Oracle® OpenSSO 업데이트 2 릴리스 노트

Beta

부품 번호: 821–2839
2010년 7월

ORACLE

101014@24999

목차

머리말

Oracle OpenSSO 8.0 업데이트 2 릴리스 노트는 OpenSSO 업데이트 2 소프트웨어 다운로드 및 설치에 대한 정보를 제공합니다. 또한 이 문서에는 OpenSSO 업데이트 1 릴리스 이후의 소프트웨어 변경 사항에 대한 정보가 포함되어 있습니다.

대상

이 릴리스 노트는 Oracle OpenSSO 8.0을 이미 설치하고 배포한 엔터프라이즈 관리자 및 개발자를 대상으로 하며, 핵심 제품 설명서에 설명된 개념 및 절차를 잘 알고 있다는 것을 전제로 합니다.

관련 설명서

이 릴리스 노트는 http://docs.sun.com/app/docs/coll/1767.1에 있는 핵심 Oracle OpenSSO 8.0 제품 설명서를 보충합니다.

관련 타사 웹 사이트 참조

본 문서는 타사 URL을 참조하여 관련된 추가 정보를 제공합니다.

주 – Oracle은 본 설명서에 언급된 타사 웹 사이트의 가용성에 대해 책임지지 않습니다. Oracle은 해당 사이트나 자원에서 또는 이를 통해 사용할 수 있는 내용, 광고, 제품 또는 기타 자료에 대해서 보증하지 않으며 책임지지 않습니다. Oracle은 해당 사이트나 자원에서 또는 이를 통해 사용할 수 있는 내용, 제품 또는 서비스의 사용과 관련이 있어나 이로 인해 발생한 또는 발생했다고 간주되는 손해나 손실에 대해 어떠한 책임도 지지 않습니다.

설명서, 지원 및 교육

추가 자원은 다음 웹 사이트를 참조하십시오.

- 설명서 (http://docs.sun.com)
- 지원 (http://www.oracle.com/us/support/systems/index.html)
- 교육 (http://education.oracle.com) – 왼쪽 탐색 막대의 Sun 링크를 클릭하십시오.

사용자 의견 환영

Oracle은 설명서의 품질과 유용성에 대한 사용자의 의견과 제안사항을 소중하게 생각합니다. 오류를 발견하거나 개선을 위한 다른 제안사항이 있으면 http://docs.sun.com으로 이동하여 피드백을 클릭하십시오. 가능한 경우 설명서의 제목 및 부품 번호와 장, 섹션 및 페이지 번호를 명시하십시오. 회신을 원하시면 알려주시기 바랍니다.

Oracle 기술 네트워크 (http://www.oracle.com/technetwork/index.html)는 Oracle 소프트웨어와 관련된 다양한 자원을 제공합니다.

- 토론 포럼 (http://forums.oracle.com)에서 기술 문제 및 솔루션에 대해 논의합니다.
- Oracle 예제 (http://www.oracle.com/technology/obe/start/index.html)에서 간편한 단계별 자습서를 사용합니다.
- 샘플 코드 (http://www.oracle.com/technology/sample_code/index.html)를 다운로드합니다.

표기 규칙

다음 표에는 이 설명서에 사용되는 표기 규칙이 설명되어 있습니다.

표 P–1 표기 규칙

서체	의미	예
AaBbCc123	명령 이름, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용	.login 파일을 편집합니다. ls -a를 사용하여 모든 파일을 표시합니다. machine_name% you have mail.
AaBbCc123	컴퓨터 화면에 출력되는 내용과 달리 사용자가 입력하는 내용	machine_name% **su** **비밀번호:**
aabbcc123	자리 표시자: 실제 이름 또는 값으로 대체	파일을 제거하는 명령은 rm *filename*입니다.

표 P–1 표기 규칙 *(계속)*

서체	의미	예
AaBbCc123	설명서 제목, 새로운 용어 및 강조할 용어	**사용자 설명서**의 6장을 읽어 보십시오. **캐시**는 로컬로 저장되는 사본입니다. 파일을 저장해서는 **안 됩니다**. **참고:** 강조 표시된 일부 항목은 온라인에서 굵게 표시됩니다.

명령의 쉘 프롬프트 예제

다음 표는 Oracle Solaris OS에 포함된 쉘에 대한 기본 UNIX 시스템 프롬프트 및 수퍼유저 프롬프트를 나타냅니다. 명령 예에 표시된 기본 시스템 프롬프트는 Oracle Solaris 릴리스에 따라 달라집니다.

표 P–2 쉘 프롬프트

쉘	프롬프트
배시 쉘, 콘 쉘 및 본 쉘	`$`
수퍼유저용 배시 쉘, 콘 쉘 및 본 쉘	`#`
C 쉘	`machine_name%`
수퍼유저용 C 쉘	`machine_name#`

◆ ◆ ◆ 1 장

1 OpenSSO 8.0 업데이트 2 정보

이 장은 다음 내용으로 구성되어 있습니다.

OpenSSO 8.0 업데이트 2의 새로운 기능

OpenSSO 8.0 업데이트 2에는 보안 토큰 서비스 및 OpenSSO Fedlet의 개선 사항이 포함되어 있습니다.

보안 토큰 서비스 개선 사항

보안 토큰 서비스에는 다음과 같은 새로운 기능이 포함되어 있습니다.

- 특정 웹 서비스 공급자 보안 토큰을 생성하기 위한 TokenType을 지원합니다.
- X509에 대한 비대칭 및 전송 바인딩과 요청자로서의 사용자 이름 보안 토큰을 지원합니다.
- OpenSSO STS가 SSL 상에서 사용자 이름을 사용하여 구성된 경우 사용자 이름 보안 토큰으로 SSL/전송 바인딩을 적용합니다.
- useKey를 웹 서비스 클라이언트 공개 키 및 웹 서비스 클라이언트 X509 보안 토큰으로 사용하여 비대칭 KeyType에 대한 SAML Holder-of-Key 보안 토큰을 발행합니다.
- WSDL은 보안 토큰 구성에 따라 동적으로 업데이트됩니다.
- 웹 서비스 공급자 공개 키를 통한 암호화를 지원합니다.
- 구성 저장소에 저장하기 전에 정적 사용자 이름 비밀번호를 암호화합니다.
- WS-Trust 요청을 통해 UserName 토큰을 On Behalf Of 보안 토큰으로 지원합니다.

- SAML Bearer 토큰 발행을 지원합니다.
- 새 웹 서비스 보안 인증 모듈 WSSAuth는 다이제스트 비밀번호 유효성 검사를 지원합니다.
- 새 OAMAuth 인증 모듈에서는 OpenSSO가 포함된 Oracle Access Manager를 사용하여 단일 사인 온(SSO)을 사용 설정합니다.

자세한 내용은 3 장, “보안 토큰 서비스 사용”을 참조하십시오.

Fedlet 개선 사항

Fedlet에는 다음과 같은 새로운 기능이 포함되어 있습니다.

- .NET Fedlet에서의 암호화를 지원합니다.
- .NET Fedlet에서의 서명을 지원합니다.
- .NET Fedlet은 단일 로그아웃을 지원합니다.
- .NET Fedlet은 서비스 공급자가 시작한 단일 사인 온(SSO) 및 아티팩트 지원을 제공합니다.
- .NET Fedlet에서의 여러 아이디 공급자 및 아이디 공급자 검색을 지원합니다.
- Fedlet에 대한 등록 정보 및 구성 파일 내의 버전 정보를 제공합니다.
- 새 비밀번호 SPI를 구현합니다.
- 속성 쿼리를 지원합니다.
- 단일 로그아웃을 지원합니다.

자세한 내용은 4 장, “Oracle OpenSSO Fedlet 사용”을 참조하십시오.

OpenSSO 8.0 업데이트 2의 하드웨어 및 소프트웨어 요구 사항

Sun OpenSSO Enterprise 8.0 Update 1 Release Notes의 “Hardware and Software Requirements For OpenSSO Enterprise 8.0 Update 1”을 참조하십시오.

새 웹 컨테이너 지원

OpenSSO 8.0 업데이트 2는 **Sun OpenSSO Enterprise 8.0 Update 1 Release Notes**의 “Support for New Web Containers”에 설명된 웹 컨테이너와 다음과 같은 새 웹 컨테이너를 지원합니다.

- Oracle WebLogic Server 10g 릴리스 3(10.3)

OpenSSO 8.0 업데이트 2 문제 및 해결 방법

CR 6959610: OpenSSO 8.0 업데이트 2 샘플이 생성 환경에서 제거되어야 합니다.

OpenSSO 8.0 업데이트 2 샘플이 잠재적인 보안 문제를 일으킬 수 있습니다.

해결 방법 생성 환경에 OpenSSO 8.0 업데이트 2를 배포하는 경우 샘플을 제거하여 잠재적인 보안 문제가 발생하지 않도록 합니다.

CR 6964648: WebLogic Server 10.3.3의 경우 새 Java 보안 권한이 필요합니다.

보안 관리자를 사용하는 Oracle WebLogic Server 10.3.3에 OpenSSO 8.0 업데이트 2를 배포하는 경우 추가 Java 보안 권한이 필요합니다.

해결 방법 WebLogic Server 10.3.3 `weblogic.policy` 파일에 다음 권한을 추가합니다.

```
permission java.lang.RuntimePermission "getClassLoader";
```

CR 6939443: LDAP 확인 또는 OCSP 확인을 통한 인증서 인증을 WebLogic Server 10.3.x에서 수행할 수 없습니다.

10.3.0 및 10.3.1과 같은 Oracle WebLogic Server의 이전 버전에서 발생하는 문제로 인해 LDAP 확인 또는 OSCP 확인을 사용하는 인증서 인증을 수행할 수 없습니다.

해결 방법 이 문제는 WebLogic Server 10.3.3에서 해결되었습니다. LDAP 확인 또는 OSCP 확인을 통한 인증서 인증을 사용하려면 WebLogic Server 10.3.3에서 OpenSSO 업데이트 2를 사용하십시오.

CR 6967026: 구성자를 GlassFish 2.1.x의 LDAPS 사용 디렉토리 서버 인스턴스에 연결할 수 없습니다.

GlassFish Enterprise Server v2.1.1 또는 v2.1.2가 OpenSSO 8.0 업데이트 2 웹 컨테이너로 배포된 경우 구성자는 LDAPS 사용 디렉토리 서버 인스턴스에 연결할 수 없습니다.

해결 방법 GlassFish가 포함된 LDAPS 사용 디렉토리 서버를 웹 컨테이너로 사용하려면 GlassFish Enterprise Server v2.1을 배포합니다.

CR 6948937: WebLogic Server 10.3.3 관리 콘솔에서 OpenSSO 8.0 업데이트 2를 활성화하면 예외가 발생합니다.

WebLogic Server 10.3.3 관리 콘솔에서 OpenSSO 8.0 업데이트 2(`opensso.war`)를 배포하는 경우 시작을 클릭하여 OpenSSO 8.0 업데이트 2가 요청 수신을 시작할 수 있도록 하면 WebLogic Server 도메인이 시작된 콘솔에서 예외가 발생합니다.

참고: OpenSSO 8.0 업데이트 2가 시작되면 시작됨 상태로 유지되며 OpenSSO 8.0 업데이트 2를 중지하고 다시 시작할 때까지 예외가 다시 발생하지 않습니다.

해결 방법 다음과 같이 OpenSSO 8 업데이트 2 `opensso-client-jdk15.war` 파일의 `saaj-impl.jar` 파일을 WebLogic Server 10.3.3 구성의 `endorsed` 디렉토리로 복사합니다.

1. Oracle WebLogic Server 10.3.3 도메인을 중지합니다.
2. 필요한 경우, OpenSSO 8.0 Update 2 `opensso.zip` 파일의 압축을 해제합니다.
3. 임시 디렉토리를 만들고 *zip-root*`/opensso/samples/opensso-client.zip` 파일의 압축을 해제합니다. *zip-root*는 `opensso.zip` 파일의 압축을 해제한 위치입니다. 예:

   ```
   cd zip-root/opensso/samples
   mkdir ziptmp
   cd ziptmp
   unzip ../opensso-client.zip
   ```

4. 임시 디렉토리를 만들고 `opensso-client-jdk15.war`에서 `saaj-impl.jar` 파일을 추출합니다. 예:

   ```
   cd zip-root/opensso/samples/ziptmp/war
   mkdir wartmp
   cd wartmp
   jar xvf ../opensso-client-jdk15.war WEB-INF/lib/saaj-impl.jar
   ```

5. WEBLOGIC_JAVA_HOME/jre/lib 디렉토리 아래에 endorsed라는 새 디렉토리를 만듭니다(endorsed가 아직 없는 경우). WEBLOGIC_JAVA_HOME은 WebLogic Server에서 사용하도록 구성된 JDK입니다.
6. saaj-impl.jar 파일을 WEBLOGIC_JAVA_HOME/jre/lib/endorsed 디렉토리로 복사합니다.
7. WebLogic Server 도메인을 시작합니다.

CR 6959373: updateschema 스크립트를 실행한 후 웹 컨테이너를 다시 시작해야 합니다.

updateschema.sh 또는 updateschema.bat 스크립트를 실행한 후 OpenSSO 8.0 업데이트 2 웹 컨테이너를 다시 시작해야 합니다.

CR 6961419: updateschema.bat 스크립트를 실행하려면 비밀번호 파일이 필요합니다.

updateschema.bat 스크립트는 여러 ssoadm 명령을 실행합니다. 따라서 Windows 시스템에서 updateschema.bat를 실행하기 전에 일반 텍스트로 amadmin 사용자의 비밀번호 사용자를 포함하는 비밀번호 파일을 만듭니다. updateschema.bat 스크립트를 실행하면 비밀번호 파일의 경로를 입력하라는 메시지가 표시됩니다. 스크립트가 종료되기 전에 비밀번호 파일이 제거됩니다.

OpenSSO 8.0 업데이트 2 설명서

이 문서뿐 아니라 다음 모음에 제공되는 추가 OpenSSO 8.0 설명서를 사용할 수 있습니다.

http://docs.sun.com/coll/1767.1

설명서 문제

OpenSSO 8.0 업데이트 2에는 다음과 같은 설명서 문제가 포함되어 있습니다.

- 16 페이지 “CR 6958580: 콘솔 온라인 도움말은 지원되지 않는 검색 에이전트에 대해 설명합니다.”
- 16 페이지 “CR 6967006 콘솔 온라인 도움말은 OAMAuth 및 WSSAuth 인증 모듈에 대한 설명을 제공하지 않습니다.”
- 16 페이지 “CR 6953582: Fedlet Java API 참조는 공용이어야 합니다.”
- 16 페이지 “CR 6953579: OpenSSO Fedlet README 파일에는 단일 로그아웃 기능에 대한 설명이 제공되어야 합니다.”

CR 6958580: 콘솔 온라인 도움말은 지원되지 않는 검색 에이전트에 대해 설명합니다.

OpenSSO 8.0 업데이트 2 관리 콘솔 온라인 도움말에는 검색 에이전트에 대한 설명이 포함되어 있지만 해당 에이전트는 지원되지 않습니다.

해결 방법 없음. 온라인 도움말의 검색 에이전트에 대한 정보를 무시합니다.

CR 6967006 콘솔 온라인 도움말은 OAMAuth 및 WSSAuth 인증 모듈에 대한 설명을 제공하지 않습니다.

OpenSSO 8.0 업데이트 1 관리 콘솔 온라인 도움말에는 OAM(Oracle Access Manager) 및 WSS(Web Services Security) 인증 모듈에 대한 설명이 포함되어 있지 않습니다.

해결 방법 이러한 인증 모듈에 대한 자세한 내용은 3 장, “보안 토큰 서비스 사용”을 참조하십시오.

CR 6953582: Fedlet Java API 참조는 공용이어야 합니다.

Fedlet Java API 공용 참조는 Oracle OpenSSO 8.0 업데이트 2 Java API 참조에서 제공되며 `http://docs.sun.com/coll/1767.1`의 문서 모음에서 찾아볼 수 있습니다.

참고: `getPolicyDecisionForFedlet` 메소드가 Java API 참조에 들어 있더라도 OpenSSO 8.0 업데이트 2는 이 메소드를 지원하지 않습니다.

CR 6953579: OpenSSO Fedlet README 파일에는 단일 로그아웃 기능에 대한 설명이 제공되어야 합니다.

Fedlet README 파일에는 단일 로그아웃 기능에 대한 설명이 포함되어 있지 않습니다.

해결 방법 Oracle OpenSSO 8.0 업데이트 2의 경우 Fedlet 단일 로그아웃 기능은 4 장, “Oracle OpenSSO Fedlet 사용”에 설명되어 있습니다.

추가 정보 및 자원

다음 위치에서도 유용한 추가 정보 및 자원을 찾을 수 있습니다.

- 17 페이지 “서비스 중단 알림 및 발표”
- 17 페이지 “문제점 보고 및 피드백 제공 방법”
- 18 페이지 “장애인을 위한 내게 필요한 옵션 기능”
- 18 페이지 “관련 타사 웹 사이트”
- 시스템에 대한 Oracle 고급 고객 서비스:

  `http://www.oracle.com/us/support/systems/advanced-customer-services/index.html`

- 소프트웨어 제품: http://www.oracle.com/us/sun/sun-products-map-075562.html
- SunSolve: http://sunsolve.sun.com/
- Sun SDN(개발자 네트워크): http://developers.sun.com/
- Sun 개발자 서비스:http://developers.sun.com/services/

서비스 중단 알림 및 발표

- SMS(서비스 관리 서비스) API(com.sun.identity.sm 패키지) 및 SMS 모델은 향후 OpenSSO 릴리스에 포함되지 않습니다.
- Unix 인증 모듈 및 Unix 인증 도우미(amunixd)는 향후 OpenSSO 릴리스에 포함되지 않습니다.
- Sun Java System Access Manager 7.1 릴리스 노트에 일반적으로 AMSDK(Access Manager SDK)라고 알려진 Access Manager com.iplanet.am.sdk 패키지와 모든 관련 API 및 XML 서식 파일이 향후 OpenSSO 릴리스에 포함되지 않는다는 것이 명시되어 있습니다.

  따라서 AMSDK가 제거되면 레거시 모드 옵션 및 지원도 제거됩니다.

  이제 마이그레이션 옵션을 사용할 수 없으며 앞으로도 지원되지 않을 예정입니다. Oracle Identity Manager는 AMSDK 대신 사용할 수 있는 사용자 프로비저닝 솔루션을 제공합니다. Identity Manager에 대한 자세한 내용은 http://www.oracle.com/products/middleware/identity-management/identity-manager.html을 참조하십시오.

문제점 보고 및 피드백 제공 방법

OpenSSO 8.0 업데이트 2 또는 후속 패치 릴리스에 대한 궁금한 점이나 문제가 있으면 지원 담당자(http://sunsolve.sun.com/)에게 문의하십시오.

이 사이트에서는 기술 자료, 온라인 지원 센터 및 Product Tracker뿐 아니라 유지 보수 프로그램 및 지원 담당자 연락처에 대한 링크가 제공됩니다. 문제에 대한 도움을 요청하는 경우 다음 정보를 제공해야 합니다.

- 문제 설명(문제가 발생한 시기와 문제가 작업에 미치는 영향 포함)
- 시스템 유형, 운영 체제 버전, 웹 컨테이너 및 버전, JDK 버전 및 OpenSSO 버전(문제에 영향을 미쳤을 수도 있는 패치 또는 기타 소프트웨어 포함)
- 문제를 재현하는 단계
- 오류 로그 또는 코어 덤프

장애인을 위한 내게 필요한 옵션 기능

이 매체를 게시한 후 출시된 내게 필요한 옵션을 얻으려면 요청 시 얻을 수 있는 섹션 508 제품 평가를 참조하여 관련 솔루션 배포에 가장 적합한 버전을 확인하십시오.

내게 필요한 옵션 구현을 위한 Oracle의 방침에 대한 자세한 내용은 http://www.oracle.com/index.html을 참조하십시오.

관련 타사 웹 사이트

본 문서는 타사 URL을 참조하여 관련된 추가 정보를 제공합니다.

주 – Oracle은 본 설명서에 언급된 타사 웹 사이트의 가용성에 대해 책임지지 않습니다. Oracle은 해당 사이트나 자원에서 또는 이를 통해 사용할 수 있는 내용, 광고, 제품 또는 기타 자료에 대해서 보증하지 않으며 책임지지 않습니다. Oracle은 해당 사이트나 자원에서 또는 이를 통해 사용할 수 있는 내용, 제품 또는 서비스의 사용과 관련이 있거나 이로 인해 발생한 또는 발생했다고 간주되는 손해나 손실에 대해 어떠한 책임도 지지 않습니다.

◆ ◆ ◆ 2 장

2 OpenSSO 8.0 업데이트 2 설치

이 장은 다음 내용으로 구성되어 있습니다.

OpenSSO 8.0 업데이트 2 설치 개요

OpenSSO 8.0 업데이트 2는 패치 **TBS**로 사용할 수 있습니다.

OpenSSO 8.0 업데이트 2(또는 후속 패치)를 설치하기 전에 본 문서에서 새로운 기능, 하드웨어 및 소프트웨어 요구 사항과 문제 및 문제 해결에 대한 정보를 확인하십시오.

OpenSSO 8.0 업데이트 2에는 다음 방법을 사용하여 설치할 수 있는 `opensso.war` 파일이 포함되어 있습니다.

- **기존 OpenSSO 8.0 배포 패치** : 이 장에 설명된 대로 업데이트 2의 `ssopatch` 유틸리티를 사용하여 기존 OpenSSO 8.0 배포를 패치합니다.

  참고 - Oracle은 OpenSSO 8.0 릴리스 패치만 지원합니다. 예를 들어 OpenSSO 8.0 업데이트 2와 함께 OpenSSO 8.0 패치가 지원됩니다.

- **새 OpenSSO 8.0 업데이트 2 배포 설치:** **Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide**에 설명된 대로 OpenSSO 8.0 업데이트 2 `opensso.war` 파일을 설치 및 구성합니다.
- **새로운 특수 WAR 파일 만들기:** `createwar` 스크립트를 사용하여 업데이트 2 `opensso.war` 파일에서 다음과 같은 새 WAR 파일 중 하나를 만듭니다.
  - OpenSSO 관리 콘솔 전용 WAR
  - Distributed Authentication UI 서버 WAR
  - OpenSSO 서버 전용 WAR(관리 콘솔 없음)
  - IDP 검색 서비스 WAR

  자세한 내용은 **Sun OpenSSO Enterprise 8.0 Update 1 Release Notes**의 4 장, "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File"를 참조하십시오.
- **기존의 특수 OpenSSO WAR 파일 패치:** **Sun OpenSSO Enterprise 8.0 Installation and Configuration Guide**의 23 장, "Patching OpenSSO Enterprise 8.0"에 설명된 대로 업데이트 2의 `ssopatch` 유틸리티를 사용하여 기존의 특수 OpenSSO 8.0 WAR 파일을 패치합니다.

주 – Access Manager 7.1 또는 Access Manager 7 2005Q4를 실행 중이며 업데이트 2로 업그레이드하려는 경우 다음 단계를 따릅니다.

1. **Sun OpenSSO Enterprise 8.0 Upgrade Guide**에 설명된 대로 Access Manager 7.x를 OpenSSO 8.0으로 업그레이드합니다.
2. 이 장에 설명된 대로 업그레이드 2 패치를 적용합니다.

OpenSSO 8.0 업데이트 2 패치

Sun은 OpenSSO 8.0 업데이트 2용 패치를 주기적으로 출시합니다. 이러한 패치에 대한 자세한 내용은 여기에서 주기적으로 확인하십시오.

패치 작업 계획

▼ OpenSSO 8.0의 패치 작업을 계획하는 방법

1 **21 페이지 "ssopatch 유틸리티 개요"를 읽습니다.**

2 **22 페이지 "ssopatch 유틸리티 설치"에 설명된 대로 해당 플랫폼에서 사용할 패치 유틸리티를 설치합니다.**

3 **기존 WAR 파일에 대한 정보를 가져와서 기존 WAR 파일이 24 페이지 "OpenSSO WAR 파일과 내부 매니페스트 비교"에 설명된 대로 사용자 정의되거나 수정되었는지 확인합니다.**

4 **24 페이지 "두 OpenSSO WAR 파일 비교"에 설명된 대로 기존 WAR 파일과 업데이트 2 WAR 파일을 비교하여 원본 WAR 파일에서 사용자 정의한 파일, 새 WAR 파일에서 업데이트된 파일 및 두 WAR 버전 간에 추가되거나 삭제된 파일을 반환합니다.**

5 **23 페이지 "OpenSSO WAR 파일 백업"에 설명된 대로 기존 Opensso WAR 파일을 백업 및 아카이브합니다.**

6 **25 페이지 "OpenSSO WAR 파일 패치"에 설명된 대로 OpenSSO WAR 파일을 패치합니다.**

7 **28 페이지 "updateschema 스크립트 실행"에 설명된 대로 updateschema 스크립트를 실행합니다.**

참고 - OpenSSO 서버 전용, 관리 콘솔 전용, Distributed Authentication UI 서버 또는 IDP 검색 서비스 WAR과 같이 opensso.war에서 생성한 특수 WAR 파일을 패치하는 경우 28 페이지 "특수 OpenSSO WAR 패치"를 참조하십시오.

ssopatch 유틸리티 개요

ssopatch 유틸리티는 Solaris 및 Linux 시스템에서 ssopatch로, Windows에서는 ssopatch.bat로 사용할 수 있는 Java 명령줄 유틸리티입니다.

참고 - OpenSSO 8.0 업데이트 2의 ssopatch에 대한 구문은 OpenSSO 8.0 릴리스 이후 크게 변경되었습니다. 새 구문에 대한 내용은 28 페이지 "updateschema 스크립트 실행"을 참조하십시오.

ssopatch 패치 유틸리티는 다음 기능을 수행합니다.

- OpenSSO WAR을 원본 매니페스트와 비교하여 WAR 파일이 사용자 정의 또는 수정되었는지 확인합니다.
- 두 OpenSSO WAR 파일을 비교하여 원본 WAR 파일에 대한 사용자 정의 내용 및 새 WAR 파일의 변경 내용 등, 두 파일의 차이점을 확인합니다.
- 새로 패치된 OpenSSO WAR 파일을 생성하는 데 필요한 파일의 스테이징 영역을 생성합니다.

OpenSSO 8.0 업데이트 2 ZIP 파일(opensso_80U2.zip)을 다운로드 및 압축 해제한 후, *zip-root*/opensso/tools 디렉토리의 ssoPatchTools.zip 파일에서 패치 유틸리티 및 관련 파일을 사용할 수 있습니다. *zip-root*는 opensso_80U2.zip을 압축 해제한 위치입니다.

ssopatch 유틸리티는 매니페스트 파일을 사용하여 특정 OpenSSO WAR 파일의 내용을 확인합니다. 매니페스트 파일은 다음을 포함하는 ASCII 텍스트 파일입니다.

- OpenSSO WAR 파일의 특정 버전을 식별하는 문자열

- OpenSSO WAR 파일의 모든 개별 파일과 각 파일에 대한 체크섬 정보

일반적으로 매니페스트 파일은 OpenSSO.manifest로 이름이 지정되고 OpenSSO WAR 파일의 META-INF 디렉토리에 저장됩니다.

ssopatch 유틸리티는 표준 출력(stdout)으로 결과를 보냅니다. 원하는 경우 파일로 출력을 리디렉션하여 ssopatch 출력을 캡처할 수 있습니다. ssopatch가 성공적으로 완료되면 영(0) 종료 코드를 반환합니다. 오류가 발생하는 경우 ssopatch는 영이 아닌 종료 코드를 반환합니다.

ssopatch 유틸리티 설치

ssopatch 유틸리티를 설치하기 전에 다음을 수행합니다.

- OpenSSO 8.0 업데이트 2 ZIP 파일(opensso_80U2.zip)을 다운로드 및 압축 해제합니다.
- JAVA_HOME 환경 변수가 JDK 1.5 이상을 가리키도록 설정합니다.

ssopatch 유틸리티를 설치하는 방법

1. *zip-root*/opensso/tools 디렉토리에서 ssoPatchTools.zip 파일을 찾습니다. *zip-root*는 opensso_80U2.zip을 압축 해제한 위치입니다.
2. ssoPatchTools.zip 파일을 압축 해제할 새 디렉토리를 만듭니다. 예: ssopatchtools
3. 새 디렉토리에서 ssoPatchTools.zip 파일을 압축 해제합니다.
4. 전체 경로를 제공하지 않고 현재 디렉토리가 아닌 다른 디렉토리에서 ssopatch 유틸리티를 실행하려면 PATH 변수에 유틸리티를 추가합니다.

다음 표는 ssoPatchTools.zip에 들어 있는 파일에 대해 설명합니다.

파일 또는 디렉토리	설명
README	ssopatch를 설명하는 Readme 파일
/lib	필수 ssopatch JAR 파일
/patch	updateschema 및 updateschema.bat 스크립트 및 관련 XML 파일
/resources	필수 등록 정보 파일
ssopatch 및 ssopatch.bat	Solaris, Linux 및 Windows 시스템용 유틸리티

OpenSSO WAR 파일 백업

시작하기 전에 기존 OpenSSO WAR 파일 및 구성 데이터를 백업합니다.

- 기존 OpenSSO WAR 파일을 안전한 위치에 복사합니다. 이렇게 하면 어떤 이유로든 업데이트 2를 되돌려야 하는 경우 WAR 파일의 백업 복사본을 재배포할 수 있습니다.
- **Sun OpenSSO Enterprise 8.0 Administration Guide**의 15 장, "Backing Up and Restoring Configuration Data"에 설명된 대로 구성 데이터를 백업합니다.

ssopatch 유틸리티 실행

ssopatch 유틸리티를 실행하려면 다음 사용법을 따릅니다.

```
ssopatch
--help|-?
[--locale|-l]

ssopatch
--war-file|-o
[--manifest|-m]
[--locale|-l]

ssopatch
--war-file|-o
--war-file-compare|-c
[--staging|-s]
[--locale|-l]
[--override|-r]
[--overwrite|-w]
```

옵션은 다음과 같습니다.

- `-war-file|-o`는 이전에 배포된 WAR 파일(예: `opensso.war`)의 경로를 지정합니다.
- `-manifest|-m`은 만들려는 매니페스트 파일의 경로를 지정합니다. 이 옵션이 제공된 경우 `-war-file|-o` 옵션이 표시하는 WAR 파일에서 매니페스트 파일이 생성됩니다.
- `-war-file-compare|-c`는 `-war-file|-o`가 표시하는 WAR 파일과 비교할 WAR 파일의 경로를 지정합니다.
- `-staging|-s`는 OpenSSO WAR의 파일을 작성할 스테이징 영역의 경로를 지정합니다.
- `-locale|-l`은 사용할 로켈을 지정합니다. 이 옵션을 지정하지 않으면 `ssopatch`는 기본 시스템 로켈을 사용합니다.
- `-override|-r`은 두 WAR 파일에 대한 개정 확인을 무시합니다. 개정 확인은 WAR 파일의 버전을 확인하고 버전이 호환 가능한 경우에만 계속 수행됩니다. 이 옵션을 사용하여 이 검사를 무시할 수 있습니다.

기본값은 false입니다(개정 확인 수행).

- `-overwrite|-w`는 기존 스테이징 영역의 파일을 덮어씁니다. 기본값은 false입니다(파일을 덮어쓰지 않음).

OpenSSO WAR 파일과 내부 매니페스트 비교

이 절차를 사용하여 다운로드한 이후 OpenSSO WAR 파일이 사용자 정의 또는 수정되었는지 확인합니다.

`ssopatch` 유틸리티는 새로운 내부 매니페스트 파일을 생성한 다음 이 내부 매니페스트를 `META-INF` 디렉토리의 원본 OpenSSO WAR 파일 내에 저장된 매니페스트와 비교합니다.

OpenSSO WAR 파일과 내부 매니페스트를 비교하는 방법

1. `ssopatch`를 실행하여 OpenSSO WAR 파일과 내부 매니페스트를 비교합니다. 예:

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Comparing manifest of Internal (Enterprise 8.0 Build 6(200810311055))
against /zip-root/opensso/deployable-war/opensso.war (generated-200905050855)
File not in original war (images/login-origimage.jpg)
File updated in new war (images/login-backimage.jpg)
File updated in new war (WEB-INF/classes/amConfigurator.properties)
Differences: 3
```

이 예는 원본 WAR 파일에 대한 변경 사항을 보여줍니다.

- `images/login-origimage.jpg`는 `opensso.war` 에는 있지만 원본 매니페스트에는 없습니다.
- `images/login-backimage.jpg`가 원본 매니페스트의 `opensso.war`에서 사용자 정의되었습니다.
- `WEB-INF/classes/amConfigurator.properties` 파일이 원본 매니페스트의 `opensso.war`에서 사용자 정의되었습니다.

두 OpenSSO WAR 파일 비교

이 절차를 통해 두 WAR 파일을 비교하여 다음과 같은 상태의 파일을 보여줍니다.

- 원본 OpenSSO WAR에서 사용자 정의됨
- 새 OpenSSO WAR 파일에서 업데이트됨
- 두 OpenSSO WAR 버전 간에 추가 또는 삭제됨

두 OpenSSO WAR 파일을 비교하는 방법

1. ssopatch를 실행하여 두 WAR 파일을 비교합니다. 예에서는 -override 옵션을 사용하여 두 WAR 파일 간의 개정 확인을 무시합니다.

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
-c /u1/opensso/deployable-war/opensso.war --override
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Original manifest: Enterprise 8.0 Build 6(200810311055)
New manifest: Enterprise 8.0 Update 2 Build 6.1(200904300525)
Versions are compatible
Generating Manifest for: /u1/opensso/deployable-war/opensso.war
Comparing manifest of /zip-root/opensso/deployable-war/opensso.war
(generated-200905050919) against
    /u1/opensso/deployable-war/opensso.war (generated-200905050920)
File updated in new war(WEB-INF/classes/amClientDetection_en.properties)
File updated in new war(WEB-INF/classes/fmSAMLConfiguration_fr.properties)
...
Differences: 1821
Customizations: 3
```

이 예는 새 WAR 파일에서 업데이트 및 사용자 정의된 파일을 표시합니다.

OpenSSO WAR 파일 패치

이 절차를 사용하여 원본 WAR 파일이 새 WAR 파일과 병합되는 새 스테이징 영역을 만듭니다.

이 작업은 각 WAR 파일에 대한 매니페스트를 비교한 후 다음을 보여줍니다.

- 원본 WAR 파일에서 사용자 정의된 파일
- 새 WAR 파일에서 업데이트된 파일
- 두 WAR 파일 버전 간에 추가되거나 제거된 파일

그런 다음 ssopatch는 해당 파일을 스테이징 디렉토리(새로 패치된 WAR을 만들고 배포하기 전에 사용자 정의를 추가해야 하는 위치)로 복사합니다.

OpenSSO WAR 파일을 패치할 스테이징 영역을 만드는 방법

1. ssopatch가 원본 opensso.war 파일을 수정하지 않더라도 패치된 opensso.war 파일을 되돌려야 하는 경우에 대비하여 이 파일을 백업하는 것이 좋습니다.
2. ssopatch를 실행하여 스테이징 영역을 만듭니다. 예:

```
./ssopatch -o /zip-root/opensso/deployable-war/opensso.war
  -c /u1/opensso/deployable-war/opensso.war --override -s /tmp/staging
Generating Manifest for: /zip-root/opensso/deployable-war/opensso.war
Original manifest: Enterprise 8.0 Build 6(200810311055)
```

```
New manifest: Enterprise 8.0 Update 2 Build 6.1(200904300525)
Versions are compatible
Generating Manifest for: /u1/opensso/deployable-war/opensso.war
Comparing manifest of /zip-root/opensso/deployable-war/opensso.war
    (generated-200905051031) against /u1/opensso/deployable-war/opensso.war
    (generated-200905051032)
File was customized in original, but not found in new war.
Staging area using original war version (samples/saml2/sae/header.jsp)
File was customized in original, but not found in new war.
Staging area using original war version
    (WEB-INF/template/opends/config/upgrade/config.ldif.4517)
File was customized in original, but not found in new war.
Staging area using original war version
    (WEB-INF/template/opends/config/upgrade/schema.ldif.4517)
Differences: 1813
Customizations: 0
```

이 예에서 `/tmp/staging`은 `ssopatch`가 파일을 복사하는 스테이징 영역입니다.

이전 단계의 결과를 사용하여 필요에 따라 스테이징 영역에서 파일을 업데이트합니다.

다음 표를 사용하여 새로 패치된 WAR 파일을 생성하기 전에 수행해야 할 작업을 확인합니다.

ssopatch 결과	**설명 및 필요한 작업**
파일이 원본 `war` *filename*에 **없습니다**.	표시된 파일이 원본 WAR 파일에는 없지만 WAR 파일의 최신 버전에는 있습니다. **작업**: 없음
파일이 새 `war` *filename*에서 **업데이트되었습니다**.	표시된 파일이 원본과 새 WAR 파일 모두에 존재하고 WAR 파일의 최신 버전에서 업데이트되었습니다. 원본 WAR 파일에서 사용자 정의가 수행되지 않았습니다. **작업**: 없음
파일이 *filename*을 **사용자 정의했습니다**.	표시된 파일이 두 WAR 파일 모두에 있고 WAR 파일의 원본 버전에서 사용자 정의되었지만 WAR 파일의 최신 버전에서 업데이트되지는 않았습니다. **작업**: 없음
*filename*을 **직접 사용자 정의해야 할 수도 있습니다**.	파일이 두 WAR 파일 모두에 있고 WAR 파일의 원본 버전에서 사용자 정의되었으며 WAR 파일의 최신 버전에서 업데이트되었습니다. **작업**: 파일에서 사용자 정의를 수행하려면 스테이징 디렉토리에서 새로 업데이트된 파일에 이를 직접 추가해야 합니다.
원본 버전에서 파일이 사용자 정의되었지만 새 `war`**에 없습니다**.	파일이 원본 WAR 파일에는 있었지만 새 WAR에는 없습니다. **작업**: 없음

다음 단계

1. 스테이징 영역의 파일에서 새 OpenSSO WAR 파일을 만듭니다. 예:

   ```
   cd /tmp/staging
   jar cvf /patched/opensso.war *
   ```

   `/patched/opensso.war`은 새로 패치된 OpenSSO WAR 파일입니다.
2. 원본 배포 URI를 사용하여 `/patched/opensso.war` 파일을 웹 컨테이너에 재배포합니다. 예: `/opensso`

OpenSSO 구성 변경 사항. 새 OpenSSO WAR 파일에 원본 WAR 파일에 없는 구성 변경 사항이 있을 수도 있습니다. 각 패치에 대한 구성 변경 사항이 있는 경우 개별적으로 기록됩니다. 구성 변경 사항에 대한 자세한 내용은 패치 설명서와 **Sun OpenSSO Enterprise 8.0 릴리스 노트**를 확인하십시오. 새 WAR 파일의 구성이 변경되지 않았더라도 OpenSSO 매니페스트 파일의 버전 문자열은 변경됩니다.

패치된 버전을 되돌려야 하는 경우 패치된 WAR 파일을 배포 해제한 다음 원본 WAR 파일을 재배포합니다.

OpenSSO WAR 매니페스트 파일 만들기

OpenSSO 매니페스트 파일은 특정 릴리스에 대한 WAR 파일의 모든 개별 파일을 각 파일에 대한 체크섬 정보로 식별하는 텍스트 파일입니다.

이 절차를 통해 OpenSSO 서버 전용, 관리 콘솔 전용, Distributed Authentication UI 서버 또는 IDP 검색 서비스 WAR과 같은 특수 OpenSSO WAR에 포함될 수 있는 매니페스트 파일을 만듭니다.

OpenSSO WAR 매니페스트 파일을 만드는 방법

1. `ssopatch`를 실행하여 OpenSSO 매니페스트 파일을 만듭니다. 예:

   ```
   ./ssopatch -o zip-root/opensso/deployable-war/opensso.war --manifest /tmp/manifest
   ```

   `opensso.war`은 기존 OpenSSO WAR 파일입니다.

   `ssopatch` 유틸리티는 `/tmp` 디렉토리에 `manifest`라는 새 매니페스트 파일을 만듭니다.
2. WAR 파일이 패치되도록 하려면 이 새 매니페스트 파일을 `opensso.war` 파일 내의 `META-INF` 디렉토리로 복사합니다. 예:

   ```
   mkdir META-INF
   cp /tmp/manifest META-INF
   jar uf opensso.war META-INF/manifest
   ```

특수 OpenSSO WAR 패치

OpenSSO 서버 전용, 관리 콘솔 전용, Distributed Authentication UI 서버 또는 IDP 검색 서비스 WAR과 같은 특수 OpenSSO WAR을 이전에 만든 경우 `ssopatch` 유틸리티를 사용하여 이를 패치할 수 있습니다.

특수 OpenSSO WAR을 패치하는 방법

1. 27 페이지 "OpenSSO WAR 매니페스트 파일 만들기"에 설명된 대로 특수 OpenSSO WAR에 대한 매니페스트 파일을 만듭니다.

   참고: 모든 사용자 정의가 수행되기 전에 Sun에서 제공된 대로 원본 OpenSSO 8.0 `opensso.war`을 기반으로 한 매니페스트 파일을 만듭니다. 매니페스트가 사용자 정의 후에 만들어진 경우 `ssopatch`는 사용자 정의가 아닌 업데이트 2의 파일을 사용할 수 있으므로 패치 후 사용자 정의를 다시 실행해야 합니다.
2. **Sun OpenSSO Enterprise 8.0 Update 1 Release Notes**의 4 장, "Creating a Specialized OpenSSO Enterprise 8.0 Update 1 WAR File"에 설명된 대로 OpenSSO 8.0 업데이트 2 `opensso.war` 파일에서 특수 OpenSSO WAR을 생성합니다.
3. `ssopatch` 유틸리티를 사용하여 이전 WAR 파일과 새 WAR 파일을 비교합니다.
4. 25 페이지 "OpenSSO WAR 파일을 패치할 스테이징 영역을 만드는 방법"에 설명된 대로 새 특수 WAR 파일에 대한 스테이징 영역을 생성합니다.
5. 새 특수 WAR 파일을 재배포합니다.

`updateschema` 스크립트 실행

`ssopatch`를 실행한 후, Solaris 또는 Linux 시스템에서는 `updateschema.sh` 또는 Windows에서는 `updateschema.bat`를 실행합니다. 스크립트는 OpenSSO 서버 버전을 업데이트하고 새 기본 서버 등록 정보를 추가하며 업데이트 2의 버그 수정 및 개선 사항에 필요한 새 속성 스키마를 추가합니다. 서버 버전을 업데이트하기 위해 `updateschema`를 실행해야 합니다.

시작하기 전에

- `updateschema.sh` 또는 `updateschema.bat` 스크립트에는 `ssoadm` 명령줄 유틸리티의 업데이트 2 버전(이상)이 필요합니다. 따라서 이 스크립트를 실행하기 전에 **Sun OpenSSO Enterprise 8.0 Update 1 Release Notes**의 3 장, "Installing the OpenSSO Enterprise 8.0 Update 1 Admin Tools"에 설명된 대로 업데이트 2 관리 도구를 설치하십시오.

- updateschema.bat 스크립트는 여러 개의 ssoadm 명령을 실행합니다. 따라서 Windows 시스템에서 updateschema.bat를 실행하기 전에 일반 텍스트로 amadmin 사용자의 비밀번호 사용자를 포함하는 비밀번호 파일을 만듭니다. updateschema.bat 스크립트를 실행하면 비밀번호 파일의 경로를 입력하라는 메시지가 표시됩니다. 스크립트가 종료되기 전에 비밀번호 파일이 제거됩니다.

updateschema 스크립트를 실행하는 방법

1. *patch-tools*/patch 디렉토리로 변경합니다. *patch-tools*는 ssoPatchTools.zip을 압축 해제한 위치입니다.
2. updateschema.sh 또는 updateschema.bat를 실행합니다. 예를 들어 Solaris 시스템의 경우 다음을 수행합니다.

   ```
   ./updateschema.sh
   ```

3. 스크립트가 메시지를 표시하면 다음 정보를 입력합니다.
   - ssoadm 유틸리티의 전체 경로(ssoadm 자체는 제외) 예: /opt/ssotools/opensso/bin
   - amadmin 비밀번호

   updateschema.sh 또는 updateschema.bat 스크립트가 메시지 또는 오류를 표준 출력으로 작성합니다.
4. OpenSSO 8.0 업데이트 2 웹 컨테이너를 다시 시작합니다.

패치 설치 되돌리기

패치 설치를 되돌려야 할 경우 원본 opensso.war 파일(또는 특수 WAR 파일)을 재배포하면 됩니다.

◆ ◆ ◆ 3 장

3 보안 토큰 서비스 사용

OpenSSO 보안 토큰 서비스는 신뢰할 수 있는 인증 서비스로서 보안 토큰을 발행 및 유효성 검사합니다. 보안 토큰 서비스는 웹 서비스 보안 공급자로서 웹 서비스 클라이언트와 OpenSSO STS 서비스 자체 간의 통신을 보안합니다. OpenSSO 8.0 업데이트 2 이후 보안 토큰 서비스의 여러 가지 기능이 개선되었습니다.

이 장은 다음 내용으로 구성되어 있습니다.

WSSAuth 인증 모듈 추가

웹 서비스 보안 인증 모듈을 사용하면 OpenSSO는 인증 토큰으로서 수신되고 웹 서비스 클라이언트에서 웹 서비스 공급자로의 서비스 요청에 포함되어 있는 다이제스트 비밀번호를 사용하여 사용자 이름의 유효성을 검사할 수 있습니다.

▼ 새 웹 서비스 보안 인증 모듈 인스턴스를 추가하는 방법

1. **Access Manager 탭에서 인증 하위 탭을 클릭합니다.**
2. **모듈 인스턴스 섹션에서 새로 만들기를 클릭합니다.**
3. **이름 필드에 이 WSSAuth 인증 모듈 인스턴스의 이름을 입력합니다.**

4 유형에는 WSSAuth를 선택합니다.

5 WSSAuth 인증 모듈 인스턴스를 구성합니다.

▼ WSSAuth 인증 모듈 인스턴스를 구성하는 방법

1 Access Manager 탭에서 인증 하위 탭을 클릭합니다.

2 모듈 인스턴스 섹션에서 구성할 WSSAuth 인증 모듈 인스턴스의 이름을 클릭합니다.

3 WSSAuth 인증 모듈 인스턴스 영역 속성에 대한 값을 제공합니다.
다음 표는 구성할 수 있는 속성의 목록과 설명을 제공합니다.

사용자 검색 속성	개발할 항목
사용자 영역	개발할 항목
사용자 비밀번호 속성	개발할 항목
인증 수준	개발할 항목

OAMAuth 인증 모듈 추가

OpenSSO는 Oracle 인증 모듈을 사용하여 이전에 Oracle Access Manager에 인증된 관리자를 인증하고 OpenSSO에 대해 단일 사인 온(SSO)할 수 있습니다. 관리자는 OpenSSO에 자격 증명을 제공할 필요가 없습니다.

▼ 새 Oracle 인증 모듈 인스턴스를 추가하는 방법

1 Access Manager 탭에서 인증 하위 탭을 클릭합니다.

2 모듈 인스턴스 섹션에서 새로 만들기를 클릭합니다.

3 이름 필드에 이 Oracle 인증 모듈 인스턴스의 이름을 입력합니다.

4 유형에는 OAMAuth를 선택합니다.

5 [확인]을 누릅니다.

6 OAMAuth 인증 모듈 인스턴스를 구성합니다.

▼ Oracle 인증 모듈 인스턴스를 구성하는 방법

1 **Access Manager 탭에서 인증 하위 탭을 클릭합니다.**

2 **모듈 인스턴스 섹션에서 구성할 OAMAuth 인증 모듈 인스턴스의 이름을 클릭합니다.**

3 **Oracle 인증 모듈 인스턴스 영역 속성에 대한 값을 제공합니다.**
다음 표는 구성할 수 있는 속성의 목록과 설명을 제공합니다.

원격 사용자 헤더 이름	개발할 항목
허용된 헤더 값	현재 값 목록에는 개발할 항목이 표시됩니다. ■ 헤더 값을 목록에 추가하려면 새 값 필드에 개발할 항목을 입력한 다음 추가를 클릭합니다. ■ 현재 값 목록에서 항목을 제거하려면 항목을 선택한 다음 제거를 클릭합니다.
인증 수준	개발할 항목

보안 토큰 생성

Oracle OpenSSO 보안 토큰 서비스(OpenSSO STS)는 웹 서비스 클라이언트와 웹 서비스 공급자 간에 트러스트 관계를 구축한 다음 양자 간의 트러스트를 중개합니다. 웹 서비스는 여러 클라이언트와 통신할 필요 없이 하나의 entity?OpenSSO STS?에서 발행된 토큰을 신뢰할 수 있습니다. 이를 통해 OpenSSO STS는 트러스트 포인트 관리 오버헤드를 크게 줄입니다.

다음 섹션에는 보안 토큰 필요성을 확인하고 이러한 필요성을 충족하는 보안 토큰을 생성 및 유효성 검사하도록 보안 토큰 서비스를 구성하기 위한 지시사항이 제공됩니다.

OpenSSO STS에 웹 서비스 공급자 등록

새 웹 서비스 공급자 보안 에이전트 프로필을 추가하는 경우 웹 서비스 공급자는 OpenSSO STS에 자동으로 등록됩니다. 자세한 내용은 다음 섹션을 참조하십시오.

웹 서비스 공급자를 OpenSSO STS에 등록하면 웹 서비스 공급자가 사용할 수 있는 웹 클라이언트 보안 토큰을 생성하도록 OpenSSO STS를 구성할 수 있습니다.

OpenSSO STS에서 웹 서비스 클라이언트 보안 토큰 요청

웹 클라이언트 보안 토큰을 생성하도록 보안 토큰 서비스를 구성하려면 웹 서비스 공급자에게 필요한 보안 토큰의 종류를 확인해야 합니다. OpenSSO STS는 Liberty Alliance 프로젝트 보안 토큰 및 웹 서비스 상호 운영성 기본 보안 프로필 보안 토큰을 지원합니다.

보안 토큰 생성 프로세스 흐름

Liberty Alliance 프로젝트 토큰, HTTP 클라이언트 또는 브라우저를 사용하여 보안을 사용 설정하는 경우 웹 서비스 클라이언트를 통해 웹 서비스 공급자로 액세스 요청을 보냅니다. 웹 서비스 보안 에이전트는 요청을 OpenSSO STS 인증 서비스로 리디렉션합니다. Liberty Alliance 프로젝트 보안 메커니즘이 존재하는 경우 HTTP 보안 에이전트는 리디렉션을 발행합니다. WS-IBS 보안이 사용되는 경우 SOAP 보안 에이전트가 리디렉션을 발행합니다.

OpenSSO STS 인증 서비스는 웹 서비스 공급자가 등록한 보안 메커니즘을 확인하고 적절한 보안 토큰을 검색합니다. 인증에 성공하면 웹 서비스 클라이언트가 SOAP 메시지 본문을 제공하고 웹 서비스측의 SOAP 보안 에이전트는 보안 헤더와 토큰을 삽입합니다. 그런 다음 요청이 WSP에 전송되기 전에 메시지가 서명됩니다.

웹 서비스 공급자 자체에 요청을 전달하기 전에 웹 서비스 공급자측의 SOAP 보안 에이전트가 SOAP 요청의 서명 및 보안 토큰을 확인합니다. 그런 다음 웹 서비스 공급자는 이를 처리하고 SOAP 보안 에이전트에서 서명한 응답을 웹 서비스 클라이언트에 다시 반환합니다. 그런 다음 응답을 웹 서비스 클라이언트로 전달하기 전에 웹 서비스 클라이언트측의 SOAP 보안 에이전트가 서명을 확인합니다.

다음 표는 Liberty Alliance 프로젝트 트랜잭션에 지원되는 토큰 목록과 간단한 설명을 제공합니다.

표 3–1 요청자 토큰 - Liberty Alliance 프로젝트

토큰	요구 사항
X.509	■ 보안 웹 서비스는 웹 서비스 클라이언트가 요청자를 식별하고 웹 서비스 공급자를 통해 인증하기 위한 수단으로 공개 키를 공급하는 PKI(공개 키 인프라)를 사용합니다. ■ 보안 웹 서비스는 웹 서비스 클라이언트가 요청자를 식별하고 웹 서비스 공급자를 통해 인증하기 위한 수단으로 공개 키를 공급하는 PKI(공개 키 인프라)를 사용합니다.

표 3–1 요청자 토큰 - Liberty Alliance 프로젝트 *(계속)*

BearerToken	■ 보안 웹 서비스는 SAML(Security Assertion Markup Language) SAML Bearer 토큰 확인 메소드를 사용합니다. ■ WSC는 웹 서비스 공급자에 대해 요청자를 인증하기 위한 수단으로 공개 키 정보와 함께 SAML 명제를 제공합니다. ■ 두 번째 서명은 명제를 SOAP 메시지에 바인딩합니다. ■ 두 번째 서명 바인딩은 Liberty Alliance 프로젝트가 정의한 규칙을 사용합니다.
SAML 토큰	■ 보안 웹 서비스는 SAML Holder-of-Key 확인 메소드를 사용합니다. ■ WSC는 SAML 명제와 디지털 서명을 SOAP 헤더에 추가합니다. ■ 보낸 사람 인증서 또는 공개 키도 서명과 함께 제공됩니다. ■ 전송은 Liberty Alliance 프로젝트에서 정의된 규칙을 사용하여 처리됩니다.

다음 표는 WS-IBS 트랜잭션에 지원되는 토큰 목록과 간단한 설명을 제공합니다.

표 3–2 요청자 토큰 - WS-IBS

토큰	요구 사항
사용자 이름	■ 보안 웹 서비스에는 사용자 이름, 비밀번호 및 필요한 경우 서명된 요청이 필요합니다. ■ 웹 서비스 소비자는 요청자를 식별하기 위한 수단으로 사용자 이름 토큰을 제공합니다. ■ 웹 서비스 소비자는 비밀번호, 공유 비밀 또는 ID를 웹 서비스 공급자에 인증할 때와 같은 비밀번호를 제공합니다.
X.509	보안 웹 서비스는 웹 서비스 소비자가 요청자를 식별하고 웹 서비스 공급자를 통해 인증을 수행하기 위한 수단으로 공개 키를 제공하는 PKI(공개 키 인프라)를 사용합니다.
SAML-Holder-Of-Key	■ 보안 웹 서비스는 SAML Holder-of-Key 확인 메소드를 사용합니다. ■ 웹 서비스 소비자는 웹 서비스 공급자에 대해 요청자를 인증하기 위한 수단으로 공개 키 정보와 함께 SAML 명제를 제공합니다. ■ 두 번째 서명은 명제를 SOAP 페이로드에 바인딩합니다.
SAML-SenderVouches	■ 보안 웹 서비스는 SAML sender-vouches 확인 메소드를 사용합니다. ■ 웹 서비스 소비자는 SAML 명제와 디지털 서명을 SOAP 헤더에 추가합니다. 보낸 사람 인증서 또는 공개 키도 서명과 함께 제공됩니다.

보안 토큰 생성 매트릭스 사용

보안 토큰 생성 매트릭스를 사용하면 웹 서비스 공급자에 필요한 웹 서비스 클라이언트 보안 토큰을 생성하도록 OpenSSO STS를 구성하는 데 도움이 됩니다. 먼저 OpenSSO STS 출력 토큰이라는 마지막 열에서 웹 서비스 공급자 토큰 요구 사항을 충족하는 설명을 찾습니다. 그런 다음 보안 토큰 서비스를 구성할 때 동일한 행의 매개 변수 값을 사용합니다. "토큰 생성 매트릭스 범례"는 테이블 제목 및 사용 가능한 옵션에 대한 정보를 제공합니다. 자세한 구성 지시사항은 5.2.3절, "보안 토큰 서비스를 구성하는 방법"을 참조하십시오. 웹 서비스 보안 및 관련 용어에 대한 일반 정보는 다음을 참조하십시오.

- http://www.oracle.com/technology/tech/standards/pdf/security.pdf
- http://download.oracle.com/docs/cd/E15523_01/web.1111/b32511/intro_security.htm#CDDHHGEE

보안 토큰 생성 매트릭스에는 자주 사용되는 보안 토큰 서비스 매개 변수 설정과 이러한 설정을 기반으로 하여 OpenSSO STS가 생성하는 보안 토큰 유형이 요약되어 있습니다.

표 3–3 보안 토큰 생성 매트릭스

행	메시지 수준 보안 바인딩	웹 서비스 클라이언트 토큰	KeyType	OnBehalfOf Token	사용 키	OpenSSO STS 출력 토큰
1	비대칭	X509	Bearer	예	아니요	SAML Bearer, 증명 키 없음
2	비대칭	사용자 이름	Bearer	예	아니요	SAML Bearer, 증명 키 없음
3	비대칭	X509	Bearer	아니요	아니요	SAML Bearer, 증명 키 없음
4	비대칭	사용자 이름	Bearer	아니요	아니요	SAML Bearer, 증명 키 없음
5	비대칭	X509	대칭	예	아니요	SAML Holder-of-Key, 대칭 증명 키
6	비대칭	사용자 이름	대칭	예	아니요	SAML Holder-of-Key, 대칭 증명 키
7	비대칭	X509	대칭	아니요	아니요	SAML Holder-of-Key, 대칭
8	비대칭	사용자 이름	대칭	아니요	아니요	SAML Holder-of-Key, 대칭 증명 키

표 3–3 보안 토큰 생성 매트릭스 *(계속)*

9	비대칭	X509	비대칭	아니요	웹 서비스 클라이언트 공개 키	SAML Holder-of-Key, 비대칭 증명 키
10	비대칭	X509	SAML sender-vouches에 대한 Oracle 독점	예	아니요	SAML sender-vouches, 증명 키 없음
11	비대칭	사용자 이름	SAML sender-vouches에 대한 Oracle 독점	예	아니요	SAML sender-vouches, 증명 키 없음
12	비대칭	X509	SAML sender-vouches에 대한 Oracle 독점	아니요	아니요	오류
13	비대칭	사용자 이름	SAML sender-vouches에 대한 Oracle 독점	아니요	아니요	오류
14	전송	사용자 이름	Bearer	예	아니요	SAML Bearer, 증명 키 없음
15	전송	사용자 이름	Bearer	아니요	아니요	SAML Bearer, 증명 키 없음
16	전송	사용자 이름	대칭	예	아니요	SAML Holder-of-Key, 대칭
17	전송	사용자 이름	대칭	아니요	아니요	SAML Holder-of-Key, 대칭 증명 키
18	전송	사용자 이름	SAML sender-vouches에 대한 Oracle 독점	예	아니요	SAML sender-vouches, 증명 키 없음
19	전송	사용자 이름	SAML sender-vouches에 대한 Oracle 독점	아니요	아니요	오류

표 3–3 보안 토큰 생성 매트릭스 *(계속)*

20	비대칭	사용자 이름	비대칭	아니요	웹 서비스 클라이언트 공개 키	오류
21	전송	사용자 이름	비대칭	아니요	웹 서비스 클라이언트 공개 키	오류
22	비대칭	X509	비대칭	예	아니요	오류
23	비대칭	사용자 이름	비대칭	예	아니요	오류
24	전송	사용자 이름	비대칭	예	아니요	오류
25	비대칭	X509	비대칭	아니요	아니요	SAML Holder-of-Key, 비대칭 증명 키
26	비대칭	X509	아니요	아니요	아니요	SAML Holder-of-Key, 비대칭 증명 키
27	비대칭	사용자 이름	아니요	아니요	아니요	SAML Holder-of-Key, 대칭 증명 키
28	전송	사용자 이름	아니요	아니요	아니요	SAML Holder-of-Key, 대칭 증명 키

보안 토큰 서비스 문제 및 해결 방법

개발할 항목

구성 문제 및 해결 방법

개발할 항목

설명서 오류 정보

개발할 항목

◆ ◆ ◆ 4 장

4 Oracle OpenSSO Fedlet 사용

이 섹션에서는 Oracle OpenSSO Fedlet에 대한 다음 정보를 제공합니다.

Oracle OpenSSO Fedlet 정보

Oracle OpenSSO Fedlet는 Java 또는 .NET 서비스 공급자 응용 프로그램과 함께 배포될 수 있는 경량의 SP(서비스 공급자) 구현으로서, 응용 프로그램이 SAMLv2 프로토콜을 사용하여 Oracle OpenSSO 8.0 업데이트 2 등의 IDP(아이디 공급자)와 통신할 수 있도록 합니다. Fedlet에는 두 가지 버전이 있으며 플랫폼에 따라 달라집니다.

- Java Fedlet은 OpenSSO 8.0에서 처음 출시되었습니다. 자세한 내용은 **Sun OpenSSO Enterprise 8.0 Deployment Planning Guide**의 5 장, “Using the OpenSSO Enterprise Fedlet to Enable Identity Federation”을 참조하십시오.
- .NET Fedlet은 OpenSSO 8.0 업데이트 1에서 출시되었습니다. 자세한 내용은 **Sun OpenSSO Enterprise 8.0 Update 1 Release Notes**의 10 장, “Using the ASP.NET Fedlet with OpenSSO Enterprise 8.0 Update 1”을 참조하십시오.

Oracle OpenSSO 8.0 업데이트 2에서 Fedlet을 다음과 같이 사용할 수 있습니다.

- OpenSSO 8.0 업데이트 2 ZIP 파일을 압축 해제한 후, 다음 파일에서 Java Fedlet과 .NET Fedlet을 모두 사용할 수 있습니다.

  zip-root`/opensso/fedlet/fedlet-unconfigured.zip`(*zip-root*는 Oracle OpenSSO 8.0 업데이트 2 ZIP 파일을 압축 해제한 위치임)
- Oracle OpenSSO 8.0 업데이트 2가 설치되면 일반 작업 아래 Fedlet 작업 흐름 만들기를 사용하여 OpenSSO 8.0 관리 콘솔에서 Java Fedlet을 만들 수 있습니다.

Oracle OpenSSO Fedlet 요구 사항

Fedlet에는 다음과 같은 요구 사항이 있습니다.

- `fedlet.war` 또는 Fedlet과 통합된 Java 서비스 공급자 응용 프로그램을 배포하려는 경우, Oracle OpenSSO 8.0 업데이트 2에서 지원되는 웹 컨테이너. 12 페이지 "OpenSSO 8.0 업데이트 2의 하드웨어 및 소프트웨어 요구 사항"을 참조하십시오.
- .NET Fedlet을 배포하려는 경우 Microsoft IIS(Internet Information Server) 7.0 이상
- JDK 1.6.x 이상

Oracle OpenSSO Fedlet 구성

이 섹션에서는 서비스 공급자 응용 프로그램을 사용하여 Fedlet을 처음 구성하는 방법을 설명합니다.

- 42 페이지 "Java Fedlet을 구성하는 방법"
- 44 페이지 ".NET Fedlet을 구성하는 방법"

Fedlet에 대한 초기 구성을 마친 후 수행하려는 추가 구성을 계속합니다. 다음과 같은 사항을 고려해야 합니다.

- Fedlet `sp.xml` 파일을 수정하는 경우 이 파일을 아이디 공급자로 다시 가져와야 합니다.
- 서비스 공급자측에서 다른 Fedlet 구성을 변경하는 경우 이 정보를 아이디 공급자 관리자에게 전달하여 아이디 공급자측에서 필요에 따라 구성을 변경할 수 있도록 합니다.

▼ Java Fedlet을 구성하는 방법

1 **아이디 공급자측에서 아이디 공급자에 대한 XML 메타데이터를 생성하고 `idp.xml`이라는 파일에 메타데이터를 저장합니다.**

Oracle OpenSSO 8.0 업데이트 2의 경우, `exportmetadata.jsp`를 사용합니다. 예:

```
http://opensso-idp.example.com:8080/opensso/saml2/jsp/exportmetadata.jsp
```

2 **서비스 공급자측에서 Fedlet ZIP 파일을 압축 해제합니다(필요한 경우).**

3 **Fedlet 홈 디렉토리를 만듭니다. 이것은 Fedlet이 메타데이터, 트러스트 그룹 및 구성 등록 정보 파일을 읽는 디렉토리입니다.**

기본 위치는 Fedlet 웹 컨테이너를 실행하는 사용자의 홈 디렉토리 아래 있는 fedlet 하위 디렉토리입니다(`user.home` JVM 등록 정보에 표시됨). 예를 들어 이 홈 디렉토리가 `/home/webservd`인 경우 Fedlet 홈 디렉토리는 다음과 같습니다.

```
/home/webservd/fedlet
```

Fedlet 기본 홈 디렉토리를 변경하려면 JVM 런타임 `com.sun.identity.fedlet.home` 등록 정보의 값을 원하는 위치로 설정합니다. 예:

```
-Dcom.sun.identity.fedlet.home=/export/fedlet/conf
```

그런 다음 Fedlet이 `/export/fedlet/conf` 디렉토리에서 메타데이터, 트러스트 그룹 및 구성 파일을 읽습니다.

4 Java Fedlet `java/conf` 디렉토리에서 Fedlet 홈 디렉토리로 다음 파일을 복사합니다.

- `sp.xml-template`
- `sp-extended.xml-template`
- `idp-extended.xml-template`
- `fedlet.cot-template`

5 Fedlet 홈 디렉토리에서 복사한 파일의 이름을 바꾸고 각 이름에서 `-template`을 삭제합니다.

6 Fedlet 홈 디렉토리에서 복사하고 이름을 바꾼 파일에서 다음 표에 표시된 대로 태그를 바꿉니다.

태그	바꾸기
`FEDLET_COT`	원격 아이디 공급자 및 Java Fedlet 서비스 공급자 응용 프로그램이 구성원으로 포함된 COT(트러스트 그룹)의 이름입니다.
`FEDLET_ENTITY_ID`	Java Fedlet 서비스 공급자 응용 프로그램의 ID(이름)입니다. 예: `fedletsp`
`FEDLET_PROTOCOL`	Java Fedlet 서비스 공급자 응용 프로그램에 대한 웹 컨테이너(예: `fedlet.war`)의 프로토콜입니다. 예: `https`
`FEDLET_HOST`	Java Fedlet 서비스 공급자 응용 프로그램에 대한 웹 컨테이너(예: `fedlet.war`)의 호스트 이름입니다. 예: `fedlet-host.example.com`
`FEDLET_PORT`	Java Fedlet 서비스 공급자 응용 프로그램에 대한 웹 컨테이너(예: `fedlet.war`)의 포트 번호입니다. 예: `80`
`FEDLET_DEPLOY_URI`	Java Fedlet 서비스 공급자 응용 프로그램의 URL입니다. 예: `http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	원격 아이디 공급자의 ID(이름)입니다. 예: `openssoidp`

참고: Fedlet 서비스 공급자 또는 아이디 공급자 엔티티 ID에 퍼센트 기호(%) 또는 쉼표(,)가 포함된 경우 `fedlet.cot` 파일에서 이름을 바꾸기 전에 문자를 제외해야 합니다. 예를 들어 "%"를 "%25"로, ","를 "%2C"로 변경합니다.

7 Java Fedlet `java/conf` 디렉토리에서 Fedlet home 디렉토리로 `FedletConfiguration.properties` 파일을 복사합니다.

8 **1단계의 아이디 공급자 표준 메타데이터 XML 파일을 Fedlet 홈 디렉토리로 복사합니다. 이 파일의 이름은 `idp.xml`로 지정되어야 합니다.**

9 **Java Fedlet XML 메타데이터 파일(`sp.xml`)을 아이디 공급자로 가져옵니다.**

Oracle OpenSSO 8.0 업데이트 2의 경우, OpenSSO 8.0 관리 콘솔에서 일반 작업 아래의 원격 서비스 공급자 등록 작업 흐름을 사용하여 Java Fedlet 서비스 공급자 메타데이터를 가져오고 Java Fedlet 서비스 공급자를 트러스트 그룹에 추가합니다.

다음 순서 요구 사항에 따라 Java Fedlet에 대한 추가 구성을 계속합니다.

▼ .NET Fedlet을 구성하는 방법

1 **아이디 공급자측에서 아이디 공급자에 대한 XML 메타데이터를 생성하고 `idp.xml`이라는 파일에 메타데이터를 저장합니다.**

Oracle OpenSSO 8.0 업데이트 2의 경우, `exportmetadata.jsp`를 사용합니다. 예:

```
http://opensso-idp.example.com:8080/opensso/saml2/jsp/exportmetadata.jsp
```

2 **서비스 공급자측에서 Fedlet ZIP 파일을 압축 해제합니다(필요한 경우).**

3 **.NET Fedlet `asp.net/conf` 폴더에서 응용 프로그램의 `App_Data` 폴더로 다음 파일을 복사합니다.**

- `sp.xml-template`
- `sp-extended.xml-template`
- `idp-extended.xml-template`
- `fedlet.cot-template`

4 **`App_Data` 폴더에서 복사한 파일의 이름을 바꾸고 각 이름에서 `-template`을 삭제합니다.**

5 **`App_Data` 폴더에서 복사하고 이름을 바꾼 파일에서 다음 표에 표시된 대로 태그를 바꿉니다.**

태그	바꾸기
`FEDLET_COT`	원격 아이디 공급자 및 .NET Fedlet 서비스 공급자 응용 프로그램이 구성원으로 포함된 COT(트러스트 그룹)의 이름입니다.
`FEDLET_ENTITY_ID`	.NET Fedlet 서비스 공급자 응용 프로그램의 ID(이름)입니다. 예: `fedletsp`
`FEDLET_DEPLOY_URI`	.NET Fedlet 서비스 공급자 응용 프로그램의 URL입니다. 예: `http://fedletsp.example.com/myFedletApp`
`IDP_ENTITY_ID`	원격 아이디 공급자의 ID(이름)입니다. 예: `openssoidp`

6 **1단계의 아이디 공급자 표준 메타데이터 XML 파일을 응용 프로그램의 `App_Data` 폴더로 복사합니다. 이 파일의 이름은 `idp.xml`로 지정되어야 합니다.**

7 **`Fedlet.dll` 및 `Fedlet.dll.config` 파일을 .NET Fedlet `asp.net/bin` 폴더에서 응용 프로그램의 `bin` 폴더로 복사합니다.**

8 **.NET Fedlet XML 메타데이터 파일(`sp.xml`)을 아이디 공급자로 가져옵니다.**

Oracle OpenSSO 8.0 업데이트 2의 경우, OpenSSO 8.0 관리 콘솔에서 일반 작업 아래의 원격 서비스 공급자 등록 작업 흐름을 사용하여 .NET Fedlet 서비스 공급자 메타데이터를 가져오고 .NET Fedlet 서비스 공급자를 트러스트 그룹에 추가합니다.

다음 순서 요구 사항에 따라 .NET Fedlet에 대한 추가 구성을 계속합니다.

OpenSSO 8.0 업데이트 2에서 제공되는 Fedlet의 새로운 기능

Oracle OpenSSO 8.0 업데이트 2에는 다음과 같은 Fedlet의 새로운 기능이 포함되어 있습니다.

Fedlet 버전 정보(CR 6941387)

Oracle OpenSSO Fedlet에는 버전 정보가 포함되어 있습니다. Fedlet 패키지(ZIP 파일)의 파일을 추출한 후 다음 파일 중 하나를 확인하여 Fedlet 버전을 판별합니다.

- Java Fedlet: `java/conf/FederationConfig.properties`
- .NET Fedlet: `asp.net/bin/Fedlet.dll.config`

Java Fedlet 비밀번호 암호화 및 해독(CR 6930477)

Java Fedlet은 `fedlet.war` 파일에 `fedletEncode.jsp`를 제공하여 `storepass` 및 `keypass` 비밀번호를 암호화합니다. 기본적으로 각 Fedlet에 대해 서로 다른 암호화 키가 생성됩니다. 이 암호화 키를 변경하려면 Fedlet `FederationConfig.properties` 파일에서 `am.encryption.pwd` 등록 정보를 설정합니다.

서명 및 암호화에 대한 Java Fedlet 지원

Java Fedlet은 암호화된 `assertion` 및 `NameID` 요소의 XML 서명 확인 및 해독과 해당 속성을 지원합니다.

▼ 서명 및 암호화를 위해 Java Fedlet을 구성하는 방법

1. **`keytool` 유틸리티를 사용하여 `keystore.jks`라는 키 저장소 파일을 만듭니다.**
2. **서명에 사용되는 개인 키(및 해당되는 경우 공용 인증서)와 암호화에 사용되는 개인 키(및 해당되는 경우 공용 인증서)를 `keystore.jks` 파일에 추가합니다.**
3. **`.storepass` 파일을 만듭니다.**
4. **`.storepass` 파일에 비밀번호를 추가합니다. 비밀번호를 암호화하려면 `fedletEncode.jsp`를 사용합니다.**
5. **`.keypass` 파일을 만듭니다.**
6. **`.keypass` 파일에 비밀번호를 추가합니다. 비밀번호를 암호화하려면 `fedletEncode.jsp`를 사용합니다.**
7. **일반 텍스트 비밀번호를 사용하는 경우 `FederationConfig.properties` 파일에 다음 줄을 주석으로 추가합니다.**

   ```
   com.sun.identity.saml.xmlsig.passwordDecoder=
       com.sun.identity.fedlet.FedletEncodeDecode
   ```

8. **`FederationConfig.properties` 파일에서 다음 속성의 전체 경로를 설정합니다. *path*는 각 파일에 대한 전체 경로입니다.**

   ```
   com.sun.identity.saml.xmlsig.keystore=path/keystore.jks
   com.sun.identity.saml.xmlsig.storepass=path/.storepass
   com.sun.identity.saml.xmlsig.keypass=path/.keypass
   ```

9. **`keytool`을 사용하여 서명 인증서를 내보냅니다. 예:**

   ```
   keytool -export -keystore keystore.jks -rfc -alias test
   ```

도구에 keystore.jks에 액세스하는 데 사용되는 비밀번호를 입력하라는 메시지가 표시되고 인증서가 생성됩니다.

10 암호화 인증서가 필요한 경우 이전 단계에 표시된 대로 keytool을 사용하여 인증서를 내보냅니다. 또는 서명과 암호화 모두에 동일한 인증서를 사용합니다.

11 KeyDescriptor XML 블록을 만들고 암호화 인증서를 블록에 추가합니다. 예를 들어 KeyDescriptor 요소의 use="signing" 태그를 보십시오.

```
<KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
</KeyDescriptor>
```

12 다른 KeyDescriptor XML 블록을 만들고 암호화 인증서를 이 블록에 추가합니다. 예를 들어 KeyDescriptor 요소의 use="encryption" 태그를 보십시오.

```
<KeyDescriptor use="encryption">
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <X509Data>
          <X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </X509Certificate>
        </X509Data>
      </KeyInfo>
<EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
  <KeySize xmlns="http://www.w3.org/2001/04/xmlenc#">128</KeySize>
</EncryptionMethod>
</KeyDescriptor>
```

13 **Java Fedlet `sp.xml` 파일에서 서명 및 암호와 인증서가 포함된 XML 블록을 `SPSSODescriptor` 요소 아래에 추가합니다. `SPSSODescriptor` 요소 샘플은 예 4–1을 참조하십시오.**

`AuthnRequestsSigned` 속성이 `true`로 설정되어 모든 인증 요청을 서명하도록 Java Fedlet을 구성합니다.

14 **Java Fedlet `sp-extended.xml` 파일에서 다음 요소에 대한 값을 설정합니다.**

- `signingCertAlias`는 키 저장소에 있는 XML 서명 인증서의 별칭을 포함합니다.
- `encryptionCertAlias`는 키 저장소에 있는 XML 암호화 인증서의 별칭을 포함합니다.

15 **Java Fedlet 서비스 공급자가 암호화하는 항목을 적용하려면 `sp-extended.xml` 파일에서 다음 속성을 `true`로 설정합니다.**

- `wantAssertionEncrypted`
- `wantNameIDEncrypted`
- `wantAttributeEncrypted`

16 **Java Fedlet 서비스 공급자가 서명하고 서명되기를 원하는 항목을 적용하려면 다음 속성을 `true`로 설정합니다.**

- `idp.xml` 파일의 `wantAuthnRequestsSigned`는 Fedlet에 서명할 항목을 알려줍니다.
- `sp.xml` 파일의 `AuthnRequestsSigned` 및 `WantAssertionsSigned`는 아이디 공급자에게 Fedlet이 서명하게 될 항목을 알려줍니다.
- `sp-extended.xml` 파일의 `wantArtifactResponseSigned`는 Fedlet에 서명할 항목을 알려줍니다.
- `sp-extended.xml` 파일의 `wantPOSTResponseSigned`
- `sp-extended.xml` 파일의 `wantLogoutRequestSigned`
- `sp-extended.xml` 파일의 `wantLogoutResponseSigned`

아이디 공급자가 특정 메시지에 대한 서명을 요구하는 경우 `idp-extended.xml` 파일에서 각 속성을 `true`로 설정합니다. `wantLogoutRequestSigned` 및 `wantLogoutResponseSigned`가 그 예에 해당합니다.

주 – `sp-extended.xml` 파일에서 속성을 설정하는 경우 이 정보를 아이디 공급자 관리자에게 전달하여 필요에 따라 아이디 공급자의 구성을 변경할 수 있습니다.

---

17 **Java Fedlet 웹 컨테이너를 다시 시작합니다.**

18 **Java Fedlet `sp.xml` 파일을 아이디 공급자로 가져옵니다.**

예 4–1 Java Fedlet 샘플 `SPSSODescriptor` 요소

```
<EntityDescriptor entityID="fedlet"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
```

```
<SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="false"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<b><KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>

</KeyDescriptor></b>
<b><KeyDescriptor use="encryption">
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <X509Data>
          <X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
          </X509Certificate>
        </X509Data>
      </KeyInfo>

<EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
<KeySize xmlns="http://www.w3.org/2001/04/xmlenc#">128</KeySize>
</EncryptionMethod>
</KeyDescriptor></b>
<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat
><AssertionConsumerService index="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://server.sun.com:7070/fedlet/fedletapplication"/>
</SPSSODescriptor>
</EntityDescriptor>
```

속성 쿼리에 대한 Java Fedlet 지원(CR 6930476)

Java Fedlet은 SAMLv2 속성 쿼리를 지원하여 Oracle OpenSSO 8.0 업데이트 2와 같은 아이디 공급자에게 특정 아이디 속성 값을 쿼리합니다. Fedlet이 쿼리를 서명하고 암호화하도록 구성할 수 있습니다. 서명은 Fedlet 쿼리를 실행하는 데 반드시 필요하지만 암호화는 선택 사항입니다.

▼ 속성 쿼리를 위해 Java Fedlet을 구성하는 방법

1 **XML 서명을 사용하여 46 페이지 "서명 및 암호화에 대한 Java Fedlet 지원"에 설명된 대로 속성 쿼리를 서명합니다.**

2 **이전 단계에서 생성된 인증서를 Fedlet `sp.xml` 파일의 `RoleDescriptor` 요소에 추가합니다. 다음 예에는 인증서를 붙여넣는 두 개의 `KeyDescriptor` 태그가 있습니다. 하나는 서명에 사용되고 다른 하나는 암호화에 사용됩니다. 암호화를 사용하지 않는 경우 `KeyDescriptor use="encryption" tag`는 필요 없습니다.**

```
<RoleDescriptor xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:query="urn:oasis:names:tc:SAML:metadata:ext:query"
      xsi:type="query:AttributeQueryDescriptorType"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            --certificate--
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
  </KeyDescriptor>
  <KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            --certificate--
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    <EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
<xenc:KeySize
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">128</xenc:KeySize>
    </EncryptionMethod>
  </KeyDescriptor>
</RoleDescriptor>
```

3 **Java Fedlet `sp-extended.xml` 파일에서 `signingCertAlias` 속성에 대한 값을 지정하고, 구성된 경우 `encryptionCertAlias` 속성에 대한 값도 지정합니다.**

아이디 공급자가 명제를 암호화하도록 구성하려면 `NameID` 요소도 암호화합니다. 그러므로 `wantNameIDEncrypted` 속성의 값을 `true`로 설정해야 합니다. XML 코드를 `AttributeQueryConfig` 요소에 추가합니다. 예:

```
<Attribute name="signingCertAlias">
      <Value>test</Value>
</Attribute>
<Attribute name="encryptionCertAlias">
      <Value>test</Value>
</Attribute>
<Attribute name="wantNameIDEncrypted">
      <Value>true</Value>
</Attribute>
```

이 예에서 `test`는 샘플 키의 별칭입니다.

4 **Java Fedlet 메타데이터 파일(`sp.xml`)을 아이디 공급자로 가져옵니다.**

또한 Fedlet에 대한 속성 쿼리를 지원하도록 아이디 공급자에서 추가 구성 단계를 수행합니다.

요청 및 응답의 .NET Fedlet 암호화 및 해독(CR 6939005)

.NET Fedlet은 NameID, Attribute 및 Assertion 요소에 대해 발신 XML 요청을 암호화하고 수신 응답을 해독할 수 있습니다.

▼ 요청 및 응답의 암호화 및 해독을 위해 .NET Fedlet을 구성하는 방법

1 **Microsoft 관리 콘솔에 대한 인증서 스냅인을 사용하여 X.509 인증서를 로컬 컴퓨터 계정 내의 개인 폴더로 가져옵니다. 이 스냅인을 사용하려면 다음 Microsoft 문서를 참조하십시오.**

`http://msdn.microsoft.com/ko-kr/library/ms788967.aspx`

2 **등록 정보 대화 상자를 보고 값을 입력하여 이 인증서의 표시 이름을 지정합니다. 4단계를 수행하기 위해 이 값을 저장합니다.**

3 **Microsoft 문서에 설명된 대로 IIS(Internet Information Server)에서 사용하는 사용자 계정에 대해 인증서에 읽기 액세스 권한을 허용하도록 적절한 권한을 설정합니다. 예:**

a. **인증서 스냅인에서 작업, 모든 작업 및 개인 키 관리로 차례로 이동합니다.**

b. **IIS(일반적으로 네트워크 서비스)를 실행하는 사용자 계정에 대해 읽기 허용 권한을 지정합니다.**

4 **.NET Fedlet의 확장 메타데이터 파일(`sp-extended.xml`)에서 `encryptionCertAlias` 속성 값으로 2단계에서 지정한 표시 이름을 지정합니다. 예:**

```
<Attribute name="encryptionCertAlias">
<Value>MyFedlet</Value>
```

5 **.NET Fedlet의 서비스 공급자 메타데이터 파일(`sp.xml`)에서 암호화 키에 대한 `KeyDescriptor`를 추가합니다.**

이전에 사용된 Microsoft 관리 콘솔에 대한 인증서 스냅인을 사용하여 `KeyDescriptor` XML 블록에 포함되도록 인증서의 공개 키를 Base64 인코딩으로 내보냅니다. 이 `KeyDescriptor` 는 `SPSSODescriptor` 내의 첫 번째 자식 요소여야 합니다. 예:

```
<KeyDescriptor use="encryption">
           <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
             <ds:X509Data>
               <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
               </ds:X509Certificate>
             </ds:X509Data>
           </ds:KeyInfo>
           <EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
           <KeySize
xmlns="http://www.w3.org/2001/04/xmlenc#">128</KeySize>
           </EncryptionMethod>
</KeyDescriptor>
```

6 **.NET 응용 프로그램과 연결된 응용 프로그램 풀을 다시 시작합니다.**

다음 순서 이 구성을 테스트하려면 샘플 응용 프로그램을 사용합니다. 또한 구성된 메타데이터를 적절하게 변경하여 아이디 공급자의 요청을 암호화하고 응답을 해독하도록 다음 속성을 설정합니다.

- Assertion: `sp-extended.xml` 메타데이터 파일에서 `wantAssertionEncrypted` 속성을 `true`로 설정하여 .NET Fedlet이 아이디 공급자로부터 수신되는 응답의 `EncryptedAssertion` 요소를 해독하도록 합니다.
- Attribute: `sp-extended.xml` 메타데이터 파일에서 `wantAttributeEncrypted` 속성을 `true`로 설정하여 .NET Fedlet이 아이디 공급자로부터 수신되는 응답의 `EncryptedAttribute` 요소를 해독하도록 합니다.

- NameID: `idp-extended.xml` 메타데이터 파일에서 `wantNameIDEncrypted` 속성을 `true`로 설정하여 .NET Fedlet이 발신되는 요청의 `NameID` 요소를 암호화하도록 합니다. `sp-extended.xml`에서 이 동일한 속성을 설정하여 .NET Fedlet이 아이디 공급자로부터 수신되는 응답의 `EncryptedID` 요소를 해독하도록 합니다.

요청 및 응답의 .NET Fedlet 서명(CR 6928530)

.NET Fedlet은 `Authn` 요청 및 로그아웃 요청과 같은 발신 XML 요청의 서명을 지원합니다.

▼ 요청 및 응답의 서명을 위해 .NET Fedlet을 구성하는 방법

1 Microsoft 관리 콘솔에 대한 인증서 스냅인을 사용하여 X.509 인증서를 로컬 컴퓨터 계정 내의 개인 폴더로 가져옵니다. 이 스냅인을 사용하려면 다음 Microsoft 문서를 참조하십시오.

`http://msdn.microsoft.com/ko-kr/library/ms788967.aspx`

2 등록 정보 대화 상자를 보고 값을 입력하여 이 인증서의 표시 이름을 지정합니다. 4단계를 수행하기 위해 이 값을 저장합니다.

3 Microsoft 문서에 설명된 대로 IIS(Internet Information Server)에서 사용하는 사용자 계정에 대해 인증서에 읽기 액세스 권한을 허용하도록 적절한 권한을 설정합니다. 예:

a. 인증서 스냅인에서 작업, 모든 작업 및 개인 키 관리로 차례로 이동합니다.

b. IIS(일반적으로 네트워크 서비스)를 실행하는 사용자 계정에 대해 읽기 허용 권한을 지정합니다.

4 .NET Fedlet의 확장 메타데이터 파일(`sp-extended.xml`)에서 `signingCertAlias` 속성 값으로 2단계에서 지정한 표시 이름을 지정합니다. 예:

```
<Attribute name="signingCertAlias">
<Value>MyFedlet</Value>
```

5 .NET Fedlet의 서비스 공급자 메타데이터 파일(`sp.xml`)에서 서명 키에 대한 `KeyDescriptor`를 추가합니다.

이전에 사용된 Microsoft 관리 콘솔에 대한 인증서 스냅인을 사용하여 `KeyDescriptor` XML 블록에 포함되도록 인증서의 공개 키를 Base64 인코딩으로 내보냅니다. 이 `KeyDescriptor`는 `SPSSODescriptor` 내의 첫 번째 자식 요소여야 합니다. 예:

```
<KeyDescriptor use="signing">
             <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                   <ds:X509Certificate>
MIICQDCCAakCBEeNB0swDQYJKoZIhvcNAQEEBQAwZzELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh
bGlmb3JuaWExFDASBgNVBAcTC1NhbnRhIENsYXJhMQwwCgYDVQQKEwNTdW4xEDAOBgNVBAsTB09w
ZW5TU08xDTALBgNVBAMTBHRlc3QwHhcNMDgwMTE1MTkxOTM5WhcNMTgwMTEyMTkxOTM5WjBnMQsw
```

```
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxMLU2FudGEgQ2xhcmExDDAK
BgNVBAoTA1N1bjEQMA4GA1UECxMHT3BlblNTTzENMAsGA1UEAxMEdGVzdDCBnzANBgkqhkiG9w0B
AQEFAAOBjQAwgYkCgYEArSQc/U75GB2AtKhbGS5piiLkmJzqEsp64rDxbMJ+xDrye0EN/q1U5Of\+
RkDsaN/igkAvV1cuXEgTL6RlafFPcUX7QxDhZBhsYF9pbwtMzi4A4su9hnxIhURebGEmxKW9qJNY
Js0Vo5+IgjxuEWnjnnVgHTs1+mq5QYTA7E6ZyL8CAwEAATANBgkqhkiG9w0BAQQFAAOBgQB3Pw/U
QzPKTPTYi9upbFXlrAKMwtFf2OW4yvGWWvlcwcNSZJmTJ8ARvVYOMEVNbsT4OFcfu2/PeYoAdiDA
cGy/F2Zuj8XJJpuQRSE6PtQqBuDEHjjmOQJ0rV/r8mO1ZCtHRhpZ5zYRjhRC9eCbjx9VrFax0JDC
/FfwWigmrW0Y0Q==
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
</KeyDescriptor>
```

6 **.NET 응용 프로그램과 연결된 응용 프로그램 풀을 다시 시작합니다.**

.NET Fedlet 단일 로그아웃(CR 6928528 및 CR 6930472)

.NET Fedlet은 아이디 공급자 시작 및 서비스 공급자 시작 단일 로그아웃을 모두 지원합니다. 단일 로그아웃을 구현하기 위해 .NET Fedlet 샘플 응용 프로그램은 `asp.net/SampleApp` 폴더에 `logout.aspx` 및 `spinitiatedslo.aspx` 파일을 포함하고 있습니다. Fedlet 단일 로그아웃 기능의 작동 방식을 보려면 .NET Fedlet 샘플 응용 프로그램을 배포하십시오.

▼ 단일 로그아웃을 위해 .NET Fedlet 서비스 공급자 응용 프로그램을 구성하는 방법

1 **.NET Fedlet을 구성하지 않은 경우 `Readme` 파일의 단계를 따릅니다.**

2 **.NET 응용 프로그램 공개 내용 내의 `logout.aspx` 및 `spinitiatedslo.aspx` 파일을 복사합니다.**

3 **응용 프로그램에 대한 구성 파일을 변경합니다.**

- `sp.xml` 파일에서 `logout.aspx` 파일의 경로가 응용 프로그램에 대한 파일의 올바른 위치를 가리키도록 합니다.
- `idp.xml` 파일에서(또는 아이디 공급자를 구성하는 동안) `spinitiatedslo.aspx` 파일의 경로가 응용 프로그램에 대한 파일의 올바른 위치를 가리키도록 합니다.

4 **로그아웃 요청 및 로그아웃 응답이 서명되도록 하려는 경우 `sp-extended.xml` 및 `idp-extended.xml` 파일에서 다음 속성을 `true`로 설정합니다.**

- `wantLogoutRequestSigned`
- `wantLogoutResponseSigned`

5 **Fedlet 서비스 공급자 메타데이터 파일(sp.xml)을 아이디 공급자로 가져옵니다.**

또한 아이디 공급자 관리자에게 Fedlet 서비스 공급자에 대해 단일 로그아웃을 구성하여 아이디 공급자 구성에 필요한 추가 변경이 수행되도록 할 수 있다는 것을 알립니다.

.NET Fedlet 서비스 공급자의 단일 사인 온(SSO) 시작(CR 6928525)

.NET Fedlet은 SAMLv2 서비스 공급자에서 시작하는 단일 사인 온(SSO)을 지원합니다. 또한 .NET Fedlet이 아티팩트를 수신한 후 실행 중인 아이디 공급자의 아티팩트 해결 서비스를 사용하여 SOAP를 통해 이를 해결할 수 있도록 하려면 아티팩트 지원이 필요합니다.

.NET Fedlet 샘플 응용 프로그램은 단일 사인 온(SSO)을 구성할 수 있는 방법을 보여줍니다. 응용 프로그램에 필요한 아티팩트가 설치된 다음, 아이디 공급자가 인증을 성공적으로 수행한 후 SAMLv2 응답이 포함된 HTTP POST를 수신하려면 특정 URI가 필요합니다. 다음 코드 예는 .NET 응용 프로그램에서 이 정보를 검색할 수 있는 방법을 보여줍니다.

예 4–2 .NET Fedlet 응용 프로그램에서 `AuthnResponse`를 검색하는 코드 예

```
AuthnResponse authnResponse = null;
try
{
      ServiceProviderUtility spu = new ServiceProviderUtility(Context);
      authnResponse = spu.GetAuthnResponse(Context);
}
catch (Saml2Exception se)
{
      // invalid AuthnResponse received
}
catch (ServiceProviderUtilityException spue)
{
      // issues with deployment (reading metadata)
}
```

응용 프로그램이 SAMLv2 응답을 수신하는 경우 `authnResponse` 객체에는 명제 정보가 입력됩니다. 샘플 응용 프로그램은 이 객체에서 속성 및 주제 정보를 검색하는 방법을 보여줍니다.

여러 아이디 공급자 및 검색 서비스에 대한 .NET Fedlet 지원(CR 6928524)

.NET Fedlet은 여러 아이디 공급자 및 아이디 공급자 검색 서비스를 지원합니다.

일부 배포에서는 Oracle OpenSSO 8.0 업데이트 2와 같은 여러 아이디 공급자로 .NET Fedlet을 구성해야 하는 경우가 있습니다. 추가할 각 아이디 공급자에 대해 다음 작업을 수행합니다.

▼ 여러 아이디 공급자에 대해 .NET Fedlet을 구성하는 방법

1. **추가 아이디 공급자에서 XML 메타데이터 파일을 가져옵니다.**

2. **추가 아이디 공급자 메타데이터 파일의 이름을 `idp`*n*`.xml`로 지정합니다.** *n*은 추가하는 아이디 공급자입니다. 예를 들어 두 번째 아이디 공급자 파일을 **`idp2.xml`**로, 세 번째를 **`idp3.xml`**로 지정합니다. 이 절차에서는 **`idp2.xml`**을 파일 이름으로 사용합니다.

3. **2단계의 `idp2.xml` 파일을 응용 프로그램의 `App_Data` 폴더로 복사합니다.**

4. **새 아이디 공급자를 .NET Fedlet 트러스트 그룹에 추가합니다.**

   새 아이디 공급자를 기존 트러스트 그룹에 추가하는 방법

   응용 프로그램의 `App_Data` 폴더에 있는 `fedlet.cot` 파일에서 쉼표(,)를 구분 기호로 사용하여 새 IDP 엔티티 ID(`idp2.xml` 메타데이터 파일의 `entityID` 속성에 표시됨)를 `sun-fm-trusted-providers` 속성의 값에 추가합니다.

   새 아이디 공급자를 새 트러스트 그룹에 추가하는 방법

   a. **응용 프로그램의 `App_Data` 폴더에 `fedlet2.cot`라는 새 파일을 만듭니다. 기존 `fedlet.cot`를 서식 파일로 사용하지만 `cot-name` 속성의 값을 새 트러스트 그룹의 이름(예: `cot2`)으로 변경합니다. 새 아이디 공급자 엔티티 ID와 Fedlet 엔티티 ID를 모두 `sun-fm-trusted-providers` 속성의 값으로 포함합니다(두 엔티티 ID를 쉼표(,)로 구분).**

   b. **`sp-extended.xml` 파일에서 새 트러스트 그룹 이름을 `cotlist` 속성의 값에 추가합니다. 예를 들어 `cot2`라는 트러스트 그룹의 경우 다음을 수행합니다.**

   ```
   <Attribute name="cotlist">
   <Value>saml2cot</Value>
   <Value>cot2</Value>
   </Attribute>
   ```

5. **응용 프로그램의 `App_Data` 폴더에서 새 아이디 공급자에 대한 확장 메타데이터로 새 `idp2-extended.xml` 파일을 만듭니다. 기존 `idp-extended.xml` 파일을 서식 파일로 사용하지만 `entityID`를 새 아이디 공급자 엔티티 ID로 변경합니다. 아이디 공급자에 대한 새 트러스트 그룹이 만들어진 경우 `cotlist` 속성의 값을 트러스트 그룹 이름으로 변경합니다. 추가 아이디 공급자는 원격 아이디여야 합니다.**

6. **Fedlet .NET 응용 프로그램과 연결된 응용 프로그램 풀을 다시 시작합니다.**

7 **Fedlet 메타데이터 XML 파일(`sp.xml`)을 추가 아이디 공급자로 가져오고 아이디 공급자 엔티티와 동일한 트러스트 그룹에 추가해야 합니다. `sp.xml` 파일을 아이디 공급자로 가져오거나 아이디 공급자 관리자에게 가져올 파일을 제공합니다.**

아이디 공급자 검색 서비스에 대한 .NET Fedlet 지원(CR 6928524)

이 시나리오에서 .NET Fedlet은 트러스트 그룹의 여러 아이디 공급자로 구성되어 있으며 아이디 공급자 검색 서비스를 사용하여 원하는 아이디 공급자를 확인하도록 Fedlet을 구성하려고 합니다.

.NET Fedlet에서 사용 중인 아이디 공급자에 대해 검색 서비스가 구성되어야 합니다. Oracle OpenSSO 8.0 업데이트 2에서의 아이디 공급자 검색 서비스 구성에 대한 자세한 내용은 `http://docs.sun.com/coll/1767.1`의 문서 모음을 참조하십시오.

▼ 아이디 공급자 검색 서비스를 사용하도록 .NET Fedlet을 구성하는 방법

1 **.NET Fedlet `fedlet.cot` 파일에서 `sun-fm-saml2-readerservice-url` 속성을 SAMLv2 판독기 서비스 URL로 설정합니다. 예:**

```
sun-fm-saml2-readerservice-url=http://discovery.common.com/opensso/saml2reader
```

2 **.NET Fedlet 응용 프로그램과 연결된 응용 프로그램 풀을 다시 시작합니다.**

Oracle OpenSSO Fedlet에 대한 일반 문제 및 해결 방법

개발할 항목

설명서 오류 정보

Fedlet Java API 참조는 Oracle OpenSSO 8.0 업데이트 2 Java API 참조에서 제공되며 `http://docs.sun.com/coll/1767.1`의 문서 모음에서 찾아볼 수 있습니다.

주 – `getPolicyDecisionForFedlet` 메소드는 OpenSSO 8.0 업데이트 2 릴리스에서 지원되지 않습니다.

◆ ◆ ◆ 5 장

5 OpenSSO 8.0 업데이트 2와 Oracle Access Manager 통합

이 장에서는 OpenSSO 8.0 업데이트 2 및 Oracle Access Manager 10g 또는 11g를 사용하여 단일 사인 온(SSO)을 구현하기 위한 지시사항을 제공합니다. 이 정보는 **Sun OpenSSO Enterprise 8.0 Integration Guide**의 3 장, "Integrating Oracle Access Manager"에 포함된 개념 정보를 보충합니다. 이 사용 사례는 Oracle Access Manager 세션을 통해 OpenSSO 보호 응용 프로그램에 대한 단일 사인 온(SSO) 경험을 제공합니다. 구성된 OpenSSO 인증 모듈은 Oracle Access Manager 세션을 기반으로 한 OpenSSO 세션을 생성합니다.

통합 단계 개요

1. 59 페이지 "시작하기 전에"
2. Unpacking the Integration Bits
3. Building source files for Oracle Access Manager in OpenSSO
4. 63 페이지 "(선택 사항) Oracle Access Manager에서 OpenSSO용 인증 체계 빌드"
5. 64 페이지 "Oracle Access Manager 및 Oracle OpenSSO STS를 사용하여 단일 사인 온(SSO) 구성"
6. 66 페이지 "단일 사인 온(SSO)을 테스트하는 방법"
7. 66 페이지 "(선택 사항) Oracle Access Manager에 Oblix AuthScheme 설치"

시작하기 전에

Oracle Access Manager와 통합을 위해 OpenSSO 8.0 업데이트 2를 설치하려면 다음 구성요소에 액세스할 수 있어야 합니다.

opensso.zip	이 zip 파일에는 OpenSSO 8.0 업데이트 2 설치 및 구성에 필요한 opensso.war 파일, 통합 소스 코드, 구성 파일 및 기타 도구가 포함되어 있습니다.

OpenSSO 에이전트	OpenSSO 에이전트는 OpenSSO가 보호하는 응용 프로그램이 실제로 Oracle Access Manager가 설정한 인증 세션을 사용할 수 있을 때 사용됩니다.
Oracle Access Manager 10g 또는 11g	Oracle 웹 사이트에서 Oracle Access Manager를 다운로드합니다. Oracle Fusion Middleware 11gR1 소프트웨어 다운로드 페이지를 참조하십시오.
Oracle Web Gate 10g 또는 11g	OpenSSO와 Oracle Webgate 모두에서 지원되는 컨테이너를 위한 Oracle Webgate를 다운로드합니다. 현재 Sun Web Server 7.x는 두 제품에서 모두 지원되는 유일한 컨테이너입니다. Oracle Fusion Middleware 11gR1 소프트웨어 다운로드 페이지를 참조하십시오.
Oracle Access Manager SDK 10g 또는 11g	Oracle Access Manager를 다운로드합니다. Oracle Access Manager 통합을 위해 OpenSSO 인증 모듈을 컴파일 및 빌드하려면 SDK가 필요합니다. Oracle Fusion Middleware 11gR1 소프트웨어 다운로드 페이지를 참조하십시오.
OpenSSO C-SDK 2.2	(선택 사항) Oracle Access Manager 자체에서 인증 모듈을 만들어서 OAM 세션을 생성하려면 OpenSSO C-SDK가 필요합니다. 이것은 OpenSSO 관점에서의 일반 사용 사례가 아닐 수 있습니다. **Sun OpenSSO Enterprise 8.0 C API Reference for Application and Web Policy Agent Developers**의 "Where is the C SDK?"를 참조하십시오.

통합 비트 압축 풀기

`opensso/integrations/oracle` 디렉토리에는 사용자 정의 인증 모듈 및 기타 플러그인을 컴파일 및 빌드하는 소스 및 구성이 포함되어 있습니다. 사용 사례 옵션 및 관련 정보는 **Sun OpenSSO Enterprise 8.0 Integration Guide**의 3 장, "Integrating Oracle Access Manager"을 참조하십시오. 다음 표에는 `opensso/integrations/oracle` 디렉토리에 있는 파일과 각 파일에 대한 설명이 요약되어 있습니다.

`README.html`	이것은 지금 읽고 있는 파일입니다.

`build.xml`	OpenSSO에서 Oracle Access Manager용 사용자 정의 인증 모듈을 빌드하기 위한 ant 빌드 파일
`config`	OpenSSO에서 Oracle Access Manager용 인증 모듈을 만드는 데 필요한 구성 파일 ■ `OblixAuthService.xml` Oracle Access Manager 인증 모듈에 대한 인증 서비스 파일 ■ `OblixAuthModule.xml` Oracle Access Manager에 대한 인증 모듈 콜백 기본적으로 이것은 빈 파일이지만 구성을 위해 있어야만 합니다. ■ `OblixAuth.properties` 인증을 위한 국제화 키를 저장하는 등록 정보 파일
`lib`	이 디렉토리는 기본적으로 비어 있습니다. 소스 라이브러리를 컴파일하려면 이 `lib` 디렉토리에 다음 라이브러리가 포함되어 있어야 합니다. ■ `jobaccess.jar` Oracle Access Manager SDK에서 이 파일을 복사합니다. ■ `openfedlib.jar`, `amserver.jar` 및 `opensso-sharedlib.jar` opensso.war에서 이 파일을 복사합니다. ■ `servlet.jar` **또는** `javaee.jar` GlassFish `lib` 디렉토리를 복사합니다. `javax.servlet.http.Cookie`와 같은 표준 Java EE 클래스가 있는 JAR 파일이 가장 적합합니다.
`source`	다음 소스 파일이 포함된 디렉토리: ■ `com/sun/identity/authentication/oblix/OblixAuthModule.java` ■ `com/sun/identity/authentication/oblix/OblixAuthModule.java` ■ `com/sun/identity/authentication/oblix/OblixPrincipal.java` ■ `com/sun/identity/saml2/plugins/OAMAdapter.java`

이 클래스는 SAML 서비스 공급자용 SAML2 플러그인 어댑터입니다. 이 클래스는 OpenSSO 세션 서비스를 사용하여 Oracle Access Manager에 대한 원격 인증을 수행합니다.

oamauth(선택 사항) 이 디렉토리에는 OpenSSO용 Oblix 인증 체계에 대한 소스 파일이 포함되어 있습니다. 이것은 C 기반 인증 모듈이며 유효성 검사를 위해 OpenSSO C-SDK를 활용합니다.

- oam/solaris/authn_api.c

  이 파일은 OpenSSO의 Oblix 사용자 정의 인증 체계를 구현합니다.

- oam/solaris/include/*.h

  인증 체계를 컴파일하는 데 필요한 모든 헤더 파일

- oam/solaris/AMAgent.properties

  샘플 OpenSSO 에이전트 구성 파일. 이것은 인증 체계가 OpenSSO 세션의 유효성을 검사하는 데 필요합니다.

OpenSSO에서 Oracle Access Manager용 소스 파일 빌드

ant 스크립트를 사용하여 소스 파일을 빌드합니다. PATH에 호환 가능한 ant 스크립트를 설치하고 구성해야 합니다.

▼ Oracle Access Manager용 소스 파일을 빌드하는 방법

1 다음 명령을 실행합니다.

```
cd $openssozipdir/integrations/oracle; ant -f build.xml
```

이 명령은 소스 파일을 빌드하고 `$openssozipdir/integrations/oracle/dist` 디렉토리에 `fam_oam_integration.jar`을 생성합니다.

2 인증 모듈을 OpenSSO WAR 파일에 번들로 묶습니다.

a. 임시 디렉토리를 만들고 `opensso.war`을 압축 해제합니다. 예:

```
# mkdir /export/tmp
# cd /export/tmp
# jar -xvf opensso.war
```

이제부터 `/export/tmp`는 WAR 스테이징 영역으로 사용되며 `$WAR_DIR` 매크로로 표시됩니다.

b. **$openssozipdir/integrations/oracle/dist/fam_oam_integration.jar을 $WAR_DIR/WEB-INF/lib로 복사합니다.**

c. **$openssozipdir/integrations/oracle/config/OblixAuth.properties를 $WAR_DIR/WEB-INF/classes로 복사합니다.**

d. **$openssozipdir/integrations/oracle/config/OblixAuthModule.xml을 $WAR_DIR/config/auth/default 및 $WAR_DIR/config/auth/default_en 디렉토리에 복사합니다.**

e. **$WAR_DIR에서 jar cvf opensso.war을 사용하여 opensso.war을 다시 압축합니다.**

예제 TBD

(선택 사항) Oracle Access Manager에서 OpenSSO용 인증 체계 빌드

참고: 이것은 일반 사용 사례가 아닙니다. SAML2 서비스 공급자 사용 사례에서처럼 필수가 아니라면 이를 빌드할 필요가 없습니다.

Oblix 인증 체계를 빌드하려면 `makefile`을 사용자 정의해야 합니다. 또한 이것은 C 기반 인증 모듈이므로 운영 체제에 따라 달라집니다.

▼ Oracle Access Manager에서 OpenSSO에 대한 인증 체계를 빌드하는 방법

시작하기 전에 인증 체계 파일은 `$openssozipdir/integrations/oracle/oamauth/solaris` 디렉토리 아래에 있습니다.

1 **OpenSSO C-SDK 2.2 버전을 다운로드 및 구성합니다.**

`authn_api.c` 파일에는 `AMAgent.propeties` 파일에 대한 참조가 포함되어 있습니다. 파일을 적절히 수정합니다.

2 **환경에 맞게 `makefile`을 사용자 정의합니다.**

예를 들어 `gcc` 컴파일 위치를 지정합니다. 또한 `LDFLAGS`가 OpenSSO C-SDK `lib` 디렉토리를 가리키도록 편집합니다.

3 **`make` 명령을 실행합니다.**

`make` 명령은 `authn_api.so` 파일을 생성합니다.

Oracle Access Manager 및 Oracle OpenSSO STS를 사용하여 단일 사인 온(SSO) 구성

▼ Oracle Access Manager 및 Oracle OpenSSO 8.0 업데이트 2를 사용하여 단일 사인 온(SSO)을 구성하는 방법

시작하기 전에: Sun Java System Web Server 7.x가 설치되고 구성되어 있어야 합니다. 웹 서버 설치 지시사항은 Sun Java System Web Server 설명서 Wiki를 참조하십시오.

1. **Sun Java System Web Server 7.x에 OpenSSO를 설치합니다.**

2. **지원되는 컨테이너에 OpenSSO 정책 에이전트를 설치하고 OpenSSO와 함께 작동하도록 에이전트를 구성합니다.**

   설치 지시사항은 **Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for J2EE Agents** 또는 **Sun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web AgentsSun OpenSSO Enterprise Policy Agent 3.0 User's Guide for Web Agents**를 참조하십시오.

3. **Oracle Access Manager를 설치 및 구성합니다.**

   Oracle Access Manager 설치 설명서 10g(10.1.4.3)를 참조하십시오.

4. **Oracle Access Manager가 포함된 Oracle Access Manager SDK를 설치 및 구성합니다.**

   Oracle Access Manager 설치 설명서 10g(10.1.4.3)를 참조하십시오.

5. **OpenSSO 서버가 설치된 것과 동일한 웹 컨테이너에 Oracle Webgate를 설치합니다. (Sun Web Server 7.x)**

   OpenSSO 웹 응용 프로그램의 `deployURI/UI/*`만 보호하도록 OpenSSO를 구성합니다. 예: `/opensso/UI/.../*`

   Oracle Access Manager 정책, 자원 및 기타 구성 세부 정보에 대해서는 Oracle Access Manager 관리 설명서를 참조하십시오. OpenSSO Enteprise에서 다른 모든 URL의 보호를 해제합니다. 이것은 간단한 단일 사인 온(SSO) 통합 시나리오에 대한 것이지만 전체 통합 및 기타 배포 종속성을 기준으로 정책을 평가합니다.

6 OpenSSO에서 인증 모듈을 구성합니다.

a. OpenSSO 콘솔에 액세스합니다.

브라우저가 인증을 위해 Oracle Access Manager로 리디렉션됩니다. 인증에 성공하면 OpenSSO의 로그인 페이지가 나타납니다. OpenSSO 관리자 사용자 이름과 비밀번호를 사용하여 로그인합니다.

b. Oracle 인증 모듈 서비스 XML 파일을 OpenSSO 구성으로 가져옵니다.

인증 모듈 서비스는 명령줄 `ssoadm` 유틸리티뿐 아니라 브라우저 기반 ssoadm.jsp에서 로드될 수 있습니다.

c. `http://host:port/opensso/ssoadm.jsp`에 액세스합니다.

d. create-service 옵션을 선택합니다.

e. `$openssozipdir/integrations/oracle/config/OblixAuthService.xml`에서 XML 파일을 복사하여 붙여넣고 제출을 클릭합니다.

이를 통해 인증 모듈 서비스가 OpenSSO 구성에 로드됩니다.

f. 인증 모듈을 인증 핵심 서비스에 등록합니다.

핵심 서비스에는 인증자 목록이 포함되어 있습니다.
`http://host:port/opensso/ssoadm.jsp`에서 `register-auth-module` 옵션을 선택합니다. `com.sun.identity.authentication.oblix.OblixAuthModule`을 인증 모듈 클래스 이름으로 입력합니다.

g. 인증 모듈이 기본 영역에 등록되었는지 확인합니다.

http://host:port/opensso URL을 사용하여 OpenSSO에 액세스합니다. OpenSSO 콘솔에서 기본 영역을 클릭한 다음 인증 탭을 클릭합니다. 새로 만들기를 클릭하여 OblixAuth라는 새 인증 모듈을 만듭니다.

h. 인증 탭에서 OblixAuth 인증 모듈을 선택합니다.

Oblix SDK 디렉토리를 구성합니다. 원격 사용자 헤더만 확인을 사용 설정하고 원격 헤더 이름을 `OAM_REMOTE_USER`로 지정합니다. 이 매개 변수는 배포를 기반으로 하여 구성될 수 있습니다.

7 (선택 사항) OpenSSO 핵심 인증 서비스에서 프로필 무시 옵션을 사용하도록 설정합니다.

OpenSSO 콘솔에서 구성 > 핵심 > 영역 속성 > 사용자 프로필로 이동합니다. 무시됨을 선택한 다음 저장을 클릭합니다.

이 구성은 인증이 성공적으로 수행된 후 OpenSSO가 기존 사용자 프로필을 검색하지 못하도록 합니다. 하지만 OpenSSO와 Oracle Access Manager가 똑같은 사용자 저장소를

사용하는 경우에는 이 단계가 필요 없습니다. 관리 콘솔 -> 구성 -> 핵심 -> 영역 속성 -> 사용자 프로필로 이동합니다. 무시됨을 선택한 다음 저장을 클릭합니다.

8 **Oracle Access Manager SDK 공유 라이브러리가 포함되도록 웹 서버 시작 스크립트를 편집합니다.**

`$ACCESSDKDIR/oblix/lib`의 공유 라이브러리를 포함하도록 `startserv` 스크립트에서 `LD_LIBRARY_PATH`를 업데이트합니다.

9 **OpenSSO와 Oracle Webgate를 모두 포함하는 Sun 웹 서버를 다시 시작합니다.**

10 **웹 에이전트의 로그인 URL 값을 `http://openssohost:openssoport/deployURI/UI/Login?module=OblixAuth`로 업데이트합니다.**

단일 사인 온(SSO)을 테스트하는 방법

OpenSSO 보호 응용 프로그램의 보호된 자원에 액세스합니다. 아직 인증되지 않은 경우 브라우저가 Oracle Access Manager 로그인 페이지로 리디렉션됩니다. 로그인에 성공하면 OpenSSO 세션이 만들어지고 마지막으로 정책 에이전트 보호 응용 프로그램 URL로 다시 리디렉션됩니다. 정책에 따라 보호된 응용 프로그램에 대한 액세스가 허용 또는 거부됩니다.

(선택 사항) Oracle Access Manager에 Oblix AuthScheme 설치

이것은 OpenSSO 세션의 유효성 검사 시 Oracle Access Manager 세션을 생성해야 하는 경우 유용합니다. 관련 사용 사례에 대한 자세한 내용은 **Sun OpenSSO Enterprise 8.0 Integration Guide**의 3 장, “Integrating Oracle Access Manager”을 참조하십시오.

Oblix 인증 체계는 C 인증 모듈로서 표시되고 이 인증 체계는 OpenSSO C-SDK 2.2 버전을 사용하여 OpenSSO 세션의 유효성을 검사합니다. Oblix의 OpenSSO 인증 체계는 `AMAgent.properties`의 OpenSSO 클라이언트측 구성에 대한 구성을 사용합니다. 인증 모듈을 구성하기 전에 이 파일을 사용자 정의해야 합니다. 빌드 지시사항에 이 파일의 위치가 나와 있습니다. 인증 체계를 구성하기 전, 컴파일된 `authn_api.so` 및 기타 C-SDK 라이브러리를 `$OAM_INSTALL_DIR/access/oblix/lib` 디렉토리로 복사해야 합니다. **Sun OpenSSO 8.0 통합 설명서**는 Oracle 인증 체계를 구성하는 방법을 설명하는 샘플 스크린샷을 보여줍니다. 이것은 참조용으로만 사용되어야 합니다. 자세한 내용은 최신 Oracle Access Manager 설명서를 참조하십시오.

OpenSSO 8.0 업데이트 2와 Oracle Access Manager 통합

이 섹션에는 OpenSSO 8.0 업데이트 2 및 Oracle Access Manager 버전 10.1.4.0.1 또는 11g를 사용한 단일 사인 온(SSO) 구현에 대한 지시사항이 제공됩니다. 이 정보는 **Sun OpenSSO Enterprise 8.0 Integration Guide**의 3 장, "Integrating Oracle Access Manager"에 포함된 개념 정보를 보충합니다. 이 사용 사례는 Oracle Access Manager 세션을 통해 OpenSSO 보호 응용 프로그램에 대한 단일 사인 온(SSO) 경험을 제공합니다. 구성된 OpenSSO 인증 모듈은 Oracle Access Manager 세션을 기반으로 한 OpenSSO 세션을 생성합니다.