.
5. 끝났으면 EnableWebAgent 를 yes 로 설정합니다. 그런 다음 파일을 저장하고 닫습니다.

   모든 로컬 구성 변경 내용이 적용됩니다. 에이전트 설정이 완료된 후 추가로 변경한 경우 해당 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다.

로컬 구성 매개 변수에 대한 변경 제한

중앙 에이전트 구성을 사용하면 로컬 웹 서버 관리자가 수정하는 구성 매개 변수를 제한할 수 있습니다. SiteMinder 관리자와 웹 서버 관리자가 다른 사람일 경우 이 방법을 사용하는 것이 좋습니다.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그인합니다.

   "시작" 화면이 표시됩니다.

2. "인프라", "에이전트 구성 개체"를 차례로 클릭합니다.

   에이전트 구성 개체 목록이 나타납니다.

   원하는 에이전트 구성 개체 행에서 편집 아이콘을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 나타납니다.

3. AllowLocalConfig 매개 변수 왼쪽에 있는 편집 아이콘을 클릭합니다.

   "매개 변수 편집" 대화 상자가 나타납니다.

4. "값" 필드의 텍스트를 지우고 다중값 옵션 단추를 클릭합니다.

5. "추가"를 클릭합니다.

   빈 필드가 나타납니다.

6. 액세스를 허용할 매개 변수의 이름을 필드에 입력합니다. 매개 변수가 여러 개일 경우에는 쉼표로 구분하십시오. 이 목록의 매개 변수만 로컬로 변경할 수 있습니다.

   예: 다음 예제에서는 EnableAuditing 매개 변수와 EnableMonitoring 매개 변수만 로컬 웹 서버에 설정할 수 있도록 허용하는 방법을 보여 줍니다.

   AllowLocalConfig=EnableAuditing,EnableMonitoring

7. (선택 사항) 매개 변수를 더 추가하려면 5 단계와 6 단계를 반복합니다.

8. "확인"을 클릭합니다.

   "매개 변수 편집" 대화 상자가 닫히고 "에이전트 구성 수정" 대화 상자가 나타납니다.

9. "제출"을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 닫히고 확인 메시지가 표시됩니다.

10. (선택 사항) 나중에 참조할 수 있도록 "설명" 필드에 변경 내용에 대해 간단히 입력합니다.

11. "예"를 클릭합니다.

    다음에 웹 에이전트가 정책 서버를 폴링하면 변경 내용이 적용됩니다.

중앙 구성과 로컬 구성 조합

다수의 웹 에이전트를 중앙에서 구성하지만 이 중 일부 웹 에이전트를 나머지 에이전트와 다르게 설정해야 하는 경우 중앙 구성과 로컬 구성을 조합하여 사용할 수 있습니다.

예를 들어 에이전트를 개별적으로 구성하지 않고 SiteMinder 네트워크에서 여러 쿠키 도메인 싱글 사인온을 구성해야 하는 경우 모든 에이전트에는 중앙 구성을 사용하고 다른 설정이 필요한 소규모 에이전트 그룹에는 로컬 구성 설정을 사용할 수 있습니다.

위의 예제에서 에이전트 구성 개체의 CookieDomain 매개 변수가 example.com 으로 설정되어 있다고 가정합니다. 그러나 네트워크의 한 웹 에이전트에 대해 다른 모든 매개 변수 값은 에이전트 구성 개체에 설정된 값을 사용하고 CookieDomain 매개 변수만 .example.net 으로 설정하려고 합니다.

예제 구성을 구현하려면

1. 관리 UI 를 사용하여 환경에 필요한 모든 매개 변수를 포함하는 에이전트 구성 개체를 생성합니다. 이때 CookieDomain 매개 변수를 .example.com 으로 설정합니다.
2. 에이전트 구성 개체의 AllowLocalConfig 매개 변수를 yes 로 설정합니다.
3. 한 웹 에이전트에 대해 CookieDomain 매개 변수 값을 example.net 으로 사용하도록 웹 서버의 로컬 구성 파일을 변경합니다. 다른 매개 변수는 수정하지 *마십시오*.

한 에이전트의 로컬 구성 파일에 있는 CookieDomain 매개 변수 값은 에이전트 구성 개체의 값보다 우선 적용되지만 에이전트 구성 개체는 다른 모든 매개 변수의 설정을 결정합니다.

제 3 장: 웹 에이전트에 사용되는 구성 파일

SiteMinder 웹 에이전트는 특정 설정을 위해 구성 파일을 사용합니다. 이러한 구성 파일 중 일부는 웹 에이전트가 있는 웹 서버에 설치됩니다. 다른 구성 파일은 SiteMinder 웹 에이전트 구성 마법사에 의해 웹 서버에 생성됩니다. 이때 웹 서버를 호스트하는 컴퓨터에 설치된 특정 웹 서버 제품과 웹 에이전트 파일이 연결됩니다.

예를 들어 Apache 웹 서버를 실행하는 32 비트 Windows 시스템에 웹 에이전트를 설치하는 경우 웹 에이전트 구성 마법사는 SiteMinder 웹 에이전트에 필요한 Apache 웹 서버 구성을 변경합니다.

Agent Connection Manager 구성 파일

웹 에이전트 설치 마법사는 다음 위치에 Agent Connection Manager 구성 파일(AgentConMgr.conf)을 설치합니다.

web_agent_home/config

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

Agent Connection Manager 구성 파일을 사용하면 작동하는 동안 수행되는 웹 에이전트 연결에 대해 자세한 추적 로그를 생성할 수 있습니다.

추가 정보:

Agent Connection Manager 추적 로그를 사용하여 자세한 에이전트 연결 데이터 수집 (페이지 422)

연결 API 구성 파일

연결 API 파일(conapi.conf)은 연결 API 를 통해 서비스를 구성하는 데 사용됩니다. 이러한 서비스에는 OneView 모니터가 포함됩니다.

웹 에이전트 설치 마법사는 다음 위치에 연결 API 구성 파일을 생성합니다.

web_agent_home/config

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

참고: OneView 모니터 사용에 대한 자세한 내용은 *항상DNA 현재 예약됨*에서 확인하십시오.

로컬 에이전트 구성 파일

웹 에이전트 설치 마법사는 다음 위치에 로컬 에이전트 구성 파일(LocalConfig.conf)을 설치합니다.

web_agent_home/config

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

이 파일을 사용하면 연결된 정책 서버의 에이전트 구성 개체에 저장된 매개 변수 설정을 사용하지 않고 웹 에이전트가 설치되어 있는 같은 웹 서버의 웹 에이전트 구성 매개 변수를 설정할 수 있습니다.

IIS 웹 에이전트의 경우 웹 에이전트 구성 마법사는 다음 위치에 로컬 에이전트 구성 파일 복사본을 만듭니다.

web_agent_home\bin\IIS

추가 정보:

로컬 에이전트 구성 (페이지 35)
LocalConfig.conf 파일 위치(프레임워크 에이전트) (페이지 39)

추적 구성 파일

추적 구성 파일(trace.conf)을 사용하면 다음 항목에 대해 추적 로그를 구성할 수 있습니다.

- 연결 API
- IPC 공급자
- TCP/IP 전송
- 모니터링 API

웹 에이전트 설치 마법사는 다음 위치에 추적 구성 파일을 생성합니다.

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

추가 정보:

Apache 오류 로그에 기록되는 IPC 세마포 관련 메시지 출력 제한 (페이지 341)

웹 에이전트 추적 구성 파일

웹 에이전트 추적 구성 파일을 사용하면 다양한 웹 에이전트 작업에 대해 추적 로그를 생성할 수 있습니다. 예를 들어 SiteMinder SSO(싱글 사인온) 기능과 관련된 웹 에이전트 작업의 추적 로그를 생성할 수 있습니다.

웹 에이전트 설치 마법사는 다음 위치에 웹 에이전트 추적 구성 파일을 생성합니다.

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

추가 정보:

추적 로깅을 설정하는 방법 (페이지 407)

SiteMinder 호스트 구성 파일

웹 에이전트 구성 마법사는 SiteMinder 웹 에이전트를 구성하는 모든 웹 서버의 다음 위치에 호스트 구성 파일(SmHost.conf)을 생성합니다.

web_agent_home/config

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

SmHost.conf 파일에는 웹 에이전트가 정책 서버에 처음으로 연결할 때 사용하는 정보가 포함되어 있습니다.

웹 에이전트 구성 파일

SiteMinder 웹 에이전트 구성 마법사는 SiteMinder 웹 에이전트를 구성하는 모든 웹 서버의 다음 위치에 웹 에이전트 구성 파일(WebAgent.conf)을 생성합니다.

web_agent_home\conf

web_agent_home

SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

기본값(UNIX/Linux 시스템): /opt/ca/webagent

이 파일은 웹 에이전트의 사용 여부(시작 또는 중지)를 설정하는 데 사용됩니다.

IIS 웹 에이전트의 경우 웹 에이전트 구성 마법사는 다음 위치에 로컬 에이전트 구성 파일 복사본을 만듭니다.

web_agent_home\bin\IIS

추가 정보:

웹 에이전트 사용 (페이지 73)
웹 에이전트 사용 안 함 (페이지 74)

제 4 장: 기본 에이전트 설정 및 정책 서버 연결

이 섹션은 다음 항목을 포함하고 있습니다.

웹 에이전트 구성 매개 변수의 기본 설정

다른 값이 별도로 지정된 경우를 제외하고 항상 웹 에이전트 구성 매개 변수의 기본 설정이 사용됩니다.

에이전트 구성 개체 또는 로컬 구성 파일에 매개 변수가 없으면 기본값이 사용됩니다.

AgentName 및 DefaultAgentName 값 설정

AgentName 매개 변수는 에이전트 아이덴티티를 지정합니다. 정책 서버는 이 아이덴티티를 사용하여 정책을 웹 에이전트에 연결합니다. 다음 매개 변수를 사용하면 에이전트 이름을 정의할 수 있습니다.

AgentName

웹 에이전트의 아이덴티티를 정의합니다. 이 아이덴티티는 에이전트를 호스트하는 각 웹 서버 인스턴스의 이름과 IP 주소 또는 FQDN 을 연결합니다.

다음과 같은 이벤트가 발생하면 AgentName 매개 변수 대신 DefaultAgentName 값이 사용됩니다.

- AgentName 매개 변수가 사용되지 않도록 설정된 경우
- AgentName 매개 변수의 값이 비어 있는 경우
- AgentName 매개 변수의 값이 기존 에이전트 개체와 일치하지 *않는* 경우

참고: 이 매개 변수에는 값을 여러 개 지정할 수 있습니다. 이 매개 변수를 에이전트 구성 개체에 설정하는 경우 다중값 옵션을 사용하고 로컬 구성 파일에 설정할 경우에는 각 값을 파일에서 별도의 행에 추가합니다.

기본값: 기본값 없음

제한: 여러 값이 허용되지만 각 AgentName 매개 변수당 4,000 자로 제한됩니다. 매개 변수 이름에 문자를 추가하여 필요한 만큼 추가 AgentName 매개 변수를 생성하십시오. 예: AgentName, AgentName1, AgentName2

제한: 32-127 범위의 7 비트 ASCII 문자를 포함하고 인쇄 가능한 문자가 하나 이상 있어야 합니다. 앰퍼샌드(&)와 별표(*)는 사용할 수 없습니다. 이 값은 대/소문자를 구분하지 않습니다. 예를 들어 MyAgent 와 myagent 는 같은 이름으로 처리됩니다.

예: myagent1,192.168.0.0 (IPV4)

예: myagent2, 2001:DB8::/32 (IPV6)

예: myagent,www.example.com

예(여러 AgentName 매개 변수): AgentName1, AgentName2, AgentName3. 각 AgentName*number* 매개 변수의 값은 4,000 자로 제한됩니다.

DefaultAgentName

요청을 처리하기 위해 에이전트가 사용하는 이름을 정의합니다. DefaultAgentName 의 값은 AgentName 매개 변수에 에이전트 이름 값이 없을 경우 IP 주소 또는 인터페이스에 대한 요청에 사용됩니다.

가상 서버를 사용하는 경우 DefaultAgentName 을 사용하여 SiteMinder 환경을 빠르게 구성할 수 있습니다. DefaultAgentName 을 사용하면 각 가상 서버에 대해 별도의 에이전트를 정의할 필요가 없습니다.

중요! DefaultAgentName 매개 변수 값을 지정하지 않으면 AgentName 매개 변수의 값에 모든 에이전트 아이덴티티 목록을 설정해야 합니다. 그렇지 않으면 정책 서버가 에이전트에 정책을 연결할 수 없습니다.

기본값: 기본값 없음

제한: 값은 하나만 사용하십시오. 여러 값은 금지됩니다.

제한: 32-127 범위의 7 비트 ASCII 문자를 포함하고 인쇄 가능한 문자가 하나 이상 있어야 합니다. 앰퍼샌드(&)와 별표(*)는 사용할 수 없습니다. 이 값은 대/소문자를 구분하지 않습니다. 예를 들어 MyAgent 와 myagent 는 같은 이름으로 처리됩니다.

가상 서버 지원을 구성하는 경우 AgentName 또는 DefaultAgentName 매개 변수의 값을 지정하십시오.

다음 단계를 수행하십시오.

1. 다음 단계 중 하나를 수행하여 AgentName 값을 지정합니다.
   - 중앙 에이전트 구성의 경우 - 관리 UI 에서 에이전트 구성 개체를 열고 원하는 값을 AgentName 매개 변수에 추가합니다.
   - 로컬 에이전트 구성의 경우 - 웹 서버에서 로컬 구성 파일을 엽니다. 그런 다음 파일에서 별도의 행에 원하는 값을 추가합니다.
2. 다음 단계 중 하나를 수행하여 DefaultAgentName 아이덴티티를 지정합니다.
   - 중앙 에이전트 구성의 경우 - 관리 UI 에서 에이전트 구성 개체를 열고 원하는 값을 DefaultAgentName 매개 변수에 추가합니다.
   - 로컬 에이전트 구성의 경우 - 웹 서버에서 로컬 구성 파일을 엽니다. 그런 다음 원하는 값을 DefaultAgentName 매개 변수에 추가합니다.

   AgentName 및 DefaultAgentName 값이 설정됩니다.

추가 정보

가상 서버 지원을 설정하는 방법 (페이지 170)

로컬 구성 매개 변수에 대한 변경 제한

중앙 에이전트 구성을 사용하면 로컬 웹 서버 관리자가 수정하는 구성 매개 변수를 제한할 수 있습니다. SiteMinder 관리자와 웹 서버 관리자가 다른 사람일 경우 이 방법을 사용하는 것이 좋습니다.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그인합니다.

   "시작" 화면이 표시됩니다.

2. "인프라", "에이전트 구성 개체"를 차례로 클릭합니다.

   에이전트 구성 개체 목록이 나타납니다.

   원하는 에이전트 구성 개체 행에서 편집 아이콘을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 나타납니다.

3. AllowLocalConfig 매개 변수 왼쪽에 있는 편집 아이콘을 클릭합니다.

   "매개 변수 편집" 대화 상자가 나타납니다.

4. "값" 필드의 텍스트를 지우고 다중값 옵션 단추를 클릭합니다.

5. "추가"를 클릭합니다.

   빈 필드가 나타납니다.

6. 액세스를 허용할 매개 변수의 이름을 필드에 입력합니다. 매개 변수가 여러 개일 경우에는 쉼표로 구분하십시오. 이 목록의 매개 변수만 로컬로 변경할 수 있습니다.

   예: 다음 예제에서는 EnableAuditing 매개 변수와 EnableMonitoring 매개 변수만 로컬 웹 서버에 설정할 수 있도록 허용하는 방법을 보여 줍니다.

   AllowLocalConfig=EnableAuditing,EnableMonitoring

7. (선택 사항) 매개 변수를 더 추가하려면 5 단계와 6 단계를 반복합니다.

8. "확인"을 클릭합니다.

   "매개 변수 편집" 대화 상자가 닫히고 "에이전트 구성 수정" 대화 상자가 나타납니다.

9. "제출"을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 닫히고 확인 메시지가 표시됩니다.

10. (선택 사항) 나중에 참조할 수 있도록 "설명" 필드에 변경 내용에 대해 간단히 입력합니다.

11. "예"를 클릭합니다.

    다음에 웹 에이전트가 정책 서버를 폴링하면 변경 내용이 적용됩니다.

에이전트 이름 일치

SiteMinder 규칙 및 정책은 에이전트 이름과 연결되어 있습니다. 정책 서버에서 알 수 없는 에이전트 이름을 사용하여 호스트에 대한 요청이 수행되면 정책 서버에서 정책을 구현할 수 없습니다. 따라서 웹 에이전트의 DefaultAgentName 또는 AgentName 매개 변수 값은 정책 서버에 정의된 에이전트 항목의 이름과 일치해야 합니다.

관리 UI 를 사용하여 정책 서버에서 에이전트를 정의합니다. "에이전트 속성" 대화 상자의 "이름" 필드에 입력하는 값은 웹 에이전트가 로컬로 구성되었는지(에이전트 구성 파일) 아니면 정책 서버를 사용하여 중앙에서 구성되었는지(에이전트 구성 개체)에 따라 DefaultAgentName 또는 AgentName 설정에 정의된 이름과 일치해야 합니다.

에이전트 이름 암호화

기본적으로 웹 에이전트는 사용자를 SSL, 양식 또는 NTC(NTLM Credential Collector)로 리디렉션하는 URL 에 에이전트 이름을 추가합니다. EncryptAgentName 매개 변수를 사용하면 에이전트가 URL 에 이름을 암호화하는지 여부와 자격 증명 수집기가 URL 을 받을 때 이름의 암호를 해독할지 여부를 제어할 수 있습니다.

EncryptAgentName 매개 변수의 기본 설정은 yes 입니다. 다음에 해당하는 경우 이 매개 변수를 no 로 설정해야 합니다.

- 타사 응용 프로그램이 자격 증명 수집기를 사용 중이고 처리를 위해 에이전트 이름을 읽을 수 있어야 하는 경우
- 양식 인증을 위해 웹 에이전트를 FCC(양식 자격 증명 수집기)로 구성하고 인증할 단일 리소스로 사용자를 안내하는 경우. 단일 리소스 대상을 구성하려면 암호화되지 않은 에이전트 이름이 필요합니다.

웹 에이전트 이름을 암호화하려면 EncryptAgentName 매개 변수를 yes 로 설정하십시오.

추가 정보

단일 리소스 대상을 사용하도록 FCC 구성 (페이지 181)

웹 에이전트와 정책 서버의 통신을 관리하는 방법

다음 절차를 사용하여 에이전트와 정책 서버 간의 통신을 관리할 수 있습니다.

- 네트워크 대기 시간 문제 조정 (페이지 62)
- 웹 서버 인스턴스가 여러 개인 경우 에이전트 관리 (페이지 63)

추가 정보:

웹 에이전트 모니터링 (페이지 388)

네트워크 대기 시간 조정

네트워크 대기 시간 문제가 있으면 웹 에이전트에서 정책 서버와 연결할 수 없습니다. 이런 문제가 나타나지 않게 하려면 에이전트 구성 개체 또는 로컬 구성 파일에서 다음 매개 변수를 사용하십시오.

AgentWaitTime

에이전트가 LLAWP(Low-level Agent Worker Process)를 사용할 수 있을 때까지 대기하는 시간(초)을 지정합니다. 이 간격이 만료되면 에이전트는 정책 서버에 연결하려고 합니다.

이 매개 변수를 설정하면 LLAWP 연결과 관련된 에이전트 시작 오류를 해결하는 데 도움이 됩니다. 기본값에서 시작하여 에이전트가 성공적으로 시작될 때까지 한 번에 5초씩 간격을 늘리는 것이 좋습니다.
로컬 구성을 사용하는 경우 에이전트 구성 개체 *대신* WebAgent.conf 파일에서 이 매개 변수를 설정하십시오.

기본값: 5

예: 다음 수식으로 권장 값을 계산하십시오.

(*정책 서버 수* x 30) + 10 = AgentWaitTime 매개 변수 값(초)

예를 들어 정책 서버가 다섯 개이면 AgentWaitTime 매개 변수의 값을 160 으로 설정하십시오. [(5x30) + 10 = 160](초)

제한: (FIPS 호환 및 FIPS 마이그레이션 모드) 5 이상

제한: (FIPS 전용 모드) 20 이상

네트워크 대기 시간 문제가 있는 *경우에만* 값을 높게 설정하십시오. 설정값이 높으면 웹 서버에서 예기치 않은 동작이 발생할 수 있습니다.

네트워크 대기 시간을 조정하려면 에이전트 구성 개체 또는 로컬 구성 파일에서 AgentWaitTime 매개 변수를 사용하도록 설정한 후 원하는 시간(초)을 지정하십시오.

웹 서버 인스턴스가 여러 개인 경우 웹 에이전트 관리

여러 웹 서버 인스턴스에서 웹 에이전트를 구성하는 경우 각 서버 인스턴스마다 자체의 웹 에이전트 캐시, 로그 파일 및 건전성 모니터링 리소스가 필요합니다. 해당 리소스가 고유한지 확인하려면 WebAgent.Conf 파일에 다음 매개 변수를 구성하십시오.

ServerPath

웹 서버 인스턴스를 여러 개 사용하도록 웹 에이전트가 구성된 경우 각 웹 서버 인스턴스에 대한 고유 경로를 지정합니다. ServerPath 는 에이전트에서 사용하는 캐싱, 로깅 및 건전성 모니터링 리소스에 대한 고유 식별자를 생성합니다.

이 값은 시스템에서 실행되는 서버 인스턴스 간에 고유한 영숫자여야 합니다. 예를 들어 서버 인스턴스가 두 개 있는 경우 한 인스턴스의 ServerPath 매개 변수 값은 MyAgent1 이고 다른 인스턴스의 값은 MyAgent2 일 수 있습니다.

기본값: 비어 있음

예: 각각 에이전트를 로드하는 웹 서버 인스턴스가 네 개 있는 경우 각 서버 인스턴스의 WebAgent.conf 파일에서 ServerPath 매개 변수를 고유한 값으로 설정합니다. ServerPath 매개 변수를 각 서버 인스턴스의 로그 파일이 저장된 디렉터리(예: *server_instance_root*/logs)로 설정할 수 있습니다.

여러 서버 인스턴스에서 웹 에이전트를 구성하려면 각 WebAgent.conf 파일에서 ServerPath 매개 변수에 고유 경로를 추가하십시오.

Windows 시스템에 대한 ServerPath 매개 변수 설정

Windows 운영 환경에 서버 인스턴스가 여러 개 있는 경우 WebAgent.conf 파일에 다음 매개 변수의 값을 지정합니다.

ServerPath

웹 서버 인스턴스를 여러 개 사용하도록 웹 에이전트가 구성된 경우 각 웹 서버 인스턴스에 대한 고유 경로를 지정합니다. ServerPath 는 에이전트에서 사용하는 캐싱, 로깅 및 건전성 모니터링 리소스에 대한 고유 식별자를 생성합니다.

이 값은 시스템에서 실행되는 서버 인스턴스 간에 고유한 영숫자여야 합니다. 예를 들어 서버 인스턴스가 두 개 있는 경우 한 인스턴스의 ServerPath 매개 변수 값은 MyAgent1 이고 다른 인스턴스의 값은 MyAgent2 일 수 있습니다.

기본값: 비어 있음

예: 각각 에이전트를 로드하는 웹 서버 인스턴스가 네 개 있는 경우 각 서버 인스턴스의 WebAgent.conf 파일에서 ServerPath 매개 변수를 고유한 값으로 설정합니다. ServerPath 매개 변수를 각 서버 인스턴스의 로그 파일이 저장된 디렉터리(예: *server_instance_root*/logs)로 설정할 수 있습니다.

참고: ServerPath 매개 변수에 지정하는 문자열에 백슬래시(\)를 사용하지 마십시오. 다른 문자는 모두 허용됩니다.

다음 Windows 플랫폼에는 ServerPath 매개 변수가 필요하지 *않습니다*.

- IIS 서버(서버 인스턴스가 항상 하나만 있음)
- Apache 2.0 서버(웹 서버 인스턴스가 하나뿐인 경우). 이러한 시스템에서 매개 변수가 지원되지만 아무 의미가 없습니다.
- Oracle iPlanet 또는 Domino 웹 서버

  이러한 서버는 Windows 의 다중 프로세스 모드에서 실행되지 *않습니다*.

UNIX 시스템에 대한 ServerPath 매개 변수 설정

ServerPath 매개 변수는 WebAgent.conf 파일에 있습니다.

웹 서버가 UNIX 플랫폼에 설치된 경우에는 각 서버 인스턴스에서 고유한 에이전트 리소스를 사용하는 것이 좋습니다.

UNIX 의 다음 서버에 대한 ServerPath 매개 변수를 설정합니다.

- Apache 2.0(IBM HTTP Server 와 같은 Apache 2.0 기반 서버 포함)
- Oracle iPlanet 웹 서버 인스턴스

참고: UNIX 시스템의 Domino 웹 서버에는 ServerPath 가 필요하지 *않습니다*.

ServerPath 매개 변수의 설정값은 시스템에서 실행되는 서버 인스턴스 간에 고유한 영숫자여야 합니다. 예를 들어 서버 인스턴스가 두 개 있는 경우 한 인스턴스의 ServerPath 매개 변수 값은 MyAgent1 이고 다른 인스턴스의 값은 MyAgent2 일 수 있습니다.

ServerPath 매개 변수가 필요한 추가 구성

아래에서는 ServerPath 매개 변수가 필요한 다른 사례를 설명합니다.

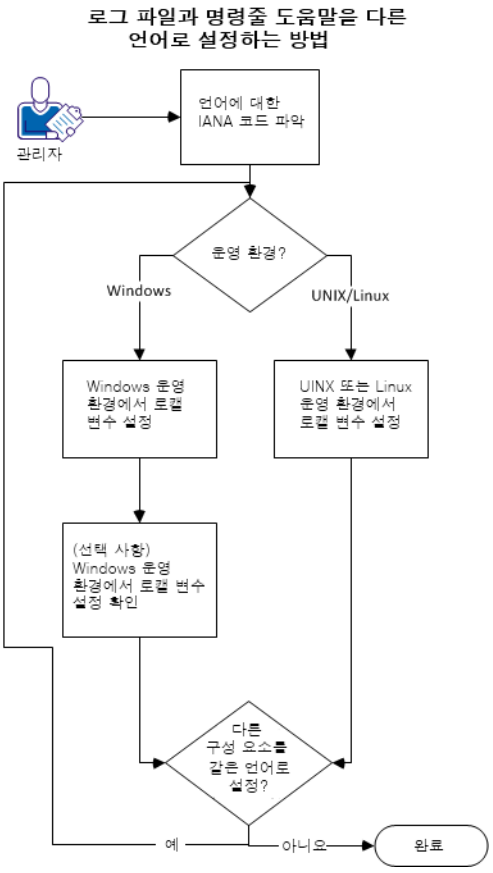
- 웹 에이전트가 세마포를 사용하여 공유 메모리를 추적하는 경우. 세마포는 운영 체제 또는 커널 저장소의 값입니다. 시스템에서 실행되는 프로세스는 이 값을 검사하여 리소스 가용성을 확인합니다. 세마포는 고유하지 않으므로 여러 에이전트에서 같은 메모리 영역을 가리킬 수 있습니다. 서버 경로 이름을 지정하면 인스턴스의 루트가 제공되고 에이전트는 세마포의 고유 키를 생성하는 데 사용되는 파일을 찾습니다.
- Windows 를 제외한 모든 플랫폼에서 서버 인스턴스가 여러 개 있는 경우 에이전트는 다음 작업 중 하나를 실행할 수 없습니다.
  - AgentName 매개 변수의 값 암호화(00-0012 오류)
  - SMSESSION 또는 SMIDENTITY 쿠키 암호화
  - 에이전트가 시작될 때 에이전트 암호화 키 업데이트
- Windows 를 제외한 모든 플랫폼에서 Apache 를 사용하는 경우 에이전트는 Apache 가 다시 시작될 때 6 개의 공유 메모리 세그먼트(세마포)를 해제하지 않습니다.
- 같은 시스템에서 서로 다른 유형의 웹 서버(예: Apache 2.0 서버와 Oracle iPlanet 서버)에 대해 각 웹 에이전트가 구성되는 경우. 이런 경우에는 각 서버의 구성에 대해 고유한 ServerPath 값을 지정해야 합니다. 서로 다른 유형의 웹 서버에서 에이전트 리소스를 공유할 수 *없습니다*.

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]
- 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 68).
2. 다음 절차 중 하나를 사용하여 운영 환경에 대한 환경 변수를 만듭니다.
   - Windows 운영 환경에서 로캘 변수를 설정합니다 (페이지 70).
   - UNIX 또는 Linux 운영 환경에서 로캘 변수를 설정합니다 (페이지 72).
3. (선택 사항) Windows 운영 환경에서 로캘 변수 설정을 확인합니다 (페이지 71).
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 70).
- UNIX 또는 Linux 운영 환경에서 로캘 변수를 설정합니다 (페이지 72).

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 68)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 72)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 5 장: 웹 에이전트 시작 및 중지

추가 정보:

WebAgent.conf 파일 위치 (페이지 36)

웹 에이전트 사용

에이전트 매개 변수를 구성한 후 에이전트에서 웹 서버의 리소스를 보호할 수 있도록 설정합니다.

참고: 리소스를 보호하려면 SiteMinder 정책 서버에도 정책을 정의해야 합니다.

다음 단계를 수행하십시오.

1. 텍스트 편집기에서 WebAgent.conf 파일을 엽니다.

   참고: 64 비트 운영 환경에 설치된 IIS 용 에이전트의 WebAgent.conf 파일은 *두 개*입니다. 이 중 한 파일은 32 비트 Windows 응용 프로그램용입니다. 다른 파일은 64 비트 응용 프로그램용입니다. IIS 용 에이전트를 시작 또는 중지할 때 *두 개의* WebAgent.conf 파일을 모두 수정하십시오.

2. EnableWebAgent 매개 변수의 값을 yes 로 변경합니다.
3. WebAgent.conf 파일을 저장한 후 닫습니다.
4. 웹 서버를 다시 시작합니다. 웹 서버가 실행되는 컴퓨터가 아니라 웹 서버 자체를 다시 시작해야 합니다.

   웹 에이전트를 사용할 수 있습니다.

웹 에이전트 사용 안 함

웹 에이전트가 더 이상 웹 서버의 리소스를 보호하지 않고 정책 서버와 통신하지 않게 하려면 해당 웹 에이전트가 사용되지 않도록 설정해야 합니다.

다음 단계를 수행하십시오.

1. 텍스트 편집기에서 WebAgent.conf 파일을 엽니다.

   참고: 64 비트 운영 환경에 설치된 IIS 용 에이전트의 WebAgent.conf 파일은 *두 개*입니다. 이 중 한 파일은 32 비트 Windows 응용 프로그램용입니다. 다른 파일은 64 비트 응용 프로그램용입니다. IIS 용 에이전트를 시작 또는 중지할 때 *두 개의* WebAgent.conf 파일을 모두 수정하십시오.

2. EnableWebAgent 매개 변수의 값을 no 로 변경합니다.
3. WebAgent.conf 파일을 저장한 후 닫습니다.
4. 웹 서버를 다시 시작합니다. 웹 서버가 실행되는 컴퓨터가 아니라 웹 서버 자체를 다시 시작해야 합니다.

   웹 에이전트를 사용할 수 없습니다.

추가 정보:

WebAgent.conf 파일 위치 (페이지 36)

apachectl 명령을 사용하여 대부분의 Apache 기반 에이전트 시작 또는 중지

UNIX 또는 Linux 운영 환경에서 apachectl 명령을 사용하여 대부분의 Apache 기반 에이전트를 시작하거나 중지하려면 먼저 제품에 대한 환경 변수를 설정해야 합니다.

참고: Apache 기반 에이전트는 apachectl -restart 옵션을 지원하지 *않습니다*. 이 절차는 Apche 기반 IBM HTTP 서버에 적용되지 *않습니다*. 대신 이 절차를 사용하십시오.

다음 단계를 수행하십시오.

1. UNIX/Linux 운영 환경의 경우 다음 스크립트를 실행하여 환경 변수를 설정합니다.

   ```
   ./ca_wa_env.sh
   ```

2. 다음 명령 중 *하나*를 사용합니다.

   ```
   apachectl -stop
   ```

   ```
   apachectl -start
   ```

제 6 장: 사용자 보호

이 섹션은 다음 항목을 포함하고 있습니다.

에이전트에서 정책 또는 키 업데이트 검사 간격 변경

웹 에이전트는 정책 서버를 정기적으로 폴링하여 다음 항목을 검사합니다.

- 업데이트된 관리 정보
- 업데이트된 정책
- 동적으로 업데이트된 에이전트 키

다음 매개 변수를 사용하여 필요에 따라 이 간격을 변경할 수 있습니다.

PSPollInterval

웹 에이전트가 정책 서버에 연결하여 정책 변경 내용이나 동적으로 업데이트된 키에 대한 정보를 검색하는 빈도(초)를 지정합니다. 높은 값(긴 간격)은 네트워크 트래픽을 감소시킵니다. 낮은 값(짧은 간격)은 네트워크 트래픽을 증가시킵니다.

기본값: 30

제한: 1

웹 에이전트가 정책 서버에서 업데이트를 확인하는 간격을 변경하려면 PSPollInterval 매개 변수에서 시간(초)을 변경하십시오.

중요! PSPollInterval 매개 변수를 늘리면 에이전트가 SiteMinder 정책 변경을 실행하는 시간에도 영향을 줍니다. 예를 들어 해고된 직원에 대한 액세스 권한을 해지하도록 10 시 30 분에 정책을 변경하고 PSPollInterval 매개 변수의 값이 3600(초)이라고 가정합니다. 웹 에이전트는 11 시 30 분이 될 때까지 변경된 정책을 실행하지 않습니다.

추가 정보:

웹 에이전트 및 동적 키 롤오버 (페이지 25)

사용자 추적 및 URL 모니터링

SiteMinder 에이전트는 다음 절차에 설명된 매개 변수를 사용하여 사용자를 추적하고 URL 을 모니터링할 수 있습니다.

익명 영역에서 사용자 아이덴티티 추적

익명 사용자가 리소스에 액세스하면 해당 사용자에게 SMIDENTITY(익명) 쿠키가 할당됩니다. 사용자가 다른 도메인으로 이동하면 해당 사용자에게 인증이 요청되고 로그인이 성공한 후 SMSESSION(로그인) 쿠키가 할당됩니다.

이 사용자가 보호된 "익명" 리소스, 즉 사용자가 자격 증명을 제공할 필요가 없는 영역의 리소스에 액세스하는 경우 해당 사용자는 사용자에 대한 두 쿠키를 모두 포함하는 도메인에 들어갈 수 있습니다. 5.x QMR 3 이상 웹 에이전트로 보호되는 리소스의 경우 웹 에이전트는 SMIDENTITY 쿠키가 아니라 SMSESSION 쿠키를 사용하여 사용자를 식별합니다.

완전히 업그레이드된 도메인에서 이전 버전의 에이전트가 SMIDENTITY 쿠키를 사용하여 사용자를 식별하는 도메인으로 사용자가 이동하는 경우에는 요청을 처리하는 웹 에이전트의 버전에 따라 사용되는 쿠키가 다릅니다.

마스터 쿠키 도메인에 보호된 리소스가 포함되고 보조 도메인에 익명 리소스가 포함되는 형식으로 쿠키 도메인이 분리되어 있는 경우 다음과 같은 태스크를 수행하는 사용자는 익명 도메인에서 익명 사용자로 간주됩니다.

1. 먼저 익명 도메인에 액세스합니다.
2. 마스터 도메인으로 이동한 후 로그인합니다.
3. 다시 익명 도메인으로 이동합니다.

감사를 사용하여 사용자 활동 또는 응용 프로그램 사용 추적

감사를 사용하면 웹 사이트에서 응용 프로그램의 사용 빈도를 측정하거나 사용자 활동을 추적할 수 있습니다. 감사는 다음 매개 변수로 제어됩니다.

EnableAuditing

웹 에이전트가 사용자 세션 캐시에 저장된 모든 성공한 권한 부여를 로깅할지 여부를 지정합니다. 사용하도록 설정하면 웹 에이전트가 정책 서버에 연결하지 않고 캐시에서 정보를 사용하는 경우에도 사용자 권한 부여 정보가 로깅됩니다. 웹 에이전트는 사용자가 리소스에 액세스할 때 사용자 이름 및 액세스 정보를 네이티브 웹 서버 로그 파일에 로깅합니다.

감사를 사용하여 사용자 활동 또는 응용 프로그램 사용을 추적하려면 이 매개 변수의 값을 yes 로 설정하십시오.

기본값: No

정책 서버와 웹 에이전트는 사용자 활동을 감사합니다. 웹 에이전트는 캐시에서 사용자에게 리소스에 액세스할 수 있는 권한이 부여될 때마다 계정 서비스에 메시지를 보냅니다. 이로 인해 계정 서비스는 웹 에이전트 및 정책 서버에 대한 성공적인 권한 부여를 추적할 수 있습니다. 웹 에이전트가 계정 서비스에 권한 부여 감사 메시지를 보낼 수 없으면 해당 리소스에 대한 액세스가 거부됩니다. 이 경우 관리 UI 에서 활동 보고서를 실행할 수 있습니다. 정책 서버의 보고서에는 각 세션에 대한 사용자 활동이 표시됩니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

URL 모니터링 개요

웹 에이전트는 웹 사이트의 정상적인 작동을 중지시키고 사이트의 보안 메커니즘을 피해 잘못된 방법으로 정보에 액세스하려는 악의적인 사용자의 공격을 방지할 수 있습니다.

웹 에이전트는 리소스 요청의 URL 을 모니터링하고 이러한 리소스에 대해 보안 정책을 적용합니다. SiteMinder 웹 에이전트에서 URL 을 해석하고 구문 분석하는 방식은 리소스가 있는 웹 서버의 방식과 다릅니다. 이러한 차이 때문에 성능 및 보안 문제가 발생하고 권한이 없는 사용자가 리소스에 액세스하는 상황이 발생할 수 있습니다. 웹 사이트를 디자인하고 SiteMinder 웹 에이전트를 구성할 때 이러한 문제를 고려해야 합니다.

공격 방지

SiteMinder 에이전트는 다음 절차에 설명된 매개 변수를 사용하여 공격을 방지할 수 있습니다.

교차 사이트 스크립팅에 대해 웹 사이트 보호

일반적으로 post 데이터 또는 URL 의 쿼리 매개 변수 데이터와 같이 브라우저에 입력한 텍스트가 문자(브라우저에 표시되면 유효한 실행 스크립트를 생성할 수 있음) 필터링 단계를 거치지 않고 응용 프로그램에 의해 표시되는 경우 CSS(교차 사이트 스크립팅) 공격이 발생할 수 있습니다.

아무 의심 없는 사용자에게 공격 URL 이 제공될 수 있습니다. 사용자가 이 URL 을 클릭하면 응용 프로그램에서 쿼리 문자열의 잘못된 매개 변수에 대한 오류 메시지와 입력 문자를 포함하는 화면 표시를 브라우저에 반환할 수 있습니다. 브라우저에 이러한 매개 변수가 표시되면 원하지 않는 스크립트가 브라우저에서 실행될 수 있습니다.

예를 들어 사용자가 검색 엔진 웹 페이지에 news 를 입력하면 일반적으로 응용 프로그램에서 빈 필드 또는 다음과 같은 응답을 반환할 수 있습니다.

```
Your search for news returned the following:
```

공격 URL 에 대해 브라우저는 다음과 같은 응답을 받을 수 있습니다.

```
news<script>BadProgram</script>
```

BadCSSChars 매개 변수는 ASCII 문자로 입력된 큰따옴표(")를 해석하지 않습니다. 큰따옴표를 잘못된 교차 사이트 스크립팅 문자로 포함하려면 ASCII 문자의 16 진수 값, 즉 %22 를 입력해야 합니다. 예를 들면 다음과 같습니다.

```
BadCSSChars="%22"
```

웹 에이전트 FCC 페이지에서 교차 사이트 스크립팅 공격 방지

웹 에이전트 FCC 페이지에 대해 교차 사이트 스크립팅 공격을 방지하려면 HTML 인코딩을 사용하여 FCC 변수 데이터가 올바르게 렌더링되도록 해야 합니다.

HTML 인코딩을 사용하면 문자가 HTML 구문이 아니라 리터럴 값으로 처리됩니다. 인코딩은 유해한 교차 사이트 스크립팅 구문이 표시되어야 할 때 이를 리터럴 텍스트로 렌더링하여 브라우저에서 HTML 양식을 렌더링하는 동안 코드가 실행되지 않도록 합니다. 공격에 악용될 수 있는 모든 구문을 인코딩할 수 있습니다.

fcchtmlencoding 매개 변수를 설정하면 에이전트는 다음과 같은 구문의 FCC 변수에 삽입된 모든 값에 HTML 인코딩 알고리즘을 적용하게 됩니다.

```
$$varname$$
```

일반적으로 차단되는 문자가 FCC 데이터에 필요한 경우에는 fcchtmlencoding 매개 변수를 사용하도록 설정하십시오.

fcchtmlencoding

웹 에이전트 FCC 페이지에 대해 교차 사이트 스크립팅 공격을 방지하기 위해 HTML 인코딩을 사용할지 여부를 지정합니다. 이 매개 변수는 문자를 차단하지 않습니다.

값: Yes, No

기본값: No

fcchtmlencoding 매개 변수는 모든 FCC 양식의 모든 변수 대체 항목에 적용됩니다. 이 매개 변수를 사용하는 에이전트는 하나 이상의 FCC 양식을 제공할 수 있습니다.

FCC 파일의 특정 문자에 HTML 인코딩을 적용하려면 다음 매개 변수를 사용하십시오.

fcchtmlencodingchars

특정 문자 값을 가져와 HTML 인코딩을 적용한 후 실제 문자 값을 FCC 파일에서 인코딩된 값으로 대체합니다.

중요! fcchtmlencodingchars 매개 변수를 사용하려면 fcchtmlencoding 매개 변수가 no 로 설정되어야 합니다.

FCC 파일의 특정 변수에 HTML 인코딩을 적용하려면 다음 함수를 사용하십시오.

HTMLENCODE

특정 변수 값을 가져와 HTML 인코딩을 적용한 후 실제 변수 값을 FCC 파일에서 인코딩된 값으로 대체합니다.

HTMLENCODE 함수의 구문은 다음과 같습니다.

```
$$htmlencode(varname)$$
```

중요! HTMLENCODE 함수를 사용하려면 fcchtmlencoding 매개 변수가 No 로 설정되어야 합니다.

교차 사이트 스크립팅을 확인하도록 웹 에이전트 구성

실행 스크립트에 포함될 수 있는 문자가 URL 에 있는지 확인하도록 웹 에이전트에 지시하려면 CSSChecking 매개 변수를 yes 로 설정하십시오. 이 매개 변수가 사용되도록 설정하면 웹 에이전트는 쿼리 문자열을 포함한 전체 URL 을 검사하여 다음과 같은 기본 문자 집합의 이스케이프된 버전과 이스케이프되지 않은 버전이 있는지 확인합니다.

- 왼쪽 및 오른쪽 꺾쇠 괄호(< 및 >)
- 작은따옴표(')

유효한 대상 도메인 정의

유해한 웹 사이트로 사용자를 리디렉션할 수 있는 피싱 시도로부터 리소스를 보호하도록 SiteMinder 에이전트를 구성하려면 다음 구성 매개 변수를 설정합니다.

ValidTargetDomain

자격 증명 수집기가 사용자를 리디렉션할 수 있는 도메인을 지정합니다. URL 의 도메인이 이 매개 변수에 설정된 도메인과 일치하지 않으면 리디렉션이 거부됩니다.

기본값: No

FCC(양식 자격 증명 수집기)를 포함한 모든 고급 인증 체계에서 이 매개 변수를 지원합니다.

ValidTargetDomain 매개 변수는 처리하는 동안 유효한 대상 도메인을 식별합니다. 사용자가 리디렉션되기 전에 에이전트는 리디렉션 URL 의 값을 이 매개 변수의 도메인과 비교합니다. 이 매개 변수가 없으면 에이전트는 사용자를 임의의 도메인에 있는 대상으로 리디렉션합니다.

ValidTargetDomain 매개 변수는 유효한 각 도메인에 대해 하나씩 지정되는 여러 개의 값을 포함할 수 있습니다.

로컬 웹 에이전트 구성의 경우 다음과 같이 각 도메인에 대한 항목을 한 행에 하나씩 지정합니다.

```
validtargetdomain=".xyzcompany.com"

validtargetdomain=".abccompany.com"
```

교차 사이트 스크립팅 공격에 대해 J2EE 응용 프로그램 보호

요청에 비표준(너무 긴 형식) 유니코드 문자를 사용하여 공격자가 교차 사이트 스크립팅 보호를 바이패스하지 않도록 할 수 있습니다.

다음 단계를 수행하십시오.

1. CSSChecking 매개 변수의 값을 yes 로 설정합니다.
2. 다음 매개 변수의 값을 yes 로 설정합니다.

   DisallowUTF8NonCanonical

   공격자가 요청의 비표준(너무 긴 형식) 유니코드(utf-8) 문자를 전송하고 교차 사이트 스크립팅 보호를 바이패스하지 못하게 합니다. 이 매개 변수의 값이 yes 일 경우 에이전트는 비표준(너무 긴 형식) 유니코드 문자를 포함하는 URL 에 대한 요청을 차단합니다.

   기본값: No

기본 CSS 문자 집합 재정의

기본적으로 에이전트는 다음과 같은 기본 크로스 사이트 스크립팅 문자 집합을 검사합니다.

- 왼쪽 및 오른쪽 꺾쇠 괄호(< 및 >)
- 아포스트로피 '

이 기본 문자 집합을 무시하려면 원하는 문자 집합을 정의하는 BadCSSChars 에이전트 매개 변수를 지정하십시오.

BadCSSChars

지정된 경우, 기본 크로스 사이트 스크립팅 문자 집합 대시 선택한 문자를 사용합니다. 이때 원하는 문자의 전체 문자열을 포함해야 합니다.

기본값: (<,',>)

예: <,> (이 경우 에이전트는 왼쪽 및 오른쪽 꺾쇠 괄호만 검색)

제한:

- 문자를 글자 그대로 지정할 수 있습니다.
- 문자를 구분하는 데 사용되는 쉼표를 포함하여 최대 4096 자의 문자를 지정할 수 있습니다.
- 하이픈으로 구분된 문자 범위를 지정할 수 있습니다. 이때 사용되는 구문은 starting_character-ending_character 입니다. 예를 들어 a-z 를 문자 범위로 입력할 수 있습니다.
- URL 인코딩 값 %22 를 사용하여 따옴표(")를 지정하십시오. ASCII 를 사용하면 안 됩니다.

에이전트는 문자 집합과 관련된 문제를 발견하면 사용자에게 액세스 거부 메시지를 반환하고 에이전트 오류 로그에 다음과 같은 메시지를 로깅합니다.

Caught Possible Cross Site Scripting Violation in URL. Exiting with HTTP 403 ACCESS FORBIDDEN. (URL 에서 교차 사이트 스크립팅 위반을 발견했습니다. HTTP 403 액세스 금지 오류가 발생하여 종료합니다.)

일부 응용 프로그램에서는 웹 서버 플랫폼에 관계없이 쿼리 문자열에 따옴표를 사용해야 합니다. 예를 들어 iNotes Web Access 와 같은 일부 Domino 응용 프로그램에서는 작은따옴표를 사용해야 합니다.

쿼리 문자열에 따옴표가 필요한 응용 프로그램을 사용하려면 BadCssChars 매개 변수에서 따옴표를 제거하십시오.

잘못된 쿼리 문자 지정

URL 의 쿼리 문자열 부분에 특정 문자를 사용하지 않으려면 다음 매개 변수를 설정하십시오.

BadQueryChars

웹 에이전트가 URL 의 쿼리 문자열 부분('?' 뒤)에서 금지하는 문자를 지정합니다.

기본값: 비어 있음(쿼리 문자열에서 모든 문자가 허용됨)

제한:

- 영어 문자에는 기본 16 진수 숫자가 적용됩니다. 다른 언어의 경우 허용하려는 언어의 문자에 해당하는 16 진수 값을 제거하십시오. 이러한 언어의 예로는 포르투갈어(브라질), 프랑스어, 일본어, 중국어 등이 포함됩니다.
- 리터럴 문자를 지정할 수 있습니다. 또한 해당 문자를 URL 로 인코딩된 형식으로 입력할 수도 있습니다. 예를 들어 a 라는 문자를 그대로 입력하거나, 이 문자를 인코딩한 값인 %61 을 입력할 수 있습니다.
- 문자를 구분하는 데 사용되는 쉼표를 포함하여 최대 4096 자의 문자를 지정할 수 있습니다.
- 하이픈으로 구분된 문자 범위를 지정할 수 있습니다. 이때 사용되는 구문은 *starting_character-ending_character* 입니다. 예를 들어 a-z 를 문자 범위로 입력할 수 있습니다.
- 따옴표(")는 URL 인코딩 값 %22 를 사용하여 지정하십시오. ASCII 를 사용하면 *안 됩니다*.

예: %25 는 쿼리에서 URL 인코딩 문자를 금지합니다.

웹 에이전트는 URL 의 쿼리 문자열에 있는 문자를 BadQueryChars 매개 변수에 정의된 해당 문자의 ASCII 값과 비교하여 쿼리 문자열에서 금지된 문자를 검색합니다. 예를 들어 이 프로세스는 다음과 같이 진행됩니다.

1. BadQueryChars 매개 변수에 다음 예제와 같이 퍼센트 기호(%)의 URL 인코딩 값이 포함됩니다.

   %25

2. 웹 에이전트가 다음 쿼리 문자열이 포함된 HTTP 요청을 받습니다.

   xxx=%0d

3. 웹 에이전트가 이전 예제에 있는 URL 을 검사하지만 URL 인코딩 값을 디코딩하지 *않습니다*. 예를 들어 웹 에이전트는 이전 예(2 단계)를 캐리지 리턴이 *아닌* 리터럴 문자열 %0d 로 해석합니다.

4. 웹 에이전트가 BadQueryChars 매개 변수의 값을 검사하고 해당 ASCII 값으로 변환합니다. 예를 들어 1 단계의 %25 가 퍼센트 기호(%)로 변환됩니다.

5. 웹 에이전트가 URL 의 각 문자를 BadQueryChars 매개 변수에서 디코딩된 ASCII 값과 비교합니다.

6. ASCII 퍼센트 기호(%)가 다음 두 위치에 모두 있기 때문에 웹 에이전트가 요청을 차단합니다.

   - URL 의 쿼리 문자열
   - BadQueryChars 매개 변수에서 디코딩된(ASCII) 값

쿼리 문자열에서 특정 문자를 차단하려면 해당 문자를 포함하도록 BadQueryChars 매개 변수의 값을 설정하십시오.

잘못된 URL 문자 지정

URL 요청에 포함할 수 없는 문자 시퀀스 집합을 나열할 수 있습니다. 에이전트는 이러한 문자 집합을 잘못된 URL 문자로 간주합니다. 웹 에이전트는 이 목록에 있는 문자 또는 문자열이 포함된 URL 요청을 거부합니다. URL 에서 "?" 앞에 나오는 문자를 검사합니다. 악의적인 웹 클라이언트에서 이러한 문자를 사용하여 SiteMinder 규칙을 재정의할 수 있으므로 웹 에이전트는 해당 문자가 포함된 URL 요청을 거부합니다.

웹 에이전트가 잘못된 URL 문자를 포함하는 URL 요청을 거부하는 경우 웹 서버는 다음 메시지 중 하나를 사용하여 응답합니다.

- 내부 서버 오류
- 웹 페이지를 찾을 수 없음(404) 오류

에이전트에서 요청을 처리하는 방식을 보려면 웹 에이전트 로그를 확인하십시오.

다음 매개 변수를 사용하여 문자를 지정합니다.

BadUrlChars

URL 요청에 사용할 수 없는 문자 시퀀스를 지정합니다. 웹 에이전트는 URL 에서 "?" 앞에 나오는 문자를 이 매개 변수의 목록과 비교하여 검사합니다. 지정된 문자가 발견되면 웹 에이전트에서 요청을 거부합니다.

다음과 같은 문자를 지정할 수 있습니다.

- 백슬래시(\)
- 두 개의 슬래시(//)
- 마침표와 슬래시(./)
- 슬래시와 마침표(/.)
- 슬래시와 별표(/*)
- 별표와 마침표(*.)
- 물결표(~)
- %2d
- %20

- %00-%1f
- %7f-%ff
- %25

문자가 여러 개일 경우에는 쉼표로 구분하십시오. 공백을 사용하면 *안 됩니다*.

물음표(?)가 잘못된 URL 문자 앞에 오면 CGI 매개 변수에 잘못된 URL 문자를 사용할 수 있습니다.

기본값: 비활성화됨(모든 문자가 허용됨).

제한:

- 영어 문자에는 기본 16 진수 숫자가 적용됩니다. 다른 언어의 경우 허용하려는 언어의 문자에 해당하는 16 진수 값을 제거하십시오. 이러한 언어의 예로는 포르투갈어(브라질), 프랑스어, 일본어, 중국어 등이 포함됩니다.
- 리터럴 문자를 지정할 수 있습니다. 또한 해당 문자를 URL 로 인코딩된 형식으로 입력할 수도 있습니다. 예를 들어 a 라는 문자를 그대로 입력하거나, 이 문자를 인코딩한 값인 %61 을 입력할 수 있습니다.
- 문자를 구분하는 데 사용되는 쉼표를 포함하여 최대 4096 자의 문자를 지정할 수 있습니다.
- 하이픈으로 구분된 문자 범위를 지정할 수 있습니다. 이때 사용되는 구문은 *starting_character-ending_character* 입니다. 예를 들어 a-z 를 문자 범위로 입력할 수 있습니다.
- 따옴표(")는 URL 인코딩 값 %22 를 사용하여 지정하십시오. ASCII 를 사용하면 *안 됩니다*.

잘못된 URL 문자를 지정하려면 차단할 문자를 포함하도록 BadURLChars 매개 변수의 값을 편집하십시오.

참고: Apache 2.0 리버스 프록시 서버 및 OWA(Outlook Web Access)를 구성하는 경우에는 BadURLChars 매개 변수가 사용되지 않도록 설정해야 합니다. OWA 에서는 BadURLChars 매개 변수에 포함될 수 있는 문자를 제한 없이 전자 메일 제목에 허용합니다.

잘못된 양식 문자 사용

다음 문자는 일반적으로 교차 사이트 스크립팅 공격에 사용됩니다.

- 왼쪽 및 오른쪽 꺾쇠 괄호(< >)
- 앰퍼샌드(&)
- 따옴표(")

인증 챌린지 중에 사용자에게 양식을 표시하기 위한 스크립팅 코드를 사용하려면 다음 매개 변수를 활성화하여 웹 에이전트가 특수 문자를 HTML 양식으로 보내기 전에 모든 특수 문자를 차단하도록 구성하십시오.

BadFormChars

양식에서 출력으로 사용하기 전에 웹 에이전트가 차단하는 문자를 지정합니다. 활성화되고 URL 의 에이전트 이름 부분에 이 매개 변수에 지정된 하나 이상의 문자가 있는 경우 로그인 페이지가 다음 오류 메시지를 반환합니다.

내부 서버 오류

기본값: Disabled (문자가 차단되지 않음)

예: <, >, &, %22

제한:

- 문자를 글자 그대로 지정할 수 있습니다.
- 문자를 구분하는 데 사용되는 쉼표를 포함하여 최대 4096 자의 문자를 지정할 수 있습니다.
- 하이픈으로 구분된 문자 범위를 지정할 수 있습니다. 이때 사용되는 구문은 starting_character-ending_character 입니다. 예를 들어 a-z 를 문자 범위로 입력할 수 있습니다.
- URL 인코딩 값 %22 를 사용하여 따옴표(")를 지정하십시오. ASCII 를 사용하면 안 됩니다.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그인합니다.
2. 이 매개 변수를 활성화하려는 에이전트 구성 개체를 엽니다.
3. BadFormChars 매개 변수 앞에 있는 # 문자를 제거하여 이 매개 변수가 사용되도록 설정합니다.

   BadFormChars 매개 변수가 기본값을 사용하여 활성화되었습니다.

4. (선택 사항) 사용하지 않을 문자를 목록에서 모두 제거합니다. 이 목록에 다른 문자를 추가할 수 있습니다. 문자는 쉼표를 사용하여 각각 구분해야 합니다.

DNS 서비스 거부 공격 방지

웹 서버에 IP 주소가 잘못된 HTTP 요청이 전달되는 경우 웹 에이전트는 해당 IP 주소를 정규화된 도메인 이름으로 확인하려고 합니다. HTTP 요청의 양이 많은 경우 서비스 거부 상태는 웹 에이전트 및 DNS 서버에도 영향을 줄 수 있습니다. 다음 매개 변수는 웹 에이전트가 DNS 조회를 수행할지 여부를 제어합니다.

DisableDNSLookup

웹 에이전트에서 DNS 조회를 수행하지 않도록 설정합니다.

다음 단계를 수행하십시오.

1. DisableDNSLookup 매개 변수가 s 로 끝나지 *않는지* 확인합니다. ACO 템플릿과 LocalConfig.conf 파일의 일부 이전 버전에서 이 오류가 나타날 수 있습니다. 올바른 매개 변수는 p 로 끝납니다.
2. DisableDNSLookup 매개 변수의 값을 yes 로 설정합니다.

중요! 이 매개 변수의 값이 yes 로 설정된 경우 쿠키 기반 기능이 제대로 작동하려면 정규화된 도메인 이름이 필요합니다.

확장명이 없는 리소스 보호

권한이 없는 사용자가 확장명이 없는 리소스에 액세스할 수 없도록 하려면 다음 매개 변수를 사용합니다.

OverrideIgnoreExtFilter

웹 에이전트에서 모든 URI 에 대해 일치하는 항목을 찾을 문자열 목록을 지정합니다. 이 매개 변수는 웹 에이전트가 일반적으로 확장명을 무시하는 리소스나 확장명이 없는 응용 프로그램이나 파일을 보호하는 데 유용합니다. URI 가 목록에 있는 문자열 중 하나와 일치하면 웹 에이전트는 정책 서버에 확인하여 리소스가 보호되는지 여부를 결정합니다.

정확한 경로보다는 좀 더 일반적인 문자열을 지정하는 것이 좋습니다. 또한 부분 문자열을 추가하여 리소스 그룹을 보호할 수도 있습니다. 예를 들어 /servlet/ 문자열은 다음 리소스를 보호합니다.

- /dira/app1/servlet/app
- /dirb/servlet/app1
- /dirc/mydir/servlet/app2

기본값: 기본값 없음

확장명이 없는 리소스를 보호하려면 보호할 리소스에 대한 문자열(마침표 제외)을 OverrideIgnoreExtFilter 매개 변수의 값에 추가합니다. 에이전트 구성 개체를 사용하는 경우 다중값 옵션을 사용하여 문자열을 추가합니다. 로컬 구성 파일을 사용하는 경우에는 각 문자열을 한 행에 하나씩 추가합니다.

POST 보존 사용 안 함

POST 보존을 사용할 필요가 없는 경우 다음 매개 변수로 이를 사용하지 않도록 설정할 수 있습니다.

PreservePostData

웹 에이전트가 요청을 리디렉션할 때 POST 데이터를 유지할지 여부를 지정합니다. 사용자가 양식 또는 인증서 인증과 같은 고급 인증을 요청받을 경우 인증 단계 중에 POST 데이터가 유지됩니다.

기본값: Yes

POST 보존이 사용되지 않도록 설정하려면 PreservePostData 매개 변수의 값을 no 로 설정하십시오.

응용 프로그램 보안

권한이 없는 사용자가 웹 에이전트에서 무시하도록 구성된 확장명을 포함하는 잘못된 파일 이름을 URL 끝에 추가할 수 있습니다. 그러면 에이전트는 권한이 없는 사용자가 리소스에 액세스할 수 있도록 허용합니다. 이러한 경우 웹 에이전트에서 액세스를 거부하도록 설정하려면 다음 매개 변수를 사용하십시오.

SecureApps

에이전트가 권한이 없는 사용자의 URL 에 권한을 부여하지 않도록 합니다. 특정 확장명으로 끝나는 파일에 대한 요청을 무시하도록 웹 에이전트가 구성된 경우 공격자가 가짜 URL 을 생성하여 리소스에 액세스하려고 할 수 있습니다.

예를 들어 다음과 같은 URL 의 리소스가 있을 수 있습니다.

```
/scripts/myapp
```

공격자는 다음 예와 같은 가짜 URL 을 생성하여 액세스 권한을 얻으려고 할 수 있습니다.

```
/scripts/myapp/junk.jpg
```

.jpg 파일에 대한 요청을 무시하도록 웹 에이전트가 설정된 경우 SecureApps 매개 변수의 값을 no 로 설정하면 /scripts/myapp/junk.jpg 에 대한 요청에 자동으로 권한이 부여됩니다.

SecureApps 매개 변수의 값을 yes 로 설정하면 웹 에이전트가 해당 리소스가 유효한지 또는 URL 이 가짜인지 확인합니다.

기본값: No

응용 프로그램의 보안을 유지하려면 SecureApps 매개 변수의 값을 yes 로 설정하십시오.

사용자 지정 응답이 X-Frame Options 를 준수하도록 설정

웹 응용 프로그램에 X-Frame-Options 응답 헤더를 사용하는 경우 에이전트의 사용자 지정 응답이 이 헤더를 올바르게 반환하도록 설정할 수 있습니다. X-frame options 헤더의 설정은 브라우저가 <frame> 또는 <iframe> 태그 사이의 콘텐츠를 사용하여 페이지를 렌더링하는지 여부를 결정합니다.

다음 매개 변수를 사용하면 에이전트의 사용자 지정 응답이 X-frame-options 를 준수할지 여부를 결정할 수 있습니다.

XFrameOptions

사용자 지정 응답이 x-frame-options 응답 헤더를 준수하는지 여부를 지정합니다. 이 매개 변수를 설정하면 올바른 x-frame-options 헤더로 사용자 지정 응답이 설정됩니다.

기본값: None

예: SAMEORIGIN

사용자 지정 응답이 x-frame options 를 준수하도록 하려면 XFrameOptions 매개 변수의 값을 yes 로 설정하십시오.

IP 주소 확인

SiteMinder 에이전트는 다음 절차에 설명된 매개 변수를 사용하여 IP 주소를 확인할 수 있습니다.

- IP 주소로 에이전트 아이덴티티 확인 (페이지 97)
- IP 주소를 비교하여 보안 위반 방지 (페이지 98)

IP 주소로 에이전트 아이덴티티 확인

가상 웹 서버에서 IP 주소와 호스트 이름을 사용하여 에이전트 이름을 확인하는 경우 웹 에이전트에서 잘못된 AgentName 값을 사용하여 요청을 평가할 수 있습니다. 이로 인해 인증되지 않은 사용자가 보호된 리소스에 액세스할 수 있게 됩니다.

다음 매개 변수를 사용하면 웹 에이전트가 가상 서버의 물리적 IP 주소를 사용하여 에이전트 이름을 확인하도록 지정할 수 있습니다.

UseServerRequestIp

가상 웹 서버의 물리적 IP 주소에 따라 AgentName 을 확인하도록 웹 에이전트에 지시합니다. 웹 서버가 가상 서버 매핑에 IP 주소를 사용하는 경우 이 매개 변수를 통해 보안을 강화할 수 있습니다. 이 매개 변수를 no 로 설정하면 웹 에이전트가 클라이언트 요청의 HTTP 호스트 헤더에 있는 호스트 이름을 기반으로 AgentName 을 확인합니다.

Domino 서버의 경우 이 매개 변수는 Domino 6.x 에서만 지원됩니다. 다른 Domino 버전의 에이전트에 대해 이 매개 변수를 설정하면 웹 에이전트가 기본 에이전트 이름을 사용합니다.

SSL 통신 및 가상 호스트에 대해 구성된 IIS 웹 에이전트의 경우 이 매개 변수를 yes 로 설정해야 합니다. IIS 는 SSL 이 설정된 호스트 이름을 사용하는 가상 호스트 매핑을 허용하지 않습니다.

기본값: No

IP 주소를 사용하여 웹 에이전트의 아이덴티티를 확인하려면 UseServerRequestIp 매개 변수를 yes 로 설정하십시오.

IP 주소를 비교하여 보안 위반 방지

인증되지 않은 시스템에서 패킷을 모니터링하고 쿠키를 도용한 후 해당 쿠키를 사용하여 다른 시스템에 액세스할 수 있습니다. 인증되지 않은 시스템에 의해 보안이 침해되지 않게 하려면 영구 쿠키 및 임시 쿠키를 사용하여 IP 검사 기능을 설정하거나 해제할 수 있습니다.

IP 검사 기능을 사용하는 경우 에이전트는 마지막 요청의 쿠키에 저장된 IP 주소를 현재 요청에 포함된 IP 주소와 비교해야 합니다. 두 IP 주소가 일치하지 않으면 에이전트에서 요청을 거부합니다.

IP 검사를 구현하는 데 사용되는 두 매개 변수는 PersistentIPCheck 와 TransientIPCheck 입니다. 다음과 같이 매개 변수를 설정하십시오.

- PersistentCookies 가 사용되도록 설정한 경우 PersistentIPCheck 를 yes 로 설정합니다.
- PersistentCookies 가 사용되도록 설정하지 않은 경우 TransientIPCheck 를 yes 로 설정합니다.

SiteMinder 아이덴티티 쿠키는 IP 검사의 영향을 받지 않습니다.

추가 정보

영구 쿠키 설정 (페이지 103)
아이덴티티 쿠키 제어 (페이지 102)

SiteMinder 브라우저 쿠키

SiteMinder 에이전트와 관련된 쿠키를 관리하려면 다음 절차에 설명된 매개 변수를 사용하십시오.

기본 인증에 쿠키 필요

다음 매개 변수를 사용하여 SiteMinder 에 쿠키가 필요한지 여부를 제어할 수 있습니다.

RequireCookies

SiteMinder 에 쿠키가 필요한지 여부를 지정합니다. 다음과 같은 기능을 수행하려면 SiteMinder 에 쿠키가 필요합니다.

- 싱글 사인온 환경의 보안 설정
- 세션 시간 만료 적용
- 유휴 시간 만료 적용

이 매개 변수의 값이 yes 인 경우 에이전트에서 HTTP 요청을 처리하려면 다음 쿠키 중 하나가 필요합니다.

- SMCHALLENGE
- SMSESSION

이 매개 변수의 값이 no 인 경우 다음과 같은 상황이 발생할 수 있습니다.

- 예기치 않게 사용자에게 자격 증명이 요청됩니다.
- 시간 만료가 엄격하게 적용되지 않습니다.

중요! 에이전트에 쿠키가 필요한 경우 사용자에게 자신의 브라우저에서 HTTP 쿠키를 수락하도록 지시하십시오. 그렇지 않으면 사용자는 보호된 모든 리소스에 대해 액세스가 거부됩니다.

기본값: Yes

쿠키를 요청하려면 RequireCookies 매개 변수의 값을 yes 로 설정하십시오.

HTTP-Only 특성을 사용하여 쿠키의 정보 보호

교차 사이트 스크립팅 공격으로부터 보호하기 위해 다음 매개 변수를 사용하여 웹 에이전트가 자신이 생성하는 모든 쿠키에 대해 HTTP-Only 특성을 설정하도록 할 수 있습니다.

UseHTTPOnlyCookies

웹 에이전트가 생성하는 쿠키에 HTTP 전용 특성을 설정하도록 웹 에이전트에 지시합니다. 웹 에이전트가 이 특성이 있는 쿠키를 사용자의 브라우저에 반환하면 원래 이 쿠키를 설정한 웹 사이트의 스크립트도 쿠키의 내용을 읽을 수 없습니다. 따라서 스크립트를 통해 권한 없는 제 3 자에게 쿠키의 중요 정보를 보내지 못하게 할 수 있습니다.

기본값: No

쿠키의 정보를 보호하려면 UseHTTPOnlyCookies 매개 변수의 값을 yes 로 설정하십시오.

보안 쿠키 설정

다음 매개 변수를 사용하면 보안 연결(HTTPS)을 통해서만 요청 브라우저와 보호된 웹 서버 간에 세션 쿠키가 전송되도록 지정할 수 있습니다.

UseSecureCookies

보안 HTTPS 연결을 통해 웹 서버에 쿠키를 보냅니다. 이 매개 변수를 설정하면 브라우저와 웹 서버 간 보안이 강화됩니다.

이 설정이 활성화되어 있으면 SSO(싱글 사인온) 환경의 사용자가 SSL 웹 서버에서 비 SSL 웹 서버로 이동할 경우 다시 인증해야 합니다. 기존 HTTP 연결을 통해 보안 쿠키를 전달할 수 없습니다.

기본값: No

SSL 연결을 통해 쿠키를 전송하려면 UseSecureCookies 매개 변수를 yes 로 설정하십시오.

추가 정보:

여러 도메인에 보안 쿠키 설정 (페이지 260)

아이덴티티 쿠키 제어

TransientIDCookies 매개 변수는 에이전트 아이덴티티 쿠키(SMIDENTITY)가 임시 쿠키인지 아니면 영구 쿠키인지를 지정합니다.

영구 쿠키는 클라이언트 시스템의 하드 디스크에 작성됩니다. 5.x QMR1 이전의 웹 에이전트에서 영구 쿠키는 7 일 동안 유효했습니다. 또한 5.x QMR1 이상의 웹 에이전트에서 영구 쿠키는 구성된 최대 세션 만료 시간 이후 7 일 동안 유효합니다. 최대 세션 만료 시간은 관리 UI 에 설정됩니다. 일반적으로 영구 쿠키는 쿠키가 만료된 후 웹 브라우저의 쿠키 파일에서 삭제되지만 브라우저에서 영구 쿠키를 다르게 처리할 수 있습니다. 기본적으로 웹 에이전트는 영구 쿠키를 사용하지 않습니다. 대신 임시 쿠키를 사용합니다.

여러 개의 브라우저 세션에 싱글 사인온을 사용하려면 영구 쿠키를 사용합니다. 영구 쿠키를 설정하는 경우 사용자는 SiteMinder 세션이 만료되기 전에 브라우저 세션을 끝낸 후 새 브라우저 세션을 시작하고 싱글 사인온 기능을 사용할 수 있습니다.

영구 쿠키는 하드 디스크에 작성되지만 임시 쿠키는 하드 디스크에 작성되지 않으며 구성된 세션 만료 시간의 영향을 받지 않습니다. 임시 쿠키는 쿠키 폴더에 유지됩니다.

아이덴티티 쿠키를 영구 쿠키로 설정하려면 TransientIDCookies 를 no 로 설정합니다. 아이덴티티 쿠키를 임시 쿠키로 설정하려는 경우에는 기본값인 yes 로 두십시오.

해당하는 IP 검사를 설정해야 합니다.

추가 정보

IP 주소를 비교하여 보안 위반 방지 (페이지 98)

영구 쿠키 설정

여러 개의 브라우저 세션에 싱글 사인온을 사용하려면 영구 쿠키를 사용합니다. 다음 단계에서는 영구 쿠키를 사용하는 한 가지 경우를 보여 줍니다.

1. 사용자가 SiteMinder 에 인증하지만 SiteMinder 세션이 만료되기 전에 브라우저 세션을 끝냅니다.
2. 사용자가 나중에 새 브라우저 세션을 시작하지만 영구 쿠키가 싱글 사인온 기능을 유지 관리합니다.

영구 쿠키는 구성된 최대 세션 만료 시간 *이후* 7 일 동안 유효합니다. 대부분의 브라우저는 쿠키 만료 후 웹 브라우저의 쿠키 파일을 삭제합니다. 일부 브라우저의 경우 영구 쿠키를 다르게 처리할 수 있습니다.

다음 단계를 수행하십시오.

1. PersistentCookies 매개 변수를 yes 로 설정합니다.

   SMSESSION 은 영구 쿠키입니다.
2. TransientIDCookies 매개 변수를 no 로 설정합니다.

   SMIDENTITY 은 영구 쿠키입니다.

에이전트 쿠키의 쿠키 경로 지정

웹 에이전트는 쿠키를 생성할 때 자동으로 루트(/) 디렉터리를 쿠키 경로로 사용합니다. 쿠키의 도메인 및 경로 특성은 요청 URL 과 비교됩니다. 쿠키가 도메인 및 경로에 대해 유효한 경우 클라이언트는 해당 쿠키를 서버에 전송합니다. 쿠키 경로에 루트 값이 사용되는 경우 클라이언트는 해당 쿠키를 도메인의 모든 요청과 함께 서버에 전송합니다.

SiteMinder 쿠키를 특정 경로 집합으로 설정하면 보호되지 않는 리소스에 대해 쿠키가 전송될 경우 발생하는 웹 트래픽을 제거할 수 있습니다. 예를 들어 쿠키 경로를 /mypackage 로 설정하면 클라이언트는 도메인에서 특정 패키지의 요청에 대해서만 쿠키를 전송합니다.

에이전트 쿠키의 쿠키 경로를 지정하려면

1. 에이전트 구성 개체 또는 로컬 에이전트 구성 파일을 엽니다.
2. 다음 매개 변수를 사용하여 쿠키 공급자의 쿠키 경로를 설정합니다.

   MasterCookiePath

   쿠키 공급자가 생성한 기본 도메인 세션 쿠키의 경로를 지정합니다. 예를 들어 이 매개 변수를 /siteminderagent 로 설정하면 쿠키 공급자가 생성하는 모든 세션 쿠키의 경로가 /siteminderagent 가 됩니다. 쿠키 공급자 에이전트에서 이 매개 변수를 설정하지 않으면 기본값이 사용됩니다.

   기본값: /(루트)

3. 다음 매개 변수를 사용하여 보조 에이전트의 쿠키 경로를 설정합니다.

   CookiePath

   다음 보조 에이전트 브라우저 쿠키의 쿠키 경로를 지정합니다.

   - xxSESSION
   - xxIDENTITY
   - xxDOMINODATA
   - xxCHALLENGE(SSL_CHALLENGE_DONE 포함)
   - xxDATA
   - xxSAVEDSESSION

   예를 들어 이 매개 변수를 /BasicAuth 로 설정하면 이전 목록의 모든 보조 에이전트가 /BasicAuth 를 경로로 사용하여 생성됩니다. 지정하지 않을 경우 기본값이 사용됩니다.

4.x 에이전트와의 호환성을 유지하기 위해 CookiePath 가 자격 증명 쿠키(예: xxxxCRED)에 추가되지 *않습니다*.

다음 쿠키에서는 *항상* 루트(/) 경로를 사용합니다.

- ONDENIEDREDIR
- TRYNO

CookiePathScope 매개 변수가 0 보다 크면 CookiePath 매개 변수 설정이 무시됩니다.

기본값: /(루트)

4. (선택 사항) 웹 에이전트가 CookiePath 값을 사용하는 대신 URL 에서 쿠키 경로를 추출하도록 구성하려면 다음 매개 변수를 0 보다 큰 값으로 설정합니다.

CookiePathScope

다음 보조 에이전트 쿠키에 대한 쿠키 경로의 범위를 지정합니다.

- xxSESSION
- xxIDENTITY
- xxDOMINODATA
- xxCHALLENGE(SSL_CHALLENGE_DONE 포함)
- xxDATA
- xxSAVEDSESSION

이 매개 변수에서 0 보다 큰 CookiePathScope 를 사용하면 CookiePath 매개 변수의 설정이 무시됩니다.

기본값: 0

추가 정보:

전체 로그오프 구성 (페이지 268)

쿠키 도메인 적용

정규화된 도메인 이름을 사용하면 쿠키가 올바로 작동할 수 있습니다. 다음과 같은 조건을 충족하는 URL 요청의 호스트 이름에 에이전트가 쿠키 도메인을 추가하도록 지정할 수 있습니다.

- 요청에 도메인을 지정하지 않았습니다.
- 요청에 IP 주소만 포함되어 있습니다.
- 다음 매개 변수를 설정하여 에이전트에 쿠키 도메인을 사용하도록 지정할 수 있습니다.

ForceCookieDomain

도메인을 지정하지 않거나 IP 주소만 포함하는 URL 요청의 호스트 이름에 웹 에이전트가 쿠키 도메인을 추가하도록 합니다. 이 매개 변수는 추가 기능을 위해 ForceFQHost 매개 변수와 함께 사용됩니다.

기본값: No

ForceFQHost

에이전트가 정규화된 도메인 이름을 사용하도록 합니다. 이 매개 변수는 구성된 DNS(Domain Name System) 서비스를 사용하여 에이전트가 아니라 DNS 서비스를 통해 URL 요청의 호스트 이름에 쿠키 도메인을 추가합니다. 웹 에이전트는 부분 URL 을 포함하는 요청을 받으면 원래 URI 에 지정된 동일한 대상 리소스로 요청을 리디렉션합니다. 이러한 리디렉션 요청에는 에이전트에서 구성된 DNS 서비스를 사용하여 확인하는 정규화된 호스트 이름이 사용됩니다. 추가 기능을 위해 이 매개 변수를 ForceCookieDomain 매개 변수와 함께 사용하십시오.

기본값: No

예: 에이전트는 http://host1/page.html 요청을 받으면 http://host1.myorg.com/page.html 로 응답합니다. 또한 http://123.113.12.1/page.html 과 같은 요청을 받으면 http://host1.myorg.com/page.html 로 응답합니다.

참고: 이러한 예제를 사용하려면 올바른 DNS 조회표가 있어야 합니다. 요청에 포함된 부분 도메인 이름을 DNS 에서 확인할 수 없는 경우 오류가 발생할 수 있습니다.

다음 단계를 수행하십시오.

1. ForceCookieDomain 매개 변수의 값을 yes 로 설정합니다.
2. ForceFQHost 매개 변수의 값을 yes 로 설정합니다.

   필요한 경우 에이전트가 호스트 이름에 쿠키 도메인을 추가합니다.

쿠키 도메인 확인 구현

자동 도메인 확인을 구현하려면 CookieDomain 매개 변수를 주석으로 처리하거나 none 으로 설정하여 웹 에이전트가 쿠키를 생성할 때 해당 쿠키가 발급된 서버에만 유효하도록 지정하십시오.

CookieDomainScope 매개 변수에 값을 추가하여 쿠키 도메인을 세부적으로 정의할 수 있습니다. 범위는 도메인 이름을 구성하는 섹션의 수를 결정합니다. 이때 각 섹션은 마침표로 구분됩니다. 도메인은 항상 "."로 시작됩니다.

CookieDomainScope 값을 0 으로 설정하면 지정된 호스트에 대해 가장 특정한 범위를 사용하도록 에이전트에 지시합니다. 1(예: 쿠키 도메인이 .com 이 되는 경우)은 HTTP 사양에서 허용되지 않는 값입니다. 이 값을 2 로 설정하면 가장 일반적인 범위를 사용하도록 에이전트에 지시합니다.

다음 표에서는 몇 가지 도메인 이름과 CookieDomainScope 값을 보여 줍니다.

도메인 이름	쿠키 도메인 범위 값	쿠키 도메인
server.myorg.com	2	.myorg.com
server.division.myorg.com	3	.division.myorg.com
	2	.myorg.com
server.subdivision.division.myorg.com	4	.subdivision.division.myorg.com
	3	.division.myorg.com
	2	.myorg.com

예를 들어 division.myorg.com 도메인의 범위는 3 입니다. 기본적으로 웹 에이전트는 범위를 2 로 가정합니다. 쿠키 도메인의 범위는 1 이 될 수 없습니다.

CookiePathScope 설정의 작동 방식

다음 표에서는 CookiePathScope 매개 변수의 값이 아래 설정과 함께 사용되는 방식을 보여 줍니다.

- URL(예: http://*fqdn*/path1/path2/path3/path4/index.html)
- CookiePath 매개 변수 값 /BasicA

CookiePath 값	CookiePathScope 값	사용되는 경로
/BasicA	0	/BasicA
/BasicA	1	/Path1
/BasicA	2	/Path1/Path2
/BasicA	3	/Path1/Path2/Path3
/BasicA	4	/Path1/Path2/Path3/Path4
/BasicA	5	/Path1/Path2/Path3/Path4
/BasicA	99	/Path1/Path2/Path3/Path4
/ 또는 "정의되지 않음"	0	/
/ 또는 "정의되지 않음"	1	/Path1
/ 또는 "정의되지 않음"	2	/Path1/Path2
/ 또는 "정의되지 않음"	3	/Path1/Path2/Path3
/ 또는 "정의되지 않음"	4	/Path1/Path2/Path3/Path4
/ 또는 "정의되지 않음"	5	/Path1/Path2/Path3/Path4
/ 또는 "정의되지 않음"	99	/Path1/Path2/Path3/Path4

이러한 설정은 기본 SSO 에도 영향을 줍니다. 예를 들어 CookiePathScope 값을 1 이상으로 설정하면 경로가 /BasicA 인 SESSION 쿠키가 /BasicB/Index.html 요청에 대해 유효하지 않으므로 사용자는 /BasicA/Index.html 과 /BasicB/Index.html 둘 다에 대해 자격 증명이 요청됩니다.

SDK 타사 쿠키에 대한 지원 구성

SiteMinder 가 아닌 다른 웹 에이전트를 조직에 사용하는 경우 다음 매개 변수를 설정하여 싱글 사인온을 지원하도록 웹 에이전트를 구성할 수 있습니다.

AcceptTPCookie

웹 에이전트가 SiteMinder 가 아닌 타사 웹 에이전트에서 생성된 세션(SMSESSION) 쿠키를 수락하도록 합니다. 타사 에이전트는 SiteMinder SDK 를 사용하여 SMSESSION 쿠키를 생성하고 읽습니다.

기본값: 기본값 없음

참고: 자세한 내용은 프로그래밍 설명서를 참조하십시오.

웹 에이전트가 SiteMinder 가 아닌 다른 웹 에이전트에서 생성한 세션 쿠키를 수락하도록 설정하려면 AcceptTPCookie 매개 변수를 yes 로 설정하십시오.

HTTPS 포트 정의

요청의 보안을 위해 SSL 을 사용하여 웹 서버에 연결하는 경우(HTTPS) 다음 매개 변수를 사용하여 HTTPS 포트 번호를 지정합니다.

HttpsPorts

SSL 을 사용하여 웹 서버에 연결하는 경우 웹 에이전트가 수신 대기하는 보안 포트를 지정합니다. 이 매개 변수의 값을 지정하는 경우 보안 요청을 처리하는 모든 웹 서버의 모든 포트를 포함해야 합니다. 값을 지정하지 않으면 웹 에이전트는 웹 서버의 컨텍스트에서 HTTP 체계를 읽습니다.

HTTPS 를 HTTP 로 변환하는 HTTPS 가속기를 사용하는 서버의 경우에는 요청이 브라우저에서 SSL 연결로 처리됩니다.

기본값: 비어 있음

예: 443

예: (여러 포트) 443,7002

HTTPS 포트를 정의하려면 HttpsPorts 매개 변수의 값을 SSL 을 사용하는 포트 번호로 설정합니다. 여러 포트 번호를 구분하려면 쉼표를 사용하십시오.

URL 의 쿼리 데이터 디코딩

정책 서버를 호출하기 전에 URL 의 쿼리 데이터를 디코딩하여 정책 서버가 올바른 리소스를 인식하도록 웹 에이전트의 Base64 알고리즘을 구성하려면 다음 매개 변수를 사용합니다.

DecodeQueryData

웹 에이전트가 정책 서버를 호출하기 전에 URL 의 쿼리 데이터를 디코딩할지 여부를 지정합니다. 해당 환경에서 다음 태스크 중 하나를 수행해야 할 경우 이 매개 변수를 yes 로 설정하십시오.

- 규칙 필터가 적절한 문자열에 대해 작동하도록 해야 하는 경우
- 쿼리 문자열의 데이터에 대해 규칙을 작성해야 하는 경우

기본값: No

정책 서버를 호출하기 전에 웹 에이전트에서 URL 의 쿼리 데이터를 디코딩하도록 구성하려면 DecodeQueryData 매개 변수의 값을 yes 로 설정하십시오.

마침표 또는 확장명이 없는 리소스를 보호하는 방법

서블릿과 같은 일부 URL 에는 마침표가 없습니다. 확장명이 없는 URL 도 있습니다. 두 경우 모두 보안상 위험합니다. 아래에서는 이러한 위험에 대해 단계별로 설명합니다.

1. 환경에 보호된 리소스인 /mydir/servlets 디렉터리가 포함되어 있습니다.
2. 웹 에이전트는 확장명이 .gif 인 리소스에 대한 요청을 무시하도록 구성됩니다.
3. 권한이 없는 사용자가 다음과 같이 존재하지 않는 파일의 이름과 .gif 확장명을 URL 끝에 추가합니다.

   /mydir/servlets/file.gif
4. 웹 에이전트는 .gif 확장명을 무시하고 권한이 없는 사용자에게 /mydir/servelets 디렉터리에 대한 액세스 권한을 부여합니다.

보안 위험을 가장 중요하게 생각하는 경우에는 에이전트가 확장명을 무시하도록 설정하면 안 됩니다. 그러나 다음과 같은 결과를 고려해야 합니다.

- 웹 에이전트가 페이지의 모든 이미지 URL 을 평가하므로 성능이 저하될 수 있습니다.
- 이전에는 인증이 필요 없었던 리소스에 대해 사용자 인증이 요청되므로 웹 사이트의 동작이 변경될 수 있습니다.

다음 옵션은 마침표가 없는 URL 을 보호할 때 사용할 수 있습니다.

- OverrideIgnoreExtFilter 기능을 사용하도록 에이전트를 구성합니다.
- 보호된 리소스에 웹 에이전트에서 무시하도록 구성된 확장명이 없는지 확인합니다.

복잡한 URI 처리

DisableDotDotRule 매개 변수는 슬래시(/)로 구분된 점 두 개를 포함하는 URI 를 웹 에이전트가 자동으로 승인할지 여부를 결정합니다.

기본값: No

DisableDotDotRule 매개 변수가 yes 로 설정되면 에이전트가 이중 점 규칙을 적용하지 *않습니다*. 예를 들어 URI 가 다음과 같은 경우

- /dir1/app.pl/file1.gif

  웹 에이전트는 IgnoreExt 매개 변수를 사용하여 해당 리소스가 자동으로 승인되는지 여부를 결정합니다.

- /dir1/okay.button.gif

  두 점이 슬래시(/)로 구분되어 있지 않으므로 에이전트가 이 URI 를 무시할 수 있습니다. 이 경우에는 이중 점 규칙을 적용할 수 없습니다.

DisableDotDotRule 매개 변수가 no(기본값)로 설정되면 웹 에이전트가 이중 점 규칙을 *적용합니다*. 웹 에이전트는 다음 URI 에 대한 요청을 시도하고 해당 요청을 정책 서버에 전달합니다.

- /dir1/app.pl/file1.gif

  이 URI 는 두 점이 슬래시로 구분되므로 이중 점 규칙의 영향을 받습니다.

  웹 서버는 /dir1/app.pl 을 대상 리소스로 간주하고 /file1.gif 를 추가 경로 정보(일반적으로 CGI 헤더에서 PATH_INFO 로 볼 수 있음)로 간주합니다.

- /dir1/okay.button.gif

  이중 점 규칙이 적용되더라도 두 점이 슬래시(/)로 구분되어 있지 않으므로 에이전트가 이 URI 를 무시하고 결과적으로 이 규칙을 적용할 수 없습니다.

중요! IgnoreExt 매개 변수와 DisableDotDotRule 매개 변수를 함께 사용하는 경우 무단 액세스 문제가 일어나지 않도록 주의하십시오. 예를 들어 /dir1/app.pl 을 보호하려는 경우 DisableDotDotRule 매개 변수를 yes 로 설정하면 에이전트에서 /dir1/app.pl/file1.gif URI 를 무시합니다. 이는 이중 점 규칙이 사용되지 않도록 설정하고 IgnoreExt 매개 변수에 .gif 를 포함했기 때문입니다. 이렇게 되면 권한이 없는 사용자가 /dir1/app.pl, 즉 보호된 응용 프로그램에 액세스할 수 있습니다.

제 7 장: SiteMinder 에이전트에 P3P 압축 정책 사용

SiteMinder 에이전트는 다음 절차에 설명된 매개 변수를 사용하여 P3P 압축 정책을 지원할 수 있습니다.

SiteMinder 웹 에이전트에서 P3P 압축 정책을 지원하는 방법

SiteMinder 는 Domino 에이전트를 *제외한* 모든 웹 에이전트에서 P3P 압축 정책을 지원합니다.

참고: P3P 에 대한 자세한 내용은 World Wide Web Consortium 웹 사이트의 P3P 페이지를 참조하십시오.

P3P 압축 정책을 지원하도록 웹 에이전트를 구성하려면 다음을 수행하십시오.

1. 웹 서버에서 P3P 압축 정책을 구성합니다.

   참고: 자세한 내용은 웹 서버 공급업체에서 제공하는 설명서를 참조하십시오.

2. P3P 압축 정책을 준수하도록 웹 에이전트를 구성합니다.

P3P 압축 정책을 준수하도록 웹 에이전트 구성

다음 매개 변수를 사용하면 웹 에이전트의 사용자 지정 응답이 P3P 응답 헤더를 준수할지 여부를 결정할 수 있습니다.

P3PCompactPolicy

사용자 지정 응답이 P3P(Platform for Privacy Preferences Project) 응답 헤더를 준수할지 여부를 결정합니다. P3P 압축 정책은 P3P 용어의 특정 요소를 나타내는 토큰을 사용합니다. P3PCompactPolicy 매개 변수를 해당 정책 구문으로 설정하면 P3P 압축 정책이 웹 에이전트에 대해 지정된 경우 사용자 지정 응답이 올바른 P3P 응답 헤더로 설정됩니다.

기본값: 기본값 없음

예: NON DSP COR CURa TAI(none, disputes, correct, current/always, tailoring 을 각각 나타냄)

P3P 압축 정책을 준수하려면 적절한 정책 구문을 P3PCompactPolicy 매개 변수에 추가합니다.

제 8 장: 세션 보호

이 섹션은 다음 항목을 포함하고 있습니다.

웹 응용 프로그램 클라이언트에 SiteMinder 동작 적용

일부 웹 응용 프로그램은 웹 브라우저 컨텍스트에서 실행되는 스크립트 엔진을 사용하여 리소스를 요청하고 내용을 표시합니다. 스크립트 엔진에서 발생하는 요청은 표준 웹 브라우저에서 보내는 요청과 마찬가지로 SiteMinder 에서 생성된 동작(예: HTTP 리디렉션 또는 챌린지)을 트리거할 수 있습니다.

웹 응용 프로그램과 제대로 통합되지 않은 상태에서 이 동작을 적용하면 웹 응용 프로그램 클라이언트가 미확정 상태가 될 수 있습니다.

웹 응용 프로그램 클라이언트 응답 ACO 매개
변수(WebAppClientResponse)를 사용하면 다음을 수행할 수 있습니다.

- 웹 브라우저 컨텍스트에서 실행되는 스크립트 엔진을 통해 발생하는 요청을 식별하도록 SiteMinder 를 구성합니다.
- SiteMinder 에서 생성된 동작(예: 챌린지)을 웹 응용 프로그램 클라이언트의 기능과 통합하도록 사용자 지정된 응답을 사용합니다.

  참고: WebAppClientResponse 매개 변수를 사용하여 SiteMinder 의 세션 관리 기능(예: 유휴 시간 또는 세션 시간 만료)을 통합하는 경우 OverLookSessionFor ACO 매개 변수도 구성하십시오.

  OverLookSessionFor 매개 변수를 사용하면 웹 응용 프로그램 클라이언트 요청이 사용자 세션을 무기한 활성화하는 것을 방지할 수 있으며 WebAppClientResponse 매개 변수를 사용하면 세션 시간 만료 후 사용자를 리디렉션하도록 SiteMinder 의 필수 기능을 통합할 수 있습니다.

추가 정보:

세션 시간 만료 후 사용자 리디렉션 (페이지 130)
메서드와 URI 를 기반으로 세션 쿠키 생성 또는 업데이트 방지 (페이지 127)

웹 응용 프로그램 클라이언트 응답

WebAppClientResponse ACO 매개 변수를 사용하여 웹 응용 프로그램 클라이언트의 기능을 구현하고 SiteMinder 보안을 유지 관리합니다.

매개 변수는 다음과 같은 기본 특성으로 구성됩니다.

```
Resource=|Method=|Status=|Body=|ContentType=|Charset=
```

다음 사항을 고려하십시오.

- 이 ACO 매개 변수를 사용하려면 하나 이상의 특성에 유효한 값을 지정해야 합니다.
- 모든 추가 특성은 선택 사항입니다.
- 여러 웹 응용 프로그램의 요청을 식별해야 하는 경우 단일 ACO 매개 변수의 각 특성에 대해 여러 개의 값을 포함할 수 있습니다.
- 웹 응용 프로그램 클라이언트 응답 기능은 기본 인증 체계에서 작동하지 않습니다.

예: WebAppClientResponse ACO 매개 변수

이 예제에서는 매개 변수의 각 특성에 유효한 값을 보여 줍니다. 그 뒤에는 각 특성의 설명이 나옵니다.

```
WebAppClientResponse:Resource=/web20/dir/*|Method=GET,POST|Status=200

|Body=C:\location\custombody_1.txt|Content-Type=application/xml|Charset=us-ascii
```

리소스

웹 응용 프로그램 클라이언트가 요청하는 URI 를 지정합니다. 요청의 URI 가 이 값과 일치하면 SiteMinder 는 해당 요청을 웹 응용 프로그램 클라이언트에서 발생한 것으로 식별합니다. 리소스에 와일드카드(*)를 포함하여 접두사 및 접미사를 일치시킬 수 있습니다.

기본값: 값 없음. 이 값을 생략하면 웹 에이전트에서 보호하는 모든 리소스가 매개 변수에 적용됩니다.

제한: 정규식이 지원되지 않습니다.

예: Resource=/web20/dir/*

예: Resource=/web20/dir/*.xml

메서드

웹 응용 프로그램 클라이언트가 요청할 때 사용하는 HTTP 메서드를 지정합니다. 요청의 HTTP 메서드가 이 값과 일치하면 SiteMinder 는 해당 요청을 웹 응용 프로그램 클라이언트에서 발생한 것으로 식별합니다.

기본값: 값 없음. 이 값을 생략하면 모든 HTTP 메서드가 매개 변수에 적용됩니다.

메서드가 여러 개인 경우 쉼표(,)로 구분합니다.

예: GET, POST

Status

웹 응용 프로그램 클라이언트 요청에 대해 SiteMinder 에서 다시 전송해야 하는 HTTP 상태를 지정합니다.

기본값: 값 없음. 이 값을 생략하면 HTTP 상태 200 이 매개 변수에 적용됩니다.

Body

웹 응용 프로그램 클라이언트 요청에 대한 응답으로 사용되는 사용자 지정 본문이 포함된 파일의 정규화된 이름을 지정합니다. 이 파일은 웹 에이전트 호스트 시스템에 있으며 다음과 같은 특징을 갖습니다.

- 텍스트 기반 파일이거나 바이너리 데이터를 포함할 수 있습니다.
- 응용 프로그램 소유자가 작성한 사용자 지정 본문을 포함할 수 있습니다.
- SiteMinder 원인 및 리디렉션 URL 을 전달하는 데 사용 가능한 사용자 지정 본문을 포함할 수 있습니다.

기본값: 값 없음. 이 값을 생략하면 SiteMinder 에서 웹 응용 프로그램 클라이언트에 본문이 없는 응답을 전달합니다.

ContentType

응답이 포함된 파일에 있는 데이터의 MIME 유형을 지정합니다.

기본값: 값 없음. 이 값을 생략하면 MIME 유형 text/plain 이 매개 변수에 적용됩니다.

SiteMinder 에서 생성한 응답이 사용자 지정 본문에 포함된 경우에는 데이터의 콘텐츠 유형이 다음 중 하나여야 합니다.

- text/*
- application/xml
- application/*+xml

Charset

본문 파일에 있는 데이터의 문자 집합을 지정합니다.

기본값: 값 없음. 이 값을 생략하면 us-ascii 형식의 문자 집합이 매개 변수에 적용됩니다.

쿠키 공급자 및 웹 응용 프로그램 클라이언트 응답

WebAppClientResponse 매개 변수를 설정할 때 다음 사항을 고려하십시오.

- 사용자가 Web 2.0 리소스에 액세스하는 경우 SiteMinder 는 쿠키 공급자의 세션 쿠키를 업데이트하지 않습니다.
- 사용자가 Web 2.0 이외의 리소스(예: .html, .jsp, .asp, .cgi)에 액세스하는 경우 SiteMinder 는 쿠키 공급자의 세션 쿠키를 정상적으로 업데이트합니다.

웹 응용 프로그램에 웹 응용 프로그램 클라이언트 응답을 적용하는 방법

웹 응용 프로그램을 사용하여 웹 응용 프로그램 클라이언트 응답을 적용하면 SiteMinder 보안을 유지 관리하면서 웹 응용 프로그램 클라이언트의 기능을 구현할 수 있습니다. 웹 응용 프로그램 클라이언트 응답을 적용하려면 다음 단계를 완료하십시오.

1. 웹 응용 프로그램 클라이언트 응답 ACO 매개 변수(WebAppClientResponse)를 구성합니다.
2. 사용자 지정 응답을 구성합니다.
3. 사용자 지정 응답을 처리하기 위해 웹 응용 프로그램을 구성합니다.

웹 응용 프로그램 클라이언트 응답 구성

웹 응용 프로그램 클라이언트의 기능을 구현하려면 웹 응용 프로그램 클라이언트 응답을 구성하십시오.

다음 단계를 수행하십시오.

1. 다음 태스크 중 하나를 수행합니다.
   - 관리 UI 에서 ACO(에이전트 구성 개체)를 열고 WebAppClientResponse 의 주석 처리를 제거합니다.
   - 로컬 에이전트 구성 파일을 열고 WebAppClientResponse 의 주석 처리를 제거합니다.
2. 다음과 같은 기본 특성 중 하나 이상의 특성에 대해 값을 입력합니다.
   - 리소스
   - 메서드
   - Status
   - Body
   - Content-Type
   - Charset

참고: 다음과 같은 제한 사항을 고려하십시오.

- 이 ACO 매개 변수를 구성할 때 하나 이상의 특성에 유효한 값을 입력해야 합니다.
- 모든 추가 특성은 선택 사항입니다.
- 여러 웹 응용 프로그램의 요청을 식별해야 하는 경우 단일 ACO 매개 변수의 각 특성에 대해 여러 개의 값을 포함할 수 있습니다.

3. 다음 태스크 중 하나를 수행합니다.
   - 관리 UI 에서 ACO 를 저장합니다.
   - 로컬 에이전트 구성 파일을 저장합니다.

사용자 지정된 응답 구성

응용 프로그램 소유자는 웹 에이전트 호스트 시스템에 있는 파일의 본문에 사용자 지정된 응답을 구성합니다. 웹 응용 프로그램 클라이언트 요청이 SiteMinder 기능을 트리거하면 웹 에이전트에서 웹 응용 프로그램 클라이언트에 대한 응답으로 이 본문을 반환합니다.

다음 사항을 고려하십시오.

- 파일에 응용 프로그램 소유자가 작성한 사용자 지정 본문을 포함할 수 있습니다.
- 텍스트 기반 파일일 수 있습니다. 텍스트 기반 파일인 경우 SiteMinder 는 웹 응용 프로그램 클라이언트에 응답을 보내기 전에 $$Reason$$ 및 $$URL$$에 대해 파일 본문을 구문 분석합니다.

  응답이 SiteMinder 에서 생성된 동작을 포함하는 경우

  - 데이터의 MIME 콘텐츠 유형이 다음 중 하나여야 합니다.
    - text/*
    - application/xml
    - application/*+xml

- 다음과 같은 자리 표시자 값이 본문에 나타나야 합니다.

  ```
  SiteminderReason=$$Reason$$
  SiteminderRedirectURL=$$URL$$
  ```

  SiteMinder 는 위의 값에 대해 본문을 구문 분석하고 트리거된 SiteMinder 기능 및 리디렉션 URL 을 삽입합니다. 다음 매개 변수 또는 정책 응답 유형에 이러한 기능과 URL 을 정의합니다.

  - IdleTimeoutURL
  - MaximumTimeoutURL
  - ForceFQHost
  - LogOffRedirectURL
  - ExpiredCookieURL
  - OnAuthAcceptRedirect
  - OnAuthRejectRedirect
  - OnAccessAcceptRedirect
  - OnAccessRejectRedirect
  - Challenge

  예: 웹 응용 프로그램 클라이언트 요청이 유휴 시간 만료를 트리거하는 경우를 가정합니다. SiteMinder 는 자리 표시자 값을 IdleTimeoutURL 및 IdleTimeoutURL 매개 변수 값에 지정된 URL 로 바꿉니다.

- 파일에 바이너리 데이터를 포함할 수 있습니다. 파일에 바이너리 데이터가 포함된 경우 SiteMinder 는 파일 본문을 구문 분석하지 않은 상태로 웹 응용 프로그램 클라이언트에 전달합니다.

사용자 지정 응답을 처리하기 위해 웹 응용 프로그램 구성

사용자 지정 응답에 SiteMinder 원인 및 리디렉션 URL 이 포함된 경우 웹 응용 프로그램을 별도로 구성하여 사용자 지정 응답을 처리합니다.

웹 에이전트 설치 마법사는 *web_agent_home*/samples 에 샘플 응용 프로그램을 설치합니다. 사용자의 특정 환경과 상황에 맞는 내용을 샘플에서 참조하십시오.

web_agent_home

웹 에이전트 설치 경로를 지정합니다.

세션 유예 기간 수정

일반적으로 웹 페이지는 여러 리소스로 구성되는데 이러한 리소스는 모두 웹 에이전트에서 보호될 수 있습니다. 단일 요청과 연결된 각 리소스에 대해 세션 쿠키가 생성됩니다. 단일 사용자 요청에 대해 세션 쿠키가 여러 개 생성되는 오버헤드를 제거하려면 다음 매개 변수를 설정합니다.

SessionGracePeriod

SiteMinder 세션(SMSESSION) 쿠키가 다시 생성되지 않는 시간(초)을 지정합니다. 다음 조건이 *모두* 충족될 경우 쿠키가 등록되지 않습니다.

- URL SMSESSION 쿠키가 없습니다.
- 수신된 SMSESSION 쿠키의 현재 시간과 마지막 액세스 시간 간의 차이가 SessionGracePeriod 보다 작거나 같습니다.
- 현재 시간과 수신된 쿠키가 유휴 상태였던 시간 간의 차이가 두 번의 유예 기간을 초과합니다. 예를 들어 유예 기간이 25 분이고 유휴 시간 만료가 60 분일 경우 10 분이 경과하면 세션이 유휴 상태가 되기까지 두 번의 유예 기간(50 분)보다 적은 시간이 남기 때문에 SiteMinder 는 10 분 후 세션 쿠키를 다시 생성합니다.

기본값: 30

세션 유예 기간을 수정하려면

1. SessionGracePeriod 매개 변수의 값을 변경합니다.
2. 1 단계에서 SessionGracePeriod 매개 변수의 설정을 늘린 경우에는 관리 UI 를 사용하여 모든 영역의 다음 두 값이 SessionGracePeriod 매개 변수의 값을 초과하지 *않도록* 해야 합니다.
   - 세션 시간 만료 값
   - 유휴 시간 만료 값

   세션 유예 기간이 변경됩니다.

   참고: 세션 시간 만료 값은 관리 UI 를 사용하여 영역을 구성할 때 설정됩니다. 세션 시간 만료를 구성하는 방법에 대한 자세한 내용은 정책 서버 설명서를 참조하십시오.

세션 업데이트 간격 수정

다음 매개 변수를 사용하면 새 쿠키를 설정하기 위해 웹 에이전트에서 쿠키 공급자로 요청을 리디렉션하는 간격을 지정할 수 있습니다.

SessionUpdatePeriod

웹 에이전트가 새 쿠키를 설정하기 위해 쿠키 공급자에 요청을 리디렉션하는 빈도(초)를 지정합니다. 마스터 쿠키를 새로 고치면 SiteMinder 세션의 유휴 시간 만료로 인해 쿠키가 만료될 가능성이 줄어듭니다.

기본값: 60

세션 업데이트 간격을 수정하려면

1. CookieProvider 매개 변수가 정의되어 있는지 확인합니다.
2. SessionUpdatePeriod 매개 변수의 값을 원하는 간격(초)으로 변경합니다.

   세션 업데이트 간격이 변경됩니다.

유효성 검사 기간 및 만료된 쿠키 URL 을 사용하여 세션 쿠키의 오용 방지

SiteMinder 는 시간을 기준으로 하는 세션 쿠키 매개 변수를 사용하므로 관리자 또는 다음 항목에 대한 액세스 권한이 있는 다른 사용자에 의해 SiteMinder 세션 쿠키가 손상될 가능성을 많이 줄일 수 있습니다.

- 웹 서버 로그
- SiteMinder 웹 에이전트 로그
- 도메인 간 싱글 사인온의 경우 도메인 사이에서 손상 위험이 있는 프록시 서버

시간을 기준으로 하는 이러한 세션 쿠키 매개 변수는 "생성 날짜" 개념을 세션 쿠키에 추가합니다. 리디렉션의 결과로 URL 세션 쿠키를 받는 에이전트는 쿠키 생성 날짜 이름/값 쌍을 검색하고 이 값을 CookieValidationPeriod 구성 매개 변수에 설정된 값과 비교합니다. 생성 날짜 값과 CookieValidationPeriod 매개 변수 값이 현재 시간을 초과하면 쿠키가 거부됩니다.

세션 쿠키가 잘못 사용되지 않게 하려면 다음 매개 변수를 설정합니다.

CookieValidationPeriod

수신 에이전트가 세션 쿠키를 수락하는 시간(초)을 지정합니다. 이 시간이 경과하면 세션 쿠키가 수락되지 않습니다. 이 필드를 사용하지 않거나 0 으로 설정하면 "유휴 시간 만료" 및 "Max Session Timeout"(최대 세션 시간 만료) 값이 되었을 때 세션 쿠키가 만료됩니다.

기본값: 비어 있음.

ExpiredCookieURL

(선택 사항) 세션 쿠키가 만료된 후 에이전트가 사용자를 리디렉션할 대상 URL 을 지정합니다. 생성 날짜와 CookieValidationPeriod 가 모두 구성되지 않은 경우 에이전트는 설정을 무시하고 쿠키를 정상적으로 처리합니다(이전 버전과의 호환성).

세션 쿠키 생성 또는 업데이트 방지

Microsoft Outlook Web Access 와 같은 일부 웹 응용 프로그램은 사용자가 현재 해당 응용 프로그램을 사용하지 않는 경우에도 백그라운드에서 HTTP 요청을 수행합니다. 예를 들어 Web Access 응용 프로그램은 사용자가 서버에서 새 전자 메일을 확인하지 않는 경우에도 HTTP 요청을 수행합니다.

이러한 요청은 SMSESSION 쿠키를 업데이트할 수 있으므로 사용자가 유휴 상태인 경우에도 세션이 만료되지 않습니다. 백그라운드 요청이 수행되는 동안 웹 에이전트가 세션 쿠키를 생성하거나 업데이트하지 않도록 설정하면 세션을 정상적으로 만료할 수 있습니다.

다음 매개 변수를 구성합니다.

OverlookSessionForMethods

웹 에이전트가 모든 HTTP 요청의 요청 메서드를 이 매개 변수에 나열된 메서드와 비교할지 여부를 지정합니다. 일치할 경우 웹 에이전트는 SMSESSION 쿠키를 생성 또는 업데이트하지 않습니다. 또한 쿠키 공급자가 구성되었을 경우 해당 요청에 대해 공급자가 업데이트되지 않습니다.

기본값: 기본값 없음

OverlookSessionForUrls

웹 에이전트가 모든 HTTP 요청의 URL 을 이 매개 변수에 나열된 URL 과 비교할지 여부를 지정합니다. 일치할 경우 웹 에이전트는 SMSESSION 쿠키를 생성 또는 업데이트하지 않습니다. 또한 쿠키 공급자가 구성되었을 경우 해당 요청에 대해 공급자가 업데이트되지 않습니다.

기본값: 기본값 없음

예: 상대 URL(예: /MyDocuments/index.html)을 사용하십시오. 절대 URL(http://fqdn.host/MyDocuments/index.html)을 사용하지 마십시오.

참고: 위의 매개 변수를 둘 다 구성하는 경우 메서드가 URL 보다 먼저 처리됩니다.

OverlookSessionAsPattern

활성화된 경우 웹 에이전트는 OverlookSessionForUrls 에 지정된 디렉터리 아래의 모든 URL 에 대해 쿠키를 생성하지 않습니다.

기본값: No

값: Yes, No

예: OverlookSessionForUrls 에 /siteminder 를 지정하고 OverlookSessionAsPattern 을 Yes 로 설정하는 경우 /siteminder/* 요청에 대해 쿠키가 생성되지 않습니다.

메서드와 URI 를 기반으로 세션 쿠키 생성 또는 업데이트 방지

Microsoft Outlook Web Access 와 같은 일부 웹 응용 프로그램은 사용자가 현재 해당 응용 프로그램을 사용하지 않는 경우에도 백그라운드에서 HTTP 요청을 수행합니다. 예를 들어 Web Access 응용 프로그램은 사용자가 서버에서 새 전자 메일을 확인하지 않는 경우에도 HTTP 요청을 수행합니다.

이러한 요청은 SMSESSION 쿠키를 업데이트할 수 있으므로 사용자가 유휴 상태인 경우에도 세션이 만료되지 않습니다. 이와 같은 백그라운드 요청이 수행되는 동안 웹 에이전트가 세션 쿠키를 생성하거나 업데이트하지 않도록 설정하면 세션을 정상적으로 만료할 수 있습니다.

메서드와 URI 를 기반으로 생성 또는 업데이트를 방지하려면

1. 다음 매개 변수를 모두 설정합니다.

   OverlookSessionForMethods

   웹 에이전트가 모든 HTTP 요청의 요청 메서드를 이 매개 변수에 나열된 메서드와 비교할지 여부를 지정합니다. 일치할 경우 웹 에이전트는 SMSESSION 쿠키를 생성 또는 업데이트하지 않습니다. 또한 쿠키 공급자가 구성되었을 경우 해당 요청에 대해 공급자가 업데이트되지 않습니다.

   기본값: 기본값 없음

   OverlookSessionForMethodUri

   웹 에이전트가 모든 HTTP 요청의 메서드 및 URI 를 이 매개 변수에 나열되는 메서드 및 URI 와 비교할지 여부를 지정합니다. 일치할 경우 웹 에이전트는 SMSESSION 쿠키를 생성 또는 업데이트하지 않습니다. 쿠키 공급자가 구성되었을 경우 해당 요청에 대해 공급자가 업데이트되지 않습니다.

   기본값: 기본값 없음

   제한: 상대 URI 을 지정하십시오. 마침표와 URL 사이에 공백을 추가하지 *마십시오*.

   예: POST,/directory/file 은 /directory/resource 에 대한 POST 요청의 SMSESSION 쿠키가 업데이트되지 않게 합니다.

   참고: 메서드가 URI 보다 먼저 처리됩니다.

보안 향상을 위해 세션 저장소에 세션 쿠키 저장

세션 쿠키는 최종 사용자의 클라이언트 컴퓨터에 저장됩니다. SiteMinder 에서 SiteMinder 세션 저장소에 저장되는 세션 쿠키를 생성하면 환경의 보안을 향상시킬 수 있습니다. 세션 쿠키를 SiteMinder 세션 저장소에 저장하면 다음 항목에 액세스할 수 있는 사람이 클라이언트 컴퓨터의 세션 쿠키를 복사한 후 재생 공격을 시도할 수 없게 됩니다.

- 웹 서버 로그
- SiteMinder 웹 에이전트 로그
- 도메인 사이에서 손상 위험이 있는 프록시 서버(여러 도메인에 대해 싱글 사인온 사용)

다음 매개 변수를 설정하여 SiteMinder 에 세션 쿠키를 저장할지 여부를 제어할 수 있습니다.

StoreSessioninServer

세션 쿠키를 클라이언트 컴퓨터에 저장할지 아니면 SiteMinder 세션 저장소에 저장할지를 지정합니다. StoreSessioninServer 매개 변수의 값이 yes 이면 세션 쿠키가 생성된 후 세션 저장소에 저장됩니다. 쿠키 공급자 및 웹 에이전트는 세션 저장소의 쿠키에 액세스합니다.

쿠키 공급자 및 웹 에이전트는 URL 의 세션 쿠키를 세션 저장소에 저장된 세션 쿠키에 해당하는 GUID 로 바꿉니다.

StoreSessioninServer 매개 변수의 값이 no 이면 세션 쿠키가 URL 에서 직접 전달됩니다.

기본값: No

다음 단계를 수행하십시오.

1. 운영 환경이 다음 조건을 충족하는지 확인합니다.
   - SiteMinder 6.0 SP5 QMR1 이상을 사용하기 위해 웹 에이전트 및 쿠키 공급자를 업그레이드합니다.
   - 웹 에이전트 및 쿠키 공급자에 DefaultAgentName 매개 변수의 값을 사용합니다.
   - 유효한 세션 저장소를 사용하여 정책 서버가 구성됩니다.
2. 웹 에이전트 및 쿠키 공급자에서 StoreSessioninServer 매개 변수의 값을 yes 로 설정합니다.

세션 쿠키 도메인 유효성 검사

다음 매개 변수를 사용하여 SiteMinder 에서 세션 쿠키 도메인의 유효성을 검사하도록 설정하면 권한이 없는 사용자가 SiteMinder 세션 쿠키를 가로채 다시 사용하는 위험을 줄일 수 있습니다.

TrackSessionDomain

세션 쿠키의 의도된 도메인을 암호화하여 세션 쿠키 자체 내에 저장하도록 웹 에이전트에 지시합니다. 이후의 요청에서 웹 에이전트는 세션 쿠키 내에 저장된 의도된 도메인과 요청된 리소스의 도메인을 비교합니다. 두 도메인이 일치하지 *않으면* 웹 에이전트가 요청을 거부합니다.

예를 들어 이 매개 변수의 값을 yes 로 설정하면 operations.example.com 에 대해 의도된 세션 쿠키가 finance.example.com 에서 제공되는 경우 해당 쿠키가 거부됩니다.

SSO 를 사용하는 SiteMinder 환경에서는 암호화된 세션 쿠키를 생성하는 웹 에이전트에 이 매개 변수를 설정하십시오. 예를 들어 SSO 환경에 a.example.com 및 b.example.com 이라는 도메인이 있다고 가정해 보십시오. a.example.com 을 보호하는 웹 에이전트가 세션 쿠키를 암호화하는 경우 관련 웹 에이전트의 TrackSessionDomain 매개 변수 값을 설정하십시오. b.example.com 을 보호하는 웹 에이전트가 쿠키를 받으면 쿠키에 저장된 의도된 도메인과 요청된 리소스의 도메인을 비교합니다.

기본값: No

SiteMinder 에서 세션 쿠키 도메인의 유효성을 검사하도록 설정하려면 TrackSessionDomain 매개 변수의 값을 yes 로 설정하십시오.

세션 시간 만료 후 사용자 리디렉션

세션 시간 만료는 관리 UI 를 사용하여 영역을 구성할 때 설정됩니다. 사용자의 SiteMinder 세션 시간이 만료되면 웹 에이전트에서 다음 중 하나를 수행합니다.

- 사용자에게 자격 증명 다시 요청
- 사용자를 다른 URL 로 리디렉션

리디렉션 URL 이 지정된 경우 사용자에게 대상 페이지가 표시됩니다. 해당 페이지가 보호되어 있지 않으면 사용자가 페이지에 바로 액세스할 수 있으며, 해당 페이지가 보호되어 있으면 페이지에 액세스하기 전에 사용자에게 자격 증명이 요청됩니다. 리디렉션 URL 이 지정되어 있지 않은 경우에는 세션 시간이 만료된 후 웹 에이전트에서 사용자에게 자격 증명을 다시 요청합니다.

세션 시간이 만료된 사용자를 사용자 지정 웹 페이지가 포함된 URL 로 리디렉션할 수 있습니다. 사용자 지정 웹 페이지에는 세션이 종료된 이유와 세션을 다시 연결할 수 있는 방법이 설명되어 있습니다. 예를 들어 "You have been logged out automatically as a security precaution. Please login again to continue." (보안상의 이유로 자동 로그아웃되었습니다. 계속하려면 다시 로그인하십시오.)라는 메시지가 표시되는 사용자 지정 웹 페이지를 생성할 수 있습니다.

세션 시간 만료 후 사용자가 다른 페이지로 리디렉션되지 않는 경우 SiteMinder 에서는 사용자에게 다시 인증을 요청합니다. 이런 경우 사용자는 재인증이 요청되는 이유를 이해하지 못하므로 혼란스러울 수 있습니다.

세션 시간 만료 후 사용자를 다른 URL 로 리디렉션하려면

1. 다음 매개 변수를 에이전트 구성 개체 또는 로컬 구성 파일에 추가합니다.

   IdleTimeoutURL

   세션의 유휴 시간 만료가 발생할 경우 웹 에이전트가 사용자를 리디렉션할 URL 을 지정합니다.

   예: http://example.mycompany.com/sessionidletimeoutpage.html

   참고: IdleTimeoutURL 은 비영구 세션에만 사용되며 영구 세션에 대해 구성되었을 경우 아무 영향이 없습니다.

   MaxTimeoutURL

세션의 최대 시간 만료가 발생할 경우 웹 에이전트가 사용자를 리디렉션할 URL 을 지정합니다.

예: http://example.mycompany.com/maxtimeoutpage.html

기본값: 기본값 없음

2. 위의 각 매개 변수에 대해 URL 하나를 입력합니다. 두 매개 변수에 같은 URL 을 사용하거나 각각 다른 URL 을 사용할 수도 있습니다.

   정책 서버에 설정된 세션의 유휴 시간 만료 및 최대 시간 만료 값이 동시에 실행되고 IdleTimeoutURL 및 MaxTimeoutURL 매개 변수가 설정된 경우 시간 만료가 발생하면 MaxTimeoutURL 매개 변수에 지정된 URL 로 사용자가 리디렉션됩니다.

여러 영역에 시간 만료 적용

사용자 세션 시간 만료는 사용자가 처음 로그인하는 영역에 의해 제어됩니다. 사용자가 싱글 사인온을 통해 새 영역에 들어가더라도 새 영역의 시간 만료 값은 첫 번째 영역에서 초기 로그인으로 연결한 세션에 의해 제어됩니다. 각 영역마다 시간 만료 값을 다르게 설정하고 각 영역에서 해당 값을 사용하게 하려면 원래 영역의 시간 만료를 재정의하면 됩니다.

이미 시간 만료된 사용자가 다른 영역에 로그인하려면 재인증이 필요합니다. 예를 들어 Realm1 의 유휴 시간 만료가 15 분이고 Realm2 의 유휴 시간 만료가 30 분인 경우 Realm1 에서 20 분 동안 유휴 상태인 사용자가 Realm2 에 로그인할 때 재인증이 요청됩니다.

원래 영역의 시간 만료를 재정의하려면 아래에 설명된 대로 웹 에이전트와 영역을 구성하십시오.

1. EnforceRealmTimeouts 매개 변수의 값을 yes 로 설정합니다.
2. 관리 UI 를 사용하여 다음 태스크를 수행합니다.
   a. 원래 시간 만료를 대체할 각 영역, 즉 SSO 기능을 통해 사용자가 액세스할 수 있는 영역에 대해 다음을 수행합니다.
      - 최대 시간 만료 값을 재정의하려면 WebAgent-OnAuthAccept-Session-Max-Timeout 응답 특성을 사용하여 응답을 생성합니다.
      - 유휴 시간 만료 값을 재정의하려면 WebAgent-OnAuthAccept-Session-Idle-Timeout 응답 특성을 사용하여 응답을 생성합니다.
   b. 위의 각 응답을 OnAuthAccept 규칙에 바인딩합니다.

   참고: 응답을 생성하는 데 대한 자세한 내용은 *정책 서버 구성 안내서*를 참조하십시오.

유효한 세션이 여러 개 있는 경우 영역 시간 만료 후 재인증되지 않도록 방지

이전 버전의 SiteMinder 는 영역 시간이 만료되면 자동으로 자격 증명에 대한 사용자 재인증을 수행했습니다. 정책 서버에 여러 세션이 있는 경우에도 이러한 인증이 발생했습니다.

이번 버전은 정책 서버가 사용자를 인증하기 전에 목록에 있는 모든 세션을 검사하도록 하는 옵션을 제공합니다.

이 옵션을 제어하는 매개 변수는 다음과 같습니다.

compatRealmtimeouts

영역 시간이 만료된 경우 정책 서버가 사용자에게 자격 증명을 요청하는지 여부를 지정합니다. 이러한 인증은 정책 서버의 첫 번째 세션이 만료된 경우 발생합니다. 정책 서버는 목록의 다른 관련 세션을 검사하지 않습니다. 이 매개 변수의 값이 yes 인 경우 정책 서버는 목록의 첫 번째 세션만 검사합니다. 그런 다음 사용자에게 인증을 요청합니다. 이 매개 변수의 값이 no 인 경우 정책 서버는 사용자에게 인증을 요청하기 전에 목록에 있는 모든 세션을 검사합니다.

기본값: No(영역 시간이 만료되는 경우 모든 세션이 검사됨)

영역 시간이 만료되는 경우 목록의 첫 번째 세션만 검사하려면 compatRealmtimeouts 매개 변수의 값을 yes 로 변경하십시오.

클라이언트 인증서를 SiteMinder 세션에 연결하는 방법(Windows)

IIS 7.x(Windows 만 해당) 및 Apache 기반 웹 서버의 경우 클라이언트 인증서를 SiteMinder 세션에 연결할 수 있습니다. 이 기능은 다음 아이덴티티가 일치하는지 확인합니다.

- SiteMinder 세션과 연결되는 사용자의 아이덴티티
- 트랜잭션에 사용되는 클라이언트 인증서의 아이덴티티

이러한 항목이 일치하지 않으면 트랜잭션이 차단됩니다.

이 기능을 사용하려면 다음 태스크를 수행하십시오.

- 클라이언트 인증서를 자동으로 얻도록 모든 웹 서버를 구성합니다. 특히 로그온 서버가 정책 적용 지점과 분리되어 있는 경우 필요합니다.

X.509 인증서 인증 체계를 사용합니다(다른 인증 체계는 지원되지 *않음*).

다음 그림에서는 클라이언트 인증서를 세션에 연결하는 방법을 보여 줍니다.

다음 단계를 수행하십시오.

1. 플러그인을 WebAgent.conf 파일에 추가합니다 (페이지 135).
2. 에이전트 구성 매개 변수를 설정합니다 (페이지 136).

플러그인 추가

클라이언트 인증서를 세션에 연결하는 첫 번째 단계는 플러그인을 추가하는 것입니다.

다음 단계를 수행하십시오.

1. 에이전트를 호스트하는 시스템에 로그인합니다.
2. 텍스트 편집기에서 다음 파일을 엽니다.

   `WebAgent.conf`
3. 다음 행을 찾습니다.

   `LoadPlugin="web_agent_home\bin\HttpPlugin.dll"`
4. 3 단계의 줄 바로 아래에 줄을 하나 추가합니다.
5. 새 줄에는 다음 내용을 추가합니다.

   `LoadPlugin="web_agent_home\bin\CertSessionLinkerPlugin.dll"`

   참고: CertSessionLinkerPlugin 은 HttpPlugin 을 따라야 합니다.
6. 파일을 저장합니다.
7. 웹 서버를 다시 시작합니다.

   플러그인이 추가되었습니다. 계속해서 구성 매개 변수를 추가합니다.

에이전트 구성 매개 변수 설정

플러그인을 추가한 후 에이전트 구성 매개 변수를 설정합니다.

다음 단계를 수행하십시오.

1. 관리 UI 에서 원하는 에이전트 구성 개체를 엽니다.
2. 다음 매개 변수의 값을 변경합니다.

   CslCertUniqueAttribute

   인증서를 고유하게 식별하는 데 사용되는 인증서의 특성을 나열합니다. 다음과 같은 인증서 특성을 사용할 수 있습니다.

   - version
   - serialnumber
   - signaturealgorithm
   - issuerdn
   - subjectdn
   - validitystart
   - validityend

   참고: 이 매개 변수에서 값의 순서는 중요하지 않습니다.

   기본값: Disabled(serialnumber 및 issuerdn 특성만 일치됨)

   CslMaxCacheEntries

   에이전트 캐시에 포함된 최대 항목 수를 지정합니다.

   참고: UNIX 에서 작동하는 Apache 기반 서버의 경우 singleprocessmode 매개 변수의 값을 no 로 설정하는 것이 좋습니다. 이 설정은 여러 요청 사이에 정보를 공유하는 다중 프로세스 캐시를 생성합니다. 이 설정은 Apache 기반 서버가 프리포크(pre-fork) 모드에서 실행되는 경우 성능을 향상시킵니다.

   기본값: 1000
3. 변경 내용을 저장하고 에이전트 구성 개체를 닫습니다.

   인증서가 세션에 연결되었습니다.

클라이언트 인증서를 세션에 연결하는 방법(UNIX)

IIS 7.x(Windows 만 해당) 및 Apache 기반 웹 서버의 경우 클라이언트 인증서를 SiteMinder 세션에 연결할 수 있습니다. 이 기능은 다음 아이덴티티가 일치하는지 확인합니다.

- SiteMinder 세션과 연결되는 사용자의 아이덴티티
- 트랜잭션에 사용되는 클라이언트 인증서의 아이덴티티

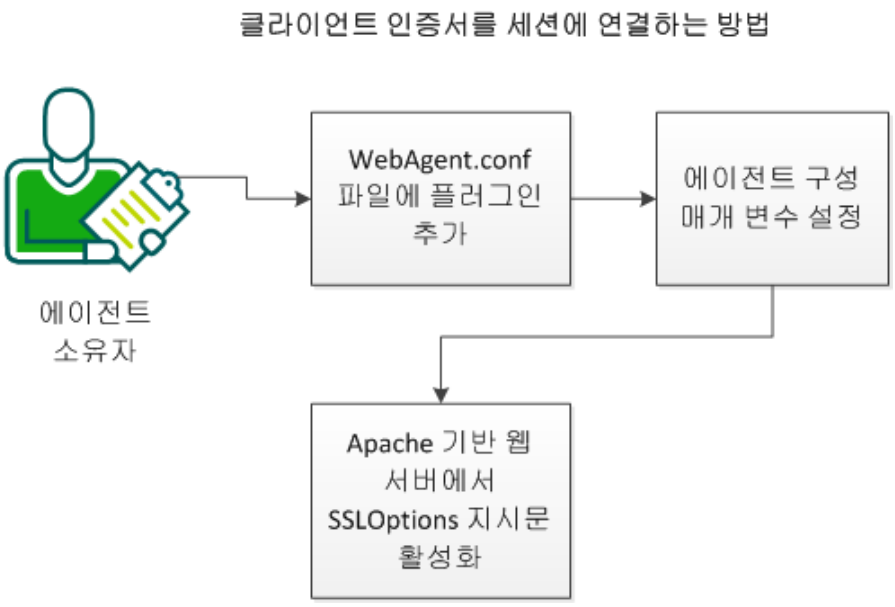
이러한 항목이 일치하지 않으면 트랜잭션이 차단됩니다.

이 기능을 사용하려면 다음 태스크를 수행하십시오.

- 클라이언트 인증서를 자동으로 얻도록 모든 웹 서버를 구성합니다. 특히 로그온 서버가 정책 적용 지점과 분리되어 있는 경우 필요합니다.

X.509 인증서 인증 체계를 사용합니다(다른 인증 체계는 지원되지 *않음*).

다음 그림에서는 클라이언트 인증서를 세션에 연결하는 방법을 보여 줍니다.



다음 단계를 수행하십시오.

1. 플러그인을 WebAgent.conf 파일에 추가합니다 (페이지 138).
2. 에이전트 구성 매개 변수를 설정합니다 (페이지 139).
3. Apache 기반 웹 서버에서 SSLOptions 지시문을 사용하도록 설정합니다 (페이지 140).

플러그인 추가

클라이언트 인증서를 세션에 연결하는 첫 번째 단계는 플러그인을 추가하는 것입니다.

다음 단계를 수행하십시오.

1. 에이전트를 호스트하는 시스템에 로그인합니다.
2. 텍스트 편집기에서 다음 파일을 엽니다.

   `WebAgent.conf`
3. 다음 행을 찾습니다.

   `LoadPlugin="web_agent_home/bin/libHttpPlugin.so"`
4. 3 단계의 줄 바로 아래에 줄을 하나 추가합니다.
5. 새 줄에는 다음 내용을 추가합니다.

   `LoadPlugin="web_agent_home/bin/libCertSessionLinkerPlugin.so"`

   참고: libCertSessionLinkerPlugin 은 libHttpPlugin 을 따라야 합니다.
6. 파일을 저장합니다.
7. 웹 서버를 다시 시작합니다.

   플러그인이 추가되었습니다. 계속해서 구성 매개 변수를 추가합니다.

에이전트 구성 매개 변수 설정

플러그인을 추가한 후 에이전트 구성 매개 변수를 설정합니다.

다음 단계를 수행하십시오.

1. 관리 UI 에서 원하는 에이전트 구성 개체를 엽니다.

   다음 매개 변수의 값을 변경합니다.

   CslCertUniqueAttribute

   인증서를 고유하게 식별하는 데 사용되는 인증서의 특성을 나열합니다. 다음과 같은 인증서 특성을 사용할 수 있습니다.

   - version
   - serialnumber
   - signaturealgorithm
   - issuerdn
   - subjectdn
   - validitystart
   - validityend

   참고: 이 매개 변수에서 값의 순서는 중요하지 않습니다.

   기본값: Disabled(serialnumber 및 issuerdn 특성만 일치됨)

CslMaxCacheEntries

에이전트 캐시에 포함된 최대 항목 수를 지정합니다.

참고: UNIX 에서 작동하는 Apache 기반 서버의 경우 singleprocessmode 매개 변수의 값을 no 로 설정하는 것이 좋습니다. 이 설정은 여러 요청 사이에 정보를 공유하는 다중 프로세스 캐시를 생성합니다. 이 설정은 Apache 기반 서버가 프리포크(pre-fork) 모드에서 실행되는 경우 성능을 향상시킵니다.

기본값: 1000

2. Apache 기반 서버의 경우 다음 매개 변수가 있으면 값이 no 인지 확인합니다.

SingleProcessMode

Apache 기반 서버(작업자 모드)에서 단일 프로세스만 CSL 캐시를 사용할 수 있는지 여부를 지정합니다. 단일 프로세스만 캐시를 사용하도록 하려면 이 값을 yes 로 설정합니다. UNIX 운영 환경에서 Apache 기반 서버는 별도의 프로세스가 각 요청을 처리하는 프리포크(prefork) 모드로 작동합니다. 여러 요청 간에 CSL 캐시를 공유하도록 UNIX 운영 환경에서 이 값을 no 로 설정하는 것이 좋습니다.

기본값: No(프리포크 모드, 다중 프로세스 캐시)

옵션: Yes(작업자 모드, 단일 프로세스 캐시)

3. 변경 내용을 저장하고 에이전트 구성 개체를 닫습니다.

계속해서 SSLOptions 지시문을 사용하도록 설정합니다.

Apache 기반 웹 서버에서 SSLOptions 지시문을 사용하도록 설정합니다.

다음 단계는 Apache 기반 서버의 SSLOptions 지시문을 사용하도록 설정하는 것입니다. 이 지시문을 사용하도록 설정하면 인증서 특성을 환경 변수로 사용할 수 있습니다.

Apache 기반 서버의 http.conf 파일에 다음 항목을 추가하십시오.

```
SSLOptions +StdEnvVars
```

제 9 장: 웹 응용 프로그램 보호

이 섹션은 다음 항목을 포함하고 있습니다.

응용 프로그램 보호 방법

SiteMinder 는 웹 응용 프로그램 보호를 위해 다음과 같은 방법을 제공합니다.

- REMOTE_USER 변수 - 인증된 사용자 이름을 응용 프로그램에 전달 (페이지 141)
- 응답 특성 (페이지 145)
- HTTP 헤더 (페이지 149)
- 사용자 지정 오류 페이지 (페이지 164)

REMOTE_USER 변수

REMOTE_USER 변수는 웹 서버에 의해 인증된 사용자의 이름을 보유합니다. 웹 서버에 에이전트가 설치되면 SiteMinder 는 웹 서버의 네이티브 인증을 바꿉니다. REMOTE_USER 변수는 비어 있습니다.

응용 프로그램에서 REMOTE_USER 변수를 사용하는 경우 REMOTE_USER 변수가 설정됩니다.

웹 서버에서 REMOTE_USER 변수를 사용하지 않는 경우에는 HTTP_SM_USER 헤더를 사용하여 응용 프로그램에 사용자 이름을 전달할 수 있습니다.

추가 정보

REMOTE_USER 변수를 설정하기 위해 웹 에이전트 구성 (페이지 142)

REMOTE_USER 변수를 설정하기 위해 웹 에이전트 구성

다음과 같이 웹 에이전트를 구성하여 REMOTE_USER 변수를 설정합니다.

- REMOTE_USER 값을 SiteMinder 로그인 사용자 이름으로 설정하려면 웹 에이전트의 SetRemoteUser 매개 변수를 yes 로 설정합니다.

  이 매개 변수의 기본값은 no 이며 REMOTE_USER 변수는 비어 있습니다.

  참고: SiteMinder 웹 에이전트 5.x QMR 2 이전 버전의 경우 SetRemoteUser 매개 변수는 IIS 웹 서버에만 영향을 주었고 Apache 및 Oracle iPlanet 에이전트는 항상 REMOTE_USER 를 SiteMinder 에 로그인한 사용자 이름으로 설정했습니다. 5.x QMR 2 이전 버전의 에이전트를 설치하거나 업그레이드하면 REMOTE_USER 를 더 이상 기본적으로 사용할 수 없음을 유의하십시오.

- 로그인한 사용자의 자격 증명 대신 특정 사용자 계정을 사용하여 REMOTE_USER 변수를 설정하려면 다음을 수행합니다.

  - SetRemoteUser 매개 변수를 yes 로 설정하여 활성화합니다.
  - RemoteUserVar 매개 변수를 설정합니다. 이 매개 변수는 HTTP-WebAgent-Header-Variable 응답 특성의 값을 사용하여 REMOTE_USER 변수를 채우도록 에이전트에 지시합니다. 이를 사용하면 레거시 응용 프로그램과 통합할 수 있습니다.

    RemoteUserVar 매개 변수를 구성하려면 응답 변수의 이름만 입력합니다. 예를 들어 HTTP-WebAgent-Header-Variable 을 "user=ajohnson"과 같이 반환하려면 RemoteUserVar 매개 변수의 값을 user 로 설정합니다.
  - 헤더 변수를 OnAuthAccept 규칙에 바인딩합니다. 이때 기존 HTTP 헤더 변수 응답을 사용하지 말고 새로 생성해야 합니다.

    참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

- 기본값, 즉 REMOTE_USER 를 빈 상태로 되돌리려면 SetRemoteUser 매개 변수를 no 로 설정합니다.

참고: SetRemoteUser 또는 RemoteUserVar 를 구성하기 전에 보안 문제를 고려하십시오.

IIS 웹 서버와 REMOTE_USER 변수

SiteMinder 에서 REMOTE_USER 변수를 사용하려면 IIS 웹 서버에 기본 인증이 필요합니다.

기본 인증이 설정된 경우 사용자가 SiteMinder 보호 리소스를 요청하면 에이전트는 사용자 이름만 포함하여 IIS 웹 서버의 HTTP_Authorization 헤더를 설정하려고 합니다. 이때 암호는 포함되지 않습니다.

HTTP_Authorization 헤더가 사용될 경우 IIS 웹 서버의 기본 인증 메커니즘은 다른 인증 챌린지보다 우선 순위가 높습니다. 따라서 IIS 웹 서버는 사용자가 자신의 챌린지에 응답하고 있다고 인식합니다.

ISAPI 필터(IIS 의 클래식 파이프라인 모드 사용)로 작동하는 에이전트는 다음과 같은 태스크를 수행하십시오.

- 요청의 사용자 컨텍스트 설정
- REMOTE_USER 헤더의 값 설정

SetRemoteUser 매개 변수의 값이 yes 이고 다음 설정이 사용되면 에이전트에서 REMOTE_USER 헤더를 채웁니다.

- DefaultUsername 및 DefaultPassword - 이러한 두 매개 변수는 에이전트에서 대부분의 활동에 사용하는 권한 있는 프록시 사용자 계정을 제어합니다.
- ForceIISProxyUser - 정상적인 동작을 무시하고 IIS 웹 서버가 프록시 사용자로 실행되도록 에이전트에 지시합니다.
- UseAnonAccess - 에이전트에서 요청에 사용자 컨텍스트를 제공하지 않도록 지시하고 기존 사용자 컨텍스트를 변경되지 않은 상태로 둡니다.
- Run in Authenticated User's Security Context(인증된 사용자의 보안 컨텍스트에서 실행) - 에이전트는 영구 세션에 저장된 자격 증명을 사용하도록 IIS 웹 서버에 지시합니다.

SetRemoteUser 매개 변수와 UseAnonAccess 매개 변수를 함께 사용할 때는 주의해야 합니다.

다음 표에서는 이러한 매개 변수가 함께 사용되는 방식을 보여 줍니다.

매개 변수 설정	결과
SetRemoteUser=yes UseAnonAccess=yes	에이전트가 사용자 보안 컨텍스트를 전달하지 *않으므로* REMOTE_USER 변수를 설정할 수 없습니다. 사용자 보안 컨텍스트가 없으면 IIS 웹 서버는 에이전트에서 수정한 HTTP_Authorization 헤더의 자격 증명을 사용합니다. 그러나 이 헤더는 사용자 이름만 포함하므로 완전하지 않습니다.
SetRemoteUser=yes UseAnonAccess=no	에이전트는 DefaultUserName, DefaultPassword 또는 ForceIISProxyUser 와 같은 다른 매개 변수의 설정에 따라 몇 가지 유형의 사용자 컨텍스트를 전달할 수 있습니다. 에이전트가 IIS 웹 서버에 보안 컨텍스트를 전달하면 IIS 웹 서버는 해당 에이전트의 자격 증명을 사용합니다. IIS 웹 서버는 불완전한 HTTP_Authorization 헤더를 무시합니다.

웹 에이전트에서 응답 특성이 사용되는 방식

SiteMinder 응답 특성은 사용자 데이터를 수집하고 해당 정보를 적용하여 각 사용자에 대해 개인화된 콘텐츠를 표시하는 방법을 응용 프로그램에 알려 줍니다.

SiteMinder 에서는 구성 가능한 응답 특성을 제공하여 응용 프로그램에 데이터를 전달하고 사용자 환경을 사용자 지정할 수 있도록 합니다.

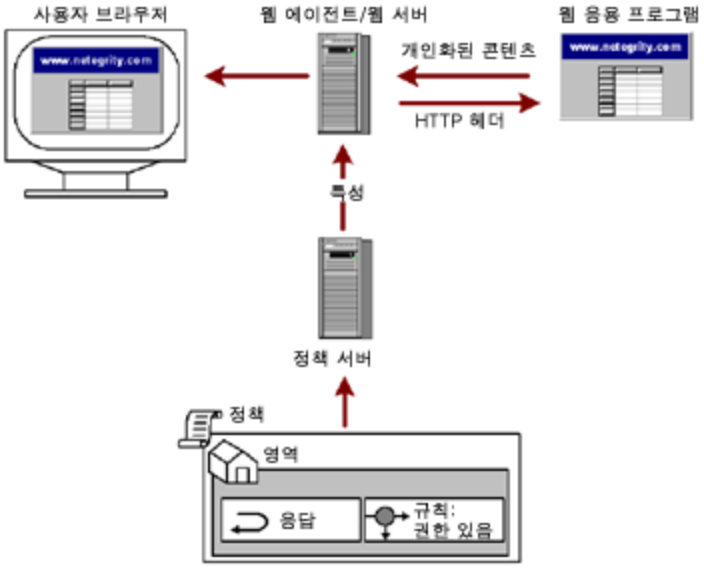
관리 UI 를 사용하여 응답을 구성한 후 정책의 특정 규칙과 응답을 연결합니다. 요청을 통해 구성된 응답을 포함하는 규칙이 트리거되면 정책 서버에서 응답 데이터를 에이전트에 보냅니다. 그러면 해당 정보가 해석되고 웹 응용 프로그램에서 사용 가능한 상태가 됩니다.

응답을 구성하는 경우 응답을 에이전트 작업과 연결합니다. HTTP 헤더 및 쿠키 응답 특성을 GET 및 POST 작업과 연결할 수 있습니다. 또한 이러한 특성을 인증 또는 권한 부여 이벤트에 연결할 수도 있습니다. 이러한 이벤트에 대해 사용자가 수락 또는 거부되는 경우 정책 서버에서 응답을 보낼 수 있습니다.

참고: 응답 특성을 구성하는 경우 에이전트 응답에 대한 웹 서버의 최대 버퍼 크기가 32 KB 임을 참고하십시오. 총 버퍼 크기를 제외하면 응답과 관련된 길이 제한은 없습니다.

헤더 및 쿠키 특성이 아닌 다른 응답 특성은 인증 또는 권한 부여 이벤트가 발생할 경우에만 사용할 수 있습니다. 이러한 이벤트에 대해 사용자가 수락되었는지 여부는 관계가 없습니다. 예를 들어 규칙에 대해 **권한 부여 이벤트** 작업을 선택한 후 **WebAgent-OnReject-Redirect** 응답 특성을 구성할 수 있습니다. 이런 경우 권한 부여가 진행되는 동안 SiteMinder 에 의해 사용자가 거부되면 에이전트는 해당 사용자를 다른 페이지로 리디렉션하여 사용자가 거부된 이유를 나타내는 메시지를 보여 줍니다.

다음 그림에서는 응답 특성이 정책 서버에서 웹 서버로 전송되는 방식을 보여 줍니다.



응답 유지 관리 태스크를 간소화하려면 각 이벤트 유형에 대해 개별적으로 응답을 정의하면 됩니다. 예를 들어 OnAccept 이벤트에 대해 응답을 정의하고 OnReject 이벤트에 대해 다른 응답을 정의합니다. 개별 응답을 생성하면 응답 값을 수정해야 할 경우 특성을 쉽게 찾을 수 있습니다.

양식 챌린지에 SM_AGENT_ATTR_USRMSG 응답 사용

SM_AGENTAPI_ATTR_USERMSG 응답을 사용하면 사용자 지정 SiteMinder 인증 체계 개발자가 사용자 지정 텍스트를 사용자 챌린지의 일부로 또는 다른 용도로 클라이언트 응용 프로그램에 반환할 수 있습니다.

v5 QMR3 이상의 웹 에이전트에는 양식 챌린지를 수행할 때 SM_AGENTAPI_ATTR_USERMSG 응답의 텍스트를 SMUSRMSG 쿠키로 변환할 수 있는 기능이 추가되었습니다.

챌린지가 완료된 후 SMUSRMSG 쿠키가 제거되도록 하기 위해 FCC 는 다음과 같이 POST 요청이 성공하면 쿠키를 소비하여 브라우저에서 삭제합니다.

- SiteMinder 네이티브 모드의 경우 에이전트는 로그인에 성공한 후 쿠키를 삭제하고 대상 URL 로 다시 리디렉션합니다.
- SiteMinder 4.x 호환성 모드의 경우 에이전트는 FORMCRED 쿠키를 생성한 후 쿠키를 삭제하고 대상 URL 로 다시 리디렉션합니다.

참고: SMUSRMSG 쿠키는 일정 기간 동안 사용자의 브라우저에 저장되며 비보안 HTTP 연결을 통해 전송될 수 있습니다. 따라서 중요한 데이터는 포함되지 않도록 해야 합니다.

웹 에이전트는 양식 챌린지가 수행되는 동안 SMUSRMSG 쿠키에 추가되는 텍스트를 URL 인코딩하여 공백 및 기타 위험한 문자를 제거하므로 HTTP 전송 시 안전합니다. FCC 는 사용자 지정 FCC 기능에 사용하기 위해 이 텍스트를 환경에 제공하기 전에 디코딩합니다.

참고: URL 인코딩은 텍스트가 SMUSRMSG 쿠키에 추가되는 경우에만 구현됩니다.

새 기능을 구현하려면 사용자 지정 인증 체계 개발자가 양식 기반 인증 체계를 사용자 지정해야 합니다. Sm_AgentApi_Login()을 호출하여 SM_AGENTAPI_CHALLENGE 가 반환되는 경우 에이전트는 Sm_AgentApi_IsProtected()에 대한 응답으로 제공된 인증 체계 URL 로 리디렉션하여 요청 사용자에 대한 인증을 시도합니다.

웹 에이전트가 HTML 양식 인증 체계 템플릿을 사용하는 인증 체계를 처리하는 경우 에이전트는 SM_AGENTAPI_ATTR_STATUS_MESSAGE 응답 특성을 검색합니다. 이 특성이 있으면 에이전트가 적절한 SMUSRMSG 쿠키를 생성하고 인증 체계 URL 로 리디렉션합니다. 그러면 양식 생성 중에 FCC 가 이 쿠키를 사용할 수 있습니다(적절한 지시문이 원하는 .FCC 원본 파일에 추가된 경우).

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

응답 특성 캐시

동적 데이터를 포함하는 만료 특성 또는 응답 특성을 캐시하도록 SiteMinder 에이전트에 지시하여 에이전트가 정책 서버에 연결한 후 정보를 업데이트하도록 할 수 있습니다. 정적 응답 특성을 구성하는 경우 정책 서버에서 값을 캐시만 할 수 있습니다. 정적 값은 변경되지 않으므로 값을 다시 계산할 필요가 없습니다. 사용자, DN 또는 활성 특성을 구성하는 경우에는 특정 간격으로 값을 캐시하거나 다시 계산하여 데이터를 최신 상태로 유지할 수 있습니다.

SiteMinder 기본 HTTP 헤더

SiteMinder 기본 HTTP 헤더는 사용자 데이터를 수집하고 해당 정보를 적용하여 각 사용자에 대해 개인화된 콘텐츠를 표시하는 방법을 응용 프로그램에 알려 줍니다.

웹 응용 프로그램 환경의 일부로 SiteMinder 에이전트는 기본 HTTP 헤더를 웹 서버에 제출하고 웹 서버는 이 헤더를 웹 응용 프로그램에 사용할 수 있도록 만듭니다. 이러한 헤더를 사용하면 기능을 포함하고 웹 응용 프로그램에서 콘텐츠를 개인화하도록 설정할 수 있습니다. 헤더에는 사용자 이름 및 사용자가 수행할 수 있는 작업 유형과 같은 정보가 저장될 수 있습니다.

에이전트는 웹 응용 프로그램에서 호출되는지 여부에 관계없이 이러한 헤더를 보내지만 일부 헤더가 사용되지 않도록 설정하여 헤더 공간을 절약할 수 있습니다.

다음과 같은 SiteMinder 기본 HTTP 헤더를 웹 에이전트에 사용할 수 있습니다.

HTTP_SM_AUTHDIRNAME

정책 서버가 사용자를 인증하는 데 기준이 되는 디렉터리의 이름을 나타냅니다. 관리자는 관리 UI 에서 이 디렉터리를 지정합니다.

HTTP_SM_AUTHDIRNAMESPACE

정책 서버가 사용자를 인증하는 데 기준이 되는 디렉터리 네임스페이스를 나타냅니다. 관리자는 관리 UI 에서 이 네임스페이스를 지정합니다.

HTTP_SM_AUTHDIROID

정책 서버 데이터베이스의 OID(디렉터리 개체 식별자)를 나타냅니다.

HTTP_SM_AUTHDIRSERVER

정책 서버가 사용자를 인증하는 데 기준이 되는 디렉터리 서버를 나타냅니다. 관리자는 관리 UI 에서 이 디렉터리 서버를 지정합니다.

HTTP_SM_AUTHREASON

실패한 인증 시도 후 또는 두 번째 인증 요청 후 웹 에이전트가 사용자에게 반환하는 코드를 나타냅니다.

HTTP_SM_AUTHTYPE

정책 서버가 사용자의 아이덴티티를 확인하는 데 사용하는 인증 체계의 유형을 나타냅니다.

HTTP_SM_DOMINOCN

Domino LDAP 디렉터리를 사용하여 사용자를 인증하는 경우 사용자의 Domino 정규 이름을 나타냅니다.

예: HTTP_SM_DOMINOCN="CN=jsmith/O=netegrity."

HTTP_SM_REALM

리소스가 있는 SiteMinder 영역을 나타냅니다.

HTTP_SM_REALMOID

리소스가 있는 영역을 식별하는 영역 개체 ID 를 나타냅니다. 타사 응용 프로그램에서 이 ID 를 사용하여 정책 서버를 호출할 수 있습니다.

HTTP_SM_SDOMAIN

에이전트의 로컬 쿠키 도메인을 나타냅니다.

HTTP_SM_SERVERIDENTITYSPEC

사용자 아이덴티티를 추적하는 정책 서버 아이덴티티 티켓을 나타냅니다. 웹 에이전트는 사용자에 대해 콘텐츠를 개인화할 수 있도록 이 티켓을 사용하여 익명 인증 체계에서 보호되는 콘텐츠에 액세스합니다.

HTTP_SM_SERVERSESSIONID

사용자 세션을 식별하는 고유 문자열을 나타냅니다.

HTTP_SM_SERVERSESSIONSPEC

사용자 세션 정보가 포함된 티켓을 나타냅니다. 정책 서버만 이 정보를 디코딩하는 방법을 알고 있습니다.

HTTP_SM_SESSIONDRIFT

정책 서버에서 세션 유효성을 검사하기 전에 웹 에이전트가 캐시의 정보를 사용하여 세션을 활성으로 유지할 수 있는 시간을 나타냅니다. 정책 서버의 세션 서버가 활성화되어 있고 세션 유효성 검사 기간이 이 헤더에 대해 구성되어 설정되어 있어야 합니다.

HTTP_SM_TIMETOEXPIRE

SiteMinder 세션의 남아 있는 시간을 나타냅니다.

HTTP_SM_TRANSACTIONID

각 사용자 요청에 대해 에이전트가 생성한 고유 ID 를 나타냅니다.

HTTP_SM_UNIVERSALID

정책 서버가 생성한 유니버설 사용자 ID 를 나타냅니다. 이 ID 는 해당 고객과 관련되고 사용자를 응용 프로그램에 대해 식별하지만 사용자 로그인과는 다릅니다.

HTTP_SM_USER

인증된 사용자의 로그인 이름을 나타냅니다. 사용자가 인증서 기반 인증과 같이 로그인 시 사용자 이름을 제공하지 않을 경우 이 변수가 설정되지 않습니다.

HTTP_SM_USERDN

정책 서버에서 지정한 인증된 사용자의 고유 이름을 나타냅니다.

익명 인증 체계의 경우 이 매개 변수는 GUID(전역 고유 식별자)를 반환합니다.

HTTP_SM_USERMSG

인증 시도 후 에이전트가 사용자에게 제공하는 텍스트를 나타냅니다. 일부 인증 체계에서는 인증 요청 텍스트 또는 인증이 실패한 이유를 제공합니다.

추가 정보

기본 HTTP 헤더 변수 사용 안 함 (페이지 162)

HTTP Header-Variable 및 HTTP Cookie-Variable

HTTP-Header-Variable 및 HTTP-Cookie-Variable 특성을 사용하면 웹 에이전트에서 정적 또는 동적 이름/값 쌍 목록을 웹 응용 프로그램에 전달할 수 있습니다. 이름/값 쌍은 리소스를 요청하는 사용자마다 고유하므로 응용 프로그램에서 해당 사용자에게 표시되는 내용을 사용자 지정할 수 있습니다.

예를 들어 관리자가 사용자의 전체 이름을 저장하는 WebAgent-HTTP-Header-Variable 응답 특성을 구성합니다. 보호된 리소스에 액세스할 수 있는 권한이 사용자에게 부여되면 웹 에이전트는 해당 사용자의 전체 이름을 웹 응용 프로그램에 전달합니다. 그런 다음 사용자의 이름이 응용 프로그램에 표시되어 고객과의 관계를 설정할 수 있습니다.

웹 응용 프로그램 환경에서 HTTP-Header-Variable 응답 특성은 HTTP_*attribute_name* 변수로 나타납니다. 여기서 *attribute_name* 은 HTTP 변수의 이름(예: USERFULLNAME)입니다. 일부 응용 프로그램 서버의 경우 밑줄을 사용하면 문제가 발생할 수 있으므로 이름에 밑줄을 추가하지 않아도 됩니다.

참고: 서버에서 특성 이름의 대시(-)를 밑줄(_)로 변환하고 모든 영문자를 대문자로 변환할 수도 있습니다.

헤더 변수 및 최종 사용자 IP 주소 유효성 검사

SiteMinder 웹 에이전트는 초기 요청 이후 같은 사용자가 보내는 요청을 받으면 후속 요청과 함께 전송된 세션 쿠키의 유효성을 검사합니다. 이때 요청하는 사용자의 IP 주소와 세션 쿠키 내에 암호화된 IP 주소를 비교하는 방식을 사용합니다. 쿠키 내의 주소는 사용자의 초기 요청이 진행되는 동안 에이전트에서 생성합니다.

방화벽, 부하 분산 장치, 캐시 장치, 프록시 등과 같이 들어오는 네트워크 트래픽을 분산시키고 관리하는 데 사용되는 메커니즘은 사용자의 IP 주소를 변경하거나, 들어오는 모든 요청이 단일 IP 주소 또는 소규모 IP 주소 그룹에 속해 있는 것처럼 나타낼 수 있습니다. 따라서 웹 에이전트의 IP 검사가 의미가 없습니다. 이러한 네트워크 환경에서 웹 에이전트는 사용자 지정 HTTP 헤더를 사용하거나 안전한 프록시 IP 주소 목록을 구성하여 IP 검사를 수행할 수 있습니다.

다음 표에서는 새로운 IP 검사 기능에 대한 용어 목록을 보여 줍니다.

용어	정의
HTTP 요청 헤더	HTTP 요청의 단일 요소를 설명하는 이름/값 쌍입니다.
사용자 지정 IP 헤더	사용자가 정의한 HTTP 요청 헤더이며, 중간 HTTP 네트워크 응용 프로그램 또는 하드웨어 장치에서 요청자의 IP 주소를 저장하는 데 사용됩니다.
IP 검사	요청의 REMOTE_ADDR 와 초기 요청 후 SMSESSION 쿠키에 저장된 REMOTE_ADDR 값을 비교하는 방식으로 웹 에이전트에서 인증 요청을 검사하는 데 사용할 수 있는 기능입니다. 이 기능은 IP 유효성 검사라고도 합니다.
REMOTE_ADDR	웹 서버에 대해 요청을 시작하는 HTTP 클라이언트의 IP 주소를 나타내는 웹 서버 변수입니다. REMOTE_IP 또는 CLIENT_IP 라고도 합니다. 이는 요청자와 대상 웹 서버 사이에 프록시 서버, NAT 방화벽 또는 기타 네트워크 서비스나 장치가 있는 경우의 요청자 IP 주소와 다릅니다.
요청자	HTTP 요청의 이니시에이터이며 대개 브라우저 사용자입니다.

용어	정의
요청자 IP 주소	원래의 HTTP 요청을 시작하는 사용자의 IP 주소입니다.
싱글 사인온	세션 중에 사용자가 자격 증명을 한 번만 입력하여 보호된 웹 사이트에 안전하게 액세스할 수 있도록 하는 기능입니다.
SMSESSION 쿠키	웹 에이전트에서 싱글 사인온 상태를 추적하는 데 사용하는 HTTP 메커니즘입니다.

사용자 지정 헤더로 IP 주소 유효성을 검사하는 방법

이제 웹 에이전트는 REMOTE_ADDR 변수 대신 사용자 지정 HTTP 헤더를 사용하여 사용자의 IP 주소를 확인할 수 있습니다. 프록시 또는 다른 장치에서 사용자 지정 클라이언트 IP 헤더를 설정하고, 웹 에이전트가 들어오는 요청에서 해당 헤더를 검색하도록 구성된 경우 에이전트는 이 헤더를 클라이언트 IP 정보의 원본으로 사용합니다.

사용자 지정 헤더를 구성하는 것 외에 프록시 IP 주소 목록도 설정할 수 있습니다. REMOTE_ADDR 가 프록시 목록의 주소와 일치하는 경우 웹 에이전트는 사용자 지정 헤더에서 사용자의 IP 주소를 검색합니다. 그렇지 않은 경우에는 REMOTE_ADDR 에서 사용자의 IP 주소를 가져옵니다.

웹 에이전트가 요청자의 IP 주소를 확인한 후에는 주소가 저장되어 요청을 처리하는 데 사용됩니다. 주소를 확인할 수 없는 경우에는 IP 주소가 알 수 없음으로 설정됩니다.

웹 에이전트는 필요한 경우 디버깅을 쉽게 할 수 있도록 클라이언트 IP 주소를 확인한 원본 위치를 로깅합니다.

IP 주소 유효성 검사 구성

다음 매개 변수를 사용하여 IP 주소 검사를 구현할 수 있습니다.

CustomIpHeader

요청자 IP 주소를 찾기 위해 에이전트가 검색할 HTTP 헤더를 지정합니다. 이 매개 변수에 값을 지정하지 않으면 빈 문자열이 기본값으로 사용됩니다. 최대 길이 제한은 없으며 올바른 HTTP 헤더 값을 포함하는 문자열이 값이 될 수 있습니다.

기본값: No

예: HTTP_ORIGINAL_IP

ProxyDefinition

사용자 지정 HTTP 헤더를 사용해야 하는 프록시(예: 캐시 장치)의 IP 주소를 지정합니다. 이 사용자 지정 헤더는 에이전트가 요청자의 IP 주소를 확인하는 데 도움이 됩니다.

기본값: 기본값 없음

제한: 문자열에 IP 주소가 포함되어야 합니다. 서버 이름 또는 정규화된 DNS 호스트 이름을 사용하면 *안 됩니다*.

RequireClientIP

에이전트에서 클라이언트 IP 주소의 유효성을 검사할지 여부를 지정합니다. 이 값을 yes 로 설정할 경우 에이전트는 브라우저 쿠키의 IP 주소가 클라이언트의 IP 주소와 일치하는지 확인합니다. 두 주소가 일치하지 *않으면* 사용자 브라우저에 403 오류 메시지가 나타납니다. 쿠키에 IP 주소가 포함되지 않은 경우에는 사용자에게 자격 증명을 요청합니다.

기본값: No(클라이언트 IP 주소의 유효성이 검사되지 않음)

참고: 이러한 설정은 TransientIPCheck 및 PersistentIPCheck 매개 변수와 관련이 없습니다.

이전 릴리스의 웹 에이전트에서 IP 주소 유효성 검사

6.x QMR 2 또는 3 웹 에이전트와 이전 버전의 에이전트를 사용하는 환경에서 IP 검사를 구성하면 싱글 사인온에 영향을 줄 수 있습니다.

v6.x QMR 2 및 5.x QMR 7 이전 버전의 웹 에이전트는 요청자 IP 주소를 확인할 수 없으므로 이러한 웹 에이전트에서 생성된 SMSESSION 쿠키는 6.x QMR 2 또는 3 웹 에이전트에서 삭제될 수 있습니다. 여기에는 SDK 를 사용하여 SMSESSION 쿠키를 생성하는 사용자 지정 에이전트, 응용 프로그램 서버 에이전트 및 싱글 사인온 환경에서 SMSESSION 쿠키를 사용하는 기타 SiteMinder 에이전트가 포함됩니다.

반대로 6.x QMR 2 및 3 웹 에이전트는 요청자의 IP 주소를 확인하지만 이전 버전의 에이전트에서 확인된 주소와는 다릅니다.

HTTP 헤더 유지

새 헤더가 생성되면 기존 HTTP 헤더를 바꾸지 않고 저장하도록 웹 에이전트를 구성할 수 있습니다. 이 기능은 응용 프로그램에서 이름은 같지만 값이 다른 SiteMinder 응답을 여러 개 생성하여 헤더에 포함해야 할 경우에 유용합니다. 같은 HTTP 헤더의 인스턴스가 여러 개 있는 경우 웹 서버는 관련된 모든 헤더 값이 쉼표로 구분된 단일 헤더를 생성하여 이를 처리합니다.

기본적으로 웹 에이전트는 잘못된 헤더 값을 사용하는 응용 프로그램을 막기 위해 헤더를 유지하지 않습니다. Oracle iPlanet, Domino 및 Apache 웹 에이전트에서 HTTP 헤더를 유지하게 하려면 PreserveHeaders 매개 변수를 yes 로 설정합니다. 기본값은 no 입니다.

HTTP 헤더 리소스가 캐시되는 방식 제어

다음 매개 변수를 설정하면 웹 에이전트에서 캐시 관련 요청 헤더를 처리하는 방식을 제어할 수 있습니다.

AllowCacheHeaders

웹 에이전트가 보호된 리소스에 대한 요청을 웹 서버에 전달하기 *전에* 해당 요청에서 다음과 같은 캐시 관련 HTTP 헤더를 제거할지 여부를 지정합니다.

- if-modified-since
- if-none-match

이 설정은 브라우저가 캐시된 페이지를 사용하는지 여부에 영향을 줍니다. IgnoreExt 매개 변수의 값을 비롯하여 자동으로 권한이 부여되는 리소스에는 영향이 *없습니다*. 자동으로 권한이 부여되는 리소스의 캐시 여부는 웹 서버 및 브라우저의 설정에 따라 결정됩니다.

이 매개 변수는 다음과 같은 값을 사용합니다.

- Yes - 에이전트가 캐시 관련 HTTP 헤더를 제거하지 *않습니다*. 세션의 유효성을 검사하기 위해 SMSESSION 쿠키가 계속 추적됩니다. 세션이 만료되면 웹 에이전트는 304 "not modified" 응답과 함께 업데이트된 SMSESSION 쿠키를 보냅니다. 이 응답은 캐시에 저장되었지만 수정되지 않은 리소스에 적용됩니다. if-modified-since HTTP 헤더에 표시된 시간은 이 동작이 발생한 시기를 알려 줍니다.

  중요! 이 매개 변수를 yes 로 설정하면 적절한 캐시 제어 헤더가 없는 개인화된 응용 프로그램의 페이지가 캐시될 수 있습니다. 이 경우 예기치 않은 동작이 발생하여 브라우저에서 중요한 데이터를 디스크에 저장하게 될 수 있습니다.

- No - 에이전트가 *보호된* 리소스에 대한 요청에서만 캐시 관련 HTTP 헤더를 제거합니다. 웹 서버는 해당 요청을 무조건적인 것으로 간주하므로 캐시 내용의 유효성은 검사되지 않습니다.
- None - 웹 에이전트가 보호된 리소스 *및* 보호되지 않은 리소스에 대한 모든 캐시 관련 헤더를 제거합니다.

종료된 세션의 경우 브라우저에서는 캐시된 내용을 사용하지 *않습니다*. AllowCacheHeaders 매개 변수의 값은 무시됩니다.

이 매개 변수의 설정은 다음 매개 변수에 영향을 줍니다.

- LogOffUri - LogOffUri 매개 변수를 사용하는 경우 AllowCacheHeaders 매개 변수의 값을 no 로 설정하십시오. 그러지 않으면 해당 세션이 적절하게 종료되지 않고, 캐시된 로그아웃 페이지가 사용자에게 제공될 수 있습니다.
- **기본값:** No
- **제한:** Yes, No, None

보호된 리소스 및 보호되지 않은 리소스에서 캐시 관련 헤더를 모두 제거하려면 AllowCacheHeaders 매개 변수의 값을 none 으로 설정하십시오.

참고: HTTP 1.1 캐싱 메커니즘에 대한 자세한 내용은 RFC 2616, 섹션 13 "Caching in HTTP"(HTTP 의 캐싱)를 참조하십시오.

HTTP 헤더 인코딩 사양 설정

HTTPHeaderEncodingSpec 설정은 모든 HTTP 헤더 값과 모든 사용자 지정 HTTP-COOKIE 응답의 인코딩에 영향을 줍니다.

특정 인코딩으로 텍스트를 지역화해야 하는 웹 응용 프로그램을 지원하려면 이 매개 변수를 사용합니다. HTTP 프로토콜을 통해 브라우저와 포털 간에 서로 쿠키가 전달되므로 선택한 인코딩이 HTTP 트래픽에서 잘못된 것으로 인식하는 문자를 쿠키에 추가하는 경우 RFC-2047 HTTPWrapSpec 을 사용하십시오.

예를 들어 일부 Shift-JIS 문자의 경우 RFC-2047 로 인코딩하지 않으면 원치 않는 결과가 발생할 수 있습니다.

Kanji 문자의 경우 SHIFT-JIS 의 상위 집합인 SECP932 를 사용할 수 있습니다. 대부분의 Kanji 인코딩 및 디코딩에 SHIFT-JIS 를 사용할 수 있지만 CP932 는 더 큰 범위의 문자 집합을 포함합니다.

HTTPWrapSpec 이 사용되는 경우 데이터는 먼저 HTTPHeaderEncodingSpec 에 따라 인코딩된 후 RFC-2047 사양에 따라 추가로 인코딩됩니다.

매개 변수 구문은 다음과 같습니다.

encoding_spec, *wrapping_spec*

encoding_spec 은 UTF-8, Shift-JIS, EUC-J 또는 ISO-2022 JP 인코딩 유형 중 하나를 나타내는 텍스트 문자열입니다. 에이전트에 사용할 인코딩 유형을 지정하십시오.

wrapping_spec 은 래핑 사양이며 RFC-2047 로 설정해야 합니다. 이 변수는 생략할 수 있지만 선택한 인코딩 유형에 의해 HTTP 프로토콜과 호환되지 않는 바이트 코드가 생성될 수 있으므로 래핑 사양을 포함하는 것이 좋습니다.

특히 더블바이트 인코딩 데이터가 포함된 사용자 지정 HTTP Cookie 응답을 사용하는 경우 이에 해당됩니다. 예를 들어 일부 Shift-JIS 문자의 경우 RFC-2047 로 인코딩하지 않으면 원치 않는 결과가 발생합니다. 또한 래핑을 포함하면 수신 응용 프로그램에 인코딩 유형 및 특징이 전달되므로 응용 프로그램에서 인코딩된 텍스트를 더 잘 해석할 수 있습니다. 예를 들어 이 매개 변수를 Shift-JIS, RFC-2047 로 설정할 수 있습니다.

RFC-2047 이 사용되는 경우 에이전트는 먼저 선택한 인코딩 사양에 따라 데이터를 인코딩한 후 RFC-2047 사양에 따라 데이터를 추가로 인코딩합니다.

참고: HTTPHeaderEncodingSpec 설정을 비워 두면 기본값 UTF-8 이 지정되고 래핑이 포함되지 않습니다.

중요! 아래 조건에 해당하는 경우 에이전트가 설치된 프록시 컴퓨터에서 HTTPHeaderEncodingSpec ACO 매개 변수의 값을 다음과 같이 설정하십시오.

```
UTF-8,RFC-2047
```

- SiteMinder 에이전트로 관리 UI 를 보호하는 경우
- 관리자의 DN 값에 영어 이외의 문자가 포함된 경우

RFC 2047 준수 안 함

기본적으로 웹 에이전트는 RFC 2047 을 따릅니다. 그러나 ConformToRFC2047 매개 변수를 no 로 설정하여 RFC 2047 을 사용하지 않도록 할 수 있습니다.

이 매개 변수가 없거나 yes 로 설정되면 웹 에이전트가 RFC 2047 을 따릅니다.

HTTP 헤더에 소문자 사용(Oracle iPlanet, Apache 및 Domino 웹 서버)

서버 응용 프로그램에서 대/소문자를 구분하는 경우 에이전트의 HTTP 헤더에 대/소문자를 지정할 수 있습니다. 기본적으로 웹 에이전트는 소문자 헤더를 사용합니다.

예를 들어 Oracle iPlanet 웹 서버는 HTTP 헤더 변수를 기본적으로 소문자로 제공합니다(예: http_sm_user).

참고: IIS 는 모든 헤더에 대문자 형식을 적용하므로 IIS 웹 에이전트에 이 기능을 사용할 수 없습니다.

소문자 헤더를 사용하려면 LowerCaseHTTP 매개 변수를 yes 로 설정합니다. 대문자 헤더 변수가 필요한 경우에는 LowerCaseHTTP 를 no 로 설정하십시오.

추가 정보:

Oracle iPlanet 웹 서버 로그에 트랜잭션 ID 기록 (페이지 345)

HTTP 헤더에 레거시 변수 사용

다음 매개 변수를 사용하면 웹 에이전트에서 HTTP 헤더에 사용할 명명 규칙을 지정할 수 있습니다.

LegacyVariables

웹 에이전트가 HTTP 헤더 이름에 밑줄을 사용할지 여부를 지정합니다. HTTP 헤더에 밑줄 문자를 사용하는 Sun Java System 과 같은 일부 웹 서버의 경우 몇몇 응용 프로그램에서 문제가 발생합니다.

이 매개 변수를 no 로 설정하면 아래의 예와 같이 HTTP 헤더에 밑줄이 사용되지 않습니다.

SM*HeaderName*

이 매개 변수를 yes 로 설정하면 아래의 예와 같이 HTTP 헤더에 밑줄이 사용됩니다.

SM_*HeaderName*

기본값: (기존 에이전트) Yes

기본값: (프레임워크 에이전트) No

레거시 변수를 사용하고 웹 에이전트에서 HTTP 헤더 이름에 밑줄을 사용할 수 있게 하려면 LegacyVariables 매개 변수의 값을 yes 로 설정하십시오.

참고: Apache 2.4.x 웹 서버에서 SMUSER, SMUSERDN 등의 SiteMinder 기본 헤더를 표시하려면 LegacyVariables 매개 변수를 No 로 설정하십시오.

기본 HTTP 헤더 변수 사용 안 함

대부분의 시스템 플랫폼에서 HTTP 헤더는 4096 바이트로 제한됩니다. 이 제한을 초과하지 않고 사용자 지정 응답 변수를 위한 공간을 허용하기 위해 SiteMinder 의 기본 HTTP 헤더 변수 중 일부가 사용되지 않도록 설정할 수 있습니다.

기본 변수는 다음과 같은 범주로 그룹화됩니다.

참고: 개별 변수를 비활성화할 수 없습니다. 여러 변수가 포함된 범주만 비활성화할 수 있습니다.

- 인증 원본 변수
  - SM_AUTHDIRNAME
  - SM_AUTHDIRSERVER
  - SM_AUTHDIRNAMESPACE
  - SM_AUTHDIROID
- 사용자 세션 변수
  - SM_SERVERSESSIONID
  - SM_SERVERSESSIONSPEC
  - SM_SERVERIDENTITYSPEC
  - SM_SESSIONDRIFT
  - SM_TIMETOEXPIRE
- 사용자 이름 변수
  - SM_USER
  - SM_USERDN
  - SM_DOMINOCN

기본 HTTP 헤더 변수가 사용되지 않도록 설정하려면 다음 태스크를 수행하십시오.

- 인증 원본 변수를 사용하지 않으려면 DisableAuthSrcVars 매개 변수의 값을 yes 로 설정합니다.
- 사용자 세션 변수를 사용하지 않으려면 DisableSessionVars 매개 변수의 값을 yes 로 설정합니다.

  기본값: No

- 사용자 이름 변수를 사용하지 않으려면 DisableUserNameVars 매개 변수의 값을 yes 로 설정합니다.

  참고: 이 범주의 변수를 사용하는 응용 프로그램 또는 CA Identity Manager 를 사용 중인 경우에는 이 매개 변수의 값을 no(사용)로 설정해야 합니다.

응용 프로그램에 대한 사용자 지정 오류 처리

사용자 지정 오류 처리를 구성하면 응용 프로그램과 관련된 오류 정보를 만들 수 있습니다. 사용자를 위해 응용 프로그램을 사용자 지정하는 경우 HTTP 500, HTTP 401 및 HTTP 403 오류 페이지로 표시되는 HTML 텍스트를 추가하거나, 사용자 지정 오류 페이지 또는 응용 프로그램을 가리키는 URL 로 사용자를 리디렉션할 수 있습니다(401 오류 제외).

다음과 같은 오류 유형에 대해 사용자 지정 처리를 구성할 수 있습니다.

- 서버 오류 - 에이전트는 HTTP 500 웹 서버 오류로 인해 나타나는 오류 페이지에 ServerErrorFile 을 사용합니다. 이러한 오류 코드는 사용자 지정 오류 페이지에 전달되고 다음을 포함합니다.
  - 웹 에이전트가 필수 HTTP 헤더에서 값을 읽을 수 없어서 발생하는 문제
  - 구문 분석할 수 없거나 오류 상태가 포함된 고급 인증 쿠키
  - 웹 에이전트와 정책 서버의 연결 문제
- 액세스 거부 오류 - 에이전트는 다음 매개 변수에 지정된 파일을 사용합니다.

  Custom401ErrorFile

  사용자가 401(권한이 없음) 브라우저 오류를 받을 때 표시할 사용자 지정 HTML 페이지를 지정합니다. 사용자가 리소스에 액세스하기 위한 적절한 권한이 없을 경우 이러한 오류가 발생합니다.

  참고: 일부 웹 서버는 선택된 사용자 지정 텍스트에 자체의 고유한 텍스트를 추가합니다. 따라서 이러한 서버의 응답 페이지는 사용자 지정할 수 없습니다.

  기본값: 기본값 없음(비어 있음)
- 쿠키 필요 오류 - RequireCookies 매개 변수가 설정되는 경우 웹 에이전트는 기본 인증이 수행되는 동안 쿠키를 설정합니다. 기본 자격 증명을 사용하여 브라우저에서 이 쿠키가 반환되지 않으면 ReqCookieErrorFile 매개 변수에 지정된 오류 페이지가 반환되고 에이전트에서 웹 서버에 대한 사용자 액세스를 거부합니다.
- 교차 사이트 스크립팅 오류 - 에이전트는 HTTP 403 교차 사이트 스크립팅 오류로 인해 나타나는 오류 페이지에 대해 CSSErrorFile 매개 변수에 지정된 파일을 사용합니다. 교차 사이트 스크립팅은 웹 사이트의 보안을 손상시킬 수 있습니다.

이러한 HTML 파일 또는 응용 프로그램을 생성한 후에는 사용자 지정 오류 페이지 또는 URL 을 웹 에이전트에 알려 주어야 합니다.

참고: 프록시 서버 또는 리버스 프록시 서버로 사용되는 Apache 서버의 경우 Apache 에이전트에서 사용자 지정 SiteMinder 오류 페이지를 반환하지 않지만 표준 Apache HTTP 500 및 403 오류 페이지를 반환합니다.

오류 처리를 설정하는 방법

응용 프로그램에서 오류 메시지가 표시되는 방식을 사용자 지정하려면 다음 태스크를 수행하십시오.

- 다음 HTTP 오류에 대해 브라우저에 표시되는 HTML 텍스트를 추가합니다.
  - 500
  - 401
  - 403
- 사용자 지정 오류 페이지 또는 응용 프로그램을 가리키는 URL 로 사용자를 리디렉션합니다.

HTTP 500 및 403 오류에만 해당: 사용자를 URL 로 리디렉션하도록 에이전트를 구성하는 경우 에이전트가 URL 에 오류 코드를 추가합니다. 추가된 URL 의 다음 예를 참조하십시오.

```
?SMError=error_code,
```

표준 HTML 오류 텍스트를 추가하는 경우에는 다음 태그 사이에 HTML 코드만 지정할 수 있습니다.

```
<body>
</body>
```

에이전트를 사용자 지정 오류 페이지 또는 URL 로 디렉션하려면 다음 태스크 중 *하나*를 수행하십시오.

- 텍스트 파일이 있는 경로를 지정합니다.

각 에이전트 구성 매개 변수의 값에 URL 을 입력합니다.

다음 표에서는 오류와 기타 이벤트 및 이에 해당하는 에이전트 구성 매개 변수를 보여 줍니다.

사용자 지정 응답을 설정할 오류 유형	사용할 구성 매개 변수
서버 오류	ServerErrorFile
액세스 거부 오류	Custom401ErrorFile
쿠키 필요 오류	ReqCookieErrorFile
CSS 문자 오류	CSSErrorFile

오류 파일은 응용 프로그램의 어디에나 있을 수 있습니다.

중요! 사용자 지정 오류 페이지로 구성하는 URL 을 보호되지 않은 상태로 두십시오.

참고: 응용 프로그램의 URL 에 HTML 태그가 필요한 경우 태그의 문자를 인코딩해야 합니다. HTML 문자를 인코딩하는 방법을 보려면 http://www.cert.org/tech_tips/ 사이트를 참조하십시오.

다음 예제에서는 오류 파일의 파일 경로 및 URL 을 보여 줍니다. 예제의 구문은 로컬 구성 파일에 대한 것입니다. 이러한 매개 변수를 에이전트 구성 개체에 설정할 수도 있습니다.

파일 경로:

```
CSSErrorFile="C:\error\error.txt"
ReqCookieErrorFile="C:\custompages\error.txt"
ServerErrorFile="C:\error\error.txt"
Custom401ErrorFile="C:\error\accessdenied.txt"
```

URL:

```
CSSErrorFile="http://www.mycompany.com/error.jsp"
ReqCookieErrorFile="http://www.myorg.com/error.asp"
ServerErrorFile="http://www.mycompany.com/error.jsp"
```

추가 정보

응용 프로그램에 대한 사용자 지정 오류 처리 (페이지 164)

사용자 지정 401 페이지에 대한 참고 사항

- Custom401errorfile 매개 변수를 URL 로 설정하면 안 됩니다.
- Custom401errorfile 의 값(사용 가능 여부는 관계없음)이 있는 경우 에이전트는 해당 파일이 변경되었는지 여부를 60 초 간격으로 확인합니다. 그러나 응답은 실제로 정적 상태여야 합니다. 예를 들어 "*user_name* 거부됨"과 같은 동적 메시지를 삽입할 수 없습니다.

  Custom401errorfile 값이 있으면 이 값의 사용 여부에 관계없이 재확인이 트리거되므로 에이전트를 다시 시작하지 않고 오류를 수정할 수 있습니다. 수정 내용은 다음에 확인할 때 적용됩니다.
- 사용자 지정 메시지 파일 텍스트는 다른 오류에 의해 노출되지 않습니다. 파일 경로 이름은 시작할 때와 오류가 발생할 때 로깅됩니다.
- 사용자 지정 범위는 웹 서버에 의해 제한되며 고유의 텍스트를 응답에 추가할 수 있습니다.
- 사용자 지정 텍스트 파일의 크기는 시스템 파일 크기 제한에 따라 결정됩니다.

제 10 장: 가상 서버 구성

이 섹션은 다음 항목을 포함하고 있습니다.

가상 서버 지원을 설정하는 방법

가상 서버는 물리적 서버에 구성하는 논리적 엔터티입니다. 이 논리적 엔터티는 독립 서버로 사용됩니다. 가상 서버를 사용하면 하나의 물리적 서버에서 여러 웹 사이트를 호스트할 수 있습니다. 예를 들어 가상 서버를 사용하여 www.mysite.com 과 www.yoursite.com 을 둘 다 호스트하는 서버를 설정할 수 있습니다.

다음과 같은 항목을 가상 서버에 할당할 수 있습니다.

- 고유한 IP 주소
- 물리적 서버와 공유하는 IP 주소
- 다른 가상 서버와 공유하는 IP 주소

각 웹 서버에 대해 웹 에이전트를 하나만 구성하지만 에이전트 아이덴티티를 여러 개 구성하여 가상 서버를 보호할 수 있습니다. 두 명의 사용자가 각각 www.mysite.com 과 www.yoursite.com 을 통해 서버에 액세스하는 경우 각 서버는 에이전트 아이덴티티로 보호됩니다. 각 가상 서버에 대해 에이전트 아이덴티티를 생성하면 사이트마다 고유한 영역 및 규칙을 정의할 수 있다는 이점이 있습니다.

웹 에이전트에 대해 정의하는 설정은 해당 웹 서버 인스턴스에 대해 정의하는 모든 가상 서버에 적용되지만 각 가상 서버는 요청을 독립적으로 처리하고 정책 서버는 각 가상 서버 요청을 개별적으로 처리합니다. 가상 서버에 대한 자세한 내용 및 가상 서버를 구성하는 방법을 보려면 웹 서버 설명서를 참조하십시오.

가상 서버 지원을 구성하려면 다음 태스크 중 *하나*를 수행하십시오.

- 각 가상 서버에 대한 에이전트 아이덴티티를 정의 및 추가하고 AgentName 매개 변수의 값을 지정한 후 가상 서버의 IP 주소 또는 호스트 헤더 이름을 할당합니다.
- 고유하게 식별해야 할 가상 서버에 대해서만 에이전트 아이덴티티를 정의합니다.
- 기본 에이전트 이름을 설정합니다.

참고: Oracle iPlanet 웹 서버의 인스턴스가 여러 개인 경우, 예를 들어 HTTP 통신용 서버와 HTTPS 통신용 서버가 있으면 WebAgent.conf 파일도 두 개입니다. 각 파일에는 에이전트 아이덴티티가 여러 개 있을 수 있습니다. Oracle iPlanet 은 이전 이름이 Sun ONE/iPlanet 인 웹 서버를 말합니다.

가상 서버의 웹 에이전트 아이덴티티 할당

각 가상 서버의 추가 웹 에이전트는 실제로 *정의되지* 않지만 웹 에이전트 아이덴티티가 *할당됩니다*. 고유한 액세스 요구 사항이 있는 가상 서버 또는 고유 영역을 보호하려면 각 서버에 고유한 에이전트 아이덴티티를 할당하고 다른 모든 가상 서버에는 기본 에이전트 이름을 사용합니다. 이렇게 하면 별도의 보호가 필요한 영역을 호스트하는 가상 서버를 보호하면서 SiteMinder 설치를 빠르게 구성할 수 있습니다.

AgentName 매개 변수 및 연결된 IP 주소는 정책 저장소에 정의된 대로 웹 서버 인터페이스와 에이전트 이름 사이의 매핑을 제공합니다. 올바른 규칙 집합 및 정책을 적용하려면 웹 에이전트가 적절한 에이전트 이름 컨텍스트에서 에이전트 API 호출을 수행해야 합니다. 정책 저장소에 대한 매핑에 에이전트 이름 또는 IP 주소가 할당되어 있지 않으면 웹 에이전트는 가상 서버에 대해 DefaultAgentName 매개 변수의 값을 사용합니다.

고유한 에이전트 아이덴티티를 사용하여 가상 서버를 보호하려면 AgentName 매개 변수에 각 가상 서버의 웹 에이전트를 추가합니다. 각 가상 서버에 대해 개별적으로 웹 에이전트를 추가하면 가상 서버마다 고유한 영역 및 규칙을 정의할 수 있습니다.

웹 에이전트 아이덴티티를 할당하려면

1. 에이전트 이름과 IP 주소를 쉼표로 구분하여 입력합니다.
2. 여러 가상 서버에서 같은 IP 주소를 공유하지만 서로 다른 포트를 사용하는 경우 IP 주소와 연결된 포트 번호를 지정합니다(예: 112.12.12.1:8080). 기본 포트를 사용 중인 경우에는 포트 번호가 필요하지 않습니다.
3. 에이전트를 여러 개 추가하려면 다음과 같이 각 항목을 한 행에 하나씩 입력합니다.

   ```
   agentname="agent1,123.123.12.12:8080"
   agentname="agent2,123.123.12.12:8081"
   agentname="agent3,123.123.12.13"
   ```

4. 에이전트 아이덴티티를 추가하는 경우 같은 구성을 사용하여 관리 UI 에서 아이덴티티를 정의해야 합니다. 에이전트 아이덴티티가 에이전트 구성에 정의된 것과 동일하게 관리 UI 에서 정의되는지 확인하십시오.

AgentName 매개 변수에 항목이 없는 경우 SiteMinder 는 가상 서버에 대해 DefaultAgentName 값을 사용합니다.

참고: DefaultAgentName 을 변경하는 경우 에이전트에 대해 정의된 것과 동일하게 관리 UI 에서 정의되는지 확인하십시오.

웹 에이전트에서 무시할 가상 서버 지정

사이트의 웹 서버가 가상 서버를 여러 개 지원하는 경우 이러한 가상 서버의 리소스 중 일부를 웹 에이전트에서 보호하지 않도록 설정할 수 있습니다. 웹 에이전트에서 보호할 웹 서버 콘텐츠를 간단한 방법으로 구별할 수 있게 하려면 다음 매개 변수를 사용합니다.

IgnoreHost

웹 에이전트에서 무시하도록 하려는 가상 서버의 정규화된 도메인 이름을 지정합니다. 이러한 가상 서버의 리소스는 자동으로 권한이 부여되고 요청한 클라이언트에 관계없이 웹 에이전트는 이러한 리소스에 대한 액세스 권한을 항상 부여받습니다. 권한 부여 결정이 정책이 아닌 웹 에이전트의 구성을 기반으로 이루어집니다.

IgnoreExt 및 IgnoreURL 등의 다른 자동 권한 부여 설정을 확인하기 전에 무시되는 호스트의 목록을 먼저 확인합니다. 따라서 이중 점 규칙은 무시되는 호스트의 리소스에 대한 권한 부여 요청을 정책 서버에 트리거하지 않으며 확장명에 의해 무시되지 않습니다.

IgnoreHost 매개 변수에서 URL 항목의 호스트 부분은 웹 에이전트가 요청된 리소스의 호스트 헤더에서 읽는 내용과 정확히 일치해야 합니다.

참고: 이 값은 대/소문자를 구분합니다.

URL 에서 특정 포트를 사용하면 이 포트를 지정해야 합니다.

중앙에서 관리되는 에이전트의 경우 에이전트 구성 개체에서 다중값 매개 변수를 사용하여 여러 서버를 나타낼 수 있습니다. 로컬 구성 파일로 구성된 에이전트의 경우 각 호스트를 파일의 각 줄에 나열하십시오.

예: (포트가 지정되어 표시되는 URL)

IgnoreHost="myserver.example.org:8080"

예: (로컬 구성 파일)

IgnoreHost="my.host.com"

IgnoreHost="your.host.com"

기본값: 기본값 없음

웹 에이전트에서 무시할 가상 서버를 지정하려면 다음 태스크를 수행하십시오.

- 중앙 구성의 경우 무시할 서버를 에이전트 구성 개체에 추가합니다. 서버가 여러 개이면 매개 변수에 다중값 설정을 사용해야 합니다.
- 로컬 구성의 경우 각 서버를 로컬 구성 파일에서 한 행에 하나씩 추가합니다.

지정된 URL 을 사용하는 리소스는 웹 에이전트에서 무시되고 이러한 리소스에는 자동으로 액세스됩니다.

추가 정보

복잡한 URI 처리 (페이지 112)

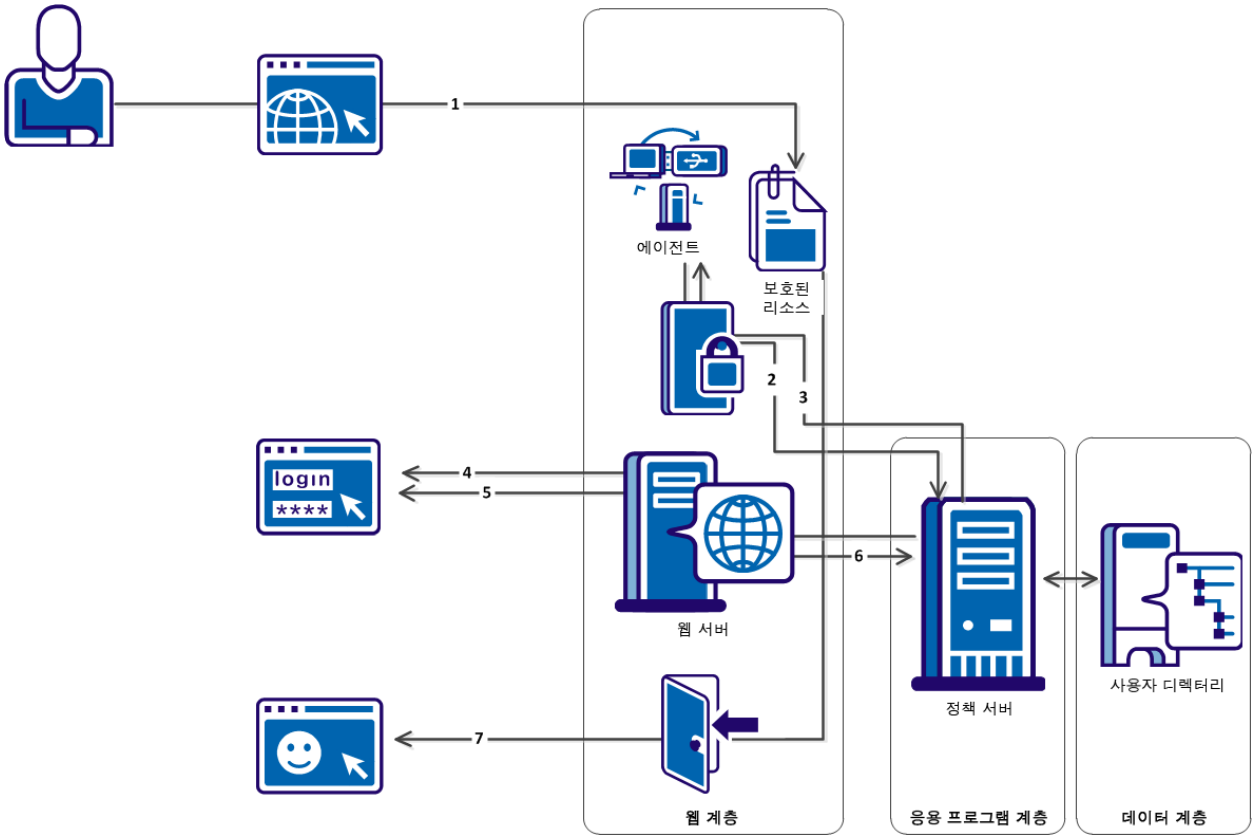
제 11 장: 양식 인증

이 섹션은 다음 항목을 포함하고 있습니다.

자격 증명 수집기의 요청 처리 방식

다음 그림에서는 FCC(양식 자격 증명 수집기)가 보호된 리소스에 대한 요청을 처리하는 방식을 보여 줍니다.

참고: 쿠키 공급자는 싱글 사인온에 대해 다른 프로세스를 사용합니다.



위의 그림에 나오는 프로세스는 다음과 같은 단계로 설명됩니다.

1. 사용자가 리소스에 대한 액세스를 요청합니다.
2. 에이전트가 정책 서버에 연결하여 해당 리소스가 보호되는지 확인합니다.
3. 정책 서버는 자격 증명 수집기가 리소스를 보호하고 있음을 에이전트에 알리고 사용 중인 자격 증명 수집기의 유형을 지정합니다.
4. 에이전트는 자격 증명 수집기의 URL 에 쿼리 데이터, 대상 리소스 및 암호화된 에이전트 이름을 추가합니다. 그런 다음 사용자를 해당 자격 증명 수집기로 리디렉션합니다.
5. 자격 증명 수집기 유형에 따라 다음 동작 중 *하나*가 발생합니다.
   - FCC 는 양식을 표시한 후 사용자의 자격 증명을 수집합니다.
   - NTC 는 사용자의 NT 자격 증명을 수집합니다.
   - SCC 는 사용자의 자격 증명을 수집합니다.
   - 사용 가능한 인증서가 없을 경우에는 SFCC 가 양식을 표시한 후 사용자 자격 증명을 수집합니다.
6. 자격 증명 수집기에서 직접 정책 서버에 사용자가 로그인하도록 합니다. 그러면 정책 서버에서 세션을 생성합니다.
7. 에이전트가 세션의 유효성을 검사하고 리소스에 액세스할 수 있는 권한을 사용자에게 부여합니다.

참고: SSL 인증 체계에 대한 자세한 내용은 정책 서버 설명서를 참조하십시오.

자격 증명 수집기의 MIME 유형

각 자격 증명 수집기는 MIME 유형과 연결됩니다. MIME 유형은 사용자가 리소스를 요청할 때 인증 챌린지를 제공하는 수집기를 나타냅니다. 다음 표에서는 각 유형을 보여 줍니다.

자격 증명 수집기	MIME 유형
양식 자격 증명 수집기	.fcc
SSL 자격 증명 수집기	.scc
쿠키 공급자	.ccc

자격 증명 수집기	MIME 유형
NTLM 자격 증명 수집기	.ntc
SSL 양식 자격 증명 수집기	.sfcc
Kerberos 자격 증명 수집기	.kcc

자격 증명 수집기를 사용하는 인증 체계를 구성하거나 여러 쿠키 도메인에 대해 싱글 사인온 환경을 설정하는 경우 관련된 MIME 유형은 다음과 같이 인증 체계 또는 싱글 사인온 구성에서 참조하는 파일의 확장명으로 사용됩니다.

- 여러 쿠키 도메인에 대해 싱글 사인온을 구성하는 경우 다음과 같이 URL 을 입력하여 쿠키 공급자를 식별합니다.

  http://myserver.company.com:80/siteminderagent/SmMakeCookie.ccc

  SmMakeCookie.ccc 는 기본 쿠키 공급자 이름입니다. 이 이름을 사용하거나 고유한 이름을 생성할 수 있지만 싱글 사인온을 시작하려면 .ccc 확장명을 사용해야 합니다.

- Windows 인증 체계를 사용하려는 경우 기본 대상 파일은 다음과 같습니다.

  /siteminderagent/ntlm/creds.ntc

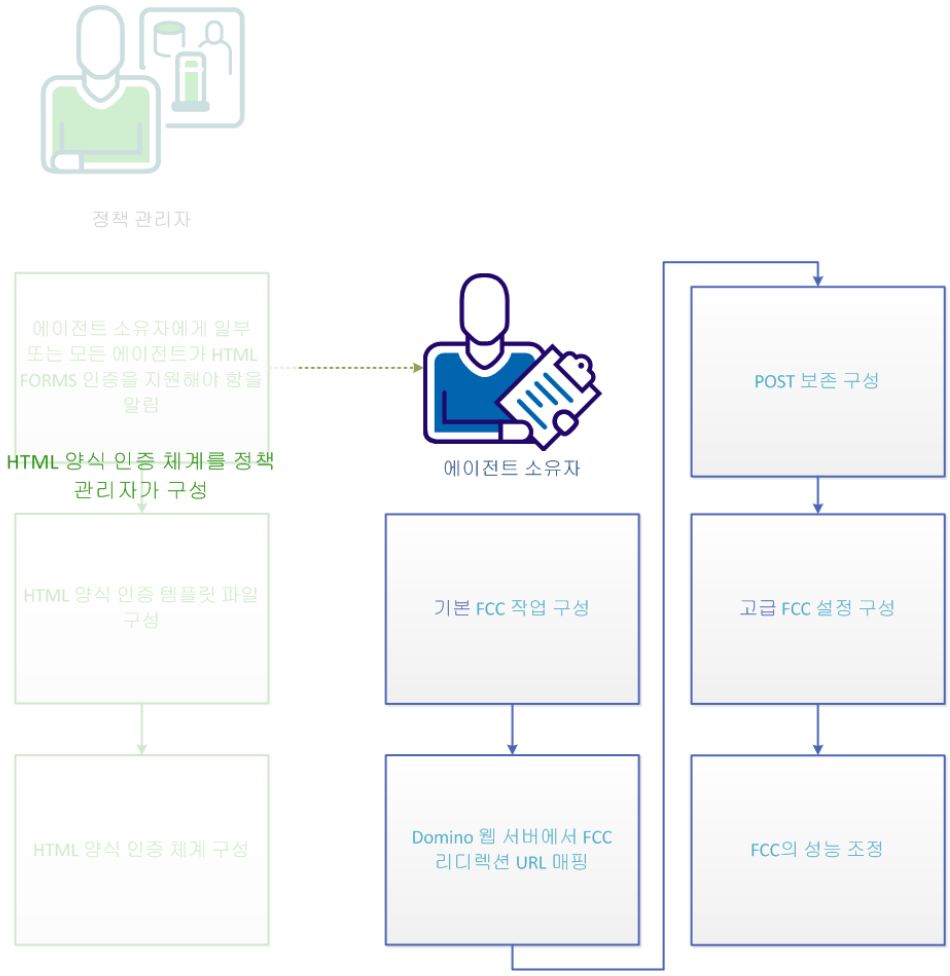
  올바른 MIME 유형이 확장명으로 지정된 파일을 사용해야 합니다.

FCC 및 SFCC 는 에이전트가 설치된 웹 서버에 실제 파일이 있어야 하는 자격 증명 수집기입니다. 이러한 수집기는 양식 기반 인증 체계에 사용됩니다. .fcc 및 .sfcc 템플릿은 사용자에게 제공되는 HTML 양식을 정의하는 데 필요합니다.

SiteMinder 에이전트에서 HTML 양식 인증을 지원하도록 구성하는 방법

SiteMinder 에서 HTML 양식 인증을 사용하여 사용자 아이덴티티의 유효성을 검사하도록 구성하려면 정책 관리자와 에이전트 소유자가 모두 구성 프로세스를 수행해야 합니다. 이 시나리오에서는 정책 관리자가 하나 이상의 에이전트에서 HTML 양식 인증 지원이 필요하다고 알릴 때 에이전트 소유자가 수행해야 하는 프로세스에 대해 설명합니다.

참고: 정책 관리자가 HTML 양식 인증을 구성하는 방법에 대한 자세한 내용은 관련 시나리오 *Configure HTML Forms Authentication*(HTML 양식 인증 구성)을 참조하십시오.



1. 기본 FCC 인증 구성 (페이지 180)
2. Domino 웹 서버에서 FCC 리디렉션을 위한 URL 매핑 (페이지 185)
3. POST 보존 구성 (페이지 185)
4. 고급 FCC 설정 구성 (페이지 189)
5. FCC 성능 조정 (페이지 206)

기본 FCC 작업 구성

HTML 양식 인증 체계로 리소스를 보호하는 에이전트의 *FCC*(*양식 자격 증명 수집기*) 구성 요소를 구성하려면 몇 가지 기본적인 구성 절차를 수행하십시오.

1. IIS 웹 서버 또는 Domino 웹 서버를 사용하는 경우 FCC 에 대한 MIME 유형 매핑을 구성 (페이지 180)합니다.

   참고: 에이전트 구성 마법사는 다음 유형의 웹 서버에 대해 SiteMinder 자격 증명 수집기에서 사용하는 올바른 MIME 유형을 자동으로 설정합니다.

   - Apache 및 Apache 기반 웹 서버
   - Oracle iPlanet 웹 서버

2. FCC 에서 사용할 웹 서버와 에이전트 아이덴티티를 매핑합니다.
3. 필요한 경우 다음과 같은 추가 설정을 구성합니다.
   - FCC 및 SCC 에서 에이전트 이름을 정규화된 호스트 이름으로 사용 (페이지 181)
   - 단일 리소스 대상을 사용하도록 FCC 구성 (페이지 181)
   - 자격 증명 수집기 리디렉션을 위해 상대 대상 사용 (페이지 182)
   - 유효한 대상 도메인 정의 (페이지 84)
   - 유효한 페더레이션 대상 도메인 정의 (페이지 183)

IIS 및 Domino 웹 서버에서 FCC 에 대한 MIME 유형 매핑 구성

IIS 및 Domino 웹 서버의 경우 웹 에이전트 구성에서 FCC 에 대한 MIME 유형 매핑을 구성하려면 FCCExt 에이전트 구성 매개 변수를 지정하십시오. MIME 유형 매핑은 파일 확장명으로 표시됩니다. 기본값을 사용하는 것이 좋습니다.

FCCExt

FCC 에 대한 MIME 유형 매핑을 지정합니다.

기본값: .fcc

제한: 올바른 파일 확장명

예: .myfcc

참고: 기본 확장명을 사용하지 않으려는 경우나 기본값이 이미 사용 중인 경우에는 원하는 확장명을 입력하십시오. 예를 들어 FCC 에 대해 FCCExt 를 .myfcc 로 설정하고 FCC 템플릿 이름을 이 확장명을 사용하는 이름(예: login.myfcc)으로 바꾸면 에이전트는 .myfcc 로 끝나는 URL 을 HTML 양식 인증 요청으로 인식합니다.

FCC 및 SCC 에서 에이전트 이름을 정규화된 호스트 이름으로 사용

양식 및 SSL 자격 증명 수집기에서 대상 URL 의 정규화된 호스트 이름을 에이전트 이름으로 사용할 수 있게 하려면 AgentNamesAreFQHostNames 구성 매개 변수를 정의해야 합니다.

예를 들어 AgentNamesAreFQHostNames 매개 변수가 Yes 로 설정되면 다음 URL 문자열의 www.nete.com 부분이 웹 에이전트 이름으로 사용됩니다.

```
url?A=1&Target=http://www.nete.com/index.html
```

다음과 같은 경우 자격 증명 수집기에서 이 매개 변수를 사용합니다.

- 에이전트 이름이 대상 에이전트의 URL 에 추가되지 않은 경우. 타사 에이전트를 사용하는 경우가 해당될 수 있습니다.
- AgentName 매개 변수에 에이전트와 호스트 이름 매핑을 구성하지 않은 경우

AgentNamesAreFQHostNames 매개 변수가 No 로 설정되는 경우 자격 증명 수집기는 DefaultAgentName 매개 변수 값을 대상 웹 에이전트의 이름으로 사용합니다.

단일 리소스 대상을 사용하도록 FCC 구성

사용자를 단일 리소스로 안내하도록 FCC 를 구성하려면 대상을 login.fcc 템플릿 파일에 하드 코드로 작성해야 합니다.

다음 단계를 수행하십시오.

1. *agent_home*/Samples 에 있는 login.fcc 파일을 엽니다.
2. @target=*target_resource* 를 FCC 에 추가합니다.
3. 다음 항목을 추가합니다.

   @smagentname=*agent_name_protecting_resource*

   예: @smagentname=mywebagent

4. EncryptAgentName 매개 변수를 no 로 설정합니다. 에이전트 이름을 파일에 하드 코드로 작성한 후에는 이름을 암호화할 수 있는 방법이 없으므로 이 매개 변수가 필요합니다.
5. 이 FCC 를 사용하는 다른 에이전트에 대해 EncryptAgentName 을 no 로 설정합니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

자격 증명 수집기 리디렉션을 위해 상대 대상 사용

필요한 경우 자격 증명 수집기 및 대상 리소스로 요청을 전달할 때 정규화된 URL 대신 상대 URI 를 사용하도록 에이전트에 지시할 수 있습니다. 상대 URI 를 사용하면 웹 에이전트가 설치된 다른 시스템의 자격 증명 수집기에서 요청을 처리하는 문제를 방지할 수 있습니다.

참고: 이 설정은 CCC(쿠키 자격 증명 수집기)를 *제외*한 모든 자격 증명 수집기에 적용됩니다. CCC 에서는 이 매개 변수에 정규화된 도메인 이름을 사용해야 합니다. 상대 URI 를 사용하면 OnAuthAccept 응답이 CCC 와 제대로 작동하지 않습니다.

일반적으로 정규화된 URL 이 자격 증명 수집기 URL 에 추가됩니다. 예를 들면 다음과 같습니다.

url?A=1&Target=http://www.nete.com/index.html.

상대 URI 만 사용하려면 TargetAsRelativeURI 매개 변수를 yes 로 설정합니다. 이 매개 변수를 yes 로 설정하면 자격 증명 수집기 URL 에 추가되는 대상 매개 변수가 상대 대상입니다(예: url?A=1&Target=/index.html). 결국 자격 증명 수집기가 대상 리소스를 보호하는 웹 에이전트로 다시 리디렉션하는 경우 상대 리디렉션이 됩니다. 또한 웹 에이전트는 슬래시(/)로 시작하지 않는 대상을 거부합니다.

이 매개 변수의 기본값은 no 이므로 항상 정규화된 URL 이 사용됩니다.

유효한 대상 도메인 정의

유해한 웹 사이트로 사용자를 리디렉션할 수 있는 피싱 시도로부터 리소스를 보호하도록 SiteMinder 에이전트를 구성하려면 다음 구성 매개변수를 설정합니다.

ValidTargetDomain

자격 증명 수집기가 사용자를 리디렉션할 수 있는 도메인을 지정합니다. URL 의 도메인이 이 매개 변수에 설정된 도메인과 일치하지 않으면 리디렉션이 거부됩니다.

기본값: No

FCC(양식 자격 증명 수집기)를 포함한 모든 고급 인증 체계에서 이 매개변수를 지원합니다.

ValidTargetDomain 매개 변수는 처리하는 동안 유효한 대상 도메인을 식별합니다. 사용자가 리디렉션되기 전에 에이전트는 리디렉션 URL 의 값을 이 매개 변수의 도메인과 비교합니다. 이 매개 변수가 없으면 에이전트는 사용자를 임의의 도메인에 있는 대상으로 리디렉션합니다.

ValidTargetDomain 매개 변수는 유효한 각 도메인에 대해 하나씩 지정되는 여러 개의 값을 포함할 수 있습니다.

로컬 웹 에이전트 구성의 경우 다음과 같이 각 도메인에 대한 항목을 한 행에 하나씩 지정합니다.

```
validtargetdomain=".xyzcompany.com"

validtargetdomain=".abccompany.com"
```

유효한 페더레이션 대상 도메인 정의

SiteMinder 가 레거시 페더레이션 SP 로 사용되는 경우 SAML 2.0 트랜잭션을 위한 IPD(아이덴티티 공급자 검색) 프로필을 구성할 수 있습니다. 사용자는 인증 요청에 대한 어설션을 생성하는 IdP 를 선택할 때 IPD 를 사용할 수 있습니다.

검색하는 동안 사용자가 악의적인 웹 사이트로 리디렉션되지 않게 할 수 있습니다. 이렇게 하려면 인증 요청을 충족하는 IdP 도메인의 유효성을 검사하도록 웹 에이전트를 구성해야 합니다.

유효성 검사 프로세스를 사용하려면 다음 매개 변수의 값을 설정합니다.

ValidFedTargetDomain

(페더레이션만 - SAML 2.0). IPD(아이덴티티 공급자 검색)를 구현할 때 페더레이션 환경에 대해 유효한 모든 도메인을 나열합니다.

SiteMinder IPD(아이덴티티 공급자 검색) 서비스가 요청을 받으면 해당 요청에서 IPDTarget 쿼리 매개 변수를 검사합니다. 이 쿼리 매개 변수는 검색 서비스가 요청을 처리한 후 리디렉션해야 하는 URL 을 나열합니다. IdP 의 경우 IPDTarget 은 SAML 2.0 싱글 사인온 서비스입니다. SP 의 경우 대상은 일반 도메인 쿠키를 사용하려고 요청하는 응용 프로그램입니다.

페더레이션 웹 서비스가 IPDTarget URL 의 도메인과 ValidFedTargetDomain 매개 변수에 지정된 도메인 목록을 비교합니다. URL 도메인이 ValidFedTargetDomain 에 구성된 도메인 중 하나와 일치하면 IPD 서비스가 사용자를 IPDTarget 매개 변수에 지정된 URL 로 리디렉션합니다. 이 경우 SP 의 URL 로 리디렉션됩니다.

일치하는 도메인이 없으면 IPD 서비스가 사용자 요청을 거부하고 브라우저를 통해 "403 사용 권한 없음" 오류가 수신됩니다. 또한 FWS 추적 로그와 affwebservices 로그에서 오류가 보고됩니다. 이러한 메시지는 IPDTarget 의 도메인이 유효한 페더레이션 대상 도메인으로 정의되지 않았음을 나타냅니다.

ValidFedTargetDomain 설정을 구성하지 않는 경우 유효성 검사가 수행되지 않고 사용자가 대상 URL 로 리디렉션됩니다.

제한: 페더레이션된 네트워크 내의 유효한 도메인

기본값: 기본값 없음

ValidFedTargetDomain 매개 변수에 유효한 도메인을 지정합니다. 이 설정은 다중값 매개 변수이므로 도메인을 여러 개 입력할 수 있습니다.

로컬 구성 파일을 수정하는 경우에는 다음과 같이 각 도메인을 하나씩 추가해야 합니다.

validfedtargetdomain=".examplesite.com"

validfedtargetdomain=".abccompany.com"

아이덴티티 공급자 검색 프로필에 대한 자세한 내용은 *Federation Security Services Guide*(Federation Security Services 안내서)를 참조하십시오.

Domino 웹 에이전트를 사용하여 FCC 리디렉션을 위한 URL 매핑

양식 인증 체계를 사용하여 Domino 뷰(.nsf) 리소스를 보호하려면 양식 자격 증명 수집기로 리디렉션되기 전에 URL 을 매핑해야 합니다.

다음 단계를 수행하십시오.

1. DominoNormalizeUrls 매개 변수의 값을 yes 로 설정합니다.
2. DominoMapUrlForRedirect 매개 변수의 값을 yes 로 설정합니다.

   FCC 로 리디렉션되기 전에 Domino URL 이 매핑됩니다.

POST 보존 구성

SiteMinder 에서는 사용자가 FCC 양식에 포스트하는 데이터를 자동으로 보존합니다. 따라서 POST 작업 중에 시간이 만료되거나 작업이 중단되더라도 양식의 데이터가 손실되지 않습니다.

기존 에이전트와 프레임워크 에이전트를 함께 사용하는 혼합 환경의 경우 다음과 같은 추가 구성 단계가 필요합니다.

- 프레임워크 에이전트와 기존 에이전트 간의 POST 보존 사용 (페이지 186)
- POST 보존 페이지 사용자 지정 (페이지 187)

POST 보존을 사용하지 않으려면 이 기능을 비활성화 (페이지 94)할 수 있습니다.

다음과 같은 경우에는 POST 보존이 지원되지 않습니다.

- ACE 인증
- FCC 에 포스트하는 사용자 지정 인증 체계

프레임워크 에이전트와 기존 에이전트 간의 POST 보존 사용

프레임워크 에이전트와 기존 에이전트는 POST 보존 데이터를 처리하는 방식이 서로 다릅니다. SiteMinder 환경에서 프레임워크 에이전트와 기존 에이전트의 조합이 사용되고, 한 유형의 에이전트에서 호스트되는 리소스가 다른 유형의 에이전트에서 호스트되는 FCC(양식 자격 증명 수집기)로 보호되는 경우 다음 매개 변수를 사용하여 올바른 템플릿 파일을 지정해야 합니다.

PostPreservationFile

다음 POST 보존 템플릿 파일 중 *하나*에 대한 경로를 지정하여 기존 에이전트와 프레임워크 에이전트 간에 POST 보존 데이터를 전송하도록 설정합니다.

- tr2fw.pptemplate - 기존 에이전트를 실행하는 서버에서 호스트되는 리소스가 프레임워크 에이전트에서 실행되는 FCC 로 보호되도록 지정합니다.
- fw2tr.pptemplate - 프레임워크 에이전트를 실행하는 서버에서 호스트되는 리소스가 기존 에이전트에서 실행되는 FCC 로 보호되도록 지정합니다.

기본값: 기본값 없음

예: *web_agent_home*/samples/forms/fw2tr.pptemplate

프레임워크 에이전트와 기존 에이전트 간의 POST 보존이 사용되도록 설정하려면

1. 다른 유형의 에이전트에서 실행 중인 FCC 로 보호되는 리소스를 확인합니다.
   a. 프레임워크 에이전트에서 실행 중인 FCC 로 보호되는 리소스를 호스트하는 기존 에이전트 목록을 생성합니다.
   b. 기존 에이전트에서 실행 중인 FCC 로 보호되는 리소스를 호스트하는 프레임워크 에이전트 목록을 생성합니다.
2. 리소스를 호스트하는 기존 에이전트, 즉 1a 단계에서 확인된 에이전트에 대해 PostPreservationFile 매개 변수의 값을 tr2fw.pptemplate 파일 경로로 설정합니다.
3. 리소스를 호스트하는 프레임워크 에이전트, 즉 1b 단계에서 확인된 에이전트에 대해 PostPreservationFile 매개 변수의 값을 fw2tr.pptemplate 파일 경로로 설정합니다.

4. 기존 에이전트와 통신하는 모든 프레임워크 웹 에이전트에 대해 다음 매개 변수의 값을 yes 로 설정합니다.

   LegacyPostPreservationEncoding

   웹 에이전트가 이전의 기존 웹 에이전트 또는 최신 프레임워크 웹 에이전트와 호환되는 방식으로 POST 보존 데이터를 인코딩할지 여부를 지정합니다. 이 매개 변수의 값을 yes 로 설정하면 인코딩이 기존 웹 에이전트와 호환됩니다. 이 매개 변수의 값을 no 로 설정하면 인코딩이 프레임워크 웹 에이전트*와만* 호환됩니다.

   기본값: No

5. 리소스를 호스트하는 웹 서버를 다시 시작합니다.

   프레임워크 에이전트와 기존 에이전트 간의 POST 보존이 가능하도록 설정됩니다.

POST 보존 페이지 사용자 지정

POST 작업 중에 시간이 만료되거나 작업이 중단될 경우 POST 보존 페이지가 표시됩니다. 대부분의 경우 POST 보존 페이지는 1 초 미만으로 나타납니다. 그러나 포스트되는 양식 데이터의 양이 많으면 POST 보존 페이지가 5 초 정도 표시될 수 있습니다.

기본적으로 POST 보존 페이지에는 다음과 같은 텍스트가 표시됩니다.

```
This page is used to hold your data while you are being authorized for your request.
You will be forwarded to continue the authorization process. If this does not happen
automatically, please click the Continue button below.
```

또한 POST 보존 페이지에는 사용자가 데이터를 응용 프로그램에 다시 포스트할 수 있도록 "Continue"(계속) 단추도 포함되어 있습니다.

POST 보존 페이지를 사용자 지정하려면 POST 보존 템플릿 파일을 생성해야 합니다.

기본 페이지의 일반적인 구조는 다음과 같습니다.

```
<HTML><HEAD><TITLE></TITLE></HEAD><BODY onLoad="document.AUTOSUBMIT.submit();">
This page is used to hold your data while you are being authorized for your
request.<BR><BR>
You will be forwarded to continue the authorization process. If this does not happen
automatically, please click the Continue button below.
<FORM NAME="AUTOSUBMIT" METHOD="POST" ACTION="$$smpostlocation$$">
<$$smpostdata$$>
<INPUT TYPE="SUBMIT" VALUE="Continue">
</FORM></BODY></HTML>
```

POST 보존 템플릿에는 POST 보존 페이지를 렌더링할 때 웹 에이전트에서 확장하는 다음 두 요소가 포함되어야 합니다.

$$smpostlocation$$

POST 보존의 첫 번째 단계 중에 자격 증명 수집기 URL 로 확장되고 POST 보존의 두 번째 단계 중에 보호된 리소스 URL 로 확장됩니다.

$$smpostdata$$

POST 보존 단계에 해당하는 두 위치에 데이터를 올바른 형식으로 포스트하는 HTML 을 포함하기 위해 확장됩니다.

이러한 요소를 제거하거나 변경하지 마십시오.

그러나 다른 요소는 변경할 수 있습니다. 예를 들어 "Continue"(계속) 단추를 제거하려면 해당 단추를 정의하는 <INPUT> 요소를 제거합니다.

```
<INPUT TYPE="SUBMIT" VALUE="Continue">
```

샘플로 제공되는 두 개의 POST 보존 템플릿 파일(fw2tr.pptemplate 및 tr2fw.pptemplate)은 다음 위치에 있습니다.

- **UNIX**: *web_agent_home*/samples_default/forms/
- **Windows**: *web_agent_home*\samples_default\forms\

web_agent_home

웹 서버에서 웹 에이전트가 설치된 디렉터리를 나타냅니다.

POST 보존 템플릿 파일을 사용하도록 웹 에이전트를 구성하려면 PostPreservationFile 에이전트 구성 매개 변수를 정의하여 템플릿 파일 경로를 지정합니다.
예를 들면 다음과 같습니다.

```
PostPreservationFile="/app/netegrity/webagent/samples_default/forms/nosubmitbutto
n.pptemplate"
```

POST 보존 사용 안 함

POST 보존을 사용할 필요가 없는 경우 다음 매개 변수로 이를 사용하지 않도록 설정할 수 있습니다.

PreservePostData

웹 에이전트가 요청을 리디렉션할 때 POST 데이터를 유지할지 여부를 지정합니다. 사용자가 양식 또는 인증서 인증과 같은 고급 인증을 요청받을 경우 인증 단계 중에 POST 데이터가 유지됩니다.

기본값: Yes

POST 보존이 사용되지 않도록 설정하려면 PreservePostData 매개 변수의 값을 no 로 설정하십시오.

고급 FCC 설정 구성

다음과 같은 고급 자격 증명 수집기 설정을 필요에 따라 구성할 수 있습니다.

- 소문자를 사용하여 URL 의 프로토콜 부분 지정 (페이지 190)
- 리디렉션 URL 의 쿼리 문자열 암호화 (페이지 191)
- 리디렉션 URL 의 쿼리 문자열을 인코딩하기 위한 FCC 지시문 (페이지 192)
- Windows 인증을 허용하도록 FCC 를 구성합니다. (페이지 195)
- FCC 와 함께 ARR 사용 (페이지 193)

소문자를 사용하여 리디렉션 URL 프로토콜 지정

양식 기반 인증 체계를 사용하여 RFC 2396 을 준수하지 않는 레거시 응용 프로그램을 보호하고 URL 의 프로토콜 부분을 소문자로 지정해야 할 경우에는 다음 매개 변수를 설정합니다.

LowerCaseProtocolSpecifier

리디렉션 URL 의 스키마(프로토콜) 부분에 소문자만 사용할지 여부를 지정합니다. 이 구성 매개 변수는 RFC 2396 을 준수하지 않는 레거시 응용 프로그램을 사용할 수 있도록 조정합니다. 이 RFC 는 응용 프로그램이 URL 의 프로토콜 부분을 대문자와 소문자로 둘 다 처리해야 함을 명시합니다. 다음과 같은 경우 이 매개 변수를 변경하십시오.

- RFC 2396 을 준수하지 않는 레거시 응용 프로그램을 사용합니다.
- 쿼리 데이터가 포함된 URL 을 리디렉션합니다.
- HTML 양식(FCC) 인증 스키마를 사용합니다.

기본값: No(HTTP, HTTPS 와 같이 대문자 사용)

예: Yes(http, https 와 같이 소문자 사용)

환경에서 URL 의 프로토콜 부분을 소문자로 지정하려면 LowerCaseProtocolSpecifier 매개 변수의 값을 yes 로 설정하십시오.

리디렉션 URL 의 쿼리 문자열 매개 변수 암호화

다음 매개 변수를 사용하면 웹 에이전트에서 리디렉션 URL 의 모든 SiteMinder 쿼리 매개 변수를 암호화할 수 있습니다.

SecureURLs

웹 에이전트가 리디렉션 URL 의 SiteMinder 쿼리 매개 변수를 암호화할지 여부를 지정합니다. 이 설정을 사용하면 고급 인증 체계에서 보호되는 요청된 리소스와 암호 서비스를 추가적으로 보호하거나 요청에서 쿠키 공급자를 호출하는 경우 보안을 강화할 수 있습니다.

중요! 웹 에이전트는 SiteMinder 구성 요소 간에 전송되는 데이터만 암호화합니다. SiteMinder 이외의 응용 프로그램에 대한 리디렉션에서 전송되는 데이터는 암호화되지 않습니다.

다음 SiteMinder 자격 증명 수집기와 응용 프로그램은 SecureUrls 기능을 지원합니다.

- HTML 양식 인증
- 인증서 및 양식 인증
- SSL 인증
- 인증서 또는 양식 인증
- NTLM 인증
- ACE 인증
- SafeWord 인증
- 사용자 자체 등록
- 쿠키 공급자를 사용한 여러 도메인 SSO(싱글 사인온)
- FCC 기반 암호 서비스(CGI 또는 JSP 기반이 아님)

기본값: No

다음 단계를 수행하십시오.

1. SecureURLs 매개 변수의 값을 yes 로 설정합니다.
2. 싱글 사인온 환경 내에서 리디렉션 URL 의 쿼리 문자열 매개 변수를 암호화하려면 싱글 사인온 환경에 있는 모든 웹 에이전트의 SecureURL 매개 변수가 같은 값으로 설정되어야 합니다.
3. 사용자 지정 FCC 를 사용하는 경우에는 smquerydata 지시문을 다른 FCC 지시문(예: TARGET)과 함께 사용자 지정 FCC 에 추가합니다.

   SiteMinder 리디렉션 URL 의 쿼리 문자열 매개 변수가 암호화됩니다.

리디렉션 URL 의 쿼리 문자열을 인코딩하기 위한 FCC 지시문

자격 증명 수집기에 대한 리디렉션 URL 의 쿼리 문자열을 암호화할 수 있습니다. 자격 증명 수집기는 쿼리 데이터를 암호화하는 데 사용되는 키를 제공합니다.

양식 인증 체계의 경우 쿼리 문자열 지시문 smquerydata 는 FCC 템플릿의 일부입니다. FCC 를 제공하는 에이전트는 FCC 가 포스트되면 이 지시문을 사용하여 암호화된 쿼리 데이터를 대상 에이전트에 보냅니다.

다음과 같은 지시문이 사용됩니다.

```
<INPUT type='hidden' name='smquerydata' value='$$smquerydata$$>
```

참고: 사용자 지정 FCC 를 사용하는 경우에는 smquerydata 지시문을 다른 FCC 지시문(예: TARGET)과 함께 사용자 지정 FCC 에 추가하십시오.

SecureUrls 매개 변수가 사용되도록 설정된 SiteMinder 12.52 SP1 에이전트는 이 기능을 지원하는 다른 에이전트에서 제공된 자격 증명 수집기와의 작업만 가능합니다.

HTML 양식 인증을 위해 ARR 를 구성하는 방법

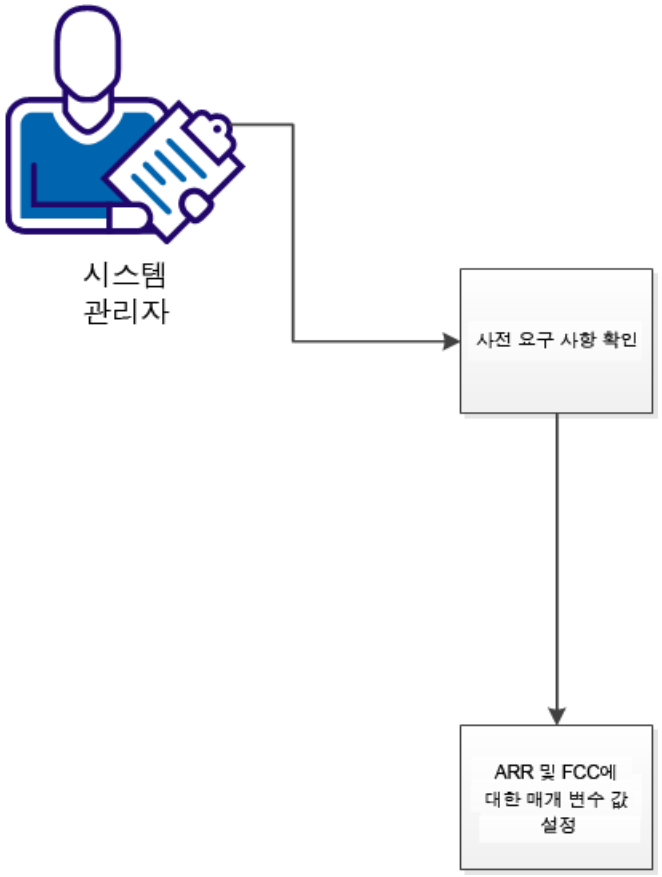
ARR(응용 프로그램 요청 라우팅)는 Microsoft IIS(인터넷 정보 서비스)에 사용할 수 있는 선택적 기능입니다. ARR 는 프록시 서버와 같은 다른 서버에 요청을 전달합니다.

IIS 웹 서버는 ARR 를 사용하여 쿠키를 다르게 처리합니다. 이 구성은 FCC 인증 체계에서 SiteMinder 쿠키가 처리되는 방식에 영향을 줍니다.

이 시나리오에서는 다음과 같은 경우 <stmdnr> 에이전트에 필요한 추가 구성 설정을 보여 줍니다.

- ARR 가 FCC 와 함께 사용됩니다.
- ARR 가 SiteMinder 및 Arcot 와 함께 사용됩니다.

다음 그림에서는 시스템 관리자가 FCC 및 ARR 에 대해 SiteMinder 를 구성하는 방법을 보여 줍니다.

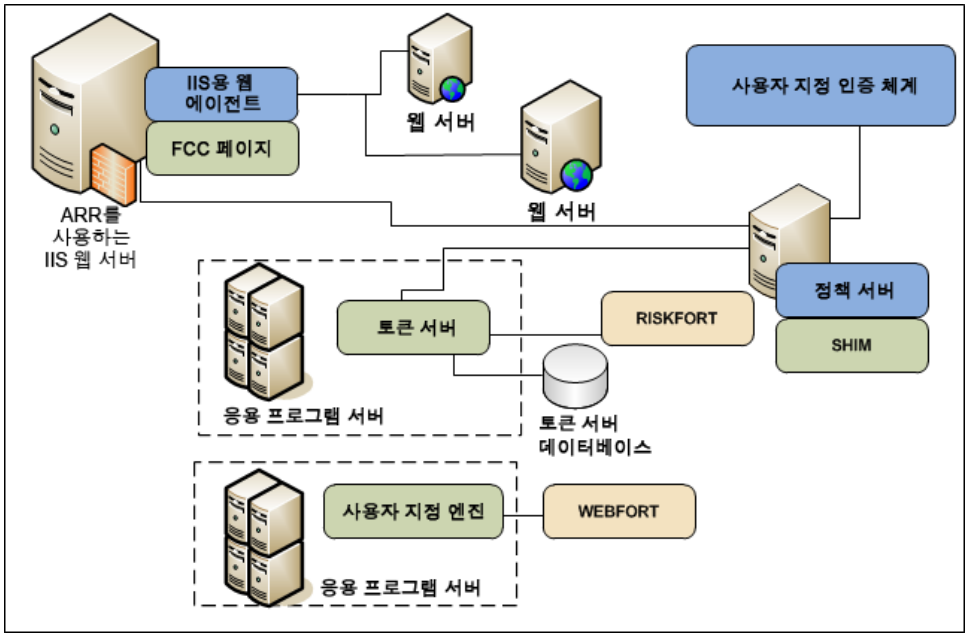


ARR 및 FCC 를 함께 사용하도록 SiteMinder 에이전트를 구성하려면 다음 단계를 수행하십시오.

1. 환경이 사전 요구 사항을 충족하는지 확인합니다 (페이지 194).
2. ARR 및 FCC 에 대한 매개 변수 값을 설정합니다 (페이지 195).

사전 요구 사항 확인

다음 그림에서는 환경의 구성 요소 및 사전 요구 사항을 보여 줍니다.



SiteMinder 및 [assign the value for dlp in your book] 환경이 다음 요구 사항을 충족하는지 확인합니다.

- 전체 SiteMinder 환경이 설치 및 구성되어 다음 구성 요소를 포함합니다.
  - ARR 를 실행하는 IIS 웹 서버 뒤에 배포된 웹 서버의 리소스를 보호하는 정책
  - (선택 사항) 설치 및 구성된 CA Arcot 구성 요소
- 다음 항목을 포함하는 IIS 웹 서버가 설치 및 구성되어 있습니다.
  - 웹 서버에 요청을 전달하도록 구성된 ARR(응용 프로그램 요청 라우팅)
  - ARR 를 실행하는 서버에 설치 및 구성된 IIS 용 SiteMinder 에이전트
  - FCC 인증 체계
- 에이전트에서 중앙 구성을 사용하는지 아니면 로컬 구성을 사용하는지 확인합니다.

ARR 및 FCC 에 대한 매개 변수 값 설정

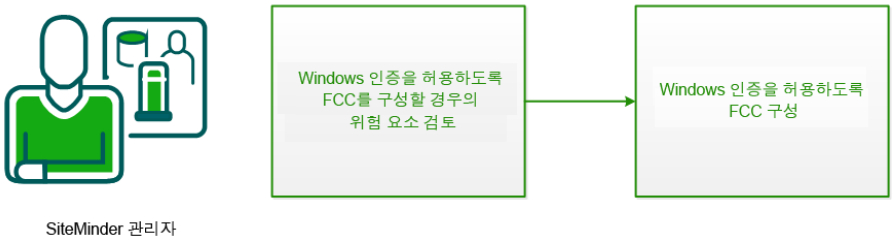
ARR 및 FCC 에 대한 매개 변수 값을 설정하려면 다음 단계를 수행하십시오.

1. 다음 목록에서 에이전트 구성 방법에 해당하는 태스크를 수행합니다.
   - 중앙 구성의 경우 에이전트 구성 개체를 엽니다 (페이지 34).
   - 로컬 구성의 경우 웹 서버에서 로컬 구성 파일을 엽니다 (페이지 39).
2. FCCCompatMode 매개 변수를 찾은 후 값을 yes 로 변경합니다.
3. CookieDomain 매개 변수를 찾은 후 값을 none 으로 변경합니다(값을 비워 두는 것이 *아님*).

Windows 인증을 허용하도록 FCC 를 구성하는 방법

SiteMinder FCC(양식 자격 증명 수집기)는 CA Services 에서 사용자 지정 인증 체계를 안전하게 트리거할 수 있도록 되어 있습니다. 따라서 FCC 는 모든 인증 체계에 대해 사용자를 인증할 수 있습니다. 그러나 FCC 는 기본적으로 Windows 인증 체계에 대해서는 인증하지 *않습니다*. 이 덕분에 특정 구성에서 공격자가 FCC 를 이용하여 유효한 Windows 사용자에 대해 SiteMinder 세션을 생성하는 문제가 발생하지 않습니다.

운영 환경에서 FCC 를 사용하여 Windows 인증 체계에 대해 인증해야 하는 경우 EnableFCCWindowsAuth 에이전트 구성 매개 변수를 지정하면 됩니다. 그러나 FCC 에서 Windows 인증을 지원하도록 설정하기 전에 이로 인한 위험 요소를 검토하고 보안에 취약한 구성에 대해 알아야 합니다.



1. Windows 인증을 허용하도록 FCC 를 구성할 경우의 위험 요소를 검토합니다 (페이지 196).
2. Windows 인증을 허용하도록 FCC 를 구성합니다. (페이지 197)

Windows 인증을 허용하도록 FCC 를 구성할 경우의 위험 요소

기본적으로 FCC 는 Windows 인증 체계에 대해 인증하지 않습니다. FCC 에서 Windows 인증을 허용하도록 설정할 수는 있습니다. 그러나 이렇게 하면 특정 구성에서 공격자가 FCC 를 사용하여 유효한 Windows 사용자에 대해 SiteMinder 세션을 생성하는 문제가 발생할 수 있습니다.

HTML 양식으로 보호된 영역과 Windows 로 보호된 영역 둘 다에 같은 SiteMinder 에이전트 이름 또는 에이전트 그룹 이름이 사용되는 구성의 경우 이런 위험에 노출됩니다. 예를 들면 단일 웹 에이전트가 HTML 양식 및 Windows 인증을 사용하여 구성된 여러 영역을 보호하도록 설정된 경우입니다.

다음 예제 시나리오를 검토하십시오.

- Resource A 는 HTML 양식 인증을 사용하여 보호된 영역에 구성되어 있습니다. FCC 는 Resource A 에 액세스하는 사용자에게 HTML 양식을 사용하여 인증을 요청합니다.
- Resource B 는 Windows 인증을 사용하여 보호된 영역에 구성되어 있습니다. Resource B 에 액세스하는 사용자는 Windows 인증을 완료합니다.
- 두 리소스 모두 같은 IIS 서버에서 호스트되고 같은 웹 에이전트에 의해 보호됩니다. 따라서 두 영역 모두 같은 에이전트 이름으로 구성됩니다.

다음과 같이 공격이 발생합니다.

1. 공격자가 HTML 양식의 TARGET 매개 변수를 "Resource A"에서 "Resource B"로 수정합니다.
2. 공격자가 유효한 Windows 사용자 이름으로 양식을 제출합니다.
3. FCC 가 인증을 위해 사용자 이름을 정책 서버에 전달합니다. SiteMinder 에서 HTML 양식 인증 체계 대신 Windows 인증 체계를 실행하고 사용자 이름 유효성이 확인됩니다.

그 결과, SiteMinder 세션이 사용자에게 반환되어 새 세션이 유효하다고 간주되는 모든 후속 요청에 대해 싱글 사인온을 설정합니다. 이제 공격자는 Windows 사용자 이름이 FCC 에 제출된 사용자를 가장합니다.

Windows 인증을 허용하도록 FCC 구성

Windows 인증을 허용하도록 FCC 를 구성하려면 다음 에이전트 구성 매개 변수를 지정합니다.

EnableFCCWindowsAuth

FCC 로 사용되는 에이전트가 SiteMinder Windows 인증 체계에서 보호하는 리소스에 대해 사용자를 인증할 수 있는지 여부를 지정합니다.

이 매개 변수는 다음과 같은 값을 사용합니다.

- Yes - FCC 에서 Windows 인증 체계에 대해 인증할 수 있습니다.

  중요! 이 매개 변수를 Yes 로 설정하면 공격자가 필수 자격 증명을 제공하지 않고 FCC 를 이용하여 Windows 사용자를 가장할 수 있습니다.

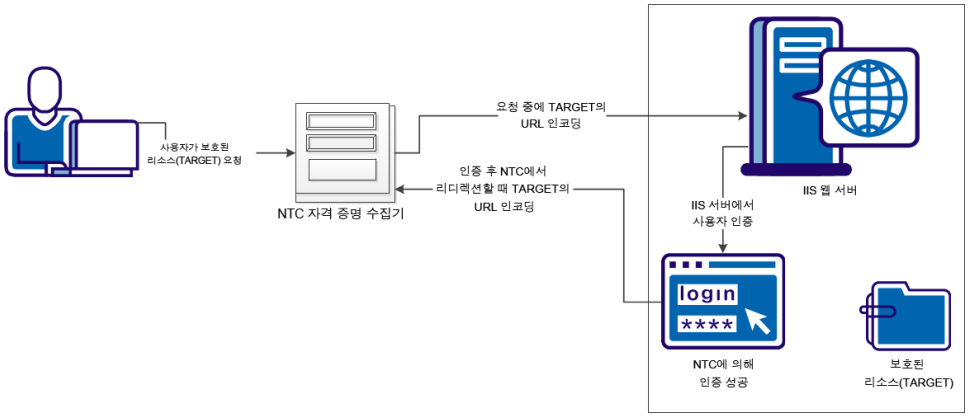
- No - FCC 에서 Windows 인증 체계에 대해 인증할 수 없습니다.

기본값: No

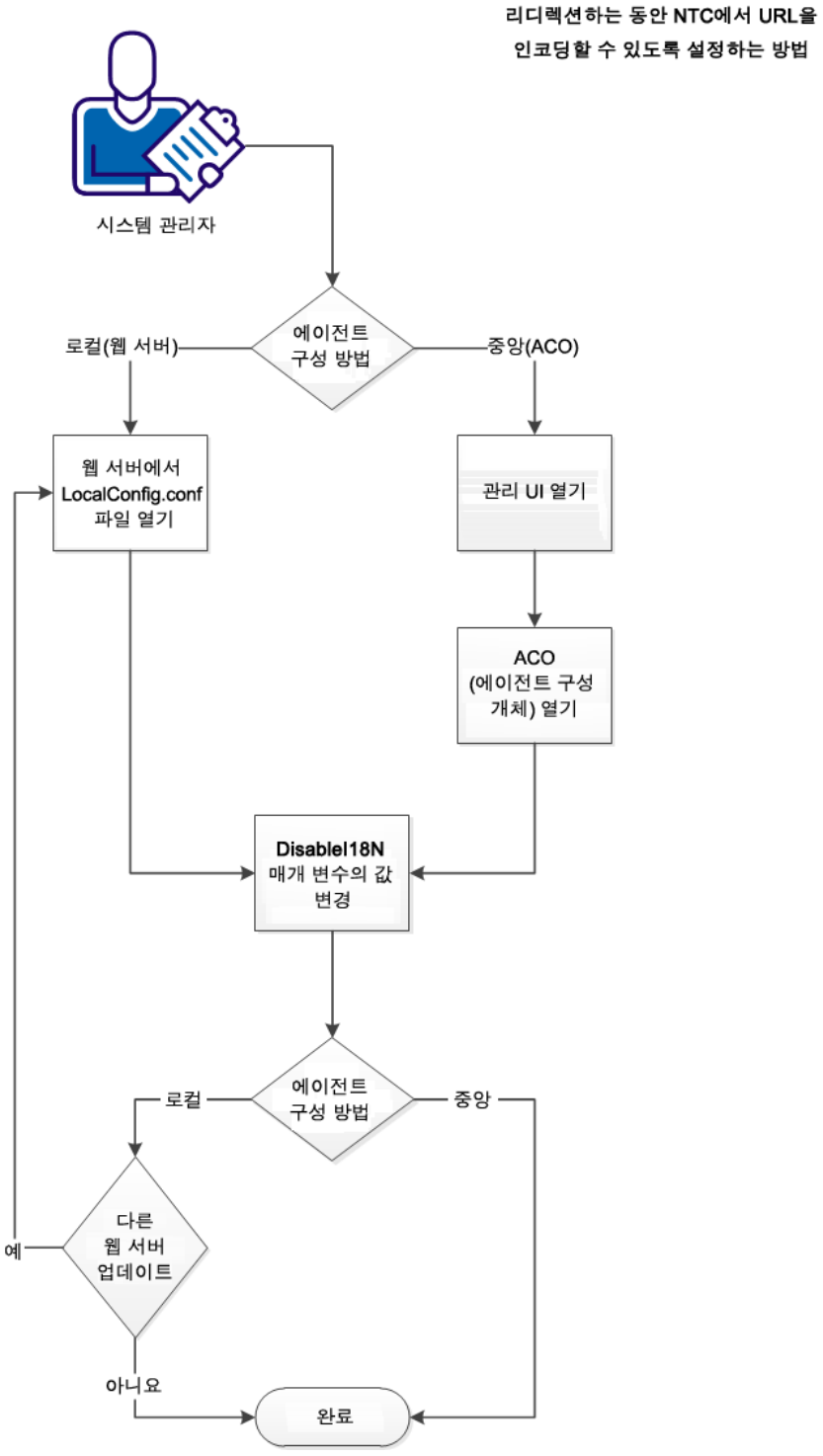
보호된 리소스로 리디렉션하는 동안 NTC 에서 URL 을 인코딩할 수 있도록 설정하는 방법

SiteMinder 는 NTC(Windows 자격 증명 수집기)를 사용하여 리소스를 보호할 수 있습니다. 사용자가 NTC 에 자격 증명을 제출하면 NTC 에서 IIS 웹 서버에 사용자가 로그인하도록 합니다. 그런 다음 IIS 웹 서버에서 사용자를 인증합니다. 인증 후 NTC 는 사용자를 보호된 리소스(TARGET)로 리디렉션합니다.

일반적으로 NTC 는 요청 중에 URL 의 TARGET 부분에 있는 문자를 인코딩하지만 인증 후 리디렉션 중에는 인코딩하지 않습니다. 리디렉션하는 동안 URL 의 TARGET 부분이 인코딩되도록 에이전트 구성을 변경할 수 있습니다. 다음 그림에서는 이러한 동작을 보여 줍니다.



다음 그림에서는 보호된 리소스에 대한 요청을 처리하는 동안 NTC 에서 URL 을 인코딩할 수 있도록 설정하는 과정을 보여 줍니다.



보호된 리소스로 리디렉션하는 동안 NTC 에서 URL 을 인코딩할 수 있도록 설정하려면 다음 단계를 수행하십시오.

1. 다음 목록에서 에이전트 구성 방법에 해당하는 절차를 선택합니다.
   - 정책 서버의 ACO(에이전트 구성 개체)를 사용하는 에이전트의 경우 다음 단계를 수행하십시오.
     a. 관리 UI 를 엽니다 (페이지 200).
     b. ACO(에이전트 구성 개체)를 열고 DisableI18N 매개 변수의 값을 변경합니다 (페이지 201).
   - 웹 서버의 로컬 구성 파일을 사용하는 에이전트의 경우 다음 단계를 수행하십시오.
     a. 텍스트 편집기에서 웹 서버의 LocalConfig.conf 파일을 열고 DisableI18N 매개 변수의 값을 변경합니다 (페이지 204).
2. 로컬 구성을 사용하는 에이전트의 경우 각 웹 서버에 대해 1c 단계를 반복합니다.

   NTC 는 보호된 리소스로 리디렉션하는 동안 인코딩된 URL 을 사용합니다.

정책 서버 개체 변경

관리 UI 를 열어 정책 서버의 개체를 변경합니다.

다음 단계를 수행하십시오.

1. 브라우저에서 다음 URL 을 엽니다.

   `https://host_name:8443/iam/siteminder/adminui`

   host_name

   정규화된 관리 UI 호스트 시스템 이름을 지정합니다.
2. "사용자 이름" 필드에 SiteMinder 슈퍼 사용자 이름을 입력합니다.
3. "암호" 필드에 SiteMinder 슈퍼 사용자 계정 암호를 입력합니다.

   참고: 슈퍼 사용자 계정 암호에 달러 -기호($) 문자가 있으면 달러 기호 문자의 각 인스턴스를 $DOLLAR$로 바꾸십시오. 예를 들어 SiteMinder 슈퍼 사용자 계정 암호가 $password 이면 "암호" 필드에 $DOLLAR$password 를 입력하십시오.
4. "서버" 드롭다운 목록에 올바른 서버 이름 또는 IP 주소가 나타나는지 확인합니다.
5. "로그인"을 선택합니다.

에이전트 구성 개체에서 DisableI18N 매개 변수의 값 변경

중앙에서 구성된 웹 에이전트에 대해 대상 URL 의 HTTP 인코딩된 문자를 처리하도록 Windows 자격 증명 수집기를 구성할 수 있습니다. 중앙에서 구성된 웹 에이전트는 정책 서버의 에이전트 구성 개체에 저장된 매개 변수 설정을 사용합니다.

다음 단계를 수행하십시오.

1. "인프라", "에이전트 구성 개체"를 차례로 클릭합니다.

   에이전트 구성 개체 목록이 나타납니다.

   원하는 에이전트 구성 개체 행에서 편집 아이콘을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 나타납니다.

2. 다음 매개 변수의 왼쪽에 있는 편집 아이콘을 클릭합니다.

   DisableI18N

   TARGET URL 의 문자에 HTTP 인코딩이 사용되는 경우 인증하는 동안 NTC(Windows 자격 증명 수집기)에서 TARGET URL 을 처리하는 방식을 지정합니다. 이 매개 변수의 값이 *no* 이면 인증하는 동안 URL 의 문자가 디코딩됩니다. 디코딩된 문자는 TARGET 리소스로 리디렉션할 때 사용됩니다. 이 매개 변수의 값이 *yes* 인 경우에는 인증하는 동안 TARGET URL 의 문자가 디코딩되지 않습니다. HTTP 인코딩을 사용하는 문자는 인증 전후에 계속 인코딩 상태를 유지합니다.

   기본값: No

   "매개 변수 편집" 대화 상자가 나타납니다.

3. "값" 필드의 텍스트를 "yes"로 변경합니다.

4. "확인"을 클릭합니다.

   "매개 변수 편집" 대화 상자가 닫히고 "에이전트 구성 수정" 대화 상자가 나타납니다.

5. 다음 매개 변수의 왼쪽에 있는 편집 아이콘을 클릭합니다.

BadUrlChars

URL 요청에 사용할 수 없는 문자 시퀀스를 지정합니다. 웹 에이전트는 URL 에서 "?" 앞에 나오는 문자를 이 매개 변수의 목록과 비교하여 검사합니다. 지정된 문자가 발견되면 웹 에이전트에서 요청을 거부합니다.

다음과 같은 문자를 지정할 수 있습니다.

- 백슬래시(\)
- 두 개의 슬래시(//)
- 마침표와 슬래시(./)
- 슬래시와 마침표(/.)
- 슬래시와 별표(/*)
- 별표와 마침표(*.)
- 물결표(~)
- %2d
- %20
- %00-%1f
- %7f-%ff
- %25

문자가 여러 개일 경우에는 쉼표로 구분하십시오. 공백을 사용하면 *안 됩니다*.

물음표(?)가 잘못된 URL 문자 앞에 오면 CGI 매개 변수에 잘못된 URL 문자를 사용할 수 있습니다.

기본값: 비활성화됨(모든 문자가 허용됨).

제한:

- 영어 문자에는 기본 16 진수 숫자가 적용됩니다. 다른 언어의 경우 허용하려는 언어의 문자에 해당하는 16 진수 값을 제거하십시오. 이러한 언어의 예로는 포르투갈어(브라질), 프랑스어, 일본어, 중국어 등이 포함됩니다.
- 리터럴 문자를 지정할 수 있습니다. 또한 해당 문자를 URL 로 인코딩된 형식으로 입력할 수도 있습니다. 예를 들어 a 라는 문자를 그대로 입력하거나, 이 문자를 인코딩한 값인 %61 을 입력할 수 있습니다.
- 문자를 구분하는 데 사용되는 쉼표를 포함하여 최대 4096 자의 문자를 지정할 수 있습니다.
- 하이픈으로 구분된 문자 범위를 지정할 수 있습니다. 이때 사용되는 구문은 *starting_character-ending_character* 입니다. 예를 들어 a-z 를 문자 범위로 입력할 수 있습니다.
- 따옴표(")는 URL 인코딩 값 %22 를 사용하여 지정하십시오. ASCII 를 사용하면 *안 됩니다.*

"매개 변수 편집" 대화 상자가 나타납니다.

6. "값" 필드에서 다음 텍스트를 제거합니다.

   ,%25

7. "확인"을 클릭합니다.

   "매개 변수 편집" 대화 상자가 닫히고 "에이전트 구성 수정" 대화 상자가 나타납니다.

8. "제출"을 클릭합니다.

   "에이전트 구성 수정" 대화 상자가 닫히고 확인 메시지가 표시됩니다.

9. (선택 사항) 나중에 참조할 수 있도록 "설명" 필드에 변경 내용에 대해 간단히 입력합니다.

10. "예"를 클릭합니다.

    다음에 웹 에이전트가 정책 서버를 폴링하면 변경 내용이 적용됩니다.

LocalConfig.conf 파일에서 DisableI18N 매개 변수의 값 변경

대상 URL 의 HTTP 인코딩된 문자를 처리하도록 Windows 자격 증명 수집기를 구성할 수 있습니다. 로컬로 구성된 웹 에이전트는 각 웹 서버의 구성 파일에 저장된 매개 변수 설정을 사용합니다.

다음 단계를 수행하십시오.

웹 서버에서 LocalConfig.conf 파일을 찾습니다. 다음 목록의 예제를 사용하면 원하는 유형의 웹 서버에서 파일을 찾을 수 있습니다.

IIS 웹 서버

web_agent_home\bin\IIS

Oracle iPlanet 웹 서버

Oracle_iPlanet_home/https-hostname/config

Apache 웹 서버

Apache_home/conf

1. 텍스트 편집기에서 LocalConfig.conf 파일을 열고 다음 매개 변수를 찾습니다.

   DisableI18N

   TARGET URL 의 문자에 HTTP 인코딩이 사용되는 경우 인증하는 동안 NTC(Windows 자격 증명 수집기)에서 TARGET URL 을 처리하는 방식을 지정합니다. 이 매개 변수의 값이 *no* 이면 인증하는 동안 URL 의 문자가 디코딩됩니다. 디코딩된 문자는 TARGET 리소스로 리디렉션할 때 사용됩니다. 이 매개 변수의 값이 *yes* 인 경우에는 인증하는 동안 TARGET URL 의 문자가 디코딩되지 않습니다. HTTP 인코딩을 사용하는 문자는 인증 전후에 계속 인코딩 상태를 유지합니다.

   기본값: No

2. DisableI18n 매개 변수의 값을 yes 로 변경합니다.

3. 다음 매개 변수를 찾습니다.

BadUrlChars

URL 요청에 사용할 수 없는 문자 시퀀스를 지정합니다. 웹 에이전트는 URL 에서 "?" 앞에 나오는 문자를 이 매개 변수의 목록과 비교하여 검사합니다. 지정된 문자가 발견되면 웹 에이전트에서 요청을 거부합니다.

다음과 같은 문자를 지정할 수 있습니다.

- 백슬래시(\)
- 두 개의 슬래시(//)
- 마침표와 슬래시(./)
- 슬래시와 마침표(/.)
- 슬래시와 별표(/*)
- 별표와 마침표(*.)
- 물결표(~)
- %2d
- %20
- %00-%1f
- %7f-%ff
- %25

문자가 여러 개일 경우에는 쉼표로 구분하십시오. 공백을 사용하면 *안 됩니다*.

물음표(?)가 잘못된 URL 문자 앞에 오면 CGI 매개 변수에 잘못된 URL 문자를 사용할 수 있습니다.

기본값: 비활성화됨(모든 문자가 허용됨).

제한:

- 영어 문자에는 기본 16 진수 숫자가 적용됩니다. 다른 언어의 경우 허용하려는 언어의 문자에 해당하는 16 진수 값을 제거하십시오. 이러한 언어의 예로는 포르투갈어(브라질), 프랑스어, 일본어, 중국어 등이 포함됩니다.
- 리터럴 문자를 지정할 수 있습니다. 또한 해당 문자를 URL 로 인코딩된 형식으로 입력할 수도 있습니다. 예를 들어 a 라는 문자를 그대로 입력하거나, 이 문자를 인코딩한 값인 %61 을 입력할 수 있습니다.
- 문자를 구분하는 데 사용되는 쉼표를 포함하여 최대 4096 자의 문자를 지정할 수 있습니다.
- 하이픈으로 구분된 문자 범위를 지정할 수 있습니다. 이때 사용되는 구문은 *starting_character-ending_character* 입니다. 예를 들어 a-z 를 문자 범위로 입력할 수 있습니다.

따옴표(")는 URL 인코딩 값 %22 를 사용하여 지정하십시오. ASCII 를 사용하면 *안 됩니다*.

4. BadURLChars 목록에서 다음 값을 제거합니다.

   ,%25

5. LocalConfig.conf 파일의 변경 내용을 저장하고 텍스트 편집기를 닫습니다.
6. 변경할 모든 웹 서버에 대해 1-5 단계를 반복합니다.

   Windows 자격 증명 수집기에서 TARGET URL 의 HTTP 인코딩된 문자를 처리할 수 있습니다.

FCC 성능 조정

다음과 같은 설정을 구성하여 자격 증명 수집기 성능을 향상시킬 수 있습니다.

- 성능 향상을 위해 FCC 영역 컨텍스트 확인 사용 안 함 (페이지 207)
- 양식 캐시 사용 (페이지 207)

성능 향상을 위해 FCC 영역 컨텍스트 확인 사용 안 함

양식 인증을 진행하는 동안 웹 에이전트는 정책 서버에 대해 IsProtected 호출을 실행하여 요청된 리소스가 보호되는지 여부를 확인합니다. 일반적으로 첫 번째 호출이 완료된 후 웹 에이전트는 정책 서버에 대해 두 번째 IsProtected 호출을 실행합니다. 두 번째 호출을 실행하면 웹 에이전트에서 FCC 를 통해 사용자가 로그인하여 보호된 리소스에 액세스할 수 있도록 영역 컨텍스트가 설정됩니다. 다음 매개 변수를 사용하면 웹 에이전트에서 두 번째 호출을 실행할지 여부를 제어할 수 있습니다.

FCCForceIsProtected

사용자가 로그인하여 보호된 리소스에 액세스할 수 있도록 웹 에이전트가 영역 컨텍스트를 설정하기 위한 추가 IsProtected 호출을 정책 서버에 대해 실행할지 여부를 지정합니다.

이 매개 변수를 no 로 설정하면 웹 에이전트는 추가 호출 대신 정책 서버에 대한 초기 IsProtected 호출에서 얻은 영역 정보를 사용합니다.

기본값: Yes

성능 향상을 위해 FCC 영역 컨텍스트 확인이 사용되지 않도록 설정하려면 FCCForceIsProtected 매개 변수의 값을 no 로 설정하십시오.

양식 캐시

양식 캐시는 양식 템플릿 데이터를 저장합니다. 템플릿 데이터를 저장하면 같은 데이터에 대해 에이전트가 .fcc 파일을 여러 번 읽을 필요가 없으므로 성능이 향상됩니다. FCC 확장명을 사용하는 리소스에 액세스하는 경우 FCC 는 해당하는 템플릿 파일을 읽고 처리합니다. 에이전트는 이러한 읽기 작업을 초당 수백 개씩 수행합니다.

양식 캐시는 양식 템플릿 파일을 쉽게 읽을 수 있도록 메모리에 저장하여 FCC 의 부담을 줄여 줍니다. 가상 메모리 액세스는 디스크 액세스보다 빠르므로 호스트 서버 사용을 줄이고 FCC 구성 요소에서 양식을 더 빠르게 처리할 수 있도록 합니다.

처리 시간이 빨라지면 각 웹 서버에 대한 요청을 처리하는 FCC 의 용량이 늘어납니다. 또한 양식 인증의 효율성이 향상됩니다.

양식 캐시 데이터

양식 캐시에 저장되는 데이터는 사전에 데이터 구조로 구문 분석된 양식 템플릿 텍스트로 구성됩니다. 이러한 데이터 구조는 FCC 처리를 최적화합니다.

이 데이터 구조에는 다음 항목이 포함됩니다.

- 국제화를 위한 양식 로캘 데이터
- UTF-8 형식의 원시 텍스트, 템플릿 지시문 정보, 요청 환경에서 대체할 함수/변수 정보 등을 포함하는 데이터 개체의 정렬된 목록

지시문, 함수 및 변수는 FCC 파일의 위에서부터 순서대로 처리됩니다.

양식 캐시 구성

양식을 캐시하여 성능을 향상시키고 필요 없는 네트워크 트래픽을 줄일 수 있습니다. 다음 매개 변수를 사용하면 양식 캐시 설정을 제어할 수 있습니다.

EnableFormCache

양식 템플릿 캐시를 제어합니다. 이 매개 변수를 yes 로 설정하면 양식 인증 성능이 향상됩니다. 캐시를 해제하려면 이 매개 변수를 no 로 설정하십시오.

기본값: Yes

FormCacheTimeout

개체가 유효하지 않은 것으로 간주되기 전에 캐시에 유지될 수 있는 시간(초)을 지정합니다. 시간 만료 간격이 만료되면 양식 템플릿 파일의 날짜와 시간이 캐시 개체가 생성된 시간과 비교됩니다. 캐시의 개체가 디스크의 파일보다 더 최근에 저장된 경우 시간 만료가 다른 간격으로 다시 설정됩니다. 그렇지 않으면 개체가 캐시에서 제거됩니다.

기본값: 600

다음 단계를 수행하십시오.

1. EnableFormCache 매개 변수의 값을 yes 로 설정합니다.
2. 양식 캐시의 시간 제한 간격을 변경하려면 FormCacheTimeout 매개 변수를 원하는 값(초)으로 설정합니다.

양식 캐시가 구성됩니다.

NTC(NTLM Credential Collector) 지정

NTC(NTLM credential collector)는 웹 에이전트 내의 응용 프로그램입니다. NTC 는 Windows 인증 체계로 보호하는 리소스에 대한 NT 자격 증명을 수집합니다. 이 체계는 IIS 웹 서버에서 Internet Explorer 브라우저로 액세스하는 리소스에 적용됩니다.

각 자격 증명 수집기에는 MIME 유형이 연결됩니다. IIS 의 경우 다음 매개 변수에 NTC MIME 유형이 정의됩니다.

NTCExt

NTC(NTLM credential collector)와 연결되는 MIME 유형을 지정합니다. 이 수집기는 Windows 인증 체계에서 보호하는 리소스에 대한 NT 자격 증명을 수집합니다. 이 인증 체계는 Internet Explorer 브라우저 사용자만 액세스하는 IIS 웹 서버의 리소스에 적용됩니다.

이 매개 변수에 여러 개의 확장명을 사용할 수 있습니다. 에이전트 구성 개체를 사용하는 경우 다중값 옵션을 선택하십시오. 로컬 구성 파일을 사용하는 경우 각 확장명을 쉼표로 구분하십시오.

기본값: .ntc

NTCExt 매개 변수에 지정되는 기본 확장명이 환경에 이미 사용되는 경우 다른 MIME 유형을 지정할 수 있습니다.

자격 증명 수집기를 트리거하는 확장명을 변경하려면 NTCExt 매개 변수에 다른 파일 확장명을 추가합니다.

4.x 유형 에이전트와 최신 유형 에이전트 간에 자격 증명 수집기 사용

이전 버전의 SiteMinder 에이전트 개체는 공유 암호 기능을 특징으로 하는 보안 모델을 사용했습니다. 공유 암호는 정책 서버 및 WebAgent.conf 파일에 저장됩니다. 이러한 에이전트를 4.x 유형 에이전트라고 합니다. SiteMinder 관리 UI 에서 에이전트 개체를 생성할 때 4.x 에이전트 기능에 대한 지원을 지정할 수 있습니다.

이후 버전의 SiteMinder 는 공유 암호 보안 모델 대신 정책 서버의 트러스트된 호스트 개체를 사용합니다.

SiteMinder 는 4.x 유형 에이전트와 이후 에이전트 간의 자격 증명 수집기 사용을 지원합니다. 이런 식의 자격 증명 수집기 사용을 혼합 모드라고 합니다. 혼합 모드 배포에는 추가 구성 단계가 필요합니다.

혼합 환경에서 자격 증명 수집기 구성

SiteMinder r6.x~SiteMinder 12.52 SP1 버전의 자격 증명 수집기는 4.x 유형의 자격 증명 수집기와 작동 방식이 다릅니다. 4.x 유형의 자격 증명 수집기는 쿠키를 사용자의 브라우저에 두고 사용자를 원래 에이전트로 다시 리디렉션했습니다.

최신 버전 SiteMinder 의 경우 자격 증명 수집기는 요청된 리소스를 보호하는 에이전트 대신 정책 서버에 사용자가 로그인하도록 합니다. 쿠키는 사용되지 *않습니다*.

참고: 자격 증명 수집기를 사용하여 사용자가 직접 로그인하는 것이 쿠키를 설정하는 것보다 좋습니다. 자격 증명 수집기를 사용하여 사용자가 로그인하면 사용자 자격 증명이 쿠키 형태로 네트워크에 전달되지 않으므로 사용자 자격 증명의 보안을 향상시킬 수 있습니다.

사용자가 로그인하려면 자격 증명 수집기에 다음과 같은 정보가 필요합니다.

- 요청된 리소스를 보호하는 에이전트의 이름
- 사용자가 제공한 자격 증명

자격 증명 수집기는 다음 과정을 수행하여 에이전트 이름을 확인합니다.

1. 자격 증명 수집기로 리디렉션할 때 원래 에이전트가 URL 의 쿼리 문자열에 추가하는 SMAGENTNAME 쿼리 매개 변수를 사용합니다.
2. 에이전트 이름이 URL 에 추가되지 않았으면 자격 증명 수집기와 연결되어 있는 AgentName 구성 매개 변수에 정의된 매핑을 사용합니다.

   AgentName 매개 변수의 각 매핑은 보호된 리소스에 대해 해당 수집기를 사용하는 호스트의 이름 및 IP 주소를 지정합니다.
3. 에이전트 이름 매핑이 구성되지 않았으면 대상 URL 의 정규화된 호스트 이름을 에이전트 이름으로 사용합니다. 이 동작은 AgentNamesAreFQHostNames 구성 매개 변수를 사용하여 결정됩니다.

   기본적으로 이 매개 변수는 사용하지 않도록 설정되어 있으므로 자격 증명 수집기는 DefaultAgentName 매개 변수의 값을 에이전트 이름으로 사용합니다.

혼합 환경에서 자격 증명 수집기를 구성하기 전에 이러한 내용을 고려하십시오.

혼합 환경에서 FCC 및 NTC 사용

FCC 와 NTC 는 요청된 리소스를 보호하는 웹 에이전트의 이름 및 사용자 자격 증명을 사용하여 요청을 처리합니다. 그러나 FCC 와 NTC 에 포스트하는 타사 에이전트 및 4.x 에이전트는 URL 을 전송할 때 에이전트 이름을 전달하지 않습니다.

다음 구성 옵션은 FCC 및 NTC 가 4.x 웹 에이전트에서 작동하는 데 도움이 됩니다.

호환성 모드 사용 - r5.x, r6.x 및 12.52 SP1 FCC/NTC 가 4.x 에이전트 또는 타사 응용 프로그램으로 보호되는 리소스에 대한 양식을 제공하도록 설정하려면 FCCCompatMode 매개 변수를 사용합니다. 기존 웹 에이전트의 FCCCompatMode 매개 변수는 기본적으로 사용되도록 설정되어 있으며 프레임워크 에이전트의 FCCCompatMode 매개 변수는 기본적으로 사용되지 않도록 설정되어 있습니다.

이 매개 변수가 사용되도록 설정하면 r5.x, r6.x 또는 12.52 SP1 에이전트가 4.x 에이전트와 같은 방식으로 양식 및 NTLM 자격 증명 수집을 처리할 수 있습니다. NTLM 자격 증명 쿠키를 사용자의 브라우저에 기록하는 이 설정은 로그인이 이루어지기 전에 에이전트로 다시 리디렉션됩니다. 이 구성은 에이전트를 상호 운용할 수 있도록 합니다.

FCCCompatMode 매개 변수의 값이 no 로 설정되면 4.x 에이전트와의 호환성 기능을 사용할 수 없습니다. 12.52 SP1 환경에서는 이 매개 변수의 값을 no 로 설정하십시오.

중요! 이 매개 변수를 no 로 설정하면 4.x 버전의 Netscape 브라우저가 지원되지 않습니다.

- 에이전트 이름 매핑 지정 - FCC 만 해당. 이전 버전과의 호환성을 비활성화한 경우에는 보호된 리소스에 대해 해당 FCC 를 사용하는 각 호스트의 이름 및 IP 주소와 AgentName 매개 변수의 매핑을 지정합니다. FCC 구성 설정에 이러한 매핑을 설정하십시오.

  예:

  myagent, 123.1.12.1

  myagent, www.sitea.com

- 호스트 이름을 에이전트 이름으로 사용 - FCC 만 해당. 처음 두 옵션을 적용할 수 없는 경우에는 AgentNamesAreFQHostNames 매개 변수의 값을 yes 로 설정할 수 있습니다. 이렇게 하면 FCC 에서 대상 URL 의 정규화된 호스트 이름을 에이전트 이름으로 사용할 수 있습니다. 예를 들어 URL 문자열이 다음과 같으면

  url?A=1&Target=http://www.nete.com/index.html

  Target 문자열의 www.nete.com 부분이 에이전트 이름으로 사용됩니다.

  기본적으로 이 매개 변수는 no 로 설정되므로 DefaultAgentName 매개 변수의 값이 에이전트 이름으로 사용됩니다.

다음 표에서는 r5.x, r6.x 또는 12.52 SP1 및 4.x FCC 와 NTC 를 구성하기 위한 지침을 제공하고 혼합 환경에서 각각 어떤 방식으로 작동하는지 설명합니다.

참고:

- NTC(NTLM credential collector)는 IIS 가 아닌 웹 서버에서 IIS 웹 서버로 사용자를 리디렉션할 수 있습니다.
- 프레임워크 웹 에이전트의 경우 FCC 호환성 모드가 사용되지 않는 상황의 지침만 참조하십시오.

리소스를 보호하는 웹 에이전트	**r5.x, r6.x 또는 12.52 SP1 FCC - FCC 호환성 모드 사용**	**r5.x, r6.x 또는 12.52 SP1 FCC - FCC 호환성 모드 사용 안 함**
r5.x, r6.x 또는 12.52 SP1	■ FCC 에서 자격 증명 쿠키를 발급합니다. ■ 인증서 **및** 양식 인증을 사용할 수 없습니다. ■ 인증서 **또는** 양식 인증을 사용할 수 없습니다.	■ FCC 에서 세션 쿠키를 발급합니다. ■ 인증서 **및** 양식 인증이 사용됩니다. ■ 인증서 **또는** 양식 인증이 사용됩니다.

리소스를 보호하는 웹 에이전트	4.x QMR 2/3/4 FCC
4.x QMR 5 또는 4.x QMR 6	■ 에이전트에서 자격 증명 쿠키를 발급합니다. ■ 인증서 및 양식 인증을 사용할 수 없습니다. ■ 인증서 또는 양식 인증이 사용됩니다.
r5.x, r6.x 또는 12.52 SP1	■ 에이전트에서 자격 증명 쿠키를 발급합니다. ■ 인증서 및 양식 인증을 사용할 수 없습니다. ■ 인증서 또는 양식 인증이 사용됩니다.

참고: SSL 인증 체계에 대한 자세한 내용은 정책 서버 설명서를 참조하십시오.

리소스를 보호하는 웹 에이전트	r5.x, r6.x 또는 12.52 SP1 FCC - FCC 호환성 모드 사용	r5.x, r6.x 또는 12.52 SP1 FCC - FCC 호환성 모드 사용 안 함
4.x QMR 5 또는 4.x QMR 6	■ NTC 에서 자격 증명 쿠키를 발급합니다.	■ NTC 에서 세션 쿠키를 발급합니다.
r5.x, r6.x 또는 12.52 SP1	■ NTC 에서 자격 증명 쿠키를 발급합니다.	■ NTC 에서 세션 쿠키를 발급합니다.
리소스를 보호하는 웹 에이전트	**4.x QMR 2/3/4 NTC**	
4.x QMR 5, 4.x QMR 6	■ 에이전트에서 자격 증명 쿠키를 발급합니다.	
r5.x, r6.x 또는 12.52 SP1	■ 에이전트에서 자격 증명 쿠키를 발급합니다.	

혼합 환경에서 SCC 사용

4.x 유형의 웹 에이전트와 r5.x, r6.x 또는 12.52 SP1 SCC 를 상호 운용할 수 있게 하려면 다음 태스크 중 하나를 수행하십시오.

- 에이전트 이름 매핑 지정: 보호된 리소스에 대해 해당 SCC 를 사용하는 각 호스트의 호스트 이름 및 IP 주소와 AgentName 매개 변수의 매핑을 지정합니다. SCC 의 에이전트 구성 매개 변수에 이러한 매핑을 생성하십시오.
- 호스트 이름을 에이전트 이름으로 사용: 에이전트 이름 매핑을 지정하지 않은 경우에는 AgentNamesAreFQHostNames 매개 변수를 Yes 로 설정할 수 있습니다. 이렇게 하면 SCC 에서 대상 URL 의 정규화된 호스트 이름을 에이전트 이름으로 사용할 수 있습니다.

  예를 들어 URL 문자열이 다음과 같으면

  ```
  url?A=1&Target=http://www.nete.com/index.html
  ```

  Target 문자열의 www.nete.com 부분이 에이전트 이름으로 사용됩니다.

  기본적으로 이 매개 변수는 no 로 설정되므로 DefaultAgentName 매개 변수의 값이 에이전트 이름으로 사용됩니다.

다음 표에서는 SCC 역할을 하는 4.x, r5.x, r6.x 또는 12.52 SP1 에이전트가 혼합 환경에서 어떤 방식으로 작동하는지 보여 줍니다.

웹 에이전트 버전	4.x QMR 2/3/4 SCC	r5.x, r6.x 또는 12.52 SP1 SCC
4.x QMR 5 또는 4.x QMR 6	■ 에이전트에서 SSL 자격 증명 쿠키를 발급합니다. ■ 처음에 SSL 을 통해 브라우저에서 서버에 연결한 경우에도 요청을 리디렉션하지 않으면 인증서를 수집할 수 없습니다.	■ AgentName 매개 변수에서 매핑을 생성하거나 AgentNamesAreFQHostNames 를 Yes 로 설정합니다. ■ SCC 에서 세션 쿠키를 발급합니다. ■ 처음에 SSL 을 통해 브라우저에서 서버에 연결한 경우에도 요청을 리디렉션하지 않으면 인증서를 수집할 수 없습니다.

웹 에이전트 버전	4.x QMR 2/3/4 SCC	r5.x, r6.x 또는 12.52 SP1 SCC
r5.x, r6.x 또는 12.52 SP1	■ 에이전트에서 SSL 자격 증명 쿠키를 발급합니다. ■ 요청을 리디렉션하지 않고 인증서를 수집할 수 있습니다.	■ SCC 에서 세션 쿠키를 발급합니다. ■ 요청을 리디렉션하지 않고 인증서를 수집할 수 있습니다.

참고: SSL 인증 체계에 대한 자세한 내용은 정책 서버 설명서를 참조하십시오.

일본어 환경에서 FCC 기반 암호 서비스용 Apache 기반 에이전트 구성

일본어 환경에서 사용되는 smpwservices.fcc 파일은 코딩 설정이 잘못되어 있습니다. 이 올바르지 않은 설정으로 인해 웹 페이지가 잘못된 문자로 표시됩니다. Apache 기반 웹 서버의 httpd.conf 파일에 지시문을 추가하여 이 문제를 해결하십시오.

다음 단계를 수행하십시오.

1. Apache 기반 웹 서버에 로그인합니다.
2. 텍스트 편집기에서 httpd.conf 파일을 엽니다.
3. 파일 끝에 빈 줄을 추가합니다.
4. 빈 줄에 다음 지시문을 추가합니다.

   ```
   BrowserMatch ".*" suppress-error-charset
   ```

5. httpd.conf 파일을 저장하고 텍스트 편집기를 닫습니다.
6. Apache 기반 웹 서버를 중지합니다.
7. Apache 기반 웹 서버를 시작합니다.

제 12 장: FCC 국제화

이 섹션은 다음 항목을 포함하고 있습니다.

FCC 국제화를 활성화하는 방법

SiteMinder 는 FCC 국제화를 위한 다음과 같은 기능을 지원합니다.

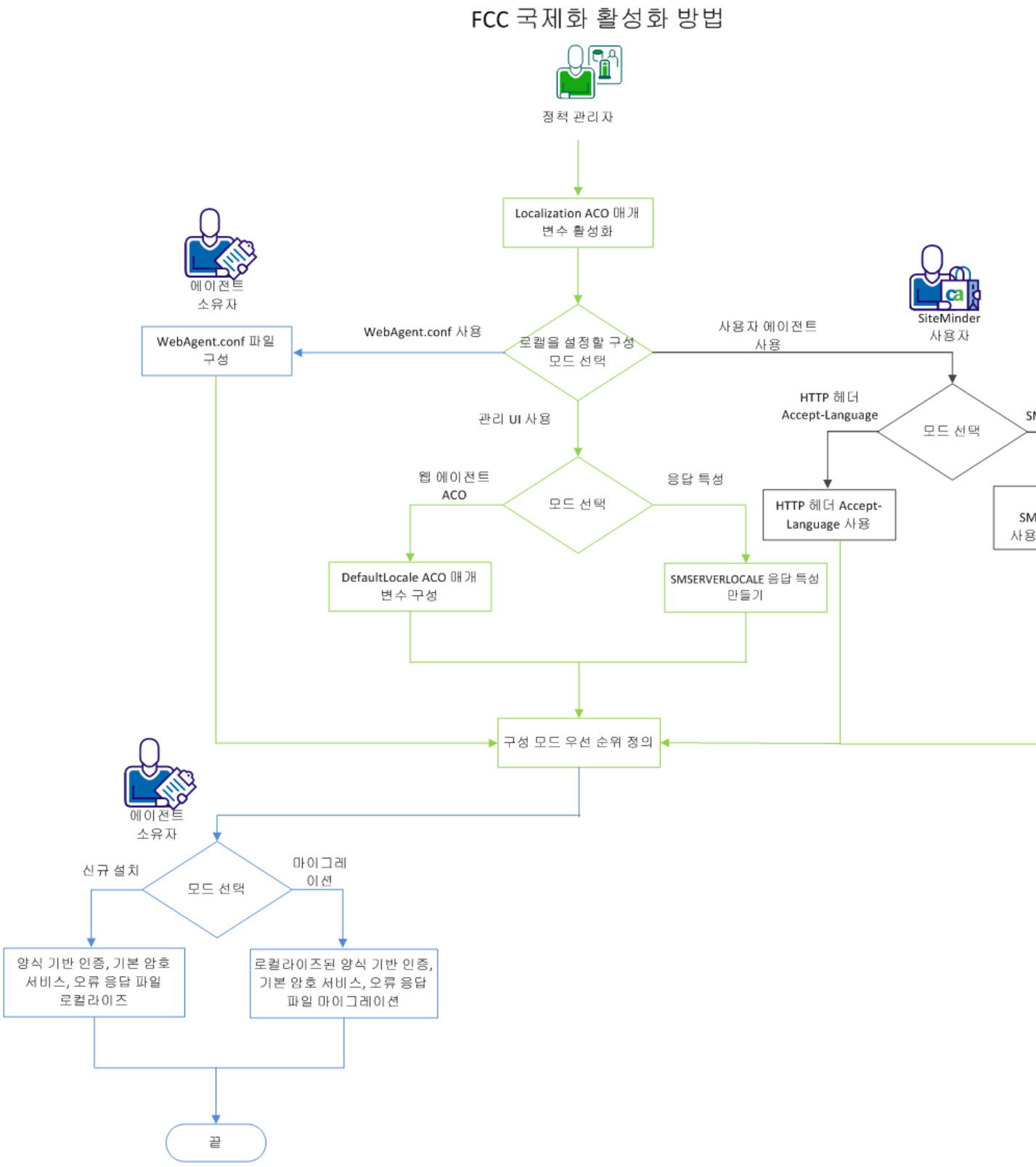
- 인증 체계에 대한 FCC
- 기본 암호 서비스
- 오류 응답 페이지

SiteMinder 가 로그인, 기본 암호 서비스, 오류 응답을 위한 HTML 페이지에 표시할 때 사용할 로캘을 설정할 수 있습니다. 조직에서 이 시나리오를 구현하려면 다음과 같은 사용자 역할이 필요합니다.

- 정책 관리자
- 에이전트 소유자
- SiteMinder 사용자

다음 다이어그램은 FCC 국제화를 위해 각 사용자가 반드시 수행해야 하는 단계를 설명합니다.

참고: 다음 다이어그램에서 파란색은 정책 관리자의 태스크를, 노란색은 에이전트 소유자의 태스크를, 검은색은 SiteMinder 사용자의 태스크를 각각 나타냅니다.


FCC 국제화 활성화 방법
정책 관리자
Localization ACO 매개 변수 활성화
에이전트 소유자
WebAgent.conf 파일 구성
WebAgent.conf 사용
로캘을 설정할 구성 모드 선택
사용자 에이전트 사용
SiteMinder 사용자
HTTP 헤더 Accept-Language
모드 선택
관리 UI 사용
웹 에이전트 ACO
모드 선택
응답 특성
HTTP 헤더 Accept-Language 사용
DefaultLocale ACO 매개 변수 구성
SMSERVERLOCALE 응답 특성 만들기
구성 모드 우선 순위 정의
에이전트 소유자
신규 설치
모드 선택
마이그레이션
양식 기반 인증, 기본 암호 서비스, 오류 응답 파일 로컬라이즈
로컬라이즈된 양식 기반 인증, 기본 암호 서비스, 오류 응답 파일 마이그레이션
끝

FCC 국제화를 활성화하려면 다음 단계를 수행해야 합니다.

1. 관리 UI 를 사용하여 Localization ACO 매개 변수를 활성화합니다 (페이지 221). 이 태스크는 정책 관리자가 수행해야 합니다.
2. 선호하는 로캘을 선택하기 위해 구성 모드를 선택합니다 (페이지 221). 이 태스크는 다음 방법으로 수행할 수 있습니다.
   - 사용자 에이전트 사용. SiteMinder 사용자는 다음 방법으로 사용자 에이전트를 사용하여 선호하는 로캘을 설정할 수 있습니다.
     - HTTP 헤더 SMCLIENTLOCALE 사용
     - HTTP 헤더 Accept-Language 사용 (페이지 222)
   - 관리 UI 사용 (페이지 222). 정책 관리자는 다음 방법으로 관리 UI 를 사용하여 선호하는 로캘을 설정할 수 있습니다.
     - SMSERVERLOCALE 응답 특성 사용 (페이지 222)
     - DefaultLocale ACO 매개 변수 사용 (페이지 223)
   - WebAgent.conf 파일 사용 (페이지 224). 에이전트 소유자는 WebAgent.conf 파일에서 Locale 매개 변수를 구성하여 선호하는 로캘을 설정할 수 있습니다.
3. 구성 모드의 우선 순위를 정의합니다 (페이지 224). 정책 관리자는 관리 UI 를 사용하여 이 태스크를 수행해야 합니다.
4. 다음 단계 중 *하나*를 수행합니다.
   - HTML 페이지를 처음 로컬라이즈하는 경우 에이전트 소유자가 다음 단계를 수행해야 합니다.
     a. 양식 기반 인증 파일 및 기본 암호 서비스 파일을 로컬라이즈합니다 (페이지 225).
     b. 오류 응답 파일을 로컬라이즈합니다 (페이지 227).
   - 로컬라이즈된 파일을 마이그레이션하려면 에이전트 소유자가 다음 단계를 수행해야 합니다.
     a. 양식 기반 인증 파일 및 기본 암호 서비스 파일을 마이그레이션합니다 (페이지 230).
     b. 로컬라이즈된 오류 응답 파일을 마이그레이션합니다 (페이지 231).

SiteMinder 가 FCC 국제화를 활성화하도록 구성되었습니다.

로컬라이제이션 매개 변수 활성화

지원되는 기능을 로컬라이즈하려면 정책 관리자가 Localization ACO 매개 변수를 true 로 설정해야 합니다.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그온합니다.
2. "인프라", "에이전트"를 차례로 클릭합니다.
3. "에이전트 구성 개체"를 클릭합니다.
4. 웹 에이전트의 에이전트 구성 개체로 이동합니다.
5. 에이전트 구성 개체의 이름을 클릭합니다.
6. "수정"을 클릭합니다.
7. Localization 매개 변수로 이동한 다음 매개 변수 옆의 오른쪽 화살표를 클릭합니다.

   "매개 변수 편집" 페이지가 열립니다.
8. 매개 변수의 값을 yes 로 설정합니다.
9. 변경 내용을 저장합니다.

선호하는 로캘 설정

Accept-Language HTTP 헤더 형식으로 여러 선호하는 로캘을 정의할 수 있습니다.

예: `es-ES,en-US;q=0.8,as-IN;q=0.6,fr-FR;q=0.4,en-IN;q=0.2`

로캘은 런타임 중에 동적으로 결정되지만 로캘은 요청 처리 중에 변경될 수 있습니다. 예를 들어, 성공적인 인증 후에 로캘이 변경될 수 있습니다.

다음 구성 모드에서 선호하는 로캘을 정의하십시오.

- 사용자 에이전트 사용
- 관리 관리 UI 사용
- WebAgent.conf 파일 사용

사용자 에이전트를 통해 선호하는 로캘 설정

사용자는 선호하는 로캘로 브라우저 설정을 구성할 수 있습니다. 정책 서버는 선호하는 로캘을 사용하여 지원되는 기능과 관련된 요청을 보냅니다.

사용자는 다음 방법을 사용하여 로캘을 설정할 수 있습니다.

- HTTP 헤더 SMCLIENTLOCALE 사용
- HTTP 헤더 Accept-Language 사용

HTTP 헤더 SMCLIENTLOCALE 사용

웹 요청에 대한 HTTP 헤더 SMCLIENTLOCALE 을 사용하여 선호하는 로캘을 설정하십시오. 웹 응용 프로그램이 선호하는 로캘을 설정하는 옵션을 제공하는 경우 사용자는 로캘을 선택할 수 있습니다. 정책 서버는 웹 요청의 SMCLIENTLOCALE 헤더를 사용하여 로캘을 받습니다. 로캘이 있는 경우 지원되는 기능이 나열된 페이지가 요청된 로캘로 표시됩니다.

HTTP 헤더 Accept-Language 사용

사용자는 로컬 웹 브라우저 설정을 사용하여 선호하는 로캘을 설정할 수 있습니다. 웹 요청의 HTTP 헤더 Accept-Language 는 로캘을 지정합니다.

관리 UI 를 통해 선호하는 로캘 설정

정책 관리자는 관리 UI 를 사용하여 선호하는 로캘을 설정할 수 있습니다. 정책 서버는 이 로캘을 사용하여 지원되는 기능과 관련된 요청을 보냅니다.

SMSERVERLOCALE 응답 헤더 변수 설정

선호하는 로캘을 응답으로 설정하려면 SMSERVERLOCALE 응답 특성을 사용하십시오.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그온합니다.
2. "정책", "도메인"을 차례로 클릭합니다.
3. "응답"을 클릭합니다.
4. 수정할 응답을 편집합니다.

5. "응답 정의" 탭을 클릭합니다.
6. 수정할 응답 특성을 편집합니다.
7. "특성 유형"을 "WebAgent-HTTP-Header 변수"로 선택합니다.
8. "변수 이름"에 SMSERVERLOCALE 을 입력합니다.
9. "변수 값"에 선호하는 로캘을 입력합니다.
10. 변경 내용을 저장합니다.

DefaultLocale ACO 속성 설정

DefaultLocale ACO 매개 변수에 선호하는 로캘 정보를 설정하십시오.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그온합니다.
2. "인프라", "에이전트"를 차례로 클릭합니다.
3. "에이전트 구성 개체"를 클릭합니다.
4. 웹 에이전트의 에이전트 구성 개체로 이동합니다.
5. 에이전트 구성 개체의 이름을 클릭합니다.
6. "수정"을 클릭합니다.

   설정과 컨트롤이 활성화됩니다.
7. DefaultLocale 매개 변수로 이동한 다음 매개 변수 옆의 오른쪽 화살표를 클릭합니다.

   "매개 변수 편집" 페이지가 열립니다.
8. 선호하는 로캘로 속성을 업데이트합니다.
9. 변경 내용을 저장합니다.

WebAgent.conf 파일을 사용하여 선호하는 로캘 설정

Locale 속성을 에이전트 소유자로서 설정하려면 WebAgent.conf 파일을 구성하십시오. 이 파일은 설치 중 지정되는 로캘 정보가 수록되어 있습니다.

다음 단계를 수행하십시오.

1. 웹 에이전트 컴퓨터에 로그온합니다.
2. 텍스트 편집기에서 WebAgent.conf 파일을 엽니다.
3. Locale 속성으로 이동하여 선호하는 로캘로 값을 설정합니다.
4. 변경 내용을 저장합니다.

구성 모드 우선 순위 정의

5 개 구성 모드의 우선 순위를 결정하려면 관리자는 ClientLocalePreferred ACO 매개 변수를 설정해야 합니다. ClientLocalePreferred 매개 변수가 true 로 설정되면 시스템은 구성 모드를 검사하여 다음 순서로 로캘을 결정합니다.

1. HTTP 헤더 SMCLIENTLOCALE
2. HTTP 헤더 Accept-Language
3. 응답 헤더 변수 SMSERVERLOCALE
4. DefaultLocale ACO 매개 변수
5. WebAgent.conf 파일의 Locale 속성

ClientLocalePreferred 가 false 로 설정되면 시스템은 구성 모드를 검사하여 다음 순서로 로캘을 결정합니다.

1. 응답 헤더 변수 SMSERVERLOCALE
2. HTTP 헤더 SMCLIENTLOCALE
3. HTTP 헤더 Accept-Language
4. DefaultLocale ACO 매개 변수
5. WebAgent.conf 파일의 Locale 속성

정책 서버는 정의된 모드를 사용하여 선호하는 로캘을 처리하고 선호하는 로캘의 목록을 만듭니다. 이 목록은 각 선호하는 로캘의 q 요소를 기준으로 정렬됩니다. 웹 요청이 전달되면 이 목록은 지원되는 기능의 HTML 페이지가 표시되는 로캘을 나타냅니다.

ClientLocalePreferred 매개 변수를 로캘의 원하는 순서로 설정하십시오.

다음 단계를 수행하십시오.

1. "인프라", "에이전트"를 차례로 클릭합니다.
2. "에이전트 구성 개체"를 클릭합니다.
3. 웹 에이전트의 에이전트 구성 개체로 이동합니다.
4. 에이전트 구성 개체의 이름을 클릭합니다.
5. "수정"을 클릭합니다.
6. 설정과 컨트롤이 활성화됩니다.
7. ClientLocalePreferred 매개 변수로 이동합니다.

   "매개 변수 편집" 페이지가 열립니다.
8. 매개 변수 값을 수정합니다.
9. 변경 내용을 저장합니다.

양식 기반 인증 및 기본 암호 서비스 파일 로컬라이즈

에이전트 소유자는 양식 기반 인증 및 기본 암호 서비스 파일을 로컬라이즈할 수 있습니다. 기본적으로, 여러 로캘로 된 샘플 양식이 제공됩니다. 이러한 샘플은 *webagent_home*/samples 디렉터리에 있습니다.

각 로캘에는 로캘 관련 파일을 위한 폴더가 있습니다. 각 폴더 및 파일에는 다음과 같은 형식의 RFC 3066 표준 명명 규칙을 따르는 로캘 언어 코드가 붙습니다.

forms_<*locale language code*>

forms_<*locale language code*>/*filename*_<*locale language code*>.fcc

예를 들면 다음과 같습니다.

영어의 경우 폴더 이름은 samples/forms_en-US, login.fcc 파일 이름은 login_en-US.fcc, WebAgent.properties 파일 이름은 WebAGent_en-US.properties 입니다.

샘플 양식 폴더와 별도로 기본 양식 폴더가 제공됩니다. 이 폴더에는 로캘 언어 코드가 없는 파일이 수록되어 있습니다. 웹 에이전트가 로캘에 대한 요청을 받으면 정책 서버는 선호하는 로캘의 목록에 요청된 로캘이 있는지 여부를 확인합니다. 그런 다음 시스템은 다음 태스크 중 *하나*를 수행합니다.

- 요청된 로캘이 목록에 있는 로캘과 일치하면 요청이 요청된 로캘로 처리됩니다.
- 요청된 로캘이 목록의 로캘과 일치하지 않으면 요청이 기본 양식 폴더의 로캘을 사용하여 처리됩니다.

새 로캘을 지원하도록 폴더 사용자 지정

기본 양식 폴더는 en-US 로 되어 있습니다. en-US 가 아닌 새 로캘을 지원하려면 에이전트 소유자가 기본 폴더 및 파일을 사용자 지정해야 합니다.

다음 단계를 수행하십시오.

1. 웹 에이전트 컴퓨터에 로그온합니다.
2. 다음 위치로 이동합니다.

   webagent_home/samples
3. 다음 형식으로 새 로캘에 대한 폴더를 만듭니다.

   forms_<*locale language code*>
4. .fcc 파일을 사용자 지정합니다.
   a. 관련 FCC 파일을 기본 양식 폴더에서 새 폴더로 복사합니다.
   b. 다음 형식으로 각 FCC 파일에 로캘의 언어 코드를 붙입니다.

      *filename*_<*locale language code*>.fcc
5. 속성 파일을 사용자 지정합니다.
   a. 관련 .properties 파일을 기본 양식 폴더에서 새 로캘 폴더로 복사합니다.
   b. 다음과 같이 각 파일에 새 로캘에 대한 언어 코드를 붙입니다.

      *filename*_<*locale language code*>.properties
6. 영어 메시지를 새 로캘의 언어로 변경하는 등, 로캘에 맞게 파일을 수정합니다.
7. 변경 내용을 저장합니다.

오류 응답 파일 로컬라이즈

에이전트 소유자는 오류 응답 파일을 로컬라이즈할 수 있습니다. 여러 로캘로 된 샘플 오류 응답 파일이 다음 디렉터리에서 제공됩니다.

webagent_home/samples/error-responses

각 로캘에는 로캘 관련 파일을 위한 폴더가 있습니다. 각 폴더 및 파일에는 다음과 같은 형식의 RFC 3066 표준 명명 규칙을 따르는 로캘 언어 코드가 붙습니다.

responses_<*locale language code*>

filename_<*locale language code*>.err

샘플 오류 응답 폴더 이외에, 로캘 언어 코드가 없는 기본 오류 응답 폴더 및 파일도 있습니다. 웹 에이전트가 특정 로캘의 요청을 받으면 정책 서버가 다음 태스크 중 하나를 수행합니다.

- 해당 로캘이 있고 오류가 발생하는 경우 해당 로캘 폴더의 오류 응답 파일을 사용하여 오류 응답이 표시됩니다.
- 해당 로캘이 없고 오류가 발생하는 경우 기본 오류 응답 폴더의 파일을 사용하여 오류 응답이 표시됩니다.

새 로캘을 지원하도록 SiteMinder 사용자 지정

기본적으로 제공되지 않는 새 로캘을 지원하려면 기본 폴더 및 파일을 사용자 지정하십시오.

선호하는 로캘에 맞게 다음과 같은 오류 응답 파일을 사용자 지정할 수 있습니다.

- CSS 파일에 대해 생성된 csserror.err
- 서버 오류에 대해 생성된 servererror.err
- 쿠키 오류에 대해 생성된 cookieerror.err
- 인증 및 인증 오류에 대해 생성된 custom401error.err

오류가 발생하면 다음 위치에서 오류 페이지를 가져옵니다.

ErrorResponseLocation_home/responses_*<locale language code>*

예:

영어 로캘의 경우 오류 응답 파일은 *ErrorResponseLocation_home*/responses_en-US 폴더에 저장됩니다. 쿠키 오류가 발생하면 cookieerror_en-US.err 오류 응답 파일을 이 폴더에서 가져옵니다.

오류 응답을 사용자 지정하려면 다음 단계를 수행하십시오.

1. 웹 에이전트를 구성합니다.
2. 웹 에이전트 ACO 를 구성합니다.

웹 에이전트 구성

오류 응답 파일을 로컬라이즈하려면 에이전트 소유자가 웹 에이전트를 구성해야 합니다.

다음 단계를 수행하십시오.

1. 웹 에이전트 컴퓨터에 로그온합니다.
2. 다음 위치로 이동합니다.

   webagent_home/samples/error-responses
3. 다음 형식으로 새 로캘에 대한 폴더를 만듭니다.

   responses_*<locale language code>*
4. 응답 파일을 기본 응답 폴더에서 새 폴더로 복사합니다.
5. 다음과 같이 각 오류 응답 파일에 새 로캘에 대한 언어 코드를 붙입니다.

   filename_<locale language code>
6. 새 로캘에 맞게 파일을 수정합니다. 예를 들어, 영어 메시지를 새 로캘에 대한 언어로 변경하십시오.

웹 에이전트 ACO 구성

오류 응답 파일을 로컬라이즈하려면 정책 관리자가 에이전트 ACO 매개 변수를 구성해야 합니다.

다음 단계를 수행하십시오.

1. 관리 UI 에 로그온합니다.
2. "인프라", "에이전트"를 차례로 클릭합니다.
3. "에이전트 구성 개체"를 클릭합니다.
4. 웹 에이전트의 에이전트 구성 개체로 이동합니다.
5. 에이전트 구성 개체의 이름을 클릭합니다.
6. "수정"을 클릭합니다.
7. 다음 매개 변수의 값이 'no'로 설정되었는지 확인합니다.
   - servererrorfile
   - csserrorfile
   - reqcookieerrorfile
   - custom401errorfile

   참고: 이러한 속성은 ErrorResponseLocation 매개 변수보다 우선 순위가 더 높습니다. 오류 응답 파일을 로컬라이즈하려면 이러한 속성을 비활성화하십시오.
8. ErrorResponseLocation 매개 변수로 이동한 다음 매개 변수 옆의 오른쪽 화살표를 클릭합니다.

   "매개 변수 편집" 페이지가 표시됩니다.
9. 오류 응답 파일의 새 경로를 입력합니다.

   참고: ErrorResponseLocation 속성의 값은 URL 이 아니어야 합니다.
10. 변경 내용을 저장합니다.

로컬라이즈된 파일 마이그레이션

12.51 이전 릴리스에 로컬라이즈된 FCC 파일, 기본 암호 서비스, 오류 응답 파일이 있는 경우, 에이전트 소유자는 이러한 파일을 재사용하기 위해 파일을 12.51 이상으로 마이그레이션할 수 있습니다.

양식 기반 인증 파일 및 기본 암호 서비스 파일 마이그레이션

에이전트 소유자는 양식 기반 인증 파일 및 기본 암호 서비스 파일을 마이그레이션할 수 있습니다.

다음 단계를 수행하십시오.

1. 웹 에이전트 컴퓨터에 로그온합니다.
2. 로컬라이즈된 파일이 있는 위치로 이동합니다.

   기본 경로: *webagent_home*/samples
3. 폴더를 만들거나 다음 단계로 건너뛰고 기존 로캘 폴더의 이름을 변경합니다. 다음 단계 중 *하나*만 수행하십시오.
   a. 다음 형식으로 폴더를 만듭니다.

      `forms_<locale language code>`
   b. 기존 로컬라이즈된 FCC 파일을 새 폴더로 복사합니다.
   c. 다음 형식으로 각 파일에 로캘의 언어 코드를 붙입니다.

      `filename_<locale language code>.fcc`
   d. 관련 .properties 파일을 기존 폴더에서 새 로캘의 양식 폴더로 복사합니다.
   e. 다음 형식으로 각 파일에 로캘의 언어 코드를 붙입니다.

      `filename_<locale language code>.properties`
4. (선택 사항) 기존 폴더의 이름을 변경합니다(이 단계는 폴더 생성 방법에 대한 대안).
   a. 다음 형식으로 기존 폴더의 이름을 변경합니다.

      `forms_<locale language code>`
   b. 다음 형식으로 로캘의 언어 코드를 사용하여 각 파일 이름을 수정합니다.

      `filename_<locale language code>.fcc`
   c. 다음 형식으로 로캘의 언어 코드를 사용하여 각 파일 이름을 수정합니다.

      `filename_<locale language code>.properties`

5. FCC 파일에서 다음 단계를 수행합니다.
   a. 하드 코드된 인코딩 매개 변수를 $$SMENC$$ 태그로 바꿉니다.

      예:

      - <meta http-equiv="Content-Type" content="text/html;charset=$$SMENC$$">
      - <INPUT TYPE=HIDDEN NAME="SMENC" VALUE="$$SMENC$$">

   b. 각 파일의 첫 번째 줄은 양식을 인코딩하는 데 사용한 문자 집합의 이름을 포함해야 합니다. 예: <!-- SiteMinder Encoding=UTF-8; -->. 여기서 UTF-8 은 양식을 인코딩하는 데 사용됩니다.
6. 변경 내용을 저장합니다.

오류 응답 파일 마이그레이션

에이전트 소유자는 로컬라이즈된 오류 응답 파일을 마이그레이션할 수 있습니다.

다음 단계를 수행하십시오.

1. 웹 에이전트 컴퓨터에 로그온합니다.
2. 로컬라이즈된 파일이 있는 위치로 이동합니다.

   기본 경로: *webagent_home*/samples/error-responses
3. 폴더를 만들려면 다음 단계를 수행하십시오. 또는, 다음 단계로 건너뛰어 기존 폴더의 이름을 변경하십시오.
   a. 다음 형식으로 폴더를 만듭니다.

      ```
      responses_<locale language code>
      ```

   b. 기존 로컬라이즈된 오류 응답 파일을 새 폴더로 복사합니다.
   c. 다음 형식으로 각 파일에 로캘의 언어 코드를 붙입니다.

      ```
      filename_<locale language code>
      ```

4. (선택 사항) 기존 폴더의 이름을 변경하려면:
   a. 다음 형식으로 기존 파일의 이름을 변경합니다.

      ```
      responses_<locale language code>
      ```

   b. 다음 형식으로 로캘의 언어 코드를 사용하여 각 파일 이름을 수정합니다.

      ```
      filename_<locale language code>
      ```

FCC 국제화가 이제 완료되었습니다.

제 13 장: 에이전트 및 암호 서비스

이 섹션은 다음 항목을 포함하고 있습니다.

FCC 암호 서비스를 구성하는 방법

암호 서비스를 구성하려면 다음 단계를 수행하십시오.

1. 관리 UI 를 엽니다.
2. SiteMinder 환경의 사용자 디렉터리와 연결되는 암호 정책을 생성합니다. "리디렉션 URL" 필드에 다음 경로를 사용하십시오.

   ```
   /siteminderagent/forms/smpwservices.fcc
   ```

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

암호 서비스 구현

SiteMinder 에서는 FCC(양식 자격 증명 수집기)를 사용하여 암호 서비스를 지원합니다.

암호 서비스를 사용하면 다음 태스크를 수행할 수 있습니다.

- 암호 서비스 URL 의 쿼리 문자열을 암호화합니다 (페이지 234).
- 다국어 암호 서비스를 지원합니다 (페이지 235).
- 암호 서비스 사용자를 정규화된 URL 로 리디렉션합니다 (페이지 236).
- 암호 서비스로 SecureID 인증을 지원합니다 (페이지 237).
- 사용자가 자신의 암호를 변경할 수 있도록 합니다. 다음 중 상황에 맞는 절차를 사용하십시오.
  - SecureURLs 매개 변수가 no 로 설정된 경우 암호 변경 (페이지 238)
  - SecureURLs 매개 변수가 yes 로 설정된 경우 암호 변경 (페이지 240)
  - 기본 인증 또는 X.509 인증서 인증 체계를 사용하는 경우 암호 변경 (페이지 242)

FCC 암호 서비스 및 URL 쿼리 암호화

FCC 암호 서비스 응용 프로그램을 사용하면 URL 의 쿼리 데이터를 암호화하고 에이전트 상호 작용의 보안을 설정할 수 있습니다. FCC 암호 서비스로 쿼리 데이터만 암호화할 수 있습니다. 다음과 같은 FCC 암호 서비스 파일이 사용됩니다.

- smpwservices.fcc

  이 FCC 는 웹 에이전트와 함께 설치되며 다음 디렉터리에 저장됩니다.

  web_agent_home/samples/forms

  암호 정책이 구성되지 않은 상태에서 암호 서비스가 호출되면 정책 서버의 SiteMinder 관리자가 NETE_PWSERVICES_REDIRECT 환경 변수를 smpwservices.fcc 의 상대 경로로 설정해야 합니다.

  경로는 다음과 같습니다.

  /siteminderagent/forms/smpwservices.fcc

  FCC 지시문 authreason 과 username 에 따라 새 FCC 에 암호 서비스가 표시됩니다.

- smpwservices.unauth

  이 파일은 암호 서비스 양식의 GET/POST 작업 중에 발생하는 오류를 처리합니다.

  이 파일은 POST 작업 중에 요청을 처리하지 못 할 경우 호출되는 다른 파일(FCC 에서 허용되지 않은 파일)과 비슷합니다. 이 FCC 는 오류 조건(예: 비어 있는 TARGET 변수)을 처리합니다. 오류 보고는 CGI 기반 암호 서비스와 동기화되고 FCC POST 로 인해 발생하는 다른 알 수 없는 오류를 처리하는 데 사용됩니다.

- smpwservicesUS-EN.properties

  이 속성 파일은 사용자에게 친숙한 메시지를 암호 서비스 양식에 표시하기 위해 smpwservices.fcc 에서 사용합니다.

  이 속성 파일에는 사용자에게 친숙한 메시지가 포함되며, 관리자는 암호 서비스 양식에 표시할 항목에 맞게 내용을 수정할 수 있습니다. 메시지 형식은 이름=값입니다.

FCC 기반 암호 서비스 변경 양식을 지역화하는 방법

다른 로캘에 대한 FCC 기반 암호 서비스의 사용자 메시지를 지역화하려면 다음 단계를 수행하십시오.

1. 웹 서버에 새 로캘을 위한 FCC 폴더를 생성하거나 현재 로캘의 기존 폴더를 사용합니다. 이 폴더에 대한 일반적인 명명 규칙은 forms*locale* 입니다.

   참고: 표시되는 디렉터리 및 파일 이름은 운영 환경과 사용 중인 웹 서버 유형에 따라 대/소문자를 구분할 수 있습니다.

2. 관련된 암호 서비스 파일을 새 폴더에 복사합니다.
3. 로캘에 맞게 파일을 수정합니다. 예를 들어 영어로 된 메시지를 로캘 언어로 변경합니다. 로캘의 모든 파일에 대해 이 단계를 반복하십시오.
4. 관리 UI 의 "암호 정책"에서 "리디렉션 URL" 필드의 값을 변경합니다.

   예를 들어 일본어 사용자용 FCC 암호 서비스를 사용하려면 다음 파일을 *web_agent_home*/samples 아래의 formsja 폴더에 복사합니다.

   - smpwservices.fcc - *web_agent_home*/samples/forms 에 있음
   - smpwservices.unauth - *web_agent_home*/samples/forms 에 있음
   - smpwservicesja.properties - 새 속성 파일

정규화된 URL 을 사용하여 암호 서비스 리디렉션

암호 서비스를 사용하는 경우 사용자가 리디렉션되는 FQDN(정규화된 도메인 이름)을 생성하도록 웹 에이전트에 지시할 수 있습니다. 다음 매개 변수를 사용하십시오.

ConstructFullPwsvcUrl

사용자를 리디렉션하기 전에 암호 서비스를 호스트하는 시스템의 서버 이름(FQDN)을 추가하도록 에이전트에 지시합니다. 이 서버 이름은 정책 서버의 암호 정책에 정의합니다.

예를 들어 이 매개 변수의 값이 yes 이고 암호 정책이 siteminderagent/forms/smpwservices.fcc 를 가리킨다고 가정해 보십시오. 그러면 웹 에이전트가 다음 URL 로 리디렉션합니다.

HTTP://*server_name*.example.com/siteminderagent/forms/smpwservices.fcc

웹 에이전트는 이 매개 변수의 값이 no 일 경우 암호 정책에 정의된 값을 사용합니다. 예를 들어 암호 정책이 하위 디렉터리만 가리킬 경우 웹 에이전트는 사용자를 이 하위 디렉터리로 리디렉션합니다.

기본값: No

예: No(암호 정책에 정의된 /siteminderagent/forms/smpwservices.fcc 로 리디렉션)

예: Yes(HTTP://*server_name*.example.com 을 암호 정책에 정의된 /siteminderagent/forms/smpwservices.fcc 에 추가)

관리 UI 에서 암호 정책의 기준 URL 은 서버 이름을 포함하지 *않습니다*. 위에서 설명한 매개 변수의 값이 yes 로 설정되면 웹 에이전트는 암호 정책에 있는 URL 로 사용자를 리디렉션합니다.

다음 표의 예제를 사용하여 ConstructFullPwsvcURl 매개 변수를 설정합니다.

원하는 작업	관리 UI 에서 암호 정책에 추가할 URL	ConstructFullPwsvcURl 설정값
특정 서버의 암호 서비스 호스트	http://server_name.example.com:80/siteminderagent/forms/smpwservices.fcc	아니요
웹 에이전트와 같은 서버의 암호 서비스 호스트(상대 URL 사용)	siteminderagent/forms/smpwservices.fcc	아니요

원하는 작업	관리 UI 에서 암호 정책에 추가할 URL	ConstructFullPwsvcURI 설정값
웹 에이전트와 같은 서버의 암호 서비스 호스트(FQDN 사용)	siteminderagent/forms/smpwservices.fcc	예

FCC 암호 서비스를 사용하여 SecureID 인증 구성

SecureID 를 인증 체계로 사용하고 환경에서 다음 두 조건이 모두 충족되는 경우 관리 UI 를 사용하여 SecureID HTML 양식 템플릿을 수정해야 합니다.

- FCC 암호 서비스 기능이 구성되었습니다.
- 웹 에이전트의 SecureUrls 매개 변수 값이 yes 로 설정됩니다.

SecureID 는 암호 서비스를 사용하여 구현되므로 인증 체계의 템플릿을 수정해야 합니다.

FCC 암호 서비스를 사용하여 SecureID 인증을 구성하려면 다음과 같이 SecureID 템플릿의 "Target" 필드에 smpwservices.fcc 파일 경로를 추가합니다.

```
/siteminderagent/forms/smpwservices.fcc
```

FCC 를 사용하여 사용자가 직접 암호를 변경할 수 있도록 설정하는 방법

원하는 경우 사용자가 자신의 암호를 변경할 수 있도록 SiteMinder 의 FCC 암호 서비스 기능을 구성할 수 있습니다.

참고: SiteMinder 웹 에이전트 구성의 SecureURLs 매개 변수가 no 로 설정된 경우 다음 프로세스를 사용하십시오.

FCC 를 사용하여 사용자가 직접 암호를 변경할 수 있도록 설정하려면 다음을 수행하십시오.

1. 사용자 디렉터리에 암호 정책을 지원하는 특성이 포함되어 있는지 확인합니다.
2. 관리 UI 를 사용하여 다음 태스크를 수행합니다.
   a. FCC 기반 암호 정책을 생성하고 원하는 리소스를 보호합니다.
   b. 권한 있는 사용자가 암호를 변경할 수 있도록 암호 정책을 구성합니다.
3. 다음 항목을 포함하는 암호 변경 URL 을 생성합니다.
   - 로그온 서버의 FQDN(예: http:logonserver.example.com)
   - FCC 기반 암호 서비스의 URI(예: siteminderagent/forms/smpwservices.fcc?)
   - SiteMinder 웹 에이전트의 이름(SMAGENTNAME)
   - 다음과 같은 대상 URL 중 *하나*
   - FCC 페이지에 포함된 암호 변경 URL 의 경우 아래 예제와 같이 (SMAGENTNAME) 및 (TARGET) 섹션에 대한 상대 값을 사용합니다.

     ```
     <a
     href="http:logonserver.example.com/siteminderagent/forms/smpwservices.fcc
     ?SMAUTHREASON=

     34&SMAGENTNAME=$$smencode(smagentname)$$&TARGET=$$smencode(target)$$">Cha
     nge Password</font></a>
     ```

   - FCC 페이지에 포함되지 *않은* 암호 변경 URL 의 경우 (SMAGENTNAME) 섹션에 대한 SiteMinder 에이전트의 이름을 하드 코드로 작성합니다. 그런 다음 아래 예제와 같이 (TARGET) 섹션에 대한 정규화된 도메인 이름 값을 하드 코드로 작성합니다.

     ```
     <a
     href="http://logonserver.example.com/siteminderagent/forms/smpwservices.f
     cc?SMAUTHREASON=34&SMAGENTNAME=Agent1&TARGET=https://logonserver.example.
     com/protected/myprotectedpage.html">Change Password</font></a>
     ```

4. 3 단계에서 생성한 암호 변경 URL 을 하나 이상의 보호되지 않은 웹 페이지의 링크로 포함합니다.
5. 다음 단계를 수행하여 암호 변경 기능을 테스트합니다.
   a. 3 단계에서 생성한 암호 변경 링크가 포함된 웹 페이지를 표시합니다.
   b. 암호 변경 링크를 클릭합니다.

      암호 변경 양식이 나타납니다.
   c. 암호 변경 양식을 작성한 후 제출합니다.

      암호가 제대로 변경되었으면 보호된 대상 리소스에 대한 링크와 함께 확인 페이지가 표시됩니다.
   d. 링크를 클릭하고 리소스가 나타나는지 확인합니다.
   e. 브라우저를 닫은 후 다시 엽니다. 그런 다음 새 암호를 사용하여 보호된 리소스에 액세스를 시도합니다.

      암호가 제대로 변경되었으면 새 암호를 사용하여 리소스에 액세스할 수 있습니다.

FCC 를 사용하여 사용자가 직접 암호를 변경할 수 있도록 설정하는 방법(SecureURLs=Yes)

원하는 경우 사용자가 자신의 암호를 변경할 수 있도록 SiteMinder 의 FCC 암호 서비스 기능을 구성할 수 있습니다.

참고: SiteMinder 웹 에이전트 구성의 SecureURLs 매개 변수가 yes 로 설정된 경우 다음 프로세스를 사용하십시오.

FCC 를 사용하여 사용자가 직접 암호를 변경할 수 있도록 설정하려면 다음을 수행하십시오.

1. 사용자 디렉터리에 암호 정책을 지원하는 특성이 포함되어 있는지 확인합니다.
2. 관리 UI 를 사용하여 다음 태스크를 수행합니다.
   a. FCC 기반 암호 정책을 생성하고 원하는 리소스를 보호합니다.
   b. 권한 있는 사용자가 암호를 변경할 수 있도록 암호 정책을 구성합니다.
   c. ValidTargetDomain 매개 변수의 값을 보호 대상 리소스의 도메인으로 설정합니다.
3. 다음 항목을 포함하는 암호 변경 URL 을 생성합니다.
   - 로그온 서버의 FQDN(예: http:logonserver.example.com)
   - FCC 기반 암호 서비스의 URI(예: siteminderagent/forms/smpwservices.fcc?)
   - SiteMinder 웹 에이전트의 이름(SMAGENTNAME)
   - 다음과 같은 대상 URL 중 *하나*
   - FCC 페이지에 포함된 암호 변경 URL 의 경우 아래 예제와 같이 (SMAGENTNAME) 및 (TARGET) 섹션에 대한 상대 값을 사용합니다.

     ```
     <a
     href="http:logonserver.example.com/siteminderagent/forms/smpwservices.fcc
     ?SMAUTHREASON=

     34&SMAGENTNAME=$$smencode(smagentname)$$&TARGET=$$smencode(target)$$">Cha
     nge Password</font></a>
     ```

   - FCC 페이지에 포함되지 *않은* 암호 변경 URL 의 경우 (SMAGENTNAME) 섹션에 대한 SiteMinder 에이전트의 이름을 하드 코드로 작성합니다. 그런 다음 아래 예제와 같이 (TARGET) 섹션에 대한 정규화된 도메인 이름 값을 하드 코드로 작성합니다.

```
<a
href="http://logonserver.example.com/siteminderagent/forms/smpwservices.f
cc?SMAUTHREASON=34&SMAGENTNAME=Agent1&TARGET=https://logonserver.example.
com/protected/myprotectedpage.html">Change Password</font></a>
```

4. 3 단계에서 생성한 암호 변경 URL 을 하나 이상의 보호되지 않은 웹 페이지의 링크로 포함합니다.
5. 웹 서버에서 다음 파일을 엽니다.

   web_agent_home/samples/forms/smpwservices.fcc

   a. 다음 행을 찾습니다.

      ```
      @smpwselfchange=0
      ```

   b. 이 행의 맨 끝에 있는 0 을 다음과 같이 1 로 변경합니다.

      ```
      @smpwselfchange=1
      ```

   c. smpwservices.fcc 파일을 저장한 후 닫습니다.
6. 3 단계에서 생성한 URL 을 하나 이상의 보호되지 않은 웹 페이지의 링크로 포함합니다.
7. 다음 단계를 수행하여 암호 변경 기능을 테스트합니다.

   a. 3 단계에서 생성한 암호 변경 링크가 포함된 웹 페이지를 표시합니다.

   b. 암호 변경 링크를 클릭합니다.

      암호 변경 양식이 나타납니다.

   c. 암호 변경 양식을 작성한 후 제출합니다.

      암호가 제대로 변경되었으면 보호된 대상 리소스에 대한 링크와 함께 확인 페이지가 표시됩니다.

   d. 링크를 클릭하고 리소스가 나타나는지 확인합니다.

   e. 브라우저를 닫은 후 다시 엽니다. 그런 다음 새 암호를 사용하여 보호된 리소스에 액세스를 시도합니다.

      암호가 제대로 변경되었으면 새 암호를 사용하여 리소스에 액세스할 수 있습니다.

사용자가 직접 암호를 변경할 수 있도록 설정하는 방법(SiteMinder X.509 인증서 및 기본 인증 체계를 사용하는 경우)

사용자가 자신의 암호를 변경할 수 있도록 SiteMinder 의 FCC 암호 서비스 기능을 구성할 수 있습니다. SiteMinder X.509 인증서 및 기본 인증 체계의 암호 변경 URL 은 HTTPS 프로토콜로 시작해야 합니다.

다음 단계를 수행하십시오.

1. 사용자 디렉터리에 암호 정책을 지원하는 특성이 포함되어 있는지 확인합니다.
2. 관리 UI 를 사용하여 다음 태스크를 수행합니다.
   a. FCC 기반 암호 정책을 생성하고 원하는 리소스를 보호합니다.
   b. 권한 있는 사용자가 암호를 변경할 수 있도록 암호 정책을 구성합니다.
3. 다음 항목을 포함하는 암호 변경 URL 을 생성합니다.
   - HTTPS 체계(프로토콜)
   - 로그온 서버의 FQDN(예: http:logonserver.example.com)
   - FCC 기반 암호 서비스의 URI(예: siteminderagent/forms/smpwservices.fcc?)
   - SiteMinder 웹 에이전트의 이름(SMAGENTNAME)
   - 다음과 같은 대상 URL 중 *하나*
   - FCC 페이지에 포함된 암호 변경 URL 의 경우 아래 예제와 같이 (SMAGENTNAME) 및 (TARGET) 섹션에 대한 상대 값을 사용합니다.

     ```
     <a
     href="https:logonserver.example.com/siteminderagent/forms/smpwservices.fcc?SMAUTHREASON=

     34&SMAGENTNAME=$$smencode(smagentname)$$&TARGET=$$smencode(target)$$">Change Password</font></a>
     ```

   - FCC 페이지에 포함되지 *않은* 암호 변경 URL 의 경우 (SMAGENTNAME) 섹션에 대한 SiteMinder 에이전트의 이름을 하드 코드로 작성합니다. 그런 다음 아래 예제와 같이 (TARGET) 섹션에 대한 정규화된 도메인 이름 값을 하드 코드로 작성합니다.

     ```
     <a
     href="https://logonserver.example.com/siteminderagent/forms/smpwservices.fcc?SMAUTHREASON=34&SMAGENTNAME=Agent1&TARGET=https://logonserver.example.com/protected/myprotectedpage.html">Change Password</font></a>
     ```

4. 3 단계에서 생성한 암호 변경 URL 을 하나 이상의 보호되지 않은 웹 페이지의 링크로 포함합니다.
5. 다음 단계를 수행하여 암호 변경 기능을 테스트합니다.
   a. 3 단계에서 생성한 암호 변경 링크가 포함된 웹 페이지를 표시합니다.
   b. 암호 변경 링크를 클릭합니다.

      암호 변경 양식이 나타납니다.
   c. 암호 변경 양식을 작성한 후 제출합니다.

      보호된 대상 리소스에 대한 링크와 함께 확인 페이지가 표시됩니다.
   d. 링크를 클릭하고 리소스가 나타나는지 확인합니다.
   e. 브라우저를 닫은 후 다시 엽니다. 그런 다음 새 암호를 사용하여 보호된 리소스에 액세스를 시도합니다.

      암호가 제대로 변경되었으면 새 암호를 사용하여 리소스에 액세스할 수 있습니다.

제 14 장: SSO

이 섹션은 다음 항목을 포함하고 있습니다.

OPTIONS 메서드를 사용하는 리소스에 자동 액세스 허용

SiteMinder 웹 에이전트는 인증된 사용자가 OPTIONS 메서드를 사용하는 리소스에 액세스하려고 하면 재인증을 요청합니다. OPTIONS 메서드를 사용하는 일부 리소스는 다음과 같습니다(이에 제한되지 않음).

- Microsoft® Word 문서
- Microsoft® Excel® 스프레드시트 문서

리소스와 연결된 응용 프로그램에서 OPTIONS 메서드를 사용하는 요청을 웹 서버에 보내기 때문에 인증 챌린지가 발생합니다. 이 요청에는 SiteMinder 쿠키가 포함되지 *않으므로* 웹 에이전트에서 인증을 시도합니다.

이러한 리소스에 액세스할 때 사용자에게 인증이 요청되지 않게 하려면

1. 다음 매개 변수의 값을 yes 로 설정합니다.

   autoauthorizeoptions

   HTTP OPTIONS 메서드를 사용하는 모든 리소스 요청에 자동으로 권한을 부여합니다.

   이 매개 변수의 값을 yes 로 설정하면 PersistentCookies 매개 변수의 값도 no 로 설정하십시오.

   제한: yes, no

2. PersistentCookies 매개 변수의 값을 no 로 설정합니다.

단일 도메인에서 싱글 사인온의 작동 방식

SiteMinder 에서는 단일 쿠키 도메인 및 여러 쿠키 도메인에 대해 싱글 사인온 기능을 제공합니다. 이 기능을 사용하면 여러 웹 서버 및 플랫폼에서 응용 프로그램 사용이 간소화되고, 사용자가 싱글 사인온 환경에서 이동할 때 재인증할 필요가 없으므로 사용자 환경이 개선됩니다.

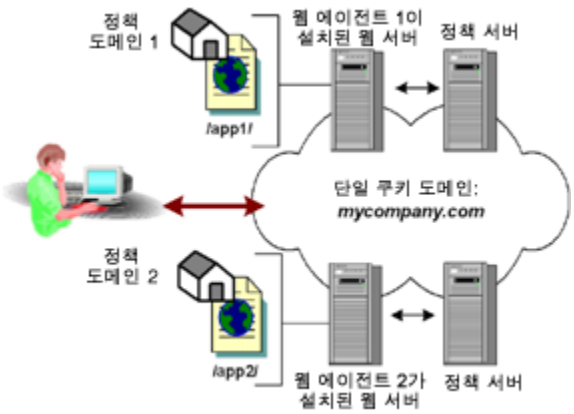
단일 도메인은 모든 리소스가 같은 쿠키 도메인에 있는 환경을 말합니다. 각 웹 에이전트의 구성에 같은 쿠키 도메인을 지정하면 해당 쿠키 도메인에 있는 여러 웹 에이전트를 싱글 사인온이 가능하도록 구성할 수 있습니다.

싱글 사인온이 가능하도록 설정되면 다음과 같은 순서로 진행됩니다.

1. 사용자가 한 번 인증됩니다.
2. 웹 에이전트가 성공한 인증을 캐시한 후 싱글 사인온 쿠키를 사용자의 브라우저에 발급합니다.
3. 싱글 사인온 쿠키에서 세션 정보를 제공하므로 사용자는 재인증 없이 다음과 같은 유형의 리소스에 액세스할 수 있습니다.
   - 다른 영역의 보호된 리소스 중 보호 수준이 *같거나 낮은* 리소스
   - 같은 쿠키 도메인의 다른 웹 서버

   사용자가 보호 수준이 *높은* 리소스에 액세스를 시도하는 경우에는 먼저 재인증해야 합니다.

다음 그림에서는 단일 쿠키 도메인의 싱글 사인온을 보여 줍니다.

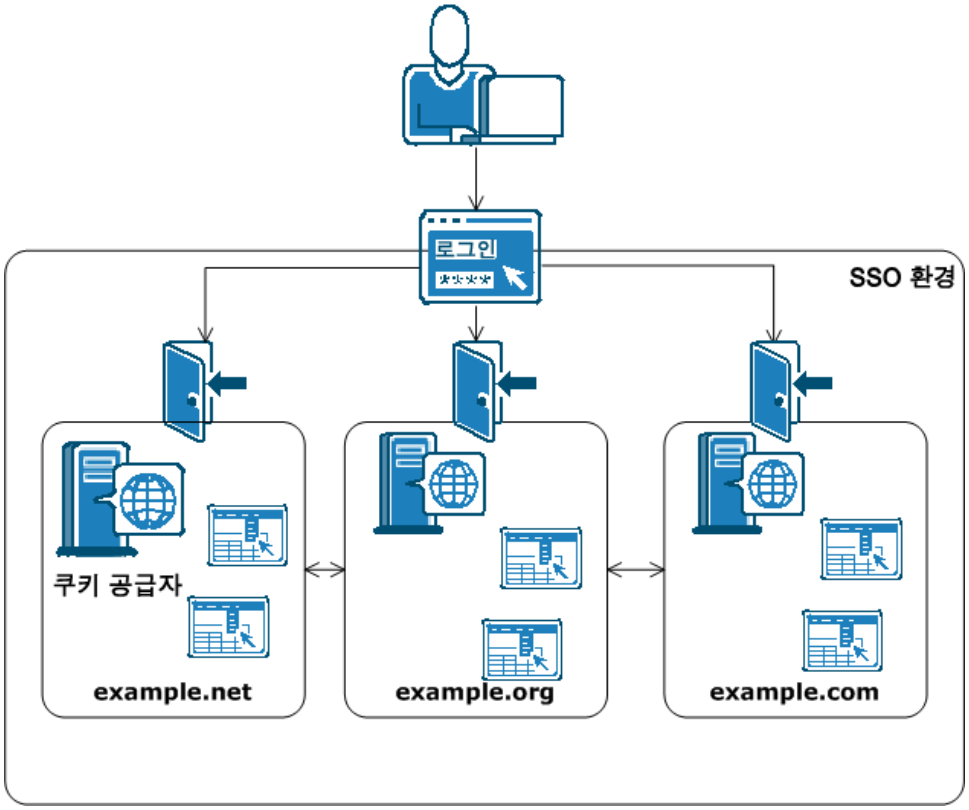


참고: 복제된 사용자 디렉터리를 복제되지 않은 정책 저장소와 함께 사용하는 경우 모든 정책 저장소에 대해 사용자 디렉터리의 이름을 동일하게 지정해야 합니다. 또한 세션 티켓을 암호화하는 세션 티켓 키가 SSO 환경의 모든 키 저장소에 대해 동일해야 합니다. 세션 티켓은 사용자 세션의 유효 기간을 결정합니다.

여러 도메인에 대한 싱글 사인온

싱글 사인온이 구성되어 있지 않으면 사용자가 다른 쿠키 도메인의 개별 서버에 있는 여러 응용 프로그램 및 리소스에 액세스할 때 로그온 과정을 거치고 자격 증명을 여러 번 입력해야 합니다. 여러 쿠키 도메인에 대해 싱글 사인온 정보를 전달하는 기능은 사용자가 한 쿠키 도메인의 사이트에서 인증한 후 자격 증명을 다시 입력하지 않고 다른 쿠키 도메인의 사이트로 이동할 수 있도록 합니다. 사용자는 이러한 원활한 이동을 통해 관련 사이트를 더 쉽게 사용할 수 있습니다.

다음 그림에서는 여러 쿠키 도메인에 대한 싱글 사인온을 보여 줍니다.



여러 쿠키 도메인에 대한 싱글 사인온 및 하드웨어 부하 분산 장치

SiteMinder 는 쿠키 공급자로 구성된 SiteMinder 웹 에이전트를 사용하여 여러 쿠키 도메인에 대해 싱글 사인온을 구현합니다.

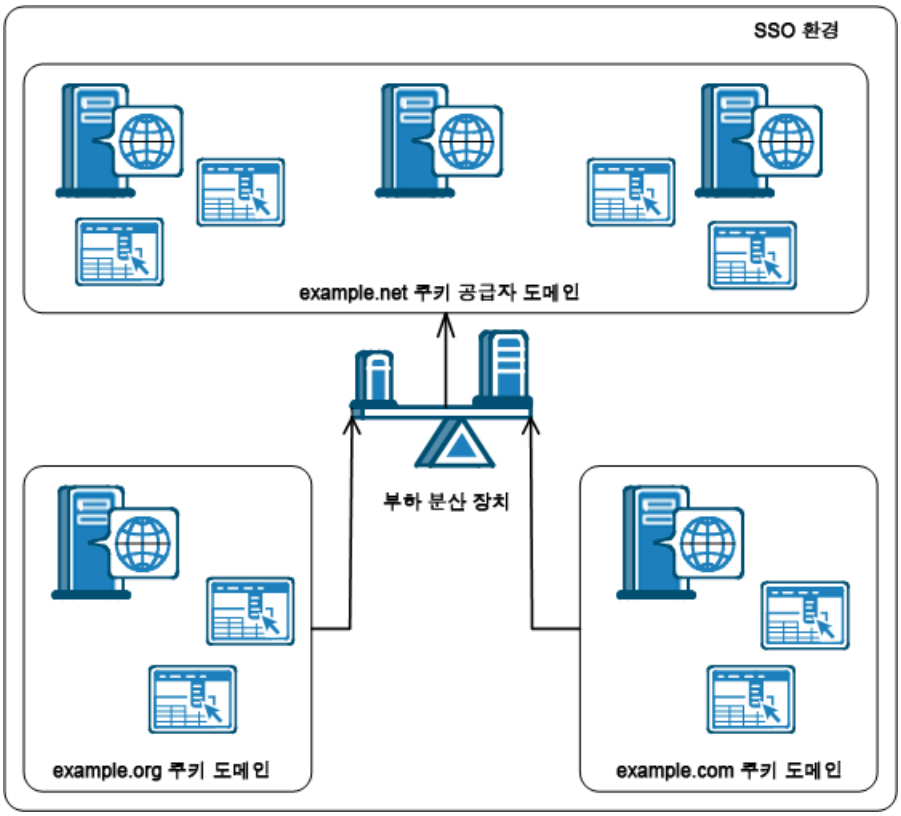
쿠키 공급자 웹 에이전트가 있는 쿠키 도메인을 쿠키 공급자 도메인이라고 합니다. 싱글 사인온 환경 내에서 다른 쿠키 도메인의 다른 모든 웹 에이전트는 하나의 쿠키 공급자를 가리킵니다.

SiteMinder 쿠키 공급자는 다음과 같은 프로세스로 동작합니다.

1. 사용자가 싱글 사인온 환경 내의 도메인에서 보호된 리소스를 요청하고 사용자에게 자격 증명이 요청됩니다.
2. 사용자가 인증되면 다음 쿠키가 사용자 브라우저에 설정됩니다.
   - 사용자가 인증된 도메인의 로컬 쿠키
   - 쿠키 공급자가 설정하는 쿠키
3. 사용자는 다음과 같은 이벤트가 발생할 때까지 다시 인증하지 않고 싱글 사인온 환경에서 여러 도메인을 탐색할 수 있습니다.
   - 사용자 세션이 만료됩니다.
   - 사용자가 세션을 끝냅니다(예: 브라우저 닫기).

싱글 사인온 환경의 에이전트는 부하 분산 기능을 사용합니까?

SSO 환경의 모든 에이전트는 단일 쿠키 공급자 도메인을 참조해야 합니다. 쿠키 공급자 도메인의 웹 서버와 SSO 환경의 다른 쿠키 도메인 사이에 부하 분산 장치를 추가하십시오. 다음 그림은 이러한 예를 보여 줍니다.



example.org 쿠키 도메인의 웹 에이전트와 example.com 쿠키 도메인의 웹 에이전트는 둘 다 example.net 이라는 같은 쿠키 공급자 도메인을 가리킵니다. 부하 분산 장치는 example.net 쿠키 공급자 도메인의 모든 웹 서버 사이에 균등하게 트래픽을 배분합니다.

참고: 여러 쿠키 도메인에 대해 SSO 를 구현하기 위해 같은 사용자 디렉터리를 사용할 필요는 없습니다. 그러나 복제된 사용자 디렉터리를 복제되지 않은 정책 저장소와 함께 사용하는 경우 모든 정책 저장소에 대해 사용자 디렉터리의 이름을 *동일하게* 지정하십시오. 또한 세션 티켓을 암호화하는 세션 티켓 키가 SSO 환경의 모든 키 저장소에 대해 동일해야 합니다. 세션 티켓은 사용자 세션의 유효 기간을 결정합니다.

싱글 사인온 및 인증 체계 보호 수준

싱글 사인온이 구성되어 있으면 한 영역에서 인증된 사용자가 재인증 없이 다른 영역의 리소스에 액세스할 수 있습니다. 단, 두 번째 영역이 첫 번째 영역보다 낮거나 같은 보호 수준이 할당된 인증 체계로 보호되어야 합니다. 사용자가 보호 수준이 높은 인증 체계로 보호되는 리소스에 액세스하려고 하면 SiteMinder 에서 자격 증명을 다시 입력하라는 메시지가 표시됩니다.

관리자는 SiteMinder 에서 관리 UI 를 사용하여 인증 체계에 보호 수준을 할당할 수 있습니다. 보호 수준의 범위는 1~20 이며 1 이 가장 낮은 수준, 20 이 가장 높은 수준을 나타냅니다. 이러한 보호 수준은 관리자가 인증 체계를 구현할 때 싱글 사인온 환경의 보안 및 유연성을 나타내는 기준이 될 수 있습니다.

예를 들어 모든 사용자가 이용할 수 있는 리소스 집합은 보호 수준 1 이 할당된 기본 인증 체계를 사용하고 회사 임원만 이용할 수 있는 다른 리소스 집합은 보호 수준 15 가 할당된 X.509 인증서 체계를 사용합니다. 기본 인증 체계로 인증된 사용자가 인증서 체계로 보호되는 리소스에 액세스하려고 하면 다시 인증해야 합니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

싱글 사인온 및 에이전트 키 관리

웹 에이전트는 웹 에이전트 간에 정보를 전달하는 쿠키를 암호화거나 암호를 해독할 때 키를 사용합니다. 에이전트는 SiteMinder 쿠키를 받으면 키를 사용하여 쿠키 내용의 암호를 해독할 수 있습니다. 정책 서버와 통신하는 모든 웹 에이전트에 대해 키가 동일한 값으로 설정되어 있어야 합니다.

키 보안을 유지하기 위해 정책 서버는 이러한 키를 생성하여 암호화한 후 SiteMinder 환경의 모든 웹 에이전트에 키를 배포할 수 있습니다. 키 변경을 자동화하면 에이전트 키를 쉽게 관리할 수 있으므로 모든 키 정보가 포함된 단일 키 저장소를 공유하는 큰 규모의 SiteMinder 설치 환경을 구현할 수 있습니다. 또한 키 변경을 자동화하면 키의 무결성을 유지할 수 있습니다.

싱글 사인온을 구성하는 방법

싱글 사인온 환경을 설정하려면 다음 단계를 수행하십시오.

1. 싱글 사인온 환경에 포함할 쿠키 도메인을 결정합니다.
2. 1 단계의 싱글 사인온 환경 내에서 쿠키 공급자 도메인으로 사용할 쿠키 도메인을 선택합니다.
3. 중앙 구성을 사용하는 에이전트의 경우 관리 UI 를 사용하여 에이전트 구성 개체를 엽니다. 로컬 구성을 사용하는 에이전트의 경우에는 각 웹 서버에서 웹 에이전트 구성 파일을 엽니다.
4. 쿠키 공급자 에이전트에 대해 다음 단계를 수행하여 구성 매개 변수를 수정합니다.
   a. 보안 향상을 위해 쿠키 공급자 기능을 제한합니다 (페이지 252).
   b. 쿠키 공급자 재생 공격을 방지합니다 (페이지 254).
   c. RequireCookies 매개 변수의 값이 yes 인지 확인합니다 (페이지 255).
   d. (선택 사항) 구성된 세션 시간이 만료될 때까지 쿠키를 유효하게 유지하려면 영구 쿠키를 사용합니다 (페이지 256). 영구 쿠키가 없으면 브라우저 쿠키는 일시적입니다. 임시 쿠키는 *하나의* 브라우저 세션에만 유효합니다.
   e. CookieDomain 매개 변수에 지정된 값이 에이전트가 설치된 시스템의 로컬 쿠키 도메인인지 확인합니다 (페이지 257).
   f. IP 주소의 유효성을 검사하려면 다음 매개 변수 중 *하나*를 설정합니다. (페이지 258)
      - 영구 쿠키를 사용하는 경우 PersistentIPCheck 매개 변수를 설정합니다.
      - 임시 쿠키를 사용하는 경우 TransientIPCheck 매개 변수를 설정합니다.
5. (선택 사항) 다음과 같은 싱글 사인온 매개 변수 설정을 수정합니다.
   - 세션 업데이트 간격 수정 (페이지 123)
   - 여러 도메인에 보안 쿠키 설정 (페이지 260)
   - 보호되지 않은 리소스에 대해 쿠키 공급자 무시 (페이지 261)
   - POST 요청에 대해 쿠키 공급자 무시 (페이지 262)
   - 싱글 사인온에 보안 URL 구성 (페이지 263)

6. SSO 환경에서 쿠키 공급자가 아닌 다른 모든 에이전트에 대해 다음 단계를 수행하여 구성 매개 변수를 설정합니다.

   a. CookieProvider 매개 변수의 값을 쿠키 공급자 도메인의 이름으로 설정합니다. 쿠키 공급자 에이전트를 호스트하는 웹 서버의 정규화된 도메인 이름을 사용하십시오 (페이지 264).

      다음과 같은 구문을 사용하십시오.

      http://*server.example.com:port*/siteminderagent/SmMakeCookie.ccc

      참고: 이 예제에서 알 수 있듯이 쿠키 공급자 이름의 확장명은 .ccc 여야 합니다.

   b. 보안 향상을 위해 쿠키 공급자 기능이 사용되지 않도록 설정합니다 (페이지 265).

7. 에이전트 구성 파일을 수정하여 매개 변수를 편집한 경우 웹 서버를 다시 시작하여 변경 내용을 적용합니다.

쿠키 공급자 기능 제한

모든 에이전트의 쿠키 공급자 기능은 기본적으로 사용되도록 설정됩니다. 도용 당한 SiteMinder SSO 쿠키를 사용하는 권한이 없는 사용자가 쿠키 공급자를 악용하고 한 도메인의 세션 쿠키를 사용하여 다른 쿠키 도메인의 세션 쿠키를 위조할 수 있습니다. 위조된 세션 쿠키는 보호된 SSO 도메인에 무단 액세스를 허용할 수도 있습니다.

다음 매개 변수를 사용하면 도용 당한 SSO 쿠키로 쿠키 공급자를 악용하고 세션 쿠키를 위조할 가능성을 없앨 수 있습니다.

LimitCookieProvider

쿠키 공급자 역할을 하는 SiteMinder 에이전트에서 쿠키 공급자 SET 요청(.ccc 리소스)을 처리하는 방식을 지정합니다. 이 매개 변수의 값이 yes 인 경우에는 쿠키 공급자 도메인에 쿠키가 없으면 SET 요청이 무시됩니다. 쿠키 공급자는 새 쿠키를 설정하지 않고 사용자를 TARGET URL 로 리디렉션합니다. 이 매개 변수의 값이 no 인 경우에는 SET 요청이 처리되고 TARGET URL 로 리디렉션하는 동안 항상 새 쿠키가 설정됩니다.

기본값: No

기본값: (smpolicy-secure.xml 을 사용하여 정책 저장소를 생성한 후) Yes

쿠키 공급자로 사용되는 에이전트와 SSO 환경에서 실행되는 다른 에이전트에 대해 최적의 보안을 유지하려면 특정 구성이 필요할 수도 있습니다.

예를 들어 SSO 환경에 세 개의 도메인이 포함된 경우를 가정해 봅니다. example.com 에 쿠키 공급자가 있고 example.org 와 example.net 이라는 SSO 도메인이 두 개 있습니다. 다음 표에서는 각 도메인에 대한 에이전트 구성 설정을 설명합니다.

Example.com(쿠키 공급자 도메인)	**Example.org(SSO 쿠키 도메인)**	**Example.net(SSO 쿠키 도메인)**
CCCExt = .ccc	**CookieProvider** = http://server1.example.com:80/siteminderagent/SmMakeCookie.ccc	**CookieProvider** = http://server1.example.com:80/siteminderagent/SmMakeCookie.ccc
IgnoreExt = (확장명 목록에 .ccc 가 포함되는지 확인)	**CCCExt** = .ccc	**CCCExt** = .ccc
EnableCookieProvider = yes	**IgnoreExt** = (확장명 목록에 .ccc 가 포함되는지 확인)	**IgnoreExt** = (확장명 목록에 .ccc 가 포함되는지 확인)
LimitCookieProvider = yes	**EnableCookieProvider** = no	**EnableCookieProvider** = no
TracksSessionDomain = yes	**TracksSessionDomain** = yes	**TracksSessionDomain** = yes
TrackCPSessionDomain = yes		

쿠키 공급자 재생 공격 방지

다음 매개 변수를 사용하면 쿠키 공급자가 재생 공격에 노출되지 않도록 할 수 있습니다.

TrackCPSessionDomain

세션 쿠키의 쿠키 도메인이 쿠키 공급자의 쿠키 도메인과 일치하는지 확인합니다. 쿠키 도메인이 여러 개이면 재생 공격이 발생할 가능성이 있습니다.

기본값: No(쿠키 공급자의 도메인이 확인되지 않음)

쿠키 공급자 재생 공격을 방지하려면 TrackCPSessionDomain 매개 변수의 값을 yes 로 설정하십시오.

에이전트는 쿠키 도메인을 비교하여 도메인이 일치하지 않으면 요청을 거부합니다.

싱글 사인온을 위한 RequireCookies 매개 변수 설정

다음 매개 변수를 사용하여 SiteMinder 에 쿠키가 필요한지 여부를 제어할 수 있습니다.

RequireCookies

SiteMinder 에 쿠키가 필요한지 여부를 지정합니다. 다음과 같은 기능을 수행하려면 SiteMinder 에 쿠키가 필요합니다.

- 싱글 사인온 환경의 보안 설정
- 세션 시간 만료 적용
- 유휴 시간 만료 적용

이 매개 변수의 값이 yes 인 경우 에이전트에서 HTTP 요청을 처리하려면 다음 쿠키 중 하나가 필요합니다.

- SMCHALLENGE
- SMSESSION

이 매개 변수의 값이 no 인 경우 다음과 같은 상황이 발생할 수 있습니다.

- 예기치 않게 사용자에게 자격 증명이 요청됩니다.
- 시간 만료가 엄격하게 적용되지 않습니다.

중요! 에이전트에 쿠키가 필요한 경우 사용자에게 자신의 브라우저에서 HTTP 쿠키를 수락하도록 지시하십시오. 그렇지 않으면 사용자는 보호된 모든 리소스에 대해 액세스가 거부됩니다.

기본값: Yes

쿠키를 요청하려면 RequireCookies 매개 변수의 값을 yes 로 설정하십시오.

싱글 사인온을 위한 영구 쿠키 사용

여러 개의 브라우저 세션에 싱글 사인온을 사용하려면 영구 쿠키를 사용합니다. 다음 단계에서는 영구 쿠키를 사용하는 한 가지 경우를 보여 줍니다.

1. 사용자가 SiteMinder 에 인증하지만 SiteMinder 세션이 만료되기 전에 브라우저 세션을 끝냅니다.
2. 사용자가 나중에 새 브라우저 세션을 시작하지만 영구 쿠키가 싱글 사인온 기능을 유지 관리합니다.

영구 쿠키는 구성된 최대 세션 만료 시간 *이후* 7 일 동안 유효합니다. 대부분의 브라우저는 쿠키 만료 후 웹 브라우저의 쿠키 파일을 삭제합니다. 일부 브라우저의 경우 영구 쿠키를 다르게 처리할 수 있습니다.

다음 단계를 수행하십시오.

1. PersistentCookies 매개 변수를 yes 로 설정합니다.

   SMSESSION 은 영구 쿠키입니다.
2. TransientIDCookies 매개 변수를 no 로 설정합니다.

   SMIDENTITY 은 영구 쿠키입니다.

쿠키 도메인 지정

CookieDomain 매개 변수는 에이전트를 설치한 서버의 쿠키 도메인을 정의합니다. 다음 매개 변수를 설정하여 도메인을 수정할 수 있습니다.

CookieDomain

에이전트의 쿠키 도메인을 정의합니다. 마침표가 두 개 이상 포함된 정규화된 도메인 이름을 사용해야 합니다. 예를 들어 쿠키 도메인을 .example.com 으로 설정하면 다음과 같은 서버가 검색됩니다.

- w1.example.com
- w2.example.com
- w3.sales.example.com

이 도메인의 모든 웹 서버는 브라우저와 쿠키를 교환할 수 있습니다. 같은 쿠키 도메인의 서버는 쿠키를 사용하여 사용자 자격 증명을 확인합니다.

이 매개 변수 값이 none 이면 에이전트는 자신의 서버에 대한 쿠키만 생성합니다. 예를 들면 myserver.example.com 입니다.

값이 비어 있거나 로컬 구성 파일에 ""가 포함되어 있으면 에이전트는 HTTP_HOST 헤더의 도메인 정보를 사용합니다. 그런 다음 CookieDomainScope 매개 변수의 설정을 사용하여 기본값을 지정합니다.

기본값: 비어 있음

예: .example.com

제한: 이 값은 대/소문자를 구분합니다. 위의 예와 같이 이 값에는 마침표가 두 개 이상 포함된 정규화된 도메인 이름을 사용해야 합니다.

참고: 이 값은 대/소문자를 구분합니다.

다음 단계를 수행하십시오.

1. CookieDomain 매개 변수의 값을 설정합니다.
2. (선택 사항) CookieDomainScope 매개 변수의 값을 설정합니다.

CookieDomainScope

도메인 이름의 섹션(마침표로 구분되는 문자) 수를 지정합니다.

이 값이 0(기본값)으로 설정되면 에이전트는 서버 전용 쿠키를 생성하지 않고 호스트별 쿠키 도메인을 선택합니다. 즉, myserver.example.com 이라는 쿠키 도메인 대신 example.com 도메인이 사용되고 myserver.metals.example.org 대신 .metals.example.org 도메인이 사용됩니다.

CookieDomainScope 매개 변수가 2 로 설정되면 쿠키 도메인은 각각 .example.com 과 .example.org 가 됩니다.

기본값: 0

예: 쿠키 도메인이 division.example.com 인 경우를 가정해 봅니다. server.division.example.com 에 대해 쿠키 도메인 범위를 설정하려면 CookieDomainScope 매개 변수의 값을 3 으로 설정합니다.

싱글 사인온 환경을 위한 IP 주소 유효성 검사 설정

인증되지 않은 시스템에서 패킷을 모니터링하고 쿠키를 도용한 후 해당 쿠키를 사용하여 다른 시스템에 액세스할 수 있습니다. 인증되지 않은 시스템에 의해 보안이 침해되지 않게 하려면 영구 쿠키 및 임시 쿠키를 사용하여 IP 검사 기능을 설정하거나 해제할 수 있습니다.

IP 검사 기능을 사용하는 경우 에이전트는 마지막 요청의 쿠키에 저장된 IP 주소를 현재 요청에 포함된 IP 주소와 비교해야 합니다. 두 IP 주소가 일치하지 않으면 에이전트에서 요청을 거부합니다.

IP 검사를 구현하는 데 사용되는 두 매개 변수는 PersistentIPCheck 와 TransientIPCheck 입니다. 다음과 같이 매개 변수를 설정하십시오.

- PersistentCookies 가 사용되도록 설정한 경우 PersistentIPCheck 를 yes 로 설정합니다.
- PersistentCookies 가 사용되도록 설정하지 않은 경우 TransientIPCheck 를 yes 로 설정합니다.

SiteMinder 아이덴티티 쿠키는 IP 검사의 영향을 받지 않습니다.

세션 업데이트 간격 수정

다음 매개 변수를 사용하면 새 쿠키를 설정하기 위해 웹 에이전트에서 쿠키 공급자로 요청을 리디렉션하는 간격을 지정할 수 있습니다.

SessionUpdatePeriod

웹 에이전트가 새 쿠키를 설정하기 위해 쿠키 공급자에 요청을 리디렉션하는 빈도(초)를 지정합니다. 마스터 쿠키를 새로 고치면 SiteMinder 세션의 유휴 시간 만료로 인해 쿠키가 만료될 가능성이 줄어듭니다.

기본값: 60

세션 업데이트 간격을 수정하려면

1. CookieProvider 매개 변수가 정의되어 있는지 확인합니다.
2. SessionUpdatePeriod 매개 변수의 값을 원하는 간격(초)으로 변경합니다.

   세션 업데이트 간격이 변경됩니다.

여러 도메인에 보안 쿠키 설정

UseSecureCookies 매개 변수를 설정하면 HTTPS 보안 연결이 사용되는 경우 웹 에이전트가 요청 브라우저 세션에 *로컬* 쿠키만 반환하도록 구성됩니다. 웹 에이전트가 쿠키 공급자로 구성된 경우에는 다른 쿠키 도메인의 리소스에 액세스하기 위해 리디렉션된 요청에 UseSecureCookies 매개 변수가 적용되지 않습니다.

보안 쿠키를 사용하도록 구성된 웹 에이전트를 쿠키 공급자로 구성하여 다른 쿠키 도메인의 웹 에이전트에 쿠키를 반환할 수 있게 하려면 UseSecureCookies 가 사용되도록 설정하고 다음 매개 변수도 구성해야 합니다.

UseSecureCPCookies

UseSecureCPCookies 를 Yes 로 설정하면 쿠키 공급자가 역시 UseSecureCookies 가 설정되어 보안 쿠키를 사용하도록 구성된 다른 쿠키 도메인의 웹 에이전트로만 쿠키를 보냅니다.

이 설정과 UseSecureCookies 가 둘 다 설정된 경우 여러 도메인 싱글 사인온 환경의 사용자는 SSL 웹 서버에서 다른 쿠키 도메인의 비 SSL 웹 서버로 이동할 때 다시 인증해야 합니다. 기존 HTTP 연결을 통해 보안 쿠키를 전달할 수 없습니다.

기본값: No

SSL 연결을 통해 여러 도메인에 쿠키를 전송하려면 쿠키 공급자에서 UseSecureCookies 및 UseSecureCPCookies 를 yes 로 설정하십시오.

추가 정보:

보안 쿠키 설정 (페이지 101)

보호되지 않은 리소스에 대해 쿠키 공급자 무시

기본적으로 에이전트는 모든 요청을 쿠키 공급자에 전달합니다. 보호되지 않은 리소스가 있는 경우 다음 매개 변수를 사용하여 네트워크 트래픽을 줄일 수 있습니다.

IgnoreCPForNotprotected

보호되지 않은 리소스 요청에 대해 쿠키 공급자가 쿼리되지 않도록 합니다. 이 매개 변수가 no 로 설정되면 웹 에이전트는 모든 요청을 쿠키 공급자에 전달합니다. 프레임워크가 아닌 기존 에이전트의 경우 이 매개 변수의 값이 웹 에이전트 로그 파일에 나타나도록 쿠키 공급자를 구성하십시오.

기본값: No

보호되지 않은 리소스가 요청될 경우 에이전트가 쿠키 공급자에 연결되지 않게 하려면 IgnoreCPForNotprotected 매개 변수의 값을 yes 로 설정합니다.

POST 요청에 대해 쿠키 공급자 무시

다음과 같이 매개 변수를 사용하여 해당 동작을 사용하도록 설정할 수 있습니다.

- 이 설정은 기존 에이전트가 쿠키 공급자의 역할을 할 수 있도록 합니다.
- 이 설정은 모든 환경에서 POST 요청을 쿠키 공급자에게 보내지 않도록 합니다.

LegacyCookieProvider

에이전트가 POST 요청을 쿠키 공급자에 보낼지 여부를 제어합니다. 에이전트가 쿠키 공급자로 작동하는 기존 에이전트에 POST 요청을 보내면 리디렉션된 요청이 GET 이 됩니다. 이 변환으로 인해 오류가 발생합니다. no 로 설정하면 에이전트가 POST 요청을 쿠키 공급자에 보냅니다. yes 로 설정하면 에이전트가 POST 요청을 쿠키 공급자에 보내지 *않습니다*.

중앙 에이전트 구성을 사용하는 경우 이 매개 변수를 에이전트 구성 개체에 추가하십시오. 이 매개 변수는 로컬 구성 파일에 있습니다.

기본값: No(POST 요청이 전송됨)

다음 동작을 사용하도록 설정하려면 LegacyCookieProvider 매개 변수의 값을 yes 로 설정하십시오.

- 기존 에이전트를 쿠키 공급자로 사용
- 쿠키 공급자에게 POST 요청을 보내지 않음

싱글 사인온에 SecureUrls 구성

싱글 사인온 네트워크에 SecureUrls 기능을 지원하는 웹 에이전트와 지원하지 않는 에이전트가 둘 다 있는 경우 사용자가 보호된 싱글 사인온 리소스를 요청하면 내부 서버 오류 메시지가 나타날 수 있습니다.

SecureUrls 기능을 지원하는 웹 에이전트의 로그에서 다음과 같이 서버 오류가 발생한 원인을 보여 줍니다.

> Error. Unable to process request, SecureUrls is disabled. (오류. SecureUrls 가 지원되지 않으므로 요청을 처리할 수 없습니다.)

참고: 싱글 사인온 환경에 있는 모든 웹 에이전트의 SecureUrls 매개 변수는 같은 값으로 설정되어야 합니다. SiteMinder 에서는 SecureUrls 매개 변수가 서로 다른 값으로 설정된 웹 에이전트 간의 상호 운용성을 지원하지 않습니다.

쿠키 공급자 지정

SSO 환경의 다른 에이전트가 쿠키 공급자를 사용하게 하려면 다음 매개 변수를 사용하여 쿠키 공급자 에이전트의 위치를 지정합니다.

CookieProvider

쿠키 공급자 에이전트가 있는 웹 서버의 URL 을 지정합니다.

쿠키 공급자는 싱글 사인온 환경의 에이전트입니다. 쿠키 공급자는 자신이 위치한 로컬 도메인의 브라우저 쿠키를 설정합니다. 이 쿠키가 설정되면 사용자는 재인증하지 않고 싱글 사인온 환경을 탐색할 수 있습니다.

다음 예제와 같이 쿠키 공급자 이름의 확장명은 .ccc 여야 합니다.

- IIS, Oracle iPlanet 및 Domino 웹 서버의 경우 URL 구문은 다음과 같습니다.

  http://*server.domain:port*/siteminderagent/SmMakeCookie.ccc

- Apache 및 Apache 기반 웹 서버의 경우 URL 구문은 다음과 같습니다.

  http://*server.domain:port*/SmMakeCookie.ccc

이 매개 변수는 다음 매개 변수에도 영향을 줍니다.

- CCCExt
- SessionUpdatePeriod

기본값: 기본값 없음

예: (IIS, Oracle iPlanet 및 Domino 웹 서버)
http://server1.example.com:80/siteminderagent/SmMakeCookie.ccc

예: (Apache 및 Apache 기반 웹 서버)
http://server1.example.com:80/SmMakeCookie.ccc

제한: 이 매개 변수에는 정규화된 도메인 이름을 사용해야 합니다.

다음 단계를 수행하십시오.

1. CookieProvider 매개 변수를 쿠키 공급자로 사용되는 웹 서버의 URL 로 설정합니다.
2. CCCExt 매개 변수의 값이 .ccc 로 설정되어 있는지 확인합니다.
3. IgnoreExt 매개 변수의 값에 .ccc 확장명을 추가합니다.
4. (선택 사항) 세션 업데이트 간격을 수정합니다.

5. SSO 환경에서 쿠키 공급자가 *아닌* 모든 웹 에이전트에 대해 1~4 단계를 반복합니다.

   쿠키 공급자가 지정됩니다.

쿠키 공급자 사용 안 함

기본적으로 모든 SiteMinder 에이전트는 쿠키 공급자로 사용될 수 있습니다. 이 설정은 SSO 환경을 더 쉽게 구성할 수 있도록 합니다. 보안 향상을 위해 다음 매개 변수를 사용하여 기본 제공되는 쿠키 공급자 기능이 사용되지 않도록 설정할 수 있습니다.

EnableCookieProvider

에이전트가 쿠키 공급자(.ccc)의 요청을 처리하는 방식을 지정합니다. 이 매개 변수 값이 yes 인 경우 에이전트가 요청을 처리합니다. 이 매개 변수 값이 no 인 경우에는 에이전트가 쿠키 공급자의 요청을 무시합니다. 에이전트는 요청된 리소스에 대한 액세스를 거부합니다. 보안을 향상시키려면 이 매개 변수 값을 no 로 설정하십시오.

기본값: Yes

기본값: (smpolicy-secure.xml 을 사용하여 정책 저장소를 생성한 후) No

에이전트가 쿠키 공급자의 요청을 처리하지 않게 하려면 EnableCookieProvider 매개 변수의 값을 no 로 설정합니다.

SSO 환경에 대해 쿠키 공급자를 사용하지 않는 경우 다음 표에 나오는 에이전트 구성 설정을 모든 에이전트에 사용하십시오.

모든 에이전트에 설정할 구성 매개 변수 값
EnableCookieProvider = no

제 15 장: 전체 로그아웃

이 섹션은 다음 항목을 포함하고 있습니다.

전체 로그오프 작동 방식

전체 로그오프 지원을 통해 웹 개발자는 사용자가 사용자 세션에서 완전히 로그오프되도록 할 수 있습니다. 그러면 사용자가 웹 브라우저를 종료하지 않고도 세션을 끝낼 수 있고 권한 없는 사용자가 열려 있는 세션의 제어권을 갖지 못하게 되므로 리소스가 보호됩니다.

전체 로그오프는 다음과 같은 순서로 진행됩니다.

1. 사용자가 로그오프 단추를 클릭합니다.
2. 웹 에이전트가 사용자 지정된 로그오프 페이지로 사용자를 리디렉션합니다.
3. 웹 에이전트가 사용자 브라우저에서 세션 및 인증 쿠키를 제거합니다.
4. 웹 에이전트가 싱글 사인온 환경에 대해 지정된 쿠키 공급자 도메인 및 로컬 쿠키 도메인에서 세션 쿠키를 제거합니다.
5. 웹 에이전트가 정책 서버를 호출하여 세션 정보를 제거하도록 지시합니다.

   사용자가 완전히 로그오프됩니다.

추가 정보:

Domino 에이전트에 대해 전체 로그오프 지원 구성 (페이지 371)

전체 로그오프 구성

전체 로그오프 기능은 다음 매개 변수로 생성하는 사용자 지정 로그아웃 페이지를 사용합니다.

LogOffUri

사용자 지정 웹 페이지의 URI 를 지정하여 전체 로그아웃 기능이 사용되도록 설정합니다. 이 사용자 지정 웹 페이지는 사용자가 성공적으로 로그오프된 후에 나타납니다. 이 페이지가 브라우저 캐시에 저장되지 않도록 구성하십시오. 그러지 않으면 브라우저에서는 사용자를 로그오프하지 않은 채 캐시의 로그아웃 페이지를 표시할 수 있습니다. 이러한 경우 권한 없는 사용자가 세션에 대한 제어권을 갖게 될 수 있습니다.

참고: CookiePath 매개 변수가 설정된 경우 LogOffUri 매개 변수의 값은 동일한 쿠키 경로를 가리켜야 합니다. 예를 들어 CookiePath 매개 변수의 값이 example.com 으로 설정되어 있으면 LogOffUri 가 example.com/logoff.html 을 가리켜야 합니다.

기본값: (CA SiteMinder Agent for SharePoint r12.0.3.0 을 *제외한* 모든 에이전트) 기본값 없음

제한: 여러 URI 값이 허용됨 정규화된 URL 을 사용하지 *마십시오. 상대* URI 를 사용하십시오.

예: (CA SiteMinder Agent for SharePoint r12.0.3.0 을 *제외한* 모든 에이전트) /Web pages/logoff.html

다음 단계를 수행하십시오.

1. 사용자를 로그오프하는 사용자 지정 HTTP 응용 프로그램을 생성합니다. 예를 들어 사용자를 지정된 URL 로 리디렉션하는 "Exit"(종료) 또는 "Sign Off"(로그오프) 단추를 추가합니다.
2. 웹 브라우저에서 캐시되지 않도록 로그아웃 페이지를 설정합니다. 이렇게 설정하면 페이지가 항상 브라우저의 캐시가 아니라 웹 서버에서 제공되므로 보안이 향상됩니다. 예를 들어 다음과 같은 메타 태그를 HTML 페이지에 추가할 수 있습니다.

   ```
   <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
   <META HTTP-EQUIV="Expires" CONTENT="-1">
   ```

   중요! 일부 웹 브라우저에서는 메타 태그를 지원하지 *않습니다.* cache-control HTTP 헤더를 대신 사용하십시오.
3. 다음 단계를 수행하여 LogOffUri 매개 변수를 구성합니다.

a. 필요한 경우 파운드 기호(#)를 삭제합니다.

b. 사용자를 로그오프할 사용자 지정 HTTP 파일의 URI 를 입력합니다. 정규화된 URL 을 사용하*면 안 됩니다*.

전체 로그아웃 기능이 구성되었습니다.

추가 정보:

에이전트 쿠키의 쿠키 경로 지정 (페이지 104)

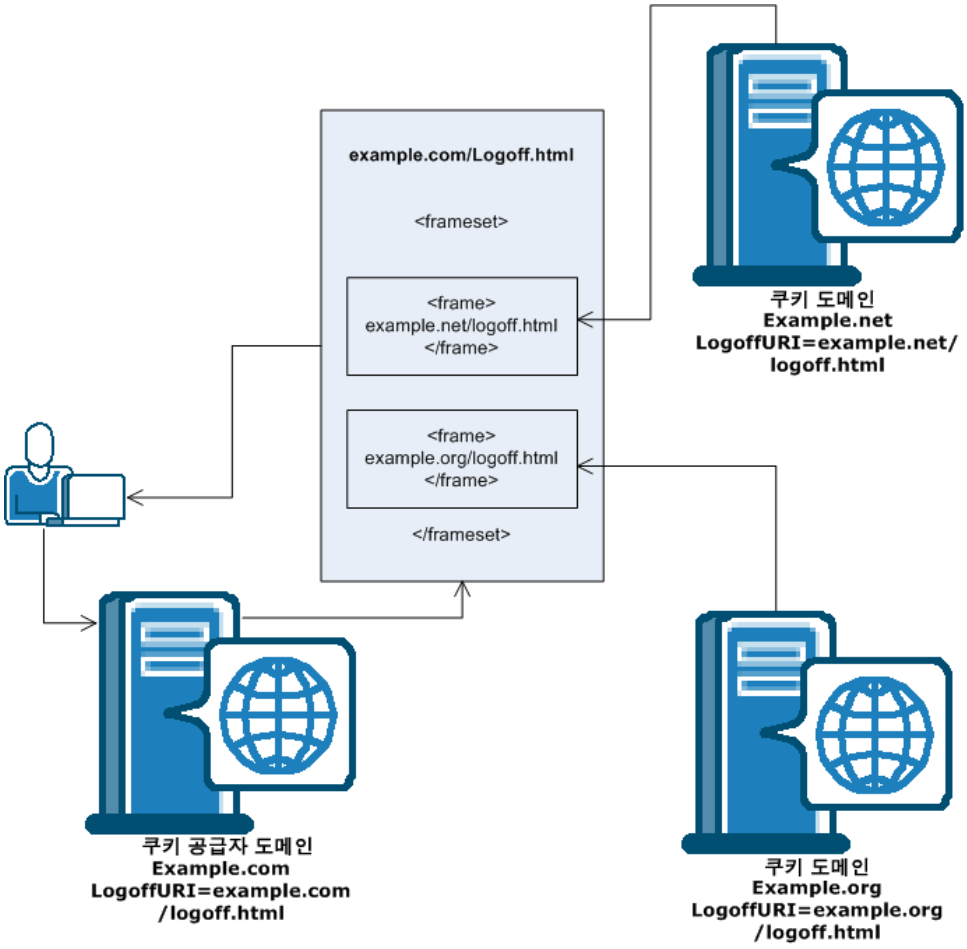
싱글 사인온에 대해 전체 로그오프를 구성하는 방법

싱글 사인온 환경에서는 웹 에이전트가 연결된 쿠키 공급자 도메인과 로컬 쿠키 도메인에서만 세션 쿠키가 제거됩니다. 여러 쿠키 도메인에 대해 싱글 사인온이 구성된 경우 SiteMinder 의 전체 로그오프 기능은 사용자가 방문한 모든 쿠키 도메인에서 사용자를 자동으로 로그오프하지 *않습니다*.

여러 쿠키 도메인에 대해 로그오프를 구성하려면 다음을 수행하십시오.

1. SSO 환경의 다른 쿠키 도메인에 대한 개별 프레임 또는 iframe 을 포함하는 중앙 로그오프 페이지 하나를 생성합니다. 이러한 프레임은 1x1 픽셀과 같이 작은 크기일 수 있습니다.
2. 1 단계에서 생성한 중앙 로그오프 페이지의 각 프레임에 대해 연결된 쿠키 도메인의 로그오프 URI 로 이동하기 위한 하이퍼링크를 추가합니다. 예를 들어 example.org 와 example.net 이라는 두 개의 쿠키 도메인이 있는 경우 다음과 같이 할 수 있습니다.
   - 한 프레임에는 example.org 의 로그오프 URI 로 연결되는 하이퍼링크를 추가합니다.
   - 다른 프레임에는 example.net 의 로그오프 URI 로 연결되는 하이퍼링크를 추가합니다.
3. 쿠키 공급자 도메인의 LogoffUri 가 중앙 로그오프 페이지를 가리키도록 구성합니다. 웹 서버가 이 로그오프 페이지를 로드하는 경우 중앙 로그오프 페이지의 프레임은 다른 쿠키 도메인에서 로그오프 페이지를 호출합니다. 그러면 사용자가 모든 쿠키 도메인에서 동시에 로그오프됩니다.

다음 그림에서는 중앙 로그오프 페이지가 사용되는 방식을 보여 줍니다.



참고: 하이퍼링크를 <frame> 태그 대신 <iframe> 태그에 추가할 수도 있습니다.

FCC 양식을 사용하여 전체 로그아웃 구성

FCC 양식을 사용하여 사용자를 인증하는 경우 이 양식을 사용하여 전체 로그아웃 기능을 구성할 수 있습니다. 이 방법은 LogoffUri 매개 변수 대신 사용될 수 있습니다.

다음 단계를 수행하십시오.

1. 텍스트 편집기에서 사용자 인증에 사용되는 .fcc 파일을 엽니다. FCC 파일은 다음 디렉터리에 있습니다.

   web_agent_home/samples/forms

   web_agent_home

   SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

   기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

   기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

   기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

   기본값(UNIX/Linux 시스템): /opt/ca/webagent

2. FCC 페이지의 맨 위에서 <_html> 태그 앞에 다음 텍스트를 추가합니다.

   ```
   @smlogout=true
   @target=http://server_name.example.com/directory/your_logout_page.html
   ```

   참고: *your_logout_page* 는 사용자에게 로그아웃되었음을 알리기 위해 생성하는 사용자 지정 html 페이지를 나타냅니다.

   FCC 양식을 사용하여 전체 로그아웃이 구성됩니다.

제 16 장: SSO 보안 영역

이 섹션은 다음 항목을 포함하고 있습니다.

보안 영역 개요

사용자는 단일 영역을 나타내는 같은 쿠키 도메인 내에 또는 서로 다른 영역을 나타내는 여러 쿠키 도메인에 대해 싱글 사인온 보안 영역을 정의할 수 있습니다. 결과적으로 사용자는 같은 영역 내에서 싱글 사인온 기능을 사용하지만 다른 영역에 들어갈 때는 영역 간에 정의된 트러스트 관계에 따라 재인증이 필요할 수 있습니다. 트러스트 관계에 포함된 영역은 해당 그룹의 영역에서 유효한 세션을 가진 사용자에게 재인증을 요청하지 않습니다.

싱글 사인온 보안 영역은 SiteMinder 웹 에이전트에 의해 완전히 구현됩니다. 각 영역은 독립된 웹 에이전트 인스턴스에 있어야 하며 같은 에이전트 인스턴스에 영역을 여러 개 생성할 수 없습니다.

보안 영역은 웹 에이전트에서 생성한 쿠키로 식별됩니다. 기본적으로 웹 에이전트는 두 개의 쿠키를 생성하는데, 하나는 SMSESSION 이라는 세션 쿠키이고 다른 하나는 SMIDENTITY 라는 아이덴티티 쿠키입니다. 보안 영역을 구성하는 경우 웹 에이전트에서 쿠키 이름에 영역 가맹이 반영되도록 고유 이름을 사용하여 세션 쿠키와 아이덴티티 쿠키를 생성할 수 있습니다.

보안 영역 정의

다음은 SSO(싱글 사인온) 보안 영역에 적용되는 용어입니다.

CAC(중앙 에이전트 구성)

웹 에이전트가 정책 저장소에 정의된 웹 에이전트 구성 개체에서 구성 속성을 빌리는 메커니즘을 식별합니다.

쿠키 공급자

여러 도메인의 웹 에이전트에서 싱글 사인온이 구현되는 메커니즘을 식별합니다. 한 도메인이 마스터 도메인으로 지정되고 다른 도메인의 웹 에이전트는 마스터 도메인의 웹 에이전트로 리디렉션되어 마스터 도메인의 쿠키가 제공됩니다.

SSO(싱글 사인온)

한 번 인증된 사용자에게 자격 증명을 다시 요청하지 않는 메커니즘을 식별합니다.

SSO 영역

단일 쿠키 도메인 내에서 응용 프로그램 SSO 를 분할하는 데 사용되는 임의의 식별자(영역 이름)로 정의된 SSO 하위 집합을 식별합니다. 같은 SSO 영역의 모든 응용 프로그램에서는 SSO 가 허용됩니다. 서로 다른 SSO 영역 간에는 영역 트러스트 관계의 정의에 따라 SSO 허용 여부가 결정됩니다.

트러스트된 SSO 영역

SSO 에 대해 로컬 영역에서 트러스트되는 외부 영역을 식별합니다.

보안 영역의 이점

SSO 보안 영역은 SiteMinder 관리자가 하나의 쿠키 도메인 내에서 싱글 사인온 환경을 분할하려는 경우 사용할 수 있는 기능입니다. 예를 들어 CA.COM 도메인을 가정해 봅니다. SiteMinder 의 표준 SSO 기능을 사용하면 CA.COM 에서 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 SiteMinder 에서 인증 요청을 받고 SiteMinder 에 로그인하여 SMSESSION 쿠키를 생성합니다.
2. 사용자가 다른 응용 프로그램 APP2 에 액세스하면 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.

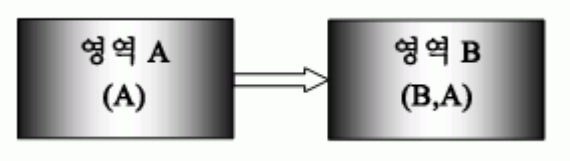
SSO 보안 영역을 사용하면 APP1 을 Z1 영역에 두고 APP2 를 Z2 영역에 둘 수 있습니다. 그런 다음 APP1 에 로그인하면 Z1SESSION 쿠키가 생성되고 APP2 에 액세스하면 Z2SESSION 쿠키가 생성됩니다. 쿠키 이름이 서로 다르므로 다른 쿠키를 덮어쓰지 않게 되고 위의 예제와 같이 사용자가 다른 응용 프로그램으로 이동할 때마다 로그인하지 않고 응용 프로그램별로 한 번만 로그인하면 됩니다.

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 영역의 사용자 세션

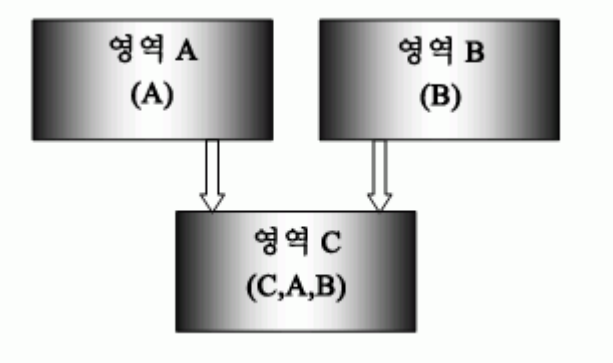
싱글 사인온 보안 영역이 모두 같은 도메인에 속할 필요는 없습니다. 실제로 영역을 여러 도메인에 나눌 수 있습니다. 그러나 웹 에이전트는 트러스트된 영역 쿠키를 검색할 때 로컬 쿠키 도메인만 대상으로 하고, 적절한 쿠키가 발견되지 않으면 자신의 영역에 대한 쿠키 공급자로 리디렉션합니다.

트러스트된 영역 순서

싱글 사인온 보안 영역은 다음과 같은 한 쌍의 매개 변수로 정의됩니다.

- 보안 영역 이름
- 순서가 지정된 트러스트된 영역 목록

트러스트된 영역이 나열되는 순서는 중요합니다. 다음 예제를 참조하십시오.



이 그림에서 영역 C 는 영역 A 와 영역 B 를 모두 트러스트합니다. 영역 A 와 영역 B 는 다른 영역을 트러스트하지 않지만 모든 영역은 자신을 트러스트합니다.

사용자가 영역 C 에서 요청하면 웹 에이전트는 트러스트된 영역 목록에 나열된 순서대로 각 영역에서 세션 쿠키 또는 아이덴티티 쿠키를 검색합니다. 이 예제의 경우 영역 C 의 트러스트된 영역 목록에는 C, A, B 로 순서가 지정되어 있습니다.

다음과 같은 순서대로 이벤트가 발생할 수 있습니다.

1. 웹 에이전트는 먼저 영역 C 에 사용자 세션이 있는지 확인합니다.
2. 세션이 없는 경우 웹 에이전트는 영역 A 에 사용자 세션이 있는지 확인합니다.
3. 세션이 없는 경우 웹 에이전트는 영역 B 에 사용자 세션이 있는지 확인합니다.
4. 검색된 각 쿠키의 세션 사양을 사용하여 로그인이 성공할 때까지 인증 요청을 처리합니다.
5. 인증 성공 후 웹 에이전트는 권한 부여를 진행합니다.
6. 쿠키가 검색되지 않거나 인증이 실패하면 에이전트는 사용자에게 자격 증명을 요청합니다.

영역이 액세스되는 순서에 따라 사용자 환경이 달라질 수 있으므로 주의하십시오. 이 예제에서는 사용자가 먼저 영역 B 에 액세스한 후 영역 C 에 액세스하면 영역 B 의 사용자 아이덴티티가 영역 C 에서도 사용됩니다. 또한 사용자가 먼저 영역 A 에 액세스한 후 영역 B 와 영역 C 에 차례대로 액세스하면 영역 C 로 이동하기 전에 영역 B 에서 사용자에게 재인증이 요청되었지만 영역 A 의 사용자 아이덴티티가 사용됩니다.

또한 최대 세션 만료 시간 및 유휴 세션 만료 시간 값이 다른 세션이 만료되는 경우도 마찬가지입니다. 이 예제에서는 영역 A 와 영역 C 에서 유효한 쿠키를 가진 사용자가 먼저 영역 C 쿠키를 사용하여 액세스합니다. 영역 C 쿠키가 만료되는 경우 영역 A 쿠키가 만료되지 않았으면 대신 사용됩니다. 따라서 자격 증명 챌린지 없이 사용자 아이덴티티가 영역 C 아이덴티티에서 영역 A 아이덴티티로 변경될 수 있습니다.

둘 이상의 웹 에이전트에서 사용하는 트러스트된 영역 목록은 다르지만 트러스트된 영역 이름은 공통으로 사용할 수 있습니다. 이 경우 에이전트는 암시적으로 서로를 트러스트하지만 같은 외부 영역을 트러스트하지 않습니다. 이 기능은 싱글 사인온을 위해 응용 프로그램을 분할할 수 있도록 합니다. 웹 에이전트는 싱글 사인온 영역 이름만 지원합니다. 해당 에이전트에서 생성되는 모든 세션, 아이덴티티 및 상태 쿠키에는 동일한 싱글 사인온 영역 이름이 사용됩니다. 따라서 두 응용 프로그램에서 동일한 싱글 사인온 트러스트 요구 사항을 공유하지 않는 경우에는 각각의 웹 에이전트 및 트러스트된 영역 목록이 있는 개별 웹 서버에서 해당 응용 프로그램이 호스트되어야 합니다.

참고: 외부 영역은 지정된 웹 에이전트에서 지원하는 영역이 아닌 다른 영역을 말합니다. 예를 들어 SSOZoneName="Z1"을 사용하여 에이전트가 구성된 경우 Z1 이 아닌 다른 영역은 모두 외부 영역으로 인식됩니다. 여기에는 기본 영역 "SM"이 포함됩니다.

싱글 사인온 기본 영역 및 트러스트된 영역 목록

SiteMinder 6.x QMR 5 이전 버전의 모든 웹 에이전트와 같이 보안 영역 이름을 지정하지 않는 웹 에이전트는 기본 영역에 속하는 것으로 간주됩니다. 이전 버전과의 호환성을 위해 암시적으로 기본 영역의 이름을 SM 으로 가정합니다. 따라서 SiteMinder 12.52 SP1 웹 에이전트에서 구성 변경 없이 SMSESSION 및 SMIDENTITY 를 기본적으로 지원할 수 있습니다.

트러스트된 영역 목록을 지정하지 않는 웹 에이전트는 고유의 싱글 사인온 영역, 즉 지정된 영역이나 기본 영역(영역 이름이 지정되지 않은 경우)만 트러스트합니다.

기본 영역 외에 다른 영역도 트러스트하도록 웹 에이전트를 구성할 수 있습니다. 또한 지정된 영역 이름만 사용하고 그 외의 다른 트러스트된 영역은 포함하지 않을 수 있습니다. 에이전트는 트러스트된 영역이 지정되어 있는지 여부에 관계없이 항상 고유의 영역을 먼저 트러스트합니다. 기본 영역이 아닌 다른 영역을 사용하는 웹 에이전트가 기본 영역도 트러스트할 수 있게 하려면 트러스트된 영역 목록에 "SM"을 추가해야 합니다.

사용자 세션이 여러 개인 경우 요청 처리

웹 에이전트는 트러스트된 영역의 순서대로 사용자 세션을 찾습니다. 유효한 사용자 세션이 있는 경우 웹 에이전트는 세션 정보를 사용하여 사용자 요청을 처리합니다. 유효한 세션이 없는 경우에는 트러스트 순서에 따라 유효한 다음 사용자 세션을 찾습니다.

확인할 다른 세션이 있으면 실패한 유효성 검사의 응답이 무시됩니다. 그렇지 않은 경우에는 정상적으로 응답이 처리됩니다. 따라서 웹 에이전트가 트러스트된 세션을 세 개 찾은 경우 처음 두 세션에서 유효성 검사가 실패하면 세 번째, 즉 마지막 세션에 대한 유효성 검사의 응답만 처리됩니다.

유효성 검사가 성공할 경우 웹 에이전트는 세션 처리를 중단하고 성공한 유효성 검사의 응답을 처리합니다. 에이전트에 유효성을 검사할 세션이 세 개 있는 경우 첫 번째 세션의 유효성 검사가 성공하면 나머지 두 세션은 무시되고 에이전트는 성공한 첫 번째 유효성 검사에 대한 응답을 처리합니다.

추가 정보

트러스트된 영역 순서 (페이지 277)

영역 간의 전이적 관계

싱글 사인온 영역 간의 트러스트 관계는 완전히 전이적이지 않습니다. 다음 다이어그램에서 볼 수 있듯이 영역 A 가 영역 B 에서 트러스트되고 영역 B 가 영역 C 에서 트러스트되는 경우 영역 A 는 영역 C 에서 트러스트되지 않을 수 있습니다.

싱글 사인온 영역의 영향을 받는 다른 쿠키

SiteMinder 에서는 상태 쿠키를 사용하여 인증 및 권한 부여와 관련된 다양한 이벤트를 관리합니다. 기본적으로 이러한 쿠키는 모두 기본 싱글 사인온 보안 영역 접두사인 SM 으로 시작합니다. 싱글 사인온 영역 이름이 새로 지정되면 쿠키 이름도 비기본 영역 이름을 반영하도록 변경됩니다. 다음은 새 싱글 사인온 영역이 정의되면 영향을 받는 쿠키 목록입니다.

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

예를 들어 Z1 이라는 영역 이름이 지정되면 웹 에이전트는 기본 인증에 사용할 Z1CHALLENGE=YES 쿠키를 생성합니다. 이제 관리자는 에이전트가 서로 충돌하지 않도록 단일 쿠키 도메인(예: ca.com)에 SiteMinder 응용 프로그램 싱글 사인온 구역을 별도로 생성할 수 있습니다. 그러면 싱글 사인온의 트러스트된 영역 목록을 사용하여 이러한 격리된 싱글 사인온 영역 사이에서 싱글 사인온을 정상적으로 구현할 수 있습니다.

싱글 사인온 영역과 권한 부여

싱글 사인온 영역을 사용하면 인증 성공 후 변경 없이 권한 부여 단계가 정상적으로 진행됩니다. 유효성 검사를 통해 유효한 세션이 식별되면 이 세션을 사용하여 나머지 요청을 처리하고 요청에서 식별된 다른 세션은 무시됩니다. 권한 부여가 실패하면 다른 세션에서 권한 부여가 성공하는지 여부를 확인하지 않고 사용자에게 자격 증명이 요청됩니다.

유효성 검사를 처음으로 통과하는 트러스트된 세션이 권한 부여에 전달됩니다. 이 세션에서 권한 부여가 실패하면 사용자에게 자격 증명이 요청됩니다.

보안 영역 구성

두 개의 싱글 사인온 매개 변수가 정책 저장소의 웹 에이전트 구성 개체에 추가되었습니다. 이러한 설정은 로컬 구성 파일에도 사용되며 설치 중에 저장된 샘플 로컬 구성 파일에 추가됩니다.

참고: 같은 에이전트 구성 개체를 통해 구성된 모든 웹 에이전트는 같은 싱글 사인온 영역에 속합니다.

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 설정은 쿠키를 해당 쿠키 도메인에 연결하는 데 도움이 됩니다. 이 매개 변수가 비어 있지 않으면 SiteMinder 가 다음 명명 규칙을 사용하여 쿠키를 생성합니다.

ZonenameCookiename.

기본값: 비어 있음(SM 을 영역 이름으로 사용하며 다음 기본 이름을 쿠키에 제공함)

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

제한: 단일 값 지정. 이 매개 변수는 영어 문자만 지원합니다.

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.

기본값: 비어 있음(SM 또는 제공되었을 경우 SSOZoneName)

제한: 다중값

에이전트에 대한 싱글 사인온 영역 지정

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 설정은 쿠키를 해당 쿠키 도메인에 연결하는 데 도움이 됩니다. 이 매개 변수가 비어 있지 않으면 SiteMinder 가 다음 명명 규칙을 사용하여 쿠키를 생성합니다.

ZonenameCookiename.

기본값: 비어 있음(SM 을 영역 이름으로 사용하며 다음 기본 이름을 쿠키에 제공함)

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

제한: 단일 값 지정. 이 매개 변수는 영어 문자만 지원합니다.

예: 값을 Z1 로 설정하면 다음 쿠키가 생성됩니다.

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

웹 에이전트에서 지원하는 싱글 사인온 영역의 이름을 입력하려면 SSOZoneName 매개 변수를 사용합니다. 이 매개 변수는 대/소문자를 구분합니다. 값을 지정하지 않을 경우 기본값은 SM 입니다. SSOZoneName 매개 변수의 값이 비어 있지 않으면 웹 에이전트에서 다음 명명 규칙을 사용하여 쿠키를 생성합니다.

zone_namecookie_name

여기서 *zone_name* 은 매개 변수 값이고 *cookie_name* 은 생성되는 쿠키의 일반 이름입니다.

다음은 이 규칙의 영향을 받는 쿠키입니다.

- SESSION
- IDENTITY
- DATA
- TRYNO
- CHALLENGE
- ONDENIEDREDIR

사용자 세션이 설정되지 않은 싱글 사인온 영역에서 사용자의 유효성이 확인되면 정책 서버에서 반환된 세션 사양을 사용하여 해당 영역에 대한 새 세션 쿠키가 생성됩니다.

새 쿠키가 생성되는 경우 사용자가 다른 영역의 이름을 변경하여 쿠키를 교체할 수 없도록 쿠키의 영역 매개 변수가 영역 이름으로 설정됩니다. 쿠키 유효성 검사 엔진은 영역 이름이 쿠키 이름에 사용된 접두사와 일치하는지 확인합니다. 이 내용은 SESSION 및 IDENTITY 쿠키에만 적용됩니다.

웹 에이전트에서 지원할 싱글 사인온 영역의 이름을 지정하려면 SSOZoneName 매개 변수에 영역 이름을 추가합니다.

트러스트 순서 및 장애 조치

SSOTrustedZone 매개 변수를 사용하여 싱글 사인온 영역의 트러스트 순서를 지정할 수 있습니다. 요청을 처리할 때 웹 에이전트는 목록에 나타나는 순서대로 각 영역에 대해 SESSION 또는 IDENTITY 쿠키를 검색합니다.

검색된 쿠키는 암호 해독된 후 호스트 이름, 싱글 사인온 영역 이름 및 만료 시간이 올바른지 테스트하여 유효성이 확인되면 트러스트된 세션 목록에 순서대로 저장됩니다. 인증하기 전에는 사용자의 활성 세션, 즉 사용자 아이덴티티가 순서가 지정된 유효한 세션 목록의 첫 번째 세션으로 간주됩니다.

인증하는 동안 웹 에이전트는 목록의 첫 번째 세션을 사용하여 유효성 검사를 호출합니다. 유효성 검사가 성공할 경우 에이전트는 계속해서 사용자 아이덴티티를 설정하고 활성 세션으로 확정합니다. 유효성 검사가 실패할 경우에는 유효성 검사가 성공하거나 확인할 세션이 더 이상 없을 때까지 다음 세션을 사용하여 새 유효성 검사를 호출합니다. 유효한 세션이 없으면 에이전트에서 사용자에게 자격 증명을 요청합니다.

한 세션에서 유효성이 확인되면 에이전트는 다른 모든 세션을 무시하고 해당 세션을 사용하여 나머지 요청을 처리하는 작업만 수행합니다. 따라서 권한 부여가 실패하면 즉시 사용자에게 자격 증명이 요청됩니다. 요청의 다른 기존 세션은 사용되지 않습니다.

제 17 장: 고급 구성 설정

이 섹션은 다음 항목을 포함하고 있습니다.

에이전트 및 프록시 서버

다음 설정을 사용하면 프록시 서버에서 실행 중인 SiteMinder 에이전트를 관리할 수 있습니다.

- 프록시 서버 뒤에 에이전트 구성 (페이지 288)
- cache-control 및 expireforproxy 헤더 설정 사용자 지정 (페이지 290)
- 프록시 헤더 사용 정보 (페이지 293)
- 보안 고려 사항 (페이지 294)

프록시 서버 뒤의 에이전트 구성

웹 에이전트가 프록시 서버 뒤에 설치되는 경우 다음 매개 변수를 사용하여 웹 에이전트를 프록시 서버와 함께 작동하도록 구성할 수 있습니다.

ProxyTrust

프록시 서버의 SiteMinder 에이전트로부터 받은 권한 부여를 트러스트하도록 대상 서버의 에이전트에 지시합니다. 대상 서버는 리버스 프록시 서버 뒤에 있는 서버입니다. 이 값을 yes 로 설정하면 권한 부여를 위해 프록시 서버의 에이전트만 정책 서버에 연결하므로 효율성이 높아집니다. 대상 서버에서 작동하는 에이전트는 사용자 권한을 다시 부여할 때 정책 서버에 연결하지 *않습니다*.

기본값: No

ExpireForProxy

클라이언트에서 콘텐츠(페이지 및 가능한 경우 헤더 또는 쿠키)를 캐시하지 않도록 합니다. 이 매개 변수의 값을 yes 로 설정하면 웹 에이전트가 다음 HTTP 헤더 중 하나를 HTTP 응답에 삽입합니다.

- Expires
- Cache-control

콘텐츠가 캐시되지 않으면 이후의 요청이 계속해서 전달됩니다.

ExpireForProxy 매개 변수를 yes 로 설정하면 웹 에이전트가 수행한 요청의 유형을 기반으로 해당 ProxyHeaders*suffix_name* 매개 변수에 지정된 문자열을 HTTP 응답에 삽입합니다.

HTTP/1.1 요청의 경우 에이전트는 다음 매개 변수의 값을 응답에 헤더로 삽입합니다.

- ProxyHeadersAutoAuth
- ProxyHeadersProtected
- ProxyHeadersUnprotected

HTTP/1.0 요청의 경우 에이전트는 다음 매개 변수의 값을 응답에 헤더로 삽입합니다.

- ProxyHeadersAutoAuth10
- ProxyHeadersProtected10
- ProxyHeadersUnprotected10

기본값: No

참고: 이 매개 변수 이름에 'proxy'라는 단어가 포함되어 있긴 하지만 이 매개 변수의 설정은 웹 브라우저의 동작이나 이 매개 변수 설정을 사용하는 SiteMinder 에이전트가 작동하는 웹 서버에 연결하는 다른 모든 클라이언트의 동작에도 영향을 줍니다.

프록시가 페이지를 캐시하지 않도록 하기 위해 웹 에이전트는 해당 페이지에 대한 Expires 헤더를 추가합니다. 이 헤더는 HTTP 1.0 사양에 지정된 대로 프록시가 페이지를 캐시하지 않도록 하기 위해 과거 날짜로 설정됩니다. 302 리디렉션의 경우 cache-control: no-cache 헤더가 대신 설정됩니다. 이 경우에도 콘텐츠 캐싱을 방지할 수는 있지만 Microsoft 고객지원에 설명된 대로 IE(Internet Explorer) 브라우저의 검색 환경에 부정적인 영향을 줄 수 있습니다.

302 리디렉션에 대해 cache-control: no-cache 를 사용하는 경우 IE 에서 바로 문서 보기를 관리하는 ActiveX 구성 요소는 브라우저의 캐시를 통해 파일을 찾습니다. 이 헤더는 브라우저에 파일을 캐시하지 않도록 지시하므로 ActiveX 구성 요소에서 파일을 찾을 수 없으며 요청을 올바로 표시할 수 없습니다. 또한 웹 에이전트의 ExpireForProxy 를 yes 로 설정하면 백엔드 서버가 프록시에 리소스를 캐시하지 않도록 지시합니다.

프록시 서버 뒤에 에이전트를 구성하려면

1. ProxyTust 매개 변수를 yes 로 설정합니다.
2. ExpireForProxy 매개 변수를 yes 로 설정합니다.
3. (선택 사항) cache-control 및 ExpireForProxy HTTP 헤더의 값을 사용자 지정합니다.

   프록시 서버 뒤에 에이전트가 구성됩니다.

추가 정보:

Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)

Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정

기존에 활성화된 응용 프로그램 파일(.doc, .pdf 등)에 영향을 주지 않고 캐시 제어 및 ExpireForProxy 헤더를 사용자 지정하여 웹 리소스를 보호할 수 있습니다. 다음 유형의 콘텐츠에 대해 특정 HTTP 헤더를 독립적으로 설정하여 웹 브라우저 또는 프록시 서버에서 콘텐츠를 캐시하는 방법을 제어할 수 있습니다.

- Auto-Authorized(자동 권한 부여)
- 보호 안 함
- 보호됨

중요! RFC 2068 에 따라 기본 설정을 변경할 때의 영향에 대해 잘 알고 있지 않다면 기본 설정을 사용하는 것이 좋습니다. 기본 설정을 변경할 계획이라면 사용자 세션이 생성된 후 보호되지 않는 페이지에 액세스할 때 유휴 시간 만료를 추적하기 위해 SiteMinder 세션 쿠키가 업데이트된다는 점에 유의하십시오. 따라서 HTTP 헤더를 캐시하는 프록시에서 보호되지 않는 페이지를 캐시하지 않아야 합니다.

헤더를 설정할 때는 다음과 같은 특성을 적용하여 프록시에서 캐시되지 않도록 하십시오.

- 모든 리디렉션이 캐시 제어를 설정합니다. 에이전트 작업에 관계없이 헤더를 캐시하지 않습니다.
- 웹 서버가 사용된 HTTP 프로토콜(1.0 또는 1.1 이상)을 기반으로 적절한 헤더를 프록시/클라이언트에 다시 보냅니다.

모든 매개 변수는 여러 헤더의 사용에 맞게 cache-control: private 및 cache-control: max-age=60 과 같이 다중값 설정을 사용하여 구성되어야 합니다.

다음과 같이 새 구성을 설정합니다.

1. ProxyHeadersDefaultTime - 기본값 60 초
2. ProxyHeadersTimeoutPercentage – 기본값 10 %
3. 다음과 같은 cache-control 헤더를 사용할 수 있습니다.

   ProxyHeadersAutoAuth

   웹 에이전트 구성에서 ExpireForProxy 매개 변수가 yes 로 설정된 경우 웹 에이전트가 클라이언트로 보낼 HTTP 응답에 삽입하는 HTTP 1.1 헤더의 값을 지정합니다. 이 헤더의 값은 자동으로 권한이 부여된 리소스를 캐시할지 여부 *또는* 그 기간을 결정합니다.

기본값: Expires: Thu, 01 Dec 1994 16:00:00 GMT

예(권장 설정): "Cache-control: max-age=60"

ProxyHeadersAutoAuth10

웹 에이전트 구성에서 ExpireForProxy 매개 변수가 yes 로 설정된 경우 웹 에이전트가 클라이언트로 보낼 HTTP 응답에 삽입하는 HTTP 1.0 헤더의 값을 지정합니다. 이 헤더의 값은 자동으로 권한이 부여된 리소스를 캐시할지 여부 또는 그 기간을 결정합니다.

기본값: Expires: Thu, 01 Dec 1994 16:00:00 GMT

예(권장 설정): "Expires: Thu, 01 Dec 1994 16:00:00 GMT"

ProxyHeadersProtected

웹 에이전트 구성에서 ExpireForProxy 매개 변수가 yes 로 설정된 경우 웹 에이전트가 클라이언트로 보낼 HTTP 응답에 삽입하는 HTTP 1.1 헤더의 값을 지정합니다. 이 헤더의 값은 보호된 리소스를 캐시할지 여부 또는 그 기간을 결정합니다.

기본값: Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

예(권장 설정): "Cache-Control: private"

ProxyHeadersProtected="Cache-Control: max-age=60"

ProxyHeadersProtected10

웹 에이전트 구성에서 ExpireForProxy 매개 변수가 yes 로 설정된 경우 웹 에이전트가 클라이언트로 보낼 HTTP 응답에 삽입하는 HTTP 1.0 헤더의 값을 지정합니다. 이 헤더의 값은 보호된 리소스를 캐시할지 여부 또는 그 기간을 결정합니다.

기본값: Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

예(권장 설정): "Expires: Thu, 01 Dec 1994 16:00:00 GMT"

ProxyHeadersUnprotected

웹 에이전트 구성에서 ExpireForProxy 매개 변수가 yes 로 설정된 경우 웹 에이전트가 클라이언트로 보낼 HTTP 응답에 삽입하는 HTTP 1.1 헤더의 값을 지정합니다. 이 헤더의 값은 보호되지 않은 리소스를 캐시할지 여부 또는 그 기간을 결정합니다.

기본값: Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

예(권장 설정): ProxyHeadersUnprotected="Cache-Control: private"

ProxyHeadersUnprotected="Cache-Control: max-age=60"

ProxyHeadersUnprotected10

웹 에이전트 구성에서 ExpireForProxy 매개 변수가 yes 로 설정된 경우 웹 에이전트가 클라이언트로 보낼 HTTP 응답에 삽입하는 HTTP 1.0 헤더의 값을 지정합니다. 이 헤더의 값은 보호되지 않은 리소스를 캐시할지 여부 또는 그 기간을 결정합니다.

기본값: Expires: Thu, 01 Dec 1994 16:00:00 GMT

Cache-Control: no-cache

예(권장 설정): "Expires: Thu, 01 Dec 1994 16:00:00 GMT"

보호되지 않는 HTTP/1.1 콘텐츠에 대해 제안된 설정의 cache-control 헤더와 같이 여러 헤더를 구성하는 경우 다음 내용에 주의하십시오.

- 구성 매개 변수가 여러 개 포함되어야 하고 각 구성 매개 변수를 쉼표(,) 또는 더하기 기호(+)로 구분할 수 없습니다.
- 이러한 구성 매개 변수의 값은 HTTP 응답 헤더이므로 RFC 2616(HTTP/1.1 의 경우), RFC 1945(HTTP/1.0 의 경우) 및 RFC 822 사양을 준수해야 합니다. HTTP/1.1 과 HTTP/1.0 은 둘 다 HTTP 헤더의 형식을 RFC 822 메시지 형식, 즉 "Name: Value"(이름, 콜론, 공백 및 값이 순서대로 나옴)로 지정합니다.

사용자가 보호되지 않은 리소스에 액세스할 때 적절한 캐시 만료 헤더를 설정하도록 웹 에이전트를 구성하지 *않으면* 기본적으로 웹 에이전트에서 이러한 헤더를 설정하지 않으므로 웹 브라우저 또는 프록시 서버에서 SMSESSION 쿠키를 캐시할 수 있습니다. 캐시된 이 쿠키는 사용자가 다른 세션 및 다른 사용자 컨텍스트를 시작한 후 웹 브라우저나 프록시 서버에 의해 다시 사용될 수 있으므로 권한 없는 가장 문제가 발생할 수 있습니다.

추가 정보:

프록시 서버 뒤의 에이전트 구성 (페이지 288)

프록시 헤더 사용 정보

- 웹 에이전트가 프록시 헤더를 전송하지 않게 하려면 ProxyHeadersUnprotected 값을 비워 둡니다. 예를 들면 다음과 같습니다.

  ProxyHeadersUnprotected=""

  참고: 큰따옴표(")를 나타내려면 작은따옴표(')를 사용하십시오. 웹 에이전트에서 작은따옴표를 자동으로 큰따옴표로 변환합니다.

- %% 또는 이와 같은 값으로 간주되는 %d 가 ProxyHeaders 행에 나타날 수 있습니다. 이 값은 IdleTimeout 과 SessionTimeout 중 더 작은 값에 ProxyHeadersTimeoutPercentage 를 곱한 값으로 대체되거나 만료 시간이 설정되지 않은 경우에는 ProxyHeadersDefaultTime 이 사용됩니다.
- 표준(1.1 이상) 및 HTTP 1.0 헤더의 값이 백엔드 서버에 대한 요청에 맞게 올바로 설정되어야 합니다.
- ExpireForProxy="YES"를 설정하면 쿼리 문자열에 SMSESSION 쿠키를 전달하는 쿠키 공급자 리디렉션이 만료됩니다.

보안 고려 사항

로그아웃 후에도 브라우저 세션을 유지할 수 있으므로 SMSESSION 쿠키를 제거해도 같은 브라우저 세션을 사용하여 이전에 캐시한 파일을 볼 수 있습니다. 프록시 서버는 로그아웃 요청을 인식하지 못하고 보호된 콘텐츠 및 보호되지 않은 콘텐츠를 시간이 만료될 때까지(cache-control: max-age=60) cache-control: private 사용자의 캐시에 유지하기 때문에 이런 문제가 발생합니다. 따라서 이러한 요청은 유효한 SMSESSION 쿠키와 함께 페이지를 반환할 수 있습니다. 보안을 유지하려면 연결 유지 기능이 사용되지 않도록 설정하거나 브라우저를 닫아야 합니다.

또한 로컬 브라우저 캐시는 전체 세션에서 로컬 캐시를 관찰하므로 private/max-age 조합의 영향을 받습니다. 따라서 보호된 리소스의 max-age 시간을 최대한 짧게 설정해야 합니다.

allowcacheheaders="FALSE" 구성 설정(기본값)이 사용되는 경우 if-modified-since 및 if-none-match 요청 헤더를 사용하면 프록시 서버가 이러한 헤더를 검색할 수 있습니다. 따라서 검색된 헤더가 프록시 서버에 따라 요청에 적용됩니다.

다음 항목을 설치하면 이 문제를 해결할 수 있습니다.

- 프록시 서버의 웹 에이전트
- 요청에서 이러한 헤더를 제거하는 다른 필터

HTTP 1.0 또는 HTTP 1.1 이상에서는 다른 헤더를 사용하여 캐싱 프록시에 대한 지침을 지정하므로 연결 유형에 따라 가장 적절하게 처리되도록 이러한 버전을 구성해야 합니다.

에이전트 및 리버스 프록시 서버

리버스 프록시 서버에 배포된 SiteMinder 에이전트를 관리하려면 다음 항목을 참조하십시오.

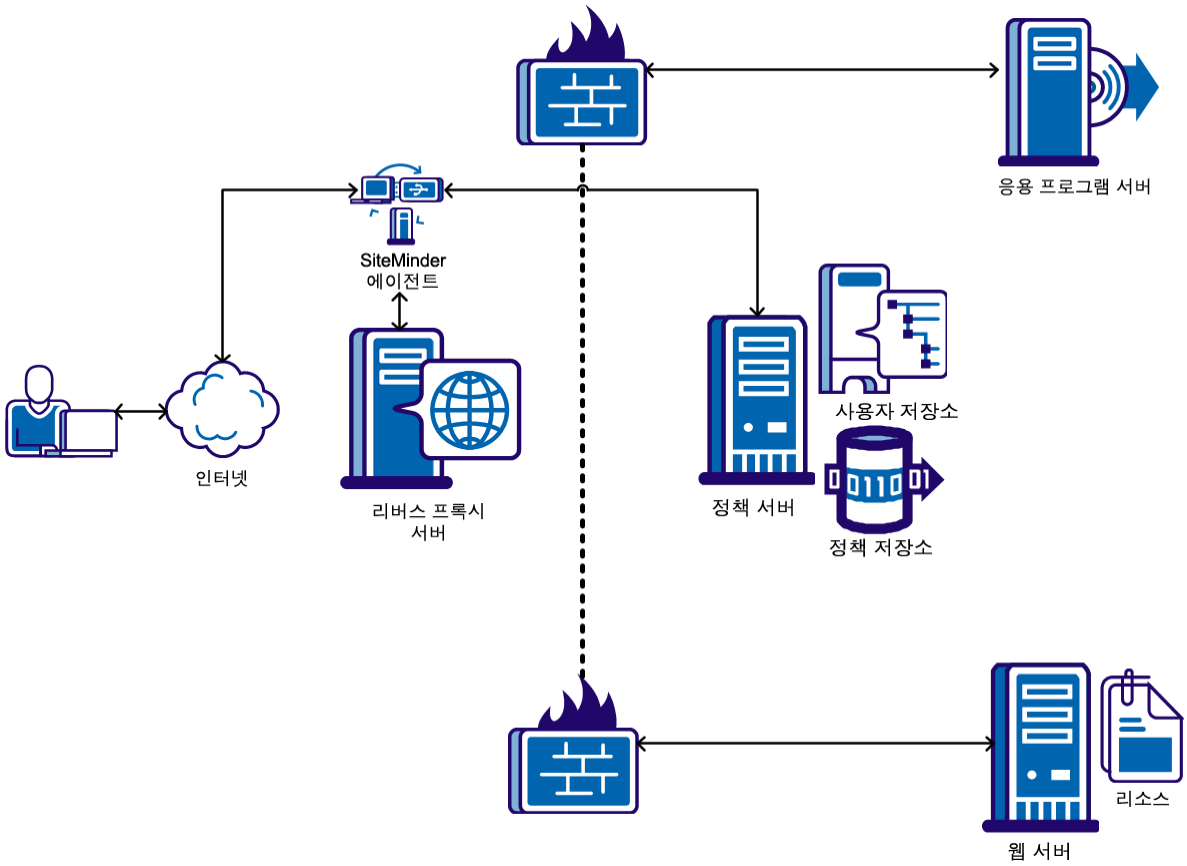
리버스 프록시 서버와 SiteMinder 의 상호 운용 방식

리버스 프록시 서버는 조직의 내부 네트워크에 요청을 전달하기 위해 엔터프라이즈를 대신하여 사용되는 프록시 서버입니다. 리버스 프록시 서버는 클라이언트가 백엔드 서버, 즉 방화벽으로 보호되는 서버의 리소스에 액세스할 수 있도록 합니다.

리버스 프록시 서버를 사용하면 다음과 같은 이점이 있습니다.

- 쿠키 도메인 내의 사용자는 재인증하지 않고 백엔드 서버의 리소스에 액세스할 수 있습니다. 다른 도메인의 사용자는 같은 백엔드 서버에 액세스하기 전에 리버스 프록시 서버 및 방화벽을 통해 인증해야 합니다.
- 사용자는 도메인 이름이 같은 여러 백엔드 서버에서 호스트되는 다양한 리소스에 액세스할 수 있습니다.
- 리버스 프록시 에이전트는 다른 SiteMinder 에이전트와 같은 기능을 지원합니다.
- SiteMinder 에이전트가 지원되지 않는 서버의 리소스를 보호합니다. 이 경우에는 리버스 프록시 서버를 백엔드 서버 앞에 배포하십시오. 그러면 지원되는 에이전트가 백엔드 서버에서 호스트되는 리소스를 보호합니다. 백엔드 서버에는 SiteMinder 에이전트가 필요하지 않습니다.

리버스 프록시 서버에 설치되는 SiteMinder 에이전트는 백엔드 서버의 리소스를 보호할 수 있습니다. 다음 그림에서는 SiteMinder 에이전트를 사용하는 리버스 프록시 서버가 설치된 네트워크를 보여 줍니다.



SiteMinder Secure Proxy Server

좀 더 정교한 리버스 프록시 솔루션을 요구하는 사용자를 위해 CA SiteMinder for Secure Proxy Server 는 Apache 또는 Oracle iPlanet 기반 SiteMinder 리버스 프록시 에이전트를 통해 다음과 같은 이점을 제공합니다.

- SSL 가속기 카드 지원 및 키와 인증서 관리를 위한 GUI 도구 등을 포함하여 완벽하게 지원되는 내장 웹 서버
- 여러 세션 체계 지원(쿠키 기반 모드, 쿠키를 사용하지 않는 모드)
- 다음과 같은 유연한 프록시 규칙 지원
  - URL 뿐만 아니라 HTTP 헤더 및 SiteMinder 응답을 사용한 규칙 지원
  - 복잡한 규칙을 쉽게 사용할 수 있음

보안 프록시 서버에서 SiteMinder 처리를 위한 SM_PROXYREQUEST HTTP 헤더

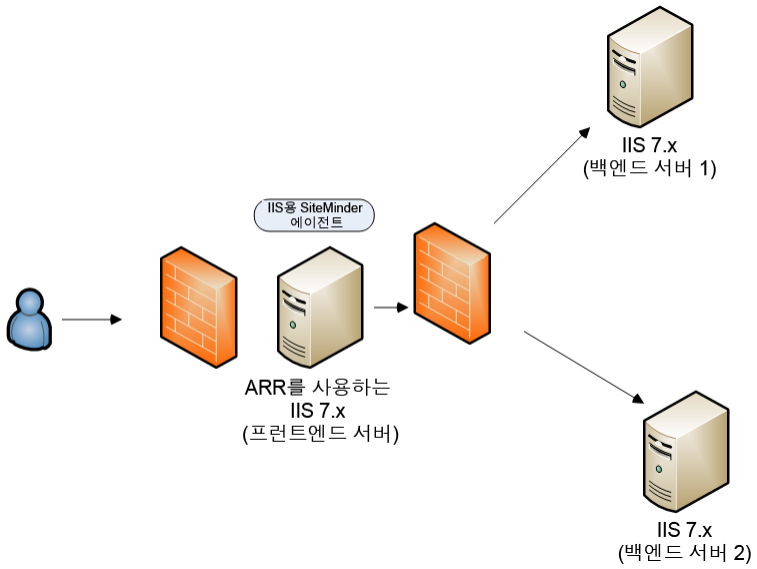
CA SiteMinder for Secure Proxy Server 는 기존의 SiteMinder 아키텍처에 새로운 계층을 도입합니다. 이 계층은 모든 요청을 엔터프라이즈 내의 대상 서버에 전달하거나 리디렉션합니다.

CA SiteMinder for Secure Proxy Server 에서 요청이 처리될 때 사용자가 요청한 URL 은 SM_PROXYREQUEST 라는 HTTP 헤더 변수에 유지됩니다. CA SiteMinder for Secure Proxy Server 에서 요청이 프록시되기 전에 사용자가 요청한 원래 URL 을 필요로 하는 다른 응용 프로그램에서는 이 헤더를 사용할 수 있습니다.

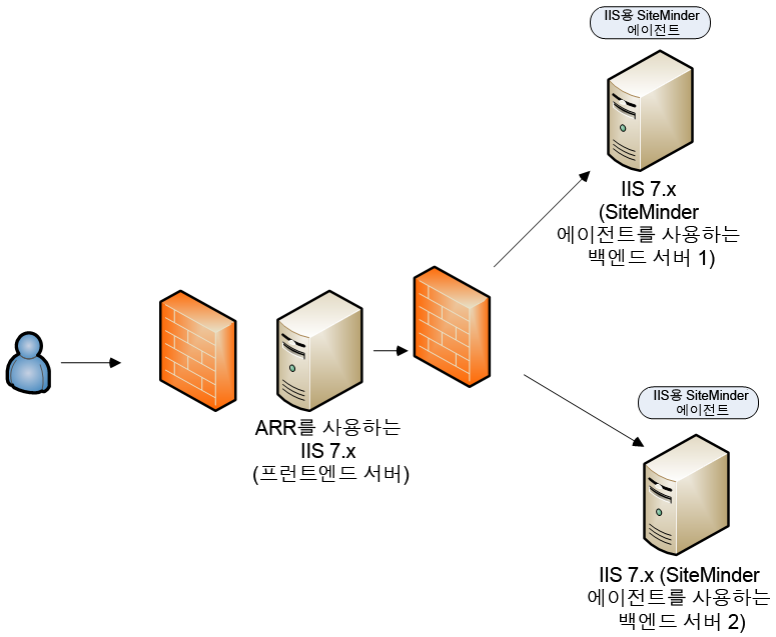
SiteMinder IIS 7.x 웹 서버 및 ARR

IIS 용 SiteMinder 12.52 SP1 에이전트는 IIS 7.x 의 ARR(응용 프로그램 요청 라우팅) 기능을 지원합니다. 다음과 같은 구성이 지원됩니다.

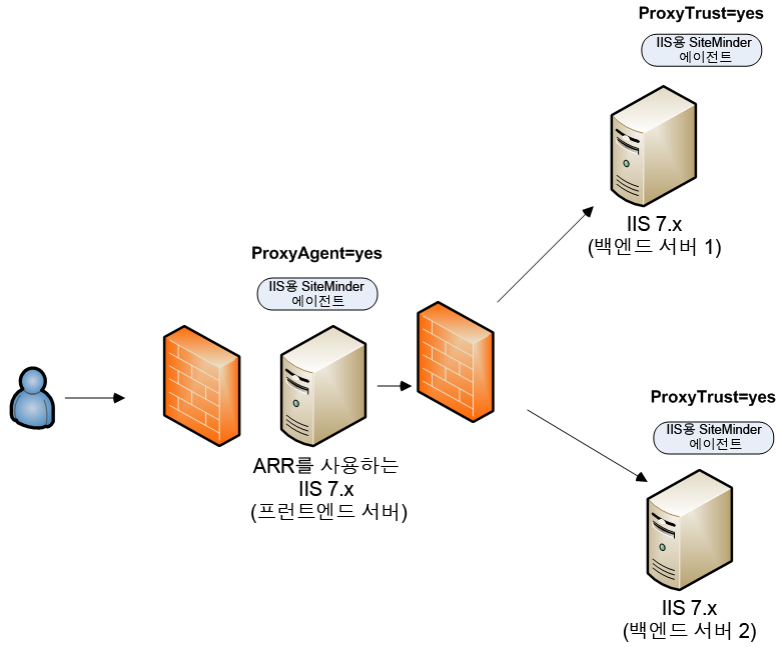
- 다음 그림과 같이 DMZ 에서 IIS 용 SiteMinder 에이전트와 ARR 를 둘 다 실행하는 IIS 7.x 웹 서버 하나 (페이지 303)



- DMZ 에서 ARR 를 실행하는 다른 IIS 7.x 서버 뒤에 IIS 용 SiteMinder 웹 에이전트를 실행하는 여러 IIS 7.x 웹 서버. 이 구성을 그림으로 나타내면 다음과 같습니다. (페이지 304)



- DMZ 에서 IIS 용 SiteMinder 에이전트와 ARR 를 둘 다 실행하는 IIS 7.x 웹 서버 하나와 ARR 서버 뒤에 IIS 용 SiteMinder 웹 에이전트를 실행하는 여러 IIS 7.x 웹 서버. 이 구성을 그림으로 나타내면 다음과 같습니다. (페이지 299)



DMZ 에 ARR 와 SiteMinder 가 있고 DMZ 뒤에 다른 IIS 용 SiteMinder 에이전트가 실행되는 IIS 7.x 서버를 설정하는 방법

IIS 용 SiteMinder 에이전트는 다음과 같은 구성으로 전체 IIS 환경을 보호합니다.

- DMZ 에서 IIS 용 SiteMinder 에이전트와 ARR(응용 프로그램 요청 라우팅)를 실행하는 IIS 7.x 웹 서버(프런트엔드 서버)
- DMZ 의 ARR 서버 *뒤에* 있는 여러 IIS 7.x 웹 서버. 각 서버는 SiteMinder 웹 에이전트 *또는* IIS 용 에이전트를 사용합니다.

  참고: 특정 SiteMinder 웹 에이전트만 리버스 프록시 서버로 작동하는 것을 지원합니다. 하지만 지원되는 SiteMinder 웹 에이전트 또는 IIS 용 에이전트를 호스트하는 웹 서버는 SiteMinder 를 실행하는 리버스 프록시 서버의 트래픽을 수락할 수 있습니다. 자세한 내용은 플랫폼 지원표를 참조하십시오.

위의 구성을 구현하려면 다음 단계를 수행하십시오.

1. DMZ 의 IIS 7.x 웹 서버에 ARR 를 설치 및 구성합니다(프런트엔드).

   참고: ARR(Application Request Routing)에 대한 자세한 내용을 보려면 IIS 웹 사이트로 이동하여 "Application Request Routing" 구문을 검색하십시오.

2. DMZ 의 IIS 7.x 웹 서버에 IIS 용 SiteMinder 에이전트를 설치 및 구성합니다(프런트엔드).

   참고: 자세한 내용은 "IIS 용 웹 에이전트 설치 안내서"를 참조하십시오.

3. DMZ 의 IIS 용 SiteMinder 에이전트에 대한 웹 에이전트 구성 매개 변수를 설정합니다 (페이지 300).

4. DMZ *뒤의* 첫 번째 IIS 7.x 웹 서버에 IIS 용 SiteMinder 에이전트를 설치 및 구성합니다(백엔드). 자세한 내용은 "IIS 용 웹 에이전트 설치 안내서"를 참조하십시오.

   참고: 이 컨텍스트에서 첫 번째 서버는 공유 구성 정보가 저장된 팜의 IIS 웹 서버입니다. 노드는 첫 번째 서버에서 공유 구성을 읽는 팜의 다른 모든 IIS 웹 서버입니다.

5. DMZ *뒤의* 다른 IIS 7.x 웹 서버 노드에 IIS 용 SiteMinder 에이전트를 설치 및 구성합니다(백엔드).

6. DMZ *뒤에서* SiteMinder 를 사용하는 모든 IIS 7.x 서버에 대한 웹 에이전트 구성 매개 변수를 설정합니다 (페이지 302). *첫 번째* 웹 서버 및 *모든* 노드를 포함해야 합니다.

DMZ 의 IIS 7.x ARR 서버에 대한 SiteMinder 웹 에이전트 구성 매개 변수 설정

이 단원에서는 다음과 같은 상황에서 IIS 용 SiteMinder 에이전트를 실행하는 웹 에이전트 구성 매개 변수를 설정하는 방법을 보여 줍니다.

- IIS 7.x 웹 서버 하나는 ARR 및 IIS 용 SiteMinder 에이전트를 사용하여 DMZ 에서 동작합니다(프런트엔드).
- DMZ 뒤의 다른 IIS 7.x 웹 서버는 ARR 서버로부터 요청을 수신하지만 IIS 용 SiteMinder 에이전트를 사용하지 *않습니다*(백엔드).

다음 단계를 수행하십시오.

1. 다음 항목을 확인합니다.
   - ARR 2.0 이 DMZ 의 웹 서버에 설치 및 구성되어 있습니다.
   - IIS 용 SiteMinder 12.52 SP1 에이전트가 DMZ 의 웹 서버에 설치 및 구성되어 있습니다.
2. 관리 UI 를 엽니다.
3. IIS 용 SiteMinder 에이전트(DMZ 에서 실행되는 프런트엔드)와 연결된 ACO(에이전트 구성 개체)를 엽니다.
4. 다음 매개 변수를 찾습니다.

   ProxyTrust

   프록시 서버의 SiteMinder 에이전트로부터 받은 권한 부여를 트러스트하도록 대상 서버의 에이전트에 지시합니다. 대상 서버는 리버스 프록시 서버 뒤에 있는 서버입니다. 이 값을 yes 로 설정하면 권한 부여를 위해 프록시 서버의 에이전트만 정책 서버에 연결하므로 효율성이 높아집니다. 대상 서버에서 작동하는 에이전트는 사용자 권한을 다시 부여할 때 정책 서버에 연결하지 *않습니다*.

   기본값: No
5. ProxyTrust 매개 변수의 값이 no 로 설정되어 있는지 확인합니다.
6. 다음 매개 변수를 찾습니다.

   ProxyAgent

   웹 에이전트가 리버스 프록시 에이전트로 사용되는지 여부를 지정합니다.

이 매개 변수의 값이 yes 이면 프런트엔드 서버의 SiteMinder 에이전트는 사용자가 요청한 원래 URL 을 SM_PROXYREQUEST HTTP 헤더에 보존합니다. 이 헤더는 보호된 리소스 및 보호되지 않은 리소스가 요청될 때마다 생성됩니다. 백엔드 서버는 이 헤더를 읽고 원래 URL 에 대한 정보를 가져올 수 있습니다.

기본값: No

7. ProxyAgent 매개 변수의 값을 yes 로 변경합니다.
8. 변경 내용을 에이전트 구성 개체에 적용합니다.

   웹 에이전트 구성 매개 변수가 설정됩니다.

DMZ 뒤에서 SiteMinder 를 사용하는 IIS 7.x 서버에 대한 웹 에이전트 구성 매개 변수 설정

이 단원에서는 다음과 같은 상황에서 IIS 용 SiteMinder 에이전트를 실행하는 웹 에이전트 구성 매개 변수를 설정하는 방법을 보여 줍니다.

- IIS 7.x 서버 하나는 ARR 를 사용하여 DMZ 에서 동작합니다(프런트엔드).
- DMZ 뒤의 다른 IIS 7.x 서버는 ARR 서버로부터 요청을 수신합니다. 또한 이러한 서버는 IIS 용 SiteMinder 에이전트를 사용합니다(백엔드).

다음 단계를 수행하십시오.

1. 다음 항목을 확인합니다.
   - ARR 2.0 이 DMZ 의 웹 서버에 설치 및 구성되어 있습니다.
   - IIS 용 SiteMinder 12.52 SP1 에이전트가 첫 번째 웹 서버에 설치 및 구성되어 있고 모든 노드는 DMZ *뒤에* 있습니다.
2. 관리 UI 를 엽니다.
3. DMZ *뒤에* 배포된 첫 번째 IIS 서버와 연결되어 있는 ACO(에이전트 구성 개체)를 엽니다.
4. 다음 매개 변수를 찾습니다.

   ProxyTrust

   프록시 서버의 SiteMinder 에이전트로부터 받은 권한 부여를 트러스트하도록 대상 서버의 에이전트에 지시합니다. 대상 서버는 리버스 프록시 서버 뒤에 있는 서버입니다. 이 값을 yes 로 설정하면 권한 부여를 위해 프록시 서버의 에이전트만 정책 서버에 연결하므로 효율성이 높아집니다. 대상 서버에서 작동하는 에이전트는 사용자 권한을 다시 부여할 때 정책 서버에 연결하지 *않습니다*.

   기본값: No
5. ProxyTrust 매개 변수의 값을 yes 로 변경합니다.
6. 다음 매개 변수를 찾습니다.

   ProxyAgent

   웹 에이전트가 리버스 프록시 에이전트로 사용되는지 여부를 지정합니다.

이 매개 변수의 값이 yes 이면 프런트엔드 서버의 SiteMinder 에이전트는 사용자가 요청한 원래 URL 을 SM_PROXYREQUEST HTTP 헤더에 보존합니다. 이 헤더는 보호된 리소스 및 보호되지 않은 리소스가 요청될 때마다 생성됩니다. 백엔드 서버는 이 헤더를 읽고 원래 URL 에 대한 정보를 가져올 수 있습니다.

기본값: No

7. ProxyAgent 매개 변수의 값이 no 로 설정되어 있는지 확인합니다.
8. 변경 내용을 에이전트 구성 개체에 적용합니다.
9. DMZ *뒤에* 배포된 IIS 서버 노드와 연결되어 있는 ACO(에이전트 구성 개체)를 엽니다.
10. 각각의 IIS 웹 서버 노드에 대해 5-10 단계를 반복하여 DMZ 뒤의 모든 노드를 구성합니다.

    웹 에이전트 구성 매개 변수가 설정됩니다.

DMZ 에 ARR 와 SiteMinder 가 있는 IIS 7.x 서버를 설정하는 방법

DMZ 에 ARR(응용 프로그램 요청 라우팅)와 IIS 용 SiteMinder 에이전트가 있는 IIS 7.x 웹 서버, 즉 프런트엔드 서버를 설정하려면 다음 단계를 수행하십시오.

1. DMZ 의 IIS 7.x 웹 서버에 ARR 를 설치 및 구성합니다(프런트엔드).

   참고: ARR(Application Request Routing)에 대한 자세한 내용을 보려면 IIS 웹 사이트로 이동하여 "Application Request Routing" 구문을 검색하십시오.

2. DMZ 의 IIS 7.x 웹 서버에 IIS 용 SiteMinder 에이전트를 설치 및 구성합니다(프런트엔드).

   참고: 자세한 내용은 "IIS 용 웹 에이전트 설치 안내서"를 참조하십시오.

DMZ 의 ARR 서버 뒤에 SiteMinder 가 실행되는 IIS 7.x 서버를 설정하는 방법

IIS 용 SiteMinder 에이전트는 ARR(응용 프로그램 요청 라우팅)를 사용하여 다음 구성을 지원합니다.

- ARR 를 실행하는 DMZ 기반 IIS 7.x 웹 서버 *뒤에* 여러 개의 백엔드 웹 서버가 동작합니다.
- SiteMinder 웹 에이전트 또는 IIS 용 에이전트를 사용하여 백엔드 서버를 보호합니다.

  참고: 특정 SiteMinder 웹 에이전트만 리버스 프록시 서버로 작동하는 것을 지원합니다. 하지만 지원되는 SiteMinder 웹 에이전트 또는 IIS 용 에이전트를 호스트하는 웹 서버는 SiteMinder 를 실행하는 리버스 프록시 서버의 트래픽을 수락할 수 있습니다. 자세한 내용은 플랫폼 지원표를 참조하십시오.

이 구성을 구현하려면 다음 단계를 수행하십시오.

1. DMZ 의 IIS 7.x 웹 서버에 ARR 를 설치 및 구성합니다(프런트엔드).

   참고: ARR(Application Request Routing)에 대한 자세한 내용을 보려면 IIS 웹 사이트로 이동하여 "Application Request Routing" 구문을 검색하십시오.

2. DMZ *뒤의* 첫 번째 IIS 7.x 웹 서버에 IIS 용 SiteMinder 에이전트를 설치 및 구성합니다(백엔드). 자세한 내용은 "IIS 용 웹 에이전트 설치 안내서"를 참조하십시오.

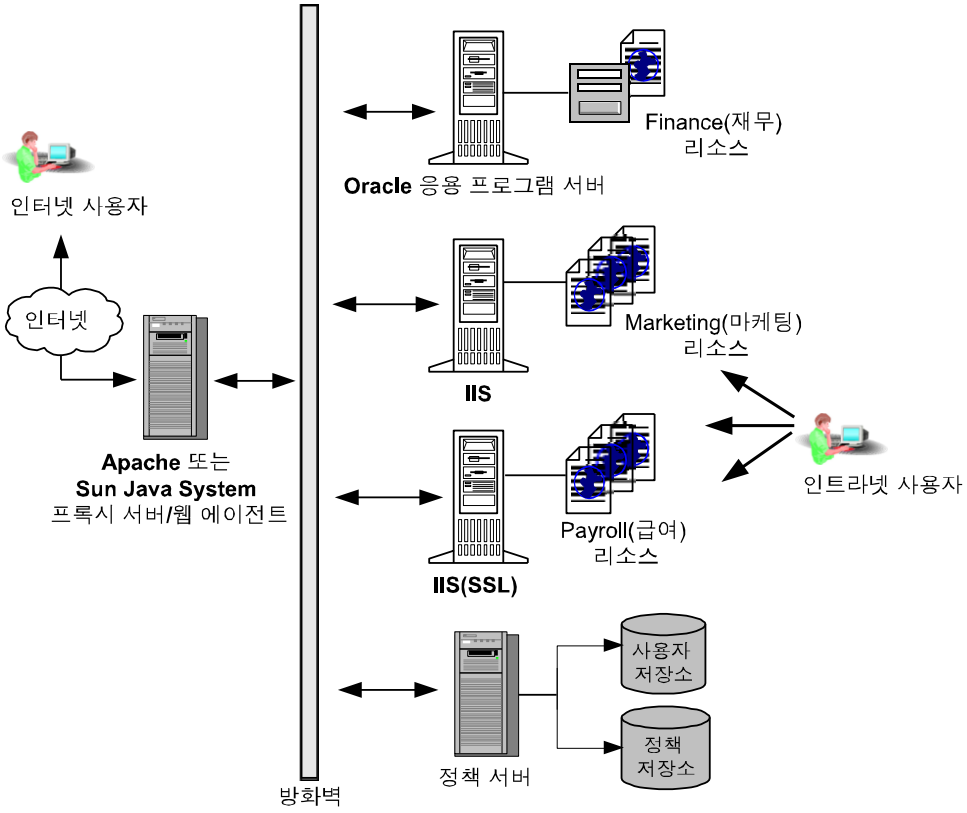
   참고: 이 컨텍스트에서 첫 번째 서버는 공유 구성 정보가 저장된 팜의 IIS 웹 서버입니다. 노드는 첫 번째 서버에서 공유 구성을 읽는 팜의 다른 모든 IIS 웹 서버입니다.

3. DMZ *뒤의* 다른 IIS 7.x 웹 서버 노드에 IIS 용 SiteMinder 에이전트를 설치 및 구성합니다(백엔드).

SiteMinder 리버스 프록시 배포 고려 사항

일반적으로 Apache 또는 Oracle iPlanet 리버스 프록시 에이전트를 배포하는 경우 보호된 리소스를 호스트하는 서버와 Apache 또는 Oracle iPlanet 웹 에이전트 사이에 방화벽이 있습니다. 또한 정책 서버도 방화벽 뒤에 있어야 합니다.

다음 그림에서는 SiteMinder 리버스 프록시 배포를 보여 줍니다.



SiteMinder 리버스 프록시 에이전트를 배포하는 경우 다음 사항을 고려하십시오.

- 응답 특성을 반환하도록 정책이 구성된 경우에는 보호된 리소스가 있는 백엔드 웹 서버와 리버스 프록시 서버 둘 다에 변수가 전달됩니다. 보호된 리소스가 요청되는 경우 정책 서버는 먼저 Apache 또는 Oracle iPlanet 서버의 에이전트에 응답 특성(CGI 또는 HTTP 변수)을 보냅니다. 그러면 에이전트는 백엔드 서버에 전달되는 요청에 응답 특성을 추가합니다.
- 백엔드 서버 또는 보호된 응용 프로그램에서 고유의 인증 기능을 제공하는 경우 인증이 사용되지 않도록 설정해야 합니다. 백엔드 인증이 사용되지 않도록 설정하면 SiteMinder 인증이 우선권을 갖습니다.

  중요! 리버스 프록시에 대한 캐시를 구성하는 경우 SMSESSION 쿠키를 포함한 모든 쿠키가 캐시된다는 것을 알아야 합니다. 자세한 내용은 Apache 또는 Oracle iPlanet 웹 서버 설명서를 참조하십시오.

추가 정보

HTTPS 포트 정의 (페이지 109)

Apache 리버스 프록시 서버를 구성하는 방법

SiteMinder 에이전트와 함께 리버스 프록시 서버 역할을 하도록 Apache 웹 서버를 구성할 수 있습니다. 다음은 Apache 리버스 프록시 서버를 구성하는 단계입니다.

1. Apache 웹 서버 구성 파일을 업데이트합니다 (페이지 307).
2. SiteMinder 에이전트에 대한 에이전트 구성 매개 변수를 업데이트합니다 (페이지 309).

Apache 웹 서버 구성 파일 업데이트

Apache 웹 서버가 SiteMinder 에이전트와 함께 리버스 프록시 서버 역할을 하도록 구성하려면 Apache 웹 서버의 구성 파일을 업데이트합니다.

다음 단계를 수행하십시오.

1. 다음 위치에 있는 httpd.conf 파일을 엽니다.

   ```
   /etc/httpd/conf/httpd.conf
   ```

2. 다음 지시문을 httpd.conf 파일에 추가합니다.

   ProxyPass

   원격 서버를 로컬 서버에 매핑할 수 있도록 합니다. 이 지시문의 값은 다음 형식을 사용합니다.

   /local_virtual_path partial_URL_of_remote_server

   예: ProxyPass /realma/ http://server.example.org/realma/

   ProxyPassReverse

   HTTP 리디렉션 응답에 대해 Apache 서버에서 위치 헤더를 조정할 수 있도록 합니다. 이 지시문의 값은 다음 형식을 사용합니다.

   /local_virtual_path partial_URL_of_remote_server

   예: ProxyPassReverse /realma/ http://server.example.org/realma/

   Apache 웹 서버에 대해 다음과 같은 Proxy Pass 설정을 구성 파일에 추가합니다.

   ```
   # SiteMinder Administrative UI
   <Location "/iam/siteminder/">
        <IfModule proxy_module>
                    ProxyPass http://hostname:port/iam/siteminder/
          ProxyPassReverse http://hostname:port/iam/siteminder/
       </IfModule>
     # Alternate unavailable page
     ErrorDocument 503 /siteminderagent/adminui/HTTP_SERVICE_UNAVAILABLE.html
   </Location>
   # CA Styles r5.1.1
   <Location "/castylesr5.1.1/">
       <IfModule proxy_module>
        ProxyPass http://hostname:port/castylesr5.1.1/
       ProxyPassReverse http://hostname:port/castylesr5.1.1/
       </IfModule>
   </Location>
   ```

   참고: *hostname:port* 는 관리 UI 를 실행하는 응용 프로그램 서버의 호스트 및 포트를 나타냅니다.

3. 구성 파일에서 다음 행의 주석 처리를 제거합니다.

   ```
   LoadModule proxy_module modules/mod_proxy.so
   ```

4. 구성 파일을 저장한 후 닫습니다.
5. Apache 웹 서버를 다시 시작합니다.

SiteMinder 에이전트에 대한 에이전트 구성 매개 변수 업데이트

Apache 리버스 프록시 서버 뒤의 Apache 기반 서버에 대해 다음과 같은 에이전트 구성 매개 변수를 업데이트합니다.

다음 단계를 수행하십시오.

1. 다음 매개 변수의 값을 yes 로 설정합니다.

   ProxyAgent

   웹 에이전트가 리버스 프록시 에이전트로 사용되는지 여부를 지정합니다.

   이 매개 변수의 값이 yes 이면 프런트엔드 서버의 SiteMinder 에이전트는 사용자가 요청한 원래 URL 을 SM_PROXYREQUEST HTTP 헤더에 보존합니다. 이 헤더는 보호된 리소스 및 보호되지 않은 리소스가 요청될 때마다 생성됩니다. 백엔드 서버는 이 헤더를 읽고 원래 URL 에 대한 정보를 가져올 수 있습니다.

   기본값: No

2. 다음 매개 변수를 설정합니다.

   ProxyTimeout

   리버스 프록시 서버 뒤에 배포된 SiteMinder 에이전트가 요청에 응답할 때까지 리버스 프록시 서버가 기다리는 시간(초)을 지정합니다.

   기본값: 120

   참고: 이 매개 변수는 Apache 기반 에이전트에만 적용됩니다.

3. (선택 사항) 다음 매개 변수를 설정합니다.

   ProxyTrust

   프록시 서버의 SiteMinder 에이전트로부터 받은 권한 부여를 트러스트하도록 대상 서버의 에이전트에 지시합니다. 대상 서버는 리버스 프록시 서버 뒤에 있는 서버입니다. 이 값을 yes 로 설정하면 권한 부여를 위해 프록시 서버의 에이전트만 정책 서버에 연결하므로 효율성이 높아집니다. 대상 서버에서 작동하는 에이전트는 사용자 권한을 다시 부여할 때 정책 서버에 연결하지 *않습니다.*

   기본값: No

4. 목록에서 다음 값을 모두 제거하여 BadURLChars 매개 변수를 편집합니다.

%

5. httpsports 매개 변수를 설정하여 SSL 을 위해 포트가 설정된 Apache 서버를 나타냅니다.
6. Apache 웹 서버를 다시 시작합니다.

참고: 에이전트 구성 매개 변수를 수정하는 데 대한 자세한 내용은 *정책 서버 구성 안내서*를 참조하십시오.

Oracle iPlanet 7.0 리버스 프록시 서버 구성

Oracle iPlanet 7.0 웹 서버를 SiteMinder 와 함께 리버스 프록시로 사용할 수 있습니다.

참고: SiteMinder 에이전트 구성 마법사는 Oracle iPlanet(이전의 Sun Java System) 웹 서버의 *기본* obj.conf *파일만* 수정합니다. SiteMinder 에서 다른 인스턴스 또는 리버스 프록시 배포를 보호하려면 기본 obj.conf 파일의 SiteMinder 설정을 각각의 *instance_name*-obj.conf 파일에 복사해야 합니다. 예를 들어 웹 서버를 설치할 때 해당 웹 서버에서 obj.conf 파일을 생성했는데 나중에 my_server.example.com 이라는 서버 인스턴스를 추가했다고 가정합니다. SiteMinder 에서 my_server.example.com 의 리소스를 보호하려면 마법사가 obj.conf 파일에 추가한 SiteMinder 설정을 my_server.example.com-obj.conf 파일에 복사해야 합니다.

다음 단계를 수행하십시오.

1. 다음 지시문을 *instance_name*-obj.conf 파일에 추가합니다.

   NameTrans

   다음 형식을 사용하여 로컬 및 원격 가상 경로를 지정합니다.

   ```
   NameTrans fn="map" from="local_virtual_path"
   name="reverse-proxy-/local_virtual_path" to="remote_virtual_path"
   ```

   예:

   ```
   NameTrans fn="map" from="/realma" name="reverse-proxy-/reamla"
   to="http://server.example.org/realma/"
   ```

2. 다음 지시문을 obj.conf 파일의 *끝에* 추가합니다.

   Object name

   다음 형식을 사용하여 NameTrans 지시문에 사용된 로컬 가상 경로의 이름 및 원격 가상 경로의 URL 을 지정합니다.

   ```
   <Object name="reverse-proxy-/local_virtual_path">
   Route fn="set-origin-server" server="http://remote_server_URL:port"
   </Object>
   ```

   예:

   ```
   <Object name="reverse-proxy-/realma">
   Route fn="set-origin-server" server="http://server.example.org:port"
   </Object>
   ```

   Object ppath

   클라이언트에 의해 서버에 제공되는 부분 경로를 지정합니다.

예:

```
<Object ppath="http:*">
Service fn="proxy-retrieve" method="*"
</Object>
```

3. 웹 서버를 다시 시작합니다.

   리버스 프록시가 구성됩니다.

HTTP 헤더 설정

다음 설정을 사용하면 URL 처리를 제어할 수 있습니다.

- URLScan 유틸리티 사용을 위해 HTTP Server 헤더 제거 (페이지 312)

URLScan 유틸리티를 사용하는 경우 HTTP Server 헤더 제거

Microsoft 의 URLScan 유틸리티를 사용하여 IIS 웹 서버가 보내는 응답에서 HTTP Server 헤더를 제거하려면 IIS 웹 에이전트의 다음 매개 변수도 설정해야 합니다.

SuppressServerHeader

IIS 웹 에이전트가 HTTP Server 헤더를 응답으로 반환하지 않도록 합니다. 이 매개 변수의 값을 no 로 설정하면 웹 에이전트가 응답에서 서버 헤더를 보내고 IIS 웹 서버가 이 헤더를 클라이언트에 전달합니다. 이 매개 변수의 값을 yes 로 설정하면 웹 에이전트가 서버 헤더를 응답으로 보내지 않습니다.

기본값: No

URLScan 유틸리티는 IIS 서버 응답에서 헤더를 제거하고 SuppressServerHeader 매개 변수는 웹 에이전트 응답에서 헤더를 제거합니다. 모든 응답의 Server 헤더가 클라이언트에 전달되지 않게 하려면 유틸리티와 매개 변수를 둘 다 설정해야 합니다.

웹 에이전트가 응답의 Server 헤더를 전달하지 않게 하려면 SuppressServerHeader 매개 변수의 값을 yes 로 설정하십시오.

URL 설정

다음 설정을 사용하면 URL 이 처리되는 방식을 제어할 수 있습니다.

- 소문자를 사용하여 프로토콜 지정 (페이지 190)
- URL 의 쿼리 데이터 디코딩 (페이지 110)
- 최대 URL 크기 설정 (페이지 314)

소문자를 사용하여 리디렉션 URL 프로토콜 지정

양식 기반 인증 체계를 사용하여 RFC 2396 을 준수하지 않는 레거시 응용 프로그램을 보호하고 URL 의 프로토콜 부분을 소문자로 지정해야 할 경우에는 다음 매개 변수를 설정합니다.

LowerCaseProtocolSpecifier

리디렉션 URL 의 스키마(프로토콜) 부분에 소문자만 사용할지 여부를 지정합니다. 이 구성 매개 변수는 RFC 2396 을 준수하지 않는 레거시 응용 프로그램을 사용할 수 있도록 조정합니다. 이 RFC 는 응용 프로그램이 URL 의 프로토콜 부분을 대문자와 소문자로 둘 다 처리해야 함을 명시합니다. 다음과 같은 경우 이 매개 변수를 변경하십시오.

- RFC 2396 을 준수하지 않는 레거시 응용 프로그램을 사용합니다.
- 쿼리 데이터가 포함된 URL 을 리디렉션합니다.
- HTML 양식(FCC) 인증 스키마를 사용합니다.

기본값: No(HTTP, HTTPS 와 같이 대문자 사용)

예: Yes(http, https 와 같이 소문자 사용)

환경에서 URL 의 프로토콜 부분을 소문자로 지정하려면 LowerCaseProtocolSpecifier 매개 변수의 값을 yes 로 설정하십시오.

URL 의 쿼리 데이터 디코딩

정책 서버를 호출하기 전에 URL 의 쿼리 데이터를 디코딩하여 정책 서버가 올바른 리소스를 인식하도록 웹 에이전트의 Base64 알고리즘을 구성하려면 다음 매개 변수를 사용합니다.

DecodeQueryData

웹 에이전트가 정책 서버를 호출하기 전에 URL 의 쿼리 데이터를 디코딩할지 여부를 지정합니다. 해당 환경에서 다음 태스크 중 하나를 수행해야 할 경우 이 매개 변수를 yes 로 설정하십시오.

- 규칙 필터가 적절한 문자열에 대해 작동하도록 해야 하는 경우
- 쿼리 문자열의 데이터에 대해 규칙을 작성해야 하는 경우

기본값: No

정책 서버를 호출하기 전에 웹 에이전트에서 URL 의 쿼리 데이터를 디코딩하도록 구성하려면 DecodeQueryData 매개 변수의 값을 yes 로 설정하십시오.

최대 URL 크기 설정

다음 매개 변수를 사용하면 웹 에이전트에서 처리할 수 있는 최대 URL 크기를 늘릴 수 있습니다.

MaxUrlSize

웹 에이전트가 처리할 수 있는 URL 의 최대 크기(바이트)를 지정합니다. 웹 서버마다 URL 길이 제한이 다르기 때문에 이 매개 변수를 설정하기 전에 해당 웹 서버 공급업체의 설명서를 확인하십시오.

기본값: 4096 B

최대 URL 크기를 변경하려면 MaxUrlSize 매개 변수에 지정된 바이트 수를 변경하십시오.

IIS 웹 서버 설정

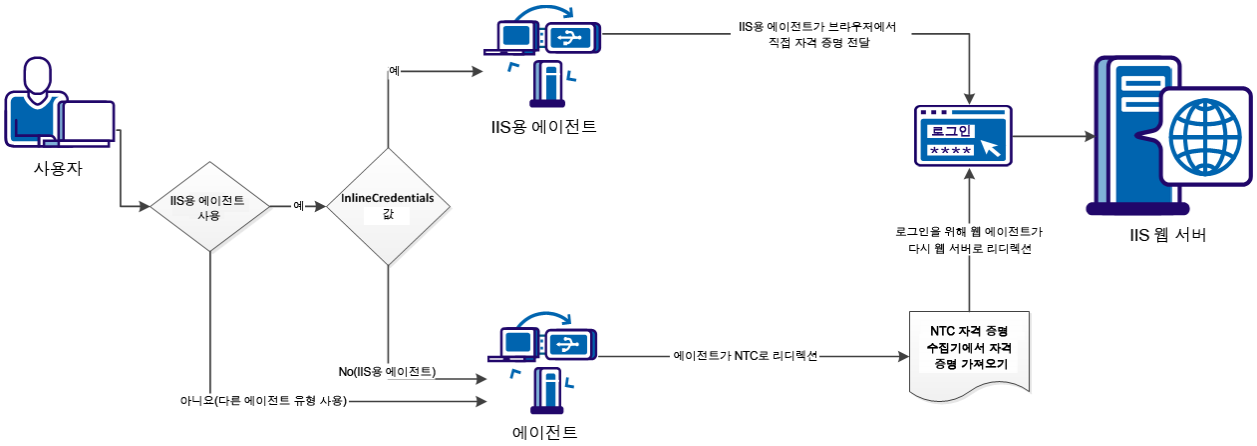
다음 설정을 사용하면 IIS 용 에이전트를 관리할 수 있습니다.

NTC로 리디렉션하지 않고 사용자 자격 증명을 가져오도록 IIS용 에이전트 구성

기본적으로 SiteMinder 에이전트는 Windows 인증 체계로 보호되는 리소스에 대한 요청을 NTC(NTLM credential collector)로 리디렉션하여 Windows 자격 증명을 가져옵니다.

NTC 로 리디렉션하지 않고 HTTP 요청에서 인라인으로 사용자 자격 증명을 가져오도록 IIS 용 SiteMinder 에이전트를 구성할 수 있습니다.

다음 그림에서는 자격 증명을 수집하는 두 방법의 차이점을 보여 줍니다.



NTC 로 리디렉션하지 않고 HTTP 요청에서 사용자 자격 증명을 가져오도록 에이전트를 구성하려면 다음과 같이 InlineCredentials 구성 매개 변수를 설정합니다.

InlineCredentials

IIS 용 에이전트에서 사용자 자격 증명을 처리하는 방식을 지정합니다. 이 매개 변수의 값이 yes 인 경우 IIS 용 에이전트는 HTTP 요청에서 직접 자격 증명을 읽습니다. 이 매개 변수의 값이 no 인 경우 에이전트는 NTC 자격 증명 수집기로 리디렉션합니다.

기본값: No

참고: 운영 환경의 SiteMinder 에이전트가 NTC 리디렉션을 사용하도록 구성된 경우에는 NT 챌린지/응답 인증을 구성하십시오.

추가 정보:

NT 챌린지/응답 인증을 지원하도록 IIS 용 에이전트 구성 (페이지 319)

IIS 서버 로그에 사용자 이름 및 트랜잭션 ID 기록

웹 에이전트는 사용자 권한 부여 요청이 성공할 때마다 고유한 트랜잭션 ID 를 생성합니다. 에이전트는 이 ID 를 HTTP 헤더에 추가합니다. 또한 다음 로그에도 ID 가 기록됩니다.

- 감사 로그
- 웹 서버 로그(서버가 쿼리 문자열을 기록하도록 구성된 경우)
- 정책 서버 로그

트랜잭션 ID 를 사용하면 지정된 응용 프로그램에 대한 사용자 활동을 추적할 수 있습니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

트랜잭션 ID 는 로그에서 기존 쿼리 문자열의 끝에 추가되는 모의 쿼리 매개 변수로 나타납니다. 다음 예제에서는 STATE=MA 로 끝나는 쿼리 문자열에 추가된 트랜잭션 ID(굵게 표시됨)를 보여 줍니다.

```
172.24.12.1, user1, 2/11/00, 15:30:10, W3SVC, MYSERVER, 192.168.100.100, 26844,
47, 101, 400, 123, GET, /realm/index.html,
STATE=MA&SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1
```

URL 에 쿼리 매개 변수가 없는 경우 에이전트는 웹 서버 로그 항목의 끝에 트랜잭션 ID 를 추가합니다. 예를 들면 다음과 같습니다.

```
172.24.12.1, user1, 2/11/00, 15:30:10, W3SVC, MYSERVER, 192.168.100.100, 26844,
47, 101, 400, 123, GET, /realma/index.html,
SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1.
```

참고: 웹 에이전트는 사용자가 리소스에 액세스할 때 사용자 이름 및 액세스 정보를 네이티브 웹 서버 로그 파일에 기록합니다.

IIS 서버의 리소스를 보호하는 에이전트는 기본적으로 SiteMinder 트랜잭션 ID 및 인증된 사용자 이름을 IIS 서버 로그에 기록하지 *않습니다*. 이러한 에이전트는 ISAPI 확장(클래식 파이프라인 모드)으로 작동할 수 있으므로 서버에서 이미 트랜잭션을 로그에 기록했습니다. 다음 매개 변수를 사용하여 에이전트가 이 정보를 추가하도록 지정할 수 있습니다.

AppendIISServerLog

인증된 사용자 이름 및 SiteMinder 트랜잭션 ID 를 IIS 서버 로그의 cs-uri-query 필드에 추가하도록 에이전트에 지시합니다. 쿼리 문자열을 포함하는 URL 의 경우 IIS 서버 로그의 cs-uri-query 필드에서 쿼리 문자열, SiteMinder 트랜잭션 ID 및 사용자 이름을 쉼표로 구분해야 합니다.

기본값: No

SetRemoteUser

레거시 응용 프로그램에 필요한 경우 *REMOTE_USER* HTTP 헤더 변수의 값을 지정합니다.

기본값: No

트랜잭션 ID 및 사용자 이름을 IIS 서버 로그에 기록하려면

1. AppendIISServerLog 매개 변수의 값을 yes 로 설정합니다.
2. SetRemoteUser 매개 변수의 값을 yes 로 설정합니다.

   사용자 이름과 트랜잭션 ID 가 IIS 서버 로그에 나타납니다.

IIS 인증을 위해 NetBIOS 이름 또는 UPN 사용

IIS 네트워크에서는 요청된 리소스의 위치에 대해 NetBIOS 이름과 도메인 이름이 다를 수 있습니다. 사용자가 보호된 리소스에 액세스를 시도하는 경우 도메인 컨트롤러가 여러 개 있으면 사용자 인증이 실패하고 웹 서버 로그에 "IIS 로그온 실패"라고 표시됩니다. 다음 매개 변수를 사용하여 IIS 웹 서버에 UPN 이름을 전달할지 아니면 NetBIOS 이름을 전달할지를 제어할 수 있습니다.

UseNetBIOSforIISAuth

IIS 6.0 웹 에이전트가 IIS 사용자 인증을 위해 UPN(사용자 프린서펄 이름) 또는 NetBIOS 이름을 IIS 6.0 웹 서버에 보낼지 여부를 지정합니다.

참고: 이 매개 변수는 Active Directory 사용자 저장소가 정책 서버와 연결된 경우에만 유효합니다.

이 매개 변수를 설정하면 SiteMinder 인증 동안 정책 서버가 Active Directory 에서 UserDN, UPN 및 NetBIOS 이름을 추출하여 이 데이터를 다시 IIS 6.0 웹 에이전트로 보냅니다.

관리 UI 에서 사용자 디렉터리에 대해 "Run in Authenticated User's Security Context"(인증된 사용자의 보안 컨텍스트에서 실행) 옵션을 선택했는지 여부와 UseNetBIOSforIIAuth 매개 변수를 설정하는 방식에 따라 사용자의 로그온 자격 증명이 다음과 같이 전송됩니다.

- UseNetBIOSforIISAuth 매개 변수를 no 로 설정하면 IIS 6.0 웹 에이전트가 UPN 이름을 보냅니다.
- UseNetBIOSforIISAuth 매개 변수를 yes 로 설정하면 웹 에이전트가 NetBIOS 이름을 보냅니다.

IIS 웹 서버는 웹 에이전트에서 받은 자격 증명으로 사용자를 인증합니다.

기본값: No

웹 에이전트에서 IIS 인증에 NetBIOS 이름을 사용하게 하려면 UseNetBIOSAuth 매개 변수를 yes 로 설정합니다.

NT 챌린지/응답 인증을 지원하도록 IIS 용 에이전트 구성

운영 환경의 SiteMinder 에이전트가 NTC 리디렉션을 사용하도록 구성된 경우에는 NT 챌린지/응답 인증을 구성하십시오.

NT 챌린지/응답 인증을 사용하면 IIS 웹 서버는 사용자가 리소스에 대한 액세스를 요청할 때 해당 사용자의 Internet Explorer 브라우저를 요청합니다.

참고: NT 챌린지/응답 인증은 Internet Explorer 브라우저에서만 동작합니다.

다음 방법 중 하나를 사용하여 NT 챌린지/응답 인증을 구현할 수 있습니다.

- 사용자가 보호된 리소스에 액세스할 때 해당 사용자에게 인증을 요청합니다. 싱글 사인온 환경의 사용자는 리소스를 처음 요청할 때만 인증 질문을 받습니다.
- 사용자가 Internet Explorer 브라우저의 자동 로그온 기능을 구성하도록 합니다.

  자동 로그온 기능을 사용하면 인증 요청 없이 사용자가 리소스에 액세스할 수 있습니다. 인증 프로세스가 수행되지만 브라우저와 서버 간 NT 챌린지/응답 프로세스가 사용자에게 보이지 않습니다. 일반적으로 자동 로그온 기능은 보안 수준이 낮은 인트라넷에서 사용자가 리소스에 원활하게 액세스할 수 있도록 하기 위해 사용됩니다. 인터넷 통신에는 자동 로그온 기능을 사용하지 않는 것이 좋습니다.

SiteMinder 에이전트는 자격 증명 수집기를 사용하여 NT 챌린지/응답 인증 체계에 대한 사용자의 Windows 자격 증명을 수집합니다. 또한 에이전트는 NTLM 자격 증명을 수집하기 위해 NTC 확장을 지원합니다.

참고: 이 기본 동작을 변경하려는 경우에만 NTCEXT 를 설정하십시오.

SiteMinder 가 NT 챌린지/응답 인증과 함께 작동하게 하려면 다음을 수행하십시오.

1. 다음 태스크를 수행하여 IIS 웹 서버에 대한 NT 챌린지/응답 인증을 설정합니다.
   a. .ntc 파일 확장명을 매핑합니다 (페이지 321).
   b. 가상 디렉터리를 생성하고 구성한 다음, NT 챌린지 및 응답 자격 증명이 필요한지 확인합니다 (페이지 322).
2. 관리 UI 에서 NT 챌린지/응답 인증을 위한 Windows 인증 체계를 구성합니다 (페이지 323).
3. NTC(NTLM credential collector)를 지정합니다 (페이지 209).

4. 관리 UI 를 사용하여 NT 챌린지/응답 인증에 대한 정책을 구성합니다.

   참고: 자세한 내용은 *정책 서버 구성 안내서*를 참조하십시오.

5. (선택 사항) 사용자가 Internet Explorer 브라우저의 자동 로그온 기능을 구성하도록 합니다 (페이지 325).

   IIS 에 대한 NT 챌린지/응답 인증이 구성됩니다.

추가 정보

NTC 로 리디렉션하지 않고 사용자 자격 증명을 가져오도록 IIS 용 에이전트 구성 (페이지 315)

.NTC 파일 확장명 매핑

IIS 웹 서버에서 NT 챌린지/응답 인증을 구성하려면 .NTC 파일 확장명을 ISAPIWebAgent.dll 응용 프로그램에 매핑합니다.

.NTC 파일 확장명을 매핑하려면

1. 인터넷 서비스 관리자를 엽니다.
2. 왼쪽 창에서 "웹 사이트"를 마우스 오른쪽 단추로 클릭하고 오른쪽 창에서 "기본 웹 사이트"를 마우스 오른쪽 단추로 클릭한 후 "속성"을 선택합니다.

   "기본 웹 사이트 속성" 대화 상자가 나타납니다.

3. "홈 디렉터리" 탭을 클릭합니다.

4. "응용 프로그램 설정" 섹션에서 "구성"을 클릭합니다.

   "응용 프로그램 구성" 대화 상자가 나타납니다.

5. "추가"를 클릭합니다.

   "응용 프로그램 확장명 매핑 추가/편집" 대화 상자가 열립니다.

   a. "실행 파일" 필드에서 "찾아보기"를 클릭하고 *web_agent_home*/bin/ISAPIWebAgent.dll 파일을 찾습니다.

   b. "Open"(열기)을 클릭합니다.

   c. "확장명" 필드에 ".ntc"를 입력합니다.

6. "확인"을 세 번 클릭합니다.

   "응용 프로그램 확장명 매핑 추가/편집" 대화 상자, "응용 프로그램 구성" 대화 상자, "기본 웹 사이트 속성" 대화 상자가 차례대로 닫힙니다. .ntc 파일 확장명이 매핑됩니다.

Windows 인증 체계를 위한 가상 디렉터리 만들기 및 구성(IIS 7.5)

SiteMinder Windows 인증 체계를 사용하려면 IIS 7.x 웹 서버에 가상 디렉터리를 구성합니다. 이 가상 디렉터리에는 자격 증명을 위한 Windows 챌린지 및 응답이 필요합니다.

다음 단계를 수행하십시오.

1. IIS(인터넷 정보 서비스) 관리자를 엽니다.
2. 왼쪽 창에서 다음 항목을 확장합니다.
   - 웹 서버 아이콘
   - 사이트 폴더
   - 기본 웹 사이트 아이콘
3. siteminderagent 가상 디렉터리를 마우스 오른쪽 단추로 클릭하고 "가상 디렉터리 추가"를 선택합니다.

   "가상 디렉터리 추가" 대화 상자가 나타납니다.

4. "별칭" 필드에 다음 값을 입력합니다.

   ```
   ntlm
   ```

5. "실제 경로" 필드 옆에 있는 "찾아보기" 단추를 클릭한 후 다음 디렉터리를 찾습니다.

   web_agent_home\samples

   가상 디렉터리가 생성됩니다.

6. 다음 단계 중 *하나*를 수행하여 가상 디렉터리를 구성합니다.

   - SiteMinder Windows 인증 체계를 사용하여 전체 웹 사이트의 모든 리소스를 보호하려면 "기본 웹 사이트" 아이콘을 클릭합니다.
   - SiteMinder Windows 인증 체계를 사용하여 전체 웹 사이트를 보호하지 *않으려면* 4 단계에서 생성한 ntlm 가상 디렉터리를 클릭합니다.

7. "인증" 아이콘을 두 번 클릭합니다.

   "인증" 대화 상자가 나타납니다.

8. 다음 단계를 수행하십시오.

   a. "익명 인증"을 마우스 오른쪽 단추로 클릭하고 "사용 안 함"을 선택합니다.

   b. "Windows 인증"을 마우스 오른쪽 단추로 클릭하고 "사용"을 선택합니다.

      Windows 인증 체계를 위한 가상 디렉터리가 구성됩니다.

   참고: 이러한 변경 내용을 적용하려면 웹 서버를 재부팅하십시오.

챌린지/응답 인증을 위해 Windows 인증 체계 구성

NT 챌린지/응답 인증을 구현하려면 Windows 인증 체계를 구성하는 정책 관리자에게 다음 값을 제공해야 합니다.

서버 이름

다음과 같은 IIS 웹 서버의 정규화된 도메인 이름입니다.

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

추가 정보

자격 증명 수집기의 MIME 유형 (페이지 177)

NTC(NTLM Credential Collector) 지정

NTC(NTLM credential collector)는 웹 에이전트 내의 응용 프로그램입니다. NTC 는 Windows 인증 체계로 보호하는 리소스에 대한 NT 자격 증명을 수집합니다. 이 체계는 IIS 웹 서버에서 Internet Explorer 브라우저로 액세스하는 리소스에 적용됩니다.

각 자격 증명 수집기에는 MIME 유형이 연결됩니다. IIS 의 경우 다음 매개 변수에 NTC MIME 유형이 정의됩니다.

NTCExt

NTC(NTLM credential collector)와 연결되는 MIME 유형을 지정합니다. 이 수집기는 Windows 인증 체계에서 보호하는 리소스에 대한 NT 자격 증명을 수집합니다. 이 인증 체계는 Internet Explorer 브라우저 사용자만 액세스하는 IIS 웹 서버의 리소스에 적용됩니다.

이 매개 변수에 여러 개의 확장명을 사용할 수 있습니다. 에이전트 구성 개체를 사용하는 경우 다중값 옵션을 선택하십시오. 로컬 구성 파일을 사용하는 경우 각 확장명을 쉼표로 구분하십시오.

기본값: .ntc

NTCExt 매개 변수에 지정되는 기본 확장명이 환경에 이미 사용되는 경우 다른 MIME 유형을 지정할 수 있습니다.

자격 증명 수집기를 트리거하는 확장명을 변경하려면 NTCExt 매개 변수에 다른 파일 확장명을 추가합니다.

Internet Explorer 의 자동 로그온 구성

Internet Explorer 브라우저 사용자가 자동 로그온 브라우저 보안 설정을 구성하면 에이전트에서 자격 증명을 요청하지 *않고* 사용자를 인증할 수 있습니다.

다음 단계를 수행하십시오.

1. Internet Explorer 브라우저를 시작합니다.
2. "인터넷 옵션" 대화 상자를 엽니다. 사용 중인 브라우저 버전에서 이 대화 상자를 여는 방법을 보려면 Internet Explorer 온라인 도움말을 참조하십시오.
3. "보안" 탭을 클릭합니다.
4. 올바른 보안 영역을 클릭합니다.
5. "사용자 지정 수준"을 클릭합니다.
6. 아래로 스크롤하여 "사용자 인증" 섹션을 찾습니다. "로그온" 옵션에서 "현재 사용자 이름 및 암호를 사용하여 자동으로 로그온" 옵션을 클릭합니다.
7. 변경 내용을 적용합니다.

   "보안 설정" 대화 상자와 "인터넷 옵션" 대화 상자를 닫습니다. 설정이 저장되고 자동 로그인이 구성됩니다.

ICAS 를 구현하는 방법

CA SiteMinder 는 Windows CardSpace 를 구현하는 ICAS(정보 카드 인증 체계)를 지원합니다. 보호된 리소스에 대한 액세스를 요청하는 사용자는 인증 카드를 선택할 수 있습니다. SiteMinder 는 이 카드에 포함된 정보를 사용하여 사용자의 아이덴티티를 확인합니다.

ICAS 를 구현하려면 다음과 같은 SiteMinder 구성 요소에서 구성 변경 작업이 필요합니다.

- SiteMinder 웹 에이전트를 호스트하는 서버
- SiteMinder 정책 서버
- smkey 데이터베이스

다음 단계를 수행하십시오.

1. 웹 서버에서 다음 태스크를 수행합니다.
   a. IIS 웹 서버에서 SSL 통신이 사용되도록 설정합니다.

      참고: 자세한 내용은 Microsoft 설명서 또는 http://support.microsoft.com/ 웹 사이트를 참조하십시오.

   b. 웹 서버 인증서를 .pfx 파일로 내보냅니다.
   c. SiteMinder InfoCard.fcc 템플릿을 사용자 지정합니다.
2. 정책 서버에서 다음 태스크를 수행합니다.
   a. 정책 서버에 JCE 를 설치합니다.
   b. 정책 서버의 java.security 파일을 업데이트합니다.
   c. 정책 서버의 config.properties 파일을 업데이트합니다.
   d. smkey 데이터베이스가 없으면 정책 서버 구성 마법사를 사용하여 새로 생성합니다.
   e. 웹 서버의 .pfx 파일 인증서를 smkey 데이터베이스에 추가합니다.
   f. 정책 서버에서 사용자 디렉터리를 구성합니다.
   g. 관리 UI 를 사용하여 CardSpace 를 위한 사용자 지정 인증 체계를 생성합니다.
   h. (선택 사항) 응답에 사용할 클레임을 세션 저장소에 저장합니다.
   i. (선택 사항) 세션 저장소에서 클레임 값 검색을 허용하여 개인 설정을 사용할 수 있도록 지정합니다.

j. (선택 사항) 저장된 클레임 값을 검색하기 위해 활성 응답을 구성합니다.

ICAS 를 위해 FCC 템플릿 구성

SiteMinder 웹 에이전트에는 SiteMinder 에서 ICAS 를 구현하는 데 사용할 수 있는 FCC(양식 자격 증명 수집기) 템플릿이 포함되어 있습니다.

다음 단계를 수행하십시오.

1. 텍스트 편집기에서 다음과 같은 기본 FCC 파일을 엽니다.

   web_agent_home\samples_default\forms\InfoCard.fcc

2. 이 파일의 *복사본*을 다음 디렉터리에 저장합니다. 복사본을 생성하면 나중에 필요할 경우를 대비해 기본 FCC 설정을 보존할 수 있습니다.

   web_agent_home\samples\forms\

3. FCC 파일 복사본에서 다음 정보를 기록합니다.

   중요! 정책 서버를 구성할 때 이 정보가 필요합니다.

   - 웹 에이전트를 호스트하는 IIS 웹 서버의 정규화된 도메인 이름
   - 2 단계에서 저장한 FCC 파일의 이름
   - 2 단계에서 저장한 FCC 파일에서 requiredClaims 매개 변수 태그의 값(따옴표 *제외*). 다음 예를 참조하십시오.

     ```
     http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier
     ```

   - (선택 사항) 허용 목록 처리가 완료된 경우 LOA(보증 수준) requiredClaims 매개 변수 태그의 값. 다음 예를 참조하십시오.

     ```
     < param name="requiredClaims" value="
     http://idmanagement.gov/icam/2009/09/imi_1.0_profile#assurancelevel1 "/>
     ```

     다음은 여러 LOA 의 URI 입니다.

     ```
     http://idmanagement.gov/icam/2009/09/imi_1.0_profile#assurancelevel1
     http://idmanagement.gov/icam/2009/09/imi_1.0_profile#assurancelevel2
     http://idmanagement.gov/icam/2009/09/imi_1.0_profile#assurancelevel3
     ```

4. (선택 사항) 텍스트 편집기를 사용하여 FCC 파일 복사본을 다음과 같이 변경합니다.

   - 사용자 지정 로고를 사용하려면 netegrity_logo.gif 파일을 원하는 그래픽으로 바꾸고 FCC 파일에서 다음 링크를 이에 맞게 업데이트합니다.

     ```
     <img alt="Logo" src="/siteminderagent/dmspages/netegrity_logo.gif">
     ```

IIS 용 SiteMinder 에이전트를 사용하는 경우 IIS 7.x 모듈 실행 순서 제어

IIS 웹 서버에 IIS 용 SiteMinder 에이전트를 설치하고 구성하는 경우 IIS 용 에이전트가 다른 모듈보다 먼저 실행됩니다. IIS 환경에서 다른 모듈을 먼저 실행해야 할 경우에는 Windows 레지스트리의 다음 위치에 설정된 번호를 변경하면 됩니다.

```
HKLM\SOFTWARE\Wow6432Node\Netegrity\SiteMinder Web Agent\Microsoft
IIS\RequestPriority
```

예를 들어 IIS 7.x 웹 서버의 다른 모듈(예: UrlScan)과 IIS 용 SiteMinder 에이전트에 지정된 실행 우선 순위가 같다고 가정해 봅니다. 이 설정을 사용하면 SiteMinder 모듈이 실행되는 시기를 제어할 수 있습니다.

다음 단계를 수행하십시오.

1. IIS 웹 서버에서 Windows 레지스트리 편집기를 엽니다.
2. 다음 키를 확장합니다.

   ```
   HKLM\SOFTWARE\Wow6432Node\Netegrity\SiteMinder Web Agent\Microsoft IIS
   ```

3. 다음 값을 찾습니다.

   ```
   RequestPriority
   ```

4. RequestPriority 값을 다음 중 원하는 값에 해당하는 번호로 변경합니다.

   PRIORITY_ALIAS_FIRST

   IIS 웹 서버의 다른 모듈보다 IIS 용 SiteMinder 에이전트를 먼저 실행합니다. 이 설정이 기본값입니다.

   예: 0(처음)

   기본값: 0

   PRIORITY_ALIAS_HIGH

   맨 처음 실행하도록 설정된 모듈과 우선 순위가 낮거나 중간 또는 마지막에 실행하도록 설정된 모듈 사이에 IIS 용 SiteMinder 에이전트 모듈을 실행합니다.

   예: 1(높음)

   PRIORITY_ALIAS_MEDIUM

   우선 순위가 높거나 맨 처음 실행하도록 설정된 모듈과 우선 순위가 낮거나 마지막에 실행하도록 설정된 모듈 사이에 IIS 용 SiteMinder 에이전트 모듈을 실행합니다.

   예: 2(중간)

PRIORITY_ALIAS_LOW

우선 순위가 높거나 중간 또는 맨 처음 실행하도록 설정된 모듈과 마지막에 실행하도록 설정된 모듈 사이에 IIS 용 SiteMinder 에이전트 모듈을 실행합니다.

예: 3(낮음)

PRIORITY_ALIAS_LAST

다른 모듈을 모두 실행한 *후에* IIS 용 SiteMinder 에이전트 모듈을 실행합니다.

예: 4(마지막)

5. 변경 내용을 저장하고 레지스트리 편집기를 닫습니다.
6. 설정을 테스트하고 원하는 모듈이 IIS 용 에이전트 모듈보다 먼저 실행되는지 확인합니다.

IIS 프록시 사용자 계정 사용(IIS 만 해당)

SiteMinder 로 보호되는 IIS 웹 서버의 리소스에 대한 IIS 권한이 없는 사용자가 해당 리소스에 액세스하려는 경우 웹 에이전트가 액세스를 거부할 수 있습니다. 예를 들어 UNIX 시스템의 LDAP 사용자 디렉터리에 저장된 사용자는 IIS 웹 서버를 사용하여 Windows 시스템에 액세스하지 못할 수 있습니다.

IIS 웹 서버에는 SiteMinder 에서 액세스 권한이 부여된 사용자에 대한 권한을 가진 기본 프록시 계정이 있습니다. 웹 에이전트는 사용자에게 유효한 Windows 보안 컨텍스트가 있는 경우에도 DefaultUserName 및 DefaultPassword 매개 변수의 값을 자격 증명으로 사용합니다.

다음 단계를 수행하십시오.

1. ForceIISProxyUser 매개 변수의 값을 다음 중 *하나*로 설정합니다.
   - 사용자의 자격 증명을 기반으로 하여 IIS 서버의 응용 프로그램에 액세스하는 경우 ForceIISProxyUser 매개 변수의 값을 yes 로 설정합니다.
   - 사용자 역할을 대신하는 특정 계정(예: 프록시)을 기반으로 하여 IIS 서버의 응용 프로그램에 액세스하는 경우 ForceIISProxyUser 매개 변수의 값을 no 로 설정합니다.

   기본값: No
2. 다음과 같은 Windows 기능을 사용하지 *않는* 경우 3 단계를 계속합니다.
   - Windows 인증 체계
   - Windows 사용자 보안 컨텍스트
3. DefaultUserName 매개 변수에 프록시 사용자 계정의 사용자 이름을 입력합니다. 도메인 계정을 사용하는 경우 로컬 컴퓨터가 해당 도메인의 일부가 아니면 다음과 같은 구문을 사용하십시오.

   DefaultUserName=*Windows_domain\acct_with_admin_privilege*

   그렇지 않은 경우에는 사용자 이름만 지정합니다.
4. DefaultPassword 매개 변수에 기존 Windows 사용자 계정과 연결된 암호를 입력합니다.

   중요! 에이전트 구성 개체에 이 매개 변수를 설정하면 값을 암호화할 수 있으므로 안전합니다. 이 매개 변수를 로컬 구성 파일에 설정하면 매개 변수 값이 암호화되지 않은 일반 텍스트로 저장됩니다.

   IIS 프록시 계정이 구성됩니다.

익명 사용자 액세스 사용

사용자가 프록시 사용자로 액세스하지 않게 하려면 다음 매개 변수를 설정합니다.

UseAnonAccess

프록시 사용자의 자격 증명을 사용하지 않고 웹 응용 프로그램을 익명 사용자로 실행하도록 IIS 웹 에이전트에 지시합니다.

기본값: No

참고: 이 매개 변수는 IIS 웹 에이전트에만 적용됩니다.

익명 사용자 액세스가 사용되도록 설정하려면 UseAnonAccess 매개 변수를 yes 로 설정하십시오.

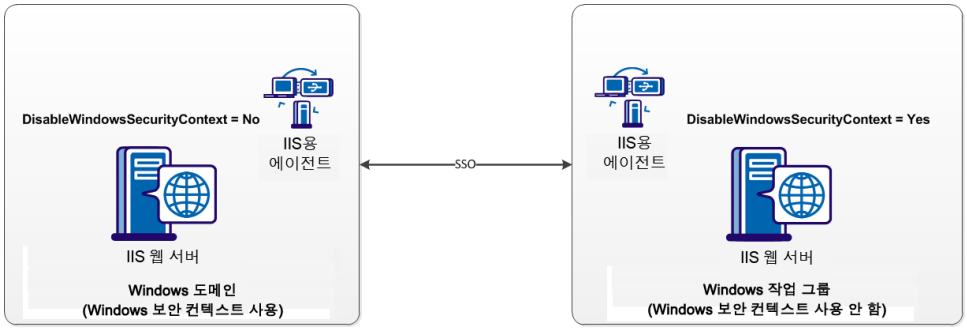
IIS 용 에이전트에서 Windows 보안 컨텍스트 사용 안 함

SiteMinder 정책 서버는 사용자 세션에서 Windows 보안 컨텍스트를 가져옵니다. 대부분의 경우 모든 에이전트에서 세션 정보를 사용할 수 있으므로 이 환경은 싱글 사인온에 적합합니다.

다음 예제에서는 싱글 사인온에 서로 다른 설정이 필요한 경우를 보여 줍니다.

- 한 SiteMinder 에이전트는 Windows 보안 컨텍스트를 사용합니다.
- 다른 SiteMinder 에이전트는 Windows 보안 컨텍스트를 사용하지 않습니다.

이러한 상황을 그림으로 나타내면 다음과 같습니다.



Windows 보안 컨텍스트를 사용하는 Windows 도메인과 Windows 보안 컨텍스트를 사용하지 않는 Windows 작업 그룹 간에 SSO 를 허용하려면 다음 매개 변수를 설정합니다.

DisableWindowsSecurityContext

에이전트에서 Windows 보안 컨텍스트가 사용되지 않도록 설정합니다. 이 매개 변수의 값이 yes 인 경우 에이전트는 사용자의 Windows 보안 컨텍스트를 무시합니다. 이 매개 변수의 값이 false 또는 no 인 경우 에이전트는 사용자 세션에 포함된 Windows 보안 컨텍스트를 사용합니다. 이 매개 변수는 보안 컨텍스트를 사용하는 Windows 환경과 사용하지 않는 Windows 환경 간의 싱글 사인온을 허용합니다.

기본값: False

제한: Yes, No

쿠키를 포함하는 서버 응답의 캐싱 방지

IIS 웹 서버는 출력 캐싱을 사용하여 응답을 저장합니다. 에이전트에 대한 응답은 쿠키를 포함합니다. IIS 웹 서버가 출력 캐시로부터 인증 응답을 보내는 경우, 다른 사용자가 캐싱된 응답에서 인증 쿠키를 받을 수 있습니다.

예를 들어, 사용자 1 이 성공적으로 인증하고 IIS 서버가 이 응답을 쿠키에 캐싱합니다. 이때 사용자 2 가 사용자 1 과 동일한 리소스에 액세스하는 경우 IIS 웹 서버는 사용자 1 에 대한 응답을 사용자 2 에게 반환할 수 있습니다.

기본적으로 쿠키를 포함하는 항목에 대한 IIS 출력 캐시는 비활성화되어 있습니다. 이전 제품 버전과의 호환성을 위해 이러한 속성을 되돌리려면 다음 매개 변수의 값을 no 로 변경하십시오.

IISCacheDisable

IIS 웹 서버가 쿠키를 포함하는 응답을 출력 캐시에 저장하는지 여부를 지정합니다. IIS 웹 서버는 SiteMinder 처리가 수행되기 전에 캐시된 응답을 전달합니다. 출력 캐시를 비활성화하면 IIS 가 각 트랜잭션을 인증 및 권한 부여하도록 만듭니다. 이 매개 변수의 값을 yes 로 설정하면 의도하지 않은 사용자에게 실수로 인증 또는 권한 부여 응답이 전달되는 것을 방지할 수 있습니다.

기본값: Yes (캐시 비활성화됨)

IIS 용 에이전트에서 쿠키를 설정해야 하는 경우 확인

IIS 용 SiteMinder 에이전트는 IIS 7.x 웹 서버가 제공하는 ARR(응용 프로그램 요청 라우팅) 기능을 지원합니다. ARR 은 Microsoft IIS 웹 서버에서 작동하며 다른 웹 서버 공급업체에서 제공하는 리버스 프록시 서버 기능과 비슷합니다.

모든 SiteMinder 에이전트는 쿠키를 처리합니다. 다음과 같은 조건에서는 쿠키 처리가 수행되는 시기를 제어해야 합니다.

- ARR 이 사용됩니다.
- FCCCompatMode 에이전트 구성 매개 변수의 값이 yes 입니다.
- SiteMinder SDK 로 개발된 사용자 지정 에이전트를 사용하고 있습니다.

에이전트가 쿠키를 처리하는 시기를 제어하면 SiteMinder 보호 수준을 적용하여 보안이 유지됩니다.

일부 SiteMinder 에이전트 배포 환경에서는 트랜잭션의 특정 시점에 SiteMinder 쿠키를 처리해야 합니다. 모든 SiteMinder 에이전트는 쿠키를 사용하고 처리합니다. 어떤 경우에는 트랜잭션에서 초기에 쿠키를 처리해야 합니다. 또는 쿠키를 나중에 처리해야 할 수도 있습니다. 적절한 시점에 쿠키를 처리하면 SiteMinder 가 리소스를 적절하게 보호할 수 있습니다.

중요! 잘못된 시간에 쿠키를 처리하면 보호 수준에 영향을 줍니다. ARR 기능이 추가 처리를 수행하도록 하려면 SiteMinder 에이전트가 쿠키를 처리하는 상대 시간을 변경해야 합니다.

다음 단계를 수행하십시오.

1. 관리 UI 에서 원하는 에이전트 구성 개체를 엽니다.
2. 다음 매개 변수를 찾습니다.

   EarlyCookieCommit

   쿠키를 처리 과정의 초기 시점에 설정할지 또는 나중에 설정할지 지정합니다. 다음과 같은 조건이 있을 경우 이 매개 변수 값을 yes 로 설정합니다.

   - IIS 웹 서버가 ARR(응용 프로그램 요청 라우팅)을 사용합니다.
   - FCCCompatMode 매개 변수의 값이 yes 입니다.
   - SiteMinder SDK 로 개발된 사용자 지정 에이전트를 사용하고 있습니다.

   이 값이 yes 일 경우 웹 에이전트가 OnAuthenticateRequest 또는 OnPostAuthenticateRequest 알림 메서드를 사용하여 요청을 처리한 후 초기에 쿠키를 커밋합니다.

   빠른 쿠키 처리가 필요한 사용자 지정 응용 프로그램에 대해 초기 SiteMinder 에이전트의 동작을 유지하려면 이 매개 변수 값을 yes 로 설정합니다.

   이 값이 no 일 경우 쿠키는 OnSendResponse 요청 알림 메서드에서 파이프라인이 끝날 때 나중에 응답으로 커밋됩니다.

   제한: IIS 7.x 용 에이전트에만 적용됩니다. 이 설정은 통합 파이프라인 모드를 사용하는 웹 응용 프로그램만 지원합니다.

   기본값: No. 쿠키가 OnSendResponse 요청 알림 메서드에서 나중에 설정됩니다.
3. 값 필드를 클릭하고 이전 매개 변수의 값을 yes 로 변경합니다.
4. "확인"을 클릭합니다.
5. "제출"을 클릭합니다.

   확인 메시지가 표시됩니다.

제 18 장: Apache 웹 서버 설정

이 섹션은 다음 항목을 포함하고 있습니다.

Apache 2.x 서버에서 HttpsPorts 매개 변수 사용

다음과 같은 조건의 경우 추가 웹 서버 구성이 필요합니다.

- HTTP_HOST 헤더의 값을 Apache 2.x 웹 서버로 변경하는 SSL 가속기 또는 기타 중간 장치를 사용합니다.
- HttpsPorts 매개 변수를 사용합니다.

다음 단계를 수행하십시오.

1. Apache 웹 서버의 httpd.conf 파일을 열고 다음과 같이 변경합니다.
   - UseCanonicalName 매개 변수의 값을 on 으로 변경합니다.
   - ServerName 매개 변수의 값을 다음 형식으로 변경합니다.

     server_name:*port_number*

     server_name

     SSL 가속기의 호스트 이름을 지정합니다.
2. 웹 에이전트의 경우 GetPortFromHeaders 매개 변수의 값을 yes 로 변경합니다.

Apache 웹 에이전트에서 레거시 응용 프로그램 사용

HTTP 1.1 을 지원하지 않는 레거시 응용 프로그램을 Apache 웹 서버에서 실행하려면 다음 매개 변수를 설정합니다.

LegacyTransferEncodingBehavior

웹 에이전트에서 사용하는 메시지 인코딩 유형을 지정합니다. 이 매개 변수의 값을 no 로 설정하면 전송 인코딩이 지원됩니다.

이 매개 변수의 값을 yes 로 설정하면 콘텐츠 인코딩이 사용됩니다. 전송 인코딩 헤더는 무시되고 콘텐츠 길이 헤더만 지원됩니다.

기본값: No

Apache 웹 서버에서 레거시 응용 프로그램을 사용하려면 LegacyTransferEncodingBehavior 매개 변수의 값을 yes 로 설정하십시오.

중요! 이 매개 변수의 값을 yes 로 설정하면 페더레이션 기능과 4KB 보다 긴 POST 데이터 보존 기능이 작동하지 않으며 큰 인증서가 인식되지 않을 수 있습니다.

HTTP HOST 요청을 사용하여 포트 번호 가져오기

실제 HTTP 헤더를 수정하지 *않고* 특정 웹 서버로 트래픽을 리디렉션하여 부하 분산을 수행하는 응용 프로그램을 사용 중인 경우에는 다음 매개 변수를 설정하여 부하 분산 장치에 사용되는 포트 대신 올바른 외부 포트로 사용자를 다시 리디렉션하도록 웹 에이전트를 구성해야 합니다.

GetPortFromHeaders

포트 번호를 웹 서버 서비스 구조가 아닌 HTTP HOST 요청 헤더에서 가져오도록 웹 에이전트에 지시합니다.

기본값: No

참고: 이 매개 변수는 Apache 웹 에이전트에 필요합니다.

Apache 웹 서버 로그에 트랜잭션 ID 기록

웹 에이전트는 사용자 권한 부여 요청이 성공할 때마다 고유한 트랜잭션 ID 를 생성합니다. 에이전트는 이 ID 를 HTTP 헤더에 추가합니다. 또한 다음 로그에도 ID 가 기록됩니다.

- 감사 로그
- 웹 서버 로그(서버가 쿼리 문자열을 기록하도록 구성된 경우)
- 정책 서버 로그

트랜잭션 ID 를 사용하면 지정된 응용 프로그램에 대한 사용자 활동을 추적할 수 있습니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

트랜잭션 ID 는 로그에서 기존 쿼리 문자열의 끝에 추가되는 모의 쿼리 매개변수로 나타납니다. 다음 예제에서는 STATE=MA 로 끝나는 쿼리 문자열에 추가된 트랜잭션 ID(굵게 표시됨)를 보여 줍니다.

```
172.24.12.1, user1, 2/11/00, 15:30:10, W3SVC, MYSERVER, 192.168.100.100, 26844,
47, 101, 400, 123, GET, /realm/index.html,
STATE=MA&SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1
```

URL 에 쿼리 매개 변수가 없는 경우 에이전트는 웹 서버 로그 항목의 끝에 트랜잭션 ID 를 추가합니다. 예를 들면 다음과 같습니다.

```
172.24.12.1, user1, 2/11/00, 15:30:10, W3SVC, MYSERVER, 192.168.100.100, 26844,
47, 101, 400, 123, GET, /realma/index.html,
SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1.
```

참고: 웹 에이전트는 사용자가 리소스에 액세스할 때 사용자 이름 및 액세스 정보를 네이티브 웹 서버 로그 파일에 기록합니다.

SiteMinder 트랜잭션 ID 를 Apache 웹 서버 로그의 SMTRANSACTIONID 헤더 변수에 기록할 수 있습니다.

다음 단계를 수행하십시오.

1. httpd.conf 파일을 엽니다.
2. SM_TRANSACTIONID 헤더 변수를 LogFormat 지시문에 추가합니다.

   예를 들면 다음과 같습니다.

   ```
   LogFormat "%h %l %u %t \"%r\" %>s %b \"%{SM_TRANSACTIONID}i\"" common
   ```

   참고: httpd.conf 파일 및 LogFormat 지시문에 대한 자세한 내용은 Apache 웹 서버 설명서를 참조하십시오.
3. 서버를 다시 시작하여 변경 내용을 적용합니다.

   트랜잭션 ID 가 Apache 웹 서버 로그에 기록됩니다.

POST 요청에서 콘텐츠 유형이 전송되는 방식 선택

Apache 웹 서버를 사용하는 경우 다음 매개 변수를 설정하여 POST 요청 중에 콘텐츠가 서버에 전송되는 방식을 제어할 수 있습니다.

LegacyStreamingBehavior

POST 요청 중에 콘텐츠가 서버에 전송되는 방식을 지정합니다. 이 매개 변수의 값을 yes 로 설정하면 다음을 *제외한* 모든 콘텐츠 유형이 스트리밍됩니다.

- text/xml
- application/x-www-form-urlencoded

이 매개 변수의 값을 no 로 설정하면 모든 콘텐츠 유형이 스풀링됩니다.

기본값: No

POST 요청에서 대부분의 콘텐츠 유형을 스트리밍하려면 LegacyStreamingBehavior 매개 변수의 값을 yes 로 변경합니다.

Apache 오류 로그에 기록되는 IPC 세마포 관련 메시지 출력 제한

기본적으로 Apache 웹 에이전트는 Apache 로깅 수준 구성에 관계없이 모든 수준(정보 및 오류)의 IPC 세마포 관련 메시지를 Apache 오류 로그에 기록합니다.

Apache 오류 로그에 출력되는 웹 에이전트 IPC 세마포 관련 메시지의 세부 정보 표시 수준을 제한하려면 *web_agent_home*/config 에 있는 trace.conf 파일에 다음 매개 변수를 추가합니다.

nete.stderr.loglevel

웹 에이전트가 Apache 오류 로그에 기록하는 IPC 세마포 관련 메시지의 수준을 지정합니다. 다음과 같은 값을 지정할 수 있습니다.

off

웹 에이전트가 IPC 세마포 관련 메시지를 Apache 오류 로그에 기록하지 않습니다.

error

웹 에이전트가 IPC 세마포 관련 오류 메시지만 Apache 오류 로그에 기록합니다.

info

(기본값) 웹 에이전트가 IPC 세마포 관련 오류 및 정보 메시지를 Apache 오류 로그에 기록합니다.

예: trace.conf 파일에 nete.stderr.loglevel 매개 변수 정의

trace.conf 파일의 다음 코드 조각에는 웹 에이전트가 IPC 세마포 관련 *오류* 메시지만 Apache 오류 로그에 기록하도록 nete.stderr.loglevel 매개 변수가 구성되어 있습니다.

```
# CA Web Agent IPC logging levels
# nete.stderr.loglevel=error
```

Stronghold 서버에서 인증서 삭제(Apache 에이전트만 해당)

Stronghold 웹 서버는 웹 에이전트에서 인증서 기반 인증을 수행할 때 사용하기 위해 클라이언트 인증서를 임시 로컬 파일에 기록합니다. Stronghold 서버는 이 파일을 사용하여 클라이언트 인증서의 정보를 인증에 사용할 수 있도록 만듭니다. 사용자가 웹 사이트를 방문하면 이러한 인증서 파일이 커지고 서버 공간을 차지하게 됩니다. 에이전트에서 사용이 끝난 인증서 파일을 삭제하도록 웹 에이전트를 구성할 수 있습니다.

인증서 파일을 삭제하려면 DeleteCerts 매개 변수를 yes 로 설정하십시오.

Oracle iPlanet 웹 서버 설정

다음 설정을 사용하면 SiteMinder 에이전트 Oracle iPlanet 서버를 관리할 수 있습니다.

Oracle iPlanet 웹 서버에서 디렉터리 검색 제한

사용자가 Oracle iPlanet 웹 서버의 디렉터리를 검색하려는 경우 SiteMinder 에서 인증 요청을 받도록 하려면 다음 매개 변수를 설정합니다.

DisableDirectoryList

웹 에이전트가 먼저 사용자의 인증을 요청하지 않고 사용자가 디렉터리의 콘텐츠를 보거나 탐색하도록 허용할지 여부를 지정합니다. 다음 조건이 *모두* 해당될 때 이 매개 변수를 사용합니다.

- 영역이 루트 리소스(/)를 보호하도록 설정된 경우
- 디렉터리의 기본 웹 페이지(예: index.html)가 이름이 바뀌거나 삭제된 경우

기본값: No

Oracle iPlanet 서버에서 디렉터리 검색을 제한하려면

1. DisableDirectoryList 매개 변수를 에이전트 구성 개체 또는 로컬 구성 파일에 추가합니다.
2. DisableDirectoryList 매개 변수의 값을 yes 로 설정합니다.

   디렉터리 검색이 제한됩니다. SiteMinder 에서는 디렉터리를 검색하려는 사용자에게 인증을 요청합니다.

Oracle iPlanet 웹 서버에 대한 여러 개의 AuthTrans 기능 처리

AuthTrans 기능은 Oracle iPlanet 웹 서버를 초기화하는 지시문입니다. Oracle iPlanet 웹 서버는 AuthTrans 기능을 obj.conf 파일에 나열된 순서대로 실행합니다. Oracle iPlanet 서버는 REQ_PROCEED 명령을 반환하는 기능을 찾을 때까지 AuthTrans 기능을 읽습니다. REQ_PROCEED 명령이 실행되면 다른 AuthTrans 기능이 실행되지 않습니다.

기본적으로 SiteMinder 는 첫 번째 AuthTrans 기능이고 REQ_PROCEED 를 반환합니다. 다른 AuthTrans 기능이 실행될 수 있게 하려면 EnableOtherAuthTrans 매개 변수를 추가하고 값을 yes 로 설정해야 합니다.

이 매개 변수의 기본값은 no 입니다. AuthTrans 기능을 여러 개 사용할 수 있게 하려면 EnableOtherAuthTrans 매개 변수를 yes 로 설정합니다.

이 매개 변수를 추가하면 SiteMinder 웹 에이전트를 다른 기능과 함께 사용할 수 있습니다.

SiteMinder 에이전트 기능은 obj.conf 파일에서 AuthTrans 지시문의 첫 번째 항목입니다. 다음과 같이 항목을 지정해야 합니다.

```
AuthTrans fn="SiteMinderAgent"
```

Oracle iPlanet 웹 서버 로그에 트랜잭션 ID 기록

Solaris 에 해당

웹 에이전트는 사용자 권한 부여 요청이 성공할 때마다 고유한 트랜잭션 ID 를 생성합니다. 에이전트는 이 ID 를 HTTP 헤더에 추가합니다. 또한 다음 로그에도 ID 가 기록됩니다.

- 감사 로그
- 웹 서버 로그(서버가 쿼리 문자열을 기록하도록 구성된 경우)
- 정책 서버 로그

트랜잭션 ID 를 사용하면 지정된 응용 프로그램에 대한 사용자 활동을 추적할 수 있습니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

트랜잭션 ID 는 로그에서 기존 쿼리 문자열의 끝에 추가되는 모의 쿼리 매개 변수로 나타납니다. 다음 예제에서는 STATE=MA 로 끝나는 쿼리 문자열에 추가된 트랜잭션 ID(굵게 표시됨)를 보여 줍니다.

```
172.24.12.1, user1, 2/11/00, 15:30:10, W3SVC, MYSERVER, 192.168.100.100, 26844,
47, 101, 400, 123, GET, /realm/index.html,
STATE=MA&SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1
```

URL 에 쿼리 매개 변수가 없는 경우 에이전트는 웹 서버 로그 항목의 끝에 트랜잭션 ID 를 추가합니다. 예를 들면 다음과 같습니다.

```
172.24.12.1, user1, 2/11/00, 15:30:10, W3SVC, MYSERVER, 192.168.100.100, 26844,
47, 101, 400, 123, GET, /realma/index.html,
SMTRANSACTIONID=0c01a8c0-01f0-38a47152-01ad-02714ae1.
```

참고: 웹 에이전트는 사용자가 리소스에 액세스할 때 사용자 이름 및 액세스 정보를 네이티브 웹 서버 로그 파일에 기록합니다.

SiteMinder 트랜잭션 ID 를 Oracle iPlanet 웹 서버 로그에 기록할 수 있습니다.

다음 단계를 수행하십시오.

1. magnus.conf 파일을 엽니다.
2. 웹 서버가 초기화될 때 기록할 기존 HTTP 서버 변수 목록에 다음 헤더 변수를 추가합니다.

   ```
   %Req->headers.SM_TRANSACTIONID%"
   ```

   참고: 에이전트 구성 개체 또는 로컬 구성 파일에서 LowerCaseHTTP 매개 변수의 값이 yes 로 설정된 경우가 아니면 헤더 변수를 대문자로 입력하십시오.

   다음 예제에서는 SMTRANSACTIONID 헤더 변수가 기존 항목의 끝에 추가되어 굵게 표시됩니다. 그러나 변수 목록에서 원하는 위치에 헤더 변수를 추가할 수 있습니다.

   ```
   Init fn="flex-init" access="D:/iPlanet/server4/https-orion/logs/access"
   format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
   \" %Req->srvhdrs.clf-status% %Req-srvhdrs.content-length% %Req->headers.-
   SM_TRANSACTIONID%"
   ```

3. Oracle iPlanet 서버를 다시 시작하여 변경 내용을 적용합니다.

   트랜잭션 ID 가 Oracle iPlanet 웹 서버 로그에 나타납니다. 다음 예제에서는 트랜잭션 ID 가 굵게 표시된 웹 서버 로그 항목을 보여 줍니다.

   ```
   11.22.33.44 - user1 [21/Nov/2003:16:12:24 -0500] "GET /Anon/index.html HTTP/1.0"
   200 748 3890b4b9-58f8-4a74df53-07f6-0002df88
   ```

추가 정보:

HTTP 헤더에 소문자 사용(Oracle iPlanet, Apache 및 Domino 웹 서버) (페이지 160)

Domino 웹 서버 설정

Domino 서버에 특별한 SiteMinder 에이전트 매개 변수가 필요한 경우가 있습니다. 별도의 설명이 없으면 이러한 매개 변수는 Domino 서버에만 사용됩니다. 다음 범주의 항목을 사용하여 Domino 리소스를 보호할 수 있습니다.

- 일반적인 정보를 보려면 다음 항목을 참조하십시오.
  - Domino 에이전트 개요 (페이지 349)
  - Domino URL 구문 (페이지 350)
  - Domino 별칭 (페이지 351)
- 기본 구성 정보를 보려면 다음 항목을 참조하십시오.
  - Domino 에이전트 구성 (페이지 352)
  - Domino 전용 에이전트 기능 구성 (페이지 353)
  - Domino 의 사용자 디렉터리 지정 (페이지 353)
  - Domino 서버에서 정책을 생성하기 위한 지침 (페이지 354)
  - Domino 의 정책 구성 (페이지 355)
  - Domino 서버 리소스에 대한 규칙 만들기 (페이지 356)
- SiteMinder 인증에 대한 내용은 다음 항목을 참조하십시오.
  - Domino 서버를 사용하여 사용자 인증 (페이지 359)
  - Domino 슈퍼 사용자로 인증 (페이지 360)
  - 실제 사용자 또는 기본 사용자로 인증 (페이지 360)
  - Domino 기본 사용자 및 Domino 슈퍼 사용자 수정 (페이지 361)
  - Encryptkey 도구를 사용하여 Domino 기본 사용자 또는 슈퍼 사용자 설정 (페이지 362)
  - SiteMinder 와 Domino 인증 조정
  - SiteMinder 에서 사용자를 인증하도록 지정 (페이지 363)
  - 인증에 SiteMinder 헤더 사용 (페이지 364)
  - Domino 세션 인증 사용 안 함 (페이지 364)
  - Domino 에서 익명 SiteMinder 인증 체계 사용 (페이지 365)
- SiteMinder FCC(양식 자격 증명 수집기) 사용에 대한 내용은 다음 항목을 참조하십시오.

- Domino 에이전트를 사용하여 인증에 필요한 자격 증명을 수집할 수 있도록 설정 (페이지 365)
- Domino 에이전트를 사용하여 FCC 리디렉션을 위한 URL 매핑 (페이지 185)
- URL 정규화 사용 안 함 (페이지 367)

■ Lotus Notes 문서 액세스를 관리하는 데 대한 내용은 다음 항목을 참조하십시오.

- Lotus Notes 문서에 대한 액세스 제어 (페이지 369)
- Lotus Notes 문서 이름 변환 (페이지 370)

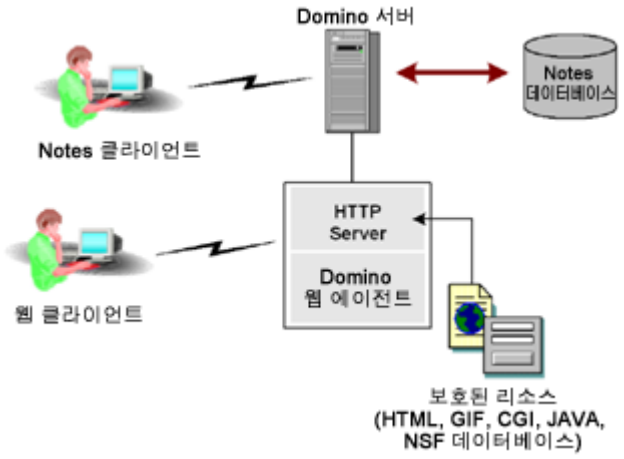
■ 위의 목록에 포함되지 않은 주제에 대한 내용은 다음 항목을 참조하십시오.

- Domino 에이전트에 대한 전체 로그아웃 지원 구성 (페이지 371)
- Domino 에이전트를 WebSphere 응용 프로그램 서버와 함께 사용 (페이지 372)

Domino 에이전트 개요

Domino 응용 프로그램 서버는 Lotus Notes 클라이언트에 보안 액세스를 제공하는 메시징 및 웹 응용 프로그램 플랫폼입니다. Domino 웹 에이전트는 Domino 응용 프로그램 서버의 HTTP 인터페이스만 보호하며 HTML, JAVA, CGI 및 기타 웹 리소스(예: 웹을 통해 제공되는 Notes)에 대한 액세스를 제어합니다. Notes 서버는 보호되지 않습니다.

다음 그림에서는 Domino 웹 에이전트가 Domino 서버와 통합되는 방식을 보여 줍니다.



Domino 는 데이터를 Notes 데이터베이스 그룹에 저장합니다. Notes 데이터베이스의 리소스는 문서, 뷰, 폼, 탐색기 등과 같은 다양한 개체가 될 수 있습니다. 이러한 개체에는 텍스트, 비디오, 그래픽 및 오디오 콘텐츠가 포함될 수 있습니다.

Notes 개체는 URL 을 사용하여 열 수 있습니다. Domino 는 Notes 개체를 웹에서 사용할 수 있도록 하기 위해 Notes 데이터베이스의 개체에서 동적으로 웹 페이지를 생성합니다. 또한 데이터베이스 뷰의 경우 뷰의 문서에 대한 URL 링크도 생성합니다. Notes 데이터베이스에서 동적으로 페이지가 생성되므로 사용자는 최신 정보를 볼 수 있습니다.

Domino URL 구문

Domino 서버의 리소스에 대한 액세스는 URL 을 기반으로 합니다. Domino 서버는 특정한 URL 구문을 사용합니다.

Domino 서버는 다음과 같은 표준 URL 을 해석할 수 있습니다.

```
http://www.example.com/index.html
```

Domino URL 명령은 다음과 같은 구문을 사용할 수 있습니다.

```
http://host/database.nsf/Domino_object?Action_Argument
```

Host

서버의 DNS 항목 또는 IP 주소를 나타냅니다.

Database

notes \data 디렉터리에 대한 상대 경로가 포함된 데이터베이스 파일 이름 또는 데이터베이스 복제본 ID 를 지정합니다.

Domino_object

데이터베이스의 개체, 즉 뷰, 문서, 양식 또는 탐색기 등을 지정합니다.

작업

Notes 개체에 대해 수행한 작업을 식별합니다. 예를 들면 ?OpenDatabase, ?OpenView, ?OpenDocument, ?OpenForm, ?ReadForm, ?EditDocument 등을 지정할 수 있습니다. URL 에 작업을 지정하지 않으면 기본값이 사용됩니다.

기본값: ?Open

Argument

Domino 서버에서 개체를 제공하는 방식을 정의합니다. 예를 들어 ?OpenView&Expand=5 라는 작업 및 인수가 제공된 경우 이 인수는 확장 형식으로 표시되는 행의 수를 지정합니다.

다음 예제에서는 financials.nsf 라는 Notes 데이터베이스의 뷰에 액세스하기 위한 URL 을 보여 줍니다.

```
http://www.example.com/financials.nsf/reports?OpenView
```

Domino 별칭

Notes 데이터베이스 규칙 중 하나는 개체의 별칭을 생성하는 것입니다. 예를 들어 별칭은 개체 이름 대신 Notes ID 또는 복제본 ID 로 리소스를 식별할 수 있습니다. 별칭을 사용하면 코드를 바꾸지 않고도 Notes 리소스의 이름을 변경할 수 있으므로 개발자가 프로그래밍을 더 쉽게 할 수 있습니다.

다음 Domino URL 은 별칭으로 식별되는 동일한 리소스에 액세스합니다.

- http://www.domino.com/85255e01001356a8852554c20756?OpenView
- http://www.domino.com/85267E00075A80C/people?OpenView
- http://www.domino.com/__852567E00075A80C.nsf/people?OpenView

리소스가 식별되는 방식에 관계없이 Domino 웹 에이전트는 모든 Domino 명명 규칙을 데이터베이스 리소스 이름을 사용하는 표준 URL 로 변환합니다. 이렇게 하면 SiteMinder 정책 저장소에 데이터 입력이 간소화됩니다.

예를 들어 다음 Domino URL 은 names.nsf 데이터베이스의 people 뷰를 가리킵니다. 해당 데이터베이스와 뷰는 복제본 ID 및 Notes ID 로 참조됩니다.

- http://www.domino.com/85255e01001356a8852554c20756?OpenView
- http://www.domino.com/85267E00075A80C/people?OpenView

Domino 웹 에이전트는 이러한 URL 을 다음과 같이 표준 URL 로 변환합니다.

- http://www.domino.com/names.nsf/people?OpenView

다음 그림에서는 별칭을 명명된 개체로 변환하는 과정을 보여 줍니다.

Domino 웹 에이전트 구성

Domino 웹 에이전트는 웹 에이전트의 모든 표준 설정을 사용하여 다음을 수행합니다.

- 웹 에이전트가 정책 서버와 통신하도록 구성
- 가상 서버에 대한 에이전트 아이덴티티 추가/제거

- 웹 에이전트 설정 수정
- 싱글 사인온 구성
- 오류 메시지 로깅 구성

정책 서버를 사용하여 중앙에서 또는 에이전트 구성 파일을 사용하여 로컬로 이러한 설정을 구성할 수 있습니다.

표준 기능뿐만 아니라 Domino 전용 매개 변수도 설정할 수 있습니다.

추가 정보

Domino 전용 에이전트 기능 구성 (페이지 353)

Domino 전용 에이전트 기능 구성

웹 에이전트 표준 설정 외에 Domino 웹 에이전트에만 설정할 수 있는 Domino 전용 구성 매개 변수도 있습니다. 이러한 설정은 Domino 에서 SiteMinder 를 통해 사용자를 인증하고 권한을 부여하는 방식을 결정합니다. 이러한 설정은 정책 서버의 에이전트 구성 개체를 사용하여 중앙에서 구성하거나 웹 서버의 에이전트 구성 파일을 사용하여 로컬로 구성할 수 있습니다.

참고: Domino 웹 에이전트는 사용자 활동을 추적하는 데 사용되는 감사 기능을 지원하지 않습니다.

Domino 의 사용자 디렉터리 지정

Domino Directory 는 모든 Domino 서버와 통합됩니다. Domino 서버의 LDAP 서비스를 활성화하면 정책 서버에서 Domino Directory 를 사용하여 사용자를 인증하고 권한을 부여할 수 있습니다. Domino 의 LDAP 서비스를 활성화하면 인증을 위해 사용자 디렉터리를 별도로 구성할 필요가 없습니다.

LDAP 서비스를 활성화하려면 Domino 서버 설명서를 참조하십시오.

추가 정보:

CA 에 문의 (페이지 4)

Domino 서버에서 정책을 생성하기 위한 지침

Domino 서버에 대한 SiteMinder 정책을 생성하는 경우 다음 지침을 고려하십시오.

- 사용자는 부모 문서와 함께 양식을 열어 해당 양식의 기본값을 볼 수 있습니다. 부모 문서는 문서를 생성하는 데 사용되는 원본 양식입니다. 권한이 없는 사용자가 액세스 권한이 없는 양식의 기본값을 볼 수 없게 하려면 SkipDominoAuth 매개 변수를 no 로 설정하십시오.
- 같은 컴퓨터의 데이터베이스를 복제하는 경우 중복된 규칙 집합을 생성하여 각 데이터베이스를 보호해야 합니다.
- Domino 에이전트에서 Notes 문서의 별칭을 양식과 연결할 수 없는 경우 각 문서에 고유한 보호 규칙이 필요합니다.
- Domino 서버는 특정 데이터베이스 문서에 대한 URL 명령에 $DefaultView, $DefaultForm, $DefaultNav, $SearchForm 등의 특수한 식별자를 사용합니다. Domino 에이전트는 이러한 식별자를 표준 URL 로 변환하여 문서에 액세스합니다.

  $defaultNav 의 경우 Domino 에이전트는 ?OpenDatabase 작업을 수행합니다. 이러한 식별자 유형에 대한 추가 규칙을 생성할 필요는 없습니다.
- Notes 데이터베이스의 별칭은 연결된 리소스를 보호합니다. 별칭이 없으면 리소스 이름 또는 설명을 지정하여 연결된 리소스를 보호합니다.
- Lotus Notes 소프트웨어에서는 유형이 다른 여러 개체에 같은 이름 및 별칭을 지정할 수 있습니다. ?Open*와 같이 ?Open 작업에 와일드카드를 사용하는 규칙을 생성하면 별칭 또는 이름을 공유하는 서로 다른 유형의 리소스를 보호할 수 있습니다.
- 양식은 해당 양식으로 생성하는 문서를 보호합니다. 양식과 함께 사용되는 작업은 ?ReadForm 입니다.
- Domino 에이전트는 확장명이 .nsf 인 파일을 보호합니다. 이 확장명을 IgnoreExt 매개 변수에 추가하지 마십시오.

Domino 의 정책 구성

Domino 서버는 같은 Notes 개체를 다양한 방법으로 나타낼 수 있습니다. 이름, ReplicaID, UniversalID 및 별칭을 사용하여 개체를 식별할 수 있습니다.

Domino 웹 에이전트가 Domino 서버와 효과적으로 통신할 수 있도록 하기 위해 Domino 에이전트는 Notes 리소스에 대한 액세스 요청을 처리할 때 개체 이름만 사용합니다. 이렇게 하면 SiteMinder 정책 저장소에서 항목을 이해할 수 있습니다.

특정 리소스에 대한 액세스 방법을 URL 로 나타내면 다음과 같습니다.

http://host/database.nsf/resource_name?Open

Domino 서버 리소스에 대한 규칙 생성

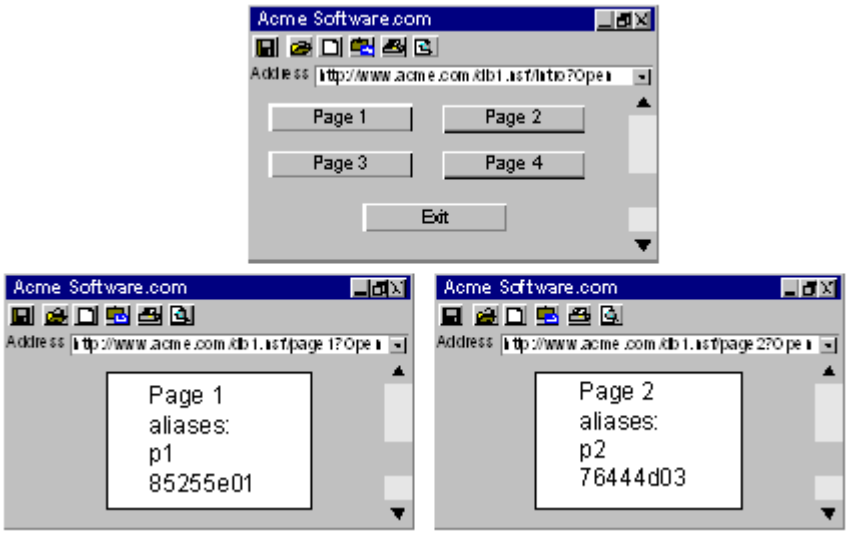
규칙을 생성할 때 Notes 데이터베이스 리소스에 대한 작업을 고려해야 합니다. 작업이 지정되지 않은 리소스에는 ?Open 작업이 기본으로 지정됩니다. SiteMinder 정책에 포함된 규칙은 기본 작업인 ?Open 및 ?Open 과
동등한 ?OpenDatabase, ?OpenView, ?OpenDocument, ?OpenFrameset 등의 작업을 설명해야 합니다.

Domino 웹 에이전트를 사용하면 정책 관리자는 같은 리소스를 가리키는 다양한 별칭에 대해 하나의 규칙을 생성할 수 있습니다. Domino 에이전트는 Domino 에서 같은 리소스를 나타내는 여러 표현을 하나의 URL 로 변환하므로 규칙이 하나만 필요합니다. Domino 에이전트의 이 기능은 SiteMinder 정책에 대한 규칙을 생성할 때 고려되어야 합니다.

관리 UI 를 사용하여 영역과 규칙을 생성합니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

다음 그림의 URL 은 db1.nsf 라는 Notes 데이터베이스가 포함된 Acme 의 Domino 서버에 대한 링크입니다. 이 데이터베이스는 page1 과 page2 라는 두 파일을 포함합니다.



예 1: 한 문서와 해당 문서의 모든 별칭 보호

page1 및 page1 의 모든 별칭에 액세스하기 위해 db1.nsf 영역에 대한 규칙을 하나만 생성합니다. Domino 에이전트는 다양한 명명 규칙을 모두 해석하고 하나의 표준 URL 형식으로 변환할 수 있습니다.

영역 및 규칙에 대해 다음을 수행하십시오.

- 영역을 만들 때 page1 이 포함된 데이터베이스에 대해 리소스 필터를 지정합니다. 예를 들어 데이터베이스의 모든 파일을 보호하려면 다음과 같이 구성합니다.

  리소스 필터: /db1.nsf/

  page1 뿐만 아니라 page1 의 모든 별칭을 보호하려면 다음과 같이 구성합니다.

  리소스 필터: /db1.nsf/page1

- page1 에 대한 작업을 보호하는 규칙을 생성하려면 "규칙 속성" 대화 상자의 "리소스" 필드에 별표(*)를 입력합니다. 예를 들면 다음과 같습니다.

  리소스: *

  이 와일드카드(*)는 정책에 바인딩된 사용자가 page1 에 대해 ?Open, ?EditDocument 등 모든 작업을 수행할 수 있음을 나타냅니다.

예 2: 같은 데이터베이스의 다른 문서 보호

db1.nsf 데이터베이스에서 page1 뿐만 아니라 page2 를 보호하려면 두 번째 규칙을 생성해야 합니다.

리소스 필터: /db1.nsf/page2

리소스: *

예 3: 같은 리소스에 대한 다른 작업 보호

일부 사용자만 ?EditDocument 작업을 수행할 수 있고 모든 사용자는 ?ReadForm 작업을 수행할 수 있는 경우와 같이 한 리소스에 대한 개별 작업을 보호하려면 다음과 같이 각 작업에 각 리소스에 대한 규칙이 필요합니다.

- 규칙 1

  리소스 필터: /db1.nsf/page1

리소스: OpenView

- 규칙 2

  리소스 필터: /db1.nsf/page1

  리소스: EditDocument

다음과 같이 하나의 규칙을 사용할 수도 있습니다.

리소스 필터: /db1.nsf/page

리소스: ?Open*

참고: "리소스" 필드의 ?Open 앞에 슬래시(/)가 없습니다.

이 리소스에 대한 별칭이 여러 개 있더라도 하나의 규칙으로 원래 페이지와 모든 별칭을 보호할 수 있습니다.

서로 다른 작업에 대한 규칙을 여러 개 생성하는 대신 단일 규칙을 지정하고 다음과 같이 와일드카드를 사용하여 모든 작업을 포함할 수 있습니다.

리소스 필터: /db1.nsf/page

리소스: ?Open*

이 규칙을 사용하여 다음과 같이 리소스를 보호합니다.

http://www.acme.com/db1.nsf/page*?Open*

참고: 리터럴 규칙을 지정하려면 정규식을 작성하십시오.

Domino 서버를 사용하여 사용자 인증

SiteMinder 에서 이미 이 프로세스를 통과한 경우에도 Domino 서버에서 사용자를 인증하고 권한을 부여해야 합니다. SiteMinder 는 사용자 및 사용자 권한 목록을 나타내는 Domino Directory 에 구성된 사용자 아이덴티티를 Domino 서버에 제공하여 Domino 인증 프로세스와 함께 작동합니다. Domino 서버는 이 아이덴티티를 사용하여 데이터베이스 리소스 액세스를 위해 사용자를 인증하고 권한을 부여합니다.

참고: 사용자 이름은 명확하게 확인되어야 합니다. 그렇지 않으면 Domino 에이전트에서 인증 요청을 거부합니다. 이로 인해 사용자 디렉터리에서 몇 가지 조정이 필요할 수 있습니다.

Domino 웹 에이전트는 Domino 서버 사용자를 다음 중 하나로 식별합니다.

- 슈퍼 사용자
- 실제 사용자
- 기본 사용자

Domino 서버와 통신할 때 Domino 웹 에이전트에서 사용하는 아이덴티티를 결정하려면 다음 매개 변수를 구성합니다.

SkipDominoAuth

서버 인증을 위해 Domino 서버에 전달할 이름을 결정합니다.

DominoSuperUser

Domino 서버의 모든 리소스에 액세스할 수 있는 사용자를 식별합니다.

DominoDefaultUser

Notes 데이터베이스에 대한 기본 액세스가 지정된 사용자, 즉 일반적인 액세스 권한을 가진 사용자를 식별합니다.

참고: DominoSuperUser 및 DominoDefaultUser 를 에이전트 구성 파일에 로컬로 구성하거나 에이전트 구성 개체를 사용하여 중앙에서 구성할 수 있습니다. 에이전트 구성 파일에서 이 설정의 값은 암호화되어 있습니다. 에이전트 구성 개체에서는 선택에 따라 이러한 값을 암호화하거나 일반 텍스트로 지정할 수 있습니다.

추가 정보

SiteMinder 에서 사용자를 인증하도록 지정 (페이지 363)
실제 사용자 또는 기본 사용자로 인증 (페이지 360)
Domino 슈퍼 사용자로 인증 (페이지 360)

Domino 슈퍼 사용자로 인증

Domino 슈퍼 사용자는 Domino 서버의 모든 리소스에 액세스할 수 있는 사용자입니다. SiteMinder 를 염두에 두고 웹 사이트 또는 포털을 디자인하는 경우에는 SiteMinder 정책을 구현하여 리소스와 응용 프로그램의 보안을 유지합니다. 따라서 Domino 서버는 Domino 자체의 보안 규칙으로 사용자 액세스를 제한할 필요가 없습니다. 이 경우 사용자는 Domino 인증을 위해 슈퍼 사용자로 식별될 수 있습니다.

사용자를 슈퍼 사용자로 식별하려면 SkipDominoAuth 매개 변수가 사용되도록 설정하고 DominoSuperUser 매개 변수의 값을 지정합니다. 이렇게 하면 SiteMinder 에서 사용자를 인증하고 Domino 에서는 사용자를 인증하지 않습니다. 또한 지정된 사용자는 Domino Directory 에 있어야 합니다.

실제 사용자 또는 기본 사용자로 인증

Domino Directory 에 정의된 사용자는 Domino 에서 해당 사용자의 이름으로 인증됩니다. 그러나 Domino Directory 에 없고 SiteMinder 에서 다른 사용자 디렉터리에 대해 인증한 사용자는 Domino 웹 에이전트에서 Domino 서버의 DominoDefaultUser 로 식별합니다.

Notes 데이터베이스에 대한 기본 액세스가 지정된 기본 사용자는 Domino 의 Depositor, Reader 또는 Author 액세스 수준과 같이 ACL 에 구성된 일반적인 액세스 권한을 갖습니다.

Domino 에이전트에서 이 값을 사용하려면 SkipDominoAuth 매개 변수를 no 로 설정합니다.

일부 Notes 데이터베이스의 경우 SiteMinder 의 보호가 필요하지 않을 수 있습니다. SiteMinder 에 의해 보호되지 않는 리소스는 기본 Domino 사용자로 인증되지 않습니다. 대신 Domino 서버에서 사용자에게 자격 증명을 요청합니다(익명 액세스를 사용할 수 없는 경우).

Domino 기본 사용자 및 Domino 슈퍼 사용자 수정

DominoDefaultUser 및 DominoSuperUser 매개 변수를 수정하려면 다음 작업 중 하나를 수행하십시오.

- 중앙에서 구성하는 경우 에이전트 구성 개체의 매개 변수를 변경합니다.

  에이전트 구성 개체에서 DominoDefaultUser 및 DominoSuperUser 설정을 수정할 수 있습니다. 이때 값을 암호화할지 아니면 일반 텍스트로 나타낼지 선택할 수 있습니다.

  참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

- encryptkey 도구를 사용하여 에이전트 구성 파일의 매개 변수를 수정합니다.

  에이전트 구성 파일에서는 DominoDefaultUser 및 DominoSuperUser 값을 암호화해야 합니다. 따라서 encryptkey 도구를 사용하여 값을 수정해야 합니다.

  중요! 에이전트 구성 파일에서 이러한 설정을 직접 편집하면 안 됩니다.

Encryptkey 를 사용하여 Domino 기본 사용자 또는 슈퍼 사용자 설정

에이전트 구성 파일에서 DominoSuperUser 또는 DominoDefaultUser 값을 설정하거나 변경하려면

1. 다음 작업 중 하나를 수행하십시오.
   - UNIX: Domino 에이전트의 bin 디렉터리를 탐색합니다. 예를 들면 다음과 같습니다.

     /$HOME/ca/SiteMinder/Web Agent/bin
   - Windows: 명령 프롬프트 창을 열고 Domino 에이전트의 Bin 디렉터리를 탐색합니다. 예를 들면 다음과 같습니다.

     C:\Program Files\ca\SiteMinder Web Agent\Bin
2. 다음 인수를 사용하여 encryptkey 도구를 실행합니다.
   - DominoSuperUser 의 경우

     encryptkey -path *path_to_Agent_config_file*

     -dominoSuperUser *new_value*
   - DominoDefaultUser 의 경우

     encryptkey -path *path_to_Agent_config_file*

     -dominoDefaultUser *new_value*

   예를 들면 다음과 같습니다.

   encryptkey -path "c:\program files\ca\SiteMinder Web Agent\Bin\Lotus Domino5\webagent.conf"

   -dominoSuperUser admin

   참고: 에이전트 구성 파일 경로에는 webagent.conf 와 같은 파일 이름이 포함되어야 합니다. 또한 경로 값에 공백이 포함된 경우에는 전체 경로를 따옴표로 묶어야 합니다.

   참고: encryptkey 도구는 SiteMinder 웹 에이전트 키트에 포함되어 있지 않습니다. 그러나 이 도구는 로컬 구성을 위해 암호화된 DominoSuperUser 설정을 생성하는 Domino 사용자에게 유용합니다. 이 도구를 다운로드하려면 고객 지원부에 문의하십시오.

SiteMinder 에서 사용자를 인증하도록 지정

Domino 가 아니라 SiteMinder 에서 사용자를 인증하도록 지정하려면 SkipDominoAuth 매개 변수를 yes 로 설정하십시오.

SkipDominoAuth 를 yes 로 설정하고 슈퍼 사용자가 정의되면 먼저 SiteMinder 에서 사용자를 식별하고 권한을 부여합니다. 그런 다음 Domino 웹 에이전트에서 해당 사용자를 Domino 서버의 슈퍼 사용자로 식별합니다. 슈퍼 사용자는 해당하는 ACL 이 있다는 가정 하에 Domino 서버의 모든 리소스에 액세스할 수 있습니다.

또한 사용자가 Domino Directory 에 저장되어 있지 않으면 권한 부여에 사용할 아이덴티티가 Domino 에 없으므로 SkipDominoAuth 매개 변수를 yes 로 설정해야 합니다.

SkipDominoAuth 를 no 로 설정하면 Domino 에서 실제 사용자 이름 또는 기본 사용자 이름을 사용하여 단독으로 사용자를 인증합니다.

다음 표에서는 SkipDominoAuth 매개 변수의 설정에 따라 사용자가 어떻게 식별되는지를 보여 줍니다.

SkipDominoAuth 값	Domino 서버에 대해 식별되는 사용자	참고
예	슈퍼 사용자	슈퍼 사용자가 Domino Directory 에 정의되어야 합니다.
없음	실제 사용자	사용자가 Domino Directory 에 있어야 합니다.
없음	기본 사용자	사용자가 Domino Directory 에 있어야 합니다.
없음	슈퍼 사용자	요청된 리소스에 자동으로 권한이 부여됩니다. 즉, 사용자에게 인증 요청이 제공되지 않습니다.

추가 정보

실제 사용자 또는 기본 사용자로 인증 (페이지 360)

인증에 SiteMinder 헤더 사용

DominoUseHeaderForLogin 및 DominoLookUpHeaderForLogin 매개 변수는 인증을 위해 Domino 사용자를 식별하는 데 사용될 수 있습니다.

DominoUseHeaderForLogin

SiteMinder 헤더 값을 Domino 웹 서버에 전달하도록 Domino 웹 에이전트에 지시합니다. Domino 서버는 헤더 데이터를 사용하여 사용자 디렉터리에서 사용자를 식별합니다.

이 매개 변수를 헤더 이름으로 설정해야 합니다. 예를 들어 DominoUseHeaderForLogin="HTTP_SM_USER"를 지정하면 웹 에이전트에서 사용자의 로그인 이름을 Domino 서버에 전달합니다.

DominoLookUpHeaderForLogin

리소스에 대한 액세스를 요청하는 사용자가 Domino 사용자 디렉터리 내에서 고유한지 여부를 Domino 웹 서버에 확인하도록 Domino 웹 에이전트에 지시합니다. Jones 라는 사용자가 리소스에 액세스하려는 경우 사용자 디렉터리에 이름이 같은 사용자가 여러 명 있으면 이러한 확인 작업이 유용합니다. 이 매개 변수가 no 로 설정되면 Domino 웹 에이전트는 Domino 웹 서버를 통해 확인을 수행하지 않습니다.

기본값: Yes

Domino 세션 인증 사용 안 함

SiteMinder 에서 인증 및 권한 부여 기능을 제공하므로 Domino 세션 인증 기능은 필요하지 않습니다. 웹 에이전트가 설치된 경우에는 Domino 세션 인증이 사용되지 않도록 설정해야 합니다.

경우에 따라 Domino 세션 인증이 사용되도록 설정하면 사용자 세션의 동작이 달라질 수 있습니다. 이러한 변경은 SiteMinder 를 사용하는 사이트의 보안에는 영향을 주지 않습니다. 이는 SiteMinder 와 Domino 세션 관리 규칙이 교차하는 부분을 반영합니다.

Domino 에서 익명 SiteMinder 인증 체계 사용

Domino 에이전트에서 익명 SiteMinder 인증 체계를 사용하려면 다음 매개 변수를 설정합니다.

DominoUserForAnonAuth

익명 사용자에 대한 값을 지정합니다. 이 값은 사용자가 익명 SiteMinder 인증 체계로 보호되는 Domino 리소스에 액세스할 때 Domino 서버에 전달됩니다.

기본값: No(익명 인증 체계 사용 *안 함*)

예: Anonymous(익명 인증 체계에 사용)

이 매개 변수는 Domino 에서 익명 SiteMinder 인증 체계를 사용하는 경우에만 적용됩니다. 사용되는 인증 체계 또는 서버 유형이 다른 경우에는 값을 변경하면 *안 됩니다*.

Domino 에이전트를 사용하여 인증에 필요한 자격 증명을 수집할 수 있도록 설정

자격 증명 수집기는 양식, SSL 및 Windows 인증 체계에 필요한 사용자 자격 증명 또는 여러 쿠키 도메인에서 싱글 사인온을 구현하는 데 필요한 사용자 자격 증명을 수집하는 응용 프로그램으로, 웹 에이전트에 포함되어 있습니다. 자격 증명 수집기에서 수집하는 자격 증명은 보호된 리소스의 특정 그룹에 대해 구성된 인증 체계 유형을 기반으로 합니다.

Domino 웹 에이전트를 자격 증명 수집기로 사용하려면 에이전트 구성 파일에 파일 확장명으로 표시되는 다양한 MIME 유형을 구성해야 합니다.

일반적으로 자격 증명 수집기는 자동으로 권한을 받습니다. 즉, 이러한 매개 변수에 파일 확장명을 추가하면 기본적으로 IgnoreExt 매개 변수에 확장명이 포함됩니다. Domino 서버는 이러한 확장명을 사용하는 파일이 포함된 URL 을 올바로 처리할 수 없으므로 Domino 에이전트에서 해당 파일을 무시해야 합니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

Domino 웹 에이전트를 사용하여 FCC 리디렉션을 위한 URL 매핑

양식 인증 체계를 사용하여 Domino 뷰(.nsf) 리소스를 보호하려면 양식 자격 증명 수집기로 리디렉션되기 전에 URL 을 매핑해야 합니다.

다음 단계를 수행하십시오.

1. DominoNormalizeUrls 매개 변수의 값을 yes 로 설정합니다.
2. DominoMapUrlForRedirect 매개 변수의 값을 yes 로 설정합니다.

   FCC 로 리디렉션되기 전에 Domino URL 이 매핑됩니다.

URL 정규화 사용 안 함

URL 정규화는 Domino 형식으로 표현된 URL 을 표준 웹 브라우저에 사용되는 URL 형식으로 수정하는 과정입니다. Domino 웹 에이전트는 Domino 웹 서버 API 를 사용하여 Domino URL 을 정규화합니다.

정규화를 수행하는 동안 Domino 서버 API 는 정규화된 URL 에 캐리지 리턴(16 진수 0x0D) 또는 줄 바꿈(16 진수 0x0A) 문자가 추가된 URL 을 주기적으로 반환합니다. 이러한 문자가 추가되는 것은 특정 Notes 데이터베이스 파일(.nsf) 및 해당 파일 내의 액세스 패턴과 관련이 있는 것 같습니다.

다음 예제에서는 캐리지 리턴이 추가되는 정규화된 URL 을 보여 줍니다.

- URL:
  http://server.ca.com:80/agentrunner.nsf/be68f4545348400461332?OpenView
- 매핑 대상 URL:
  http://server.ca.com:80/agentrunner.nsf/AgentContext?OpenView
- 정규화된 URL:
  http://xxxxx.ca.com:port/agentrunner.nsf/0x0d/AgentContext?OpenView

필요한 경우 다음 매개 변수를 사용하여 Domino 리소스 ID 가 포함된 URL 은 정규화되지 않도록 설정할 수 있습니다.

DominoNormalizeUrls

SiteMinder 웹 에이전트가 Domino URL 을 양식 자격 증명 수집기에 리디렉션하기 전에 친숙한 URL 이름으로 변환할지 여부를 지정합니다.

Domino URL 을 변환하려면 MapUrlsForRedirect 매개 변수도 yes 로 설정해야 합니다.

DominoNormalizeUrls 매개 변수를 no 로 설정하면 MapUrlsForRedirect 매개 변수를 yes 로 설정해도 URL 이 정규화되지 *않습니다.*

중요! DominoNormalizeUrls 매개 변수를 no 로 설정하면 Notes 데이터베이스 내의 개별 문서를 보호할 수 없으며 Domino 웹 서버의 하위 디렉터리 또는 전체 데이터베이스만 보호할 수 있습니다.

기본값: Yes

정규화를 해제하고 URL 이 변경되지 않게 하려면 DominoNormalizeUrls 매개 변수를 no 로 설정하십시오.

Lotus Notes 문서에 대한 액세스 제어

웹 에이전트는 Domino 의 Lotus Notes 문서를 보호하기 위해 좀 더 세분화된 수준을 제공합니다. 다음 매개 변수는 이러한 보호를 제어합니다.

DominoLegacyDocumentSupport

웹 에이전트가 Domino 환경에서 보호되는 Lotus Notes 문서에 대한 사용자 요청을 처리하는 방식을 지정합니다. 이 매개 변수를 yes 로 설정하면 사용자에게 요청된 문서에 대해서만 ReadForm 권한이 부여됩니다.

기본값: No

DominoLegacyDocumentSupport 매개 변수를 사용하면 Notes 문서에 액세스할 때 사용자가 요청한 작업을 처리하도록 웹 에이전트를 구성할 수 있습니다. 이 기능을 통해 Domino 에 대한 보호 수준을 좀 더 세분화할 수 있습니다.

Notes 문서에는 이름이 없습니다. Notes 문서는 해당 문서를 생성할 때 사용된 폼에 대한 참조와 함께 데이터베이스에 저장됩니다. 사용자가 Notes 문서를 요청할 경우 Domino 웹 에이전트는 요청을 URL 로 변환하여 해당 문서에 대한 폼을 찾습니다. 이 URL 은 원래 Domino 작업을 포함합니다. 폼이 없을 경우에는 아무 것도 사용되지 않습니다.

예를 들면 다음과 같습니다.

"http://server.domain.com/db.nsf?OpenDocument"

URL 에 지정된 문서에 대해 ?OpenDocument 또는 ?EditDocument 와 같이 사용자가 요청한 Domino 작업을 수행하도록 웹 에이전트를 구성하려면 DominoLegacyDocumentSupport 매개 변수를 no 로 설정하십시오.

예를 들어 URL 요청이 다음과 같으면

http://www.dominoserver.com/names.nsf/9348730948938987785784395880920398579834 9?EditDocument

Domino 에이전트에서 이 URL 을 다음과 같이 변환합니다.

http://www.dominoserver.com/names.nsf/Person?EditDocument

여기서 Person 은 원래 URL 에서 NotesID 로 식별된 문서를 생성할 때 사용한 폼의 이름입니다.

Notes 문서 액세스에 대해 4.6 이전 동작으로 Domino 웹 에이전트를 되돌려 ?ReadForm 작업만 허용되게 하려면 이 매개 변수를 yes 로 설정하십시오. 레거시 문서 지원이 설정된 경우에는 Domino 에이전트에서 이전 예제의 URL 을 다음과 같이 변환합니다.

http://www.dominoserver.com/names.nsf/Person?ReadForm

Notes 문서 이름 변환

뷰나 폼과 달리 Notes 문서는 이름이 없으며 문서를 생성할 때 사용된 폼에 대한 참조와 함께 데이터베이스에 저장됩니다. 사용자가 문서에 액세스하려는 경우 Domino 웹 에이전트에서 해당 문서를 읽을 수 있는 이름으로 변환할 수 없으면 문서를 생성한 폼의 이름을 사용하여 URL 을 생성합니다. 이 내용은 문서에만 적용됩니다. 원본 폼이 없으면 포함된 폼이 사용되고, 포함된 폼도 없으면 Domino 식별자 $defaultForm 을 사용하여 문서가 보호됩니다.

예를 들어 받는 URL 이 다음과 같으면

http://www.domino.com/names.nsf/8567489d60034we50938450098?OpenDocument

에이전트에서 다음 URL 을 사용합니다.

http://www.domino.com/names.nsf/Person?ReadForm

이 예제에서 Person 은 문서 이름입니다.

Domino 에이전트에 대해 전체 로그오프 지원 구성

전체 로그오프 기능은 다음 매개 변수로 생성하는 사용자 지정 로그아웃 페이지를 사용합니다.

LogOffUri

사용자 지정 웹 페이지의 URI 를 지정하여 전체 로그아웃 기능이 사용되도록 설정합니다. 이 사용자 지정 웹 페이지는 사용자가 성공적으로 로그오프된 후에 나타납니다. 이 페이지가 브라우저 캐시에 저장되지 않도록 구성하십시오. 그러지 않으면 브라우저에서는 사용자를 로그오프하지 않은 채 캐시의 로그아웃 페이지를 표시할 수 있습니다. 이러한 경우 권한 없는 사용자가 세션에 대한 제어권을 갖게 될 수 있습니다.

참고: CookiePath 매개 변수가 설정된 경우 LogOffUri 매개 변수의 값은 동일한 쿠키 경로를 가리켜야 합니다. 예를 들어 CookiePath 매개 변수의 값이 example.com 으로 설정되어 있으면 LogOffUri 가 example.com/logoff.html 을 가리켜야 합니다.

기본값: (CA SiteMinder Agent for SharePoint r12.0.3.0 을 *제외한* 모든 에이전트) 기본값 없음

제한: 여러 URI 값이 허용됨 정규화된 URL 을 사용하지 *마십시오. 상대* URI 를 사용하십시오.

예: (CA SiteMinder Agent for SharePoint r12.0.3.0 을 *제외한* 모든 에이전트) /Web pages/logoff.html

다음 단계를 수행하십시오.

1. 사용자를 로그오프하는 사용자 지정 HTTP 응용 프로그램을 생성합니다. 예를 들어 사용자를 지정된 URL 로 리디렉션하는 "Exit"(종료) 또는 "Sign Off"(로그오프) 단추를 추가합니다.
2. 웹 브라우저에서 캐시되지 않도록 로그아웃 페이지를 설정합니다. 이렇게 설정하면 페이지가 항상 브라우저의 캐시가 아니라 웹 서버에서 제공되므로 보안이 향상됩니다. 예를 들어 다음과 같은 메타 태그를 HTML 페이지에 추가할 수 있습니다.

   ```
   <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
   ```

   ```
   <META HTTP-EQUIV="Expires" CONTENT="-1">
   ```

   중요! 일부 웹 브라우저에서는 메타 태그를 지원하지 *않습니다.* cache-control HTTP 헤더를 대신 사용하십시오.
3. 다음 단계를 수행하여 LogOffUri 매개 변수를 구성합니다.

a. 필요한 경우 파운드 기호(#)를 삭제합니다.

b. 사용자를 로그오프할 사용자 지정 HTTP 파일의 URI 를 입력합니다. 정규화된 URL 을 사용하*면 안 됩니다*.

전체 로그아웃 기능이 구성되었습니다.

추가 정보

전체 로그오프 작동 방식 (페이지 267)

Domino 에이전트를 WebSphere 응용 프로그램 서버와 함께 사용

Domino 웹 서버는 요청을 WebSphere 서버에 전달하기 전에 가로채는 필터 플러그인을 제공하여 WebSphere 응용 프로그램 서버에 대한 프런트엔드 서버로 사용됩니다.

보호되지 않은 SiteMinder 리소스를 Domino 서버에서 인증하도록 지정

Domino 서버의 리소스를 SiteMinder 에서 보호하지 않으려는 경우를 가정해 봅니다. 대신 Domino 서버를 사용하여 해당 리소스를 보호할 수 있습니다. 이러한 리소스를 보호하려면 다음 매개 변수를 설정하십시오.

UseDominoUserForUnprotected

SiteMinder 가 *아니라* Domino *서버에서만* 보호하는 리소스에 대한 요청을 Domino 서버에서 Domino 사용자로 인증할지 여부를 지정합니다.

이 매개 변수의 값이 yes 인 경우 에이전트는 Domino 사용자를 Domino 서버에 전달합니다. 그러면 Domino 서버에서 사용자를 인증합니다. 이 매개 변수의 값이 no 이거나 매개 변수를 사용할 수 없는 경우에는 에이전트가 Domino 사용자를 Domino 서버에 전달하지 *않습니다*. 그러면 Domino 서버에서 사용자를 인증하지 않습니다.

기본값: 사용 안 함

다음 단계를 수행하십시오.

1. 위의 매개 변수를 찾습니다.
2. 매개 변수 앞에 있는 #(주석) 문자를 제거합니다.
3. 매개 변수의 값을 yes 로 변경합니다.

이전 버전과의 호환성 설정

다음 설정을 사용하면 SiteMinder 에이전트의 이전 버전과의 호환성을 관리할 수 있습니다.

- 레거시 URL 인코딩 조정 (페이지 374)
- POST 요청에서 콘텐츠 유형이 전송되는 방식 결정 (페이지 341)
- HOST 헤더를 전송하지 않는 테스트 도구 수용 (페이지 375)

레거시 URL 인코딩 조정

CA 에서 사용하는 레거시 URL 인코딩에는 달러 기호($)가 사용됩니다. 달러 기호가 문제가 될 경우에는 다음 매개 변수를 설정하여 웹 에이전트에서 달러 기호 대신 하이픈(-)이 사용되도록 할 수 있습니다.

LegacyEncoding

웹 에이전트가 레거시 URL 에서 달러 기호 문자($)를 하이픈(-)으로 바꾸도록 강제합니다. 또한 MSR, 암호 서비스 및 DMS 와의 호환성도 보장합니다. 이 매개 변수를 no 로 설정하면 웹 에이전트가 SM 문자열을 -SM-으로 변환합니다. 이 매개 변수를 yes 로 설정하면 웹 에이전트가 달러 기호 문자($)를 변환하지 *않습니다*.

기본값: (프레임워크 에이전트) No

기본값: (기존 에이전트) Yes

달러 기호 대신 하이픈을 사용하여 레거시 URL 을 인코딩하려면 LegacyEncoding 매개 변수의 값을 no 로 설정하십시오.

POST 요청에서 콘텐츠 유형이 전송되는 방식 선택

Apache 웹 서버를 사용하는 경우 다음 매개 변수를 설정하여 POST 요청 중에 콘텐츠가 서버에 전송되는 방식을 제어할 수 있습니다.

LegacyStreamingBehavior

POST 요청 중에 콘텐츠가 서버에 전송되는 방식을 지정합니다. 이 매개 변수의 값을 yes 로 설정하면 다음을 *제외한* 모든 콘텐츠 유형이 스트리밍됩니다.

- text/xml
- application/x-www-form-urlencoded

이 매개 변수의 값을 no 로 설정하면 모든 콘텐츠 유형이 스풀링됩니다.

기본값: No

POST 요청에서 대부분의 콘텐츠 유형을 스트리밍하려면 LegacyStreamingBehavior 매개 변수의 값을 yes 로 변경합니다.

HOST 헤더를 전송하지 않는 테스트 도구 수용

SiteMinder 웹 에이전트는 HTTP 요청의 HOST 헤더 값을 사용하여 다음 설정을 확인합니다.

- 에이전트 이름
- 서버 이름
- 서버 IP 주소

HTTP 버전 0.9 와 1.0 은 HOST 헤더를 사용하지 않기 때문에 SiteMinder 웹 에이전트는 HTTP 버전 1.1 요청만 수락합니다. 이로 인해 HOST 헤더를 보내지 않는 일부 테스트 도구에서 문제가 발생하는데, 웹 에이전트가 해당 요청을 거부하기 때문입니다.

SiteMinder 12.52 SP1 는 HOST 헤더 값을 정의하는 데 사용되는 새로운 에이전트 구성 매개 변수를 지원합니다. 웹 에이전트는 HOST 헤더를 포함하지 *않는* 요청에 이 값을 사용합니다.

HOST 헤더를 전송하지 않는 테스트 도구를 수용하려면

1. 다음 항목 중 하나를 엽니다.
   - 중앙 구성을 사용하는 경우 에이전트 구성 개체를 엽니다.
   - 로컬 구성을 사용하는 경우 LocalConfig.conf 파일을 엽니다.
2. 다음 매개 변수를 추가합니다.

   DefaultHostName

   HOST 헤더의 값을 정의합니다. 이 매개 변수를 에이전트 구성 개체 또는 LocalConfig.conf 파일에 추가하면 HOST 헤더 없이 HTTP 버전 0.9 또는 버전 1.0 요청을 보내는 테스트 또는 성능 도구를 사용할 수 있습니다. 이 매개 변수를 설정하지 않으면 웹 에이전트가 *HTTP 1.1 요청만* 수락합니다.

   기본값: 없음(비어 있음)

   예: webserver.example.com
3. 이 매개 변수의 값을 원하는 호스트 이름으로 설정합니다. 이전 예제를 참조하십시오.
4. 다음 항목 중 *하나*를 저장한 후 닫습니다.
   - 중앙 구성을 사용하는 경우 에이전트 구성 개체를 저장한 후 닫습니다.

- 로컬 구성을 사용하는 경우 LocalConfig.conf 파일을 저장한 후 닫습니다.

웹 에이전트는 HOST 헤더가 없는 HTTP 요청의 DefaultHostName 값을 대체합니다.

페더레이션 도메인에 대한 에이전트 설정

SiteMinder 가 레거시 페더레이션 SP 로 사용되는 경우 SAML 2.0 트랜잭션을 위한 IPD(아이덴티티 공급자 검색) 프로필을 구성할 수 있습니다. 사용자는 인증 요청에 대한 어설션을 생성하는 IdP 를 선택할 때 IPD 를 사용할 수 있습니다.

검색하는 동안 사용자가 악의적인 웹 사이트로 리디렉션되지 않게 할 수 있습니다. 이렇게 하려면 인증 요청을 충족하는 IdP 도메인의 유효성을 검사하도록 웹 에이전트를 구성해야 합니다.

유효성 검사 프로세스를 사용하려면 다음 매개 변수의 값을 설정합니다.

ValidFedTargetDomain

(페더레이션만 - SAML 2.0). IPD(아이덴티티 공급자 검색)를 구현할 때 페더레이션 환경에 대해 유효한 모든 도메인을 나열합니다.

SiteMinder IPD(아이덴티티 공급자 검색) 서비스가 요청을 받으면 해당 요청에서 IPDTarget 쿼리 매개 변수를 검사합니다. 이 쿼리 매개 변수는 검색 서비스가 요청을 처리한 후 리디렉션해야 하는 URL 을 나열합니다. IdP 의 경우 IPDTarget 은 SAML 2.0 싱글 사인온 서비스입니다. SP 의 경우 대상은 일반 도메인 쿠키를 사용하려고 요청하는 응용 프로그램입니다.

페더레이션 웹 서비스가 IPDTarget URL 의 도메인과 ValidFedTargetDomain 매개 변수에 지정된 도메인 목록을 비교합니다. URL 도메인이 ValidFedTargetDomain 에 구성된 도메인 중 하나와 일치하면 IPD 서비스가 사용자를 IPDTarget 매개 변수에 지정된 URL 로 리디렉션합니다. 이 경우 SP 의 URL 로 리디렉션됩니다.

일치하는 도메인이 없으면 IPD 서비스가 사용자 요청을 거부하고 브라우저를 통해 "403 사용 권한 없음" 오류가 수신됩니다. 또한 FWS 추적 로그와 affwebservices 로그에서 오류가 보고됩니다. 이러한 메시지는 IPDTarget 의 도메인이 유효한 페더레이션 대상 도메인으로 정의되지 않았음을 나타냅니다.

ValidFedTargetDomain 설정을 구성하지 않는 경우 유효성 검사가 수행되지 않고 사용자가 대상 URL 로 리디렉션됩니다.

제한: 페더레이션된 네트워크 내의 유효한 도메인

기본값: 기본값 없음

ValidFedTargetDomain 매개 변수에 유효한 도메인을 지정합니다. 이 설정은 다중값 매개 변수이므로 도메인을 여러 개 입력할 수 있습니다.

로컬 구성 파일을 수정하는 경우에는 다음과 같이 각 도메인을 하나씩 추가해야 합니다.

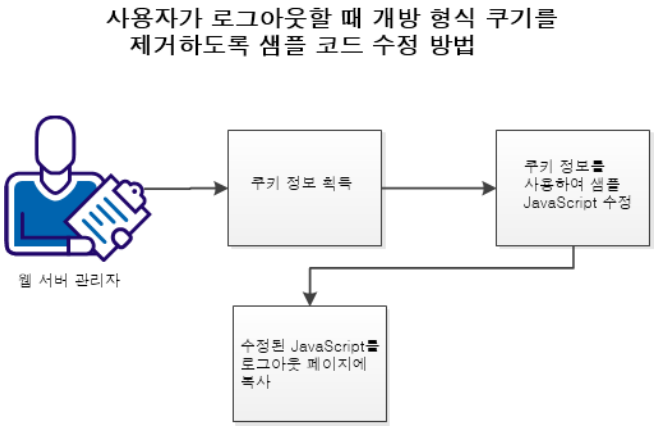
validfedtargetdomain=".examplesite.com"

validfedtargetdomain=".abccompany.com"

아이덴티티 공급자 검색 프로필에 대한 자세한 내용은 *Federation Security Services Guide*(Federation Security Services 안내서)를 참조하십시오.

사용자가 로그아웃할 때 개방 형식 쿠키를 제거하도록 샘플 코드를 수정하는 방법

SiteMinder 에서는 개방 형식 쿠키를 인식하거나 처리하거나 삭제할 수 *없습니다*. 사용자가 로그아웃할 때 개방 형식 쿠키를 제거하는 클라이언트 측 스크립트를 직접 생성하십시오.



다음 단계를 수행하십시오.

1. 쿠키 정보를 얻습니다 (페이지 378).
2. 쿠키 정보를 사용하여 샘플 JavaScript 코드를 수정합니다 (페이지 379).
3. 수정한 JavaScript 코드를 로그아웃 페이지에 복사합니다 (페이지 381).

쿠키 정보 가져오기

클라이언트 측 로그아웃 스크립트를 생성하려면 개방 형식 쿠키에 대한 다음 정보가 필요합니다.

- 개방 형식 쿠키의 이름(예: SMOFC)
- 쿠키 경로(예: /[슬래시는 개방 형식 쿠키의 루트 디렉터리를 나타냄])
- 쿠키가 생성된 도메인(예: .example.com)

이 정보는 에이전트 소유자 또는 웹 서버 관리자로부터 얻으십시오.

쿠키 정보를 사용하여 샘플 JavaScript 코드 수정

개방 형식 쿠키에 대한 정보를 얻은 후에는 샘플 JavaScript 코드를 수정하십시오.

다음 단계를 수행하십시오.

1. 샘플 JavaScript 코드를 텍스트 편집기에 복사합니다.

```
<html>
    <head>
        <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
        <META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">
        <META HTTP-EQUIV="Expires" CONTENT="-1">

        <!-- JavaScript to remove cookie from browser -->
        <script>
            // This function takes the cookie name, path and domain
            // and constructs a expired cookie so that the browser removes the
cookie from its store
            function eraseCookie(name, path, domain)
            {
                if (name)
                {
                  var delCookie = name + '=; expires=Thu, 01-Jan-70 00:00:01
GMT';
                  if (path && path.length > 0) delCookie += ';path=' + path;
                  if (domain && domain.length > 0) delCookie += ';domain=' +
domain;
                  document.cookie = delCookie;
                }
            }

            function showCookie(name)
            {
                var ckVal = null;
                var tC = document.cookie.split('; ');
                for (var i = tC.length - 1; i >= 0; i--)
                {
                    var x = tC[i].split('=');

                    if (name == x[0] && x[1])
                    {
                        ckVal = unescape(x[1]);
                        break;
                    }
                }

                if (ckVal)
```

```
                alert( name + ' = ' + ckVal);
            else
                alert('Cookie ' + name + ' does not exist');
        }
    </script>
</head>

<body>
    <p><a href="javascript:showCookie('SMOFC')" class="page">개방 형식 쿠키를 표시하려면 클릭하십시오.</a><br />
    <p><a href="javascript:eraseCookie('SMOFC', '/', 'example.com')" class="page">개방 형식 쿠키를 제거하려면 클릭하십시오.</a><br />
</body>
</html>
```

2. 다음 기본값을 모두 개방 형식 쿠키의 값으로 대체합니다.

   이름

   개방 형식 쿠키 이름으로 대체하십시오.

   예: SMOFC

   경로

   개방 형식 쿠키 경로로 대체하십시오.

   예: \

   도메인

   개방 형식 쿠키 도메인으로 대체하십시오.

   예: example.com

3. 샘플 JavaScript 의 다른 요소를 사용 환경에 필요한 대로 변경합니다.
4. 수정한 JavaScript 를 저장하고 텍스트 편집기를 닫습니다.

수정한 JavaScript 코드를 로그아웃 페이지에 복사

로그아웃 페이지를 수정한 JavaScript 코드로 업데이트하십시오. 이 코드는 사용자가 로그아웃할 때 개방 형식 쿠키를 제거합니다.

다음 단계를 수행하십시오.

1. 텍스트 편집기를 사용하여 웹 서버의 로그아웃 페이지를 엽니다.
2. 수정한 JavaScript 코드를 로그아웃 페이지에 복사합니다.
3. 변경 내용을 페이지에 저장하고 텍스트 편집기를 닫습니다.
4. 각 웹 서버에 대해 1-3 단계를 반복합니다.

제 19 장: 성능

이 섹션은 다음 항목을 포함하고 있습니다.

저장된 자격 증명의 만료 시간 설정

사용자가 자격 증명을 저장하도록 선택하는 경우 정책 서버는 해당 사용자의 자격 증명을 포함하는 영구 쿠키를 생성하도록 웹 에이전트에 지시합니다. 웹 에이전트는 인증할 사용자에게 자격 증명을 요청하지 않고 이 쿠키에 저장된 자격 증명을 기반으로 사용자를 인증할 수 있습니다. 다음 매개 변수를 사용하여 영구 쿠키가 유지되는 기간을 제어할 수 있습니다.

SaveCredsTimeout

사용자 자격 증명이 포함된 영구 쿠키가 저장되는 시간 간격을 지정합니다. 이 시간 간격 중에 웹 에이전트는 사용자를 쿠키에 저장된 데이터를 사용하여 인증합니다. 이 시간 간격이 만료되면 쿠키가 제거되고 웹 에이전트는 사용자에게 다시 인증을 요청합니다.

기본값: 720(30 일)

저장된 자격 증명의 만료 시간을 설정하려면 SaveCredsTimeout 매개 변수에 원하는 시간을 입력하십시오.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

웹 에이전트 캐시

웹 에이전트는 사용자 세션 및 리소스 정보를 캐시 메모리에 저장합니다. 이 방법은 사용자가 액세스를 요청할 때마다 정책 서버에서 정보를 검색할 필요가 없으므로 웹 에이전트의 효율성을 높일 수 있습니다.

캐시 설정을 구성하면 이러한 정보가 저장되는 방식을 관리할 수 있습니다. 캐시의 항목 수에 따라 캐시 크기가 결정됩니다. 각 캐시의 전체 항목 수는 지정된 최대 캐시 크기를 초과할 수 없습니다.

참고: 웹 에이전트 캐시 설정의 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다.

다음과 같은 지침이 캐시 관리에 적용됩니다.

- 캐시가 꽉 차면 가장 오래 전에 사용한 항목을 새 항목으로 바꿉니다.
- 리소스 캐시의 경우 ResourceCacheTimeout 매개 변수의 값을 초과하면 항목이 제거됩니다.
- 사용자 세션 캐시의 경우 각 영역에 설정한 세션 시간 제한 값에 따라 항목이 제거됩니다.

SiteMinder 는 정책이 수정되면 캐시된 리소스 정보를 비웁니다. 또한 관리 UI 를 사용하여 사용자 및 리소스 캐시를 수동으로 비울 수도 있습니다.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

다음 매개 변수를 사용하면 에이전트의 캐시를 관리할 수 있습니다.

- 익명 사용자 캐시 (페이지 385)
- 리소스 캐시의 최대 크기 설정 (페이지 386)
- 사용자 세션 캐시의 최대 크기 설정 (페이지 387)
- 리소스 항목이 캐시에 유지되는 시간 제어 (페이지 388)
- 리소스 캐시 사용 안 함 (페이지 388)

익명 사용자 캐시

다음 매개 변수를 사용하면 익명 사용자 정보를 캐시에 저장하도록 웹 에이전트를 구성할 수 있습니다.

CacheAnonymous

웹 에이전트가 익명 사용자 정보를 캐시할지 여부를 지정합니다. 다음과 같은 경우 이 매개 변수를 설정할 수 있습니다.

- 웹 사이트의 방문자가 대부분 익명 사용자이고 이러한 사용자의 세션 정보를 저장하려는 경우
- 웹 사이트의 방문자가 등록된 사용자와 익명 사용자가 섞여 있는 경우

  익명 사용자 정보가 캐시를 채워 등록된 사용자에 대한 공간이 부족해지지 않도록 이 매개 변수를 해제할 수도 있습니다.

기본값: No

익명 사용자 정보를 캐시에 저장하려면 CacheAnonymous 매개 변수의 값을 yes 로 설정하십시오.

리소스 캐시의 최대 크기 설정

다음 매개 변수를 사용하면 웹 페이지와 같이 웹 에이전트가 추적하는 리소스 캐시 항목의 최대 개수를 설정할 수 있습니다.

MaxResourceCacheSize

웹 에이전트가 리소스 캐시에 유지하는 최대 항목 수를 지정합니다. 항목에는 다음 정보가 포함됩니다.

- 리소스가 보호되는지 여부에 대한 정책 서버 응답
- 응답과 함께 반환된 추가 특성

최대 수에 도달하면 새 리소스 레코드가 가장 오래된 리소스 레코드를 대체합니다.

이 값을 높은 수로 설정하는 경우 충분한 시스템 메모리를 사용할 수 있어야 합니다.

OneView 모니터를 사용하여 웹 에이전트 통계를 보면 ResourceCacheCount 에 대해 표시되는 값이 MaxResourceCacheSize 매개 변수에 지정한 값보다 크다는 것을 알 수 있습니다. 이는 오류가 아닙니다. 웹 에이전트는 MaxResourceCacheSize 매개 변수를 기준으로 사용하며 MaxResourceCacheSize 매개 변수는 리소스 캐시에서 평균 크기 항목의 최대 수를 나타내기 때문에 때로는 값이 다를 수 있습니다. 실제 캐시 항목은 미리 결정된 평균 크기보다 크거나 작을 수 있으므로 실제 최대 항목 수는 지정된 값보다 크거나 작을 수 있습니다.

참고: 프레임워크 에이전트처럼 공유 메모리를 사용하는 웹 에이전트의 경우 캐시가 MaxResourceCacheSize 값에 따라 상수 크기로 사전 할당되므로 커지지 않습니다.

기본값: (Domino 웹 서버) 1000

기본값: (IIS 및 Sun Java System 웹 서버) 700

기본값: (Apache 웹 서버) 750

리소스 캐시의 최대 크기를 설정하려면

1. MaxResourceCacheSize 매개 변수의 값을 원하는 최대 리소스 수로 설정합니다.
2. 프레임워크 에이전트의 경우 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다.

   리소스 캐시의 최대 크기가 변경됩니다.

사용자 세션 캐시의 최대 크기 설정

다음 매개 변수를 사용하면 에이전트가 세션 캐시에 유지하는 최대 사용자 수를 설정할 수 있습니다.

MaxSessionCacheSize

에이전트가 세션 캐시에 유지하는 최대 사용자 수를 지정합니다. 세션 캐시에는 성공적으로 인증한 사용자의 세션 ID 가 저장됩니다. 세션 중에 해당 영역 내의 다른 리소스에 액세스하는 인증된 사용자는 정책 서버 대신 세션 캐시를 사용하여 인증됩니다. 최대 수에 도달하면 에이전트는 가장 오래된 사용자 레코드를 새로운 사용자 레코드로 대체합니다.

지속된 기간 동안 리소스에 액세스하여 리소스를 사용할 것으로 예상되는 사용자의 수를 기반으로 이 매개 변수의 값을 지정하십시오. 이 값을 높은 수로 설정하는 경우 충분한 시스템 메모리를 사용할 수 있어야 합니다.

참고: 캐시 크기에 관계없이 웹 에이전트의 세션 캐시에 저장된 모든 항목은 1 시간 후 *자동으로 만료됩니다*.

기본값: (Domino 웹 서버) 1000

기본값: (IIS 및 Oracle iPlanet 웹 서버) 700

기본값: (Apache 웹 서버) 750

사용자 세션 캐시의 최대 크기를 설정하려면

1. MaxSessionCacheSize 매개 변수의 값을 원하는 최대 사용자 수로 설정합니다.
2. 프레임워크 에이전트의 경우 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다.

   사용자 세션 캐시의 최대 크기가 변경됩니다.

리소스 항목이 캐시에 유지되는 시간 제어

다음 매개 변수를 사용하면 리소스 항목이 캐시에 유지되는 시간을 변경할 수 있습니다.

ResourceCacheTimeout

리소스 항목이 캐시에서 유지되는 시간(초)을 지정합니다. 이 시간 간격이 초과된 후 사용자가 보호된 리소스에 액세스하려고 하면 웹 에이전트는 캐시된 항목을 제거하고 정책 서버에 연결합니다.

기본값: 600(10 분)

참고: 이 매개 변수의 값을 변경하면 웹 서버를 다시 시작해야 변경 내용이 적용됩니다.

리소스 항목이 캐시에 유지되는 시간을 변경하려면 ResourceCacheTimeout 매개 변수를 원하는 시간(초)으로 설정하십시오.

리소스 캐시 사용 안 함

고유한 동적 URL 을 사용하는 응용 프로그램을 보호하는 경우 리소스 캐시가 사용되지 않도록 설정할 수 있습니다. 해당 응용 프로그램에 사용되는 URL 은 고유하므로 캐시에서 읽을 수 없습니다.

리소스 캐시가 사용되지 않도록 설정하려면 MaxResourceCacheSize 값을 0 으로 변경하십시오.

웹 에이전트 모니터링

다음 방법을 사용하면 에이전트 성능을 모니터링할 수 있습니다.

- OneView 모니터를 사용하여 에이전트 모니터링 (페이지 389)
- CA Wily Introscope 를 사용하여 에이전트 모니터링 (페이지 390)

추가 정보:

웹 에이전트와 정책 서버의 통신을 관리하는 방법 (페이지 61)

OneView 모니터를 사용하여 웹 에이전트 모니터링

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 SiteMinder OneView 모니터를 제어합니다.

EnableMonitoring

SiteMinder 웹 에이전트가 정책 서버에 모니터링 정보를 보낼지 여부를 지정합니다.

기본값: No

웹 에이전트에서 SiteMinder OneView 모니터를 사용하게 하려면 EnableMonitoring 매개 변수를 yes 로 설정하십시오.

참고: 자세한 내용은 정책 서버 설명서를 참조하십시오.

CA Wily Introscope 를 사용하여 웹 에이전트 모니터링

CA Wily Introscope 를 이미 조직에 사용하는 경우 다음 매개 변수를 설정하여 SiteMinder 웹 에이전트의 건전성을 모니터링할 수 있습니다.

EnableIntroscopeApiSupport

SiteMinder 웹 에이전트에 대한 정보를 수집하고 플러그인을 사용하여 CA Wily Introscope 에 보냅니다. 이 매개 변수에는 다음 설정이 사용됩니다.

- yes 로 설정하면 Wily 플러그인이 API 를 호출하여 데이터를 수집합니다.
- no 로 설정하면 Wily 플러그인이 데이터와 함께 HTTP 헤더를 생성합니다.
- both 로 설정하면 Wily 플러그인이 API 를 호출하고 데이터와 함께 HTTP 헤더를 생성합니다.
- none 으로 설정하면 데이터가 수집되지 않습니다.

기본값: No

제한: Yes, Both, No, None

예: (HTTP 헤더) sm-wa-perf-counters = *server_name*.example.com:6180,86117203,86118343,1,0,0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,1125,0,15,1,1,750,750,

CA Wily Introscope 를 사용하여 웹 에이전트의 건전성을 모니터링하려면 EnableIntroscopeApiSupport 매개 변수 값을 다음 중 *하나*로 설정하십시오.

- Yes
- Both
- No

보호되지 않은 리소스 무시

보호하지 *않을* 리소스에 대한 요청을 무시하면 SiteMinder 의 성능을 향상시킬 수 있습니다. 다음과 같은 매개 변수를 사용할 수 있습니다.

- 특정 파일 확장명을 무시하여 오버헤드 감소 (페이지 392)
- 에이전트에서 무시할 가상 서버 지정 (페이지 173)
- URL 의 쿼리 데이터 무시 (페이지 395)
- URI 무제한 액세스 허용 (페이지 397)

보호되지 않은 리소스의 파일 확장명을 무시하여 오버헤드 감소

다음 매개 변수를 사용하여 웹 에이전트에서 특정 유형의 리소스 요청을 무시하도록 지정하면 SiteMinder 오버헤드를 줄일 수 있습니다.

IgnoreExt

웹 에이전트가 SiteMinder 정책을 확인하지 않고 요청을 웹 서버에 전달하는 리소스 유형을 지정합니다. 웹 에이전트는 이 매개 변수에서 지정하는 항목이 SiteMinder 정책으로 보호되는 영역에 있을 경우에도 해당 항목에 대한 액세스를 허용합니다.

다음 조건 중 하나에 해당되는 리소스에 대한 요청은 무시할 수 있습니다.

- 리소스가 웹 에이전트에서 무시하도록 구성한 확장명 중 하나로 끝납니다.
- 보호된 리소스의 URI 에 마침표(.)가 하나 포함되어 있습니다.

  예를 들어 요청되는 리소스의 URI 가 /my.dir/이면 웹 에이전트는 요청을 웹 서버에 직접 전달합니다.

기본값: .class, .gif, .jpg, .jpeg, .png, .fcc, .scc, .sfcc, .ccc, .ntc

중요! IgnoreExt 매개 변수는 신중하게 설정하십시오. 고려해야 할 몇 가지 보안 문제가 있기 때문입니다.

기본적으로 에이전트는 슬래시(/)로 구분된 점이 두 개 이상 포함된 리소스에 대한 요청을 무시하지 *않습니다*. 웹 에이전트는 다음 예제에서 설명하는 단계를 사용하여 리소스 요청을 처리합니다.

1. .gif 확장명이 IgnoreExt 매개 변수에 추가됩니다. 확장명이 .gif 인 리소스에 대한 요청이 웹 에이전트에서 무시됩니다.
2. 다음 URI 에 대한 요청이 발생합니다.

   /dir1/app.pl/file1.gif
3. 일부 웹 서버의 경우 file1.gif 리소스를 제공하는 대신 /dir1/app.pl 을 응용 프로그램으로 실행하므로 웹 에이전트는 정책 서버에 대해 /dir1/app.pl/file1.gif 를 확인합니다.

   웹 서버를 참조하지 않고 /dir1/app.pl/file1.gif 에 대한 액세스 권한을 부여하면 보안 위반 문제가 발생할 수 있습니다.

보호되지 않은 리소스의 파일 확장명을 무시하여 오버헤드를 줄이려면 무시할 리소스 확장명을 IgnoreExt 매개 변수의 값에 추가합니다.

웹 에이전트에서 무시할 가상 서버 지정

사이트의 웹 서버가 가상 서버를 여러 개 지원하는 경우 이러한 가상 서버의 리소스 중 일부를 웹 에이전트에서 보호하지 않도록 설정할 수 있습니다. 웹 에이전트에서 보호할 웹 서버 콘텐츠를 간단한 방법으로 구별할 수 있게 하려면 다음 매개 변수를 사용합니다.

IgnoreHost

웹 에이전트에서 무시하도록 하려는 가상 서버의 정규화된 도메인 이름을 지정합니다. 이러한 가상 서버의 리소스는 자동으로 권한이 부여되고 요청한 클라이언트에 관계없이 웹 에이전트는 이러한 리소스에 대한 액세스 권한을 항상 부여받습니다. 권한 부여 결정이 정책이 아닌 웹 에이전트의 구성을 기반으로 이루어집니다.

IgnoreExt 및 IgnoreURL 등의 다른 자동 권한 부여 설정을 확인하기 전에 무시되는 호스트의 목록을 먼저 확인합니다. 따라서 이중 점 규칙은 무시되는 호스트의 리소스에 대한 권한 부여 요청을 정책 서버에 트리거하지 않으며 확장명에 의해 무시되지 않습니다.

IgnoreHost 매개 변수에서 URL 항목의 호스트 부분은 웹 에이전트가 요청된 리소스의 호스트 헤더에서 읽는 내용과 정확히 일치해야 합니다.

참고: 이 값은 대/소문자를 구분합니다.

URL 에서 특정 포트를 사용하면 이 포트를 지정해야 합니다.

중앙에서 관리되는 에이전트의 경우 에이전트 구성 개체에서 다중값 매개 변수를 사용하여 여러 서버를 나타낼 수 있습니다. 로컬 구성 파일로 구성된 에이전트의 경우 각 호스트를 파일의 각 줄에 나열하십시오.

예: (포트가 지정되어 표시되는 URL)

IgnoreHost="myserver.example.org:8080"

예: (로컬 구성 파일)

IgnoreHost="my.host.com"

IgnoreHost="your.host.com"

기본값: 기본값 없음

웹 에이전트에서 무시할 가상 서버를 지정하려면 다음 태스크를 수행하십시오.

- 중앙 구성의 경우 무시할 서버를 에이전트 구성 개체에 추가합니다. 서버가 여러 개이면 매개 변수에 다중값 설정을 사용해야 합니다.

- 로컬 구성의 경우 각 서버를 로컬 구성 파일에서 한 행에 하나씩 추가합니다.

지정된 URL 을 사용하는 리소스는 웹 에이전트에서 무시되고 이러한 리소스에는 자동으로 액세스됩니다.

추가 정보

복잡한 URI 처리 (페이지 112)

URL 의 쿼리 데이터 무시

IgnoreQueryData 매개 변수는 웹 에이전트에서 URL 을 처리하는 방식에 영향을 줍니다. 웹 에이전트에서 규칙을 처리할 때 전체 URL 을 캐시하거나 쿼리 문자열이 포함된 URI 를 정책 서버에 전달하지 않으려는 경우 다음 매개 변수를 사용하면 성능이 향상됩니다.

IgnoreQueryData

웹 에이전트가 전체 URL(쿼리 문자열 포함)을 캐시하고 규칙 처리를 위해 정책 서버에 보낼지 여부를 지정합니다. 전체 URL 문자열에는 다음 예와 같이 URI, 후크(?) 및 몇 가지 쿼리 데이터가 포함됩니다.

URI?*query_data*

요청의 주체였던 URL 은 기본적으로 캐시됩니다. 이후의 요청에서는 캐시를 검색하여 일치 항목을 찾습니다. 동일한 URI 에 대한 요청에서 쿼리 데이터가 다르면 일치 항목이 검색되지 않습니다. 쿼리 데이터를 무시하면 성능이 향상됩니다.

IgnoreQueryData 매개 변수를 yes 로 설정하면 다음과 같은 동작이 수행됩니다.

- URL 이 후크에서 잘립니다. URI 만 캐시되고 정책 서버로 전송됩니다. 리디렉션의 적절한 상태를 유지하기 위해 쿼리 데이터가 다른 위치에 유지됩니다.
- 후크 앞의 부분만 규칙 처리를 위해 정책 서버로 전송됩니다.
- 다음 예에서 두 URI 는 동일한 리소스로 처리됩니다.

  /myapp?data=1

  /myapp?data=2

IgnoreQueryData 매개 변수를 no 로 설정하면 다음과 같은 동작이 수행됩니다.

- 전체 URL 이 캐시됩니다.
- 전체 URI 가 규칙 처리를 위해 정책 서버로 전송됩니다.
- 다음 예에서 두 URI 는 다른 리소스로 처리됩니다.

  /myapp?data=1

  /myapp?data=2

기본값: No

처리할 URI 만 정책 서버에 전달하도록 웹 에이전트를 지정하려면 IgnoreQueryData 매개 변수의 값을 yes 로 설정하십시오.

중요! URL 쿼리 데이터를 사용하는 정책이 있을 경우 이 설정을 활성화하지 마십시오.

URI 무제한 액세스 허용

특정 URI 를 SiteMinder 로 보호하지 않으려는 경우 다음 매개 변수를 설정하여 웹 에이전트에서 해당 URI 를 무시하고 무제한 액세스를 허용하도록 지정할 수 있습니다.

IgnoreUrl

URL 내에서 보호되지 않을 URI 를 지정합니다. 이 URI 와 관련된 리소스에 액세스하려고 하는 사용자는 인증 요청을 받지 않습니다. 웹 에이전트는 문자열에서 세 개의 슬래시 뒤의 URI 부분을 무시합니다. 예를 들어 이 매개 변수를 다음 값으로 설정합니다.

```
http://www.example.com/directory
```

웹 에이전트는 다음 URI 를 무시합니다.

```
directory
```

웹 에이전트는 지정된 URI 가 다른 도메인에 속해 있어도 해당 URI 를 항상 무시합니다. 예를 들어 웹 에이전트는 다음 모든 URL 에서 이전에 표시된 URI 를 무시합니다.

```
http://www.example.com/directory
http://www.example.net/directory
http://www.example.org/directory
```

참고: 이 값은 대/소문자를 구분합니다.

기본값: 기본값 없음

예: (로컬 구성 파일의 여러 URI 사용)

IgnoreUrl="http://www.example.com/directory"

IgnoreUrl="http://www.example.com/directory2"

예: (도메인을 지정하지 않고 하나의 URI 만 사용)

IgnoreUrl="/*resource*/"

URI 에 대한 무제한 액세스를 허용하려면 다음 태스크를 수행하십시오.

- 중앙 구성의 경우 무시할 URI 가 포함된 정규화된 도메인 이름을 에이전트 구성 개체에 추가합니다. URI 가 여러 개이면 매개 변수에 다중값 설정을 사용해야 합니다.
- 로컬 구성의 경우 정규화된 도메인 이름 및 URI 를 로컬 구성 파일에서 한 행에 하나씩 추가합니다.

지정된 URI 를 사용하는 리소스는 웹 에이전트에서 무시되고 이러한 리소스에는 자동으로 액세스됩니다.

제 20 장: 로깅 및 추적

이 섹션은 다음 항목을 포함하고 있습니다.

시작 이벤트 로그

디버깅에 도움이 되도록 하기 위해 시작 이벤트가 로그에 기록됩니다. 각 메시지는 문제에 대한 정보를 제공합니다. 이러한 로그는 다음 위치에 저장됩니다.

- Windows 시스템의 경우 시작 이벤트가 Windows 응용 프로그램 이벤트 로그에 기록됩니다.
- UNIX 시스템의 경우 시작 이벤트가 STDERR 에 전달됩니다. Apache 서버는 STDERR 를 Apache error_log 파일에 매핑하므로 이러한 이벤트도 해당 로그에 기록됩니다.

오류 로그 및 추적 로그

웹 에이전트 로깅 기능을 사용하여 웹 에이전트 및 정책 서버와의 통신 성능을 모니터링할 수 있습니다. 로깅 기능은 성능을 분석하고 문제를 해결하기 위한 SiteMinder 프로세스 작업에 대한 정확하고 포괄적인 정보를 제공합니다.

로그는 프로그램 실행 중에 발생하는 이벤트의 레코드입니다. 로그는 각각 프로그램 실행 중에 발생한 몇 가지 이벤트를 설명하는 일련의 로그 메시지로 구성됩니다. 로그 메시지는 로그 파일에 기록됩니다.

참고: IIS 에이전트는 첫 번째 사용자 요청이 제출된 후에만 로그 파일을 생성합니다. 또한 Apache 2.0 웹 에이전트는 Apache 서버가 시작될 때 로그 파일을 생성합니다.

웹 에이전트에서는 다음과 같은 로그 파일을 사용합니다.

오류 로그

프로그램 및 작업 수준 오류를 포함합니다. 웹 에이전트가 정책 서버와 통신할 수 없는 경우를 예로 들 수 있습니다. 이 로그의 경우 세부 출력 수준을 사용자 지정할 수 없습니다. 오류 로그에는 다음과 같은 유형의 메시지가 포함됩니다.

오류 메시지

네트워크 오류와 같은 외부 문제로 인해 기능이 예상대로 작동하지 않는 경우 또는 잘못되었거나 비정상적인 프로그램 동작을 나타내는 프로그램 수준 오류를 포함합니다. 작업 수준 오류도 포함됩니다. 이 유형의 오류는 파일 열기 또는 사용자 인증과 같은 작업이 실패하는 경우입니다.

정보 메시지

서버 시작 또는 중지와 같은 일부 이벤트가 실행되었거나 작업이 수행되었음을 사용자 또는 관리자에게 알려 주는 메시지를 포함합니다.

경고 메시지

비정상적이거나 잠재적 문제를 나타내는 이벤트 및 상태가 있음을 사용자 또는 관리자에게 알려 주는 경고를 포함합니다. 경고 메시지는 문제가 발생했음을 나타내는 것은 아닙니다.

추적 로그

직접 구성할 수 있는 자세한 경고 및 정보 메시지를 포함합니다. 예를 들면 추적 메시지와 흐름 상태 메시지입니다. 이 파일에는 헤더 정보 및 쿠키 변수와 같은 데이터도 포함됩니다. 추적 로그에는 다음과 같은 메시지가 포함됩니다.

추적 메시지

추적 및 디버깅에 사용할 목적으로 프로그램 작업과 관련된 자세한 정보를 제공합니다. 일반적으로 정상 작업 중에는 추적 메시지가 해제됩니다. 정보, 경고 및 오류 메시지와 달리 추적 메시지는 원본 코드에 포함되며 쉽게 지역화할 수 없습니다. 또한 추적 메시지는 메시지 자체뿐만 아니라 현재 사용자 또는 영역 이름과 같은 중요한 데이터를 포함할 수 있습니다.

웹 에이전트를 구성할 때 오류 로그 파일과 추적 로그 파일의 위치를 둘 다 지정합니다. 오류 및 추적 로그는 웹 에이전트가 올바로 작동하지 않는 경우 문제를 해결하는 데 유용합니다.

참고: Windows 플랫폼에 설치된 에이전트의 경우 웹 에이전트 로그를 생성하려면 EnableWebAgent 매개 변수를 yes 로 설정해야 합니다. EnableWebAgent 가 기본값인 no 로 설정된 상태에서 로깅 매개 변수를 설정하면 UNIX 플랫폼의 에이전트에 대해서만 에이전트 로그가 생성됩니다.

추가 정보

오류 로깅 설정 및 사용 (페이지 403)
추적 로깅 구성 (페이지 408)

로그 파일에 표시되는 매개 변수 값

웹 에이전트는 구성 매개 변수 및 매개 변수 값을 웹 에이전트 오류 로그 파일에 나열하지만 기존 에이전트와 프레임워크 에이전트의 처리 방식에 차이가 있습니다.

프레임워크 에이전트는 구성 매개 변수 및 매개 변수 값을 에이전트 구성 개체 또는 로컬 구성 파일에 입력한 그대로 로그 파일에 기록합니다. 즉, 잘못된 값이 지정된 매개 변수를 포함하여 모든 매개 변수가 로그 파일에 기록됩니다.

기존 에이전트는 매개 변수 값을 먼저 처리한 후 기록합니다. 즉, 매개 변수의 값이 올바르면 해당 매개 변수와 값이 로그 파일에 기록됩니다. 잘못된 값이 지정된 매개 변수는 로그 파일에 기록되지 *않습니다.*

오류 로깅 설정 및 사용

오류 로그에는 다음과 같은 설정이 필요합니다.

- 로깅을 사용할 수 있어야 합니다.
- 로그 파일 위치를 지정해야 합니다.

오류 로깅을 설정하고 옵션(예: 로그 데이터 추가)을 결정하는 매개 변수는 정책 서버의 에이전트 구성 개체 또는 로컬 구성 파일에 정의됩니다.

IIS 또는 Apache 웹 서버에 설치된 에이전트에서는 로컬 구성 파일에 로컬로 설정된 로그 매개 변수를 동적으로 구성할 수 없습니다. 에이전트를 다시 시작하면 변경 내용이 적용됩니다. 그러나 이러한 로그 설정을 정책 서버의 에이전트 구성 개체에서 동적으로 저장하고 업데이트할 수 있습니다.

참고: IIS 에이전트는 첫 번째 사용자 요청이 제출된 후에만 로그 파일을 생성합니다. 또한 Apache 2.0 웹 에이전트는 Apache 서버가 시작될 때 로그 파일을 생성합니다.

다음 단계를 수행하십시오.

1. 로그 파일이 없는 경우 로그 파일 및 관련 디렉터리를 생성합니다.
2. LogFile 매개 변수의 값을 yes 로 설정합니다.

   참고: 웹 서버의 로컬 구성 파일에서 이 매개 변수의 값을 yes 로 설정하면 정책 서버에 정의된 로깅 설정이 무시됩니다. 예를 들어 LocalConfig.conf 파일에서 이 매개 변수의 값이 yes 로 설정되어 있다고 가정해 봅니다. 이 경우에는 해당 에이전트 구성 개체에서 AllowLocalConfig 매개 변수의 값이 no 로 설정되어 있더라도 에이전트에서 로그 파일을 생성합니다. 또한 LocalConfig.conf 파일에서 관련 로깅 매개 변수를 설정하여 에이전트 구성 개체의 다른 설정을 무시할 수도 있습니다.
3. 다음 매개 변수를 사용하여 오류 파일의 이름을 포함한 전체 경로를 지정합니다.

   LogFileName

   로그 파일의 이름을 포함한 전체 경로를 지정합니다.

   기본값: No

   예: (Windows) *web_agent_home*\log\WebAgent.log

   예: (UNIX/LInux) /export/iPlanet/servers/https-jsmith/logs/WebAgent.log

LogFileName32

32 비트 응용 프로그램을 보호하는 64 비트 Windows 운영 환경에서 IIS 용 SiteMinder 웹 에이전트에 대한 로그 파일의 전체 경로를 지정합니다. 32 비트 응용 프로그램은 64 비트 Windows 운영 환경의 Wow64 모드에서 실행됩니다. 로깅하도록 설정된 경우 이 매개 변수가 설정되지 않으면 IIS 용 웹 에이전트에서 로그 파일 이름에 _32 를 추가합니다.

기본값: No

제한: Windows 64 비트 운영 환경에만 적용됩니다. 경로 끝에 파일 이름을 지정하십시오.

예: (Wow64 모드를 사용하는 Windows 64 비트 운영 환경) *web_agent_home*\log\WebAgent32.log

4. (선택 사항) 정책 서버의 에이전트 구성 개체 또는 로컬 구성 파일에서 다음 매개 변수를 설정합니다.

LogAppend

새 로그 정보를 기존 로그 파일의 끝에 추가합니다. 이 매개 변수가 no 로 설정되면 로깅을 호출할 때마다 전체 로그 파일이 다시 작성됩니다.

기본값: No

LogFileSize

로그 파일 크기 제한(MB)을 지정합니다. 현재 로그 파일의 크기가 이 값이 되면 새 로그 파일이 생성됩니다. 새 로그 파일의 이름은 다음과 같은 명명 규칙을 사용하여 지정됩니다.

- 프레임워크 에이전트의 경우 원래 이름에 시퀀스 번호를 추가하여 새 로그 파일 이름을 지정합니다. 예를 들어 myfile.log 라는 로그 파일은 크기 제한에 도달하면 myfile.log.1 로 이름이 바뀝니다.
- 기존 에이전트의 경우 원래 이름에 날짜 및 타임스탬프를 추가하여 새 로그 파일 이름을 지정합니다. 예를 들어 myfile.log 라는 로그 파일은 크기 제한에 도달하면 myfile.log.09-18-2003-16-07-07 로 이름이 바뀝니다.

이전 파일은 수동으로 보관하거나 제거하십시오.

기본값: 0(롤오버 안 함)

예: 80

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.

기본값: Yes

로컬 구성 파일을 사용하는 경우 다음과 같이 설정됩니다.

```
LogFile="yes"
LogFileName="/export/iPlanet/servers/https-myserver/logs/errors.log"
LogAppend="no"
LogFileSize="80"
LogLocalTime="yes"
```

오류 로깅을 사용할 수 있습니다.

TLI 로깅 사용

에이전트와 정책 서버 간의 연결을 검사하려는 경우 전송 계층 인터페이스 로깅이 사용되도록 설정합니다.

TLI 로깅이 사용되도록 설정하려면

1. 다음 환경 변수를 웹 서버에 추가합니다.

   ```
   SM_TLI_LOG_FILE
   ```

2. 다음과 같이 디렉터리 및 로그 파일 이름을 변수 값으로 지정합니다.

   directory_name/log_file_name.log

3. 에이전트가 사용되도록 설정되었는지 확인합니다.
4. 웹 서버를 다시 시작합니다.

   TLI 로깅을 사용할 수 있습니다.

저장되는 로그 파일의 수 제한

에이전트에 유지되는 로그 파일의 수를 제한할 수 있습니다. 예를 들어 에이전트 로그가 저장되는 시스템의 디스크 공간을 절약하려면 다음 매개 변수를 사용하여 로그 파일의 수를 제한할 수 있습니다.

LogFilesToKeep

유지되는 에이전트 로그 파일의 수를 지정합니다. 다음 경우에 새 로그 파일이 생성됩니다.

- 에이전트가 시작되는 경우
- LogFileSize 매개 변수 값에 지정된 로그 파일의 크기 제한에 도달하는 경우

이 매개 변수의 값을 변경해도 유지하려는 개수를 초과하는 기존 로그 파일이 자동으로 삭제되지는 *않습니다*. 예를 들어 시스템에 500 개의 로그 파일이 저장되어 있는데 그 중 50 개만 유지하려는 경우 에이전트가 나머지 450 개의 파일을 삭제하지 *않습니다*.

이 매개 변수의 값을 0 으로 설정하면 모든 로그 파일이 유지됩니다.

기본값: 0

다음 단계를 수행하십시오.

1. 시스템의 기존 로그 파일을 보관하거나 삭제합니다.
2. LogAppend 매개 변수의 값을 no 로 설정합니다.
3. LogFilesToKeep 매개 변수 값을 유지할 로그 파일의 수로 변경합니다.

추적 로깅을 설정하는 방법

추적 로깅을 설정하려면 다음을 수행하십시오.

1. 추적 로깅을 구성하고 사용할 수 있도록 설정합니다.
2. 다음 목록을 검토하여 추적 로그에 기록할 항목을 결정합니다.
   - 추적 로그 구성 요소 및 하위 구성 요소
   - 추적 메시지 데이터 필드
   - 데이터 필드 필터
3. 기본 추적 구성 파일을 복제합니다.
4. 기록할 항목을 포함하도록 복제 파일을 수정합니다.
5. 에이전트를 다시 시작합니다.

추적 로깅 구성

추적 로깅을 사용하려면 먼저 추적 로그 파일의 이름, 위치 및 매개 변수를 지정하여 추적 로깅을 구성해야 합니다. 이러한 설정은 파일 자체의 크기와 형식을 제어합니다. 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.

추적 로깅을 구성하려면

1. 웹 서버에서 WebAgentTrace.conf 파일을 찾습니다. 그런 다음 이 파일을 복제합니다.

   참고: IIS 용 SiteMinder 에이전트를 실행하고 WoW64 모드를 사용하여 64 비트 시스템의 32 비트 응용 프로그램을 보호하는 경우에는 복제 파일을 두 개 생성해야 합니다. 64 비트 Windows 운영 환경에는 32 비트 응용 프로그램과 64 비트 응용 프로그램을 위한 디렉터리가 따로 있습니다.

2. 에이전트 구성 개체 또는 로컬 구성 파일을 엽니다.

3. TraceFile 매개 변수를 yes 로 설정합니다.

   참고: 웹 서버의 로컬 구성 파일에서 이 매개 변수의 값을 yes 로 설정하면 정책 서버에 정의된 로깅 설정이 무시됩니다. 예를 들어 LocalConfig.conf 파일에서 이 매개 변수의 값이 yes 로 설정되어 있다고 가정해 봅니다. 이 경우에는 해당 에이전트 구성 개체에서 AllowLocalConfig 매개 변수의 값이 no 로 설정되어 있더라도 에이전트에서 로그 파일을 생성합니다. 또한 LocalConfig.conf 파일에서 관련 로깅 매개 변수를 설정하여 에이전트 구성 개체의 다른 설정을 무시할 수도 있습니다.

4. 다음 매개 변수를 사용하여 추적 로그 파일의 전체 경로를 지정합니다.

   TraceFileName

   추적 로그 파일의 전체 경로를 지정합니다.

   기본값: 기본값 없음

   제한: 이 매개 변수에 파일 이름을 지정하십시오.
   예: *web_agent_home*\log\trace.log

   TraceFileName32

64 비트 Windows 운영 환경에서 실행되고 32 비트 응용 프로그램을 보호하는 IIS 용 SiteMinder 에이전트에 대한 추적 파일의 전체 경로를 지정합니다. IIS 용 SiteMinder 에이전트가 64 비트 Windows 운영 환경에 설치되어 있고 32 비트 Windows 응용 프로그램을 보호하는 경우 이 매개 변수를 설정하십시오. 32 비트 응용 프로그램은 64 비트 Windows 운영 환경의 Wow64 모드에서 실행됩니다. 추적 로깅이 가능하도록 설정된 경우 이 매개 변수가 설정되지 않으면 IIS 용 웹 에이전트에서 파일 이름에 _32 를 추가합니다.

기본값: 기본값 없음

제한: Windows 64 비트 운영 환경에만 적용됩니다. 경로 끝에 추적 파일 이름을 지정하십시오.

예: (Wow64 모드를 사용하는 Windows 64 비트 운영 환경)
web_agent_home\log\WebAgentTrace32.log

5. 다음 매개 변수를 사용하여 1 단계에서 생성한 WebAgentTrace.conf 복제 파일의 전체 경로를 지정합니다.

TraceConfigFile

모니터링할 구성 요소 및 이벤트를 결정하는 WebAgentTrace.conf 구성 파일의 위치를 지정합니다.

기본값: 기본값 없음

예: *web_agent_home*\config\WebAgentTrace.conf

TraceConfigFile32

모니터링할 구성 요소 및 이벤트를 결정하는 WebAgentTrace.conf 구성 파일의 위치를 지정합니다. IIS 용 SiteMinder 에이전트가 64 비트 Windows 운영 환경에 설치되어 있고 32 비트 Windows 응용 프로그램을 보호하는 경우 이 매개 변수를 설정하십시오. 32 비트 응용 프로그램은 64 비트 Windows 운영 환경의 Wow64 모드에서 실행됩니다. 로깅하도록 설정된 경우 이 매개 변수가 설정되지 않으면 IIS 용 웹 에이전트가 파일 이름에 _32 를 추가합니다.

기본값: 기본값 없음

제한: Windows 64 비트 운영 환경에만 적용됩니다. 경로 끝에 구성 파일 이름을 지정하십시오.

예: (Wow64 모드를 사용하는 Windows 64 비트 운영 환경)
web_agent_home\config\WebAgentTrace32.conf

참고: 이 파일을 사용하려면 웹 서버를 다시 시작해야 합니다.

6. 에이전트 구성 개체 또는 로컬 구성 파일에서 다음 매개 변수를 설정하여 추적 로그 파일의 정보 형식을 정의합니다.

TraceAppend

로깅이 호출될 때마다 전체 파일을 다시 쓰지 않고 기존 로그 파일의 끝에 새 로깅 정보를 추가합니다.

기본값: No

IIS 에이전트는 첫 번째 사용자 요청이 제출된 후에만 로그 파일을 생성합니다. 또한 Apache 2.0 웹 에이전트는 Apache 서버가 시작될 때 로그 파일을 생성합니다.

TraceDelimiter

추적 파일에서 필드를 구분하는 사용자 지정 문자를 지정합니다.

기본값: 기본값 없음

예: |

TraceFileSize

추적 파일의 최대 크기(MB)를 지정합니다. 이 제한에 도달하면 웹 에이전트가 새 파일을 생성합니다.

기본값: 0(새 로그 파일이 생성되지 않음)

예: 20(MB)

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.

기본값: Yes

7. 웹 에이전트에서 원하는 활동을 모니터링하도록 구성하려면 WebAgentTrace.conf 파일을 편집합니다.

   프레임워크 웹 에이전트에서는 에이전트 구성 파일에 로컬로 설정된 로그 매개 변수를 동적으로 구성할 수 없습니다. 따라서 매개 변수를 수정하는 경우 변경 내용을 적용하려면 웹 서버를 다시 시작해야 합니다. 그러나 정책 서버의 에이전트 구성 개체에 로그 설정을 구성하는 경우에는 이러한 로그 설정을 동적으로 저장하고 업데이트할 수 있습니다.

   참고: IIS 에이전트는 첫 번째 사용자 요청이 제출된 후에만 로그 파일을 생성합니다. 또한 Apache 2.0 웹 에이전트는 Apache 서버가 시작될 때 로그 파일을 생성합니다.

8. 웹 에이전트에서 새 추적 구성 파일을 사용하도록 웹 서버를 다시 시작합니다.

추적 로그 구성 요소 및 하위 구성 요소

SiteMinder 에이전트에서는 특정 SiteMinder 구성 요소를 모니터링할 수 있습니다. 구성 요소를 모니터링하는 경우 해당 구성 요소의 모든 이벤트가 추적 로그에 기록됩니다. 각 구성 요소에는 에이전트에서 모니터링할 수 있는 하위 구성 요소가 하나 이상 있습니다. 에이전트에서 구성 요소의 일부 이벤트만 기록하게 하려면 모니터링할 하위 구성 요소만 지정할 수 있습니다.

예를 들어 웹 서버에서 에이전트에 대한 싱글 사인온 메시지만 기록하려는 경우 웹 에이전트 구성 요소와 SSO 하위 구성 요소를 지정하면 됩니다.

다음과 같은 구성 요소 및 하위 구성 요소를 사용할 수 있습니다.

AgentFramework

모든 에이전트 프레임워크 메시지를 기록합니다. 프레임워크 에이전트에만 적용됩니다. 다음 하위 구성 요소를 사용할 수 있습니다.

- 관리
- Filter
- HighLevelAgent
- LowLevelAgent
- LowLevelAgentWP

AffiliateAgent

별도 구매 제품인 Federation Security Services 의 일부인 4.x 가맹 에이전트와 관련된 웹 에이전트 메시지를 기록합니다. 프레임워크 에이전트에만 적용됩니다. 다음 하위 구성 요소를 사용할 수 있습니다.

- RequestProcessing

SAMLAgent

SAML 가맹 에이전트와 관련된 웹 에이전트 메시지입니다. 프레임워크 에이전트에만 적용됩니다. 다음 하위 구성 요소를 사용할 수 있습니다.

- RequestProcessing

WebAgent

모든 웹 에이전트 로그 메시지를 기록합니다. IIS 6.0 또는 Apache 2.0 에이전트를 *제외한* 모든 에이전트에 적용됩니다. 다음 하위 구성 요소를 사용할 수 있습니다.

- AgentCore
- Cache
- authentication
- 응답
- Management
- SSO
- Filter

Agent_Functions

모든 에이전트 API 메시지를 기록합니다. 다음 하위 구성 요소를 사용할 수 있습니다.

- Init
- UnInit
- IsProtected
- 로그인
- ChangePassword
- Validate
- Logout
- Authorize
- Audit
- FreeAttributes
- UpdateAttributes
- GetSessionVariables
- SetSessionVariables
- DeleteSessionVariables
- 터널
- GetConfig
- DoManagement

Agent_Con_Manager

에이전트 API 의 내부 처리와 관련된 메시지를 기록합니다. 다음 하위 구성 요소를 사용할 수 있습니다.

- RequestHandler
- Cluster
- 서버
- WaitQueue
- Management
- Statistics

추적 메시지 데이터 필드

메시지에 포함할 데이터 필드를 지정하면 특정 구성 요소의 각 추적 메시지에 포함되는 항목을 정의할 수 있습니다.

데이터 필드에 사용되는 구문은 다음과 같습니다.

```
data:data_field1,data_field2,data_field3
```

일부 데이터 필드는 다음과 같이 표시됩니다.

```
data:message,date,time,user,agentname,IPAddr
```

각 메시지의 필드에 대한 데이터가 없는 경우도 있으므로 빈 필드가 발생할 수 있습니다. 예를 들어 RealmOID 를 데이터 필드로 선택하면 일부 추적 메시지에만 영역 OID 가 표시됩니다.

다음과 같은 데이터 필드를 사용할 수 있습니다.

Message

실제 추적 메시지를 포함합니다.

SrcFile

추적 메시지의 원본 파일 및 행 수를 포함합니다.

Pid

프로세스 ID 를 포함합니다.

Tid

스레드 ID 를 포함합니다.

Date

날짜를 포함합니다.

Time

시간을 포함합니다.

PreciseTime

밀리초를 나타내는 시간을 포함합니다.

Function

추적 메시지가 있는 코드의 함수를 포함합니다.

User

사용자 이름을 포함합니다.

Domain

SiteMinder 도메인을 포함합니다.

Realm

SiteMinder 영역을 포함합니다.

AgentName

사용되는 에이전트 이름을 포함합니다.

TransactionID

트랜잭션 ID 를 포함합니다.

DomainOID

SiteMinder 도메인 OID 를 포함합니다.

IPAddr

클라이언트 IP 주소를 포함합니다.

RequestIPAddr

에이전트가 있는 서버의 IP 주소를 포함합니다.

IPPort

클라이언트 IP 포트를 포함합니다.

CertSerial

인증서 일련 번호를 포함합니다.

SubjectDN

인증서의 주체 DN 을 포함합니다.

IssuerDN

인증서의 발급자 DN 을 포함합니다.

SessionSpec

SiteMinder 세션 사양을 포함합니다.

SessionID

SiteMinder 세션 ID 를 포함합니다.

UserDN

사용자 DN 을 포함합니다.

Resource

요청된 리소스를 포함합니다.

Action

요청된 작업을 포함합니다.

RealmOID

영역 OID 를 포함합니다.

ResponseTime

CA 웹 에이전트나 SDK 에이전트 및 API 응용 프로그램과 연결된 정책 서버의 평균 응답 시간(밀리초)을 포함합니다.

참고: ResponseTime 을 추적 로그에 출력하려면 WebAgentTrace.conf 파일 또는 정책 서버 ACO(에이전트 구성 개체)에 지정된 다른 파일에 ResponseTime 데이터 필드와 Agent_Con_Manager 구성 요소를 포함하고 정책 서버를 다시 시작해야 합니다. Agent_Con_Manager 구성 요소, 즉 Agent API Connection Manager 는 정책 서버로부터 응답을 받을 때마다 ResponseTime 을 계산하고 평균 시간을 유지합니다. 추적 로그에서 ResponseTime 을 찾으려면 [PrintStats]를 검색하십시오.

추적 메시지 데이터 필드 필터

특정 문제에 초점을 맞추려면 데이터 필드 값을 기반으로 필터를 지정하여 추적 로그의 출력 범위를 좁힐 수 있습니다. 예를 들어 index.html 페이지에 문제가 있는 경우 추적 구성 파일에 Resource:==/html 을 지정하여 접미사가 html 인 리소스를 필터링할 수 있습니다. 각 필터는 파일에서 한 행에 하나씩 지정해야 합니다.

필터를 지정할 때 다음 구문을 사용합니다.

data_field:*filter*

다음과 같은 유형의 필터를 사용할 수 있습니다.

- ==(정확히 일치)
- !=(같지 않음)

필터를 지정할 때 다음과 같이 부울 논리를 사용할 수 있습니다.

Action:!=get(get 을 제외한 모든 작업)

Resource:==/html(/html 로 끝나는 모든 리소스)

추적 로그의 내용 결정

WebAgentTrace.conf 파일은 추적 로그의 내용을 결정합니다. 웹 서버에 있는 WebAgentTrace.conf 파일의 설정을 수정하여 추적 로그에 나타낼 구성 요소 및 데이터 항목을 제어할 수 있습니다. 이 파일을 편집할 때 다음 사항을 고려하십시오.

- 항목은 대/소문자를 구분합니다.

  구성 요소, 데이터 필드 또는 필터를 지정하는 경우 값이 WebAgentTrace.conf 파일 명령의 옵션과 정확히 일치해야 합니다.
- 구성 설정 행의 주석 처리를 제거해야 합니다.
- 기존 에이전트 위에 새 에이전트를 설치하기 전에 WebAgentTrace.conf 파일을 수정하면 이 파일을 덮어씁니다. 먼저 파일을 백업하거나 이름을 바꾸십시오. 설치 후 변경 내용을 새 파일에 통합할 수 있습니다.

다음 단계를 수행하십시오.

1. WebAgentTrace.conf 파일을 엽니다.

   참고: 원본 파일을 복제하고 복사본을 변경하는 것이 좋습니다. 복사본을 수정하면 기본 설정이 보존됩니다.

2. 다음 단계를 수행하여 구성 요소 및 하위 구성 요소를 추가합니다.

   a. 에이전트 유형과 일치하는 섹션을 찾습니다. 예를 들어 서버에 Apache 2.0 에이전트가 설치된 경우에는 다음과 비슷한 행을 찾습니다.

   ```
   # For Apache 2.0, Apache 2.2, IIS 7.0 and SunOne Web Agents
   ```

   b. 해당 섹션에서 다음 행을 찾습니다.

   ```
   #components:
   ```

   c. 이 행의 주석 처리를 제거합니다. 그런 다음 콜론 뒤에 원하는 구성 요소 이름을 추가합니다. 구성 요소가 여러 개이면 다음과 같이 쉼표로 구분하십시오.

   ```
   components:  AgentFramework, HTTPAgent
   ```

   d. (선택 사항) 구성 요소 이름 뒤에 원하는 하위 구성 요소의 이름을 추가합니다. 하위 구성 요소 이름은 다음과 같이 슬래시로 구분하십시오.

   ```
   components:  AgentFramework/Administration
   ```

3. 다음 단계를 수행하여 데이터 필드 및 필터를 추가합니다.
   a. 해당 섹션에서 다음 행을 찾습니다.

      ```
      #data:
      ```

   b. 이 행의 주석 처리를 제거합니다. 그런 다음 콜론 뒤에 원하는 데이터 필드를 추가합니다. 데이터 필드가 여러 개이면 다음과 같이 쉼표로 구분하십시오.

      ```
      data: Date, Time, Pid, Tid, TransactionID, Function, Message, IPAddr
      ```

   c. (선택 사항) 데이터 필드 뒤에 콜론, 부울 연산자 및 원하는 값을 지정하여 해당 데이터 필드에 필터를 추가합니다. 필터에 지정하는 값은 정확히 일치해야 합니다. 다음 예제에서는 특정 IP 주소에 대한 활동을 기록하는 필터를 보여 줍니다.

      ```
      data: Date, Time, Pid, Tid, TransactionID, Function, Message,
      IPAddr:==127.0.0.1
      ```

      참고: 각 필터는 파일에서 별도의 행에 지정해야 합니다.
4. 변경 내용을 저장하고 파일을 닫습니다.
5. 웹 서버를 다시 시작하여 변경 내용을 적용합니다.

추적 로그의 내용이 결정되었습니다.

저장되는 추적 로그 파일의 수 제한

SiteMinder 에이전트에 유지되는 추적 로그의 수를 제한할 수 있습니다. 예를 들어 에이전트 로그가 저장되는 시스템의 디스크 공간을 절약하려면 다음 매개 변수를 사용하여 추적 로그의 수를 제한할 수 있습니다.

TraceFilesToKeep

유지되는 SiteMinder 에이전트 추적 로그 파일의 수를 지정합니다. 다음 경우 새 추적 로그가 생성됩니다.

- 에이전트가 시작되는 경우
- TraceFileSize 매개 변수 값에 지정된 추적 로그의 크기 제한에 도달하는 경우

이 매개 변수의 값을 변경해도 유지하려는 개수를 초과하는 기존 추적 로그가 자동으로 삭제되지는 *않습니다*. 예를 들어 시스템에 500 개의 추적 로그가 저장되어 있는데 그 중 50 개만 유지하려는 경우 에이전트가 나머지 450 개의 추적 로그를 삭제하지 *않습니다*.

이 매개 변수의 값을 0 으로 설정하면 모든 추적 로그가 유지됩니다.

기본값: 0

다음 단계를 수행하십시오.

1. 시스템의 기존 추적 로그를 보관하거나 삭제합니다.
2. TraceAppend 매개 변수의 값을 no 로 설정합니다.
3. TraceFilesToKeep 매개 변수 값을 유지할 추적 로그의 수로 변경합니다.

Agent Connection Manager 추적 로그를 사용하여 자세한 에이전트 연결 데이터 수집

웹 에이전트와 정책 서버 간 연결에 대한 자세한 정보를 수집하려면 Agent Collection Manager 에서 수집한 정보를 포함하는 추적 로그 파일을 생성합니다.

자세한 웹 에이전트 연결 데이터를 수집하려면

1. 에이전트 구성 개체 또는 로컬 구성 파일을 엽니다.
2. TraceFile 매개 변수의 값을 yes 로 설정합니다.

   참고: 웹 서버의 로컬 구성 파일에서 이 매개 변수의 값을 yes 로 설정하면 정책 서버에 정의된 로깅 설정이 무시됩니다. 예를 들어 LocalConfig.conf 파일에서 이 매개 변수의 값이 yes 로 설정되면 정책 서버의 해당 에이전트 구성 개체에서 AllowLocalConfig 매개 변수의 값이 no 로 설정된 경우에도 로그 파일이 생성됩니다. 또한 정책 서버의 추적 로그 설정을 무시하려면 LocalConfig.conf 파일에서 파일 이름, 크기 등을 정의하는 관련 추적 로깅 매개 변수를 설정하십시오.
3. 에이전트 연결 데이터에 대한 추적 로그 파일의 전체 경로를 TraceFileName 매개 변수에 지정합니다. 이 파일에는 추적 로그 출력이 포함됩니다.
4. TraceConfigFile 매개 변수 값을 다음 파일의 전체 경로로 설정합니다.

   web_agent_home/config/AgentConMgr.conf

   web_agent_home

   SiteMinder 에이전트가 설치된 디렉터리를 나타냅니다.

   기본값(Windows 32 비트의 SiteMinder 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent

   기본값(Windows 64 비트의 SiteMinder IIS 웹 에이전트 설치만 해당): C:\Program Files\CA\webagent\win64

   기본값(64 비트 시스템에서 작동하는 Windows 32 비트 응용 프로그램 - Wow64 모드의 IIS 용 SiteMinder 웹 에이전트만 해당): C:\Program Files (x86)\webagent\win32

   기본값(UNIX/Linux 시스템): /opt/ca/webagent
5. 다음 매개 변수를 설정하여 에이전트 연결 데이터에 대한 추적 로그 파일의 형식을 정의합니다.

   TraceAppend

로깅이 호출될 때마다 전체 파일을 다시 쓰지 않고 기존 로그 파일의 끝에 새 로깅 정보를 추가합니다.

기본값: No

TraceDelimiter

추적 파일에서 필드를 구분하는 사용자 지정 문자를 지정합니다.

기본값: 기본값 없음

예: |

TraceFileSize

추적 파일의 최대 크기(MB)를 지정합니다. 이 제한에 도달하면 웹 에이전트가 새 파일을 생성합니다.

기본값: 0(새 로그 파일이 생성되지 않음)

예: 20(MB)

TraceFormat

추적 파일에서 메시지를 표시하는 방식을 지정합니다. 다음 옵션 중 *하나*를 선택하십시오.

- default - 필드를 대괄호([])로 묶습니다.
- fixed - 고정 너비로 필드를 사용합니다.
- delim - 선택한 문자를 사용하여 필드를 구분합니다.
- xml - XML 같은 태그를 사용합니다. 웹 에이전트에서는 DTD 또는 스타일시트가 제공되지 *않습니다*.

기본값: default(대괄호)

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.

기본값: Yes

6. 웹 서버를 다시 시작하여 새 설정을 적용합니다.

자세한 웹 에이전트 연결 정보가 수집됩니다.

참고: SiteMinder 12.52 SP1 에서는 BusyHandleCount 및 FreeHandleCount 특성이 사용되지 않습니다.

제 21 장: 에이전트 구성 문제 해결

제 22 장: 에이전트 오류 코드

IIS 용 에이전트 문제 해결 로그

증상:

IIS 용 SiteMinder 에이전트의 문제를 해결하는 데 도움이 되는 IIS 7.x 로그 파일이 있는지 확인해야 합니다.

해결책:

다음 로그 파일을 여십시오.

web_agent_home\log\IIS70Trace.log

이 로그 파일에는 다음과 같은 유형의 정보가 포함됩니다.

- 응용 프로그램 풀 ID
- 응용 프로그램 풀 파이프라인 모드
- 필터 유형(ISAPI 필터가 아닌 네이티브 HTTP 모듈)
- 32 비트 또는 64 비트

로그 파일에 중복 LLAWP 오류 표시

증상:

로그 파일에 다음과 같은 오류가 표시됩니다.

```
Duplicate LLAWP
```

해결책:

응용 프로그램 풀이 재순환되는 경우 이 오류가 발생합니다. 현재 LLAWP 프로세스가 완전히 종료되기 전에 응용 프로그램 풀에서 새 LLAWP 프로세스를 시작하려는 경우입니다.

이 오류가 나타나지 않게 하려면 순환 겹침이 허용되지 않도록 IIS 서버의 응용 프로그램 풀을 구성하십시오. 그러면 응용 프로그램 풀에서 새 LLAWP 프로세스가 시작되기 전에 현재 LLAWP 프로세스가 중지될 때까지 대기합니다.

참고: 자세한 내용을 보려면 IIS 웹 사이트에서 "DisallowOverlappingRotation"을 검색하십시오.

사용자 지정 오류 페이지가 나타나지 않음

Oracle Directory Enterprise Edition(이전의 Oracle iPlanet Directory Server Enterprise Edition)에 해당

증상:

다음 구성 매개 변수를 설정했지만 사용자에게 일반 서버 오류 메시지가 표시됩니다.

CSSErrorFile

사용자가 교차 사이트 스크립팅 문자가 포함된 URL 을 열려고 할 경우 사용자에게 표시할 사용자 지정 오류 메시지 파일 또는 URL 의 위치를 지정합니다.

기본값: 기본값 없음

ServerErrorFile

사용자에게 서버 오류가 발생할 경우 사용자 지정 오류 페이지를 표시하도록 웹 에이전트에 지시합니다. 이 매개 변수에 파일 경로 또는 URL 을 지정하십시오.

기본값: 기본값 없음

해결책:

다음 단계를 수행하십시오.

1. 웹 서버의 *instance_name*-obj.conf 파일을 엽니다.
2. 다음 행을 찾습니다.

   ```
   AuthTrans fn="SiteMinderAgent"
   ```

3. 다음과 같이 이 행의 끝에 UseOutputStreamSize="0"을 추가합니다.

   ```
   AuthTrans fn="SiteMinderAgent" UseOutputStreamSize="0"
   ```

4. 파일을 저장한 후 웹 서버를 다시 시작합니다.

추적 메시지를 초기화할 수 없음

IIS 7.x 에 해당

증상:

같은 IIS 웹 서버에서 32 비트 에이전트와 64 비트 에이전트를 실행하고 있습니다. 32 비트 에이전트에 대한 추적 로그를 기록해야 하는데 다음과 같은 메시지가 표시됩니다.

```
[INFO] LLAWP: Unable to initialize tracing.
```

64 비트 에이전트 추적 로그는 영향을 받지 않습니다.

해결책:

에이전트에 대해 다음 구성 매개 변수가 정의되어 있는지 확인하십시오.

TraceConfigFile32

모니터링할 구성 요소 및 이벤트를 결정하는 WebAgentTrace.conf 구성 파일의 위치를 지정합니다. IIS 용 SiteMinder 에이전트가 64 비트 Windows 운영 환경에 설치되어 있고 32 비트 Windows 응용 프로그램을 보호하는 경우 이 매개 변수를 설정하십시오. 32 비트 응용 프로그램은 64 비트 Windows 운영 환경의 Wow64 모드에서 실행됩니다. 로깅하도록 설정된 경우 이 매개 변수가 설정되지 않으면 IIS 용 웹 에이전트가 파일 이름에 _32 를 추가합니다.

기본값: 기본값 없음

제한: Windows 64 비트 운영 환경에만 적용됩니다. 경로 끝에 구성 파일 이름을 지정하십시오.

예: (Wow64 모드를 사용하는 Windows 64 비트 운영 환경) *web_agent_home*\config\WebAgentTrace32.conf

LogFileName32

32 비트 응용 프로그램을 보호하는 64 비트 Windows 운영 환경에서 IIS 용 SiteMinder 웹 에이전트에 대한 로그 파일의 전체 경로를 지정합니다. 32 비트 응용 프로그램은 64 비트 Windows 운영 환경의 Wow64 모드에서 실행됩니다. 로깅하도록 설정된 경우 이 매개 변수가 설정되지 않으면 IIS 용 웹 에이전트에서 로그 파일 이름에 _32 를 추가합니다.

기본값: No

제한: Windows 64 비트 운영 환경에만 적용됩니다. 경로 끝에 파일 이름을 지정하십시오.

예: (Wow64 모드를 사용하는 Windows 64 비트 운영 환경)
web_agent_home\log\WebAgent32.log

TraceFileName32

64 비트 Windows 운영 환경에서 실행되고 32 비트 응용 프로그램을 보호하는 IIS 용 SiteMinder 에이전트에 대한 추적 파일의 전체 경로를 지정합니다. IIS 용 SiteMinder 에이전트가 64 비트 Windows 운영 환경에 설치되어 있고 32 비트 Windows 응용 프로그램을 보호하는 경우 이 매개 변수를 설정하십시오. 32 비트 응용 프로그램은 64 비트 Windows 운영 환경의 Wow64 모드에서 실행됩니다. 추적 로깅이 가능하도록 설정된 경우 이 매개 변수가 설정되지 않으면 IIS 용 웹 에이전트에서 파일 이름에 _32 를 추가합니다.

기본값: 기본값 없음

제한: Windows 64 비트 운영 환경에만 적용됩니다. 경로 끝에 추적 파일 이름을 지정하십시오.

예: (Wow64 모드를 사용하는 Windows 64 비트 운영 환경)
web_agent_home\log\WebAgentTrace32.log

에이전트와 정책 서버가 방화벽으로 분리된 경우 KeepAlive 가 사용되도록 설정

증상:

에이전트와 정책 서버 사이에서 방화벽을 사용하고 있는데, 페이지에 액세스하려고 할 때 에이전트에서 500 오류가 반환되는 경우가 있습니다.

해결책:

다음 단계를 수행하여 에이전트에서 KeepAlive 가 사용되도록 설정하십시오.

1. 웹 에이전트를 호스트하는 컴퓨터에서 다음 환경 변수를 찾습니다.

   SM_ENABLE_TCP_KEEPALIVE

2. 앞의 환경 변수의 값을 1 로 설정합니다.

일본어 페이지가 잘못 렌더링됨(153202, 153609)

증상:

다음과 같은 이유 때문에 사용자가 다른 페이지로 리디렉션되는 경우 결과 페이지의 콘텐츠가 제대로 렌더링되지 않습니다.

- 새 등록
- 암호 변경
- 암호 만료됨

해결책:

Apache 웹 서버의 http.conf 파일에 다음 행을 추가하십시오.

```
"BrowserMatch ".*" suppress-error-charset"
```

http.conf 파일을 저장한 후 웹 서버를 다시 시작하여 이 설정을 적용하십시오.

영어가 아닌 입력 문자에 정크 문자가 포함됨

UNIX/Linux 에 해당

증상:

영어가 아닌 일부 입력 문자가 콘솔 창에 올바로 표시되지 않습니다.

해결책:

콘솔 창의 터미널 설정을 확인하십시오. 콘솔이 입력 문자의 높은(8) 비트를 삭제하지 않는지 확인하십시오. 다음 명령을 실행합니다.

```
stty –istrip
```

제 23 장: 에이전트 오류 코드

이 섹션은 다음 항목을 포함하고 있습니다.

00-0001

원인:

IP 주소에서 에이전트 이름을 확인할 수 없습니다.

조치:

에이전트 구성을 검토하고 웹 서버에서 제공하는 각 호스트 주소에 해당 AgentName 이 매핑되어 있는지 또는 DefaultAgentName 이 제대로 설정되어 있는지 확인하십시오.

00-0002

원인:

BadUrlChars 매개 변수에 정의된 문자 또는 잘못된 문자가 URL 에서 검색되었습니다.

조치:

다음 작업 중 하나를 수행하십시오.

- 잘못된 문자를 URL 에서 제거합니다.
- BadUrlChars 매개 변수 목록에서 문자를 제거하여 해당 문자가 차단되지 않도록 합니다.

00-0004

원인:

SSLCRED 쿠키에 오류 상태가 포함되어 있습니다.

조치:

SCC(보안 자격 증명 수집기)로 사용되는 웹 에이전트를 조사하여 구성을 확인하십시오.

일반적으로 이 오류는 SCC 에이전트가 환경에서 자격 증명을 가져올 수 없는 경우에만 발생합니다. 이 오류는 구성 오류가 발생할 수 있음을 나타냅니다.

00-0005

원인:

FORMCRED 쿠키에 오류 상태가 포함되어 있습니다.

조치:

FCC(양식 자격 증명 수집기)로 사용되는 웹 에이전트를 조사하여 구성을 확인하십시오.

일반적으로 이 오류는 FCC 에이전트가 환경에서 자격 증명을 가져올 수 없는 경우에만 발생합니다. 이 오류는 구성 오류가 발생할 수 있음을 나타냅니다.

00-0006

원인:

NTLM 으로 보호된 리소스가 리소스 캐시에 없습니다.

조치:

Windows 인증 체계 설정을 조사하여 구성을 확인하십시오.

00-0007

원인:

ASCII 인코딩 오류가 있습니다. 이 오류는 웹 에이전트 내부 오류입니다.

조치:

웹 서버 및 웹 에이전트를 조사하여 서비스 불안정을 진단하십시오.

고객 지원부에 웹 에이전트 로그 및 구성 파일 검토를 요청하십시오.

추가 정보:

CA 에 문의 (페이지 4)

00-0008

원인:

SSL 인증이 실패했습니다. 이 오류는 인증서가 잘못되었거나 사용자가 인증되지 않았음을 나타냅니다.

조치:

다른 인증서를 사용하거나 SSL 인증 체계 구성을 조사하여 문제를 확인하십시오.

00-0009

원인:

SSL 자격 증명이 없거나 잘못되었습니다.

조치:

다른 인증서 또는 사용자 이름/암호 쌍을 사용하십시오. 또는 SSL 인증 체계 구성을 조사하여 문제를 확인하십시오.

00-0010

원인:

액세스가 거부되었습니다. 이 오류는 액세스를 차단하는 일반 오류를 나타냅니다.

조치:

웹 에이전트 및 정책 서버 로그를 조사하여 오류의 근본 원인을 확인하십시오.

00-0011

원인:

자격 증명 수집기 오류입니다. 이 오류는 액세스를 차단하는 양식 또는 SSL 기반 고급 인증의 일반 오류를 나타냅니다.

조치:

다음 작업을 수행하십시오.

- 웹 에이전트 및 정책 서버 로그를 조사하여 오류의 근본 원인을 확인합니다.
- 고급 인증 체계 설정을 조사하여 문제를 확인합니다.

00-0012

원인:

암호화 오류입니다. 이 오류는 웹 에이전트 내부 오류를 나타냅니다.

조치:

다음 작업을 수행하십시오.

- 웹 서버 및 웹 에이전트를 조사하여 서비스 불안정을 진단합니다.
- 키 저장소 설정을 검토하여 사용 중인 에이전트 키가 올바른지 확인합니다.
- 고객 지원부에 웹 에이전트 로그 및 구성 파일 검토를 요청합니다.

추가 정보:

CA 에 문의 (페이지 4)

00-0013

원인:

에이전트 구성 오류입니다. 시작할 때 오류가 하나 이상 발생하여 웹 에이전트를 올바로 구성할 수 없습니다.

조치:

다음 작업을 수행하십시오.

- Windows 의 경우 응용 프로그램 이벤트 로그에서 자세한 정보를 확인합니다.
- Apache 에이전트의 경우 Apache 오류 로그에서 자세한 정보를 확인합니다.
- Oracle iPlanet UNIX 에이전트의 경우 셸 프롬프트에서 Oracle iPlanet 을 시작하고 STDERR 를 통해 표시되는 오류를 검색합니다.
- SmHost.conf 파일이 있는지(호스트가 제대로 등록됨) 그리고 올바른 항목이 포함되어 있는지 확인합니다.
- 에이전트 구성 파일에 올바른 SmHost.conf 파일을 가리키는 HostConfigFile 항목이 포함되어 있는지 확인합니다.
- AgentConfigObject 에 올바른 값이 포함되어 있는지 확인합니다.

00-0014

원인:

사용자를 로그아웃할 수 없습니다.

조치:

다음 파일에서 자세한 정보를 확인하십시오.

- 웹 에이전트 로그 파일
- 웹 에이전트 추적 파일
- 정책 서버 로그 파일
- 정책 서버 추적 파일

00-0015

원인:

SiteMinder 계정 서버에서 감사 요청에 대해 SM_AGENTAPI_NO 를 응답했습니다.

조치:

다음 파일에서 자세한 정보를 확인하십시오.

- 정책 서버 로그 파일
- 정책 서버 추적 파일

00-0016

원인:

FQ 호스트 이름을 확인할 수 없습니다.

조치:

웹 에이전트 로그를 검토하여 에이전트가 확인하려고 하는 호스트 이름을 판별하십시오. 호스트 이름이 올바르면 해당 에이전트가 실행되는 웹 서버의 DNS 설정을 검토해야 합니다.

00-0017

원인:

리디렉션 대상이 잘못되었습니다.

조치:

이 메시지를 보고하는 웹 에이전트의 로그 파일을 검사하여 처리 중인 URL(대개 FCC 또는 다른 고급 인증 URL)을 찾고 TARGET CGI 매개 변수의 값이 올바른지 확인하십시오.

10-0001

원인:

'SERVER_NAME' HTTP 변수를 읽을 수 없습니다.

조치:

웹 브라우저 및 웹 서버가 HTTP 1.0 과 호환되는지 확인하십시오.

10-0002

원인:

'URL' HTTP 변수를 읽을 수 없습니다.

조치:

웹 브라우저 및 웹 서버가 HTTP 1.0 과 호환되는지 확인하십시오.

10-0003

원인:

'method' HTTP 변수를 읽을 수 없습니다.

조치:

웹 브라우저 및 웹 서버가 HTTP 1.0 과 호환되는지 확인하십시오.

10-0004

원인:

'host' HTTP 변수를 읽을 수 없습니다.

조치:

웹 브라우저 및 웹 서버가 HTTP 1.0 과 호환되는지 확인하십시오.

10-0005

원인:

'URI' HTTP 변수를 읽을 수 없습니다.

조치:

웹 브라우저 및 웹 서버가 HTTP 1.0 과 호환되는지 확인하십시오.

10-0007

원인:

URL 이 너무 깁니다.

조치:

MaxUrlSize 매개 변수 설정을 늘리십시오. 기본 설정은 4096 바이트입니다.

추가 정보:

최대 URL 크기 설정 (페이지 314)

20-0001

원인:

SiteMinder 계정 서버에 연결할 수 없거나 예기치 않은 정책 서버 오류가 발생했습니다.

조치:

다음 작업을 수행하십시오.

- 정책 서버 로그에서 오류에 대한 자세한 정보를 확인합니다.
- 정책 서버를 ping 하여 웹 에이전트와 정책 서버 간 연결을 확인합니다. 에이전트와 정책 서버 사이에 방화벽이 구성된 경우에는 다음 서비스 포트가 방화벽으로 차단되지 않는지 확인해야 합니다.
  - 44441(계정)
  - 44442(인증)
  - 44443(권한 부여)

20-0002

원인:

SiteMinder 인증 서버에 연결할 수 없거나 예기치 않은 정책 서버 오류가 발생했습니다.

조치:

다음 작업을 수행하십시오.

- 정책 서버 로그에서 오류에 대한 자세한 정보를 확인합니다.
- 정책 서버를 ping 하여 웹 에이전트와 정책 서버 간 연결을 확인합니다. 에이전트와 정책 서버 사이에 방화벽이 구성된 경우에는 다음 서비스 포트가 방화벽으로 차단되지 않는지 확인해야 합니다.
  - 44441(계정)
  - 44442(인증)
  - 44443(권한 부여)

20-0003

원인:

SiteMinder 권한 부여 서버에 연결할 수 없거나 예기치 않은 정책 서버 오류가 발생했습니다.

조치:

다음 작업을 수행하십시오.

- 정책 서버 로그에서 오류에 대한 자세한 정보를 확인합니다.
- 정책 서버를 ping 하여 웹 에이전트와 정책 서버 간 연결을 확인합니다. 에이전트와 정책 서버 사이에 방화벽이 구성된 경우에는 다음 서비스 포트가 방화벽으로 차단되지 않는지 확인해야 합니다.
  - 44441(계정)
  - 44442(인증)
  - 44443(권한 부여)

30-0026

원인:

암호 서비스 리디렉션 URL 을 사용할 수 없습니다.

조치:

암호 서비스 리디렉션 URL 을 구성했는지 확인하십시오.

부록 A: 에이전트 매개 변수

이 섹션은 다음 항목을 포함하고 있습니다.

에이전트 구성 매개 변수 목록

다음 표에서는 에이전트 구성 매개 변수를 보여 줍니다.

설정할 매개 변수	참조 항목
AcceptTPCookie	SDK 타사 쿠키에 대한 지원 구성 (페이지 109)
AgentConfigObject	로컬 구성 파일에만 있는 매개 변수 (페이지 40)
AgentName	AgentName 및 DefaultAgentName 값 설정 (페이지 56)
AgentNamesAreFQHostNames	혼합 환경에서 자격 증명 수집기 구성 (페이지 211)
AgentWaitTime	네트워크 대기 시간 조정 (페이지 62)
AllowCacheHeaders	HTTP 헤더 리소스가 캐시되는 방식 제어 (페이지 157)
AllowLocalConfig	로컬 구성 구현 (페이지 41) 로컬 구성 매개 변수에 대한 변경 제한 (페이지 44)
AppendIISServerLog	IIS 서버 로그에 사용자 이름 및 트랜잭션 ID 기록 (페이지 317)
autoauthorizeoptions	OPTIONS 메서드를 사용하는 리소스에 자동 액세스 허용 (페이지 245)
BadCSSChars	교차 사이트 스크립팅에 대해 웹 사이트 보호 (페이지 82)
BadFormChars	잘못된 양식 문자 지정
BadQueryChars	잘못된 쿼리 문자 지정 (페이지 88)
BadUrlChars	잘못된 URL 문자 지정 (페이지 90)
CacheAnonymous	익명 사용자 캐시 (페이지 385)
CCCExt	쿠키 공급자 지정 (페이지 264)
ConformToRFC2047	RFC 2047 준수 안 함 (페이지 236)

설정할 매개 변수	참조 항목
ConstructFullPwsvcUrl	정규화된 URL 을 사용하여 암호 서비스 리디렉션 (페이지 236)
CookieDomain	싱글 사인온을 구성하는 방법 (페이지 251)
CookieDomainScope	쿠키 도메인 확인 구현 (페이지 107)
CookiePath	에이전트 쿠키의 쿠키 경로 지정 (페이지 104)
CookiePathScope	에이전트 쿠키의 쿠키 경로 지정 (페이지 104)
CookieProvider	싱글 사인온을 구성하는 방법 (페이지 251)
CookieValidationPeriod	유효성 검사 기간 및 만료된 쿠키 URL 을 사용하여 세션 쿠키의 오용 방지 (페이지 124)
CSSChecking	교차 사이트 스크립팅을 확인하도록 웹 에이전트 구성 (페이지 84)
CSSErrorFile	오류 처리를 설정하는 방법 (페이지 165)
Custom401ErrorFile	오류 처리를 설정하는 방법 (페이지 165)
CustomIpHeader	IP 주소 유효성 검사 구성 (페이지 155)
DecodeQueryData	쿼리 데이터 디코딩 (페이지 110)
DefaultAgentName	AgentName 및 DefaultAgentName 값 설정 (페이지 56)
DefaultHostName	HOST 헤더를 전송하지 않는 테스트 도구 수용 (페이지 375)
DefaultPassword	IIS 프록시 사용자 계정 사용 (페이지 331)
DefaultUsername	IIS 프록시 사용자 계정 사용 (페이지 331)
DeleteCerts	Stronghold 서버에서 인증서 삭제 (페이지 342)
DisableAuthSrcVars	기본 HTTP 헤더 변수 사용 안 함 (페이지 162)
DisableDirectoryList	Sun Java System 서버에서 디렉터리 검색 제한 (페이지 343)
DisableDNSLookups	DNS DOS 공격 방지 (페이지 93)
DisableDotDotRule	복잡한 URI 처리 (페이지 112)
DisableSessionVars	기본 HTTP 헤더 변수 사용 안 함 (페이지 162)
DisableUserNameVars	기본 HTTP 헤더 변수 사용 안 함 (페이지 162)
DisableWindowsSecurityContext	IIS 의 에이전트에서 Windows 보안 컨텍스트 사용 안 함 (페이지 332)
DisallowUTF8NonCanonical	교차 사이트 스크립팅 공격에 대해 J2EE 응용 프로그램 보호

설정할 매개 변수	참조 항목
DLPExclusionList	DLP 콘텐츠 분류에서 리소스 제외 **참고:** 자세한 내용은 *SiteMinder 구현 안내서*를 참조하십시오.
DLPSupportEnabled	SharePoint 에이전트 구성 개체 수정 **참고:** 자세한 내용은 *SiteMinder 구현 안내서*를 참조하십시오.
DominoDefaultUser	Domino 서버를 사용하여 사용자 인증 (페이지 359)
DominoLegacyDocumentSupport	Lotus Notes 문서에 대해 사용자가 요청한 작업 처리 (페이지 369)
DominoLookUpHeaderForLogin	인증에 SiteMinder 헤더 사용 (페이지 364)
DominoMapUrlForRedirect	Domino 웹 에이전트를 사용하여 FCC 리디렉션을 위한 URL 매핑 (페이지 185)
DominoNormalizeUrls	Domino 웹 에이전트를 사용하여 FCC 리디렉션을 위한 URL 매핑 (페이지 185)
DominoSuperUser	Domino 슈퍼 사용자로 인증 (페이지 360)
DominoUseHeaderForLogin	인증에 SiteMinder 헤더 사용 (페이지 364)
DominoUserForAnonAuth	Domino 에서 익명 SiteMinder 인증 체계 사용 (페이지 365)
EarlyCookieCommit	IIS 용 에이전트에서 쿠키를 설정해야 하는 경우 확인 (페이지 334)
EnableAuditing	사용자 활동을 추적하도록 감사 구성 (페이지 80)
EnableCookieProvider	쿠키 공급자 사용 안 함 (페이지 265)
EnableFCCWindowsAuth	Windows 인증을 허용하도록 FCC 구성 (페이지 195)
EnableFormCache	양식 캐시 구성 (페이지 209)
EnableIntroscopeApiSupport	CA Technologies Wily Introscope 를 사용하여 웹 에이전트 모니터링 (페이지 390)
EnableMonitoring	OneView 모니터를 사용하여 웹 에이전트 모니터링 (페이지 389)
EnableOtherAuthTrans	여러 개의 AuthTrans 기능 처리 (페이지 344)
EnableWebAgent	웹 에이전트 사용 (페이지 73)
EncryptAgentName	에이전트 이름 암호화 (페이지 60)
EnforceRealmTimeouts	여러 영역에 시간 만료를 적용하는 방법 (페이지 132)

설정할 매개 변수	참조 항목
ExpiredCookieURL	유효성 검사 기간 및 만료된 쿠키 URL 을 사용하여 세션 쿠키의 오용 방지 (페이지 124)
ExpireForProxy	프록시 서버 뒤의 에이전트 구성 (페이지 212)
FCCCompatMode	혼합 환경에서 FCC 및 NTC 사용 (페이지 212)
FCCExt	IIS 및 Domino 웹 서버에 대한 자격 증명 수집기 설정
FCCForceIsProtected	양식 인증을 위해 FCC 에서 영역 컨텍스트 설정 (페이지 207)
Fcchtmlencoding	웹 에이전트 FCC 페이지에서 교차 사이트 스크립팅 공격 방지
ForceCookieDomain	쿠키 도메인 적용 (페이지 106)
ForceFQHost	쿠키 도메인 적용 (페이지 106)
ForceIISProxyUser	IIS 프록시 사용자 계정 사용 (페이지 331)
FormCacheTimeout	양식 캐시 구성 (페이지 209)
GetPortFromHeaders	HTTP HOST 요청을 사용하여 포트 번호 가져오기 (페이지 338)
HostConfigFile	로컬 구성 파일에만 있는 매개 변수 (페이지 40)
HTTPHeaderEncodingSpec	HTTP 헤더 인코딩 사양 설정
HttpsPorts	HTTPS 포트 정의 (페이지 109)
IdleTimeoutURL	세션 시간 만료 후 사용자 리디렉션 (페이지 130)
IgnoreCPForNotprotected	보호되지 않은 리소스에 대해 쿠키 공급자 무시 (페이지 261)
IgnoreExt	보호되지 않은 리소스의 파일 확장명을 무시하여 오버헤드 감소 (페이지 392)
IgnoreHost	웹 에이전트에서 무시할 가상 서버 지정 (페이지 173)
IgnoreQueryData	쿼리 데이터 무시 (페이지 395)
IgnoreUrl	URI 무제한 액세스 허용 (페이지 397)
IISCacheDisable	쿠키를 포함하는 서버 응답의 캐싱 방지 (페이지 333)
LegacyCookieProvider	성능 향상을 위해 FCC 영역 컨텍스트 확인 사용 안 함 (페이지 207)
LegacyEncoding	레거시 URL 인코딩 조정 (페이지 374)
LegacyStreamingBehavior	POST 요청에서 콘텐츠 유형이 전송되는 방식 선택 (페이지 341)

설정할 매개 변수	참조 항목
LegacyTransferEncodingBehavior	Apache 웹 에이전트에서 레거시 응용 프로그램 사용 (페이지 338)
LegacyVariables	HTTP 헤더에 레거시 변수 사용
LimitCookieProvider	쿠키 공급자 기능 제한 (페이지 252)
LoadPlugin	프레임워크 에이전트의 WebAgent.conf 파일 (페이지 37)
localconfigfile	프레임워크 에이전트의 WebAgent.conf 파일 (페이지 37)
LogAppend	오류 로깅 설정 및 사용 (페이지 403)
LogFile	오류 로깅 설정 및 사용 (페이지 403)
LogFileName	오류 로깅 설정 및 사용 (페이지 403)
LogFileSize	오류 로깅 설정 및 사용 (페이지 403)
LogFilesToKeep	저장되는 로그 파일의 수 제한 (페이지 406)
LogLocalTime	오류 로깅 설정 및 사용 (페이지 403)
LogoffUri	싱글 사인온에 대해 전체 로그오프를 구성하는 방법 (페이지 269)
LowerCaseHTTP	HTTP 헤더에 소문자 사용 (페이지 160)
LowerCaseProtocolSpecifier	소문자를 사용하여 URL 프로토콜 지정 (페이지 190)
MasterCookiePath	에이전트 쿠키의 쿠키 경로 지정 (페이지 104)
MaxResourceCacheSize	리소스 캐시의 최대 크기 설정 (페이지 386)
MaxSessionCacheSize	사용자 세션 캐시의 최대 크기 설정 (페이지 387)
MaxTimeoutURL	세션 시간 만료 후 사용자 리디렉션 (페이지 130)
MaxUrlSize	최대 URL 크기 설정 (페이지 314)
NTCExt	NTC(NTLM Credential Collector) 지정 (페이지 209)
OverlookSessionAsPattern	세션 쿠키 생성 또는 업데이트 방지
OverlookSessionForMethods	세션 쿠키 생성 또는 업데이트 방지
OverlookSessionForMethodUri	메서드와 URI 를 기반으로 세션 쿠키 생성 또는 업데이트 방지 (페이지 127)
OverlookSessionForUrls	세션 쿠키 생성 또는 업데이트 방지
OverrideIgnoreExtFilter	확장명이 없는 리소스 보호 (페이지 94)

설정할 매개 변수	참조 항목
P3PCompactPolicy	P3P 압축 정책을 준수하도록 웹 에이전트 구성 (페이지 114)
PersistentCookies	영구 쿠키 설정 (페이지 103)
PersistentIPCheck	IP 주소를 비교하여 보안 위반 방지 (페이지 98)
PostPreservationFile	프레임워크 에이전트와 기존 에이전트 간의 POST 보존 사용 (페이지 186)
PreserveHeaders	HTTP 헤더 유지 (페이지 156)
PreservePostData	POST 보존 사용 또는 사용 안 함 (페이지 94)
ProxyAgent	SiteMinder 리버스 프록시 배포 고려 사항 (페이지 305)
ProxyDefinition	SiteMinder 리버스 프록시 배포 고려 사항 (페이지 305)
ProxyHeadersAutoAuth	Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)
ProxyHeadersAutoAuth10	Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)
ProxyHeadersProtected	Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)
ProxyHeadersProtected10	Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)
ProxyHeadersUnprotected	Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)
ProxyHeadersUnprotected10	Cache-Control 및 ExpireForProxy 헤더 설정 사용자 지정 (페이지 290)
ProxyTimeout	SiteMinder 리버스 프록시 배포 고려 사항 (페이지 305)
ProxyTrust	프록시 서버 뒤의 에이전트 구성 (페이지 288)
PSPollInterval	에이전트에서 정책 또는 키 업데이트 검사 간격 변경 (페이지 78)
RemoteUserVar	REMOTE_USER 변수를 설정하기 위해 웹 에이전트 구성 (페이지 142)
ReqCookieErrorFile	오류 처리 설정 (페이지 165)
RequireClientIP	IP 주소 유효성 검사 구성 (페이지 155)
RequireCookies	기본 인증에 쿠키 필요 (페이지 100)

설정할 매개 변수	참조 항목
ResourceCacheTimeout	리소스 항목이 캐시에 유지되는 시간 제어 (페이지 388)
SaveCredsTimeout	저장된 자격 증명의 만료 시간 설정 (페이지 383)
SCCExt	IIS 및 Domino 웹 서버에 대한 자격 증명 수집기 설정
SecureApps	응용 프로그램 보안 (페이지 95)
SecureURLs	싱글 사인온에 SecureUrls 구성 (페이지 263)
ServerErrorFile	오류 처리 설정 (페이지 165)
SessionGracePeriod	세션 유예 기간 수정 (페이지 122)
SessionUpdatePeriod	세션 업데이트 간격 수정 (페이지 123)
SetRemoteUser	REMOTE_USER 변수를 설정하기 위해 웹 에이전트 구성 (페이지 142)
SFCCExt	IIS 및 Domino 웹 서버에 대한 자격 증명 수집기 설정
SkipDominoAuth	Domino 서버를 사용하여 사용자 인증 (페이지 359)
SSOTrustedZone	트러스트 순서 및 장애 조치 (페이지 286)
SSOZoneName	보안 영역 구성 (페이지 282)
StoreSessioninServer	단일 사용 세션 쿠키 설정 (페이지 128)
SuppressServerHeader	URLScan 유틸리티를 사용하는 경우 HTTP Server 헤더 제거 (페이지 312)
TargetAsRelativeURI	자격 증명 수집기 리디렉션을 위해 상대 대상 사용 (페이지 182)
TraceAppend	추적 로깅 구성 (페이지 408)
TraceConfigFile	추적 로깅 구성 (페이지 408)
TraceDelimiter	추적 로깅 구성 (페이지 408)
TraceFile	추적 로깅 구성 (페이지 408)
TraceFileName	추적 로깅 구성 (페이지 408)
TraceFileSize	추적 로깅 구성 (페이지 408)
TraceFilesToKeep	저장되는 추적 로그 파일의 수 제한 (페이지 421)
TraceFormat	추적 로깅 구성 (페이지 408)
TrackCPSessionDomain	쿠키 공급자 재생 공격 방지 (페이지 254)

설정할 매개 변수	참조 항목
TrackSessionDomain	세션 쿠키 도메인 유효성 검사 (페이지 129)
TransientIDCookies	아이덴티티 쿠키 제어 (페이지 102)
TransientIPCheck	IP 주소를 비교하여 보안 위반 방지 (페이지 98)
UseAnonAccess	익명 사용자 액세스 사용 (페이지 332)
UseDominoUserForUnprotected	보호되지 않은 리소스를 Domino 서버에서 인증하도록 지정 (페이지 373)
UseHTTPOnlyCookies	HTTP-Only 특성을 사용하여 쿠키의 정보 보호 (페이지 101)
UseNetBIOSforIISAuth	IIS 인증을 위해 NetBIOS 이름 또는 UPN 사용 (페이지 319)
UseSecureCookies	보안 쿠키 설정 (페이지 101)
UseSecureCPCookies	여러 도메인에 보안 쿠키 설정 (페이지 260)
UseServerRequestIp	IP 주소로 에이전트 아이덴티티 확인 (페이지 97)
ValidFedTargetDomain	유효한 페더레이션 대상 도메인 정의 (페이지 183)
ValidTargetDomain	유효한 대상 도메인 정의 (페이지 84)
WebAppClientResponse	웹 응용 프로그램 클라이언트에 SiteMinder 동작 적용 (페이지 115)
XFrameOptions	사용자 지정 응답이 X-Frame Options 를 준수하도록 설정 (페이지 96)

제 24 장: SiteMinder Support Matrix(SiteMinder 지원표)

플랫폼 지원표 찾기

운영 체제 및 기타 필수 타사 구성 요소가 지원되는지 여부를 확인하려면 플랫폼 지원표를 사용하십시오.

다음 단계를 수행하십시오.

1. CA Support 사이트로 이동합니다.
2. "Product Pages"(제품 페이지)를 클릭합니다.
3. 제품 이름을 입력하고 Enter 키를 클릭합니다.
4. "Popular Links"(인기 있는 링크)를 열고 "Informational Documentation Index"(정보 설명서 색인)를 클릭합니다.
5. "Platform Support Matrices"(플랫폼 지원표)를 클릭합니다.

참고: Oracle Developer Network 에서 최신 JDK 및 JRE 버전을 다운로드할 수 있습니다.

기술 파트너 및 CA 검증된 제품

파트너 및 검증된 제품의 최신 목록입니다.