



Red Hat Enterprise Linux 9

IdM의 인증서 관리

인증서 발행, 인증서 기반 인증 구성, Red Hat Enterprise Linux 9의 ID 관리에서 인증서
유효성 제어

Red Hat Enterprise Linux 9 IdM의 인증서 관리

인증서 발행, 인증서 기반 인증 구성, Red Hat Enterprise Linux 9의 ID 관리에서 인증서 유효성 제어

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

법적 공지

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Managing_certificates_in_IdM.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution-Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

초록

이 문서에서는 IdM 인증 기관에서 발급한 인증서 관리, 인증에 인증서를 사용하도록 사용자 계정 구성, Red Hat Enterprise Linux 9의 Identity Management의 인증서 유지 관리에 대해 설명합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	6
RED HAT 문서에 관한 피드백 제공	7
1장. IDENTITY MANAGEMENT의 공개 키 인증서	8
1.1. IDM의 인증 기관	8
1.2. 인증서 및 KERBEROS 비교	9
1.3. IDM에서 사용자를 인증하기 위해 인증서 사용 및 장단점	9
2장. 통합 IDM CA를 사용하여 사용자, 호스트 및 서비스의 인증서 관리	11
2.1. IDM 웹 UI를 사용하여 사용자, 호스트 또는 서비스에 대한 새 인증서 요청	11
2.2. CERTUTIL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청	12
2.3. OPENSSEL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청	13
2.4. 추가 리소스	15
3장. IDM 사용자, 호스트 및 서비스용 외부 서명된 인증서 관리	16
3.1. IDM CLI를 사용하여 외부 CA에서 발급한 인증서 추가, IDM 사용자, 호스트 또는 서비스	16
3.2. IDM 웹 UI를 사용하여 IDM 사용자, 호스트 또는 서비스에 외부 CA에서 발급한 인증서 추가	16
3.3. IDM CLI를 사용하여 IDM 사용자, 호스트 또는 서비스 계정에서 외부 CA에서 발급한 인증서 제거	17
3.4. IDM 웹 UI를 사용하여 IDM 사용자, 호스트 또는 서비스 계정에서 외부 CA에서 발급한 인증서 제거	18
3.5. 추가 리소스	19
4장. IDM에서 작동하도록 인증서 형식 변환	20
4.1. IDM의 인증서 형식 및 인코딩	20
4.2. 외부 인증서를 IDM 사용자 계정으로 로드하도록 변환	22
4.2.1. 사전 요구 사항	22
4.2.2. IdM CLI에서 외부 인증서를 변환하여 IdM 사용자 계정으로 로드합니다.	22
4.2.3. IdM 사용자 계정으로 로드하기 위해 IdM 웹 UI에서 외부 인증서를 변환	24
4.3. 브라우저에 인증서 로드 준비	25
4.3.1. NSS 데이터베이스에서 인증서 및 개인 키를 PKCS #12 파일로 내보내기	25
4.3.2. 인증서 및 개인 키 PEM 파일을 PKCS #12 파일로 결합	26
4.4. IDM의 인증서 관련 명령 및 형식	26
5장. IDENTITY MANAGEMENT에서 인증서 프로필 생성 및 관리	28
5.1. 인증서 프로필이란 무엇입니까?	28
5.2. 인증서 프로파일 생성	29
5.3. CA 액세스 제어 목록이란 무엇입니까?	31
5.4. 인증서 프로필에 대한 액세스를 제어하기 위한 CA ACL 정의	32
5.5. 인증서 프로필 및 CA ACL을 사용하여 인증서 발급	34
5.6. 인증서 프로필 수정	36
5.7. 인증서 프로필 구성 매개변수	37
6장. IDM의 인증서 유효성 관리	41
6.1. IDM CA에서 발급한 기존 인증서의 유효성 관리	41
6.2. IDM CA에서 발행한 향후 인증서의 유효성 관리	42
6.3. IDM WEBUI에서 인증서 만료 날짜 보기	42
6.4. CLI에서 인증서 만료일 보기	43
6.5. 통합 IDM CA로 인증서 해지	43
6.5.1. 인증서 해지 이유	43
6.5.2. IdM WebUI를 사용하여 통합 IdM CA에서 인증서 취소	44
6.5.3. IdM CLI를 사용하여 통합 IdM CA에서 인증서 취소	45
6.6. 통합 IDM CA의 인증서 복원	46
6.6.1. IdM WebUI를 사용하여 통합 IdM CA의 인증서 복원	46

6.6.2. IdM CLI를 사용하여 통합 IdM CA의 인증서 복원	47
7장. 스마트 카드 인증을 위한 ID 관리 구성	48
7.1. 스마트 카드 인증을 위한 IDM 서버 구성	49
7.2. 스마트 카드 인증을 위한 IDM 클라이언트 구성	52
7.3. IDM 웹 UI에서 사용자 항목에 인증서 추가	54
7.4. IDM CLI에서 사용자 항목에 인증서 추가	56
7.5. 스마트 카드 관리 및 사용을 위한 도구 설치	57
7.6. 스마트 카드에 인증서 저장	58
7.7. 스마트 카드를 사용하여 IDM 로그인	61
7.8. 스마트 카드 인증을 사용하여 GDM 액세스 구성	63
7.9. 스마트 카드 인증을 사용하여 SU 액세스 구성	64
8장. IDM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서 구성	65
8.1. 스마트 카드 인증	66
8.2. 신뢰 구성 및 인증서 사용에 필요한 WINDOWS SERVER 설정	67
8.3. SFTP를 사용하여 ACTIVE DIRECTORY에서 인증서 복사	67
8.4. ADCS 인증서를 사용하여 스마트 카드 인증을 위한 IDM 서버 및 클라이언트 구성	68
8.5. PCABUNDLE 파일 변환	71
8.6. 스마트 카드 관리 및 사용을 위한 도구 설치	72
8.7. 스마트 카드에 인증서 저장	72
8.8. SSSD.CONF에서 시간 제한 설정	75
8.9. 스마트 카드 인증을 위한 인증서 매핑 규칙 생성	76
9장. IDENTITY MANAGEMENT에서 인증서 매핑 규칙 구성	78
9.1. 스마트 카드에서 인증을 구성하기 위한 인증서 매핑 규칙	78
9.1.1. Active Directory 도메인을 사용하는 신뢰에 대한 인증서 매핑 규칙	79
9.1.2. IdM의 ID 매핑 규칙 구성 요소	79
9.1.3. 일치하는 규칙에 사용할 인증서에서 발급자 가져오기	81
9.1.4. 추가 리소스	82
9.2. IDM에 저장된 사용자의 인증서 매핑 구성	82
9.2.1. IdM 웹 UI에 인증서 매핑 규칙 추가	83
9.2.2. IdM CLI에 인증서 매핑 규칙 추가	84
9.2.3. IdM 웹 UI에서 사용자 항목에 인증서 매핑 데이터 추가	85
9.2.4. IdM CLI에서 사용자 항목에 인증서 매핑 데이터 추가	87
9.3. AD 사용자 항목이 전체 인증서가 포함되어 있는 사용자에게 대한 인증서 매핑 구성	88
9.3.1. IdM 웹 UI에 인증서 매핑 규칙 추가	89
9.3.2. IdM CLI에 인증서 매핑 규칙 추가	90
9.4. AD가 사용자 인증서를 사용자 계정에 매핑하도록 구성된 경우 인증서 매핑 구성	91
9.4.1. IdM 웹 UI에 인증서 매핑 규칙 추가	91
9.4.2. IdM CLI에 인증서 매핑 규칙 추가	92
9.4.3. AD 측에서 인증서 매핑 데이터 확인	93
9.5. AD 사용자 항목에 인증서가 없거나 매핑 데이터가 없는 경우 인증서 매핑 구성	94
9.5.1. IdM 웹 UI에 인증서 매핑 규칙 추가	94
9.5.2. IdM CLI에 인증서 매핑 규칙 추가	96
9.5.3. IdM 웹 UI에서 AD 사용자 ID 재정의에 인증서 추가	96
9.5.4. IdM CLI에서 AD 사용자 ID 재정의에 인증서 추가	98
9.6. 여러 ID 매핑 규칙을 하나로 결합	99
10장. IDM 클라이언트의 데스크탑에 저장된 인증서로 인증 구성	102
10.1. 웹 UI에서 인증 인증을 위한 IDENTITY MANAGEMENT SERVER 구성	102
10.2. 새 사용자 인증서 요청 및 클라이언트로 내보내기	103
10.3. 인증서와 사용자가 연결되어 있는지 확인	105
10.4. 인증서 인증을 사용하도록 브라우저 구성	106

10.5. IDENTITY MANAGEMENT USER로 인증서를 사용하여 IDENTITY MANAGEMENT 웹 UI에 인증	110
10.6. 인증서를 사용하여 CLI에 인증할 수 있도록 IDM 클라이언트 구성	111
11장. IDM CA 갱신 서버 사용	113
11.1. IDM CA 갱신 서버에 대한 설명	113
11.2. IDM CA 갱신 서버 변경 및 재설정	115
11.3. IDM에서 자체 서명된 CA로 외부에서 전환	117
11.4. 외부 서명 인증서로 IDM CA 갱신 서버 갱신	118
12장. IDM이 오프라인 상태일 때 만료된 시스템 인증서 갱신	122
12.1. CA 갱신 서버에서 만료된 시스템 인증서 갱신	122
12.2. 갱신 후 IDM 도메인의 다른 IDM 서버 확인	123
12.3. 웹 서버 및 LDAP 서버 인증서 교체	125
13장. IDM CA 서버에서 CRL 생성	128
13.1. IDM 서버에서 CRL 생성 중지	128
13.2. IDM 복제본 서버에서 CRL 생성 시작	129
14장. CA 갱신 서버 및 CRL 게시자 역할을 수행하는 서버 해제	131
15장. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기	135
15.1. CERTMONGER 개요	135
15.2. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기	137
15.3. 서비스 인증서를 요청하는 CERTMONGER의 통신 흐름	139
15.4. CERTMONGER에서 추적한 인증서 요청의 세부 정보 보기	143
15.5. 인증서 추적 시작 및 중지	145
15.6. 인증서를 수동으로 업데이트	146
15.7. CERTMONGER는 CA 복제본에서 IDM 인증서 추적을 재개합니다.	147
16장. RHEL 시스템 역할을 사용하여 인증서 요청	149
16.1. 인증서 문제 및 업데이트 시스템 역할	149
16.2. 인증서 문제 및 업데이트 시스템 역할을 사용하여 자체 서명된 새 인증서 요청	150
16.3. 인증서 문제 및 업데이트 시스템 역할을 사용하여 IDM CA에서 새 인증서 요청	152
16.4. 인증서 발행 및 업데이트 시스템 역할을 사용하여 인증서 발행 전 또는 이후에 실행할 명령 지정	154
17장. 인증서 서브 세트만 신뢰하도록 애플리케이션 제한	157
17.1. 경량 하위 CA 관리	158
17.1.1. IdM 웹 UI에서 하위 CA 생성	159
17.1.2. IdM 웹 UI에서 하위 CA 삭제	160
17.1.3. IdM CLI에서 하위 명령 생성	162
17.1.4. IdM CLI에서 하위 명령 비활성화	164
17.1.5. IdM CLI에서 하위 명령 삭제	165
17.2. IDM WEBUI에서 하위 CA 인증서 다운로드	167
17.3. 웹 서버 및 클라이언트 인증에 대한 CA ACL 생성	167
17.3.1. IdM CLI에서 CA ACL 보기	167
17.3.2. webserver-ca에서 발행한 인증서를 사용하여 웹 클라이언트에 인증하는 웹 서버에 대한 CA ACL 생성	168
17.3.3. webclient-ca에서 발행한 인증서를 사용하여 웹 서버에 인증하는 사용자 웹 브라우저의 CA ACL 생성	170
17.4. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기	172
17.5. 서비스 인증서를 요청하는 CERTMONGER의 통신 흐름	174
17.6. 단일 인스턴스 APACHE HTTP SERVER 설정	178
17.7. APACHE HTTP SERVER에 TLS 암호화 추가	179
17.8. APACHE HTTP SERVER에서 지원되는 TLS 프로토콜 버전 설정	182
17.9. APACHE HTTP SERVER에서 지원되는 암호 설정	184
17.10. TLS 클라이언트 인증서 인증 구성	185
17.11. 새 사용자 인증서 요청 및 클라이언트로 내보내기	187

17.12. 인증서 인증을 사용하도록 브라우저 구성	189
18장. 특정 관련 인증서 그룹 무효화를 신속하게	192
18.1. IDM CLI에서 CA ACL 비활성화	192
18.2. IDM 하위 명령 비활성화	194
19장. IDM 상태 점검을 사용하여 인증서 확인	195
19.1. IDM 인증서 상태 점검 테스트	195
19.2. HEALTHCHECK 도구를 사용하여 인증서 확인	197
20장. IDM 상태 점검을 사용하여 시스템 인증서 확인	199
20.1. 시스템 인증서 상태 점검 테스트	199
20.2. HEALTHCHECK를 사용하여 시스템 인증서 확인	200

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오.

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
 1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **피드백** 버튼이 있는지 확인합니다.
 2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
 3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
 4. 표시된 지침을 따릅니다.
- Bugzilla를 통해 피드백을 제출하려면 새 티켓을 생성합니다.
 1. [Bugzilla](#) 웹 사이트로 이동합니다.
 2. 구성 요소로 **문서**를 사용합니다.
 3. **설명** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
 4. **버그 제출**을 클릭합니다.

1장. IDENTITY MANAGEMENT의 공개 키 인증서

이 장에서는 IdM(Identity Management)에서 사용자, 호스트 및 서비스를 인증하는 데 사용되는 X.509 공개 키 인증서에 대해 설명합니다. 인증 외에도 X.509 인증서를 사용하면 디지털 서명 및 암호화를 통해 개인 정보 보호, 무결성 및 비회전을 제공할 수 있습니다.

인증서에는 다음 정보가 포함됩니다.

- 인증서가 인증되는 주체입니다.
- 인증서에 서명한 CA인 발급자입니다.
- 인증서 유효성의 시작 및 종료일입니다.
- 인증서의 유효한 사용입니다.
- 주체의 공개 키입니다.

공개 키로 암호화된 메시지는 해당 개인 키로만 암호를 해독할 수 있습니다. 인증서와 공개 키를 공개적으로 사용할 수 있지만 사용자, 호스트 또는 서비스는 개인 키 시크릿을 유지해야 합니다.

1.1. IDM의 인증 기관

인증 기관은 신뢰 계층에서 작동합니다. 내부 인증 기관(CA)이 있는 IdM 환경에서 모든 IdM 호스트, 사용자 및 서비스는 CA에서 서명한 인증서를 신뢰합니다. 이 루트 CA 외에도 IdM은 루트 CA에서 인증서 서명 기능을 부여한 하위 CA를 지원합니다. 이러한 하위 CA가 서명할 수 있는 인증서는 VPN 인증서(예: VPN 인증서)의 인증서입니다. 마지막으로 IdM은 외부 CA 사용을 지원합니다. [아래](#) 표는 IdM에서 개별 유형의 CA를 사용하는 세부 사항을 보여줍니다.

표 1.1. IdM에서 통합 및 외부 CA 사용 비교

CA 이름	설명	사용	유용한 링크
ipa CA	Dogtag 업스트림 프로젝트 기반의 통합 CA	통합 CA는 사용자, 호스트 및 서비스에 대한 인증서를 생성, 취소 및 발행할 수 있습니다.	ipa CA를 사용하여 새 사용자 인증서를 요청하고 클라이언트로 내보내기
IdM 하위 CA	ipa CA에 종속된 통합 CA	IdM 하위 CA는 ipa CA에서 인증서에 서명할 수 있는 CA입니다. 종종 이러한 인증서는 특정 유형(예: VPN 인증서)입니다.	인증서 서브 세트만 신뢰하도록 애플리케이션 제한
외부 CA	외부 CA는 통합 IdM CA 또는 해당 하위 CA 이외의 CA입니다.	IdM 툴을 사용하여 이러한 CA에서 발행한 인증서를 사용자, 서비스 또는 호스트에 추가하고 이를 제거합니다.	IdM 사용자, 호스트 및 서비스용 외부 서명된 인증서 관리

인증서 보기에서는 자체 서명된 IdM CA에서 서명하고 외부에서 서명할 때 차이가 없습니다.

CA의 역할에는 다음과 같은 용도가 포함됩니다.

- 디지털 인증서를 발급합니다.

- 인증서에 서명하면 인증서에 이름이 지정된 주체가 공개 키가 있음을 인증합니다. 주체는 사용자, 호스트 또는 서비스일 수 있습니다.
- 인증서를 해지할 수 있으며 CRL(Certificate Revocation Lists) 및 OCSCSP(Online Certificate Status Protocol)를 통해 해지 상태를 제공합니다.

추가 리소스

- [CA 서비스 계획](#)을 참조하십시오.

1.2. 인증서 및 KERBEROS 비교

인증서는 Kerberos 티켓에서 수행하는 것과 유사한 기능을 수행합니다. Kerberos는 안전하지 않은 네트워크를 통해 통신하는 노드가 안전한 방식으로 서로의 ID를 검증할 수 있도록 티켓 기반으로 작동하는 컴퓨터 네트워크 인증 프로토콜입니다. 다음 표에서는 Kerberos 및 X.509 인증서 비교를 보여줍니다.

표 1.2. 인증서 및 Kerberos 비교

특성	Kerberos	X.509
인증	있음	있음
개인 정보	선택 사항	있음
무결성	선택 사항	있음
관련된 암호화 유형	symmetric	encryptedal
기본 유효성	짧은 (1일)	긴(2년)

기본적으로 Identity Management의 Kerberos는 통신 당사자의 ID만 보장합니다.

1.3. IDM에서 사용자를 인증하기 위해 인증서 사용 및 장단점

IdM에서 사용자를 인증하기 위해 인증서를 사용하는 이점은 다음과 같습니다.

- 스마트 카드의 개인 키를 보호하는 pin은 일반적으로 덜 복잡하고 일반 암호보다 쉽게 기억할 수 있습니다.
- 장치에 따라 스마트 카드에 저장된 개인 키를 내보낼 수 없습니다. 이는 추가 보안을 제공합니다.
- 스마트 카드는 로그아웃을 자동으로 수행할 수 있습니다. IdM은 사용자가 독자에서 스마트 카드를 제거할 때 사용자를 로그아웃하도록 구성할 수 있습니다.
- 개인 키를 도용하려면 스마트 카드에 대한 실제 물리적 액세스가 필요하므로 해킹 공격으로부터 스마트 카드를 보호해야 합니다.
- 스마트 카드 인증은 2단계 인증의 예입니다. 이 인증에는 (카드)와 알 수 있는 내용(Pributional)이 모두 필요합니다.
- 스마트 카드는 암호보다 유연합니다. 이메일 암호화와 같은 다른 용도로 사용할 수 있는 키를 제공하기 때문입니다.

- IdM 클라이언트인 공유 시스템에서 스마트 카드를 사용하면 일반적으로 시스템 관리자에게 추가 구성 문제가 발생하지 않습니다. 실제로 스마트 카드 인증은 공유 머신에 이상적인 옵션입니다.

IdM에서 사용자를 인증하기 위해 인증서를 사용하는 단점은 다음과 같습니다.

- 사용자는 스마트 카드 또는 인증서를 가져 와서 효과적으로 잠길 수 있습니다.
- pin을 여러 번 제거하면 카드가 잠길 수 있습니다.
- 일반적으로 일종의 보안 담당자 또는 승인자의 요청과 승인 사이에 중간 단계가 있습니다. IdM에서 보안 담당자 또는 관리자는 **ipa cert-request** 명령을 실행해야 합니다.
- 스마트 카드와 독자는 공급 업체 및 드라이버별 경향이 있습니다. 독자가 다른 카드에 사용할 수 있지만 특정 공급 업체의 스마트 카드가 다른 공급 업체의 독자 또는 설계되지 않은 독자 유형에서 작동하지 않을 수 있습니다.
- 인증서 및 스마트 카드에는 관리자를 위한 스테이프 학습 곡선이 있습니다.

2장. 통합 IDM CA를 사용하여 사용자, 호스트 및 서비스의 인증서 관리

이 장에서는 통합 CA, **ipa** CA 및 해당 하위 CA를 사용하여 IdM(Identity Management)에서 인증서를 관리하는 방법을 설명합니다.

이 장에서는 다음 섹션을 설명합니다.

- [IdM 웹 UI를 사용하여 사용자, 호스트 또는 서비스에 대한 새 인증서 요청](#) .
- IdM CLI를 사용하여 IdM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서를 요청합니다.
 - [certutil을 사용하여 IdM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청](#)
 - **certutil** 유틸리티를 사용하여 IdM CA에서 새 사용자 인증서를 요청하고 IdM 클라이언트로 내보내는 특정 예는 [새 사용자 인증서 요청 및 클라이언트에 내보내기를 참조하십시오](#).
 - [openssl을 사용하여 IdM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청](#)

certmonger 유틸리티를 사용하여 IdM CA에서 서비스에 대한 새 인증서를 요청할 수도 있습니다. 자세한 내용은 [certmonger를 사용하여 IdM CA에서 서비스에 대한 새 인증서 요청](#) 에서 참조하십시오.

사전 요구 사항

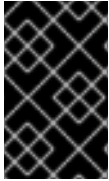
- IdM 배포에 통합 CA가 포함되어 있습니다.
 - IdM에서 CA 서비스를 계획하는 방법에 대한 자세한 내용은 [CA 서비스 계획](#)을 참조하십시오.
 - 통합 DNS 및 통합 CA를 루트 CA로 사용하여 IdM 서버를 설치하는 방법에 대한 자세한 내용은 통합 CA [와 통합된 DNS를 사용하여 IdM 서버](#) 설치를 참조하십시오.
 - 통합된 DNS 및 외부 CA를 사용하여 IdM 서버를 설치하는 방법에 대한 자세한 내용은 [IdM 서버 설치](#)를 참조하십시오. 통합 DNS를 루트 CA로 사용하는 통합 DNS 사용
 - 통합 DNS 및 통합 CA를 루트 CA로 통합하지 않고 IdM 서버를 설치하는 방법에 대한 자세한 내용은 통합 CA [를 루트 CA로 통합 CA와 함께 통합 DNS](#) 설치를 참조하십시오.
 - [선택 사항] IdM 배포는 인증서로 인증하는 사용자를 지원합니다.
 - IdM 클라이언트 파일 시스템에 저장된 인증서를 사용하여 사용자 인증을 지원하도록 IdM 배포를 구성하는 방법에 대한 자세한 내용은 IdM 클라이언트 [데스크탑에 저장된 인증서로 인증 구성](#)을 참조하십시오.
 - IdM 클라이언트에 삽입된 스마트 카드에 저장된 인증서로 사용자 인증을 지원하도록 IdM 배포를 구성하는 방법에 대한 자세한 내용은 [스마트 카드 인증을 위한 ID 관리](#) 구성을 참조하십시오.
 - Active Directory 인증서 시스템에서 발행한 스마트 카드를 사용하여 사용자 인증을 지원하도록 IdM 배포를 구성하는 방법에 대한 자세한 내용은 [IdM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서](#) 구성을 참조하십시오.

2.1. IDM 웹 UI를 사용하여 사용자, 호스트 또는 서비스에 대한 새 인증서 요청

이 섹션에서는 IdM(Identity Management) 웹 UI를 사용하여 통합 IdM 인증 기관(CA): **ipa** CA 또는 그 하위 CA 중 하나에서 IdM 엔터티에 대한 새 인증서를 요청하는 방법을 설명합니다.

IdM 엔티티는 다음과 같습니다.

- 사용자
- 호스트
- 서비스



중요

일반적으로 서비스는 개인 키가 저장되는 전용 서비스 노드에서 실행됩니다. 서비스의 개인 키를 IdM 서버에 복사하는 것은 안전하지 않은 것으로 간주됩니다. 따라서 서비스에 대한 인증서를 요청할 때 서비스 노드에 CSR(인증서 서명 요청)을 만듭니다.

사전 요구 사항

- IdM 배포에는 통합 CA가 포함되어 있습니다.
- IdM 관리자로 IdM 웹 UI에 로그인되어 있습니다.

절차

1. **Identity (ID)** 탭에서 사용자, 호스트 또는 서비스 하위 탭을 선택합니다.
2. 사용자, 호스트 또는 서비스의 이름을 클릭하여 구성 페이지를 엽니다.

그림 2.1. 호스트 목록

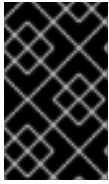
Hosts			
Search		Refresh	Delete
		+ Add	Actions v
☐	Host name	Description	Enrolled
☐	server.example.com		True
Showing 1 to 1 of 1 entries.			

3. **Actions → New Certificate** (새 인증서)를 클릭합니다.
4. 선택 사항: 발행 CA 및 프로필 ID를 선택합니다.
5. 화면에서 **certutil** 명령줄(CLI) 유틸리티를 사용하는 방법에 대한 지침을 따릅니다.
6. **Issue** 를 클릭합니다.

2.2. CERTUTIL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청

certutil 유틸리티를 사용하여 표준 IdM 상황에서 IdM(Identity Management) 사용자, 호스트 또는 서비스에 대한 인증서를 요청할 수 있습니다. 호스트 또는 서비스 Kerberos 별칭이 인증서를 사용할 수 있도록 하려면 **openssl** 유틸리티를 사용하여 대신 인증서를 요청합니다.

이 섹션에서는 **certutil** 을 사용하여 **ipa**, IdM 인증 기관(CA)에서 IdM 사용자, 호스트 또는 서비스에 대한 인증서를 요청하는 방법을 설명합니다.



중요

일반적으로 서비스는 개인 키가 저장되는 전용 서비스 노드에서 실행됩니다. 서비스의 개인 키를 IdM 서버에 복사하는 것은 안전하지 않은 것으로 간주됩니다. 따라서 서비스에 대한 인증서를 요청할 때 서비스 노드에 CSR(인증서 서명 요청)을 만듭니다.

사전 요구 사항

- IdM 배포에는 통합 CA가 포함되어 있습니다.
- IdM 관리자로 IdM 명령줄 인터페이스(CLI)에 로그인되어 있습니다.

절차

1. 인증서 데이터베이스에 대한 임시 디렉터리를 생성합니다.

```
# mkdir ~/certdb/
```

2. 새 임시 인증서 데이터베이스를 만듭니다. 예를 들면 다음과 같습니다.

```
# certutil -N -d ~/certdb/
```

3. CSR을 생성하고 출력을 파일로 리디렉션합니다. 예를 들어 4096비트 인증서에 대한 CSR을 생성하고 이를 `CN=server.example.com,O=EXAMPLE.COM` 으로 설정하려면 다음을 실행합니다.

```
# certutil -R -d ~/certdb/ -a -g 4096 -s "CN=server.example.com,O=EXAMPLE.COM" -8  
server.example.com > certificate_request.csr
```

4. 인증서 요청 파일을 IdM 서버에서 실행 중인 CA에 제출합니다. 새로 발급된 인증서와 연결할 Kerberos 주체를 지정합니다.

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM의 **ipa cert-request** 명령에서는 다음과 같은 기본값을 사용합니다.

- **calPAserviceCert** 인증서 프로파일
사용자 지정 프로필을 선택하려면 **--profile-id** 옵션을 사용합니다.
- 통합 IdM 루트 CA, **ipa**
하위 CA를 선택하려면 **--ca** 옵션을 사용합니다.

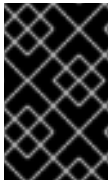
추가 리소스

- **ipa cert-request --help** 명령의 출력을 참조하십시오.
- [Identity Management에서 인증서 프로파일 생성 및 관리](#)를 참조하십시오.

2.3. OPENSSL을 사용하여 IDM CA에서 사용자, 호스트 또는 서비스에 대한 새 인증서 요청

호스트 또는 서비스의 Kerberos 별칭이 인증서를 사용할 수 있도록 하려면 **openssl** 유틸리티를 사용하여 IdM(Identity Management) 호스트 또는 서비스에 대한 인증서를 요청할 수 있습니다. 표준 상황에서는 **certutil** 유틸리티를 사용하여 새 인증서를 요청하는 것이 좋습니다.

이 섹션에서는 **openssl** 을 사용하여 IdM 호스트 또는 **ipa** 에서 IdM 인증 기관의 인증서를 요청하는 방법을 설명합니다.



중요

일반적으로 서비스는 개인 키가 저장되는 전용 서비스 노드에서 실행됩니다. 서비스의 개인 키를 IdM 서버에 복사하는 것은 안전하지 않은 것으로 간주됩니다. 따라서 서비스에 대한 인증서를 요청할 때 서비스 노드에 CSR(인증서 서명 요청)을 만듭니다.

사전 요구 사항

- IdM 배포에는 통합 CA가 포함되어 있습니다.
- IdM 관리자로 IdM 명령줄 인터페이스(CLI)에 로그인되어 있습니다.

절차

1. Kerberos 보안 주체 `test/server.example.com` 에 대해 하나 이상의 별칭을 만듭니다. 예: `test1/server.example.com` 및 `test2/server.example.com`.
2. CSR에서 `dnsName(server.example.com)` 및 `otherName(test2/server.example.com)`에 `subjectAltName`을 추가합니다. 이렇게 하려면 UPN `otherName` 및 `subjectAltName`을 지정하는 다음 행을 포함하도록 **openssl.conf** 파일을 구성하십시오.

```
otherName=1.3.6.1.4.1.311.20.2.3;UTF8:test2/server.example.com@EXAMPLE.COM
DNS.1 = server.example.com
```

3. **openssl** 을 사용하여 인증서 요청을 생성합니다.

```
openssl req -new -newkey rsa:2048 -keyout test2service.key -sha256 -nodes -out
certificate_request.csr -config openssl.conf
```

4. 인증서 요청 파일을 IdM 서버에서 실행 중인 CA에 제출합니다. 새로 발급된 인증서와 연결할 Kerberos 주체를 지정합니다.

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

IdM의 **ipa cert-request** 명령에서는 다음과 같은 기본값을 사용합니다.

- **calPAserviceCert** 인증서 프로파일
사용자 지정 프로필을 선택하려면 **--profile-id** 옵션을 사용합니다.
- 통합 IdM 루트 CA, **ipa**
하위 CA를 선택하려면 **--ca** 옵션을 사용합니다.

추가 리소스

- **ipa cert-request --help** 명령의 출력을 참조하십시오.
- [Identity Management에서 인증서 프로필 생성 및 관리](#)를 참조하십시오.

2.4. 추가 리소스

- [통합 IdM CA로 인증서 해지를 참조하십시오.](#)
- [통합 IdM CA가 있는 인증서 복원을 참조하십시오.](#)
- [인증서 서브 세트만 신뢰하도록 애플리케이션 제한을 참조하십시오.](#)

3장. IDM 사용자, 호스트 및 서비스용 외부 서명된 인증서 관리

이 장에서는 IdM(Identity Management) 명령줄 인터페이스(CLI)와 IdM 웹 UI를 사용하여 외부 인증 기관(CA)에서 발급한 사용자, 호스트 또는 서비스 인증서를 추가하거나 제거하는 방법을 설명합니다.

3.1. IDM CLI를 사용하여 외부 CA에서 발급한 인증서 추가, IDM 사용자, 호스트 또는 서비스

IdM(Identity Management) 관리자는 IdM(Identity Management) CLI를 사용하여 IdM 사용자, 호스트 또는 서비스 계정에 외부 서명된 인증서를 추가할 수 있습니다.

사전 요구 사항

- 관리 사용자의 티켓 개발 티켓이 있습니다.

절차

- IdM 사용자에게 인증서를 추가하려면 다음을 입력합니다.

```
$ ipa user-add-cert user --certificate=MIQTPrajQAwg...
```

명령을 사용하려면 다음 정보를 지정해야 합니다.

- 사용자의 이름
- Base64로 인코딩된 DER 인증서

참고

인증서 내용을 명령줄에 복사하고 붙여넣는 대신 인증서를 DER 형식으로 변환한 다음 Base64로 다시 코딩할 수 있습니다. 예를 들어 **user_cert.pem** 인증서를 **사용자**에 추가하려면 다음을 입력합니다.

```
$ ipa user-add-cert user --certificate="$(openssl x509 -outform der -in user_cert.pem | base64 -w 0)"
```

옵션을 추가하지 않고 **ipa user-add-cert** 명령을 실행할 수 있습니다.

IdM 호스트에 인증서를 추가하려면 다음을 입력합니다.

- **ipa host-add-cert**

IdM 서비스에 인증서를 추가하려면 다음을 입력합니다.

- **ipa service-add-cert**

추가 리소스

- [통합 IdM CA를 사용하여 사용자, 호스트 및 서비스의 인증서 관리](#)

3.2. IDM 웹 UI를 사용하여 IDM 사용자, 호스트 또는 서비스에 외부 CA에서 발급한 인증서 추가

IdM(Identity Management) 관리자는 IdM(Identity Management) 웹 UI를 사용하여 IdM 사용자, 호스트 또는 서비스 계정에 외부 서명된 인증서를 추가할 수 있습니다.

사전 요구 사항

- IdM(Identity Management) 웹 UI에 관리자로 로그인되어 있습니다.

절차

1. **Identity** 탭을 열고 **사용자, 호스트 또는 서비스** 하위 탭을 선택합니다.
2. 사용자, 호스트 또는 서비스의 이름을 클릭하여 구성 페이지를 엽니다.
3. **Certificates** (인증서) 항목 옆에 있는 **Add** (추가)를 클릭합니다.

그림 3.1. 사용자 계정에 인증서 추가

The screenshot shows the IdM web UI for user 'demouser'. The 'Identity Settings' section includes fields for Job Title, First name (Demo), Last name (User), Full name (Demo User), Display name (Demo User), Initials (DU), GECOS (Demo User), and Class. The 'Account Settings' section includes fields for User login (demouser), Password (*****), Password expiration (2016-07-14 10:14:12Z), UID (373000005), GID (373000005), Principal alias (demouser@IDM.EXAMPLE.COM), Kerberos principal expiration (YYYY-MM-DD hh : mn UTC), Login shell (/bin/sh), Home directory (/home/demouser), SSH public keys (Add), and Certificates (Add). The 'Add' button for Certificates is highlighted with a red box.

4. Base64 또는 PEM으로 인코딩된 형식으로 인증서를 붙여넣고 텍스트 필드에 인증서를 붙여넣고 추가를 클릭합니다.
5. **저장**을 클릭하여 변경 사항을 저장합니다.

3.3. IDM CLI를 사용하여 IDM 사용자, 호스트 또는 서비스 계정에서 외부 CA에서 발급한 인증서 제거

IdM(Identity Management) 관리자는 IdM(Identity Management) CLI를 사용하여 IdM 사용자, 호스트 또는 서비스의 계정에서 외부 서명된 인증서를 제거할 수 있습니다.

사전 요구 사항

- 관리 사용자의 티켓 개발 티켓이 있습니다.

절차

- IdM 사용자의 인증서를 제거하려면 다음을 입력합니다.

```
$ ipa user-remove-cert user --certificate=MIQTPrajQAwg...
```

명령을 사용하려면 다음 정보를 지정해야 합니다.

- 사용자의 이름
- Base64로 인코딩된 DER 인증서



참고

인증서 내용을 명령줄에 복사하고 붙여넣는 대신 인증서를 DER 형식으로 변환한 다음 Base64로 다시 코딩할 수 있습니다. 예를 들어 사용자에서 **user_cert.pem** 인증서를 제거하려면 다음을 입력합니다.

```
$ ipa user-remove-cert user --certificate="$(openssl x509 -outform der -in user_cert.pem | base64 -w 0)"
```

옵션을 추가하지 않고 **ipa user-remove-cert** 명령을 대화형으로 실행할 수 있습니다.

IdM 호스트에서 인증서를 제거하려면 다음을 입력합니다.

- **ipa host-remove-cert**

IdM 서비스에서 인증서를 제거하려면 다음을 입력합니다.

- **ipa service-remove-cert**

추가 리소스

- [통합 IdM CA를 사용하여 사용자, 호스트 및 서비스의 인증서 관리](#)

3.4. IDM 웹 UI를 사용하여 IDM 사용자, 호스트 또는 서비스 계정에서 외부 CA에서 발급한 인증서 제거

IdM(Identity Management) 관리자는 IdM(Identity Management) 웹 UI를 사용하여 IdM 사용자, 호스트 또는 서비스의 계정에서 외부 서명된 인증서를 제거할 수 있습니다.

사전 요구 사항

- **IdM(Identity Management) 웹 UI에 관리자로 로그인되어 있습니다.**

절차

1. **Identity** 탭을 열고 사용자,호스트 또는 서비스 하위 탭을 선택합니다.
2. 사용자, 호스트 또는 서비스의 이름을 클릭하여 구성 페이지를 엽니다.
3. 삭제할 인증서 옆에 있는 작업을 클릭하고 삭제를 선택합니다.
4. 저장을 클릭하여 변경 사항을 저장합니다.

3.5. 추가 리소스

[role="_additional-resources"]

- **Ansible** 플레이북을 사용하여 **IdM** 서비스 항목에 외부 서명 인증서가 있는지 확인

4장. IDM에서 작동하도록 인증서 형식 변환

이 사용자 사례에서는 **IdM** 시스템 관리자로서 특정 **IdM** 명령과 함께 올바른 형식의 인증서를 사용하고 있는지 확인하는 방법을 설명합니다. 예를 들어 다음과 같은 경우 유용합니다.

- 사용자 프로필에 외부 인증서를 로드하고 있습니다. 자세한 내용은 [외부 인증서를 변환하여 IdM 사용자 계정으로 로드합니다.](#)
- 스마트 카드 인증을 위해 **IdM** 서버를 구성하거나 스마트 카드 인증을 위해 **IdM** 클라이언트를 구성할 때 외부 **CA** 인증서를 사용하고 있으며 사용자는 외부 인증 기관에서 발급한 인증서와 함께 스마트 카드를 사용하여 **IdM**에 인증할 수 있습니다.
- **NSS** 데이터베이스의 인증서를 인증서와 개인 키가 모두 포함된 **pkcs #12** 형식으로 내보내고 있습니다. 자세한 내용은 [NSS 데이터베이스에서 PKCS #12 파일로 인증서 및 개인 키 내보내기를 참조하십시오.](#)

4.1. IDM의 인증서 형식 및 인코딩

IdM의 스마트 카드 인증이 포함된 인증서 인증은 사용자가 제공하는 인증서 또는 사용자의 **IdM** 프로필에 저장된 인증서 데이터를 비교하여 진행합니다.

시스템 구성

IdM 프로필에 저장된 내용은 해당 개인 키가 아닌 인증서일 뿐입니다. 인증하는 동안 사용자는 해당 개인 키를 보유하는 경우에도 표시되어야 합니다. 사용자는 인증서와 개인 키가 모두 포함된 **PKCS #12** 파일을 제공하거나 두 개의 파일(인증서 포함)과 개인 키를 포함하는 파일을 제공하여 이를 수행합니다.

따라서 사용자 프로필에 인증서를 로드하는 것과 같은 프로세스는 개인 키가 포함되지 않은 인증서 파일만 허용합니다.

마찬가지로 시스템 관리자가 외부 **CA** 인증서를 제공할 때 공개 데이터만 제공합니다. 개인 키가 없는 인증서만 제공합니다. **IdM** 서버 또는 스마트 카드 인증을 위한 **IdM** 클라이언트를 구성하기 위한 **ipa-advise** 유틸리티에서는 입력 파일에 개인 키가 아닌 외부 **CA**의 인증서가 포함되어 있어야 합니다.

인증서 인코딩

두 가지 일반적인 인증서 인코딩이 있습니다: 개인 정보 보호 전자 메일 (**PEM**) 및 **Distinguished Encoding** 규칙 (**DER**). **base64** 형식은 **PEM** 형식과 거의 동일하지만 **-----BEGIN CERTIFICATE-----**

END CERTIFICATE----END CERTIFICATE ----- 헤더 및 **footer**는 포함되어 있지 않습니다.

DER를 사용하여 인코딩한 인증서는 바이너리 **X509** 디지털 인증서 파일입니다. 바이너리 파일로, 인증서는 사람이 읽을 수 없습니다. **DER** 파일은 때때로 **.der** 파일 이름 확장자를 사용하지만 **.crt** 및 **.cer** 파일 이름이 확장된 파일에도 **DER** 인증서가 포함될 수 있습니다. 키를 포함하는 **DER** 파일의 이름은 **.key**일 수 있습니다.

PEM Base64를 사용하여 인코딩한 인증서는 사람이 읽을 수 있는 파일입니다. 파일에는 **ASCII(Base64)**가 **"-----BEGIN"** 줄이 접두사가 붙은 데이터가 포함되어 있습니다. **PEM** 파일은 경우에 따라 **.pem** 파일 이름 확장자를 사용하지만 **.crt** 및 **.cer** 파일 이름 확장자가 있는 파일에 **PEM** 인증서가 포함된 경우가 있습니다. 키가 포함된 **PEM** 파일의 이름은 **.key**로 지정할 수 있습니다.

다른 **ipa** 명령은 수락하는 인증서 유형에 대한 다른 제한 사항이 있습니다. 예를 들어 **ipa user-add-cert** 명령은 **base64** 형식으로 인코딩된 인증서만 허용하지만 **ipa-server-certinstall**은 **PEM**, **DER**, **PKCS #7**, **PKCS #8** 및 **PKCS #12** 인증서를 허용합니다.

표 4.1. 인증서 인코딩

인코딩 형식	사람이 읽을 수 있는	일반적인 파일 이름 확장	인코딩 형식을 허용하는 샘플 IdM 명령
PEM/base64	있음	.pem, .crt, .cer	ipa user-add-cert, ipa-server-certinstall, ...
DER	없음	.der, .crt, .cer	ipa-server-certinstall, ...

IdM의 인증서 관련 명령 및 형식은 명령에서 허용하는 인증서 형식과 함께 추가 **ipa** 명령을 나열합니다.

사용자 인증

웹 **UI**를 사용하여 **IdM**에 액세스하는 경우 사용자는 두 브라우저의 데이터베이스에 두 가지를 저장하여 인증서에 해당하는 개인 키를 보유하고 있음을 증명합니다.

CLI를 사용하여 **IdM**에 액세스하는 경우 사용자는 다음 방법 중 하나로 인증서에 해당하는 개인 키를 보유하고 있음을 증명합니다.

- 사용자는 **kinit -X** 명령의 **X509_user_identity** 매개변수 값으로, 인증서와 키를 모두 포함하는 스마트 카드 모듈에 대한 경로를 추가합니다.

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' idm_user
```

- 사용자는 두 개의 파일을 **kinit -X** 명령의 **X509_user_identity** 매개변수 값으로 추가합니다. 하나는 인증서와 다른 개인 키를 포함합니다.

```
$ kinit -X X509_user_identity='FILE:`/path/to/cert.pem,/path/to/cert.key`' idm_user
```

유용한 인증서 명령

주체 및 발급자와 같은 인증서 데이터를 보려면 다음을 수행합니다.

```
$ openssl x509 -noout -text -in ca.pem
```

두 인증서의 차이가 있는 행을 비교하려면 다음을 수행합니다.

```
$ diff cert1.crt cert2.crt
```

두 인증서의 행을 두 개의 인증서가 두 열에 표시된 출력과 비교하려면 다음을 수행합니다.

```
$ diff cert1.crt cert2.crt -y
```

4.2. 외부 인증서를 IDM 사용자 계정으로 로드하도록 변환

이 섹션에서는 사용자 항목에 추가하기 전에 외부 인증서가 올바르게 인코딩되고 포맷되는지 확인하는 방법을 설명합니다.

4.2.1. 사전 요구 사항

- **Active Directory** 인증 기관에서 인증서를 발급했으며 **PEM** 인코딩을 사용하는 경우 **PEM** 파일이 **UNIX** 형식으로 변환되었는지 확인합니다. 파일을 변환하려면 **eponymous** 패키지에서 제공하는 **dos2unix** 유틸리티를 사용합니다.

4.2.2. IdM CLI에서 외부 인증서를 변환하여 IdM 사용자 계정으로 로드합니다.

IdM CLI는 첫 번째 및 마지막 행(**-----BEGIN CERTIFICATE---** 및 **---END CERTIFICATE-----**)이 제거된 **PEM** 인증서만 허용합니다.

다음 절차에 따라 외부 인증서를 **PEM** 형식으로 변환하고 **IdM CLI**를 사용하여 **IdM** 사용자 계정에 추가합니다.

절차

1. 인증서를 **PEM** 형식으로 변환합니다.

- 인증서가 **DER** 형식인 경우:

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

- 파일이 **PKCS #12** 형식인 경우 공통 파일 이름 확장자가 **.pfx** 및 **.p12** 이고 인증서, 개인 키 및 기타 데이터가 포함되어 있는 경우 **openssl pkcs12** 유틸리티를 사용하여 인증서를 추출하십시오. 메시지가 표시되면 파일에 저장된 개인 키를 보호하는 암호를 입력합니다.

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2. 관리자의 자격 증명을 가져옵니다.

```
$ kinit admin
```

3. 다음 방법 중 하나에서 **IdM CLI** 를 사용하여 사용자 계정에 인증서를 추가합니다.

- **ipa user-add-cert** 명령에 문자열을 추가하기 전에 **sed** 유틸리티를 사용하여 **PEM** 파일의 첫 번째 및 마지막 행(**--END CERTIFICATE----** 및 **---END CERTIFICATE-----**)을 제거합니다.

```
$ ipa user-add-cert some_user --certificate="$(sed -e '/BEGIN
CERTIFICATE/d;/END CERTIFICATE/d' cert.pem)"
```

- 첫 번째 및 마지막 행(**-----BEGIN CERTIFICATE---** 및 **---END CERTIFICATE-----**)을 **ipa user-add-cert** 명령으로 복사 및 붙여 넣습니다.

```
$ ipa user-add-cert some_user --
certificate=MIIDlzCCAn+gAwIBAgIBATANBgkqhki...
```



참고

첫 번째 및 마지막 행(---BEGIN CERTIFICATE--- 및 ---END CERTIFICATE---)을 먼저 제거하지 않고 인증서가 포함된 PEM 파일을 `ipa user-add-cert` 명령에 직접 전달할 수 없습니다.

```
$ ipa user-add-cert some_user --cert=some_user_cert.pem
```

이 명령을 실행하면 "**ipa: ERROR: Base64 디코딩 failed: Incorrect padding**" 오류 메시지가 표시됩니다.

4.

선택적으로 인증서가 시스템에서 수락되었는지 확인하려면 다음을 수행하십시오.

```
[idm_user@r8server]$ ipa user-show some_user
```

4.2.3. IdM 사용자 계정으로 로드하기 위해 IdM 웹 UI에서 외부 인증서를 변환

다음 절차에 따라 외부 인증서를 PEM 형식으로 변환하고 IdM 웹 UI의 IdM 사용자 계정에 추가합니다.

절차

1.

CLI 를 사용하여 인증서를 PEM 형식으로 변환합니다.

•

인증서가 DER 형식인 경우:

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

•

파일이 PKCS #12 형식인 경우 공통 파일 이름 확장자가 `.pfx` 및 `.p12` 이고 인증서, 개인 키 및 기타 데이터가 포함되어 있는 경우 `openssl pkcs12` 유틸리티를 사용하여 인증서를 추출하십시오. 메시지가 표시되면 파일에 저장된 개인 키를 보호하는 암호를 입력합니다.

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2.

편집기에서 인증서를 열고 콘텐츠를 복사합니다. IdM 웹 UI에서 PEM 및 base64 형식을 모두 수락해야 하므로 "-----BEGIN CERTIFICATE-----" 및 "-END CERTIFICATE-----" 헤더와 footer 행 모두를 포함할 수 있습니다.

3. **IdM 웹 UI**에서 보안 담당자로 로그인합니다.
4. **Identity → Users → some_user** 로 이동합니다.
5. 인증서 옆에 있는 추가를 클릭합니다.
6. 인증서의 **PEM** 형식 콘텐츠를 여는 창에 붙여넣습니다.
7. 추가를 클릭합니다.

시스템에서 인증서를 수락한 경우 사용자 프로필의 인증서 중 목록을 볼 수 있습니다.

4.3. 브라우저에 인증서 로드 준비

브라우저로 사용자 인증서를 가져오기 전에 인증서 및 해당 개인 키가 **PKCS #12** 형식이어야 합니다. 추가 준비 작업이 필요한 두 가지 일반적인 상황이 있습니다.

- 인증서는 **NSS** 데이터베이스에 있습니다. 이 상황에서 진행되는 방법에 대한 자세한 내용은 [NSS 데이터베이스에서 인증서 및 개인 키 내보내기를 PKCS #12 파일로](#) 참조하십시오.
- 인증서와 개인 키는 두 개의 별도의 **PEM** 파일에 있습니다. 이 상황에서 진행되는 방법에 대한 자세한 내용은 [인증서 및 개인 키 PEM 파일을 PKCS #12 파일로 결합을](#) 참조하십시오.

그 후, **PEM** 형식으로 **CA** 인증서를 둘 다 가져오고 **PKCS #12** 형식의 사용자 인증서를 브라우저에 가져오려면 [브라우저 설정 절차에 따라 ID 관리 웹 UI를 사용하여 ID 관리 웹 UI에 대한 인증 및 ID 관리 사용자](#).

4.3.1. NSS 데이터베이스에서 인증서 및 개인 키를 PKCS #12 파일로 내보내기

절차

1. **pk12util** 명령을 사용하여 **NSS** 데이터베이스에서 **PKCS12** 형식으로 인증서를 내보냅니다. 예를 들어 **~/certdb** 디렉터리에 저장된 **NSS** 데이터베이스에서 **some_user nickname**을 사용하

여 인증서를 `~/some_user.p12` 파일로 내보내려면 다음을 실행합니다.

```
$ pk12util -d ~/certdb -o ~/some_user.p12 -n some_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

2.

`.p12` 파일에 적절한 권한을 설정합니다.

```
# chmod 600 ~/some_user.p12
```

PKCS #12 파일에는 개인 키도 포함되어 있으므로 다른 사용자가 파일을 사용하지 못하도록 보호해야 합니다. 그렇지 않으면 사용자가 가장할 수 있습니다.

4.3.2. 인증서 및 개인 키 PEM 파일을 PKCS #12 파일로 결합

이 섹션에서는 인증서와 별도의 **PEM** 파일에 저장된 해당 키를 **PKCS #12** 파일로 결합하는 방법을 설명합니다.

절차

- `certfile.cer`에 저장된 인증서와 `certfile.key`에 저장된 키를 인증서와 키가 모두 포함된 `certfile.p12` 파일로 결합하려면 다음을 수행합니다.

```
$ openssl pkcs12 -export -in certfile.cer -inkey certfile.key -out certfile.p12
```

4.4. IdM의 인증서 관련 명령 및 형식

표 **IdM 인증서 명령 및 형식**은 허용 가능한 형식을 사용하여 IdM에 인증서 관련 명령을 표시합니다.

표 4.2. IdM 인증서 명령 및 형식

명령	허용 가능한 형식	참고
<code>ipa user-add-cert some_user --certificate</code>	base64 PEM 인증서	
<code>ipa-server-certinstall</code>	PEM 및 DER 인증서, PKCS#7 인증서 체인, PKCS#8 및 원시 개인 키, PKCS#12 인증서 및 개인 키	

명령	허용 가능한 형식	참고
ipa-cacert-manage 설치	DER; PEM; PKCS#7	
ipa-cacert-manage 갱신 --external-cert-file	PEM 및 DER 인증서; PKCS#7 인증서 체인	
ipa-ca-install --external-cert-file	PEM 및 DER 인증서; PKCS#7 인증서 체인	
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	해당 없음	< cert_serial > 일련 번호가 있는 인증서로 PEM 인코딩 file.pem 파일을 생성합니다.
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	해당 없음	< cert_serial > 일련 번호가 있는 인증서로 PEM 인코딩 file.pem 파일을 생성합니다. --chain 옵션을 사용하면 PEM 파일에 인증서 체인을 포함한 인증서가 포함됩니다.
ipa cert-request --certificate-out=FILE /path/to/req.csr	해당 없음	새 인증서로 PEM 형식으로 req.csr 파일을 생성합니다.
ipa cert-request --certificate-out=FILE /path/to/req.csr	해당 없음	새 인증서로 PEM 형식으로 req.csr 파일을 생성합니다. --chain 옵션을 사용하면 PEM 파일에 인증서 체인을 포함한 인증서가 포함됩니다.

5장. IDENTITY MANAGEMENT에서 인증서 프로필 생성 및 관리

인증서 프로필은 인증서에 서명할 때 인증서 서명 요청(CSR)이 허용되는지, 인증서 서명 요청(CSR)이 인증서에 존재하는 기능 및 확장이 있는지 확인할 때 **CA**(인증 기관)에서 사용됩니다. 인증서 프로필은 특정 유형의 인증서를 발급하는 것과 연결되어 있습니다. 인증서 프로필과 **CA ACL**(액세스 제어 목록)을 결합하면 사용자 정의 인증서 프로필에 대한 액세스를 정의하고 제어할 수 있습니다.

이 절차에서는 인증서 프로필을 생성하는 방법을 설명하여 **S/MIME** 인증서를 예로 사용합니다. 일부 이메일 프로그램은 **S/MIME**(Secure Multipurpose Internet mail Extension) 프로토콜을 사용하여 디지털 서명 및 암호화된 이메일을 지원합니다. **S/MIME**를 사용하여 이메일 메시지를 서명하거나 암호화하려면 메시지의 발신자가 **S/MIME** 인증서를 가져야 합니다.

- [인증서 프로필이란?](#)
- [인증서 프로파일 생성](#)
- [CA 액세스 제어 목록](#)
- [인증서 프로필에 대한 액세스를 제어하기 위한 CA ACL 정의](#)
- [인증서 프로필 및 CA ACL을 사용하여 인증서 발급](#)
- [인증서 프로필 수정](#)
- [인증서 프로필 구성 매개변수](#)

5.1. 인증서 프로필이란 무엇입니까?

인증서 프로필을 사용하여 인증서의 콘텐츠와 다음과 같은 인증서를 발급하기 위한 제약 조건을 확인할 수 있습니다.

- 인증서 서명 요청을 연결하는 데 사용할 서명 알고리즘입니다.

- 인증서의 기본 유효성입니다.
- 인증서를 취소하는 데 사용할 수 있는 해지 이유는 다음과 같습니다.
- 보안 주체의 공통 이름이 주체 대체 이름 필드에 복사되는 경우 **If the common name of the principal is copied to the subject alternative name field.**
- 인증서에 존재해야 하는 기능 및 확장 기능입니다.

단일 인증서 프로필은 특정 유형의 인증서를 발급하는 것과 연결되어 있습니다. **IdM**에서 사용자, 서비스 및 호스트에 대해 다른 인증서 프로필을 정의할 수 있습니다. **IdM**에는 기본적으로 다음 인증서 프로필이 포함됩니다.

- **calPAserviceCert**
- **IECUserRoles**
- **NetNamespaces_PKINIT_Certs (internally)**

또한 특정 용도로 인증서를 발행할 수 있는 사용자 정의 프로필을 생성하고 가져올 수 있습니다. 예를 들어 특정 프로필의 사용을 한 명의 사용자 또는 하나의 그룹으로만 제한하여 다른 사용자 및 그룹이 인증을 위해 인증서를 발급하는 데 해당 프로필을 사용하지 못하도록 할 수 있습니다. 사용자 정의 인증서 프로필을 생성하려면 **ipa certprofile** 명령을 사용합니다.

추가 리소스

- **ipa help certprofile** 명령을 참조하십시오.

5.2. 인증서 프로파일 생성

이 절차에서는 **S/MIME** 인증서를 요청하는 프로필 구성 파일을 만들어 명령줄을 통해 인증서 프로필을 생성하는 방법을 설명합니다.

절차

1. 기존 기본 프로필을 복사하여 사용자 정의 프로필을 생성합니다.

```
$ ipa certprofile-show --out smime.cfg calPAserviceCert
-----
Profile configuration stored in file 'smime.cfg'
-----
Profile ID: calPAserviceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE
```

2. 텍스트 편집기에서 새로 생성된 프로필 구성 파일을 엽니다.

```
$ vi smime.cfg
```

3. 프로필 ID 를 프로필 사용을 반영하는 이름으로 변경합니다(예: **smime**).



참고

새로 생성된 프로필을 가져올 때 **profileid** 필드가 있는 경우 명령줄에 지정된 ID와 일치해야 합니다.

4. 확장 키 사용 구성을 업데이트합니다. 기본 확장 키 사용 확장 구성은 **TLS** 서버 및 클라이언트 인증에 사용됩니다. 예를 들어 **S/MIME**의 경우 이메일 보호를 위해 확장 키 사용을 구성해야 합니다.

```
policyset.serverCertSet.7.default.params.exKeyUsageOIDs=1.3.6.1.5.5.7.3.4
```

5. 새 프로필을 가져옵니다.

```
$ ipa certprofile-import smime --file smime.cfg \
--desc "S/MIME certificates" --store TRUE
-----
Imported profile "smime"
-----
Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
```

검증 단계

- 새 인증서 프로파일 가져왔는지 확인합니다.

```
$ ipa certprofile-find
```

```
-----
4 profiles matched
-----
```

```
Profile ID: calPAserviceCert
```

```
Profile description: Standard profile for network services
```

```
Store issued certificates: TRUE
```

```
Profile ID: IECUserRoles
```

```
Profile description: User profile that includes IECUserRoles extension from request
```

```
Store issued certificates: TRUE
```

```
Profile ID: KDCs_PKINIT_Certs
```

```
Profile description: Profile for PKINIT support by KDCs
```

```
Store issued certificates: TRUE
```

```
Profile ID: smime
```

```
Profile description: S/MIME certificates
```

```
Store issued certificates: TRUE
```

```
-----
Number of entries returned 4
-----
```

추가 리소스

- `ipa help certprofile` 을 참조하십시오.
- [RFC 5280, 섹션 4.2.1.12](#) 를 참조하십시오.

5.3. CA 액세스 제어 목록이란 무엇입니까?

CA ACL(인증 기관 액세스 제어 목록) 규칙은 보안 주체에 인증서를 발급하는 데 사용할 수 있는 프로 파일을 정의합니다. **CA ACL**을 사용하여 이 작업을 수행할 수 있습니다. 예를 들면 다음과 같습니다.

- 특정 프로 파일을 사용하여 인증서를 발급할 수 있는 사용자, 호스트 또는 서비스 확인
- 인증서를 발급할 수 있는 **IdM** 인증 기관 또는 하위 **CA** 확인

예를 들어, **CA ACL**을 사용하면 런던에 위치한 사무실에서 작업 중인 직원은 런던 사무실 관련 **IdM** 사용자 그룹의 멤버인 사용자에게만 사용할 수 있습니다.

CA ACL 규칙 관리를 위한 **ipa caacl** 유틸리티를 사용하면 권한 있는 사용자가 지정된 **CA ACL**을 추가, 표시, 수정 또는 삭제할 수 있습니다.

추가 리소스

- **ipa help caacl** 을 참조하십시오.

5.4. 인증서 프로필에 대한 액세스를 제어하기 위한 **CA ACL** 정의

이 절차에서는 **caacl** 유틸리티를 사용하여 사용자 정의 인증서 프로필에 있는 사용자가 액세스할 수 있도록 **CA ACL**(액세스 제어 목록) 규칙을 정의하는 방법을 설명합니다. 이 경우 절차에서는 **S/MIME** 사용자 그룹 및 **CA ACL**을 생성하여 해당 그룹의 사용자가 **smime** 인증서 프로필에 액세스할 수 있도록 하는 방법을 설명합니다.

사전 요구 사항

- **IdM** 관리자의 인증 정보를 가져왔는지 확인합니다.

절차

1. 인증서 프로필의 사용자를 위한 새 그룹을 생성합니다.

```
$ ipa group-add smime_users_group
-----
Added group "smime users group"
-----
Group name: smime_users_group
GID: 75400001
```

2. **smime_user_group** 그룹에 추가할 새 사용자를 생성합니다.

```
$ ipa user-add smime_user
First name: smime
Last name: user
-----
Added user "smime_user"
-----
```

```

User login: smime_user
First name: smime
Last name: user
Full name: smime user
Display name: smime user
Initials: TU
Home directory: /home/smime_user
GECOS: smime user
Login shell: /bin/sh
Principal name: smime_user@IDM.EXAMPLE.COM
Principal alias: smime_user@IDM.EXAMPLE.COM
Email address: smime_user@idm.example.com
UID: 1505000004
GID: 1505000004
Password: False
Member of groups: ipausers
Kerberos keys available: False

```

3.

smime_user 를 **smime_users_group** 그룹에 추가합니다.

```

$ ipa group-add-member smime_users_group --users=smime_user
Group name: smime_users_group
GID: 1505000003
Member users: smime_user
-----
Number of members added 1
-----

```

4.

그룹의 사용자가 인증서 프로필에 액세스할 수 있도록 **CA ACL**을 만듭니다.

```

$ ipa caacl-add smime_acl
-----
Added CA ACL "smime_acl"
-----
ACL name: smime_acl
Enabled: TRUE

```

5.

CA ACL에 사용자 그룹을 추가합니다.

```

$ ipa caacl-add-user smime_acl --group smime_users_group
ACL name: smime_acl
Enabled: TRUE
User Groups: smime_users_group
-----
Number of members added 1
-----

```

6.

CA ACL에 인증서 프로필을 추가합니다.

```
$ ipa caacl-add-profile smime_acl --certprofile smime
ACL name: smime_acl
Enabled: TRUE
Profiles: smime
User Groups: smime_users_group
-----
Number of members added 1
-----
```

검증 단계

- 생성한 **CA ACL**의 세부 정보를 확인합니다.

```
$ ipa caacl-show smime_acl
ACL name: smime_acl
Enabled: TRUE
Profiles: smime
User Groups: smime_users_group
...
```

추가 리소스

- **ipa man** 페이지를 참조하십시오.
- **ipa help caacl** 을 참조하십시오.

5.5. 인증서 프로필 및 **CA ACL**을 사용하여 인증서 발급

CA ACL(인증 기관 액세스 제어 목록)에서 허용하는 경우 인증서 프로필을 사용하여 인증서를 요청할 수 있습니다. 이 절차에서는 **CA ACL**을 통해 액세스 권한이 부여된 사용자 정의 인증서 프로필을 사용하여 사용자에게 **S/MIME** 인증서를 요청하는 방법을 설명합니다.

사전 요구 사항

- 인증서 프로필이 생성되었습니다.
- 사용자가 필요한 인증서 프로필을 사용하여 인증서를 요청할 수 있는 **CA ACL**이 생성되었습니다.



참고

cert-request 명령을 수행하는 사용자가 **CA ACL** 검사를 바이패스할 수 있습니다.

- 는 **admin** 사용자입니다.
- **CA ACL** 요청에서 **CA ACL** 권한이 무시되어 있습니다.

절차

1. 사용자에게 인증서 요청을 생성합니다. 예를 들어 **OpenSSL**을 사용합니다.

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout private.key -out
cert.csr -subj '/CN=smime_user'
```

2. IdM CA에서 사용자의 새 인증서를 요청합니다.

```
$ ipa cert-request cert.csr --principal=smime_user --profile-id=smime
```

선택적으로 **--ca 하위-CA_name** 옵션을 명령에 전달하여 루트 **CA** 대신 하위 **CA**에서 인증서를 요청합니다.

검증 단계

- 새로 발급한 인증서가 사용자에게 할당되어 있는지 확인합니다.

```
$ ipa user-show user
User login: user
...
Certificate: MIICfzCCAWcCAQA...
...
```

추가 리소스

- **ipa(a)** 매뉴얼 페이지를 참조하십시오.

- **ipa help user-show** 명령을 참조하십시오.
- **ipa help cert-request** 명령을 참조하십시오.
- **openssl(1ssl)** 매뉴얼 페이지를 참조하십시오.

5.6. 인증서 프로파일 수정

이 절차에서는 **ipa certprofile-mod** 명령을 사용하여 명령줄을 통해 직접 인증서 프로파일을 수정하는 방법을 설명합니다.

절차

1. 수정 중인 인증서 프로파일의 인증서 프로파일 ID를 확인합니다. IdM에 현재 저장된 모든 인증서 프로파일을 표시하려면 다음을 수행합니다.

```
# ipa certprofile-find

-----
4 profiles matched
-----
Profile ID: calPAserviceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE

Profile ID: IECUserRoles
...

Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates: TRUE
-----
Number of entries returned
-----
```

2. 인증서 프로파일 설명을 수정합니다. 예를 들어 기존 프로파일을 사용하여 **S/MIME** 인증서에 대한 사용자 정의 인증서 프로파일을 생성한 경우 새 사용과 함께 설명을 변경합니다.

```
# ipa certprofile-mod smime --desc "New certificate profile description"
-----
Modified Certificate Profile "smime"
-----
```

```

Profile ID: smime
Profile description: New certificate profile description
Store issued certificates: TRUE

```

3. 텍스트 편집기에서 고객 인증서 프로파일 파일을 열고 요구 사항에 맞게 수정합니다.

```
# vi smime.cfg
```

인증서 프로파일 구성 파일에서 구성할 수 있는 옵션에 대한 자세한 내용은 [인증서 프로파일 구성 매개 변수](#)를 참조하십시오.

4. 기존 인증서 프로파일 구성 파일을 업데이트합니다.

```
# ipa certprofile-mod _profile_ID_ --file=smime.cfg
```

검증 단계

- 인증서 프로파일 이 업데이트되었는지 확인합니다.

```

$ ipa certprofile-show smime
Profile ID: smime
Profile description: New certificate profile description
Store issued certificates: TRUE

```

추가 리소스

- [ipa\(a\)](#) 매뉴얼 페이지를 참조하십시오.
- [ipa help certprofile-mod](#) 를 참조하십시오.

5.7. 인증서 프로파일 구성 매개변수

인증서 프로파일 구성 매개변수는 **CA 프로파일** 디렉터리인 `/var/lib/pki/pki-tomcat/ca/profiles/ca.cfg` 파일에 저장됩니다. 프로파일의 모든 매개 변수(기본값, 입력, 출력 및 제약 조건)는 단일 정책 세트 내에서 구성됩니다. 인증서 프로파일 에 설정된 정책에는 `name policyset.policyName.policyNumber` 가 있습니다. 예를 들어 `policy set serverCertSet`의 경우

```

policyset.list=serverCertSet
policyset.serverCertSet.list=1,2,3,4,5,6,7,8

```

```

policysset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl
policysset.serverCertSet.1.constraint.name=Subject Name Constraint
policysset.serverCertSet.1.constraint.params.pattern=CN=[^,]+,+.+
policysset.serverCertSet.1.constraint.params.accept=true
policysset.serverCertSet.1.default.class_id=subjectNameDefaultImpl
policysset.serverCertSet.1.default.name=Subject Name Default
policysset.serverCertSet.1.default.params.name=CN=$request.req_subject_name.cn$, OU=pki-
ipa, O=IPA
policysset.serverCertSet.2.constraint.class_id=validityConstraintImpl
policysset.serverCertSet.2.constraint.name=Validity Constraint
policysset.serverCertSet.2.constraint.params.range=740
policysset.serverCertSet.2.constraint.params.notBeforeCheck=false
policysset.serverCertSet.2.constraint.params.notAfterCheck=false
policysset.serverCertSet.2.default.class_id=validityDefaultImpl
policysset.serverCertSet.2.default.name=Validity Default
policysset.serverCertSet.2.default.params.range=731
policysset.serverCertSet.2.default.params.startTime=0

```

각 정책 세트에는 평가해야 하는 순서대로 정책 ID 번호별로 인증서 프로필에 대해 구성된 정책 목록이 포함되어 있습니다. 서버는 수신하는 각 요청에 대해 설정된 각 정책 세트를 평가합니다. 단일 인증서 요청이 수신되면 하나의 세트가 평가되고 프로필의 다른 집합이 무시됩니다. 이중 키 쌍을 발급하면 첫 번째 정책 세트가 첫 번째 인증서 요청에 대해 평가되고 두 번째 세트는 두 번째 인증서 요청에 대해 평가됩니다. 듀얼 키 쌍을 발행할 때 단일 인증서 또는 두 개 이상의 세트를 발행할 때 두 개 이상의 정책 세트가 필요하지 않습니다.

표 5.1. 인증서 프로필 구성 파일 매개변수

매개변수	설명
desc	최종 항목 페이지에 표시되는 인증서 프로필에 대한 무료 텍스트 설명입니다. 예를 들어 desc=이 인증서 프로필은 에이전트 인증을 사용하여 서버 인증서를 등록하는 것입니다.
enable	최종 엔터티 페이지를 통해 액세스할 수 있도록 프로필을 활성화합니다. 예를 들면 enable=true 입니다.
auth.instance_id	인증서 요청을 인증하는 데 사용할 인증 관리자 플러그인을 설정합니다. 자동 등록의 경우 인증이 성공하면 CA에서 즉시 인증서를 발행합니다. 인증에 실패하거나 인증 플러그인이 지정되지 않은 경우 에이전트에서 수동으로 승인하도록 요청이 대기됩니다. 예: auth.instance_id=AgentCertAuth.

매개변수	설명
authz.acl	권한 부여 제약 조건을 지정합니다. 이는 그룹 계산 ACL(액세스 제어 목록)을 설정하는 데 주로 사용됩니다. 예를 들어 caCMCUserCert 매개변수를 사용하려면 CMC 요청의 서명자가 인증서 관리자 에이전트 그룹에 속해야 합니다. authz.acl=group="Certificate Manager 에이전트s 디렉터리 기반 사용자 인증서 갱신에서는 이 옵션을 사용하여 원래 요청자 및 현재 authenticated 사용자가 동일한지 확인합니다. 권한 부여를 평가하기 전에 엔터티를 인증(바인 또는 기본적으로 시스템에 로그인)해야 합니다.
name	인증서 프로파일의 이름입니다. 예를 들어 name=Agent-Authenticated Server Certificate Enrollment. 이 이름은 최종 사용자 등록 또는 갱신 페이지에 표시됩니다.
input.list	인증서 프로파일 허용된 입력을 이름으로 나열합니다. 예: input.list=i1,i2.
input.input_id.class_id	입력 ID(input.list에 나열된 입력 이름)별 입력의 java 클래스 이름을 나타냅니다. 예를 들어 input.i1.class_id=certReqInputImpl.
output.list	인증서 프로파일의 가능한 출력 형식을 이름으로 나열합니다. 예: output.list=o1.
output.output_id.class_id	output.list로 이름이 지정된 출력 형식의 Java 클래스 이름을 지정합니다. 예를 들어 output.o1.class_id=certOutputImpl.
policyset.list	구성된 인증서 프로파일 규칙을 나열합니다. 이중 인증서의 경우 하나의 규칙 세트는 서명 키와 다른 규칙 세트에 적용됩니다. 단일 인증서는 하나의 인증서 프로파일 규칙 세트만 사용합니다. 예를 들면 policyset.list=serverCertSet 입니다.
policyset.policyset_id.list	평가해야 하는 순서대로 정책 ID 번호로 인증서 프로파일 구성 정책 내의 정책을 나열합니다. 예를 들어 policyset.serverCertSet.list=1,2,3,4,5,6,7,8 입니다.
policyset.policyset_id.policy_number.constraint.class_id	프로파일 규칙에 구성된 기본값에 대해 설정된 제약 조건 플러그인의 Java 클래스 이름을 나타냅니다. 예를 들어 policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl입니다.

매개변수	설명
policyset.policyset_id.policy_number.constraint.name	계약 조건의 사용자 정의 이름을 제공합니다. 예를 들어 policyset.serverCertSet.1.constraint.name=Subject Name Constraint입니다.
policyset.policyset_id.policy_number.constraint.params.attribute	계약 조건에 허용되는 특성의 값을 지정합니다. 가능한 속성은 계약 조건 유형에 따라 다릅니다. 예를 들어 policyset.serverCertSet.1.constraint.params.pattern=CN=.*입니다.
policyset.policyset_id.policy_number.default.class_id	프로필 규칙에 설정된 기본 클래스의 Java 클래스 이름을 지정합니다. 예: policyset.serverCertSet.1.default.class_id=userSubjectNameDefaultImpl
policyset.policyset_id.policy_number.default.name	기본값의 사용자 정의 이름을 지정합니다. 예: policyset.serverCertSet.1.default.name=Subject Name Default
policyset.policyset_id.policy_number.default.params.attribute	기본값에 허용되는 특성의 값을 지정합니다. 사용 가능한 속성은 기본값 유형에 따라 다릅니다. 예를 들어 policyset.serverCertSet.1.default.params.name=CN=(Name)\$request.requestor_name\$입니다.

6장. IDM의 인증서 유효성 관리

IdM(Identity Management)에서는 향후 발급하려는 기존 인증서와 인증서의 유효성을 모두 관리할 수 있지만 방법은 다릅니다.

6.1. IDM CA에서 발급한 기존 인증서의 유효성 관리

IdM에서 인증서의 만료 날짜를 확인하는 다음 방법을 사용할 수 있습니다.

- [IdM WebUI의 만료 날짜 보기.](#)
- [CLI에서 만료 날짜 보기.](#)

다음과 같은 방법으로 **IdM CA**에서 발급한 기존 인증서의 유효성을 관리할 수 있습니다.

- 원래 **CSR**(인증서 서명 요청) 또는 개인 키에서 생성된 새 **CSR**을 사용하여 새 인증서를 요청하여 인증서를 갱신합니다. 다음 유틸리티를 사용하여 새 인증서를 요청할 수 있습니다.

certmonger

certmonger 를 사용하여 서비스 인증서를 요청할 수 있습니다. 인증서가 만료되기 전에 **certmonger** 는 인증서를 자동으로 갱신하여 서비스 인증서의 지속적인 유효성을 보장합니다. 자세한 내용은 [certmonger를 사용하여 서비스에 대한 IdM 인증서 가져오기](#)를 참조하십시오.

certutil

certutil 을 사용하여 사용자, 호스트 및 서비스 인증서를 갱신할 수 있습니다. 사용자 인증서 요청에 대한 자세한 내용은 [새 사용자 인증서 요청 및 클라이언트로 내보내기](#)를 참조하십시오.

OpenSSL

openssl 을 사용하여 사용자, 호스트 및 서비스 인증서를 갱신할 수 있습니다.

- 인증서를 취소하십시오. 자세한 내용은 다음을 참조하십시오.
 - [IdM WebUI를 사용하여 통합 IdM CA에서 인증서 해지.](#)

- **IdM CLI를 사용하여 통합 IdM CA에서 인증서 취소**
- 인증서가 일시적으로 취소된 경우 인증서를 복원합니다. 자세한 내용은 다음을 참조하십시오.
 - **IdM WebUI를 사용하여 통합 IdM CA의 인증서 복원.**
 - **IdM CLI를 사용하여 통합 IdM CA가 있는 인증서 복원.**

6.2. IDM CA에서 발행한 향후 인증서의 유효성 관리

IdM CA에서 발행한 향후 인증서의 유효성을 관리하려면 인증서 프로필을 수정, 가져오기 또는 생성합니다. 자세한 내용은 **ID 관리에서 인증서 프로필 생성 및 관리**를 참조하십시오.

6.3. IDM WEBUI에서 인증서 만료 날짜 보기

IdM WebUI를 사용하여 **IdM CA**에서 발행한 모든 인증서의 만료 날짜를 확인할 수 있습니다.

사전 요구 사항

- 관리자의 자격 증명을 확보했는지 확인합니다.

절차

1. 인증 메뉴에서 인증서 > 인증서 를 클릭합니다.
2. 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.

그림 6.1. 인증서 목록

Certificates		
Subject ▼ Search  Refresh  + Issue		
☐	Serial Number	Subject
☐	1	CN=Certificate Authority,O=EXAMPLE.COM
☐	2	CN=OCSP Subsystem,O=EXAMPLE.COM
☐	3	CN=server.example.com,O=EXAMPLE.COM
☐	4	CN=CA Subsystem O=EXAMPLE.COM

3.

인증서 정보 페이지에서 **Expires On** 정보를 찾습니다.

6.4. CLI에서 인증서 만료일 보기

CLI(명령줄 인터페이스)를 사용하여 인증서의 만료 날짜를 확인할 수 있습니다.

절차

- openssl 유틸리티를 사용하여 사람이 읽을 수 있는 형식으로 파일을 엽니다.

```
$ openssl x509 -noout -text -in ca.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O = IDM.EXAMPLE.COM, CN = Certificate Authority
    Validity
      Not Before: Oct 30 19:39:14 2017 GMT
      Not After : Oct 30 19:39:14 2037 GMT
```

6.5. 통합 IDM CA로 인증서 해지

6.5.1. 인증서 해지 이유

취소된 인증서가 유효하지만 인증에 사용할 수 없습니다. 이유 6: **Certificate Hold** 를 제외한 모든 취소는 영구적입니다.

기본 해지 이유는 0: 지정되지 않음입니다.

표 6.1. 취소 이유

ID	이유	설명
0	지정되지 않음	
1	key Compromised	인증서를 발급한 키는 더 이상 신뢰되지 않습니다. 가능한 원인: 토큰 손실, 부적절하게 액세스한 파일.
2	CA Compromised	인증서를 발급한 CA는 더 이상 신뢰되지 않습니다.
3	ffiliation changed	가능한 원인은 다음과 같습니다. * 한 사람이 퇴사하거나 다른 부서로 이동했습니다. * 호스트 또는 서비스가 중단됩니다.
4	대체됨	최신 인증서가 현재 인증서를 교체했습니다.
5	운영 중단	호스트 또는 서비스가 해제되어 있습니다.
6	Certificate Hold	인증서가 일시적으로 취소됩니다. 나중에 인증서를 복원할 수 있습니다.
8	CRL에서 제거	인증서가 CRL(인증서 해지 목록)에 포함되어 있지 않습니다.
9	권한 삭제	사용자, 호스트 또는 서비스는 더 이상 인증서를 사용할 수 없습니다.
10	특성 기관 (AA) Compromise	AA 인증서는 더 이상 신뢰되지 않습니다.

6.5.2. IdM WebUI를 사용하여 통합 IdM CA에서 인증서 취소

인증서의 개인 키를 분실한 것을 알고 있는 경우, 악용을 방지하기 위해 인증서를 취소해야 합니다. IdM WebUI를 사용하여 IdM CA에서 발행한 인증서를 취소하려면 이 절차를 완료합니다.

절차

1. 인증 > 인증서 > 인증서 를 클릭합니다.
2. 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.

그림 6.2. 인증서 목록

Certificates		
Subject ▼ Search  Refresh  + Issue		
☐	Serial Number	Subject
☐	1	CN=Certificate Authority,O=EXAMPLE.COM
☐	2	CN=OCSP Subsystem,O=EXAMPLE.COM
☐	3	CN=server.example.com,O=EXAMPLE.COM
☐	4	CN=CA Subsystem,O=EXAMPLE.COM

3.

인증서 정보 페이지에서 **Actions** → **Revoke Certificate**.를 클릭합니다.

4.

취소 이유를 선택하고 **Revoke** 를 클릭합니다. 자세한 내용은 [인증서 해지 이유](#)를 참조하십시오.

6.5.3. IdM CLI를 사용하여 통합 IdM CA에서 인증서 취소

인증서의 개인 키를 분실한 것을 알고 있는 경우, 악용을 방지하기 위해 인증서를 취소해야 합니다. IdM CLI를 사용하여 IdM CA에서 발행한 인증서를 취소하려면 이 절차를 완료합니다.

절차

- **ipa cert-revoke** 명령을 사용하고 다음을 지정합니다.
 - 인증서 일련 번호
 - 해지 이유의 ID 번호를 참조하십시오. 자세한 내용은 [인증서 해지 이유](#)를 참조하십시오.

예를 들어, 일련 번호 1032 로 인증서를 취소하려면 1: **Key Compromised** 을 입력하십시오.

```
$ ipa cert-revoke 1032 --revocation-reason=1
```

새 인증서를 요청하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오.

- 새 사용자 인증서 요청 및 클라이언트로 내보내기;
- **certmonger**를 사용하여 서비스에 대한 **IdM** 인증서를 가져옵니다.

6.6. 통합 IdM CA의 인증서 복원

6: Certificate Hold 로 인해 인증서를 취소한 경우 인증서의 개인 키가 손상되지 않은 경우 다시 복원할 수 있습니다. 인증서를 복원하려면 다음 절차 중 하나를 사용합니다.

- **IdM WebUI**를 사용하여 통합 **IdM CA**로 인증서 복원
- **IdM CLI**를 사용하여 통합 **IdM CA**로 인증서를 복원합니다.

6.6.1. IdM WebUI를 사용하여 통합 IdM CA의 인증서 복원

IdM WebUI를 사용하여 이유 **6: Certificate Hold** 로 인해 취소된 **IdM** 인증서를 복원하려면 이 절차를 완료합니다.

절차

1. 인증 메뉴에서 인증서 > 인증서 를 클릭합니다.
2. 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.

그림 6.3. 인증서 목록

Certificates		
Subject ▼ Search 🔍 Refresh ↻ Issue +		
☐	Serial Number	Subject
☐	1	CN=Certificate Authority,O=EXAMPLE.COM
☐	2	CN=OCSP Subsystem,O=EXAMPLE.COM
☐	3	CN=server.example.com,O=EXAMPLE.COM
☐	4	CN=CA Subsystem O=EXAMPLE.COM

3.

인증서 정보 페이지에서 **Actions** → 인증서 복원 을 클릭합니다.

6.6.2. IdM CLI를 사용하여 통합 IdM CA의 인증서 복원

이유 6: **Certificate Hold** 로 인해 취소된 IdM 인증서를 복원하려면 IdM CLI를 사용하려면 이 절차를 완료합니다.

절차

- **ipa cert-remove-hold** 명령을 사용하여 인증서 일련 번호를 지정합니다. 예를 들어 다음과 같습니다.

```
$ ipa cert-remove-hold 1032
```

7장. 스마트 카드 인증을 위한 ID 관리 구성

스마트 카드를 기반으로 하는 인증은 암호에 대한 대안입니다. 개인 키 및 인증서 형식으로 스마트 카드에 사용자 자격 증명을 저장할 수 있으며 특수 소프트웨어 및 하드웨어가 해당 카드에 액세스하는 데 사용됩니다. 스마트 카드를 **reader** 또는 **USB** 포트에 배치하고 암호를 제공하는 대신 스마트 카드의 **Pin** 코드를 제공하십시오.

IdM(Identity Management)은 다음을 사용하여 스마트 카드 인증을 지원합니다.

- **IdM** 인증 기관에서 발급한 사용자 인증서
- 외부 인증 기관에서 발급한 사용자 인증서

이 사용자 사례에서는 두 가지 유형의 인증서 모두에 대해 **IdM**에서 스마트 카드 인증을 설정하는 방법을 보여줍니다. 사용자 사례에서 **smartcard_ca.pem CA** 인증서는 신뢰할 수 있는 외부 인증 기관의 인증서가 포함된 파일입니다.

사용자 스토리에는 다음 모듈이 포함되어 있습니다.

- [스마트 카드 인증을 위한 IdM 서버 구성](#)
- [스마트 카드 인증을 위한 IdM 클라이언트 구성](#)
- [IdM 웹 UI에서 사용자 항목에 인증서 추가](#)
- [IdM CLI에서 사용자 항목에 인증서 추가](#)
- [스마트 카드 관리 및 사용을 위한 도구 설치](#)
- [스마트 카드에 인증서 저장](#)

- 스마트 카드를 사용하여 **IdM** 로그인
- 스마트 카드 인증을 사용하여 **GDM** 액세스 구성
- 스마트 카드 인증을 사용하여 **su** 액세스 구성

7.1. 스마트 카드 인증을 위한 ID 서버 구성

EXAMPLE.ORG 도메인의 인증 기관에서 발급한 인증서가 발행된 사용자에게 대해 스마트 카드 인증을 활성화하려면 **LDAP** 고유 이름(DN)이 **CN=Certificate Authority,DC=EXAMPLE,DC=ORG** **LE,DC=ORG** ; 스크립트를 구성하여 스크립트를 실행할 수 있도록 인증 기관의 인증서를 받아야 합니다. 예를 들어 기관에서 발급한 인증서가 있는 웹 페이지에서 인증서를 다운로드할 수 있습니다. 자세한 내용은 [Configuring a browser to enable certificate authentication](#) 에서 **1 - 4a** 단계를 참조하십시오.

IdM 인증 기관에서 인증서를 발급한 **IdM** 사용자에게 대한 스마트 카드 인증을 활성화하려면 **IdM CA**가 실행 중인 **IdM** 서버의 **/etc/ipa/ca.crt** 파일에서 **CA** 인증서를 받으십시오.

이 섹션에서는 스마트 카드 인증을 위해 **IdM** 서버를 구성하는 방법을 설명합니다. 먼저 **PEM** 형식으로 **CA** 인증서로 파일을 가져온 다음 빌드된 **ipa-adviser** 스크립트를 실행합니다. 마지막으로 시스템 구성을 다시 로드합니다.

사전 요구 사항

- **IdM** 서버에 대한 루트 액세스 권한이 있습니다.
- 루트 **CA** 인증서와 모든 하위 **CA** 인증서가 있습니다.

절차

1. 구성을 수행할 디렉터리를 만듭니다.

```
[root@server]# mkdir ~/SmartCard/
```

2. 디렉터리로 이동합니다.

```
[root@server]# cd ~/SmartCard/
```

3.

PEM 형식으로 파일에 저장된 관련 **CA** 인증서를 가져옵니다. **CA** 인증서가 **DER**와 같은 다른 형식의 파일에 저장된 경우 **PEM** 형식으로 변환합니다. **IdM** 인증 기관 인증서는 **/etc/ipa/ca.crt** 파일에 있습니다.

DER 파일을 **PEM** 파일로 변환합니다.

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

4.

쉽게 구성을 수행할 디렉터리에 인증서를 복사합니다.

```
[root@server SmartCard]# cp /etc/ipa/ca.crt ~/SmartCard/
[root@server SmartCard]# cp /tmp/smartcard_ca.pem ~/SmartCard/
```

5.

선택적으로 외부 인증 기관의 인증서를 사용하는 경우 **openssl x509** 유틸리티를 사용하여 **PEM** 형식의 파일 내용을 보고 **Issuer** 및 **Subject** 값이 올바른지 확인합니다.

```
[root@server SmartCard]# openssl x509 -noout -text -in smartcard_ca.pem | more
```

6.

관리자의 권한을 사용하여 기본 제공된 **ipa-adviser** 유틸리티로 구성 스크립트를 생성합니다.

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-adviser config-server-for-smart-card-auth > config-
server-for-smart-card-auth.sh
```

config-server-for-smart-card-auth.sh 스크립트는 다음 작업을 수행합니다.

- **IdM Apache HTTP** 서버를 구성합니다.
- **KDC(Key Distribution Center)**에서 **Kerberos(PKINIT)**의 초기 인증용 공개 키 암호화가 가능합니다.
- 스마트 카드 인증 요청을 수락하도록 **IdM 웹 UI**를 구성합니다.

7.

스크립트를 실행하고 루트 **CA** 및 하위 **CA** 인증서가 포함된 **PEM** 파일을 인수로 추가합니다.

```
[root@server SmartCard]# chmod +x config-server-for-smart-card-auth.sh
[root@server SmartCard]# ./config-server-for-smart-card-auth.sh smartcard_ca.pem
ca.crt
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



참고

하위 **CA** 인증서 및 **CA** 또는 하위 **CA** 인증서가 만료되지 않았는지 루트 **CA** 인증서를 인수로 추가해야 합니다.

8.

선택적으로 사용자 인증서를 발급한 인증 기관이 **OCSP** 응답자(**Online Certificate Status Protocol**) 응답을 제공하지 않는 경우 **IdM** 웹 **UI**에 대한 인증을 비활성화해야 할 수 있습니다.

a.

`/etc/httpd/conf.d/ssl.conf` 파일에서 **SSL_OCSP_Enable** 매개변수를 **off** 로 설정합니다.

SSL_OCSP_Enable off

b.

변경 사항이 즉시 적용되도록 **Apache** 데몬(**httpd**)을 다시 시작합니다.

```
[root@server SmartCard]# systemctl restart httpd
```



주의

IdM CA에서 발급한 사용자 인증서만 사용하는 경우 **OCSP** 검사를 비활성화하지 마십시오. **OCSP** 응답자는 **IdM**의 일부입니다.

OCSP 검사를 계속 활성화하는 방법에 대한 자세한 내용은 사용자 인증서가 **OCSP** 서비스 요청을 수신하는 **CA**에 대한 정보가 포함되어 있지 않은 경우, [Apache mod_ssl 구성 옵션](#)의 **SSL_OCSPDefaultResponder** 지시문을 참조하십시오.

이제 서버가 스마트 카드 인증을 위해 구성됩니다.



참고

전체 토폴로지에서 스마트 카드 인증을 활성화하려면 각 **IdM** 서버에서 절차를 실행합니다.

7.2. 스마트 카드 인증을 위한 **IDM** 클라이언트 구성

이 섹션에서는 스마트 카드 인증을 위해 **IdM** 클라이언트를 구성하는 방법을 설명합니다. 인증에 스마트 카드를 사용하는 동안 연결하려는 각 **IdM** 시스템, 클라이언트 또는 서버에서 절차를 실행해야 합니다. 예를 들어 호스트 **A**에서 호스트 **B**로 **ssh** 연결을 활성화하려면 호스트 **B**에서 스크립트를 실행해야 합니다.

관리자로 다음 절차를 실행하여 스마트 카드 인증을 활성화합니다.

- **ssh** 프로토콜

자세한 내용은 [스마트 카드 인증을 사용하여 SSH 액세스 구성](#)을 참조하십시오.

- 콘솔 로그인
- **Gnome Display Manager (GDM)**
- **su** 명령

IdM 웹 UI 인증 절차는 필요하지 않습니다. **IdM** 웹 UI에 인증하려면 두 개의 호스트가 포함되며, 둘 다 **IdM** 클라이언트일 필요가 없습니다.

- 시스템 - **IdM** 도메인 외부에 있을 수 있음 - 브라우저가 실행 중인 경우
- **httpd**가 실행 중인 **IdM** 서버

다음 절차에서는 **IdM** 서버가 아닌 **IdM** 클라이언트에서 스마트 카드 인증을 구성한다고 가정합니다. 따라서 구성 스크립트를 생성하는 **IdM** 서버와 스크립트를 실행할 **IdM** 클라이언트의 두 개의 컴퓨터가 필요합니다.

사전 요구 사항

- 스마트 카드 인증을 위한 **IdM** 서버 구성에 설명된 대로 스마트 카드 인증을 위해 **IdM** 서버가 구성되어 있습니다.
- **IdM** 서버 및 **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.
- 루트 **CA** 인증서 및 하위 **CA** 인증서에 액세스할 수 있습니다.
- 원격 사용자가 성공적으로 로그인할 수 있도록 **--mkhomedir** 옵션과 함께 **IdM** 클라이언트를 설치했습니다. 홈 디렉터리를 생성하지 않으면 기본 로그인 위치는 **root**입니다.

절차

1.

IdM 서버에서 관리자 권한을 사용하여 **ipa-adviser** 로 구성 스크립트를 생성합니다.

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-adviser config-client-for-smart-card-auth > config-client-for-smart-card-auth.sh
```

config-client-for-smart-card-auth.sh 스크립트는 다음 작업을 수행합니다.

- 스마트 카드 데몬을 구성합니다.
 - 시스템 전체의 신뢰 저장소를 설정합니다.
 - 데스크탑에 스마트 카드 로그인을 허용하도록 **SSSD(System Security Services Daemon)**를 구성합니다.
- 2.
- IdM** 서버에서 **IdM** 클라이언트 시스템에서 선택한 디렉터리에 스크립트를 복사합니다.

```
[root@server SmartCard]# scp config-client-for-smart-card-auth.sh
root@client.idm.example.com:/root/SmartCard/
Password:
config-client-for-smart-card-auth.sh    100% 2419   3.5MB/s  00:00
```

3.

IdM 서버에서 이전 단계에서 사용된 것과 동일한 디렉터리에 편의를 위해 **PEM** 형식으로 **CA** 인증서 파일을 복사합니다.

```
[root@server SmartCard]# scp {smartcard_ca.pem,ca.crt}
root@client.idm.example.com:/root/SmartCard/
Password:
smartcard_ca.pem          100% 1237   9.6KB/s  00:00
ca.crt                    100% 2514  19.6KB/s  00:00
```

4.

클라이언트 시스템에서 스크립트를 실행하고 **CA** 인증서가 포함된 **PEM** 파일을 인수로 추가합니다.

```
[root@client SmartCard]# kinit admin
[root@client SmartCard]# chmod +x config-client-for-smart-card-auth.sh
[root@client SmartCard]# ./config-client-for-smart-card-auth.sh smartcard_ca.pem
ca.crt
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



참고

하위 **CA** 인증서 및 **CA** 또는 하위 **CA** 인증서가 만료되지 않았는지 루트 **CA** 인증서를 인수로 추가해야 합니다.

이제 클라이언트가 스마트 카드 인증을 위해 구성됩니다.

7.3. IDM 웹 UI에서 사용자 항목에 인증서 추가

다음 절차에서는 **IdM** 웹 **UI**의 사용자 항목에 외부 인증서를 추가하는 방법을 설명합니다.

전체 인증서를 업로드하는 대신 **IdM**의 사용자 항목에 인증서 매핑 데이터를 업로드할 수도 있습니다. 전체 인증서 또는 인증서 매핑 데이터를 포함하는 사용자 항목은 해당 인증서 매핑 규칙과 함께 사용하여

시스템 관리자를 위한 스마트 카드 인증 구성을 원활하게 수행할 수 있습니다. 자세한 내용은 참조하십시오.

스마트 카드에서 인증을 구성하기 위한 인증서 매핑 규칙입니다.



참고

IdM 인증 기관에서 사용자 인증서를 발급한 경우 인증서가 이미 사용자 항목에 저장되어 있으며 이 섹션을 건너뛸 수 있습니다.

사전 요구 사항

- 사용자 항목에 추가할 인증서가 있습니다. **You have the certificate that you want to add to the user entry.**

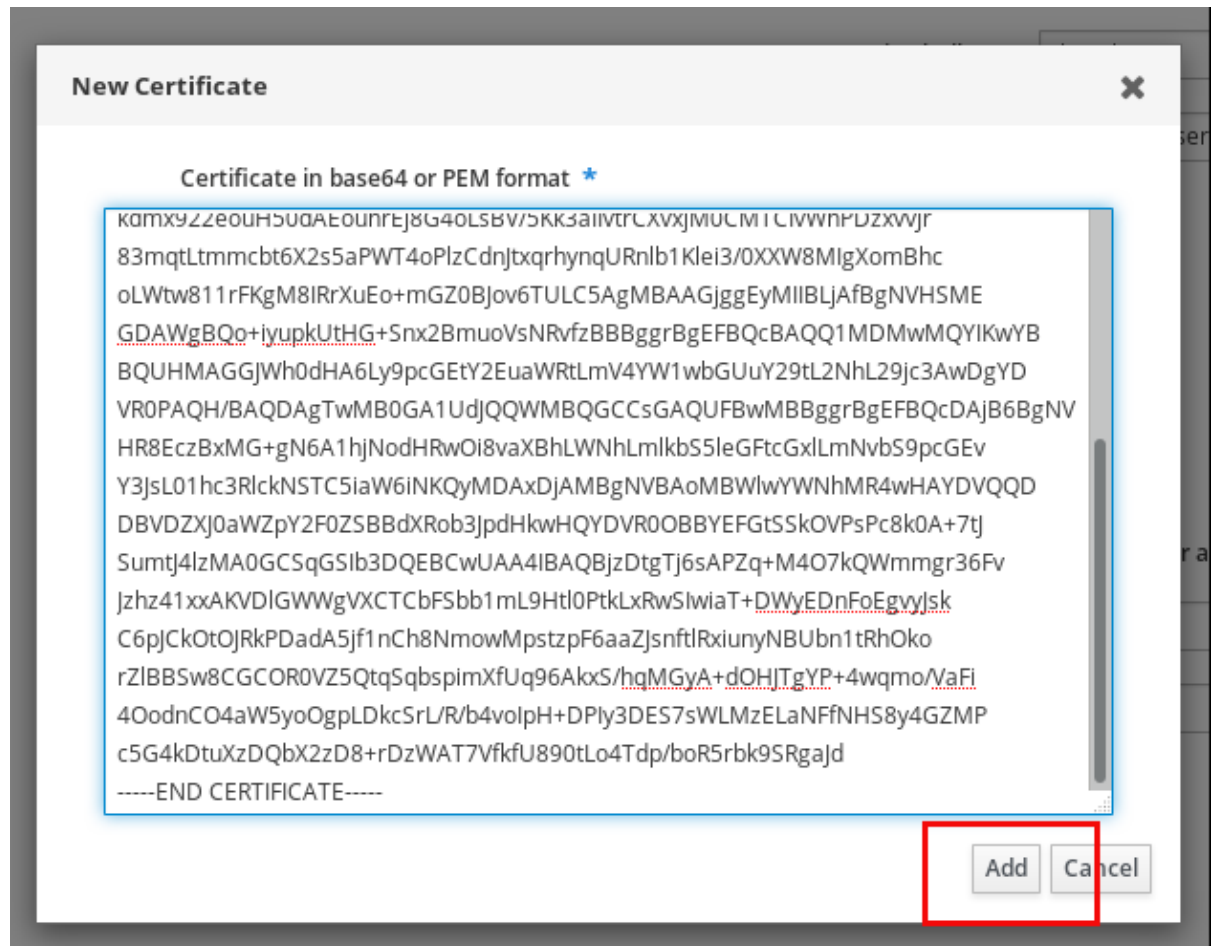
절차

1. 다른 사용자에게 인증서를 추가하려면 **IdM 웹 UI**에 관리자로 로그인합니다. 고유한 프로필에 인증서를 추가하는 경우 관리자의 인증 정보가 필요하지 않습니다.
2. 사용자 → 활성 사용자 → **sc_user** 로 이동합니다.
3. 인증서 옵션을 찾고 추가를 클릭합니다.
4. 명령줄 인터페이스의 **cat** 유틸리티 또는 텍스트 편집기를 사용하여 **PEM** 형식으로 인증서를 표시합니다.

```
[user@client SmartCard]$ cat testuser.crt
```

5. **CLI**의 인증서를 복사하여 웹 **UI**에서 열린 창에 붙여넣습니다.
6. 추가를 클릭합니다.

그림 7.1. IdM 웹 UI에 새 인증서 추가



이제 **sc_user** 항목에 외부 인증서가 포함됩니다.

7.4. IDM CLI에서 사용자 항목에 인증서 추가

다음 절차에서는 **IdM CLI**에서 사용자 항목에 외부 인증서를 추가하는 방법을 설명합니다.

전체 인증서를 업로드하는 대신 **IdM**의 사용자 항목에 인증서 매핑 데이터를 업로드할 수도 있습니다. 전체 인증서 또는 인증서 매핑 데이터를 포함하는 사용자 항목은 해당 인증서 매핑 규칙과 함께 사용하여 시스템 관리자를 위한 스마트 카드 인증 구성을 원활하게 수행할 수 있습니다. 자세한 내용은 참조하십시오.

스마트 카드에서 인증을 구성하기 위한 인증서 매핑 규칙입니다.



참고

IdM 인증 기관에서 사용자 인증서를 발급한 경우 인증서가 이미 사용자 항목에 저장되어 있으며 이 섹션을 건너뛸 수 있습니다.

사전 요구 사항

- 사용자 항목에 추가할 인증서가 있습니다. **You have the certificate that you want to add to the user entry.**

절차

1. 다른 사용자에게 인증서를 추가하려면 **IdM CLI**에 관리자로 로그인합니다.

```
[user@client SmartCard]$ kinit admin
```

고유한 프로필에 인증서를 추가하는 경우 관리자의 인증 정보가 필요하지 않습니다.

```
[user@client SmartCard]$ kinit sc_user
```

2. **ipa user-add-cert** 명령에서 예상되는 형식인 헤더 및 푸너가 제거되고 연결된 인증서가 포함된 환경 변수를 생성합니다.

```
[user@client SmartCard]$ export CERT=`openssl x509 -outform der -in testuser.crt | base64 -w0 -`
```

testuser.crt 파일의 인증서는 **PEM** 형식이어야 합니다.

3. **ipa user-add-cert** 명령을 사용하여 **sc_user**의 프로필에 인증서를 추가합니다.

```
[user@client SmartCard]$ ipa user-add-cert sc_user --certificate=$CERT
```

이제 **sc_user** 항목에 외부 인증서가 포함됩니다.

7.5. 스마트 카드 관리 및 사용을 위한 도구 설치

스마트 카드를 구성하려면 인증서를 생성하고 스마트 카드에 저장할 수 있는 도구가 필요합니다.

다음이 필요합니다.

- 인증서 관리에 도움이 되는 **gnutls-utils** 패키지를 설치합니다.
- 스마트 카드로 작업할 라이브러리 및 유틸리티 세트를 제공하는 **opensc** 패키지를 설치합니다.
- 스마트 카드 리더와 통신하는 **pcscd** 서비스를 시작합니다.

절차

1. **opensc** 및 **gnutls-utils** 패키지를 설치합니다.

```
# dnf -y install opensc gnutls-utils
```

2. **pcscd** 서비스를 시작합니다.

```
# systemctl start pcscd
```

pcscd 서비스가 실행 중인지 확인합니다.

7.6. 스마트 카드에 인증서 저장

이 섹션에서는 구성하는데 도움이 되는 **pkcs15-init** 툴을 사용하여 스마트 카드 구성에 대해 설명합니다.

- 스마트 카드 삭제
- 새로운 **pins** 및 선택적 **pin Unblocking Keys (PUKs)** 설정

- 스마트 카드에서 새 슬롯 생성
- 인증서, 개인 키 및 공개 키를 슬롯에 저장
- 스마트 카드 설정을 잠그십시오 (일부 스마트 카드에는 이러한 유형의 최종화가 필요합니다)

사전 요구 사항

- **pkcs15-init** 툴이 포함된 **opensc** 패키지가 설치됩니다.
- 자세한 내용은 [스마트 카드 관리 및 사용을 위한 도구](#) 설치를 참조하십시오.
- 카드가 리더에 삽입되고 컴퓨터에 연결되어 있습니다.
 - 스마트 카드에 저장할 개인 키, 공개 키 및 인증서가 있습니다. 이 절차에서는 **testuser.key**, **testuserpublic.key** 및 **testuser.crt** 가 개인 키, 공개 키 및 인증서에 사용되는 이름입니다.
 - 현재 스마트 카드 사용자 **Pin** 및 **Security Officer pin (SO-PIN)**

절차

1. 스마트 카드를 지우고 **Pin**을 사용하여 사용자를 인증합니다.

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

카드가 삭제되었습니다.

2. 스마트 카드를 초기화하고, 사용자 **Pin** 및 **PUK**를 설정하고, 보안 사무실자 **Pin** 및 **PUK**를 설정합니다.

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
  --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

`pkcs15-init` 톨은 스마트 카드에 새 슬롯을 만듭니다.

3.

슬롯의 레이블 및 인증 ID를 설정합니다.

```
$ pkcs15-init --store-pin --label testuser \
  --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

레이블은 사람이 읽을 수 있는 값(이 경우 `testuser`)으로 설정됩니다. `auth-id` 는 두 개의 16진수 값이어야 합니다. 이 경우 `01` 로 설정됩니다.

4.

스마트 카드의 새 슬롯에 개인 키를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
  --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

`--id` 에 지정하는 값은 개인 키 및 인증서를 저장할 때 동일해야 합니다. `--id` 에 대한 값을 지정하지 않으면 톨에 의해 더 복잡한 값이 계산되므로 자체 값을 더 쉽게 정의할 수 있습니다.

5.

스마트 카드의 새 슬롯에 인증서를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
  --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6.

(선택 사항) 스마트 카드의 새 슬롯에서 공개 키를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-public-key testuserpublic.key
  --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

공개 키가 개인 키 및/또는 인증서에 해당하는 경우 해당 개인 키 및/또는 인증서와 동일한 ID를 지정해야 합니다.

7.

(선택 사항) 일부 스마트 카드는 설정을 잠금하여 카드를 종료해야 합니다.

\$ pkcs15-init -F

이 단계에서 스마트 카드에는 새로 생성된 슬롯에 인증서, 개인 키 및 공개 키가 포함됩니다. 또한 사용자 Pin 및 PUK 및 Security Officer Pin 및 PUK를 생성했습니다.

7.7. 스마트 카드를 사용하여 IDM 로그인

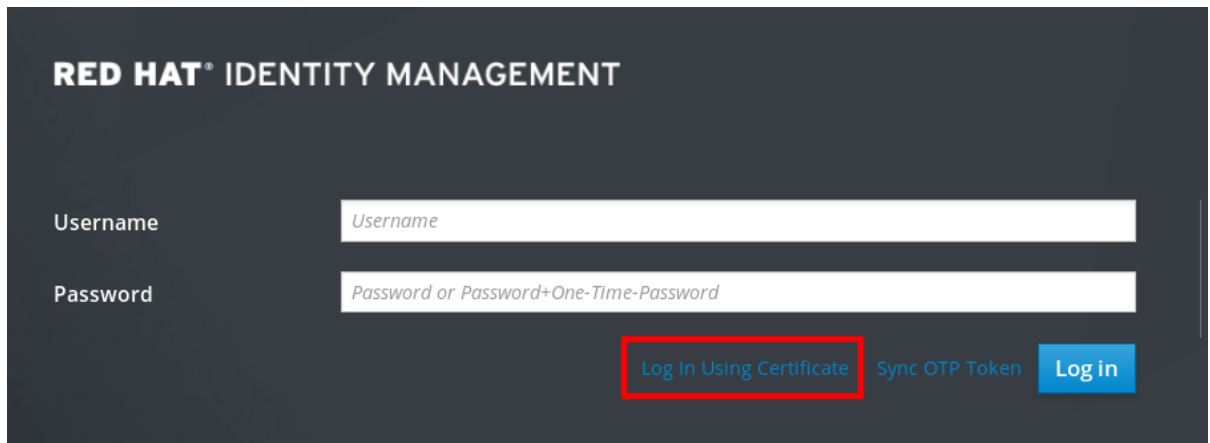
이 섹션에서는 IdM 웹 UI에 로그인하는 데 스마트 카드를 사용하는 방법에 대해 설명합니다.

사전 요구 사항

- 웹 브라우저는 스마트 카드 인증을 사용하도록 구성되어 있습니다.
- 스마트 카드 인증을 위해 IdM 서버가 구성되었습니다.
- 스마트 카드에 설치된 인증서는 IdM 서버에 알려져 있습니다.
- 스마트 카드 잠금을 해제하려면 pin이 필요합니다.
- 스마트 카드가 리더에 연결되어 있습니다.

절차

1. 브라우저에서 IdM 웹 UI를 엽니다.
2. 인증서를 사용하여 로그인을 클릭합니다.



RED HAT® IDENTITY MANAGEMENT

Username

Password

Log In Using Certificate Sync OTP Token **Log in**

3.

Password Required (암호 필요) 대화 상자가 열리면, 스마트 카드 잠금을 해제하는 **pin**을 추가하고 **OK** 버튼을 클릭합니다.

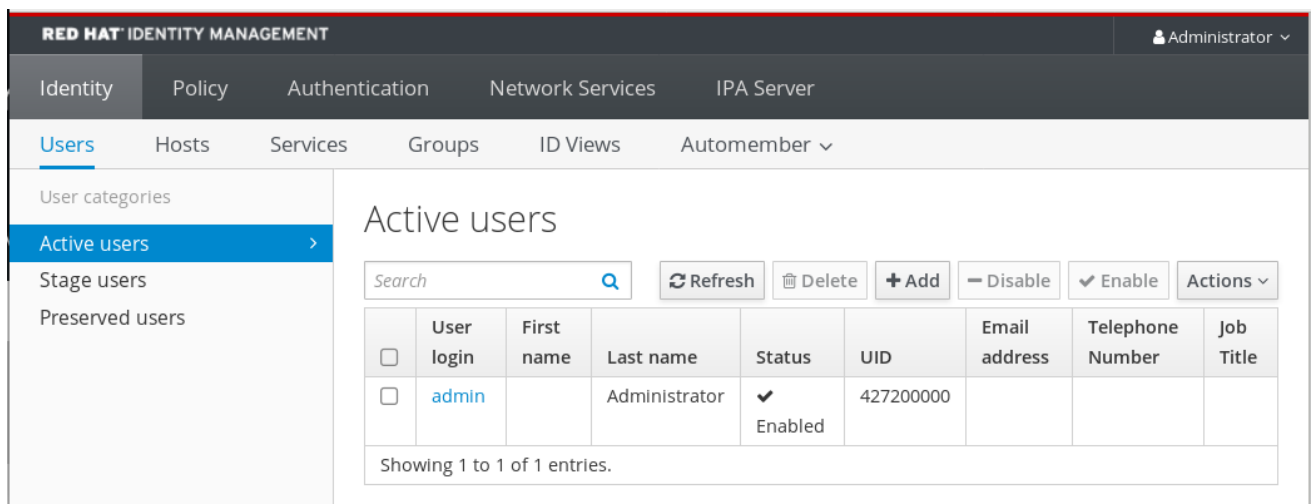
사용자 식별 요청 대화 상자가 열립니다.

스마트 카드에 두 개 이상의 인증서가 포함된 경우 드롭다운 목록에서 인증에 사용할 인증서를 선택합니다. 인증서 선택.

4.

OK 버튼을 클릭합니다.

이제 **IdM 웹 UI**에 성공적으로 로그인했습니다.



RED HAT® IDENTITY MANAGEMENT Administrator ▾

Identity Policy Authentication Network Services IPA Server

Users Hosts Services Groups ID Views Automember ▾

User categories

Active users >

Stage users

Preserved users

Active users

Search

☐	User login	First name	Last name	Status	UID	Email address	Telephone Number	Job Title
☐	admin		Administrator	✓ Enabled	427200000			

Showing 1 to 1 of 1 entries.

7.8. 스마트 카드 인증을 사용하여 GDM 액세스 구성

Gnome Desktop Manager (GDM)에는 인증이 필요합니다. 그러나 암호를 사용할 수는 있지만 인증에 스마트 카드를 사용할 수도 있습니다.

이 섹션에서는 **GDM**에 액세스하기 위한 스마트 카드 인증에 대해 설명합니다.

스마트 카드 인증 사용의 장점은 사용자 계정이 **ID** 관리 도메인의 일부인 경우 **TGT**(티켓 통합 티켓)도 얻는 것입니다.

사전 요구 사항

- 스마트 카드에는 인증서와 개인 키가 포함되어 있습니다.
- 사용자 계정은 **IdM** 도메인의 멤버입니다.
- 스마트 카드의 인증서는 다음을 통해 사용자 항목에 매핑됩니다.
 - 특정 사용자 항목에 인증서 할당. 자세한 내용은 [IdM 웹 UI의 사용자 항목에 인증서 추가](#) 또는 [IdM CLI의 사용자 항목에 인증서 추가](#)를 참조하십시오.
 - 계정에 적용되는 인증서 매핑 데이터입니다. 자세한 내용은 [스마트 카드에서 인증을 구성하기 위한 인증서 매핑 규칙](#)을 참조하십시오.

절차

1. 리더에 스마트 카드를 삽입합니다.
2. 스마트 카드 **Pin**을 입력합니다.
3. **Sign In** 을 클릭합니다.

RHEL 시스템에 성공적으로 로그인했으며 **IdM** 서버에서 제공하는 **TGT**가 있습니다.

검증 단계

- 터미널 창에서 **klist** 를 입력하고 결과를 확인합니다.

```
$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: example.user@REDHAT.COM

Valid starting    Expires          Service principal
04/20/2020 13:58:24 04/20/2020 23:58:24  krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 04/27/2020 08:58:15
```

7.9. 스마트 카드 인증을 사용하여 **SU** 액세스 구성

다른 사용자로 변경하려면 인증이 필요합니다. 암호 또는 인증서를 사용할 수 있습니다. 이 섹션에서는 **su** 명령과 함께 스마트 카드를 사용하는 방법을 설명합니다. 즉, **su** 명령을 입력한 후 스마트 카드 **Pin**을 입력하라는 메시지가 표시됩니다.

사전 요구 사항

- 스마트 카드에는 인증서와 개인 키가 포함되어 있습니다.
- 카드가 리더에 삽입되고 컴퓨터에 연결되어 있습니다.

절차

- 터미널 창에서 **su** 명령을 사용하여 다른 사용자로 변경합니다.

```
$ su - example.user
PIN for smart_card
```

구성이 성공하면 스마트 카드 **Pin**을 입력하라는 메시지가 표시됩니다.

8장. IDM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서 구성

이 시나리오에서는 다음 상황을 설명합니다.

- 배포는 **IdM(Identity Management)**과 **AD(Active Directory)** 간의 가장 큰 신뢰를 기반으로 합니다.
- 계정이 **AD**에 저장되어 있는 사용자에게 스마트 카드 인증을 허용하려고 합니다.
- 인증서는 **ADCS(Active Directory 인증서 서비스)**에 생성 및 저장됩니다.

구성은 다음 단계에서 수행됩니다.

- **Active Directory**에서 **IdM** 서버 및 클라이언트로 **CA** 및 사용자 인증서 복사
- **ADCS** 인증서를 사용하여 스마트 카드 인증을 위한 **IdM** 서버 및 클라이언트 구성
- **Pcabundle (PKCS#12)** 파일을 변환하여 인증서 및 개인 키를 스마트 카드에 저장할 수 있습니다.
- **sssd.conf** 파일에서 시간 제한 설정
- 스마트 카드 인증을 위한 인증서 매핑 규칙 생성

사전 요구 사항

- **IdM(Identity Management)** 및 **AD(Active Directory)** 신뢰가 설치됨

자세한 내용은 **IdM**과 **AD** 간의 신뢰 설치를 참조하십시오.
- **ADCS(Active Directory 인증서 서비스)**가 설치되고 사용자를 위한 인증서가 생성됩니다.

8.1. 스마트 카드 인증

스마트 카드는 카드에 저장된 인증서를 사용하여 개인 인증을 제공할 수 있는 물리적 장치입니다. 개인 인증은 사용자 암호와 동일한 방식으로 스마트 카드를 사용할 수 있음을 의미합니다.

개인 키 및 인증서 형식으로 스마트 카드에 사용자 자격 증명을 저장할 수 있으며 특수 소프트웨어 및 하드웨어가 해당 카드에 액세스하는 데 사용됩니다. 스마트 카드를 **reader** 또는 **USB** 소켓에 배치하고 암호를 제공하는 대신 스마트 카드의 **Pin** 코드를 제공합니다.

특정 **IdM** 클라이언트에서 스마트 카드 인증이 작동하는 방법을 구성할 수 있습니다.

- 사용자는 사용자 이름 및 암호로 또는 스마트 카드로 인증할 수 있습니다.
- 사용자는 스마트 카드로 인증할 수 있으며 암호는 허용되지 않습니다.
- 사용자는 제거 시 기능 잠금으로 로그아웃하는 데 스마트 카드를 사용할 수 있으며 암호는 허용되지 않습니다.

IdM(Identity Management)은 다음을 사용하여 스마트 카드 인증을 지원합니다.

- **IdM** 인증 기관에서 발급한 사용자 인증서. 자세한 내용은 [스마트 카드 인증을 위한 ID 관리](#) 구성을 참조하십시오.
- **ADCS** 인증 기관에서 발급한 사용자 인증서입니다. 자세한 내용은 [IdM의 스마트 카드 인증을 위해 ADCS에서 발급한 인증서](#) 구성을 참조하십시오.
- **RHEL** 시스템에서 생성된 로컬 인증 기관에서 발급한 사용자 인증서입니다. 자세한 내용은 [스마트 카드로 로컬 인증서 구성 및 가져오기](#)를 참조하십시오.
- 외부 인증 기관에서 발급한 사용자 인증서입니다.



참고

스마트 카드 인증을 사용하려면 하드웨어 요구 사항: [RHEL9의 스마트 카드 지원을 참조하십시오.](#)

8.2. 신뢰 구성 및 인증서 사용에 필요한 WINDOWS SERVER 설정

이 섹션에서는 **Windows Server**에서 구성해야 하는 사항에 대해 요약합니다.

- **ADCS(Active Directory 인증서 서비스)**가 설치됨
- 인증 기관 생성
- [선택 사항] 인증 기관 웹 등록을 사용하는 경우 인터넷 정보 서비스(IIS)를 구성해야 합니다.

인증서를 내보냅니다.

- **key**에는 **2048 비트** 이상이 있어야 합니다.
- 개인 키 포함
- 다음과 같은 형식의 인증서가 필요합니다: 개인 정보 교환 **Exchange#150- PKCS #12(.PFX)**
 - 인증서 개인 정보 보호 활성화

8.3. SFTP를 사용하여 ACTIVE DIRECTORY에서 인증서 복사

스마트 카드 인증 기능을 사용하려면 다음 인증서 파일을 복사해야 합니다.

- **CER** 형식의 루트 **CA** 인증서: **IdM** 서버의 **adcs-winservice-ca.cer**.

•

etcd 형식의 개인 키가 있는 사용자 인증서: **IdM** 클라이언트의 **aduser1.pfx**.



참고

이 절차에서는 **SSH** 액세스가 허용됩니다. **SSH**를 사용할 수 없는 경우 사용자는 **AD** 서버에서 **IdM** 서버 및 클라이언트로 파일을 복사해야 합니다.

절차

1.

IdM 서버에서 연결하고 **adcs-winservice-ca.cer** 루트 인증서를 **IdM** 서버에 복사합니다.

```
root@idmsvr ~]# sftp Administrator@winsvr.ad.example.com
Administrator@winsvr.ad.example.com's password:
Connected to Administrator@winsvr.ad.example.com.
sftp> cd <Path to certificates>
sftp> ls
adcs-winservice-ca.cer  aduser1.pfx
sftp>
sftp> get adcs-winservice-ca.cer
Fetching <Path to certificates>/adcs-winservice-ca.cer to adcs-winservice-ca.cer
<Path to certificates>/adcs-winservice-ca.cer      100% 1254  15KB/s 00:00
sftp quit
```

2.

IdM 클라이언트에서 연결하고 **aduser1.pfx** 사용자 인증서를 클라이언트에 복사합니다.

```
[root@client1 ~]# sftp Administrator@winsvr.ad.example.com
Administrator@winsvr.ad.example.com's password:
Connected to Administrator@winsvr.ad.example.com.
sftp> cd /<Path to certificates>
sftp> get aduser1.pfx
Fetching <Path to certificates>/aduser1.pfx to aduser1.pfx
<Path to certificates>/aduser1.pfx      100% 1254  15KB/s 00:00
sftp quit
```

이제 **CA** 인증서가 **IdM** 서버에 저장되고 사용자 인증서는 클라이언트 시스템에 저장됩니다.

8.4. ADCS 인증서를 사용하여 스마트 카드 인증을 위한 IDM 서버 및 클라이언트 구성

IdM 환경에서 스마트 카드 인증을 사용하려면 **IdM(Identity Management)** 서버 및 클라이언트를 구성해야 합니다. **IdM**에는 필요한 모든 변경을 수행하는 **ipa-adviser** 스크립트가 포함되어 있습니다.

- 필요한 패키지 설치
- **IdM 서버 및 클라이언트를 설정합니다.**
- **CA 인증서를 예상 위치에 복사**

IdM 서버에서 ipa-adviser 를 실행할 수 있습니다.

이 절차에서는 다음을 설명합니다.

- **IdM 서버:** 스마트 카드 인증을 위해 **ipa-adviser** 스크립트를 준비하여 **IdM 서버**를 구성합니다.
- **IdM 서버:** 스마트 카드 인증을 위해 **ipa-adviser** 스크립트를 준비하여 **IdM 클라이언트**를 구성합니다.
- **IdM 서버:** **AD** 인증서를 사용하여 **IdM 서버**에서 **ipa-adviser** 서버 스크립트 적용.
- 클라이언트 스크립트를 **IdM 클라이언트 시스템**으로 이동합니다.
- **IdM 클라이언트:** **AD** 인증서를 사용하여 **IdM 클라이언트**에서 **ipa-adviser** 클라이언트 스크립트 적용.

사전 요구 사항

- 인증서가 **IdM 서버**에 복사되었습니다.
- **Kerberos** 티켓을 받습니다.
- 관리 권한이 있는 사용자로 로그인합니다.

절차

1. IdM 서버에서 **ipa-adviser** 스크립트를 사용하여 클라이언트를 구성합니다.

```
[root@idmserver ~]# ipa-adviser config-client-for-smart-card-auth > sc_client.sh
```

2. IdM 서버에서 **ipa-adviser** 스크립트를 사용하여 서버 구성을 수행합니다.

```
[root@idmserver ~]# ipa-adviser config-server-for-smart-card-auth > sc_server.sh
```

3. IdM 서버에서 스크립트를 실행합니다.

```
[root@idmserver ~]# sh -x sc_server.sh adcs-winsrv-ca.cer
```

- IdM Apache HTTP 서버를 구성합니다.
- KDC(Key Distribution Center)에서 Kerberos(PKINIT)의 초기 인증용 공개 키 암호화가 가능합니다.
- 스마트 카드 인증 요청을 수락하도록 IdM 웹 UI를 구성합니다.

4. **sc_client.sh** 스크립트를 클라이언트 시스템에 복사합니다.

```
[root@idmserver ~]# scp sc_client.sh root@client1.idm.example.com:/root
Password:
sc_client.sh          100% 2857  1.6MB/s  00:00
```

5. Windows 인증서를 클라이언트 시스템에 복사합니다.

```
[root@idmserver ~]# scp adcs-winsrv-ca.cer root@client1.idm.example.com:/root
Password:
adcs-winsrv-ca.cer    100% 1254  952.0KB/s  00:00
```

6. 클라이언트 시스템에서 클라이언트 스크립트를 실행합니다.

```
[root@idmclient1 ~]# sh -x sc_client.sh adcs-winsrv-ca.cer
```

CA 인증서는 IdM 서버 및 클라이언트 시스템의 올바른 형식으로 설치되며 다음 단계는 사용자 인증서를 스마트 카드 자체에 복사하는 것입니다.

8.5. PCABUNDLE 파일 변환

Pcabundle (PKCS#12) 파일을 스마트 카드에 저장하기 전에 다음을 수행해야 합니다.

- 파일을 PEM 형식으로 변환
- 개인 키와 인증서를 두 개의 서로 다른 파일로 추출

사전 요구 사항

- Ptekton 파일이 IdM 클라이언트 시스템에 복사됩니다.

절차

1. IdM 클라이언트에서 PEM 형식으로 되어 있습니다.

```
[root@idmclient1 ~]# openssl pkcs12 -in aduser1.pfx -out aduser1_cert_only.pem -
clcerts -nodes
Enter Import Password:
```

2. 키를 별도의 파일에 추출합니다.

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -nocerts -out adduser1.pem >
aduser1.key
```

3. 공용 인증서를 별도의 파일에 추출합니다.

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -clcerts -nokeys -out
aduser1_cert_only.pem > aduser1.crt
```

이 시점에서 **aduser1.key** 및 **aduser1.crt** 를 스마트 카드에 저장할 수 있습니다.

8.6. 스마트 카드 관리 및 사용을 위한 도구 설치

스마트 카드를 구성하려면 인증서를 생성하고 스마트 카드에 저장할 수 있는 도구가 필요합니다.

다음이 필요합니다.

- 인증서 관리에 도움이 되는 **gnutls-utils** 패키지를 설치합니다.
- 스마트 카드로 작업할 라이브러리 및 유틸리티 세트를 제공하는 **opensc** 패키지를 설치합니다.
- 스마트 카드 리더와 통신하는 **pcscd** 서비스를 시작합니다.

절차

1. **opensc** 및 **gnutls-utils** 패키지를 설치합니다.

```
# dnf -y install opensc gnutls-utils
```

2. **pcscd** 서비스를 시작합니다.

```
# systemctl start pcscd
```

pcscd 서비스가 실행 중인지 확인합니다.

8.7. 스마트 카드에 인증서 저장

이 섹션에서는 구성하는데 도움이 되는 **pkcs15-init** 툴을 사용하여 스마트 카드 구성에 대해 설명합니다.

- 스마트 카드 삭제

- 새로운 pins 및 선택적 pin Unblocking Keys (PUKs) 설정
- 스마트 카드에서 새 슬롯 생성
- 인증서, 개인 키 및 공개 키를 슬롯에 저장
- 스마트 카드 설정을 잠그십시오 (일부 스마트 카드에는 이러한 유형의 최종화가 필요합니다)

사전 요구 사항

- **pkcs15-init** 툴이 포함된 **opensc** 패키지가 설치됩니다.

자세한 내용은 [스마트 카드 관리 및 사용을 위한 도구](#) 설치를 참조하십시오.
- 카드가 리더에 삽입되고 컴퓨터에 연결되어 있습니다.
- 스마트 카드에 저장할 개인 키, 공개 키 및 인증서가 있습니다. 이 절차에서는 **testuser.key**, **testuserpublic.key** 및 **testuser.crt** 가 개인 키, 공개 키 및 인증서에 사용되는 이름입니다.
- 현재 스마트 카드 사용자 **Pin** 및 **Security Officer pin (SO-PIN)**

절차

1. 스마트 카드를 지우고 **Pin**을 사용하여 사용자를 인증합니다.

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

카드가 삭제되었습니다.

2. 스마트 카드를 초기화하고, 사용자 **Pin** 및 **PUK**를 설정하고, 보안 사무실자 **Pin** 및 **PUK**를 설

정합니다.

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
  --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

`speeds ks15-init` 톨은 스마트 카드에 새 슬롯을 만듭니다.

3.

슬롯의 레이블 및 인증 ID를 설정합니다.

```
$ pkcs15-init --store-pin --label testuser \
  --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

레이블은 사람이 읽을 수 있는 값(이 경우 `testuser`)으로 설정됩니다. `auth-id` 는 두 개의 16진수 값이어야 합니다. 이 경우 `01` 로 설정됩니다.

4.

스마트 카드의 새 슬롯에 개인 키를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
  --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

`--id` 에 지정하는 값은 개인 키 및 인증서를 저장할 때 동일해야 합니다. `--id` 에 대한 값을 지정하지 않으면 톨에 의해 더 복잡한 값이 계산되므로 자체 값을 더 쉽게 정의할 수 있습니다.

5.

스마트 카드의 새 슬롯에 인증서를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_cert \
  --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6.

(선택 사항) 스마트 카드의 새 슬롯에서 공개 키를 저장하고 레이블을 지정합니다.

```
$ pkcs15-init --store-public-key testuserpublic.key
--label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



참고

공개 키가 개인 키 및/또는 인증서에 해당하는 경우 해당 개인 키 및/또는 인증서와 동일한 ID를 지정해야 합니다.

7.

(선택 사항) 일부 스마트 카드는 설정을 잠금하여 카드를 종료해야 합니다.

```
$ pkcs15-init -F
```

이 단계에서 스마트 카드에는 새로 생성된 슬롯에 인증서, 개인 키 및 공개 키가 포함됩니다. 또한 사용자 Pin 및 PUK 및 Security Officer Pin 및 PUK를 생성했습니다.

8.8. SSSD.CONF에서 시간 제한 설정

스마트 카드 인증서로 인증하는 데 SSSD에서 사용하는 기본 시간 초과보다 오래 걸릴 수 있습니다. 제한 시간 초과를 다음을 통해 발생할 수 있습니다.

- 느린 리더
- 전달은 가상 환경으로 물리적 장치를 형성합니다.
- 스마트 카드에 저장된 인증서가 너무 많습니다.
- OCSP(Online Certificate Status Protocol)가 인증서를 확인하는 데 사용되는 경우 OCSP(Online Certificate Status Protocol) 응답 속도

이 경우 **sssd.conf** 파일에서 다음과 같은 시간 초과를 늘릴 수 있습니다(예: 60초).

- **p11_child_timeout**

- **krb5_auth_timeout**

사전 요구 사항

- **root**로 로그인해야 합니다.

절차

1. **sssd.conf** 파일을 엽니다.

```
[root@idmclient1 ~]# vim /etc/sss/sss.conf
```

2. **p11_child_timeout**의 값을 변경합니다.

```
[pam]
p11_child_timeout = 60
```

3. **NetNamespace 5_auth_timeout**의 값을 변경합니다.

```
[domain/IDM.EXAMPLE.COM]
krb5_auth_timeout = 60
```

4. 설정을 저장합니다.

이제 스마트 카드와의 상호 작용은 시간 초과로 인증이 실패하기 전에 **1 분 (60 초)** 동안 실행할 수 있습니다.

8.9. 스마트 카드 인증을 위한 인증서 매핑 규칙 생성

AD(Active Directory) 및 **IdM(Identity Management)**에 계정이 있는 사용자에게 하나의 인증서를 사용하려면 **IdM** 서버에 인증서 매핑 규칙을 생성할 수 있습니다.

이러한 규칙을 생성하면 사용자는 두 도메인에서 스마트 카드로 인증할 수 있습니다.

인증서 매핑 규칙에 대한 자세한 내용은 [스마트 카드에서 인증을 구성하기 위한 인증서 매핑 규칙을 참](#)

조하십시오.

9장. IDENTITY MANAGEMENT에서 인증서 매핑 규칙 구성

9.1. 스마트 카드에서 인증을 구성하기 위한 인증서 매핑 규칙

인증서 매핑 규칙은 **IdM(Identity Management)** 관리자가 특정 사용자의 인증서에 액세스할 수 없는 경우 사용자가 시나리오에서 인증서를 사용하여 인증할 수 있는 편리한 방법입니다. 이러한 액세스 부족은 일반적으로 인증서가 외부 인증 기관에서 발행되었다는 사실 때문에 발생합니다. 특수 사용 사례는 **IdM** 도메인이 신뢰 관계에 있는 **AD(Active Directory)**의 인증서 시스템에서 발급한 인증서로 표시됩니다.

IdM 환경이 스마트 카드를 사용하는 많은 사용자가 큰 경우에도 인증서 매핑 규칙이 편리합니다. 이 경우 전체 인증서를 추가하는 것이 복잡할 수 있습니다. 주체와 발급자는 대부분의 시나리오에서 예측할 수 없으므로 전체 인증서보다 먼저 더 쉽게 추가할 수 있습니다. 시스템 관리자는 인증서 매핑 규칙을 생성하고 특정 사용자에게 인증서를 발급하기 전에 사용자 항목에 인증서 매핑 데이터를 추가할 수 있습니다. 인증서가 발급되면 전체 인증서가 사용자 항목에 아직 업로드되지 않은 경우에도 인증서를 사용하여 로그인할 수 있습니다.

또한 인증서를 정기적으로 갱신해야 하므로 인증서 매핑 규칙은 관리 오버헤드를 줄입니다. 사용자 인증서가 갱신되면 관리자는 사용자 항목을 업데이트할 필요가 없습니다. 예를 들어 매핑이 **Subject** 및 **Issuer** 값을 기반으로 하고 새 인증서에는 이전 인증서와 동일한 제목 및 발급자가 있는 경우 매핑은 계속 적용됩니다. 대조적으로 전체 인증서가 사용된 경우 관리자는 이전 인증서를 교체하기 위해 사용자 항목에 새 인증서를 업로드해야 합니다.

인증서 매핑을 설정하려면 다음을 수행합니다.

1. 관리자는 인증서 매핑 데이터(일반적으로 발급자 및 주체) 또는 전체 인증서를 사용자 계정으로 로드해야 합니다.
2. 관리자는 사용자에게 **IdM**에 성공적으로 로그인할 수 있도록 인증서 매핑 규칙을 생성해야 합니다.
 - a. **whose account contains a certificate mapping data entry**
 - b. 인증서 매핑 데이터 항목이 인증서의 정보와 일치하는 **whose certificate mapping data entry matches the information on the certificate**

매핑 규칙을 구성하는 개별 구성 요소와 이를 가져와서 사용하는 방법에 대한 자세한 내용은 **IdM의 ID 매핑 규칙 구성 및 일치하는 규칙에서 사용할 수 있도록 인증서의 발행자 가져오기**를 참

조하십시오.

이후, 최종 사용자가 파일 **시스템** 또는 **스마트 카드**에 저장된 인증서를 제공하면 인증이 성공적으로 수행됩니다.

9.1.1. Active Directory 도메인을 사용하는 신뢰에 대한 인증서 매핑 규칙

이 섹션에서는 **IdM** 배포가 **Active Directory(AD)** 도메인과 신뢰 관계에 있는 경우 가능한 다양한 인증서 매핑 사용 사례에 대해 간단히 설명합니다.

인증서 매핑 규칙은 신뢰할 수 있는 **AD** 인증서 시스템에서 발급한 스마트 카드 인증서가 있는 사용자의 **IdM** 리소스에 액세스할 수 있는 편리한 방법입니다. **AD** 구성에 따라 다음 시나리오가 가능합니다.

- **AD**에서 인증서를 발행하지만 사용자와 인증서가 **IdM**에 저장된 경우, **IdM** 측에서 인증 요청의 매핑 및 전체 처리가 수행됩니다. 이 시나리오 구성에 대한 자세한 내용은 **IdM에 저장된 사용자의 인증서 매핑** 구성을 참조하십시오.
- 사용자가 **AD**에 저장된 경우 인증 요청 처리가 **AD**에서 수행됩니다. 세 가지 하위 사례가 있습니다.
 - **AD** 사용자 항목에는 전체 인증서가 포함되어 있습니다. 이 시나리오에서 **IdM**을 구성하는 방법에 대한 자세한 내용은 **AD 사용자 항목에 전체 인증서가 포함된 사용자의 인증서 매핑** 구성을 참조하십시오.
 - **AD**는 사용자 인증서를 사용자 계정에 매핑하도록 구성됩니다. 이 경우 **AD** 사용자 항목에 전체 인증서가 포함되어 있지 않지만 대신 **altSecurityIdentities** 라는 속성이 포함됩니다. 이 시나리오에서 **IdM**을 구성하는 방법에 대한 자세한 내용은 **AD가 사용자 인증서를 사용자 계정에 매핑하도록 구성된 경우 인증서** 매핑 구성을 참조하십시오.
 - **AD** 사용자 항목에는 전체 인증서와 매핑 데이터가 포함되어 있지 않습니다. 이 경우 유일한 해결책은 **ipa idoverrideuser-add** 명령을 사용하여 **IdM**의 **AD** 사용자 ID 덮어쓰기에 전체 인증서를 추가하는 것입니다. 자세한 내용은 **AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑** 구성을 참조하십시오.

9.1.2. IdM의 ID 매핑 규칙 구성 요소

이 섹션에서는 **IdM**의 **ID 매핑** 규칙 구성 요소 및 구성 방법에 대해 설명합니다. 각 구성 요소에는 재정의할 수 있는 기본값이 있습니다. 웹 **UI** 또는 **CLI**에서 구성 요소를 정의할 수 있습니다. **CLI**에서 **ID** 매핑

규칙은 **ipa certmaprule-add** 명령을 사용하여 생성됩니다.

매핑 규칙

매핑 규칙 구성 요소는 인증서를 하나 이상의 사용자 계정과 연결(또는 *매핑*)합니다. 규칙은 인증서를 의도한 사용자 계정과 연결하는 **LDAP** 검색 필터를 정의합니다.

다른 **CA**(인증 기관)에서 발급한 인증서에는 다른 속성이 있을 수 있으며 다른 도메인에서 사용할 수 있습니다. 따라서 **IdM**은 매핑 규칙을 무조건 적용하지 않고 적절한 인증서에만 적용합니다. 적절한 인증서는 *일치하는 규칙*을 사용하여 정의됩니다.

매핑 규칙 옵션을 비워 두면 인증서가 **DER** 인코딩 바이너리 파일로 **userCertificate** 속성에서 검색됩니다.

--maprule 옵션을 사용하여 **CLI**에서 매핑 규칙을 정의합니다.

일치 규칙

일치하는 규칙 구성 요소는 매핑 규칙을 적용할 인증서를 선택합니다. 기본 일치 규칙은 **digitalSignature** 키 사용과 함께 인증서와 일치하며 **clientAuth**는 확장된 키 사용량과 일치합니다.

--matchrule 옵션을 사용하여 **CLI**에서 일치하는 규칙을 정의합니다.

도메인 목록

domain list는 **ID** 매핑 규칙을 처리할 때 **IdM**에서 사용자를 검색할 **ID** 도메인을 지정합니다. 옵션을 지정하지 않으면 **IdM**에서 **IdM** 클라이언트가 속하는 로컬 도메인에서만 사용자를 검색합니다.

domain 옵션을 사용하여 **CLI**에서 도메인을 정의합니다.

우선 순위

인증서에 여러 규칙이 적용되는 경우 우선 순위가 가장 높은 규칙이 우선합니다. 다른 모든 규칙은 무시됩니다.

-

숫자 값이 낮으면 **ID** 매핑 규칙의 우선 순위가 높습니다. 예를 들어 우선 순위 **1**이 있는 규칙은 우선 순위가 **2**인 규칙보다 우선 순위가 높습니다.

- 규칙에 우선순위 값이 정의되어 있지 않으면 우선순위가 가장 낮습니다.

--priority 옵션을 사용하여 CLI에서 매핑 규칙 우선 순위를 정의합니다.

인증서 매핑 규칙 예

CLI를 사용하여 **EXAMPLE.ORG** 조직의 스마트 카드 **CA** 에서 발행한 인증서에 대한 인증을 허용하는 **simple_rule** 이라는 인증서의 인증서 매핑 규칙을 정의하려면 **IdM**의 사용자 계정의 **certmapdata** 항목과 일치하는 한 해당 인증서의 인증이 허용됩니다.

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<l>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

9.1.3. 일치하는 규칙에 사용할 인증서에서 발급자 가져오기

이 절차에서는 인증서에서 발급자 정보를 가져와서 인증서 매핑 규칙의 일치하는 규칙에 붙여넣을 수 있도록 하는 방법을 설명합니다. 일치하는 규칙에 필요한 발급자 형식을 가져오려면 **openssl x509** 유틸리티를 사용합니다.

사전 요구 사항

- .pem 또는 .crt 형식의 사용자 인증서가 있습니다.

절차

1. 인증서에서 사용자 정보를 가져옵니다. 다음을 사용하여 **openssl x509** 인증서 표시 및 서명 유틸리티를 사용합니다.
 - 인코딩된 요청 버전의 출력을 방지하기 위한 **-noout** 옵션
 - 발급자 이름을 출력하는 **-issuer** 옵션
 - 인증서를 읽을 입력 파일 이름을 지정하는 **-in** 옵션
 - 가장 구체적인 상대 고유 이름(**RDN**)으로 출력을 표시하는 **RFC2253** 값이 포함된 -

nameopt 옵션

입력 파일에 ID 관리 인증서가 포함된 경우 명령 출력은 발급자가 다음 정보를 사용하여 정의되었음을 표시합니다.

```
# openssl x509 -noout -issuer -in idm_user.crt -nameopt RFC2253
issuer=CN=Certificate Authority,O=REALM.EXAMPLE.COM
```

입력 파일에 Active Directory 인증서가 포함된 경우 명령 출력은 도메인 구성 요소 정보를 사용하여 Issuer가 정의되었음을 표시합니다.

```
# openssl x509 -noout -issuer -in ad_user.crt -nameopt RFC2253
issuer=CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM
```

2.

필요한 경우 인증서 발행자가 ad.example.com 도메인의 추출된 AD-ignition2012R2-CA 여야 함을 지정하는 일치 규칙을 기반으로 CLI에 새 매핑 규칙을 생성하려면 인증서의 주체가 IdM의 사용자 계정의 certmapdata 항목과 일치해야 합니다.

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
```

9.1.4. 추가 리소스

- **ss-certmap(5)** 매뉴얼 페이지를 참조하십시오.

9.2. IDM에 저장된 사용자의 인증서 매핑 구성

이 사용자 사례에서는 인증 인증이 IdM에 저장되는 경우 시스템 관리자가 IdM에서 인증서 매핑을 활성화하기 위해 수행해야 하는 단계를 설명합니다. 이 섹션에서는 다음을 설명합니다.

- 매핑 규칙에 지정된 조건과 일치하는 인증서를 사용하여 IdM 사용자가 인증서 매핑 규칙을 설정하는 방법 및 인증서 매핑 데이터 항목이 IdM에 인증할 수 있도록 하는 방법.
- 인증서 매핑 데이터 항목에 지정된 값이 포함된 한 사용자가 여러 인증서를 사용하여 인증할 수 있도록 IdM 사용자 항목에 인증서 매핑 데이터를 입력하는 방법.

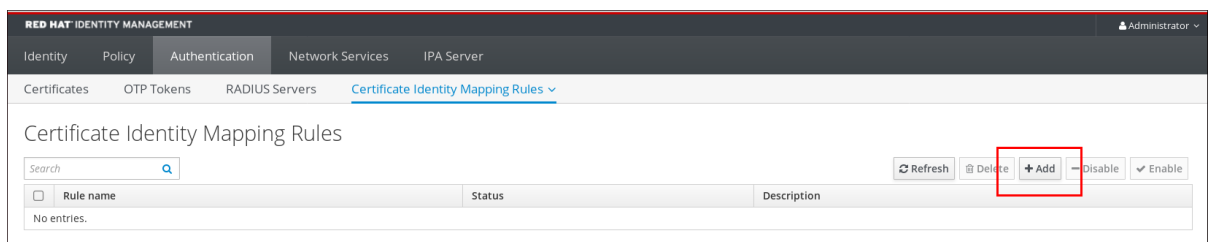
사전 요구 사항

- 사용자에게 **IdM**에 계정이 있습니다.
- 관리자에게는 사용자 항목에 추가할 전체 인증서 또는 인증서 매핑 데이터가 있습니다.

9.2.1. IdM 웹 UI에 인증서 매핑 규칙 추가

1. **IdM 웹 UI**에 관리자로 로그인합니다.
2. 인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.
3. 추가를 클릭합니다.

그림 9.1. IdM 웹 UI에 새 인증서 매핑 규칙 추가



4. 규칙 이름을 입력합니다.
5. 매핑 규칙을 입력합니다. 예를 들어, **IdM**에서 제공되는 인증서의 **Issuer** 및 **Subject** 항목을 검색하고 제공된 인증서의 두 항목에 있는 정보를 인증하기로 결정하십시오.

```
(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
```

6. 일치하는 규칙을 입력합니다. 예를 들어 **EXAMPLE.ORG** 조직의 **Smart Card CA** 에서 발급한 인증서만 허용하려면 사용자를 **IdM**에 인증하십시오.

```
<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
```

그림 9.2. IdM 웹 UI에서 인증서 매핑 규칙 세부 정보 입력

7. 대화 상자 하단에 있는 **Add** (추가)를 클릭하여 규칙을 추가하고 상자를 닫습니다.
8. **SSSD(System Security Services Daemon)**는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

이제 스마트 카드 인증서에서 찾은 매핑 규칙에 지정된 데이터 유형을 **IdM** 사용자 항목의 인증서 매핑 데이터와 비교하는 인증서 매핑 규칙을 설정했습니다. 일치 항목을 찾으려면 일치하는 사용자를 인증합니다.

9.2.2. IdM CLI에 인증서 매핑 규칙 추가

1. 관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2. 매핑 규칙을 입력하고 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. 예를 들어, 표시된 인증서의 **Issuer** 및 **Subject** 항목을 **IdM** 검색을 수행하고 제공되는 인증서의 두 항목에 있는 정보에 대해 인증하기로 결정하려면 **EXAMPLE.ORG** 조직의 **Smart Card CA** 에서 발급한 인증서만 인식합니다.

```
# ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card
CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})'
```

Added Certificate Identity Mapping Rule "rule_name"

```

Rule name: rule_name
Mapping rule: (ipacertmapdata=X509:<|>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})
Matching rule: <ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
Enabled: TRUE

```

3.

SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

이제 스마트 카드 인증서에서 찾은 매핑 규칙에 지정된 데이터 유형을 **IdM** 사용자 항목의 인증서 매핑 데이터와 비교하는 인증서 매핑 규칙을 설정했습니다. 일치 항목을 찾으면 일치하는 사용자를 인증합니다.

9.2.3. IdM 웹 UI에서 사용자 항목에 인증서 매핑 데이터 추가

1.

IdM 웹 UI에 관리자로 로그인합니다.

2.

사용자 → 활성 사용자 → **idm_user** 로 이동합니다.

3.

인증서 매핑 데이터 옵션을 찾아 추가를 클릭합니다.

4.

If you have the certificate of **idm_user** at your responsibility:

a.

명령줄 인터페이스에서 **cat** 유틸리티 또는 텍스트 편집기를 사용하여 인증서를 표시합니다.

```

[root@server ~]# cat idm_user_certificate.pem
-----BEGIN CERTIFICATE-----
MIIFFTCCA/2gAwIBAgIBejANBgkqhkiG9w0BAQsFADA6MRgwFgYDVQQKDA9JRE
0u
RVhBTvBMRs5DT00xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0
x
ODA5MDIxODE1MzlaFw0yMDA5MDIxODE1MzlaMCwxGDAWBgNVBAoMD0IETS5F
WEFN
[...output truncated...]

```

b.

인증서를 복사합니다.

c.

IdM 웹 UI에서 인증서 옆에 있는 **Add** (추가)를 클릭하고 인증서가 열리는 창에 붙여넣습니다.

그림 9.3. 사용자의 인증서 매핑 데이터 추가: certificate

User: demouser
demouser is a member of:

Settings | User Groups | Netgroups | Roles | HBAC Rules | Sudo Rules

Refresh Revert Save Actions

Identity Settings

Job Title:
 First name *: Demo
 Last name *: User
 Full name *: Demo User
 Display name: Demo User
 Initials: DU
 GECOS: Demo User
 Class:

Account Settings

User login: demouser
 Password: *****
 Password expiration: 2016-07-14 10:14:41Z
 UID: 373000005
 GID: 373000005
 Principal alias: demouser@IDM.EXAMPLE.COM

 Kerberos principal expiration: YYYY-MM-DD hh : mn UTC
 Login shell: /bin/sh
 Home directory: /home/demouser
 SSH public keys:
 Certificates:

대안으로, **idm_user**의 인증서가 현재 폐기에 없지만 발급자 및 인증서의 주체를 알고 있는 경우 발급자 및 제목의 라디오 버튼을 확인하고 두 개의 각 상자에 값을 입력합니다.

그림 9.4. 사용자의 인증서 매핑 데이터 추가: issuer 및 subject

GID: 1997000009

Add Certificate Mapping Data

☐ Certificate mapping data
 Certificate mapping data

☒ Issuer and subject

Issuer *: O=EXAMPLE.ORG,CN=Smart Card CA

Subject *: CN=test,O=EXAMPLE.ORG

Certificate mapping

5.

추가를 클릭합니다.

6.

필요한 경우 **.pem** 형식으로 전체 인증서에 액세스할 수 있는 경우 사용자 및 인증서가 연결되어 있는지 확인합니다.

a.

sss_cache 유틸리티를 사용하여 **SSSD** 캐시에서 **idm_user** 레코드를 무효화하고 **idm_user** 정보를 다시 로드합니다.

```
# sss_cache -u idm_user
```

b.

IdM 사용자의 인증서가 포함된 파일 이름으로 **ipa certmap-match** 명령을 실행합니다.

```
# ipa certmap-match idm_user_cert.pem
```

```
-----
```

```
1 user matched
```

```
-----
```

```
Domain: IDM.EXAMPLE.COM
```

```
User logins: idm_user
```

```
-----
```

```
Number of entries returned 1
```

```
-----
```

출력은 이제 **idm_user**에 인증서 매핑 데이터를 추가하고 해당 매핑 규칙이 존재하는지 확인합니다. 즉 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 **idm_user**로 인증할 수 있습니다.

9.2.4. IdM CLI에서 사용자 항목에 인증서 매핑 데이터 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

사용 중인 **idm_user** 인증서가 있는 경우 **ipa user-add-cert** 명령을 사용하여 사용자 계정에 인증서를 추가합니다.

```
# CERT=`cat idm_user_cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n\'`
# ipa user-add-certmapdata idm_user --certificate $CERT
```

또는 **idm_user**의 인증서가 현재 사용되었지만 **Issuer** 및 **Identitym_user**의 주체 인증서를

알고 있는 경우:

```
# ipa user-add-certmapdata idm_user --subject "O=EXAMPLE.ORG,CN=test" --issuer
"CN=Smart Card CA,O=EXAMPLE.ORG"
-----
Added certificate mappings to user "idm_user"
-----
User login: idm_user
Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

3.

필요한 경우 .pem 형식으로 전체 인증서에 액세스할 수 있는 경우 사용자 및 인증서가 연결되어 있는지 확인합니다.

a.

sss_cache 유틸리티를 사용하여 **SSSD** 캐시에서 **idm_user** 레코드를 무효화하고 **idm_user** 정보를 다시 로드합니다.

```
# sss_cache -u idm_user
```

b.

IdM 사용자의 인증서가 포함된 파일 이름으로 **ipa certmap-match** 명령을 실행합니다.

```
# ipa certmap-match idm_user_cert.pem
-----
1 user matched
-----
Domain: IDM.EXAMPLE.COM
User logins: idm_user
-----
Number of entries returned 1
-----
```

출력은 이제 **idm_user**에 인증서 매핑 데이터를 추가하고 해당 매핑 규칙이 존재하는지 확인합니다. 즉 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 **idm_user**로 인증할 수 있습니다.

```
/include::modules/identity-management/proc_add-certmapdata-to-user.adoc[leveloffset=+1]
```

9.3. AD 사용자 항목이 전체 인증서가 포함되어 있는 사용자에게 대한 인증서 매핑 구성

이 사용자 사례에서는 **IdM** 배포가 **AD(Active Directory)**와 함께 신뢰되는 경우 **IdM**에서 인증서 매핑을 활성화하는 데 필요한 단계를 설명하고, **AD**의 사용자 항목에 전체 인증서가 포함되어 있습니다.

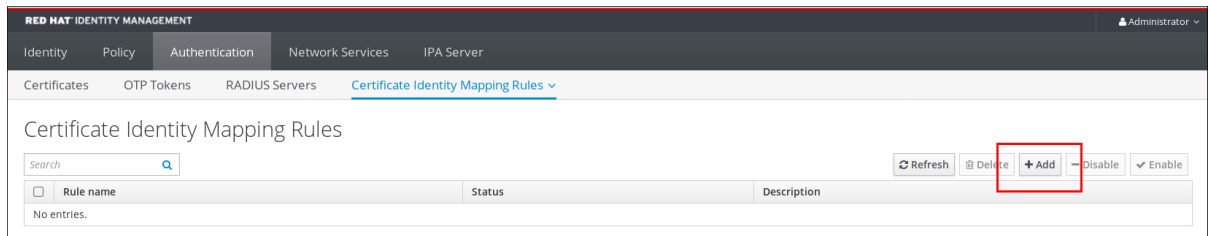
사전 요구 사항

- 사용자에게 **IdM**에 계정이 없습니다.
- 사용자에는 인증서를 포함하는 **AD** 계정이 있습니다.
- **IdM** 관리자는 **IdM** 인증서 매핑 규칙을 기반으로 할 수 있는 데이터에 액세스할 수 있습니다.

9.3.1. IdM 웹 UI에 인증서 매핑 규칙 추가

1. **IdM 웹 UI**에 관리자로 로그인합니다.
2. 인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.
3. 추가를 클릭합니다.

그림 9.5. IdM 웹 UI에 새 인증서 매핑 규칙 추가



4. 규칙 이름을 입력합니다.
5. 매핑 규칙을 입력합니다. 인증을 위해 **IdM**에 제공되는 전체 인증서를 **AD**에서 사용할 수 있는 것과 비교하려면 다음을 수행하십시오.

(userCertificate;binary={cert!bin})

6. 일치하는 규칙을 입력합니다. 예를 들어 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발급한 인증서만 인증하려면 다음을 수행합니다.

<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com

그림 9.6. AD에 저장된 인증서가 있는 사용자의 인증서 매핑 규칙

7.

추가를 클릭합니다.

8.

SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **CLI**에서 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

9.3.2. IdM CLI에 인증서 매핑 규칙 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

매핑 규칙을 입력하고 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. **AD**에서 사용할 수 있는 것과 비교하여 인증을 위해 제공되는 전체 인증서를 설정하려면 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발행한 인증서만 인증할 수 있습니다.

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "simpleADrule"
-----
```

```
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

3.

SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

9.4. AD가 사용자 인증서를 사용자 계정에 매핑하도록 구성된 경우 인증서 매핑 구성

이 사용자 사례에서는 **IdM** 배포가 **AD(Active Directory)**를 신뢰하고 **AD**의 사용자 항목이 인증서 매핑 데이터를 포함하는 경우 **IdM**에서 인증서 매핑을 활성화하는 데 필요한 단계를 설명합니다.

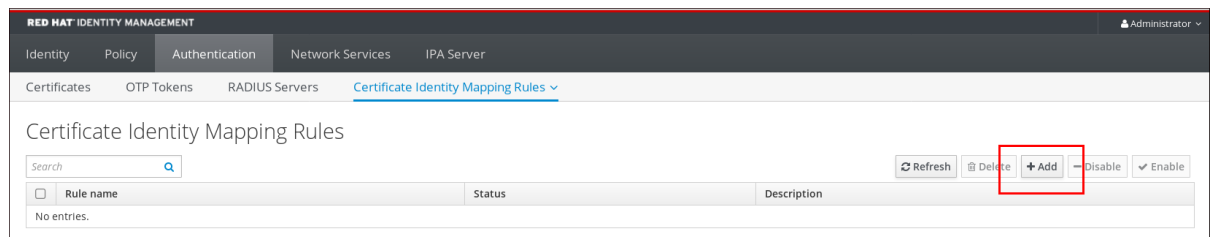
사전 요구 사항

- 사용자에게 **IdM**에 계정이 없습니다.
- 사용자에게는 **altSecurityIdentities** 속성이 포함된 **AD** 계정이 있으며, **AD**는 **IdM** **certmapdata** 속성과 동일합니다.
- **IdM** 관리자는 **IdM** 인증서 매핑 규칙을 기반으로 할 수 있는 데이터에 액세스할 수 있습니다.

9.4.1. IdM 웹 UI에 인증서 매핑 규칙 추가

1. **IdM 웹 UI**에 관리자로 로그인합니다.
2. 인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.
3. 추가를 클릭합니다.

그림 9.7. IdM 웹 UI에 새 인증서 매핑 규칙 추가



4. 규칙 이름을 입력합니다.

5.

매핑 규칙을 입력합니다. 예를 들어 **AD DC**에서 제공되는 인증서의 **Issuer** 및 **Subject** 항목을 검색하고 제공된 인증서의 두 항목에 있는 정보를 인증하기로 결정하십시오.

```
(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})
```

6.

일치하는 규칙을 입력합니다. 예를 들어 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발급한 인증서만 허용하여 사용자를 **IdM**에 인증하려면 다음을 수행합니다.

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7.

도메인에 로그인합니다.

```
ad.example.com
```

그림 9.8. AD가 매핑용으로 구성된 경우 인증서 매핑 규칙

8.

추가를 클릭합니다.

9.

SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **CLI**에서 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

9.4.2. IdM CLI에 인증서 매핑 규칙 추가

1.

관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2.

매핑 규칙을 입력하고 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. 예를 들어 제공되는 인증서의 **Issuer** 및 **Subject** 항목을 **AD** 검색을 수행하고 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발급한 인증서만 허용하려면 다음을 수행합니다.

```
# ipa certmaprule-add ad_configured_for_mapping_rule --matchrule
'<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule
'(<altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})' --
domain=ad.example.com

-----
Added Certificate Identity Mapping Rule "ad_configured_for_mapping_rule"
-----

Rule name: ad_configured_for_mapping_rule
Mapping rule: (<altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

3.

SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

9.4.3. AD 측에서 인증서 매핑 데이터 확인

altSecurityIdentities 속성은 **IdM**의 **certmapdata** 사용자 속성에 해당하는 **AD(Active Directory)**입니다. 신뢰할 수 있는 **AD** 도메인을 사용자 계정에 매핑하도록 구성된 시나리오에서 **IdM**에서 인증서 매핑을 구성할 때 **IdM** 시스템 관리자는 **AD**의 사용자 항목에 **altSecurityIdentities** 속성이 올바르게 설정되었는지 확인해야 합니다.

AD에 저장된 사용자에 대한 올바른 정보가 포함되어 있는지 확인하려면 **ldapsearch** 명령을 사용하십시오.

•

예를 들어 아래 명령을 입력하여 다음 조건이 적용되는 **adserver.ad.example.com** 서버로 확인합니다.

○

altSecurityIdentities 속성은 **ad_user**의 사용자 항목에 설정됩니다.

○

matchrule에서 다음과 같은 조건이 적용됩니다.

■ **ad_user**가 AD에 인증하는 데 사용하는 인증서는 **ad.example.com** 도메인의 **AD-ROOT-CA**에서 발행되었습니다.

■ 제목은 < S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user:

```
$ ldapsearch -o ldif-wrap=no -LLL -h adserver.ad.example.com \
-p 389 -D cn=Administrator,cn=users,dc=ad,dc=example,dc=com \
-W -b cn=users,dc=ad,dc=example,dc=com "(cn=ad_user)" \
altSecurityIdentities
Enter LDAP Password:
dn: CN=ad_user,CN=Users,DC=ad,DC=example,DC=com
altSecurityIdentities: X509:<l>DC=com,DC=example,DC=ad,CN=AD-ROOT-
CA<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user
```

9.5. AD 사용자 항목에 인증서가 없거나 매핑 데이터가 없는 경우 인증서 매핑 구성

이 사용자 사례에서는 **IdM** 배포가 **AD(Active Directory)**와 함께 **IdM** 배포가 신뢰할 수 있는 경우 **IdM**에서 인증서 매핑을 활성화하는 데 필요한 단계를 설명하고, **AD**의 사용자 항목에 전체 인증서와 인증서 매핑 데이터가 포함되어 있지 않습니다.

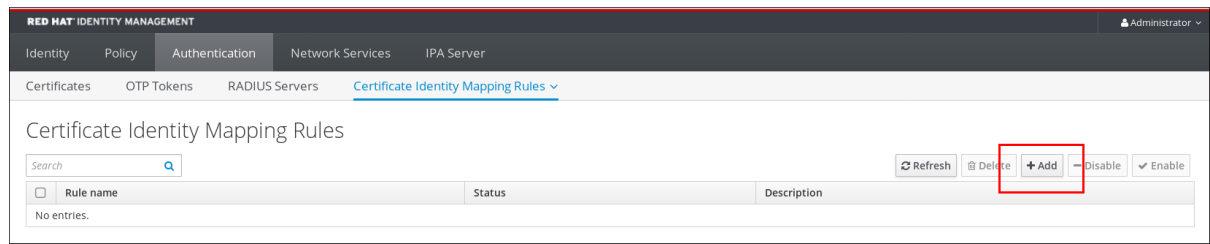
사전 요구 사항

- 사용자에게 **IdM**에 계정이 없습니다.
- 사용자는 **AD**에 전체 인증서와 **altSecurityIdentities** 속성을 모두 포함하는 계정을 가지고 있으며, **AD**는 **IdM certmapdata** 속성과 동일합니다.
- **IdM** 관리자는 **IdM**의 **AD** 사용자 ID 덮어쓰기에 추가할 전체 **AD** 사용자 인증서가 있습니다.

9.5.1. IdM 웹 UI에 인증서 매핑 규칙 추가

1. **IdM 웹 UI**에 관리자로 로그인합니다.
2. 인증 → 인증서 ID 매핑 규칙 → 인증서 ID 매핑 규칙으로 이동합니다.
3. 추가를 클릭합니다.

그림 9.9. IdM 웹 UI에 새 인증서 매핑 규칙 추가



4.

규칙 이름을 입력합니다.

5.

매핑 규칙을 입력합니다. IdM에서 AD 사용자 항목의 사용자 ID 오버라이드 항목에 저장된 인증서와 비교하여 IdM에 제공되는 전체 인증서를 보유하려면 다음을 수행합니다.

```
(userCertificate;binary={cert!bin})
```

6.

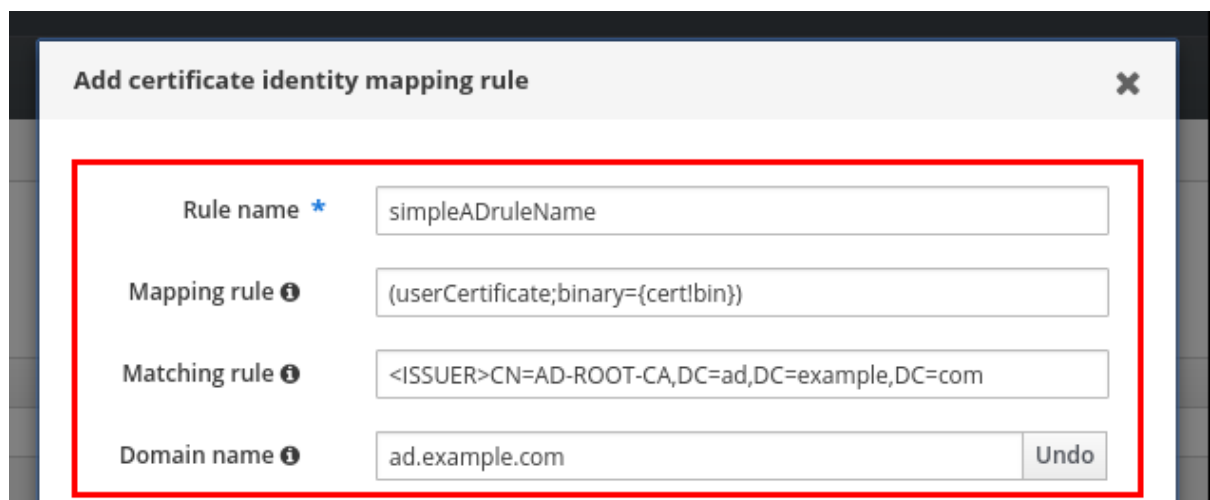
일치하는 규칙을 입력합니다. 예를 들어 AD.EXAMPLE.COM 도메인의 AD-ROOT-CA 에서 발급한 인증서만 인증하려면 다음을 수행합니다.

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7.

도메인 이름을 입력합니다. 예를 들어 ad.example.com 도메인에서 사용자를 검색하려면 다음을 수행합니다.

그림 9.10. 인증서 또는 AD에 저장된 데이터를 매핑하지 않은 사용자의 인증서 매핑 규칙



8.

추가를 클릭합니다.

9.

SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 CLI에서 SSSD를 다시 시작합니다.

```
# systemctl restart sssd
```

9.5.2. IdM CLI에 인증서 매핑 규칙 추가

1. 관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2. 매핑 규칙을 입력하고 매핑 규칙이 기반으로 하는 일치 규칙을 입력합니다. IdM의 사용자 ID 덮어쓰기 항목에 저장된 인증서에 비해 인증을 위해 제공되는 전체 인증서를 보유하려면 **AD.EXAMPLE.COM** 도메인의 **AD-ROOT-CA** 에서 발행한 인증서만 인증할 수 있습니다.

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
-----
Added Certificate Identity Mapping Rule "simpleADrule"
-----
```

```
Rule name: simpleADrule
Mapping rule: (userCertificate;binary={cert!bin})
Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
Domain name: ad.example.com
Enabled: TRUE
```

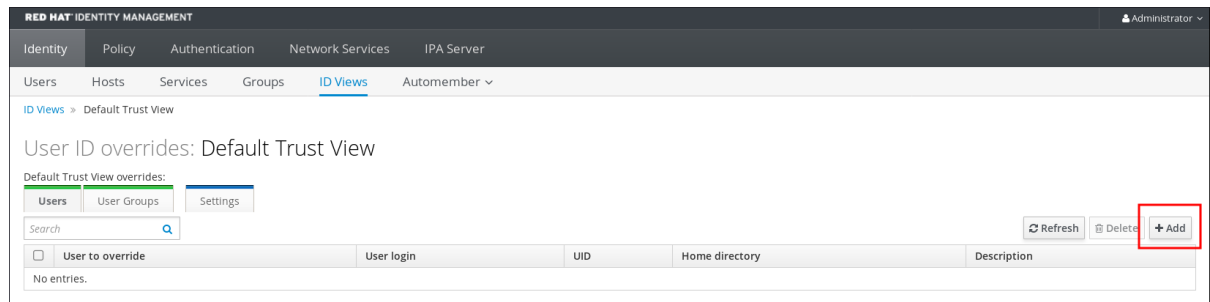
3. SSSD(System Security Services Daemon)는 인증서 매핑 규칙을 주기적으로 다시 읽습니다. 새로 생성된 규칙을 즉시 로드하도록 하려면 SSSD를 다시 시작합니다.

```
# systemctl restart sssd
```

9.5.3. IdM 웹 UI에서 AD 사용자 ID 재정의에 인증서 추가

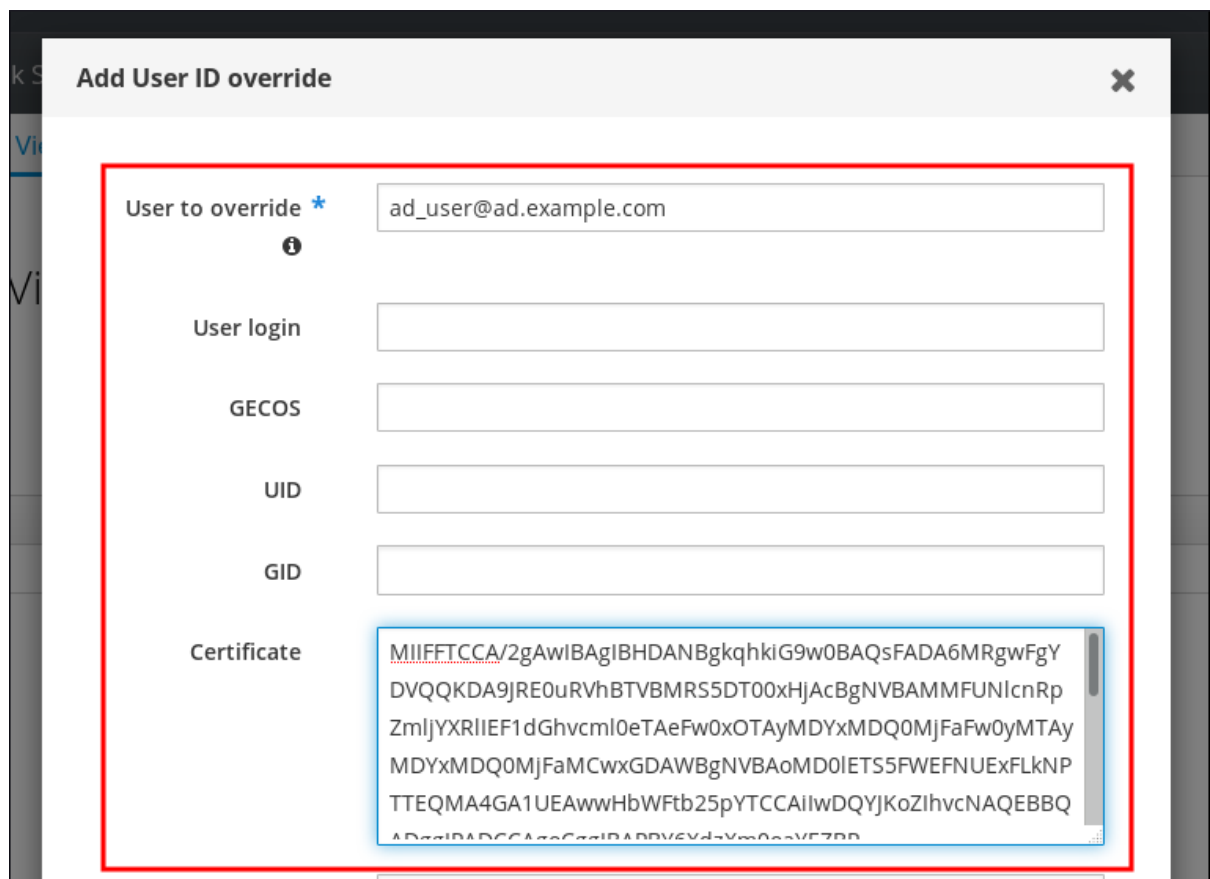
1. Identity → ID Views → Default Trust View 로 이동합니다.
2. 추가를 클릭합니다.

그림 9.11. IdM 웹 UI에서 새 사용자 ID 덮어쓰기 추가



3. **User to override** 필드에 **ad_user@ad.example.com** 을 입력합니다.
4. **ad_user** 의 인증서를 복사하여 인증서 필드에 붙여넣습니다.

그림 9.12. AD 사용자에게 대한 사용자 ID 재정의 구성



5. **추가**를 클릭합니다.

검증 단계

- 사용자 및 인증서가 연결되었는지 확인합니다.

○

sss_cache 유틸리티를 사용하여 **SSSD** 캐시에서 **ad_user@ad.example.com** 레코드를 무효화하고 **ad_user@ad.example.com** 정보를 다시 로드합니다.

```
# sss_cache -u ad_user@ad.example.com
```

o

AD 사용자의 인증서가 포함된 파일 이름으로 **ipa certmap-match** 명령을 실행합니다.

```
# ipa certmap-match ad_user_cert.pem
-----
1 user matched
-----
Domain: AD.EXAMPLE.COM
User logins: ad_user@ad.example.com
-----
Number of entries returned 1
-----
```

출력은 인증서 매핑 데이터가 **ad_user@ad.example.com**에 추가되고 **AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가에 정의된 해당 매핑 규칙이 있는지** 확인합니다. 즉, 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 **ad_user@ad.example.com**로 인증할 수 있습니다.

9.5.4. IdM CLI에서 AD 사용자 ID 재정의에 인증서 추가

추가 리소스

1. 관리자의 자격 증명을 가져옵니다.

```
# kinit admin
```

2. 인증서 **Blob**을 **CERT** 라는 새 변수에 저장합니다.

```
# CERT=`cat ad_user_cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n'`
```

3. **ipa idoverrideuser-add-cert** 명령을 사용하여 사용자 계정에 **ad_user@ad.example.com**의 인증서를 추가합니다.

```
# ipa idoverrideuser-add-cert ad_user@ad.example.com --certificate $CERT
```

검증 단계

- 사용자 및 인증서가 연결되었는지 확인합니다.
- `sss_cache` 유틸리티를 사용하여 SSSD 캐시에서 `ad_user@ad.example.com` 레코드를 무효화하고 `ad_user@ad.example.com` 정보를 다시 로드합니다.

```
# sss_cache -u ad_user@ad.example.com
```

- AD 사용자의 인증서가 포함된 파일 이름으로 `ipa certmap-match` 명령을 실행합니다.

```
# ipa certmap-match ad_user_cert.pem
-----
1 user matched
-----
Domain: AD.EXAMPLE.COM
User logins: ad_user@ad.example.com
-----
Number of entries returned 1
-----
```

출력은 인증서 매핑 데이터가 `ad_user@ad.example.com`에 추가되고 **AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가에 정의된 해당 매핑 규칙이 있는지** 확인합니다. 즉, 정의된 인증서 매핑 데이터와 일치하는 인증서를 사용하여 `ad_user@ad.example.com`로 인증할 수 있습니다.

9.6. 여러 ID 매핑 규칙을 하나로 결합

추가 리소스

여러 ID 매핑 규칙을 하나의 결합된 규칙으로 결합하려면 `|` (또는) 문자를 사용하여 개별 매핑 규칙 앞에 `|`를 사용하고 `()` 대괄호를 사용하여 분리합니다.

인증서 매핑 필터 예 1

```
$ ipa certmaprule-add ad_cert_for_ipa_and_ad_users \
--maprule='(|(ipacertmapdata=X509:<|>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<|>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
--matchrule='<|ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' \
--domain=ad.example.com
```

위 예에서 **--maprule** 옵션의 필터 정의에는 다음과 같은 기준이 포함됩니다.

- **ipacertmapdata=X509:<I>{issuer_dn!nss_x500}** 는 IdM의 인증서 **매핑 규칙**에 설명된 대로 스마트 카드 인증서의 주체 및 발행자를 IdM 사용자 계정의 값으로 연결하는 필터입니다.
- **altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}={subject_dn!ad_x500}**는 스마트 카드 인증서의 주체 및 발급자를 AD 사용자 계정의 값에 연결하는 필터입니다. **altSecurityIdentities=X509:<I>{issuer_dn!ad_x500} is a filter that links the subject and issuer from altSecurityIdentities attribute in the AD user account.**
- **--domain=ad.example.com** 옵션을 추가하면 지정된 인증서에 매핑된 사용자가 로컬 **idm.example.com** 도메인에서만 아니라 **ad.example.com** 도메인에서도 검색됩니다.

--maprule 옵션의 필터 정의는 논리 연산자 | (또는)를 허용하므로 여러 기준을 지정할 수 있습니다. 이 경우 규칙은 조건 중 하나 이상을 충족하는 모든 사용자 계정을 매핑합니다.

인증서 매핑 필터 예 2

```
$ ipa certmaprule-add ipa_cert_for_ad_users \
--maprule='((userCertificate;binary={cert!bin}))(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}))(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
--matchrule='<ISSUER>CN=Certificate Authority,O=REALM.EXAMPLE.COM' \
--domain=idm.example.com --domain=ad.example.com
```

위 예에서 **--maprule** 옵션의 필터 정의에는 다음과 같은 기준이 포함됩니다.

- **userCertificate;binary={cert!bin}** 은 전체 인증서가 포함된 사용자 항목을 반환하는 필터입니다. AD 사용자의 경우 이 유형의 필터를 생성하는 것은 **AD 사용자 항목에 인증서 또는 매핑 데이터가 없는 경우 인증서 매핑 규칙 추가**에 자세히 설명되어 있습니다.
- **ipacertmapdata=X509:<I>{issuer_dn!nss_x500}** 는 IdM의 인증서 **매핑 규칙**에 설명된 대로 스마트 카드 인증서의 주체 및 발행자를 IdM 사용자 계정의 값에 연결하는 필터입니다.

-

Alt SecurityIdentities=X509:<|>{issuer_dn!ad_x500}={subject_dn!ad_x500}는 스마트 카드 인증서의 주체 및 발행자를 AD 사용자 계정의 값에 연결하는 필터입니다. **AltSecurityIdentities=X509:<|>{ issuer_dn!ad_x500} is altSecurityIdentities attribute in altSecurityIdentities attribute in the AD user map.**

--maprule 옵션의 필터 정의는 논리 연산자 | (또는)를 허용하므로 여러 기준을 지정할 수 있습니다. 이 경우 규칙은 조건 중 하나 이상을 충족하는 모든 사용자 계정을 매핑합니다.

10장. IDM 클라이언트의 데스크탑에 저장된 인증서로 인증 구성

IdM(Identity Management)을 구성하면 **IdM** 시스템 관리자가 인증 기관(**CA**)이 사용자에게 발행한 인증서를 사용하여 **IdM** 웹 **UI** 및 **CLI**(명령줄 인터페이스)에 인증할 수 있습니다.

웹 브라우저는 **IdM** 도메인에 포함되지 않은 시스템에서 실행할 수 있습니다.

이 사용자 스토리는 **IdM** 클라이언트의 데스크탑에 저장된 인증서를 사용하여 **ID** 관리 웹 **UI** 및 **CLI**에 대한 로그인을 효과적으로 구성하고 테스트하는 방법에 대한 지침을 제공합니다. 이 사용자 스토리를 보면,

- 인증서에 이미 **인증서가 있는 경우 새 사용자 인증서 요청 및 클라이언트 내보내기** 를 건너뛸 수 있습니다.
- **IdM CA** 에서 **사용자의 인증서를 발급한 경우 인증서와 사용자가 함께 연결되어 있는지 여부** 를 건너뛸 수 있습니다.



참고

ID 관리 사용자만 인증서를 사용하여 웹 **UI**에 로그인할 수 있습니다. **Active Directory** 사용자는 사용자 이름과 암호를 사용하여 로그인할 수 있습니다.

10.1. 웹 UI에서 인증 인증을 위한 IDENTITY MANAGEMENT SERVER 구성

IdM(Identity Management) 관리자는 사용자가 인증서를 사용하여 **IdM** 환경에 인증할 수 있습니다.

절차

Identity Management 관리자:

1. **Identity Management** 서버에서 관리자 권한을 얻고 서버를 구성하는 셸 스크립트를 생성합니다.
 - a. `ipa-adviser config-server-for-smart-card-auth` 명령을 실행하고 출력을 파일에 저장합니다(예: `server_certificate_script.sh`):

```
# kinit admin
# ipa-adviser config-server-for-smart-card-auth > server_certificate_script.sh
```

b.

pkexec 유틸리티를 사용하여 파일에 실행 권한을 추가합니다.

```
# chmod +x server_certificate_script.sh
```

2.

Identity Management 도메인의 모든 서버에서 **server_certificate_script.sh** 스크립트를 실행합니다.

a.

IdM CA가 인증서 인증을 활성화하려는 사용자의 인증서를 발급한 유일한 인증 기관인 경우 입력으로서 **IdM** 인증 기관의 경로 **/etc/ipa/ca.crt** 입니다.

```
# ./server_certificate_script.sh /etc/ipa/ca.crt
```

b.

다른 외부 **CA**가 인증서 인증을 활성화하려는 사용자의 인증서에 서명하면 관련 **CA** 인증서가 입력으로 발생합니다.

```
# ./server_certificate_script.sh /tmp/ca1.pem /tmp/ca2.pem
```



참고

전체 토폴로지에서 활성화된 사용자에게 인증서 인증을 사용하려면 나중에 시스템에 추가하는 각 새 복제본에서 스크립트를 실행해야 합니다.

10.2. 새 사용자 인증서 요청 및 클라이언트로 내보내기

IdM(Identity Management) 관리자는 **IdM** 환경에서 사용자의 인증서를 생성하고 사용자에게 인증서 인증을 활성화하려는 **IdM** 클라이언트에 내보낼 수 있습니다.



참고

인증서를 사용하여 인증하려는 사용자에게 인증서가 이미 있는 경우 이 섹션을 건너뛸 수 있습니다.

절차

1.

필요한 경우 새 디렉터리(예: **~/certdb/**)를 생성하고 임시 인증서 데이터베이스를 만듭니다.

메시지가 표시되면 **NSS** 인증서 **DB** 암호를 생성하여 후속 단계에서 생성할 인증서에 대한 키를 암호화합니다.

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2.

CSR(인증서 서명 요청)을 생성하고 출력을 파일로 리디렉션합니다. 예를 들어 **IDM.EXAMPLE.COM** 영역에 있는 **idm_user** 사용자의 **4096** 비트 인증서에 대해 **certificate_request.csr** 이라는 이름으로 **CSR**을 생성하려면 인증서 개인 키의 **nickname**을 쉽게 찾을 수 있도록 **idm_user** 로 설정하고 **CN=idm_user,O=IDMM.EXALE:COM**으로 설정합니다.

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3.

메시지가 표시되면 **certutil** 을 사용하여 임시 데이터베이스를 생성할 때 입력한 암호와 동일한 암호를 입력합니다. 다음으로 중지될 때까지 **randlomly**를 계속 입력합니다.

Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

4.

인증서 요청 파일을 서버에 제출합니다. 새로 발급된 인증서, 인증서를 저장할 출력 파일, 선택적으로 인증서 프로필과 연결할 **Kerberos** 주체를 지정합니다. 예를 들어 **IECUserRoles** 프로필의 인증서를 얻으려면 사용자 역할 확장이 추가된 프로필인 **idm_user@IDM.EXAMPLE.COM** 주체의 경우 **~/idm_user.pem** 파일에 저장합니다.

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --certificate-out=~/idm_user.pem
```

5.

NSS 데이터베이스에 인증서를 추가합니다. **-n** 옵션을 사용하여 이전에 **CSR**을 생성할 때 사용한 것과 동일한 **nickname**을 설정하여 인증서가 **NSS** 데이터베이스의 개인 키와 일치하도록

합니다. **t** 옵션은 신뢰 수준을 설정합니다. 자세한 내용은 **certutil(1)** 매뉴얼 페이지를 참조하십시오. **i** 옵션은 입력 인증서 파일을 지정합니다. 예를 들어 **NSS** 데이터베이스에 **~/certdb/** 데이터베이스의 **~/idm_user.pem** 파일에 저장된 **idm_user nickname**을 사용하여 인증서를 추가하려면 다음을 수행합니다.

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6.

NSS 데이터베이스의 키가 닉네임 (**orphan**) 으로 표시되지 않는지 확인합니다. 예를 들어 **~/certdb/** 데이터베이스에 저장된 인증서가 분리되지 않았는지 확인하려면 다음을 수행합니다.

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395  NSS Certificate
DB:idm_user
```

7.

pk12util 명령을 사용하여 **NSS** 데이터베이스에서 **PKCS12** 형식으로 인증서를 내보냅니다. 예를 들어 **idm_user nickname**을 사용하여 인증서를 **/root/certdb NSS** 데이터베이스에서 **~/idm_user.p12** 파일로 내보내려면 다음을 실행합니다.

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8.

idm_user 의 인증서 인증을 활성화할 호스트에 인증서를 전송합니다.

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9.

인증서가 전송된 호스트에서 **.pkcs12** 파일이 저장된 디렉토리를 보안상의 이유로 **'other'** 그룹에 저장할 수 없도록 합니다.

```
# chmod o-rwx /home/idm_user/
```

10.

보안상의 이유로 서버에서 임시 **NSS** 데이터베이스 및 **.pkcs12** 파일을 제거하십시오.

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

10.3. 인증서와 사용자가 연결되어 있는지 확인



참고

IdM CA에서 사용자 인증서를 발급한 경우 이 섹션을 건너뛸 수 있습니다.

인증서 인증이 작동하려면 인증서가 **IdM(Identity Management)**에 인증할 사용자에게 인증서가 연결되어 있는지 확인해야 합니다.

- ID 관리 환경에 포함되지 않은 인증 기관에서 인증서를 제공하는 경우 사용자 계정 링크를 통해 인증서와 관련된 절차에 따라 사용자와 [인증서를 연결합니다](#).
- ID 관리 **CA**에서 인증서를 제공하는 경우 인증서가 이미 사용자 항목에 자동으로 추가되고 인증서를 사용자 계정에 연결할 필요가 없습니다. **IdM**에서 새 인증서를 생성하는 방법에 대한 자세한 내용은 [새 사용자 인증서 요청 및 클라이언트로 내보내기](#)를 참조하십시오.

10.4. 인증서 인증을 사용하도록 브라우저 구성

WebUI를 사용하여 **IdM(Identity Management)**에 로그인할 때 인증서로 인증할 수 있으려면 사용자와 관련 인증 기관(**CA**) 인증서를 **Mozilla Firefox** 또는 **Google Chrome** 브라우저로 가져와야 합니다. 브라우저가 실행 중인 호스트 자체는 **IdM** 도메인의 일부가 될 필요가 없습니다.

IdM은 다음 브라우저를 지원하여 **WebUI** 연결을 지원합니다.

- **Mozilla Firefox 38 이상**
- **Google Chrome 46 이상**

다음 절차에서는 **Mozilla Firefox 57.0.1** 브라우저를 구성하는 방법을 보여줍니다.

사전 요구 사항

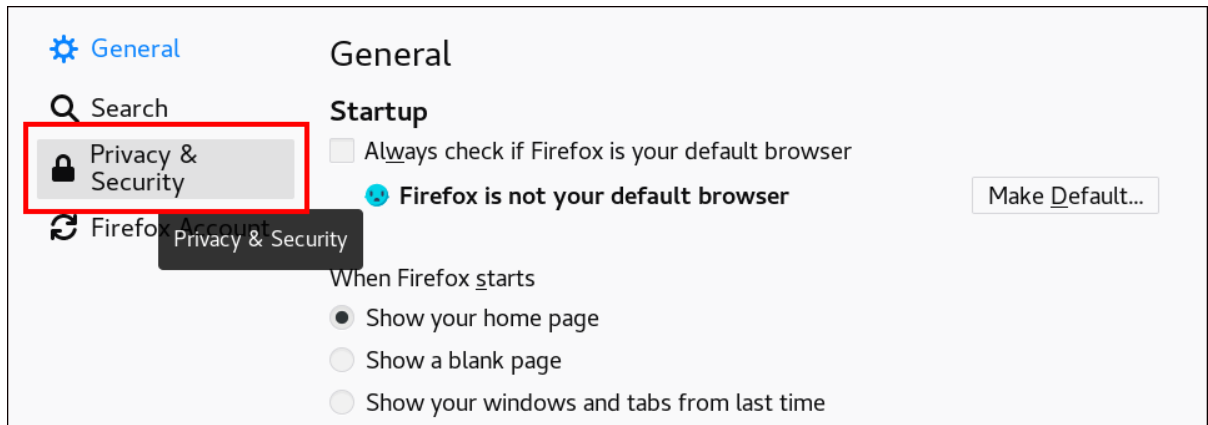
- **PKCS#12** 형식으로든 브라우저로 가져올 [사용자 인증서](#)가 있습니다.

절차

1.

Firefox를 연 다음 기본 설정 → 개인 정보 보호 및 보안 으로 이동합니다.

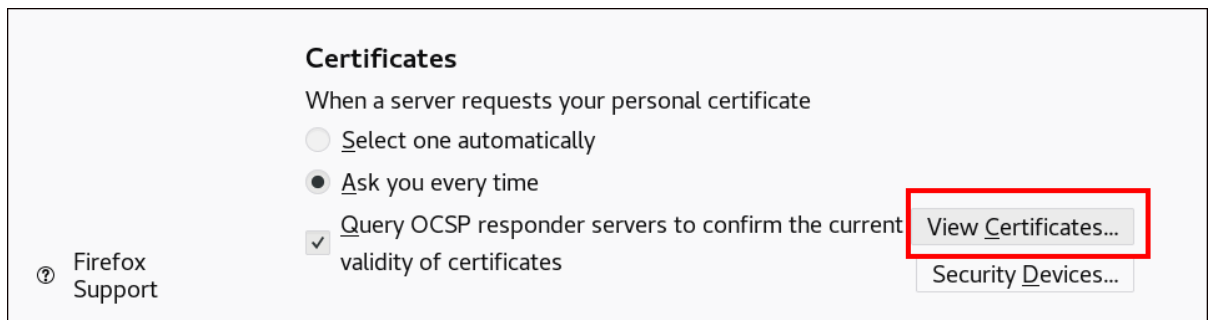
그림 10.1. 기본 설정의 개인 정보 및 보안 섹션



2.

View Certificates (인증서 보기)를 클릭합니다.

그림 10.2. 개인 정보 보호 및 보안의 인증서 보기



3.

인증서 탭에서 가져오기 를 클릭합니다. **PKCS12** 형식으로 사용자 인증서를 찾아서 연 다음 확인 및 확인을 클릭합니다.

4.

Firefox에서 **Identity Management** 인증 기관을 신뢰할 수 있는 기관으로 인식했는지 확인합니다.

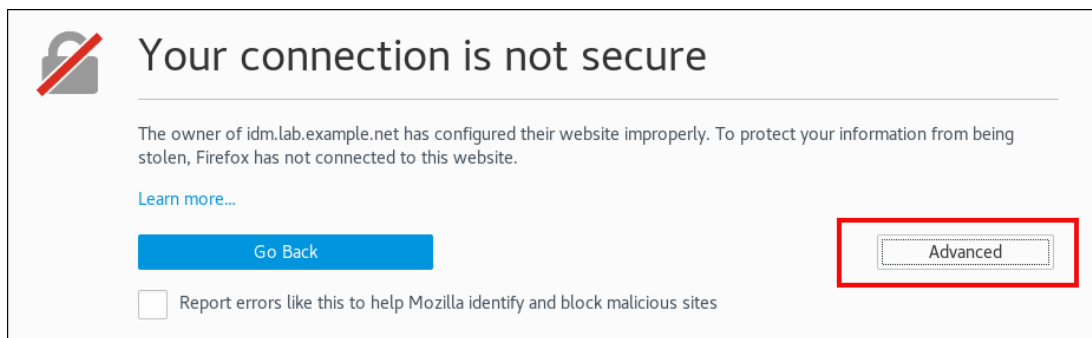
a.

IdM CA 인증서를 로컬에 저장합니다.

•

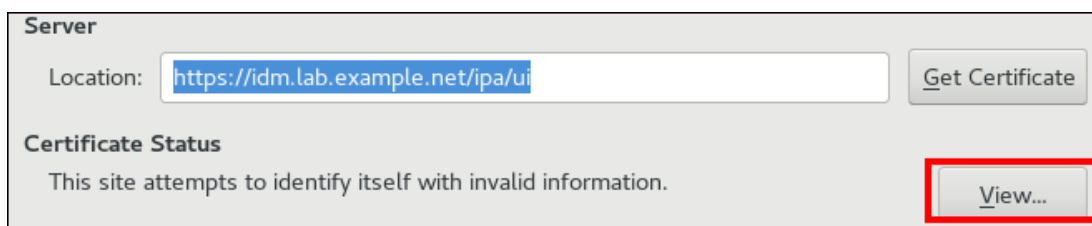
Firefox 주소 표시줄에 **IdM** 서버 이름을 작성하여 **IdM** 웹 UI로 이동합니다. **Insecure Connection** 경고 페이지에서 **Advanced** 를 클릭합니다.

그림 10.3. 비보안 연결



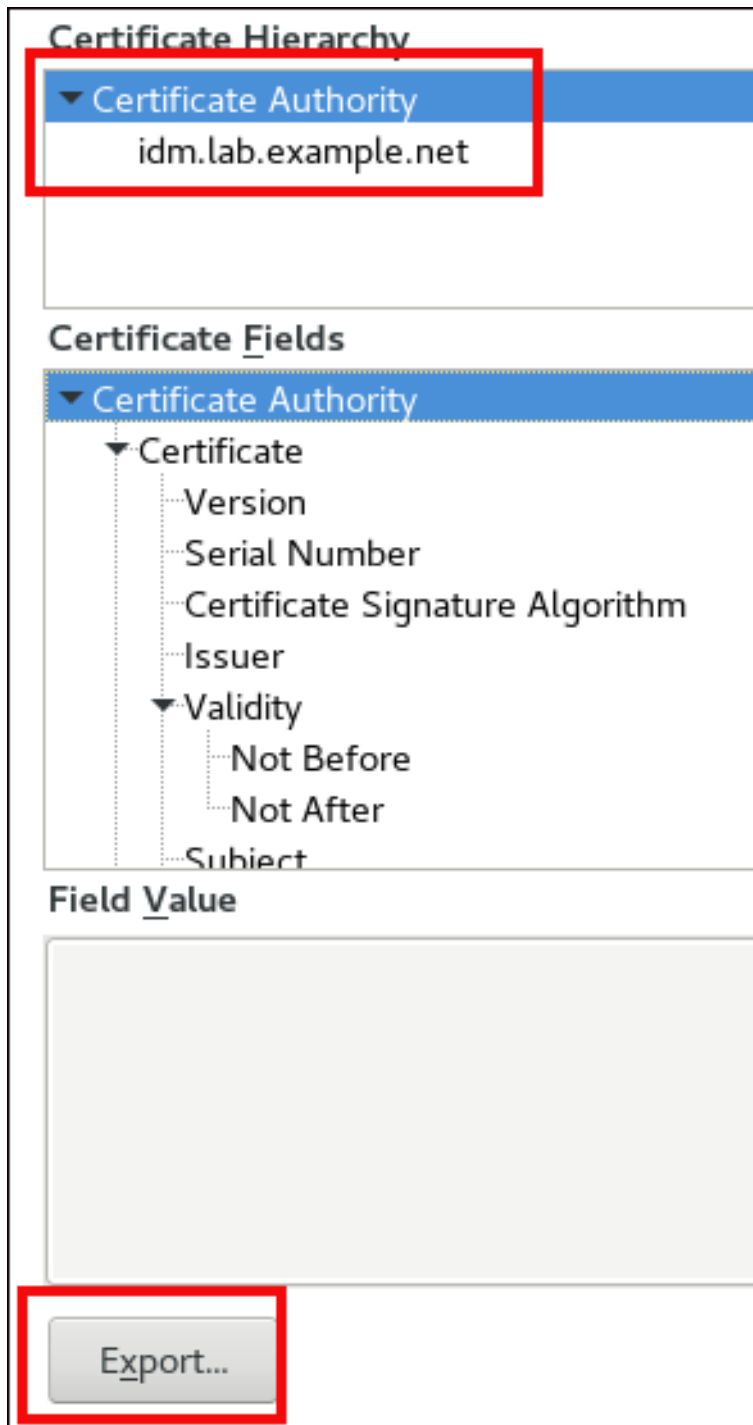
- 예외를 추가합니다. **View** 를 클릭합니다.

그림 10.4. 인증서 세부 정보 보기



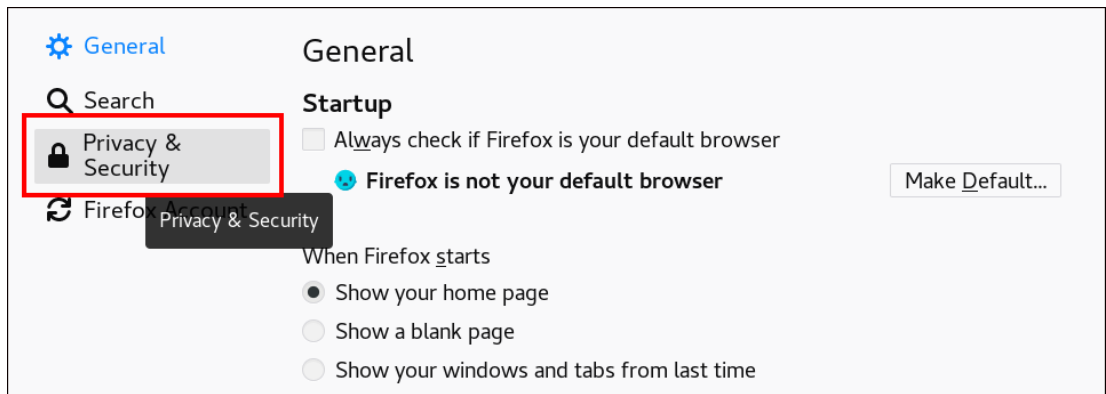
- **Details** (세부 정보) 탭에서 **Certificate Authority** 필드를 강조 표시합니다.

그림 10.5. CA 인증서 내보내기



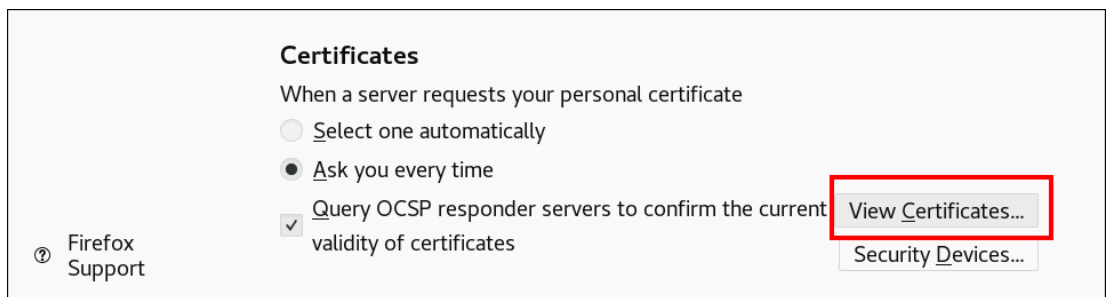
- **Export.** 예를 들어 **CertificateAuthority.crt** 파일과 같은 **CA** 인증서를 저장한 다음 닫기 를 클릭하고 취소를 클릭합니다.
- b. **IdM CA** 인증서를 신뢰할 수 있는 인증 기관 인증서로 **Firefox**로 가져옵니다.
- **Firefox**를 열고 기본 설정으로 이동하고 개인 정보 및 보안을 클릭합니다.

그림 10.6. 기본 설정의 개인 정보 및 보안 섹션



• **View Certificates (인증서 보기)**를 클릭합니다.

그림 10.7. 개인 정보 보호 및 보안의 인증서 보기



• **Authorities (권한) 탭에서 Import (가져오기)**를 클릭합니다.
CertificateAuthority.crt 파일에서 이전 단계에서 저장한 **CA** 인증서를 찾아서 엽니다.
 인증서를 신뢰하여 웹 사이트를 식별한 다음 확인을 클릭하고 확인을 클릭합니다.

5.

Identity Management User로 인증서를 사용하여 **ID** 관리 웹 **UI**를 계속 인증합니다.

10.5. IDENTITY MANAGEMENT USER로 인증서를 사용하여 IDENTITY MANAGEMENT 웹 UI에 인증

이 절차에서는 **ID** 관리 클라이언트의 데스크탑에 저장된 인증서를 사용하여 **IdM(Identity Management)** 웹 **UI**에 대한 사용자로 인증을 설명합니다.

절차

1. 브라우저에서 **에서 ID 관리 웹 UI**로 이동합니다(예: <https://server.idm.example.com/ipa/ui>).
2. 인증서를 사용하여 로그인을 클릭합니다.

. ID 관리 웹 UI에서 인증서를 사용하여 로그인

3.

사용자 인증서가 이미 선택되어 있어야 합니다. **Remember this decision.**를 선택 해제한 다음 확인을 클릭합니다.

이제 인증서에 해당하는 사용자로 인증되었습니다.

추가 리소스

-

스마트 카드 인증을 위한 ID 관리 구성을 참조하십시오.

10.6. 인증서를 사용하여 CLI에 인증할 수 있도록 IDM 클라이언트 구성

IdM 클라이언트의 CLI(명령줄 인터페이스)에서 IdM 사용자에게 대한 인증서 인증을 수행하려면 IdM 사용자의 인증서 인증서 및 개인 키를 IdM 클라이언트에 가져옵니다. 사용자 인증서 생성 및 전송에 대한 자세한 내용은 [새 사용자 인증서 요청 및 클라이언트로 내보내기](#)를 참조하십시오.

절차

-

IdM 클라이언트에 로그인하고 사용자 인증서와 개인 키가 포함된 .p12 파일이 있어야 합니다. Kerberos 티켓 부여 티켓(TGT)을 가져오고 캐시하려면 `X509_username:/path/to/file.p12` 속성과 함께 `-X` 옵션을 사용하여 사용자의 X509 ID 정보를 찾을 위치를 지정합니다. 예를 들어 `~/idm_user.p12` 파일에 저장된 사용자의 ID 정보를 사용하여 `idm_user`에 대한 TGT를 얻으려면 다음을 실행합니다.

```
$ kinit -X X509_idm_user='PKCS12:~/idm_user.p12' idm_user
```



참고

또한 이 명령은 **.pem** 파일 형식인 **kinit -X X509_username='FILE:/path/to/cert.pem,/path/to/key' user_principal**을 지원합니다.

11장. IDM CA 갱신 서버 사용

11.1. IDM CA 갱신 서버에 대한 설명

CA(인증 기관)를 사용하는 IdM(Identity Management) 배포에서 CA 갱신 서버는 IdM 시스템 인증서를 유지 관리하고 갱신합니다. 이를 통해 강력한 IdM 배포를 수행할 수 있습니다.

IdM 시스템 인증서는 다음과 같습니다.

- IdM CA 인증서
- OCSP 서명 인증서
- IdM CA 하위 시스템 인증서
- IdM CA 감사 서명 인증서
- IdM 갱신 에이전트 (RA) 인증서
- KRA 전송 및 스토리지 인증서

시스템 인증서를 특징으로 설정하는 것은 해당 키가 모든 CA 복제본에서 공유된다는 것입니다. 대조적으로 IdM 서비스 인증서(예: LDAP, HTTP 및 PKINIT 인증서)에는 서로 다른 IdM CA 서버에서 키 쌍 및 제목 이름이 다릅니다.

IdM 토폴로지에서는 기본적으로 첫 번째 IdM CA 서버는 CA 갱신 서버입니다.



참고

업스트림 설명서에서 IdM CA는 Dogtag 라고 합니다.

CA 갱신 서버의 역할

IdM 배포에 IdM CA, IdM CA 하위 시스템 및 IdM RA 인증서가 중요합니다. 각 인증서는 `/etc/pki/pki-tomcat/` 디렉터리에 있는 NSS 데이터베이스에 저장되고 LDAP 데이터베이스 항목으로 저장됩니다. LDAP에 저장된 인증서는 NSS 데이터베이스에 저장된 인증서와 일치해야 합니다. 일치하지 않는 경우 IdM 프레임워크와 IdM CA 간에 인증 오류가 발생하고 IdM CA와 LDAP 간에 인증 오류가 발생합니다.

모든 IdM CA 복제본에는 모든 시스템 인증서에 대한 추적 요청이 있습니다. 통합 CA가 있는 IdM 배포에 CA 갱신 서버가 없는 경우 각 IdM CA 서버는 시스템 인증서 갱신을 독립적으로 요청합니다. 이로 인해 다양한 시스템 인증서 및 인증 오류가 발생하는 다른 CA 복제본이 발생합니다.

하나의 CA 복제본을 갱신 서버로 지정하면 필요에 따라 시스템 인증서를 정확히 한 번 갱신할 수 있으므로 인증 오류가 발생하지 않습니다.

CA 복제본에서 certmonger 서비스의 역할

모든 IdM CA 복제본에서 실행되는 certmonger 서비스는 dogtag-ipa-ca-renew-agent 갱신 도우미를 사용하여 IdM 시스템 인증서를 추적합니다. 갱신 도우미 프로그램은 CA 갱신 서버 구성을 읽습니다. CA 갱신 서버가 아닌 각 CA 복제본에서 갱신 도우미는 ca_renewal LDAP 항목에서 최신 시스템 인증서를 검색합니다. 정확히 certmonger 갱신 시도가 발생할 때 발생하는 결정적이지 않기 때문에 dogtag-ipa-ca-renew-agent 도우미는 CA 갱신 서버가 인증서를 실제로 갱신하기 전에 시스템 인증서를 업데이트하려고 하는 경우가 있습니다. 이 경우 이전의 곧 만료되는 인증서가 CA 복제본의 certmonger 서비스로 반환됩니다. certmonger 서비스를 통해 이미 데이터베이스에 저장된 인증서와 동일한 인증서이며 CA 갱신 서버에서 업데이트된 인증서를 검색할 수 있을 때까지 개별 시도 사이에 일정 지연으로 인증서를 갱신하려고 합니다.

IdM CA 갱신 서버의 올바른 기능

포함된 CA가 있는 IdM 배포는 IdM CA와 함께 설치된 IdM 배포 또는 IdM CA 서버가 나중에 설치된 IdM 배포입니다. 포함된 CA가 포함된 IdM 배포에는 항상 하나의 CA 복제본이 갱신 서버로 구성되어 있어야 합니다. 갱신 서버는 온라인 상태여야 하며, 다른 서버와 함께 올바르게 복제해야 합니다.

`ipa server-del, ipa-replica-manage del, ipa-csreplica-manage del` 또는 `ipa-server-install --uninstall` 명령을 사용하여 현재 CA 갱신 서버가 삭제되면 다른 CA 복제본이 CA 갱신 서버로 자동으로 할당됩니다. 이 정책은 갱신 서버 구성이 유효한 상태로 유지됩니다.

이 정책은 다음 상황을 다루지 않습니다.

- 오프라인 갱신 서버

갱신 서버가 연장된 기간 동안 오프라인 상태인 경우 갱신 기간이 누락될 수 있습니다. 이 경우 인증서가 만료될 때까지 모든 갱신 CA 서버가 현재 시스템 인증서를 다시 설치합니다. 이 경우

만료된 인증서도 다른 인증서에 대해 갱신 오류가 발생할 수 있으므로 **IdM** 배포가 중단됩니다.

- 복제 문제

갱신 서버 및 기타 **CA** 복제본 간에 복제 문제가 있을 수 있지만 다른 **CA** 복제본은 만료되기 전에 업데이트된 인증서를 검색하지 못할 수 있습니다.

이러한 상황을 방지하려면 복제 계약이 올바르게 작동하는지 확인하십시오. 자세한 내용은 **RHEL 7 Linux 도메인 ID, 인증 및 정책 가이드의 일반 또는 특정 복제 문제 해결 지침을 참조하십시오.**

11.2. IDM CA 갱신 서버 변경 및 재설정

CA(인증 기관) 갱신 서버가 해제되는 경우 **IdM**(Identity Management)은 **IdM CA** 서버 목록에서 새 **CA** 갱신 서버를 자동으로 선택합니다. 시스템 관리자가 선택에 영향을 미칠 수 없습니다.

새 **IdM CA** 갱신 서버를 선택하려면 시스템 관리자가 수동으로 교체를 수행해야 합니다. 현재 갱신 서버를 해제하는 프로세스를 시작하기 전에 새 **CA** 갱신 서버를 선택합니다.

현재 **CA** 갱신 서버 구성이 유효하지 않으면 **IdM CA** 갱신 서버를 재설정합니다.

CA 갱신 서버를 변경하거나 재설정하려면 이 절차를 완료합니다.

사전 요구 사항

- **IdM** 관리자 인증 정보가 있습니다.

절차

1. **IdM** 관리자 자격 증명을 가져옵니다.

```
~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. 필요한 경우 배포의 **IdM** 서버에 새 **CA** 갱신 서버가 되는 데 필요한 **CA** 역할이 있는지 확인하려면 다음을 수행합니다.

```
~]$ ipa server-role-find --role 'CA server'
```

```
-----  
2 server roles matched  
-----
```

```
Server name: server.idm.example.com  
Role name: CA server  
Role status: enabled
```

```
Server name: replica.idm.example.com  
Role name: CA server  
Role status: enabled  
-----
```

```
Number of entries returned 2  
-----
```

배포에는 두 개의 CA 서버가 있습니다.

3.

필요한 경우 현재 CA 갱신 서버인 CA 서버를 찾으려면 다음을 입력합니다.

```
~]$ ipa config-show | grep 'CA renewal'  
IPA CA renewal master: server.idm.example.com
```

현재 갱신 서버는 **server.idm.example.com** 입니다.

4.

갱신 서버 구성을 변경하려면 **--ca-renewal-master-server** 옵션과 함께 **ipa config-mod** 유틸리티를 사용합니다.

```
~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com | grep 'CA  
renewal'  
IPA CA renewal master: replica.idm.example.com
```

중요

다음을 사용하여 새 CA 갱신 서버로 전환할 수도 있습니다.

- **ipa-cacert-manage --renew** 명령 이 명령은 둘 다 CA 인증서를 갱신 하고 새 CA 갱신 서버를 실행하는 CA 서버를 만듭니다.
- **ipa-cert-fix** 명령 이 명령은 만료된 인증서로 인해 오류가 발생하면 배포를 복구합니다. 또한 명령을 새 CA 갱신 서버로 실행하는 CA 서버를 만듭니다.

자세한 내용은 [IdM이 오프라인인 경우 만료된 시스템 인증서 갱신을 참조하십시오.](#)

11.3. IDM에서 자체 서명된 CA로 외부에서 전환

외부에 서명된 IdM(Identity Management) 인증 기관(CA)의 자체 서명된 인증서로 전환하려면 이 절차를 완료합니다. 자체 서명된 CA에서는 CA 인증서 갱신이 자동으로 관리됩니다. 시스템 관리자는 CSR(인증서 서명 요청)을 외부 기관에 제출할 필요가 없습니다.

외부에 서명된 자체 서명된 CA로 전환하면 CA 인증서만 대체됩니다. 이전 CA에서 서명한 인증서는 계속 유효하며 여전히 사용 중입니다. 예를 들어 자체 서명된 CA로 이동한 후에도 LDAP 인증서의 인증서 체인은 변경되지 않고 유지됩니다.

external_CA certificate > IdM CA certificate > LDAP certificate

사전 요구 사항

- IdM CA 갱신 서버 및 모든 IdM 클라이언트 및 서버에 대한 루트 액세스 권한이 있습니다.

절차

1. IdM CA 갱신 서버에서 CA 인증서를 자체 서명으로 갱신합니다.

```
# ipa-cacert-manage renew --self-signed
Renewing CA certificate, please wait
CA certificate successfully renewed
The ipa-cacert-manage command was successful
```

2.

나머지 **IdM** 서버 및 클라이언트에 **root** 로 **SSH** 를 수행합니다. 예를 들어 다음과 같습니다.

```
# ssh root@idmclient01.idm.example.com
```

3.

IdM 클라이언트에서 서버의 인증서로 로컬 **IdM** 인증서 데이터베이스를 업데이트합니다.

```
[idmclient01 ~]# ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

4.

선택적으로 업데이트가 완료되고 새 **CA** 인증서가 **/etc/ipa/ca.crt** 파일에 추가되었는지 확인하려면 다음을 수행합니다.

```
[idmclient01 ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -
print_certs -text -noout
[...]
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 39 (0x27)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
    Validity
      Not Before: Jul  1 16:32:45 2019 GMT
      Not After : Jul  1 16:32:45 2039 GMT
    Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
  [...]

```

새 **CA** 인증서가 이전 **CA** 인증서를 사용하여 나열되므로 출력에 업데이트가 성공한 것으로 표시됩니다.

11.4. 외부 서명 인증서로 **IDM CA** 갱신 서버 갱신

이 섹션에서는 **CSR**(인증서 서명 요청)에 서명하도록 외부 **CA**를 사용하여 **IdM**(Identity Management) **CA**(인증 기관) 인증서를 갱신하는 방법을 설명합니다. 이 구성에서 **IdM CA** 서버는 외부 **CA**의 하위 **CA**입니다. 외부 **CA**는 **AD CS**(Active Directory 인증서 서버)일 수 있지만 반드시 필요한 것은 아닙니다.

외부 인증 기관이 **AD CS**인 경우 **CSR**에서 **IdM CA** 인증서에 원하는 템플릿을 지정할 수 있습니다. 인증서 템플릿은 인증서 요청이 수신될 때 **CA**에서 사용하는 정책 및 규칙을 정의합니다. **AD**의 인증서 템플릿은 **IdM**의 인증서 프로필에 해당합니다.

OID(Object Identifier)를 통해 특정 **AD CS** 템플릿을 정의할 수 있습니다. **OID**는 분산 애플리케이션의 데이터 요소, 구문 및 기타 부분을 고유하게 식별하기 위해 다양한 발행 기관에서 발행한 고유한 숫자 값입니다.

또는 특정 **AD CS** 템플릿을 이름으로 정의할 수 있습니다. 예를 들어 **IdM CA**가 **AD CS**에 제출한 **CSR**에 사용된 기본 프로필 이름은 **subCA**입니다.

CSR에서 **OID** 또는 이름을 지정하여 프로필을 정의하려면 **external-ca-profile** 옵션을 사용합니다. 자세한 내용은 **ipa-cacert-manage man** 페이지를 참조하십시오.

미리 작성된 인증서 템플릿을 사용하는 것 외에도 **AD CS**에 사용자 지정 인증서 템플릿을 생성하고 **CSR**에서 사용할 수도 있습니다.

사전 요구 사항

- **IdM CA** 갱신 서버에 대한 루트 액세스 권한이 있습니다.

절차

현재 **CA** 인증서가 자체 서명되었는지 여부에 관계없이 **IdM CA**의 인증서를 외부 서명으로 갱신하려면 이 절차를 완료합니다.

1.

외부 **CA**에 제출할 **CSR**을 생성합니다.

- 외부 **CA**가 **AD CS**인 경우 **--external-ca-type=ms-cs** 옵션을 사용합니다. 기본 **subCA** 템플릿과 다른 템플릿을 원하는 경우 **--external-ca-profile** 옵션을 사용하여 지정합니다.

```
~]# ipa-cacert-manage renew --external-ca --external-ca-type=ms-cs [--external-ca-profile=PROFILE]
```

Exporting CA certificate signing request, please wait

The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:

```
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
```

The ipa-cacert-manage command was successful

- 외부 **CA**가 **AD CS**가 아닌 경우:

```
~]# ipa-cacert-manage renew --external-ca
Exporting CA certificate signing request, please wait
The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-
manage as:
ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-
cert-file=/path/to/external_ca_certificate
The ipa-cacert-manage command was successful
```

출력은 **CSR**이 생성되었으며 **/var/lib/ipa/ca.csr** 파일에 저장되어 있음을 보여줍니다.

2.

/var/lib/ipa/ca.csr에 있는 **CSR**을 외부 **CA**에 제출합니다. 이 프로세스는 외부 **CA**로 사용할 서비스에 따라 다릅니다.

3.

발행된 인증서와 권한 있는 **Blob**에 대해 발행된 **CA**의 **CA** 인증서 체인(**CA** 인증서 체인)을 검색합니다.

-

외부 **CA**가 **AD CS**가 아닌 경우 **PEM** 파일입니다.

-

외부 **CA**가 **AD CS**인 경우 **Base_64** 인증서입니다.

이 프로세스는 모든 인증서 서비스에 따라 다릅니다. 일반적으로 웹 페이지 또는 알림 이메일의 다운로드 링크를 사용하면 관리자가 필요한 모든 인증서를 다운로드할 수 있습니다.

외부 **CA**가 **AD CS**이고 **Microsoft Windows** 인증 기관 관리 창을 통해 알려진 템플릿이 있는 **CSR**을 제출한 경우 **AD CS**가 인증서를 즉시 발행하고 **Save Certificate** 대화 상자가 **AD CS** 웹 인터페이스에 표시되어 발급된 인증서를 저장할 위치를 묻는 메시지가 표시됩니다.

4.

ipa-cacert-manage renew 명령을 다시 실행하여 전체 인증서 체인을 제공하는 데 필요한 모든 **CA** 인증서 파일을 추가합니다. **--external-cert-file** 옵션을 여러 번 사용하여 필요한 만큼 파일을 지정합니다.

```
~]# ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-
cert-file=/path/to/external_ca_certificate_1 --external-cert-
file=/path/to/external_ca_certificate_2
```

5.

모든 **IdM** 서버 및 클라이언트에서 서버의 인증서로 로컬 **IdM** 인증서 데이터베이스를 업데이트합니다.

```
[client ~]$ ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

6.

선택적으로 업데이트가 완료되고 새 **CA** 인증서가 **/etc/ipa/ca.crt** 파일에 추가되었는지 확인하려면 다음을 수행합니다.

```
[client ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -print_certs
-text -noout
[...]
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 39 (0x27)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
    Validity
      Not Before: Jul  1 16:32:45 2019 GMT
      Not After : Jul  1 16:32:45 2039 GMT
    Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
  [...]

```

새 **CA** 인증서가 이전 **CA** 인증서를 사용하여 나열되므로 출력에 업데이트가 성공한 것으로 표시됩니다.

12장. IDM이 오프라인 상태일 때 만료된 시스템 인증서 갱신

시스템 인증서가 만료된 경우 **IdM(Identity Management)**이 시작되지 않습니다. **IdM**은 **ipa-cert-fix** 툴을 사용하여 이 경우에도 시스템 인증서 갱신을 지원합니다.

- 호스트에 **ipactl start --ignore-service-failures** 명령을 입력하여 **LDAP** 서비스가 실행 중인 지 확인합니다.

12.1. CA 갱신 서버에서 만료된 시스템 인증서 갱신

이 섹션에서는 만료된 **IdM** 인증서에 **ipa-cert-fix** 툴을 적용하는 방법을 설명합니다.



중요

CA 갱신 서버가 아닌 **CA(Certificate Authority)** 호스트에서 **ipa-cert-fix** 툴을 실행하고 유틸리티가 공유 인증서를 갱신하는 경우 해당 호스트는 도메인의 새 **CA** 갱신 서버가 자동으로 됩니다. 불일치를 방지하려면 도메인에 항상 하나의 **CA** 갱신 서버만 있어야 합니다.

사전 요구 사항

- 관리 권한이 있는 서버에 로그인합니다.

절차

1. **ipa-cert-fix** 툴을 시작하여 시스템을 분석하고 갱신이 필요한 만료된 인증서를 나열합니다.

```
# ipa-cert-fix
...
The following certificates will be renewed:

Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial: 13
  Expires: 2019-05-12 05:55:47
...
Enter "yes" to proceed:
```

2. **yes** 를 입력하여 갱신 프로세스를 시작합니다.

```

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial: 268369925
  Expires: 2021-08-14 02:19:33
...

Becoming renewal master.
The ipa-cert-fix command was successful

```

ipa-cert-fix 가 만료된 모든 인증서를 갱신하기까지 최대 1분 정도 걸릴 수 있습니다.

3.

필요한 경우 모든 서비스가 실행되고 있는지 확인합니다.

```

# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful

```

이 시점에서 인증서가 갱신되어 서비스가 실행됩니다. 다음 단계는 **IdM** 도메인의 다른 서버를 확인하는 것입니다.

참고

여러 **CA** 서버에서 인증서를 복구해야 하는 경우:

1. 토폴로지에서 **LDAP** 복제가 작동하는지 확인한 후 먼저 위 절차에 따라 하나의 **CA** 서버에서 **ipa-cert-fix** 를 실행합니다.
2. 다른 **CA** 서버에서 **ipa-cert-fix** 를 실행하기 전에, 공유 인증서의 불필요한 갱신을 방지하기 위해 **getcrt-resubmit** (다른 **CA** 서버의 경우)을 통해 공유 인증서에 대한 **Certmonger** 갱신을 트리거합니다.

12.2. 갱신 후 IDM 도메인의 다른 IDM 서버 확인

ipa-cert-fix 툴을 사용하여 **CA** 갱신 서버의 인증서를 갱신한 후 다음을 수행해야 합니다.

- 도메인에서 다른 모든 **IdM(Identity Management)** 서버를 재시작합니다.
- **certmonger**에서 인증서를 갱신했는지 확인합니다.
- 만료된 시스템 인증서가 있는 다른 **CA(인증 기관)** 복제본이 있는 경우 **ipa-cert-fix** 툴을 사용하여 해당 인증서를 갱신합니다.

사전 요구 사항

- 관리 권한을 사용하여 서버에 로그인합니다.

절차

1. **--force** 매개변수를 사용하여 **IdM**을 다시 시작합니다.

```
# ipactl restart --force
```

--force 매개변수를 사용하면 **ipactl** 유틸리티에서 개별 서비스 시작 실패를 무시합니다. 예를 들어 서버가 만료된 인증서가 있는 **CA**인 경우 **pki-tomcat** 서비스가 시작되지 않습니다. 이는 **--force** 매개변수를 사용하므로 예상되고 무시됩니다.

2. 다시 시작한 후 **certmonger** 서비스가 인증서를 갱신했는지 확인합니다(인증서 상태가 **MONITORING**으로 표시됨).

```
# getcert list | egrep '^Request|status:|subject:'
Request ID '20190522120745':
    status: MONITORING
    subject: CN=IPA RA,O=EXAMPLE.COM 201905222205
Request ID '20190522120834':
    status: MONITORING
    subject: CN=Certificate Authority,O=EXAMPLE.COM 201905222205
...
```

certmonger 가 복제본의 공유 인증서를 갱신하는 데 다소 시간이 걸릴 수 있습니다.

- 3.

서버가 **CA**이기도 한 경우 이전 명령은 **pki-tomcat** 서비스에서 사용하는 인증서에 **CA_UNREACHABLE** 을 보고합니다.

```
Request ID '20190522120835':
status: CA_UNREACHABLE
subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
...
```

4.

이 인증서를 갱신하려면 **ipa-cert-fix** 유틸리티를 사용하십시오.

```
# ipa-cert-fix
Dogtag sslserver certificate:
Subject: CN=ca2.example.com,O=EXAMPLE.COM
Serial: 3
Expires: 2019-05-11 12:07:11

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
Subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
Serial: 15
Expires: 2019-08-14 04:25:05

The ipa-cert-fix command was successful
```

이제 모든 **IdM** 인증서가 갱신되어 올바르게 작동합니다.

12.3. 웹 서버 및 LDAP 서버 인증서 교체

IdM(Identity Management) 시스템 관리자는 **IdM** 서버에서 실행되는 웹(또는 **httpd**) 및 **LDAP**(또는 디렉터리) 서비스의 인증서를 수동으로 교체할 수 있습니다. 예를 들어 **certmonger** 유틸리티가 인증서를 자동으로 갱신하도록 구성되지 않았거나 인증서가 외부 **CA**(인증 기관)에서 서명한 경우 이 기능이 필요할 수 있습니다.

이 예에서는 **server.idm.example.com** **IdM** 서버에서 실행되는 서비스에 대한 인증서를 설치합니다. 외부 **CA**에서 인증서를 가져옵니다.



참고

HTTP 및 **LDAP** 서비스 인증서에는 다양한 **IdM** 서버에서 키 쌍 및 제목 이름이 다르므로 각 **IdM** 서버에서 개별적으로 인증서를 갱신해야 합니다.

사전 요구 사항

- IdM 서버에 대한 루트 액세스 권한이 있습니다.
- Directory Manager 비밀번호를 알고 있습니다.
- 외부 CA, *ca_certificate_chain_file.crt* 의 CA 인증서 체인을 저장하는 파일에 액세스할 수 있습니다.

절차

1. *ca_certificate_chain_file.crt* 에 포함된 인증서를 IdM에 대한 추가 CA 인증서로 설치합니다.

```
# ipa-cacert-manage install
```

2. *ca_certificate_chain_file.crt* 의 인증서로 로컬 IdM 인증서 데이터베이스를 업데이트합니다.

```
# ipa-certupdate
```

3. OpenSSL 유틸리티를 사용하여 개인 키 및 CSR(인증서 서명 요청)을 생성합니다.

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout new.key -out new.csr -
addext "subjectAltName = DNS:ipa-ca.idm.example.test" -subj
'/CN=server.idm.example.com,O=IDM.EXAMPLE.COM'
```

CSR을 외부 CA에 제출합니다. 이 프로세스는 외부 CA로 사용할 서비스에 따라 다릅니다. CA가 인증서에 서명한 후 인증서를 IdM 서버로 가져옵니다.

4. IdM 서버에서 Apache 웹 서버의 이전 개인 키와 인증서를 새 키 및 새로 서명된 인증서로 바꿉니다.

```
# ipa-server-certinstall -w --pin=password new.key new.crt
```

위의 명령에서:

- **w** 옵션은 웹 서버에 인증서를 설치하도록 지정합니다.

- **--pin** 옵션은 개인 키를 보호하는 암호를 지정합니다.

5. 메시지가 표시되면 디렉터리 관리자 암호를 입력합니다.

6. **LDAP** 서버의 이전 개인 키와 인증서를 새 키 및 새로 서명된 인증서로 바꿉니다.

```
# ipa-server-certinstall -d --pin=password new.key new.cert
```

위의 명령에서:

- **d** 옵션은 **LDAP** 서버에 인증서를 설치하도록 지정합니다.

- **--pin** 옵션은 개인 키를 보호하는 암호를 지정합니다.

7. 메시지가 표시되면 디렉터리 관리자 암호를 입력합니다.

8. **httpd** 서비스를 다시 시작합니다.

```
# systemctl restart httpd.service
```

9. **Directory** 서비스를 다시 시작하십시오.

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

추가 리소스

- [IdM에서 작동하도록 인증서 형식 변환](#)
- [ipa-server-certinstall\(1\) 매뉴얼 페이지](#)

13장. IDM CA 서버에서 CRL 생성

IdM 배포 시 포함된 CA(인증 기관)를 사용하는 경우 하나의 IdM(Identity Management) 서버에서 다른 IdM(Identity Management)으로 CRL(Certificate Revocation List)을 생성해야 할 수도 있습니다. 예를 들어 서버를 다른 시스템으로 마이그레이션하려는 경우 필요할 수 있습니다.

CRL을 생성하도록 하나의 서버만 구성합니다. CRL 게시자 역할을 수행하는 IdM 서버는 일반적으로 CA 갱신 서버 역할을 수행하는 동일한 서버이지만 필수는 아닙니다. CRL 게시자 서버를 해제하기 전에 CRL 게시자 서버 역할을 수행하도록 다른 서버를 선택하고 구성하십시오.

이 장에서는 다음을 설명합니다.

- IdM 서버에서 CRL 생성 중지.
- IdM 복제본에서 CRL 생성 시작

13.1. IDM 서버에서 CRL 생성 중지

IdM CRL 게시자 서버에서 CRL(Certificate Revocation List) 생성을 중지하려면 `ipa-crlgen-manage` 명령을 사용합니다. 생성을 비활성화하기 전에 서버가 실제로 CRL을 생성하는지 확인합니다. 그러면 비활성화할 수 있습니다.

사전 요구 사항

- `root`로 로그인해야 합니다.

절차

1. 서버가 CRL을 생성 중인지 확인합니다.

```
[root@server ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2.

서버에서 **CRL** 생성을 중지합니다.

```
[root@server ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL
generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3.

서버가 **CRL** 생성을 중지했는지 확인합니다.

```
[root@server ~]# ipa-crlgen-manage status
```

서버가 **CRL** 생성을 중지했습니다. 다음 단계는 **IdM** 복제본에서 **CRL** 생성을 활성화하는 것입니다.

13.2. IDM 복제본 서버에서 CRL 생성 시작

ipa-crlgen-manage 명령을 사용하여 **IdM CA** 서버에서 **CRL**(Certificate Revocation List)을 생성할 수 있습니다.

사전 요구 사항

- **RHEL** 시스템은 **IdM** 인증 기관 서버여야 합니다.
- **root**로 로그인해야 합니다.

절차

1.

CRL 생성을 시작합니다.

```
[root@replica1 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
```

CRL generation enabled on the local host. Please make sure to have only a single CRL generation master.

The ipa-crlgen-manage command was successful

2.

CRL이 생성되었는지 확인합니다.

[root@replica1 ~]# ipa-crlgen-manage status

CRL generation: enabled

Last CRL update: 2019-10-31 12:10:00

Last CRL Number: 7

The ipa-crlgen-manage command was successful

14장. CA 갱신 서버 및 CRL 게시자 역할을 수행하는 서버 해제

CA(인증 기관) 갱신 서버 역할과 CRL(Certificate Revocation List) 게시자 역할을 모두 수행하는 하나의 서버가 있을 수 있습니다. 이 서버를 오프라인으로 전환하거나 해제해야 하는 경우 다른 **CA** 서버를 선택하고 구성하여 이러한 역할을 수행합니다.

이 예에서는 **CA** 갱신 서버 및 **CRL** 게시자 역할을 수행하는 호스트 **server.idm.example.com** 을 해제해야 합니다. 이 절차에서는 **CA** 갱신 서버 및 **CRL** 게시자 역할을 호스트 **replica.idm.example.com** 에 전송하고 **IdM** 환경에서 **server.idm.example.com** 을 제거합니다.



참고

CA 갱신 서버 및 **CRL** 게시자 역할을 모두 수행하도록 동일한 서버를 구성할 필요가 없습니다.

사전 요구 사항

- **IdM** 관리자 인증 정보가 있습니다.
- 해제 중인 서버의 루트 암호가 있습니다.
- **IdM** 환경에 두 개 이상의 **CA** 복제본이 있습니다.

절차

1. **IdM** 관리자 자격 증명을 가져옵니다.

```
[user@server ~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. (선택 사항) **CA** 갱신 서버 및 **CRL** 게시자 역할을 수행하는 서버가 확실하지 않은 경우:

- a. 현재 **CA** 갱신 서버를 표시합니다. **IdM** 서버에서 다음 명령을 실행할 수 있습니다.

```
[user@server ~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: server.idm.example.com
```

b.

호스트가 현재 **CRL** 게시자인지 테스트합니다.

```
[user@server ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

CRL을 생성하지 않는 **CA** 서버는 **CRL** 생성: **disabled** 를 표시합니다.

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: disabled
The ipa-crlgen-manage command was successful
```

CRL 게시자 서버를 찾을 때까지 **CA** 서버에 이 명령을 계속 입력합니다.

c.

이러한 역할을 충족하도록 승격할 수 있는 다른 모든 **CA** 서버를 표시합니다. 이 환경에는 두 개의 **CA** 서버가 있습니다.

```
[user@server ~]$ ipa server-role-find --role 'CA server'
-----
2 server roles matched
-----
Server name: server.idm.example.com
Role name: CA server
Role status: enabled
Server name: replica.idm.example.com
Role name: CA server
Role status: enabled
-----
Number of entries returned 2
-----
```

3.

replica.idm.example.com 을 **CA** 갱신 서버로 설정합니다.

```
[user@server ~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com
```

4.

server.idm.example.com 에서 :

a.

인증서 업데이트 프로그램 작업을 비활성화합니다.

```
[root@server ~]# pki-server ca-config-set ca.certStatusUpdateInterval 0
```

- b. IdM 서비스를 다시 시작하십시오.

```
[user@server ~]$ ipactl restart
```

5. replica.idm.example.com 에서 :

- a. 인증서 업데이트 프로그램 작업을 활성화합니다.

```
[root@server ~]# pki-server ca-config-unset ca.certStatusUpdateInterval
```

- b. IdM 서비스를 다시 시작하십시오.

```
[user@replica ~]$ ipactl restart
```

6. server.idm.example.com 에서 CRL 생성을 중지합니다.

```
[user@server ~]$ ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL
generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

7. replica.idm.example.com 에서 CRL 생성을 시작합니다.

```
[user@replica ~]$ ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

8. **server.idm.example.com** 에서 **IdM** 서비스를 중지합니다.

```
[user@server ~]$ ipactl stop
```

9. **replica.idm.example.com** 에서 **IdM** 환경에서 **server.idm.example.com** 을 삭제합니다.

```
[user@replica ~]$ ipa server-del server.idm.example.com
```

10. **server.idm.example.com** 에서 **ipa-server-install --uninstall** 명령을 루트 계정으로 사용합니다.

```
[root@server ~]# ipa-server-install --uninstall
...
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

검증 단계

- 현재 **CA** 갱신 서버를 표시합니다.
- ```
[user@replica ~]$ ipa config-show | grep 'CA renewal'
IPA CA renewal master: replica.idm.example.com
```
- **replica.idm.example.com** 호스트에서 **CRL**을 생성하는지 확인합니다.

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

#### 추가 리소스

- [IdM CA 갱신 서버 변경 및 재설정](#)
- [IdM CA 서버에서 CRL 생성](#)
- [IdM 복제본 설치 제거](#)

## 15장. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기

### 15.1. CERTMONGER 개요

**IdM(Identity Management)**이 통합된 **IdM 인증 기관(CA)**과 함께 설치되면 **certmonger** 서비스를 사용하여 시스템 및 서비스 인증서를 추적하고 갱신합니다. 인증서가 만료일에 도달하면 **certmonger**는 다음을 통해 갱신 프로세스를 관리합니다.

- 원래 요청에 제공된 옵션을 사용하여 인증서 서명 요청(**CSR**)을 다시 생성합니다.
- **IdM API cert-request** 명령을 사용하여 **IdM CA**에 **CSR**을 제출합니다.
- **IdM CA**에서 인증서 수신.
- 원래 요청으로 지정한 경우 사전 저장 명령을 실행합니다.
- 업데이트 요청에 지정된 위치에 새 인증서 설치: **NSS** 데이터베이스 또는 파일에 있습니다.
- 원래 요청으로 지정한 경우 저장 후 명령을 실행합니다. 예를 들어 **post-save** 명령은 서비스가 새 인증서를 선택하도록 **certmonger**에 관련 서비스를 재시작하도록 지시할 수 있습니다.

#### 인증서 certmonger 트랙

인증서는 시스템 및 서비스 인증서로 나눌 수 있습니다.

서비스 인증서(예: **HTTP,LDAP** 및 **PKINIT**)와 달리, 서로 다른 서버에 키 쌍과 제목 이름이 다른 경우 **IdM** 시스템 인증서와 해당 키는 모든 **CA** 복제본에서 공유합니다. **IdM** 시스템 인증서는 다음과 같습니다.

- **IdM CA 인증서**
- **OCSP 서명 인증서**

- **IdM CA 하위 시스템 인증서**
- **IdM CA 감사 서명 인증서**
- **IdM 갱신 에이전트 (RA) 인증서**
- **KRA 전송 및 스토리지 인증서**

**certmonger** 서비스는 통합된 **CA**를 사용하여 **IdM** 환경 설치 중에 요청된 **IdM** 시스템 및 서비스 인증서를 추적합니다. **certmonger**는 **IdM** 호스트에서 실행되는 다른 서비스를 위해 시스템 관리자가 수동으로 요청한 인증서도 추적합니다. **certmonger**는 외부 **CA** 인증서 또는 사용자 인증서를 추적하지 않습니다.

#### **certmonger** 구성 요소

**certmonger** 서비스는 두 가지 주요 구성 요소로 구성됩니다.

- **certmonger** 데몬 - 인증서 목록을 추적하고 갱신 명령 시작
- 시스템 관리자가 **certmonger** 데몬에 적극적으로 명령을 보낼 수 있는 **CLI**(명령줄 인터페이스)용 **getcert** 유틸리티입니다.

특히 시스템 관리자는 **getcert** 유틸리티를 사용하여 다음을 수행할 수 있습니다.

- [새 인증서 요청](#)
- [certmonger](#) 트랙의 인증서 목록 보기
- [인증서 추적 시작 또는 중지](#)
- [인증서 업데이트](#)

## 15.2. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기

**IdM(Identity Management)** 클라이언트에서 실행되는 브라우저와 웹 서비스 간 통신이 안전하고 암호화되도록 하려면 **TLS** 인증서를 사용합니다. **IdM CA(인증 기관)**에서 웹 서비스의 **TLS** 인증서를 가져옵니다.

이 섹션에서는 **certmonger** 를 사용하여 서비스에 대한 **IdM** 인증서 (**HTTP/my\_company.idm.example.com@IDM.EXAMPLE.COM**)를 **IdM** 클라이언트에서 실행하는 방법을 설명합니다.

**certmonger** 를 사용하여 인증서를 자동으로 요청한다는 것은 **certmonger** 가 갱신이 발생할 때 인증서를 관리하고 갱신한다는 것을 의미합니다.

**certmonger** 가 서비스 인증서를 요청할 때 발생하는 상황을 시각적으로 표현하려면 **15.3절. “서비스 인증서를 요청하는 **certmonger**의 통신 흐름”** 을 참조하십시오.

### 사전 요구 사항

- 웹 서버는 **IdM** 클라이언트로 등록되어 있습니다.
- 프로시저를 실행 중인 **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.
- 인증서를 요청하는 서비스는 **IdM**에 사전 존재할 필요가 없습니다.

### 절차

1. **HTTP** 서비스가 실행 중인 **my\_company.idm.example.com** **IdM** 클라이언트에서 **HTTP/my\_company.idm.example.com@IDM.EXAMPLE.COM** 주체에 해당하는 서비스에 대한 인증서를 요청하며 이를 지정합니다.
  - 인증서는 로컬 **/etc/pki/tls/certs/httpd.pem** 파일에 저장됩니다.
  - 개인 키는 로컬 **/etc/pki/tls/private/httpd.key** 파일에 저장해야 합니다.
  - **SubjectAltName** 에 대한 **extensionRequest**가 **my\_company.idm.example.com** 의

DNS 이름을 사용하여 서명 요청에 추가됩니다.

```
ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

위의 명령에서:

- **ipa-getcert request** 명령은 인증서를 IdM CA에서 가져올 수 있도록 지정합니다. **ipa-getcert request** 명령은 **getcert request -c IPA** 의 바로 가기입니다.
- **g** 옵션은 아직 없는 경우 생성할 키 크기를 지정합니다.
- **d** 옵션은 요청에 추가할 **SubjectAltName DNS** 값을 지정합니다.
- **-C** 옵션은 인증서를 가져온 후 **httpd** 서비스를 다시 시작하도록 **certmonger** 에 지시합니다.
- 특정 프로필로 인증서를 발행하도록 지정하려면 **-T** 옵션을 사용합니다.
- 지정된 **CA**에서 이름이 지정된 발행자를 사용하여 인증서를 요청하려면 **-X ISSUER** 옵션을 사용합니다.

2.

필요한 경우 요청 상태를 확인하려면 다음을 수행합니다.

```
ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
 status: MONITORING
 stuck: no
 key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
 certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
 CA: IPA
[...]
```

출력은 요청이 **MONITORING** 상태에 있음을 보여줍니다. 즉, 인증서를 가져온 것입니다. 키 쌍과 인증서의 위치는 요청된 위치입니다.

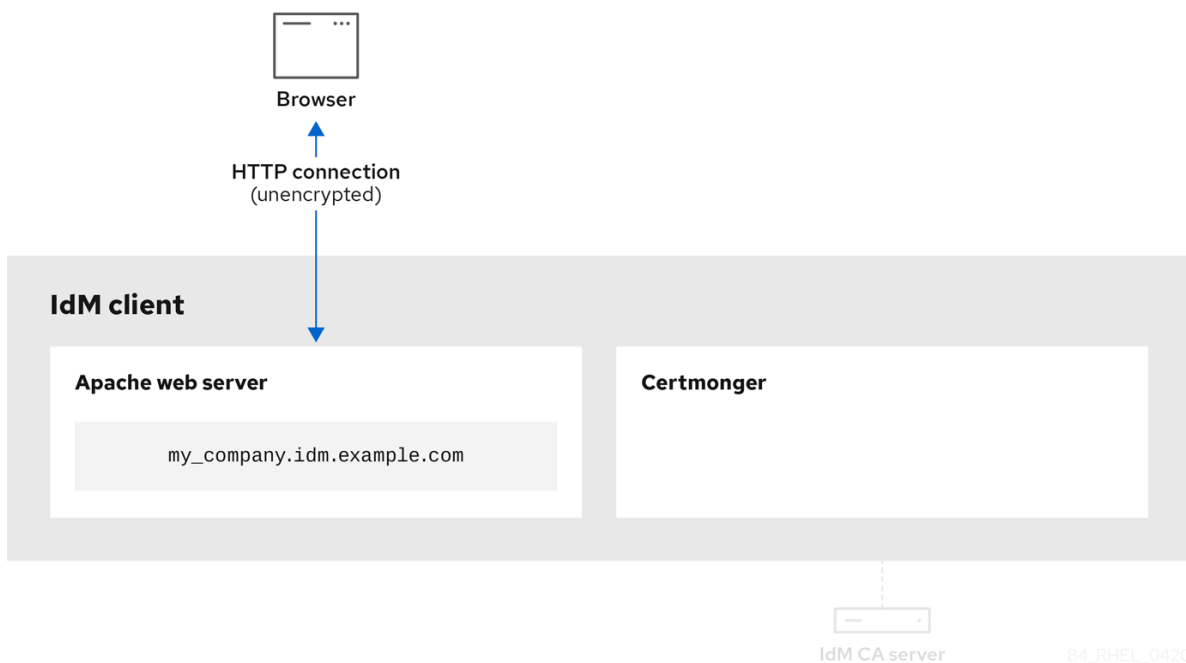
### 15.3. 서비스 인증서를 요청하는 CERTMONGER의 통신 흐름

이 섹션의 다이어그램은 **certmonger**가 **IdM(Identity Management)** 인증 기관(**CA**) 서버에서 서비스 인증서를 요청할 때 발생하는 단계를 보여줍니다. 시퀀스는 다음 다이어그램으로 구성됩니다.

- 암호화되지 않은 통신
- 서비스 인증서 요청
- 서비스 인증서를 발급하는 **IdM CA**
- 서비스 인증서 적용
- **certmonger**에서 이전 인증서가 만료되는 경우 새 인증서를 요청합니다.

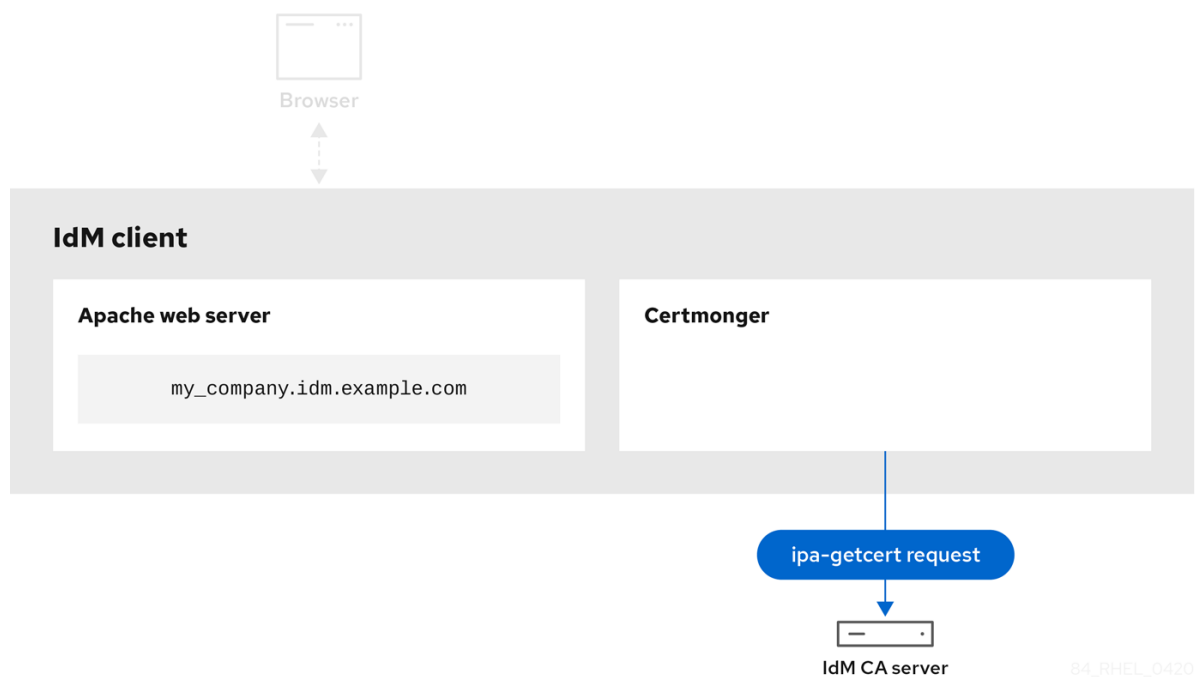
**암호화되지 않은 통신**에서는 초기 상황을 보여줍니다. **HTTPS** 인증서가 없으면 웹 서버와 브라우저 간 통신이 암호화되지 않습니다.

그림 15.1. 암호화되지 않은 통신



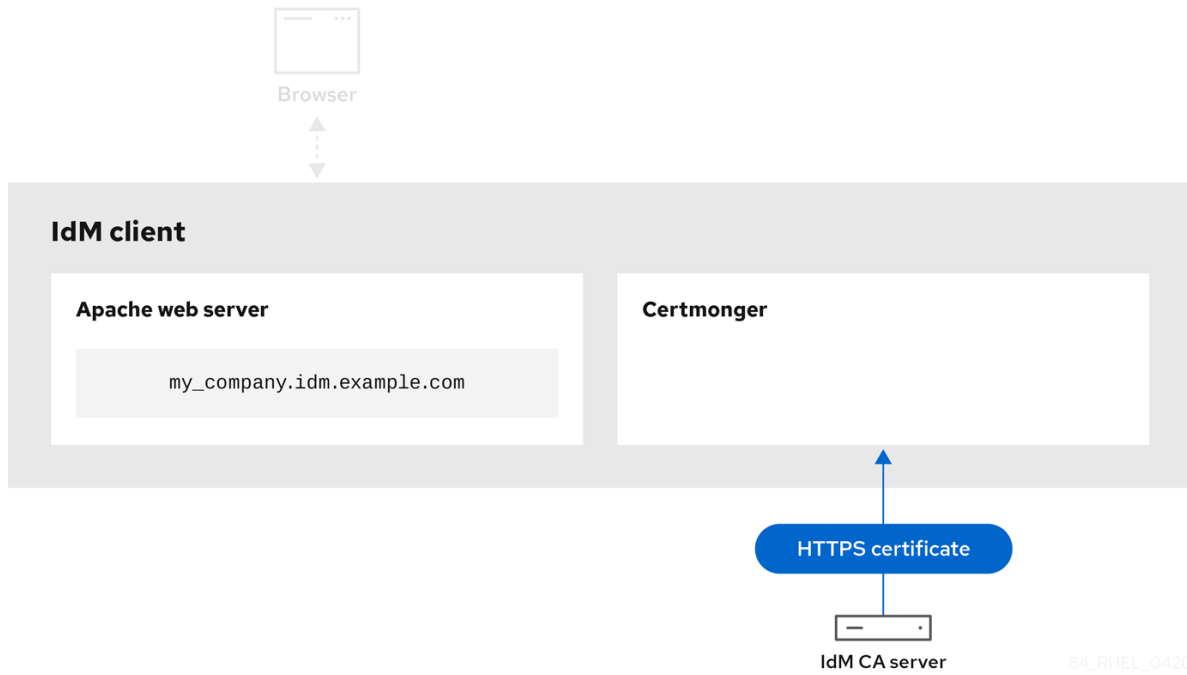
서비스 인증서를 요청하는 **certmonger** 는 **certmonger** 를 사용하여 **Apache** 웹 서버에 대한 **HTTPS** 인증서를 수동으로 요청하는 시스템 관리자에게 보여줍니다. 웹 서버 인증서를 요청할 때 **certmonger**는 **CA**와 직접 통신하지 않습니다. **IdM**을 통해 프록시합니다.

그림 15.2. 서비스 인증서 요청



서비스 인증서를 발행하는 **IdM CA**는 웹 서버의 **HTTPS** 인증서를 발행하는 **IdM CA**를 보여줍니다.

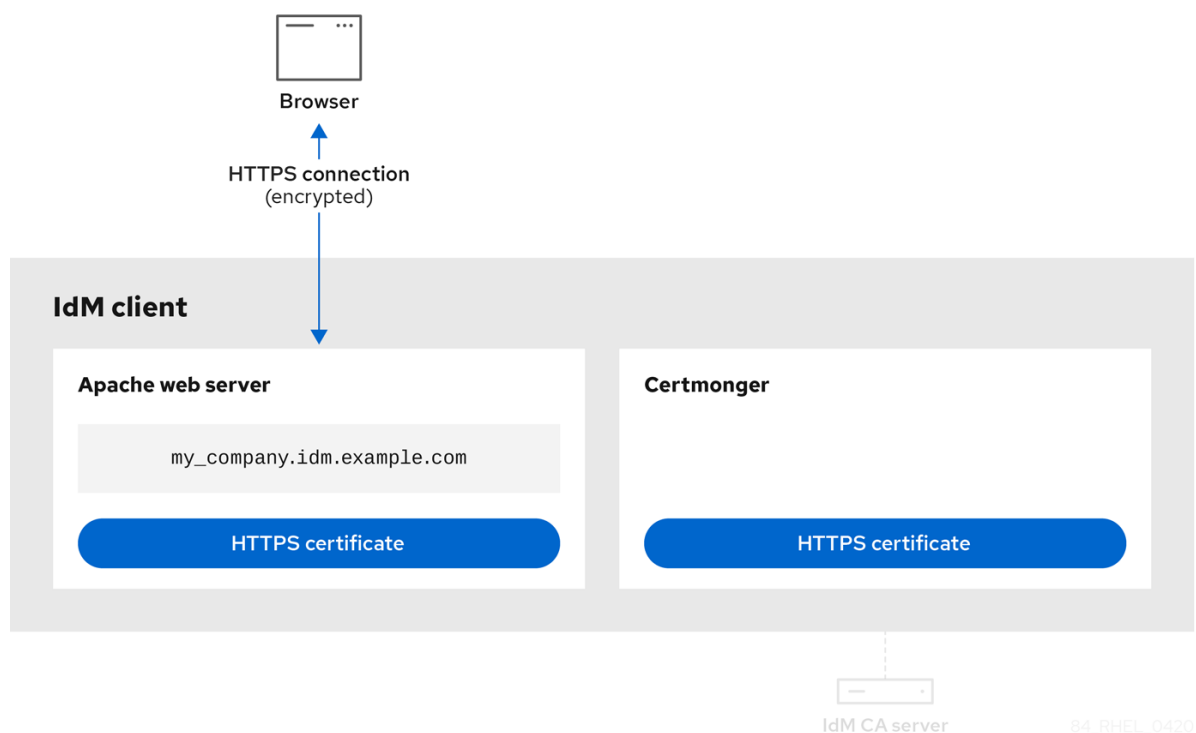
그림 15.3. 서비스 인증서를 발급하는 IdM CA



84\_RHEL\_0420

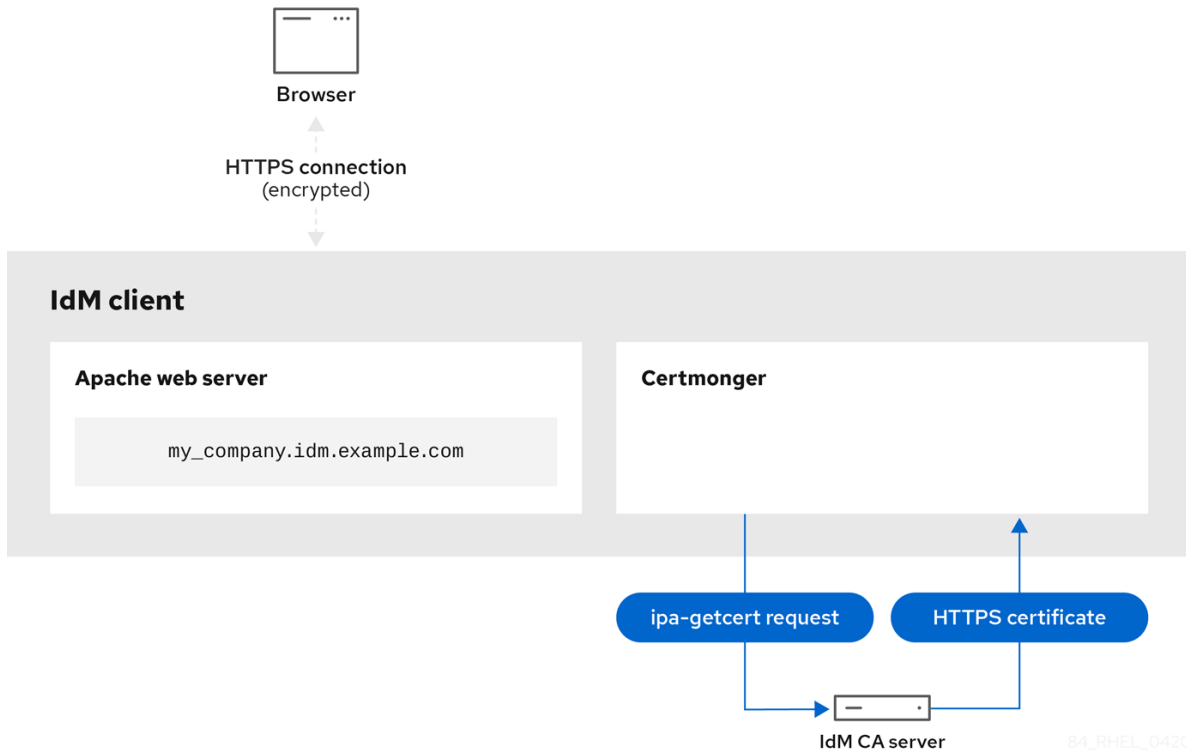
서비스 인증서를 적용하는 **certmonger** 는 IdM 클라이언트의 적절한 위치에 **HTTPS** 인증서를 배치하고 이를 수행하라는 지침이 있는 **certmonger**를 사용하여 **httpd** 서비스를 다시 시작합니다. **Apache** 서버에서는 **HTTPS** 인증서를 사용하여 자체와 브라우저 간 트래픽을 암호화합니다.

그림 15.4. 서비스 인증서 적용



만료일이 다가오는 경우 **certmonger**에서 새 인증서를 요청하는 **certmonger**가 인증서 만료 전에 **IdM CA**에서 서비스 인증서 갱신을 자동으로 요청합니다. **IdM CA**는 새 인증서를 발행합니다.

그림 15.5. **certmonger**에서 이전 인증서가 만료되는 경우 새 인증서를 요청합니다.



#### 15.4. CERTMONGER에서 추적한 인증서 요청의 세부 정보 보기

**certmonger** 서비스는 인증서 요청을 모니터링합니다. 인증서에 대한 요청이 성공적으로 서명되면 인증서가 생성됩니다. **certmonger**는 결과 인증서를 포함하여 인증서 요청을 관리합니다. 이 섹션에서는 **certmonger**에서 관리하는 특정 인증서 요청의 세부 정보를 확인하는 방법을 설명합니다.

##### 절차

- - 인증서 요청을 지정하는 방법을 알고 있는 경우 해당 특정 인증서 요청의 세부 사항만 나열합니다. 예를 들어 다음을 지정할 수 있습니다.
  - 요청 ID
  - 인증서의 위치
  - 인증서 닉네임

예를 들어 요청 ID가 **20190408146**인 인증서의 세부 정보를 보려면 **-v** 옵션을 사용하여 인증서에 대한 요청이 실패한 경우 오류의 모든 세부 정보를 볼 수 있습니다.

```
getcert list -i 20190408143846 -v
Number of certificates and requests being tracked: 16.
Request ID '20190408143846':
 status: MONITORING
 stuck: no
 key pair storage: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/dirsrv/slapd-
IDM-EXAMPLE-COM/pwdfile.txt'
 certificate: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB'
 CA: IPA
 issuer: CN=Certificate Authority,O=IDM.EXAMPLE.COM
 subject: CN=r8server.idm.example.com,O=IDM.EXAMPLE.COM
 expires: 2021-04-08 16:38:47 CEST
 dns: r8server.idm.example.com
 principal name: ldap/server.idm.example.com@IDM.EXAMPLE.COM
 key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
 eku: id-kp-serverAuth,id-kp-clientAuth
 pre-save command:
 post-save command: /usr/libexec/ipa/certmonger/restart_dirsrv IDM-EXAMPLE-
COM
 track: yes
 auto-renew: yes
```

출력에는 인증서에 대한 여러 정보가 표시됩니다. 예를 들면 다음과 같습니다.

- 위의 예에서 인증서 위치. **/etc/dirsrv/slapd-IDM-EXAMPLE-COM** 디렉터리의 **NSS** 데이터베이스입니다.
- 인증서 닉네임; 위의 예에서 **Server-Cert**입니다.
- 위 예제에서 편을 저장하는 파일은 **/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt**입니다.
- 인증서를 갱신하는 데 사용할 **CA**(인증 기관)는 **IPA CA**입니다.
- 위의 예에서 만료 날짜는 **2021-04-08 16:38:47 CEST**입니다.
-

인증서 상태; 위의 예에서 **MONITORING** 상태는 인증서가 유효하고 추적됨을 나타냅니다.

○

위의 예에서 **post-save** 명령은 **LDAP** 서비스를 다시 시작합니다.

●

인증서 요청을 지정하는 방법을 모르는 경우 **certmonger**가 모니터링 중이거나 획득하려는 모든 인증서의 세부 정보를 나열합니다.

### # getcert list

추가 리소스

●

**getcert** 목록 도움말 페이지를 참조하십시오.

## 15.5. 인증서 추적 시작 및 중지

이 섹션에서는 **getcert stop-tracking** 및 **getcert start-tracking** 명령을 사용하여 인증서를 모니터링하는 방법을 설명합니다. 두 명령은 **certmonger** 서비스에서 제공합니다. 인증서 추적을 활성화하면 **IdM(Identity Management)** 인증 기관(CA)에서 다른 **IdM** 클라이언트의 시스템으로 발급한 인증서를 가져온 경우 특히 유용합니다. 인증서 추적을 활성화하는 것도 다음 프로비저닝 시나리오의 최종 단계가 될 수 있습니다.

1.

**IdM** 서버에서 아직 존재하지 않는 시스템의 인증서를 생성합니다.

2.

새 시스템을 생성합니다.

3.

새 시스템을 **IdM** 클라이언트로 등록합니다.

4.

의 **IdM** 서버에서 **IdM** 클라이언트로 인증서와 키를 가져옵니다.

5.

**certmonger**를 사용하여 인증서 추적을 시작하여 만료될 때 갱신되도록 합니다.

절차

•

20190408143846의 요청 ID가 있는 인증서 모니터링을 비활성화하려면 다음을 실행합니다.

```
getcert stop-tracking -i 20190408143846
```

자세한 내용은 `getcert stop-tracking man` 페이지를 참조하십시오.

•

개인 키가 `/tmp/some_key.key` 파일에 저장되어 있는 `/tmp/some_cert.crt` 파일에 저장된 인증서의 모니터링을 활성화하려면 다음을 수행합니다.

```
getcert start-tracking -c IPA -f /tmp/some_cert.crt -k /tmp/some_key.key
```

`certmonger`는 인증서를 발급한 CA 유형을 자동으로 식별할 수 없습니다. 따라서 IdM CA에서 인증서를 발급한 경우 `getcert start-tracking` 명령에 IPA 값이 있는 `-c` 옵션을 추가합니다. `-c` 옵션을 추가하도록 생략하면 `certmonger`가 `NEED_CA` 상태가 됩니다.

자세한 내용은 `getcert start-tracking man` 페이지를 참조하십시오.



#### 참고

두 명령에서는 인증서를 조작하지 않습니다. 예를 들어, `getcert stop-tracking`은 인증서를 삭제하거나 NSS 데이터베이스 또는 파일 시스템에서 제거하지만 모니터링된 인증서 목록에서 인증서를 제거하기만 하면 됩니다. 마찬가지로 `getcert start-tracking`은 모니터링된 인증서 목록에 인증서만 추가합니다.

## 15.6. 인증서를 수동으로 업데이트

인증서가 만료일이 되면 `certmonger` 데몬은 CA(인증 기관) 도우미를 사용하여 갱신 명령을 자동으로 발행하고, 갱신된 인증서를 가져와 이전 인증서를 새 인증서로 대체합니다.

`getcert resubmit` 명령을 사용하여 사전에 인증서를 수동으로 갱신할 수도 있습니다. 이렇게 하면 인증서에서 포함하는 정보를 업데이트할 수 있습니다(예: SAN(주체 대체 이름)).

이 섹션에서는 인증서를 수동으로 갱신하는 방법에 대해 설명합니다.

### 절차

•

20190408143846의 요청 ID로 인증서를 갱신하려면 다음을 수행합니다.

```
getcert resubmit -i 20190408143846
```

특정 인증서에 대한 요청 ID를 얻으려면 **getcert list** 명령을 사용합니다. 자세한 내용은 **getcert list man** 페이지를 참조하십시오.

## 15.7. CERTMONGER는 CA 복제본에서 IDM 인증서 추적을 재개합니다.

다음 절차에서는 인증서 추적이 중단된 후 통합 인증 기관을 사용한 IdM 배포에 중요한 IdM(Identity Management) 시스템 인증서 추적을 재개하는 방법을 설명합니다. 시스템 인증서를 갱신하는 동안 IdM 호스트에서 또는 복제 토폴로지가 제대로 작동하지 않아 IdM 호스트가 중단될 수 있습니다. 또한 **certmonger**를 통해 IdM 서비스 인증서 추적, 즉 HTTP,LDAP 및 PKINIT 인증서의 추적을 재개하는 방법도 설명합니다.

### 사전 요구 사항

•

시스템 인증서 추적을 재개할 호스트는 IdM CA 갱신 서버가 아닌 IdM 인증 기관(CA)이기도 합니다.

### 절차

1.

하위 시스템 CA 인증서에 대한 pin을 가져옵니다.

```
grep 'internal=' /var/lib/pki/pki-tomcat/conf/password.conf
```

2.

하위 시스템 CA 인증서에 추적을 추가하고 아래 명령에서 [내부 Pin]을 이전 단계에서 얻은 PIN으로 교체합니다.

```
getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "caSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
-T caCACert "caSigningCert cert-pki-ca"
```

```
getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "auditSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
-T caSignedLogCert "auditSigningCert cert-pki-ca"
```

```
getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "ocspSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
```

```
-T caOCSPCert "ocspSigningCert cert-pki-ca"
```

```
getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "subsystemCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
-T caSubsystemCert "subsystemCert cert-pki-ca"
```

```
getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "Server-Cert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
-T caServerCert "Server-Cert cert-pki-ca"
```

3.

나머지 **IdM** 인증서, **HTTP,LDAP,IPA** 갱신 에이전트 및 **PKINIT** 인증서에 대한 추적을 추가합니다.

```
getcert start-tracking -f /var/lib/ipa/certs/httpd.crt -k /var/lib/ipa/private/httpd.key -p
/var/lib/ipa/passwds/idm.example.com-443-RSA -c IPA -C
/usr/libexec/ipa/certmonger/restart_httpd -T caIPAServiceCert
```

```
getcert start-tracking -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM -n "Server-Cert" -c IPA
-p /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt -C
'/usr/libexec/ipa/certmonger/restart_dirsrv -T caIPAServiceCert "IDM-EXAMPLE-COM"
```

```
getcert start-tracking -f /var/lib/ipa/ra-agent.pem -k /var/lib/ipa/ra-agent.key -c
dogtag-ipa-ca-renew-agent -B /usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_ra_cert -T caSubsystemCert
```

```
getcert start-tracking -f /var/kerberos/krb5kdc/kdc.crt -k
/var/kerberos/krb5kdc/kdc.key -c dogtag-ipa-ca-renew-agent -B
/usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_kdc_cert -T KDCs_PKINIT_Certs
```

4.

**certmonger** 를 다시 시작하십시오.

```
systemctl restart certmonger
```

5.

**certmonger** 가 시작된 후 1 분 정도 기다린 다음 새 인증서의 상태를 확인합니다.

```
getcert list
```

추가 리소스

•

**IdM** 시스템 인증서가 모두 만료된 경우 **CA** 갱신 서버 및 **CRL** 게시자 서버이기도 하는 **IdM CA** 서버에서 **IdM** 시스템 인증서를 수동으로 갱신하려면 [기술 센터 지원\(KCS\) 솔루션](#)을 참조하십시오. 그런 다음 이 [KCS 솔루션](#)에 설명된 절차에 따라 토폴로지의 다른 모든 **CA** 서버에서 **IdM** 시스템 인증서를 수동으로 갱신합니다.

## 16장. RHEL 시스템 역할을 사용하여 인증서 요청

인증서 문제 및 업데이트 시스템 역할을 사용하여 인증서를 발급하고 관리할 수 있습니다.

이 장에서는 다음 주제를 다룹니다.

- [인증서 시스템 역할](#)
- [인증서 시스템 역할을 사용하여 자체 서명된 새 인증서 요청](#)
- [인증서 시스템 역할을 사용하여 IdM CA에서 새 인증서 요청](#)

### 16.1. 인증서 문제 및 업데이트 시스템 역할

인증서 문제 및 업데이트 시스템 역할을 사용하여 **Ansible Core**를 사용하여 **TLS** 및 **SSL** 인증서를 발급 및 갱신할 수 있습니다.

이 역할은 **certmonger**를 인증서 공급자로 사용하며 현재 자체 서명된 인증서를 발행 및 갱신하고 **IdM** 통합 인증 기관(**CA**)을 사용합니다.

인증서 문제 및 업데이트 시스템 역할에서 **Ansible** 플레이북에서 다음 변수를 사용할 수 있습니다.

#### certificate\_wait

작업이 인증서가 발급될 때까지 기다려야 하는지 여부를 지정하려면 다음을 수행합니다.

#### certificate\_requests

실행할 각 인증서 및 해당 매개 변수를 나타냅니다.

#### 추가 리소스

- [/usr/share/ansible/roles/rhel-system-roles.certificate/README.md](#) 파일을 참조하십시오.

•

**RHEL 시스템 역할 시작**을 참조하십시오.

## 16.2. 인증서 문제 및 업데이트 시스템 역할을 사용하여 자체 서명된 새 인증서 요청

인증서 문제 및 업데이트 시스템 역할을 통해 **Ansible Core**를 사용하여 자체 서명된 인증서를 발행할 수 있습니다.

이 프로세스는 **certmonger** 공급자를 사용하고 **getcert** 명령을 통해 인증서를 요청합니다.



참고

기본적으로 **certmonger** 는 만료되기 전에 인증서를 자동으로 갱신합니다. **Ansible** 플레이북의 **auto\_renew** 매개변수를 **no** 로 설정하여 비활성화할 수 있습니다.

### 사전 요구 사항

•

**Ansible Core** 패키지는 제어 시스템에 설치됩니다.

•

플레이북을 실행할 시스템에 **rhel-system-roles** 패키지가 설치되어 있습니다.

**RHEL** 시스템 역할에 대한 자세한 내용과 이를 적용하는 방법에 대한 자세한 내용은 **RHEL 시스템 역할 시작하기**를 참조하십시오.

### 절차

1.

**선택 사항:** 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2.

인벤토리 파일을 열고 인증서를 요청할 호스트를 정의합니다. 예를 들면 다음과 같습니다.

```
[webserver]
server.idm.example.com
```

3.

플레이북 파일을 만듭니다(예: **request-certificate.yml**):

- 인증서를 요청할 호스트를 포함하도록 (예: **webserver**)를 설정합니다.
  - 다음을 포함하도록 **certificate\_requests** 변수를 설정합니다.
    - **name** 매개 변수를 원하는 인증서 이름(예: **mycert**)으로 설정합니다.
    - **dns** 매개 변수를 인증서에 포함할 도메인으로 설정합니다(예: **\*.example.com**).
    - **ca** 매개 변수를 자체 서명 으로 설정합니다.
  - 역할에 **rhel-system-roles.certificate** 역할을 설정합니다.
- 이 예에서는 플레이북 파일입니다.

```

- hosts: webserver

vars:
 certificate_requests:
 - name: mycert
 dns: "*.example.com"
 ca: self-sign

roles:
 - rhel-system-roles.certificate
```

4. 파일을 저장합니다.
5. 플레이북을 실행합니다.

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

#### 추가 리소스

- [/usr/share/ansible/roles/rhel-system-roles.certificate/README.md](#) 파일을 참조하십시오.

•

**ansible-playbook(1)** 매뉴얼 페이지를 참조하십시오.

### 16.3. 인증서 문제 및 업데이트 시스템 역할을 사용하여 IDM CA에서 새 인증서 요청

인증서 문제 및 업데이트 시스템 역할을 사용하면 통합 인증 기관(CA)과 함께 IdM 서버를 사용하는 동안 **anible-core** 를 사용하여 인증서를 발행할 수 있습니다. 따라서 IdM을 CA로 사용할 때 여러 시스템의 인증서 신뢰 체인을 효율적이고 일관되게 관리할 수 있습니다.

이 프로세스는 **certmonger** 공급자를 사용하고 **getcert** 명령을 통해 인증서를 요청합니다.



#### 참고

기본적으로 **certmonger** 는 만료되기 전에 인증서를 자동으로 갱신합니다. **Ansible** 플레이북의 **auto\_renew** 매개변수를 **no** 로 설정하여 비활성화할 수 있습니다.

#### 사전 요구 사항

•

**Ansible Core** 패키지는 제어 시스템에 설치됩니다.

•

플레이북을 실행할 시스템에 **rhel-system-roles** 패키지가 설치되어 있습니다.

**RHEL** 시스템 역할에 대한 자세한 내용과 이를 적용하는 방법에 대한 자세한 내용은 [RHEL 시스템 역할 시작하기](#)를 참조하십시오.

#### 절차

1.

**선택 사항:** 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2.

인벤토리 파일을 열고 인증서를 요청할 호스트를 정의합니다. 예를 들면 다음과 같습니다.

```
[webserver]
server.idm.example.com
```

3.

플레이북 파일을 만듭니다(예: `request-certificate.yml`):

- 인증서를 요청할 호스트를 포함하도록 (예: `webserver`)를 설정합니다.
- 다음을 포함하도록 `certificate_requests` 변수를 설정합니다.
  - `name` 매개 변수를 원하는 인증서 이름(예: `mycert`)으로 설정합니다.
  - `dns` 매개 변수를 인증서에 포함할 도메인으로 설정합니다(예: `www.example.com`).
  - `principal` 매개 변수를 설정하여 `HTTP/www.example.com@EXAMPLE.COM` 와 같은 Kerberos 보안 주체를 지정합니다.
  - `ca` 매개 변수를 `ipa` 로 설정합니다.
- 역할에 `rhel-system-roles.certificate` 역할을 설정합니다.

이 예에서는 플레이북 파일입니다.

```

- hosts: webserver
 vars:
 certificate_requests:
 - name: mycert
 dns: www.example.com
 principal: HTTP/www.example.com@EXAMPLE.COM
 ca: ipa

 roles:
 - rhel-system-roles.certificate
```

4.

파일을 저장합니다.

5.

플레이북을 실행합니다.

■

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

#### 추가 리소스

- [/usr/share/ansible/roles/rhel-system-roles.certificate/README.md](#) 파일을 참조하십시오.
- [ansible-playbook\(1\)](#) 매뉴얼 페이지를 참조하십시오.

### 16.4. 인증서 발행 및 업데이트 시스템 역할을 사용하여 인증서 발행 전 또는 이후에 실행할 명령 지정

인증서 시스템 문제 및 업데이트 역할을 사용하면 **Ansible Core**를 사용하여 인증서가 발행되거나 갱신된 후 이전과 후에 명령을 실행할 수 있습니다.

다음 예에서 관리자는 **www.example.com**의 자체 서명된 인증서를 발행하거나 갱신하기 전에 **httpd** 서비스를 중지한 후 나중에 다시 시작합니다.



#### 참고

기본적으로 **certmonger**는 만료되기 전에 인증서를 자동으로 갱신합니다. **Ansible** 플레이북의 **auto\_renew** 매개변수를 **no**로 설정하여 비활성화할 수 있습니다.

#### 사전 요구 사항

- **Ansible Core** 패키지는 제어 시스템에 설치됩니다.
- 플레이북을 실행할 시스템에 **rhel-system-roles** 패키지가 설치되어 있습니다.

**RHEL** 시스템 역할에 대한 자세한 내용과 이를 적용하는 방법에 대한 자세한 내용은 [RHEL 시스템 역할 시작하기](#)를 참조하십시오.

#### 절차

1. **선택 사항:** 인벤토리 파일을 생성합니다(예: **inventory.file**):

```
$ touch inventory.file
```

2.

인벤토리 파일을 열고 인증서를 요청할 호스트를 정의합니다. 예를 들면 다음과 같습니다.

```
[webserver]
server.idm.example.com
```

3.

플레이북 파일을 만듭니다(예: `request-certificate.yml`):

- 인증서를 요청할 호스트를 포함하도록 (예: `webserver`)를 설정합니다.
- 다음을 포함하도록 `certificate_requests` 변수를 설정합니다.
  - `name` 매개 변수를 원하는 인증서 이름(예: `mycert`)으로 설정합니다.
  - `dns` 매개변수를 인증서에 포함할 도메인으로 설정합니다(예: `www.example.com`).
  - `ca` 매개 변수를 사용하여 인증서를 발급할 CA(예: `self-sign`)로 설정합니다.
  - 이 인증서를 발행하거나 `systemctl stop httpd.service` 와 같이 갱신하기 전에 실행할 명령에 `run_before` 매개변수를 설정합니다.
  - 이 인증서가 실행된 후 실행할 `run_after` 매개 변수를 `systemctl start httpd.service` 와 같이 갱신합니다.
- 역할에 `rhel-system-roles.certificate` 역할을 설정합니다.

이 예에서는 플레이북 파일입니다.

```

- hosts: webserver
 vars:
 certificate_requests:
 - name: mycert
 dns: www.example.com
 ca: self-sign
```

```
run_before: systemctl stop httpd.service
run_after: systemctl start httpd.service
```

```
roles:
- rhel-system-roles.certificate
```

4. 파일을 저장합니다.
5. 플레이북을 실행합니다.

```
$ ansible-playbook -i inventory.file request-certificate.yml
```

#### 추가 리소스

- [/usr/share/ansible/roles/rhel-system-roles.certificate/README.md](#) 파일을 참조하십시오.
- [ansible-playbook\(1\)](#) 매뉴얼 페이지를 참조하십시오.

## 17장. 인증서 서브 세트만 신뢰하도록 애플리케이션 제한

**IdM(Identity Management)** 설치가 통합된 **CA(Certificate System)** 인증 기관(CA)으로 구성된 경우 경량의 하위 CA를 생성할 수 있습니다. 생성하는 모든 하위 CA는 인증서 시스템의 기본 CA인 **ipa CA**로 분류됩니다.

이 컨텍스트의 **경량 하위 CA**는 **하위 CA**가 특정 목적으로 인증서를 발급한다는 것을 의미합니다. 예를 들어, 경량 하위 CA를 사용하면 가상 사설 네트워크(VPN) 게이트웨이 및 웹 브라우저와 같은 서비스를 구성하여 **하위 CA A**에서 발급한 인증서만 허용할 수 있습니다. **하위 CA B**에서만 발급한 인증서만 수락하도록 다른 서비스를 구성하면 **하위 CA**, 즉 **ipa CA** 및 둘 사이의 중간 하위 CA에서 발급한 인증서를 수락하지 못합니다.

하위 CA의 중간 인증서를 취소하면 이 하위 CA에서 발행한 모든 인증서는 올바르게 구성된 클라이언트에서 자동으로 유효하지 않은 것으로 간주됩니다. 루트 CA, ipa 또는 다른 하위 CA에서 직접 발급한 기타 모든 인증서는 유효한 상태로 유지됩니다.

이 섹션에서는 **Apache** 웹 서버의 예제를 사용하여 인증서 서브 세트만 신뢰하도록 애플리케이션을 제한하는 방법을 설명합니다. IdM 클라이언트에서 실행 중인 웹 서버가 **webserver-ca IdM** 하위 CA에서 발급한 인증서를 사용하도록 제한하고, 사용자가 **webclient-ca IdM** 하위 명령에서 발급한 사용자 인증서를 사용하여 웹 서버에 인증해야 하는 경우 이 섹션을 완료합니다.

수행하는데 필요한 단계는 다음과 같습니다.

1. **IdM 하위 CA 생성**
2. **IdM WebUI에서 하위 CA 인증서 다운로드**
3. 사용자, 서비스 및 CA 및 사용된 인증서 프로필의 올바른 조합을 지정하는 **CA ACL**을 생성합니다.
4. **IdM 하위 CA에서 IdM 클라이언트에서 실행 중인 웹 서비스에 대한 인증서 요청**
5. 단일 인스턴스 **Apache HTTP Server** 설정

6.

[Apache HTTP 서버에 TLS 암호화 추가](#)

7.

[Apache HTTP Server에서 지원되는 TLS 프로토콜 버전 설정](#)

8.

[Apache HTTP Server에서 지원되는 암호 설정](#)

9.

[웹 서버에서 TLS 클라이언트 인증서 인증 구성](#)

10.

[IdM 하위 CA에서 사용자의 인증서를 요청하고 클라이언트로 내보냅니다.](#)

11.

[브라우저로 사용자 인증서를 가져와서 하위 CA 인증서를 신뢰하도록 브라우저를 구성합니다.](#)

### 17.1. 경량 하위 CA 관리

이 섹션에서는 경량의 하위 인증 기관(sub-CA)을 관리하는 방법을 설명합니다. 생성하는 모든 하위 CA는 인증서 시스템의 기본 CA인 ipa CA로 분류됩니다. 하위 CA를 비활성화하고 삭제할 수도 있습니다.

#### 참고

- 하위 CA를 삭제하면 해당 하위 CA에 대한 해지 검사는 더 이상 작동하지 않습니다. 향후 만료 시간이 아닌 해당 하위 CA에서 발행한 인증서가 더 이상 없는 경우에만 하위 CA를 삭제합니다.
- 해당 하위 CA에서 발급한 만료되지 않은 인증서가 있는 동안 하위 CA만 비활성화해야 합니다. 하위 CA에서 발급한 모든 인증서가 만료된 경우 해당 하위 CA를 삭제할 수 있습니다.
- IdM CA를 비활성화하거나 삭제할 수 없습니다.

하위 CA 관리에 대한 자세한 내용은 다음을 참조하십시오.

- [IdM 웹 UI에서 하위 CA 생성](#)
- [IdM 웹 UI에서 하위 CA 삭제](#)
- [IdM CLI에서 하위 명령 생성](#)
- [IdM CLI에서 하위 명령 비활성화](#)
- [IdM CLI에서 하위 명령 삭제](#)

#### 17.1.1. IdM 웹 UI에서 하위 CA 생성

이 절차에서는 IdM WebUI를 사용하여 **webserver-ca** 및 **webclient-ca** 라는 새 하위 CA를 생성하는 방법을 설명합니다.

##### 사전 요구 사항

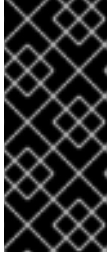
- 관리자의 자격 증명을 확보했는지 확인합니다.

##### 절차

1. **Authentication (인증) 메뉴에서 Certificates (인증서)를 클릭합니다.**
2. 인증 기관을 선택하고 추가를 클릭합니다.
3. **webserver-ca 하위 CA의 이름을 입력합니다. 주체 DN(예: CN=WEBSERVER,O=IDM.EXAMPLE.COM)을 주체 DN 필드에 입력합니다. 주체 DN은 IdM CA 인프라에서 고유해야 합니다.**
4. **webclient-ca 하위 CA의 이름을 입력합니다. Subject DN CN=WEBCIENT,O=IDM.EXAMPLE.COM 을 Subject DN 필드에 입력합니다.**
5. 명령줄 인터페이스에서 **ipa-certupdate** 명령을 실행하여 **webserver-ca** 및 **webclient-ca** 하

위 CA 인증서에 대한 **certmonger** 추적 요청을 생성합니다.

```
[root@ipaserver ~]# ipa-certupdate
```



#### 중요

하위 CA를 생성한 후 **ipa-certupdate** 명령을 실행하는 것을 잊어버린 경우 **end-entity** 인증서가 만료되지 않은 경우에도 하위 CA에서 발급한 종료 인증서가 유효하지 않은 것으로 간주됩니다.

#### 검증

- 

새 하위 CA의 서명 인증서가 IdM 데이터베이스에 추가되었는지 확인합니다.

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L
```

| Certificate Nickname                                           | Trust Attributes   |
|----------------------------------------------------------------|--------------------|
|                                                                | SSL,S/MIME,JAR/XPI |
| caSigningCert cert-pki-ca                                      | CTu,Cu,Cu          |
| Server-Cert cert-pki-ca                                        | u,u,u              |
| auditSigningCert cert-pki-ca                                   | u,u,Pu             |
| caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc | u,u,u              |
| ocspSigningCert cert-pki-ca                                    | u,u,u              |
| subsystemCert cert-pki-ca                                      | u,u,u              |



#### 참고

새 하위 CA 인증서는 인증서 시스템 인스턴스가 설치된 모든 복제본으로 자동으로 전송됩니다.

### 17.1.2. IdM 웹 UI에서 하위 CA 삭제

이 절차에서는 IdM WebUI에서 경량의 하위 캐시를 삭제하는 방법을 설명합니다.

## 참고

- 하위 **CA**를 삭제하면 해당 하위 **CA**에 대한 해지 검사는 더 이상 작동하지 않습니다. 향후 만료 시간이 아닌 해당 하위 **CA**에서 발행한 인증서가 더 이상 없는 경우에만 하위 **CA**를 삭제합니다.
- 해당 하위 **CA**에서 발급한 만료되지 않은 인증서가 있는 동안 하위 **CA**만 비활성화해야 합니다. 하위 **CA**에서 발급한 모든 인증서가 만료된 경우 해당 하위 **CA**를 삭제할 수 있습니다.
- **IdM CA**를 비활성화하거나 삭제할 수 없습니다.

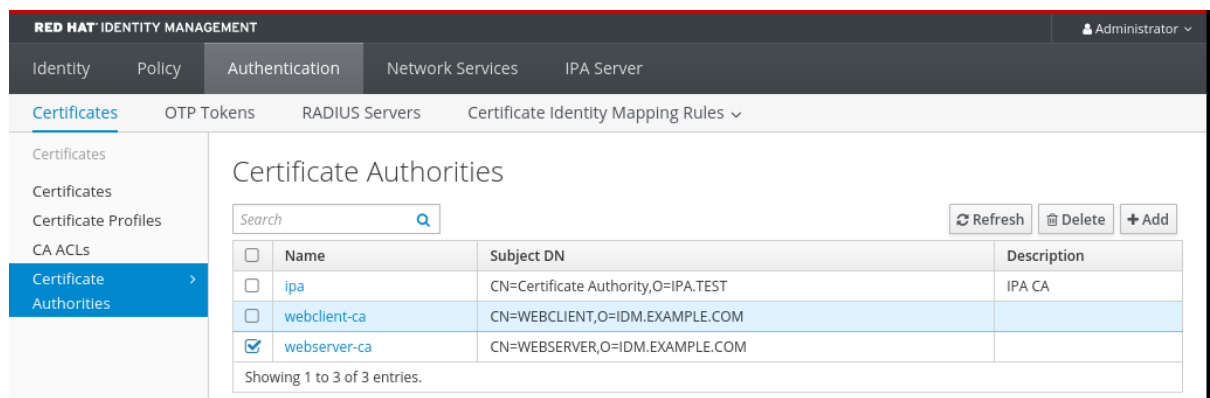
## 사전 요구 사항

- 관리자의 자격 증명을 확보했는지 확인합니다.
- **IdM CLI**에서 하위 **CA**를 비활성화했습니다. **IdM CLI에서 하위 CA 비활성화**를 참조하십시오.

## 절차

1. **IdM 웹 UI**에서 **Authentication (인증)** 탭을 열고 **Certificates (인증서)** 하위 탭을 선택합니다.
2. 인증 기관을 선택합니다.
3. 제거할 하위 **CA**를 선택하고 삭제 버튼을 클릭합니다.

그림 17.1. IdM 웹 UI에서 하위 명령 삭제



4.

**Delete** 를 클릭하여 확인합니다.하위 **CA**는 인증 기관 목록에서 제거됩니다.**17.1.3. IdM CLI에서 하위 명령 생성**

이 절차에서는 **IdM CLI**를 사용하여 **webserver-ca** 및 **webclient-ca** 라는 새 하위 **CA**를 생성하는 방법을 설명합니다.

## 사전 요구 사항

- 관리자의 자격 증명을 확보했는지 확인합니다.
- **CA** 서버인 **IdM** 서버에 로그인되어 있는지 확인합니다.

## 절차

1.

**ipa ca-add** 명령을 입력하고 **webserver-ca** 하위 **CA**의 이름과 제목 **Distinguished Name (DN)**을 지정합니다.

```
[root@ipaserver ~]# ipa ca-add webserver-ca --
subject="CN=WEBSERVER,O=IDM.EXAMPLE.COM"

Created CA "webserver-ca"

Name: webserver-ca
Authority ID: ba83f324-5e50-4114-b109-acca05d6f1dc
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

이름

**CA**의 이름입니다.

기관 ID

**CA**의 개별 **ID**를 자동으로 생성합니다.제목 **DN**

**Distinguished Name (DN)**입니다. 주체 DN은 **IdM CA** 인프라에서 고유해야 합니다.

## 발급자 DN

하위 **CA** 인증서를 발급한 상위 **CA**입니다. 모든 하위 **CA**는 **IdM** 루트 **CA**의 하위 항목으로 생성됩니다.

2.

웹 클라이언트에 인증서를 발급하기 위해 **webclient-ca** 하위 **CA**를 생성합니다.

```
[root@ipaserver ~]# ipa ca-add webclient-ca --
subject="CN=WEBCLIENT,O=IDM.EXAMPLE.COM"

Created CA "webclient-ca"

Name: webclient-ca
Authority ID: 8a479f3a-0454-4a4d-8ade-fd3b5a54ab2e
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

3.

**ipa-certupdate** 명령을 실행하여 **webserver-ca** 및 **webclient-ca** 하위 **CA** 인증서에 대한 **certmonger** 추적 요청을 생성합니다.

```
[root@ipaserver ~]# ipa-certupdate
```



### 중요

하위 **CA**를 생성하고 하위 **CA** 인증서가 만료된 후 **ipa-certupdate** 명령을 실행하는 것을 잊어버린 경우 엔드 고유 인증서가 만료되지 않은 경우에도 해당 하위 **CA**에서 발급한 엔드 엔티티 인증서가 유효하지 않은 것으로 간주됩니다.

## 검증 단계

•

새 하위 **CA**의 서명 인증서가 **IdM** 데이터베이스에 추가되었는지 확인합니다.

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L
```

| Certificate Nickname         | Trust Attributes   |
|------------------------------|--------------------|
|                              | SSL,S/MIME,JAR/XPI |
| caSigningCert cert-pki-ca    | CTu,Cu,Cu          |
| Server-Cert cert-pki-ca      | u,u,u              |
| auditSigningCert cert-pki-ca | u,u,Pu             |

```
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca u,u,u
subsystemCert cert-pki-ca u,u,u
```



참고

새 하위 **CA** 인증서는 인증서 시스템 인스턴스가 설치된 모든 복제본으로 자동으로 전송됩니다.

#### 17.1.4. IdM CLI에서 하위 명령 비활성화

다음 절차에서는 **IdM CLI**에서 하위 **CA**를 비활성화하는 방법을 설명합니다. 하위 **CA**에서 발급한 만료되지 않은 인증서가 여전히 있는 경우 삭제할 수 없지만 비활성화할 수 있습니다. 하위 **CA**를 삭제하면 해당 하위 **CA**에 대한 해지 검사는 더 이상 작동하지 않습니다.

##### 사전 요구 사항

- 관리자의 자격 증명을 확보했는지 확인합니다.

##### 절차

1. **ipa ca-find** 명령을 실행하여 삭제 중인 하위 **CA**의 이름을 확인합니다.

```
[root@ipaserver ~]# ipa ca-find

3 CAs matched

Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Name: webserver-ca
Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST

Number of entries returned 3

```

2.

**ipa ca-disable** 명령을 실행하여 하위 **CA**(이 예에서는 **webserver-ca**)를 비활성화합니다.

```
ipa ca-disable webserver-ca
```

```

Disabled CA "webserver-ca"

```

### 17.1.5. IdM CLI에서 하위 명령 삭제

이 절차에서는 **IdM CLI**에서 경량 하위 **CA**를 삭제하는 방법을 설명합니다.

#### 참고

- 하위 **CA**를 삭제하면 해당 하위 **CA**에 대한 해지 검사는 더 이상 작동하지 않습니다. 향후 만료 시간이 아닌 해당 하위 **CA**에서 발행한 인증서가 더 이상 없는 경우에만 하위 **CA**를 삭제합니다.
- 해당 하위 **CA**에서 발급한 만료되지 않은 인증서가 있는 동안 하위 **CA**만 비활성화해야 합니다. 하위 **CA**에서 발급한 모든 인증서가 만료된 경우 해당 하위 **CA**를 삭제할 수 있습니다.
- **IdM CA**를 비활성화하거나 삭제할 수 없습니다.

#### 사전 요구 사항

- 관리자의 자격 증명을 확보했는지 확인합니다.

#### 절차

1.

하위 **CA** 및 **CA** 목록을 표시하려면 **ipa ca-find** 명령을 실행합니다.

```
ipa ca-find
```

```

3 CAs matched

```

```
Name: ipa
```

```
Description: IPA CA
```

```
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
```

```
Subject DN: CN=Certificate Authority,O=IPA.TEST
```

```
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webserver-ca
Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```

Number of entries returned 3

```

2.

**ipa ca-disable** 명령을 실행하여 하위 **CA**(이 예에서는 **webserver-ca**)를 비활성화합니다.

```
ipa ca-disable webserver-ca
```

```

Disabled CA "webserver-ca"

```

3.

이 예에서 **sub-CA**를 삭제합니다. **webserver-ca**:

```
ipa ca-del webserver-ca
```

```

Deleted CA "webserver-ca"

```

## 검증

•

**ipa ca-find** 를 실행하여 **CA** 및 하위 **CA** 목록을 표시합니다. **webserver-ca** 가 더 이상 목록에 없습니다.

```
ipa ca-find
```

```

2 CAs matched

```

```
Name: ipa
Description: IPA CA
Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
Subject DN: CN=Certificate Authority,O=IPA.TEST
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

```
Name: webclient-ca
Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
Subject DN: CN=WEBCIENT,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IPA.TEST
```

-----  
**Number of entries returned 2**  
 -----

## 17.2. IDM WEBUI에서 하위 CA 인증서 다운로드

### 사전 요구 사항

- **IdM 관리자의 인증 정보를 가져왔는지 확인합니다.**

### 절차

1. 인증 메뉴에서 인증서 > 인증서 를 클릭합니다.

그림 17.2. 인증서 목록의 하위 CA 인증서

|                          |           |                                |     |       |
|--------------------------|-----------|--------------------------------|-----|-------|
| <input type="checkbox"/> | 268173326 | CN=WEBSERVER,O=IDM.EXAMPLE.COM | ipa | VALID |
| <input type="checkbox"/> | 268238849 | CN=idm_user,O=IDM.EXAMPLE.COM  | ipa | VALID |

2. 하위 CA 인증서의 일련 번호를 클릭하여 인증서 정보 페이지를 엽니다.
3. 인증서 정보 페이지에서 작업 > 다운로드를 클릭합니다.
4. CLI에서 하위 CA 인증서를 `/etc/pki/tls/private/` 디렉터리로 이동합니다.

```
mv path/to/the/downloaded/certificate /etc/pki/tls/private/sub-ca.crt
```

## 17.3. 웹 서버 및 클라이언트 인증에 대한 CA ACL 생성

**CA ACL**(인증 기관 액세스 제어 목록) 규칙은 사용자, 서비스 또는 호스트에 인증서를 발급하는 데 사용할 수 있는 프로필을 정의합니다. 프로필, 주체 및 그룹을 연결함으로써 **CA ACL**은 보안 주체 또는 그룹을 통해 특정 프로필을 사용하여 인증서를 요청할 수 있습니다.

예를 들어, **CA ACL**을 사용하면, 관리자는 런던에 위치한 사무실에서 작업 중인 직원들에게 런던 관련 그룹의 멤버인 사용자에게만 사용할 수 있는 프로필의 사용을 제한할 수 있습니다.

### 17.3.1. IdM CLI에서 CA ACL 보기

IdM 배포에서 사용할 수 있는 **CA ACL**(인증 기관 액세스 제어 목록) 목록과 특정 **CA ACL**의 세부 정보를 보려면 이 섹션을 완료합니다.

## 절차

1. IdM 환경의 모든 **CA ACL**을 보려면 **ipa caacl-find** 명령을 입력합니다.

```
$ ipa caacl-find

1 CA ACL matched

ACL name: hosts_services_calPAserviceCert
Enabled: TRUE
```

2. **CA ACL**의 세부 정보를 보려면 **ipa caacl-show** 명령을 입력하고 **CA ACL** 이름을 지정합니다. 예를 들어 **hosts\_services\_calPAserviceCert** **CA ACL**의 세부 정보를 보려면 다음을 입력합니다.

```
$ ipa caacl-show hosts_services_calPAserviceCert
ACL name: hosts_services_calPAserviceCert
Enabled: TRUE
Host category: all
Service category: all
CAs: ipa
Profiles: calPAserviceCert
Users: admin
```

### 17.3.2. webserver-ca에서 발행한 인증서를 사용하여 웹 클라이언트에 인증하는 웹 서버에 대한 **CA ACL** 생성

이 섹션에서는 **HTTP/my\_company.idm.example.com@IDM.EXAMPLE.COM** 서비스에 대한 인증서를 요청할 때 시스템 관리자가 **webserver-ca** 하위 **CA** 및 **calPAserviceCert** 프로필을 사용해야 하는 **CA ACL**을 생성하는 방법을 설명합니다. 사용자가 다른 하위 **CA** 또는 다른 프로필의 인증서를 요청하면 요청이 실패합니다. 유일한 예외는 활성화된 또 다른 일치하는 **CA ACL**이 있는 경우입니다. 사용 가능한 **CA ACL**을 보려면 **IdM CLI**에서 **CA ACL** 보기를 참조하십시오.

## 사전 요구 사항

- **HTTP/my\_company.idm.example.com@IDM.EXAMPLE.COM** 서비스가 IdM의 일부인지 확인합니다.
- IdM 관리자의 인증 정보가 있는지 확인합니다.

## 절차

1.

**ipa caacl** 명령을 사용하여 **CA ACL**을 생성하고 이름을 지정합니다.

```
$ ipa caacl-add TLS_web_server_authentication

Added CA ACL "TLS_web_server_authentication"

ACL name: TLS_web_server_authentication
Enabled: TRUE
```

2.

**ipa caacl-mod** 명령을 사용하여 **CA ACL**을 수정하여 **CA ACL** 설명을 지정합니다.

```
$ ipa caacl-mod TLS_web_server_authentication --desc="CAACL for web servers
authenticating to web clients using certificates issued by webserver-ca"

Modified CA ACL "TLS_web_server_authentication"

ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
```

3.

**CA ACL**에 **webserver-ca** 하위 **CA**를 추가합니다.

```
$ ipa caacl-add-ca TLS_web_server_authentication --ca=webserver-ca
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca

Number of members added 1

```

4.

**ipa caacl-add-service** 를 사용하여 보안 주체가 인증서를 요청할 수 있는 서비스를 지정합니다.

```
$ ipa caacl-add-service TLS_web_server_authentication --
service=HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
```

```

Number of members added 1

```

5.

`ipa caacl-add-profile` 명령을 사용하여 요청된 인증서의 인증서 프로필을 지정합니다.

```
$ ipa caacl-add-profile TLS_web_server_authentication --
certprofiles=calPAserviceCert
 ACL name: TLS_web_server_authentication
 Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
 Enabled: TRUE
 CAs: webserver-ca
 Profiles: calPAserviceCert
 Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM

Number of members added 1

```

새로 생성된 **CA ACL**을 즉시 사용할 수 있습니다. 기본적으로 생성된 후 활성화됩니다.



#### 참고

**CA ACL**의 핵심은 특정 보안 주체 또는 그룹에서 들어오는 요청에 대해 허용되는 **CA** 및 프로필 조합을 지정하는 것입니다. **CA ACL**은 인증서 검증 또는 신뢰에 영향을 미치지 않습니다. 인증서 사용 방식에는 영향을 미치지 않습니다.

### 17.3.3. webclient-ca에서 발행한 인증서를 사용하여 웹 서버에 인증하는 사용자 웹 브라우저의 **CA ACL** 생성

이 섹션에서는 시스템 관리자가 인증서를 요청할 때 **webclient-ca** 하위 **CA** 및 **IECUserRoles** 프로필을 사용해야 하는 **CA ACL**을 생성하는 방법을 설명합니다. 사용자가 다른 하위 **CA** 또는 다른 프로필의 인증서를 요청하면 요청이 실패합니다. 유일한 예외는 활성화된 또 다른 일치하는 **CA ACL**이 있는 경우입니다. 사용 가능한 **CA ACL**을 보려면 **IdM CLI**에서 **CA ACL** 보기를 참조하십시오.

#### 사전 요구 사항

- **IdM** 관리자의 인증 정보를 가져왔는지 확인합니다.

#### 절차

1.

`ipa caacl` 명령을 사용하여 **CA ACL**을 생성하고 이름을 지정합니다.

```
$ ipa caacl-add TLS_web_client_authentication

Added CA ACL "TLS_web_client_authentication"

ACL name: TLS_web_client_authentication
Enabled: TRUE
```

2.

**ipa caacl-mod** 명령을 사용하여 **CA ACL**을 수정하여 **CA ACL** 설명을 지정합니다.

```
$ ipa caacl-mod TLS_web_client_authentication --desc="CAACL for user web
browsers authenticating to web servers using certificates issued by webclient-ca"

Modified CA ACL "TLS_web_client_authentication"

ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
```

3.

**CA ACL**에 **webclient-ca** 하위 **CA**를 추가합니다.

```
$ ipa caacl-add-ca TLS_web_client_authentication --ca=webclient-ca
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
CAs: webclient-ca

Number of members added 1

```

4.

**ipa caacl-add-profile** 명령을 사용하여 요청된 인증서의 인증서 프로필을 지정합니다.

```
$ ipa caacl-add-profile TLS_web_client_authentication --certprofiles=IECUserRoles
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
CAs: webclient-ca
Profiles: IECUserRoles

Number of members added 1

```

5.

**ipa caacl-mod** 명령을 사용하여 **CA ACL**을 수정하여 **CA ACL**이 모든 **IdM** 사용자에게 적용 되도록 지정합니다.

```
$ ipa caacl-mod TLS_web_client_authentication --usercat=all
```

```

Modified CA ACL "TLS_web_client_authentication"

```

```
ACL name: TLS_web_client_authentication
Description: CAACL for user web browsers authenticating to web servers using
certificates issued by webclient-ca
Enabled: TRUE
User category: all
CAs: webclient-ca
Profiles: IECUserRoles
```

새로 생성된 **CA ACL**을 즉시 사용할 수 있습니다. 기본적으로 생성된 후 활성화됩니다.



#### 참고

**CA ACL**의 핵심은 특정 보안 주체 또는 그룹에서 들어오는 요청에 대해 허용되는 **CA** 및 프로필 조합을 지정하는 것입니다. **CA ACL**은 인증서 검증 또는 신뢰에 영향을 미치지 않습니다. 인증서 사용 방식에는 영향을 미치지 않습니다.

## 17.4. CERTMONGER를 사용하여 서비스에 대한 IDM 인증서 가져오기

**IdM** 클라이언트에서 실행되는 브라우저와 웹 서비스 간 통신이 안전하고 암호화되도록 하려면 **TLS** 인증서를 사용합니다. 웹 브라우저가 **webserver-ca** 하위 **CA**에서 발급한 인증서를 신뢰하도록 제한하고 기타 **IdM** 하위 **CA**는 없는 경우 **webserver-ca** 하위 **CA**에서 웹 서비스의 **TLS** 인증서를 가져옵니다.

이 섹션에서는 **certmonger**를 사용하여 서비스에 대한 **IdM** 인증서 (**HTTP/my\_company.idm.example.com@IDM.EXAMPLE.COM**)를 **IdM** 클라이언트에서 실행하는 방법을 설명합니다.

**certmonger**를 사용하여 인증서를 자동으로 요청한다는 것은 **certmonger**가 갱신이 발생할 때 인증서를 관리하고 갱신한다는 것을 의미합니다.

**certmonger**가 서비스 인증서를 요청할 때 발생하는 상황을 시각적으로 표현하려면 **17.5절. “서비스 인증서를 요청하는 certmonger의 통신 흐름”**을 참조하십시오.

#### 사전 요구 사항

- 웹 서버는 **IdM** 클라이언트로 등록되어 있습니다.

- 프로시저를 실행 중인 **IdM** 클라이언트에 대한 루트 액세스 권한이 있습니다.
- 인증서를 요청하는 서비스는 **IdM**에 사전 존재할 필요가 없습니다.

## 절차

1.

**HTTP** 서비스가 실행 중인 **my\_company.idm.example.com** **IdM** 클라이언트에서 **HTTP** **/my\_company.idm.example.com@IDM.EXAMPLE.COM** 주체에 해당하는 서비스에 대한 인증서를 요청하며 이를 지정합니다.

- 인증서는 로컬 **/etc/pki/tls/certs/httpd.pem** 파일에 저장됩니다.
- 개인 키는 로컬 **/etc/pki/tls/private/httpd.key** 파일에 저장해야 합니다.
- **webserver-ca** 하위 **CA**는 발급 기관입니다.
- **SubjectAltName**에 대한 **extensionRequest**가 **my\_company.idm.example.com**의 **DNS** 이름을 사용하여 서명 요청에 추가됩니다.

```
ipa-getcert request -K HTTP/my_company.idm.example.com -k
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -X webserver-ca -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

위의 명령에서:

- **ipa-getcert request** 명령은 인증서를 **IdM CA**에서 가져올 수 있도록 지정합니다. **ipa-getcert request** 명령은 **getcert request -c IPA**의 바로 가기입니다.
- **g** 옵션은 아직 없는 경우 생성할 키 크기를 지정합니다.
- **d** 옵션은 요청에 추가할 **SubjectAltName DNS** 값을 지정합니다.

- **X** 옵션은 인증서 발행자가 **ipa** 가 아닌 **webserver-ca** 여야 함을 지정합니다.
- **-C** 옵션은 인증서를 가져온 후 **httpd** 서비스를 다시 시작하도록 **certmonger** 에 지시합니다.
- 특정 프로필로 인증서를 발행하도록 지정하려면 **-T** 옵션을 사용합니다.

2.

필요한 경우 요청 상태를 확인하려면 다음을 수행합니다.

```
ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
 status: MONITORING
 stuck: no
 key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
 certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
 CA: IPA
 issuer: CN=WEBSERVER,O=IDM.EXAMPLE.COM
```

[...]

출력은 요청이 **MONITORING** 상태에 있음을 보여줍니다. 즉, 인증서를 가져온 것입니다. 키 쌍과 인증서의 위치는 요청된 위치입니다.

## 17.5. 서비스 인증서를 요청하는 CERTMONGER의 통신 흐름

이 섹션의 다이어그램은 **certmonger** 가 **IdM(Identity Management)** 인증 기관(**CA**) 서버에서 서비스 인증서를 요청할 때 발생하는 단계를 보여줍니다. 시퀀스는 다음 다이어그램으로 구성됩니다.

- 암호화되지 않은 통신
- 서비스 인증서 요청
- 서비스 인증서를 발급하는 **IdM CA**
- 서비스 인증서 적용

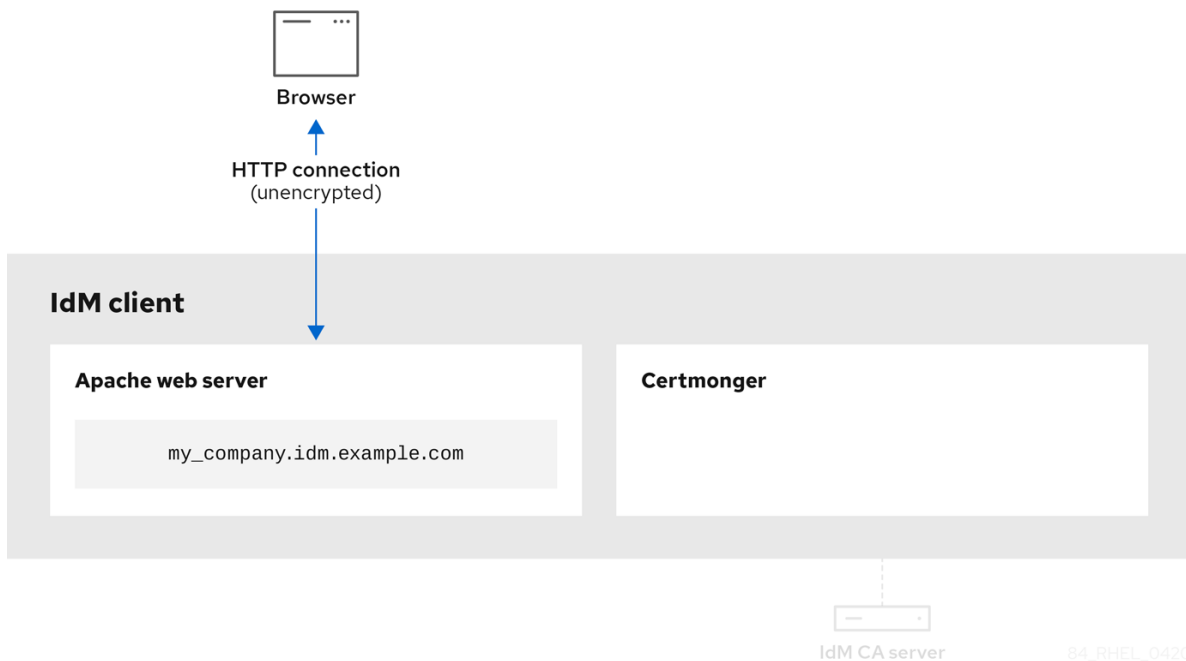
•

**certmonger**에서 이전 인증서가 만료되는 경우 새 인증서를 요청합니다.

다이어그램에서 **webserver-ca** 하위 **CA**는 일반 **IdM CA** 서버로 표시됩니다.

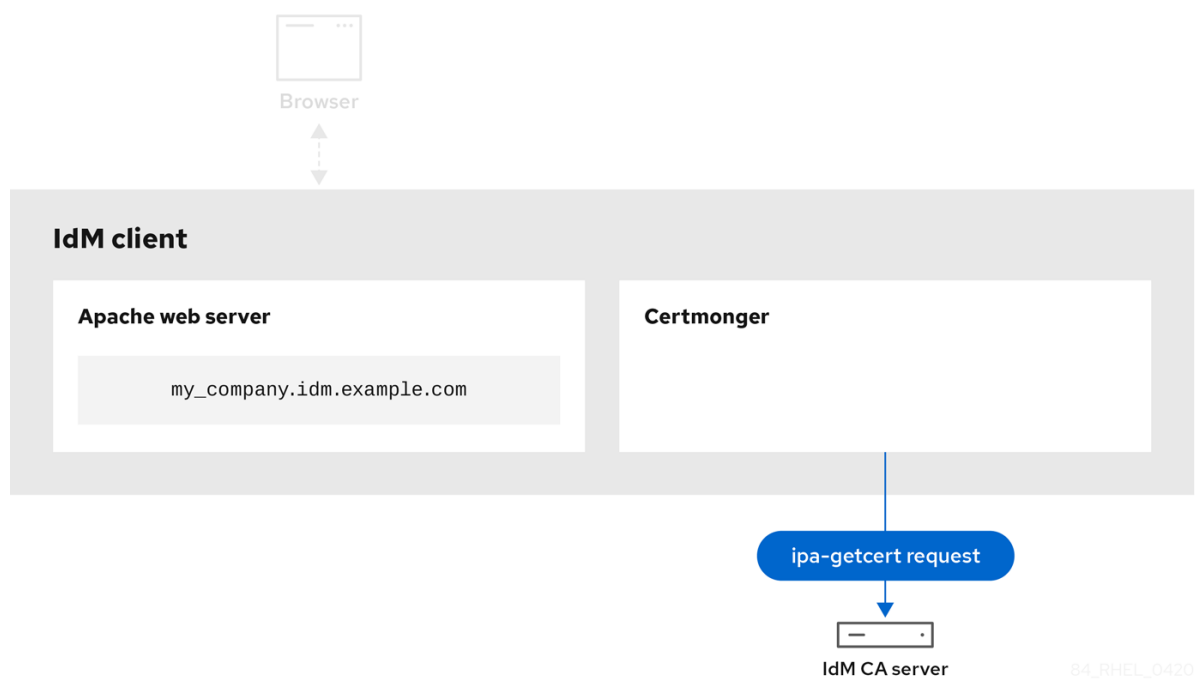
**암호화되지 않은 통신**에서는 초기 상황을 보여줍니다. **HTTPS** 인증서가 없으면 웹 서버와 브라우저 간 통신이 암호화되지 않습니다.

그림 17.3. 암호화되지 않은 통신



**서비스 인증서를 요청하는 certmonger**는 **certmonger**를 사용하여 **Apache** 웹 서버에 대한 **HTTPS** 인증서를 수동으로 요청하는 시스템 관리자에게 보여줍니다. 웹 서버 인증서를 요청할 때 **certmonger**는 **CA**와 직접 통신하지 않습니다. **IdM**을 통해 프록시합니다.

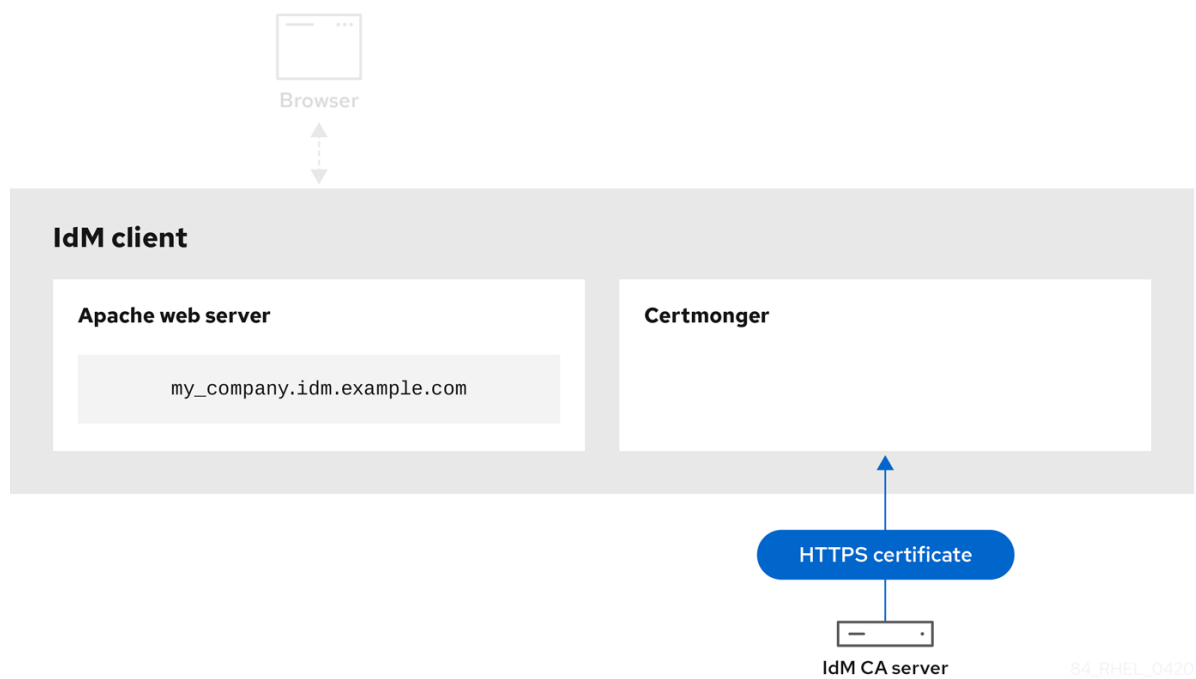
그림 17.4. 서비스 인증서 요청



84\_RHEL\_0420

서비스 인증서를 발행하는 IdM CA는 웹 서버의 HTTPS 인증서를 발행하는 IdM CA를 보여줍니다.

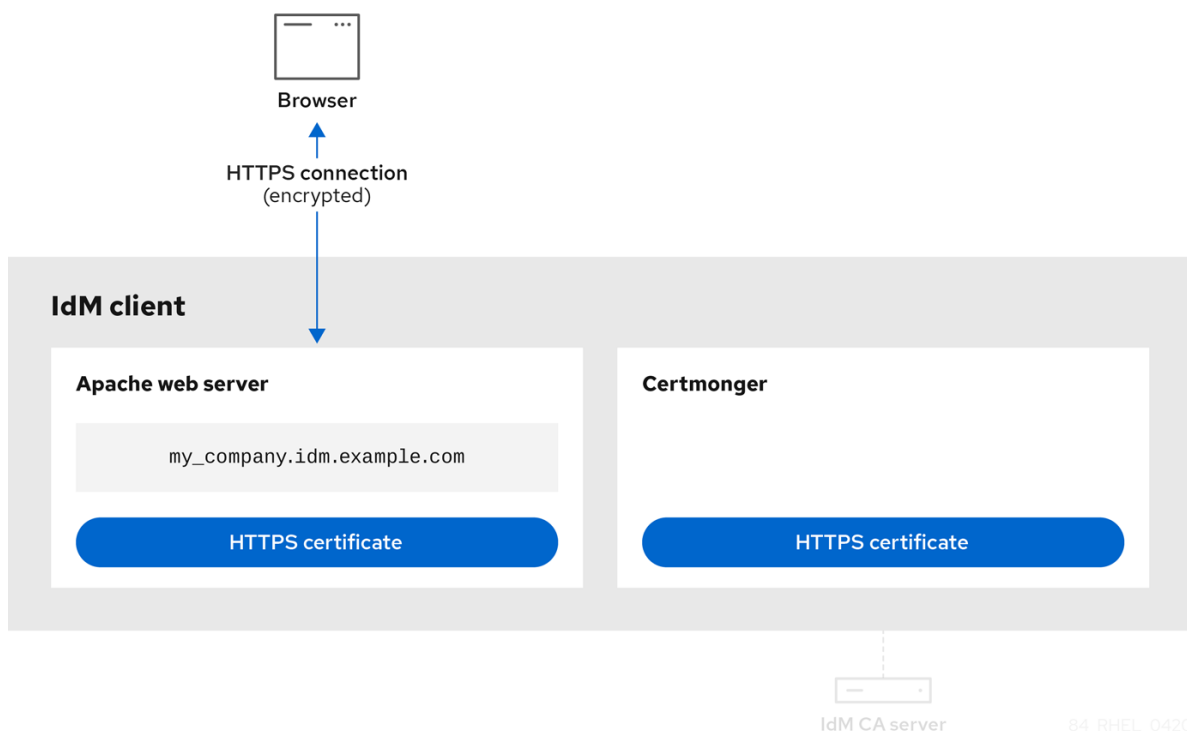
그림 17.5. 서비스 인증서를 발급하는 IdM CA



84\_RHEL\_0420

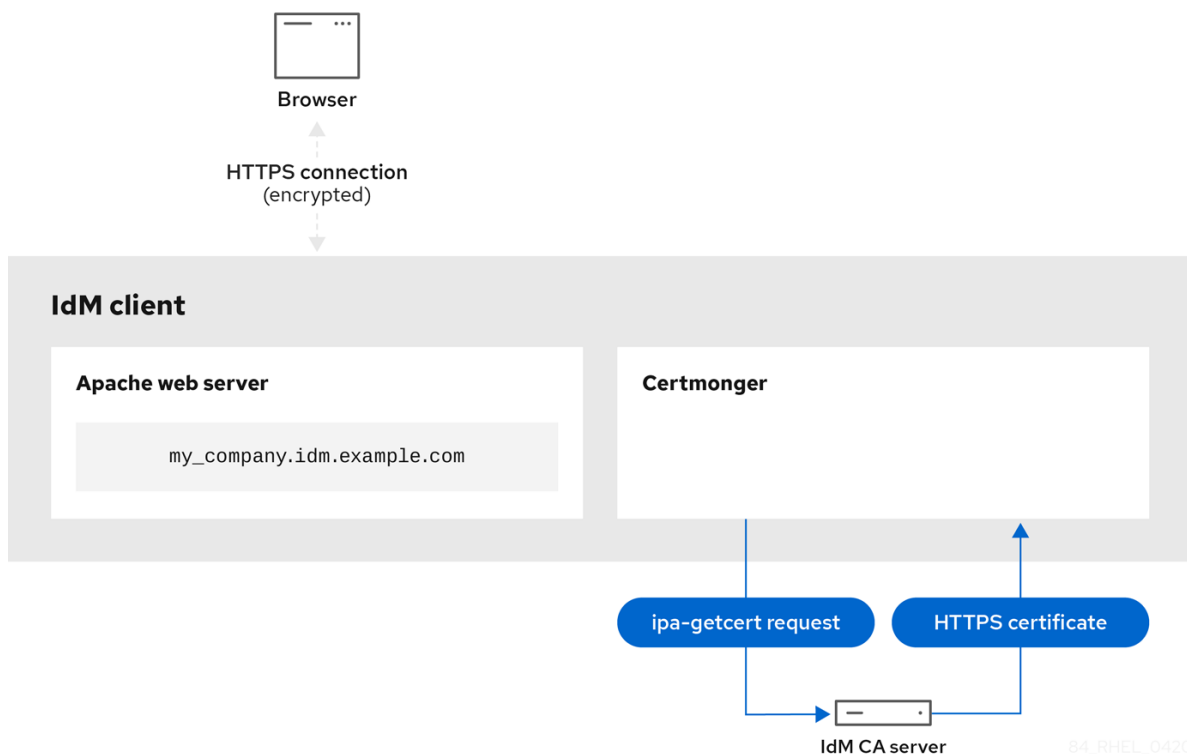
서비스 인증서를 적용하는 **certmonger**는 **IdM** 클라이언트의 적절한 위치에 **HTTPS** 인증서를 배치하고 이를 수행하라는 지침이 있는 **certmonger**를 사용하여 **httpd** 서비스를 다시 시작합니다. **Apache** 서버에서는 **HTTPS** 인증서를 사용하여 자체와 브라우저 간 트래픽을 암호화합니다.

그림 17.6. 서비스 인증서 적용



만료일이 다가오는 경우 **certmonger**에서 새 인증서를 요청하는 **certmonger**가 인증서 만료 전에 **IdM CA**에서 서비스 인증서 갱신을 자동으로 요청합니다. **IdM CA**는 새 인증서를 발행합니다.

그림 17.7. **certmonger**에서 이전 인증서가 만료되는 경우 새 인증서를 요청합니다.



## 17.6. 단일 인스턴스 **APACHE HTTP SERVER** 설정

이 섹션에서는 정적 **HTML** 콘텐츠를 제공하기 위해 단일 인스턴스 **Apache HTTP Server**를 설정하는 방법을 설명합니다.

웹 서버가 서버와 연결된 모든 도메인에 동일한 콘텐츠를 제공해야 하는 경우 이 섹션의 절차를 따릅니다. 도메인마다 다른 콘텐츠를 제공하려면 이름 기반 가상 호스트를 설정합니다. 자세한 내용은 [Apache 이름 기반 가상 호스트 구성](#)을 참조하십시오.

### 절차

1. **httpd** 패키지를 설치합니다.

```
dnf install httpd
```

2. 로컬 방화벽에서 **TCP 포트 80**을 엽니다.

```
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --reload
```

3.

**httpd** 서비스를 활성화하고 시작합니다.

```
systemctl enable --now httpd
```

4.

선택 사항: **/var/www/html/** 디렉터리에 **HTML** 파일을 추가합니다.



#### 참고

콘텐츠를 **/var/www/html/**에 추가할 때 기본적으로 **httpd**가 실행되는 사용자가 파일 및 디렉터리를 읽을 수 있어야 합니다. 콘텐츠 소유자는 **root** 사용자 및 **root** 사용자 그룹 또는 다른 사용자 또는 관리자가 선택한 그룹일 수 있습니다. 콘텐츠 소유자가 **root** 사용자 및 **root** 사용자 그룹인 경우 다른 사용자가 파일을 읽을 수 있어야 합니다. 모든 파일 및 디렉터리에 대한 **SELinux** 컨텍스트는 기본적으로 **/var/www** 디렉터리 내의 모든 콘텐츠에 적용되는 **httpd\_sys\_content\_t**이어야 합니다.

#### 검증 단계

•

웹 브라우저에서 **http://my\_company.idm.example.com/** 또는 **http://server\_IP/**에 연결합니다.

**/var/www/html/** 디렉터리가 비어 있거나 **index.html** 또는 **index.htm** 파일이 포함되어 있지 않은 경우 **Apache**에서 **Red Hat Enterprise Linux Test Page**를 표시합니다. **/var/www/html/**에 다른 이름이 있는 **HTML** 파일이 포함된 경우 해당 파일에 **URL**을 입력하여 로드할 수 있습니다 (예: **http://server\_IP/example.html** 또는 **http://my\_company.idm.example.com/example.html**).

#### 추가 리소스

•

**Apache** 설명서를 참조하십시오. [Apache HTTP Server 수동 설치](#)를 참조하십시오.

•

**httpd.service(8)** 매뉴얼 페이지를 참조하십시오.

### 17.7. APACHE HTTP SERVER에 TLS 암호화 추가

이 섹션에서는 **idm.example.com** 도메인에 대해 **my\_company.idm.example.com** **Apache HTTP** 서버에서 **TLS** 암호화를 활성화하는 방법을 설명합니다.

## 사전 요구 사항

- **my\_company.idm.example.com** Apache HTTP 서버가 설치되어 실행 중입니다.
- **webserver-ca** 하위 CA에서 TLS 인증서를 가져와서 17.4절. “**certmonger를 사용하여 서비스에 대한 IdM 인증서 가져오기**”에 설명된 대로 **/etc/pki/tls/certs/httpd.pem** 파일에 저장했습니다. 다른 경로를 사용하는 경우 절차의 해당 단계를 조정합니다.
- 해당 개인 키는 **/etc/pki/tls/private/httpd.key** 파일에 저장됩니다. 다른 경로를 사용하는 경우 절차의 해당 단계를 조정합니다.
- **webserver-ca** CA 인증서는 **/etc/pki/tls/private/sub-ca.crt** 파일에 저장됩니다. 다른 경로를 사용하는 경우 절차의 해당 단계를 조정합니다.
- 클라이언트 및 **my\_company.idm.example.com** 웹 서버는 서버의 호스트 이름을 웹 서버의 IP 주소로 확인합니다.

## 절차

1. **mod\_ssl** 패키지를 설치합니다.

```
dnf install mod_ssl
```

2. **/etc/httpd/conf.d/ssl.conf** 파일을 편집하고 다음 설정을 **<VirtualHost \_default\_:443>** 지시문에 추가합니다.

- a. 서버 이름을 설정합니다.

```
ServerName my_company.idm.example.com
```



## 중요

서버 이름은 인증서의 **Common Name** 필드에 설정된 항목과 일치해야 합니다.

b.

선택 사항: 인증서에 **SAN**( 주체 **Alt Names** ) 필드에 추가 호스트 이름이 포함된 경우 이러한 호스트 이름에 **TLS** 암호화도 제공하도록 **mod\_ssl** 을 구성할 수 있습니다. 이를 구성하려면 해당 이름으로 **ServerAliases** 매개변수를 추가합니다.

```
ServerAlias www.my_company.idm.example.com
server.my_company.idm.example.com
```

c.

개인 키, 서버 인증서 및 **CA** 인증서에 대한 경로를 설정합니다.

```
SSLCertificateKeyFile "/etc/pki/tls/private/httpd.key"
SSLCertificateFile "/etc/pki/tls/certs/httpd.pem"
SSLCACertificateFile "/etc/pki/tls/certs/ca.crt"
```

3.

보안상의 이유로 **root** 사용자만 개인 키 파일에 액세스할 수 있도록 구성합니다.

```
chown root:root /etc/pki/tls/private/httpd.key
chmod 600 //etc/pki/tls/private/httpd.key
```



#### 주의

권한이 없는 사용자가 개인 키에 액세스한 경우 인증서를 취소하고 새 개인 키를 만들고 새 인증서를 요청합니다. 그렇지 않으면 **TLS** 연결이 더 이상 안전하지 않습니다.

4.

로컬 방화벽에서 **443** 포트를 엽니다.

```
firewall-cmd --permanent --add-port=443/tcp
firewall-cmd --reload
```

5.

**httpd** 서비스를 다시 시작합니다.

```
systemctl restart httpd
```



## 참고

개인 키 파일을 암호로 보호한 경우 **httpd** 서비스가 시작될 때마다 이 암호를 입력해야 합니다.

- 브라우저를 사용하여 **https://my\_company.idm.example.com** 에 연결합니다.

## 추가 리소스

- **Apache** 설명서의 **SSL/TLS** 암호화 설명서를 참조하십시오.
- [Apache HTTP Server 수동 설치](#)를 참조하십시오.
- [RHEL 8에서 TLS의 보안 고려 사항](#)

## 17.8. APACHE HTTP SERVER에서 지원되는 TLS 프로토콜 버전 설정

기본적으로 **RHEL**의 **Apache HTTP Server**는 최신 브라우저와 호환되는 안전한 기본값을 정의하는 시스템 전체 암호화 정책을 사용합니다. 예를 들어 **DEFAULT** 정책은 **apache**에서 **TLSv1.2** 및 **TLSv1.3** 프로토콜 버전만 사용하도록 정의합니다.

이 섹션에서는 **my\_company.idm.example.com** **Apache HTTP Server**에서 지원하는 **TLS** 프로토콜 버전을 수동으로 구성하는 방법을 설명합니다. 환경에서 특정 **TLS** 프로토콜 버전만 활성화해야 하는 경우 절차를 따르십시오. 예를 들면 다음과 같습니다.

- 환경에 해당 클라이언트가 클라이언트가 취약한 **TLS1 (TLSv1.0)** 또는 **TLS1.1** 프로토콜을 사용할 수도 있어야 하는 경우
- **Apache**가 **TLSv1.2** 또는 **TLSv1.3** 프로토콜만 지원하도록 구성하려는 경우

## 사전 요구 사항

- [Apache HTTP 서버에 TLS 암호화 추가에 설명된 대로 my\\_company.idm.example.com 서버에서 TLS 암호화가 활성화됩니다.](#)

## 절차

1. `/etc/httpd/conf/httpd.conf` 파일을 편집하고 다음 설정을 TLS 프로토콜 버전을 설정하려는 `<VirtualHost>` 지시문에 추가합니다. 예를 들어 TLSv1.3 프로토콜만 활성화하려면 다음을 수행합니다.

```
SSLProtocol -All TLSv1.3
```

2. `httpd` 서비스를 다시 시작합니다.

```
systemctl restart httpd
```

## 검증 단계

1. 다음 명령을 사용하여 서버가 TLSv1.3을 지원하는지 확인합니다:

```
openssl s_client -connect example.com:443 -tls1_3
```

2. 다음 명령을 사용하여 서버가 TLSv1.2를 지원하지 않는지 확인합니다.

```
openssl s_client -connect example.com:443 -tls1_2
```

서버가 프로토콜을 지원하지 않으면 명령에서 오류를 반환합니다.

```
140111600609088:error:1409442E:SSL routines:ssl3_read_bytes:tlsv1 alert protocol version:ssl/record/rec_layer_s3.c:1543:SSL alert number 70
```

3. 선택 사항: 다른 TLS 프로토콜 버전에 대해 명령을 반복합니다.

## 추가 리소스

- `update-crypto-policies(8)` 도움말 페이지를 참조하십시오.
- [전체 시스템 암호화 정책 사용](#)을 참조하십시오.
- `SSLProtocol` 매개변수에 대한 자세한 내용은 **Apache** 설명서의 `mod_ssl` 문서를 참조하십시오.

- [Apache HTTP Server 수동 설치](#)를 참조하십시오.

## 17.9. APACHE HTTP SERVER에서 지원되는 암호 설정

기본적으로 **Apache HTTP** 서버는 최근 브라우저와 호환되는 안전한 기본값을 정의하는 시스템 전체 암호화 정책을 사용합니다. 시스템 전체에서 허용하는 암호 목록은 `/etc/crypto-policies/back-ends/openssl.config` 파일을 참조하십시오.

이 섹션에서는 `my_company.idm.example.com` **Apache HTTP** 서버가 지원하는 암호를 수동으로 구성하는 방법을 설명합니다. 환경에 특정 암호가 필요한 경우 절차를 따르십시오.

### 사전 요구 사항

- [Apache HTTP 서버에 TLS 암호화 추가](#)에 설명된 대로 `my_company.idm.example.com` 서버에서 TLS 암호화가 활성화됩니다.

### 절차

1. `/etc/httpd/conf/httpd.conf` 파일을 편집하고 **SSLCipherSuite** 매개변수를 TLS 암호를 설정하려는 `<VirtualHost>` 지시문에 추가합니다.

```
SSLCipherSuite
"EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:!SHA1:!SHA256"
```

이 예제에서는 **EECDH+AESGCM**, **EDH+AESGCM**, **AES256+EECDH** 및 **AES256+EDH** 암호만 활성화하며 **SHA1** 및 **SHA256** 메시지 인증 코드(MAC)를 사용하는 모든 암호를 비활성화합니다.

2. **httpd** 서비스를 다시 시작합니다.

```
systemctl restart httpd
```

### 검증 단계

1. **Apache HTTP Server**에서 지원하는 암호 목록을 표시하려면 다음을 수행합니다.

- a. **nmap** 패키지를 설치합니다.

```
dnf install nmap
```

- b. **nmap** 유틸리티를 사용하여 지원되는 암호를 표시합니다.

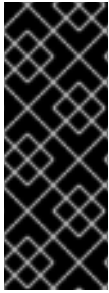
```
nmap --script ssl-enum-ciphers -p 443 example.com
...
PORT STATE SERVICE
443/tcp open https
| ssl-enum-ciphers:
| TLSv1.2:
| ciphers:
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdhe_x25519) - A
| TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh_2048) - A
| TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdhe_x25519) - A
|
|_
```

#### 추가 리소스

- **update-crypto-policies(8)** 도움말 페이지를 참조하십시오.
- [전체 시스템 암호화 정책 사용](#)을 참조하십시오.
- **SSLCipherSuite** 매개 변수에 대한 자세한 내용은 **Apache** 설명서의 **mod\_ssl** 설명서를 참조하십시오.
- [Apache HTTP Server 수동 설치](#)를 참조하십시오.

### 17.10. TLS 클라이언트 인증서 인증 구성

클라이언트 인증서 인증을 사용하면 관리자는 인증서를 사용하여 인증한 사용자만 **my\_company.idm.example.com** 웹 서버의 리소스에 액세스할 수 있습니다. 이 섹션에서는 **/var/www/html/Example/** 디렉터리에 대한 클라이언트 인증서 인증을 구성하는 방법을 설명합니다.



## 중요

**my\_company.idm.example.com** Apache 서버에서 TLS 1.3 프로토콜을 사용하는 경우 특정 클라이언트에 추가 구성이 필요합니다. 예를 들어 Firefox에서 **about:config** 메뉴의 **security.tls.enable\_post\_handshake\_auth** 매개 변수를 **true**로 설정합니다. 자세한 내용은 [Red Hat Enterprise Linux 8의 전송 계층 보안 버전 1.3](#)을 참조하십시오.

## 사전 요구 사항

- [Apache HTTP](#) 서버에 TLS 암호화 추가에 설명된 대로 **my\_company.idm.example.com** 서버에서 TLS 암호화가 활성화됩니다.

## 절차

1. **/etc/httpd/conf/httpd.conf** 파일을 편집하고 다음 설정을 클라이언트 인증을 구성하려는 **<VirtualHost>** 지시문에 추가합니다.

```
<Directory "/var/www/html/Example/">
 SSLVerifyClient require
</Directory>
```

**SSLVerifyClient require**를 설정해야 클라이언트가 **/var/www/html/Example/** 디렉터리의 콘텐츠에 액세스하기 전에 서버가 클라이언트 인증서의 유효성을 검사해야 합니다.

2. **httpd** 서비스를 다시 시작합니다.

```
systemctl restart httpd
```

## 검증 단계

1. **curl** 유틸리티를 사용하여 클라이언트 인증 없이 **https://my\_company.idm.example.com/Example/** URL에 액세스합니다.

```
$ curl https://my_company.idm.example.com/Example/
curl: (56) OpenSSL SSL_read: error:1409445C:SSL routines:ssl3_read_bytes:tlsv13 **alert certificate required**, errno 0
```

이 오류는 **my\_company.idm.example.com** 웹 서버에 클라이언트 인증서 인증이 필요함을 나타냅니다.

- 2.

클라이언트 개인 키 및 인증서와 **CA** 인증서를 **curl**에 전달하여 클라이언트 인증으로 동일한 URL에 액세스합니다.

```
$ curl --cacert ca.crt --key client.key --cert client.crt
https://my_company.idm.example.com/Example/
```

요청이 성공하면 **curl**은 **/var/www/html/Example/** 디렉터리에 저장된 **index.html** 파일을 표시합니다.

#### 추가 리소스

- **Apache** 설명서의 **mod\_ssl** 구성 방법 설명서를 참조하십시오.
- **Apache HTTP Server** 수동 설치를 참조하십시오.

#### 17.11. 새 사용자 인증서 요청 및 클라이언트로 내보내기

**IdM(Identity Management)** 관리자는 **IdM** 클라이언트에서 실행되는 웹 서버를 구성하여 웹 브라우저를 사용하여 서버에 액세스하여 특정 **IdM** 하위 명령에서 발급한 인증서로 인증하는 사용자를 요청할 수 있습니다. 사용자가 웹 브라우저를 사용하여 웹 서버에 액세스하려는 호스트로 인증서 및 해당 개인 키를 내보내려면 이 섹션을 완료합니다. 그런 다음 인증서와 개인 키를 브라우저로 가져옵니다.

#### 절차

1.

필요한 경우 새 디렉터리(예: **~/certdb/**)를 생성하고 임시 인증서 데이터베이스를 만듭니다. 메시지가 표시되면 **NSS** 인증서 **DB** 암호를 생성하여 후속 단계에서 생성할 인증서에 대한 키를 암호화합니다.

```
mkdir ~/certdb/
certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2.

**CSR**(인증서 서명 요청)을 생성하고 출력을 파일로 리디렉션합니다. 예를 들어 **IDM.EXAMPLE.COM** 영역에 있는 **idm\_user** 사용자의 **4096** 비트 인증서에 대해 **certificate\_request.csr** 이라는 이름으로 **CSR**을 생성하려면 인증서 개인 키의 **nickname**을 쉽

게 찾을 수 있도록 `idm_user` 로 설정하고 `CN=idm_user,O=IDMM.EXALE :COM`으로 설정합니다.

```
certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s
"CN=idm_user,O=IDM.EXAMPLE.COM" > certificate_request.csr
```

3.

메시지가 표시되면 `certutil` 을 사용하여 임시 데이터베이스를 생성할 때 입력한 암호와 동일한 암호를 입력합니다. 다음으로 중지될 때까지 `randlonly`를 계속 입력합니다.

Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the creation of your key. One of the easiest ways to create a random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

4.

인증서 요청 파일을 서버에 제출합니다. 새로 발급된 인증서, 인증서를 저장할 출력 파일, 선택적으로 인증서 프로필과 연결할 `Kerberos` 주체를 지정합니다. 인증서를 발행할 `IdM` 하위 `CA` 를 지정합니다. 예를 들어, `webclient-ca` 의 `idm_user@IDM.EXAMPLE.COM` 주체에 대한 사용자 역할 확장이 추가된 프로필인 `IECUserRoles` 프로필의 인증서를 얻으려면 `~/idm_user.pem` 파일에 인증서를 저장합니다.

```
ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --ca=webclient-ca --certificate-out=~/idm_user.pem
```

5.

`NSS` 데이터베이스에 인증서를 추가합니다. `-n` 옵션을 사용하여 이전에 `CSR`을 생성할 때 사용한 것과 동일한 `nickname`을 설정하여 인증서가 `NSS` 데이터베이스의 개인 키와 일치하도록 합니다. `t` 옵션은 신뢰 수준을 설정합니다. 자세한 내용은 `certutil(1)` 매뉴얼 페이지를 참조하십시오. `i` 옵션은 입력 인증서 파일을 지정합니다. 예를 들어 `NSS` 데이터베이스에 `~/certdb/` 데이터베이스의 `~/idm_user.pem` 파일에 저장된 `idm_user` `nickname`을 사용하여 인증서를 추가하려면 다음을 수행합니다.

```
certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6.

`NSS` 데이터베이스의 키가 낙네임 (`orphan`) 으로 표시되지 않는지 확인합니다. 예를 들어 `~/certdb/` 데이터베이스에 저장된 인증서가 분리되지 않았는지 확인하려면 다음을 수행합니다.

```
certutil -K -d ~/certdb/
< 0> rsa 5ad14d41463b87a095b1896cf0068ccc467df395 NSS Certificate
DB:idm_user
```

7.

**pk12util** 명령을 사용하여 **NSS** 데이터베이스에서 **PKCS12** 형식으로 인증서를 내보냅니다. 예를 들어 **idm\_user** nickname을 사용하여 인증서를 **/root/certdb** **NSS** 데이터베이스에서 **~/idm\_user.p12** 파일로 내보내려면 다음을 실행합니다.

```
pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8.

**idm\_user**의 인증서 인증을 활성화할 호스트에 인증서를 전송합니다.

```
scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9.

인증서가 전송된 호스트에서 **.pkcs12** 파일이 저장된 디렉토리를 보안상의 이유로 **'other'** 그룹에 저장할 수 없도록 합니다.

```
chmod o-rwx /home/idm_user/
```

10.

보안상의 이유로 서버에서 임시 **NSS** 데이터베이스 및 **.pkcs12** 파일을 제거하십시오.

```
rm ~/certdb/
rm ~/idm_user.p12
```

## 17.12. 인증서 인증을 사용하도록 브라우저 구성

**WebUI**를 사용하여 **IdM(Identity Management)**에 로그인할 때 인증서로 인증할 수 있으려면 사용자와 관련 인증 기관(**CA**) 인증서를 **Mozilla Firefox** 또는 **Google Chrome** 브라우저로 가져와야 합니다. 브라우저가 실행 중인 호스트 자체는 **IdM** 도메인의 일부가 될 필요가 없습니다.

**IdM**은 다음 브라우저를 지원하여 **WebUI** 연결을 지원합니다.

- **Mozilla Firefox 38 이상**
- **Google Chrome 46 이상**

다음 절차에서는 **Mozilla Firefox 57.0.1** 브라우저를 구성하는 방법을 보여줍니다.

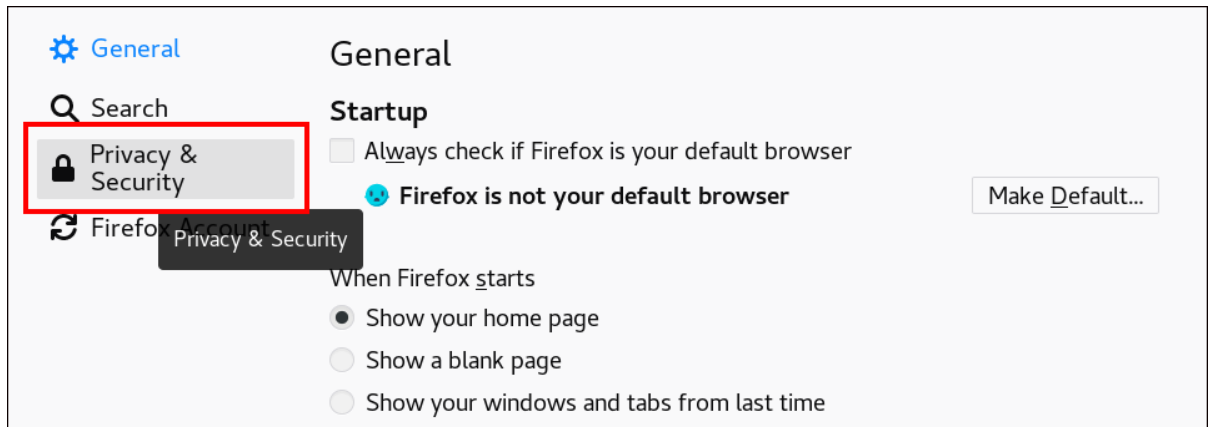
#### 사전 요구 사항

- **PKCS#12** 형식으로든 브라우저로 가져올 **사용자 인증서** 가 있습니다.
- 하위 **CA 인증서**를 다운로드하여 **PEM** 형식으로 폐기합니다.

#### 절차

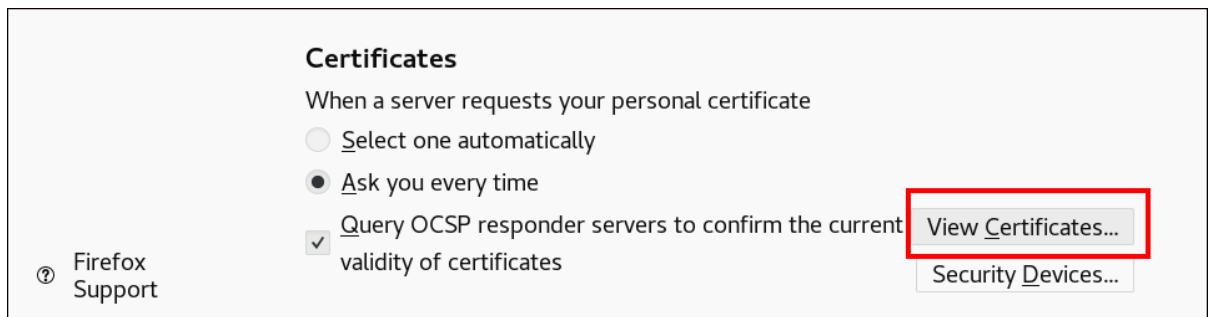
1. **Firefox**를 연 다음 기본 설정 → 개인 정보 보호 및 보안 으로 이동합니다.

그림 17.8. 기본 설정의 개인 정보 및 보안 섹션



2. **View Certificates** (인증서 보기)를 클릭합니다.

그림 17.9. 개인 정보 보호 및 보안의 인증서 보기



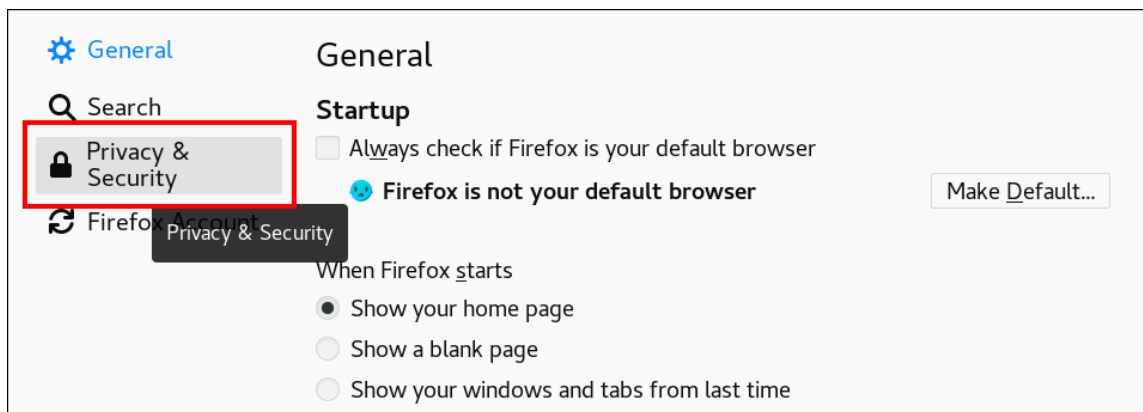
3. 인증서 탭에서 가져오기 를 클릭합니다. **PKCS12** 형식으로 사용자 인증서를 찾아서 연 다음 확인 및 확인을 클릭합니다.
- 4.

IdM 하위 CA가 Firefox에서 신뢰할 수 있는 기관으로 인식되도록 하려면 IdM WebUI에서 하위 CA 인증서를 신뢰할 수 있는 인증 기관 인증서로 다운로드할 때 저장한 IdM 하위 CA 인증서를 가져옵니다.

a.

Firefox를 열고 기본 설정으로 이동하고 개인 정보 및 보안을 클릭합니다.

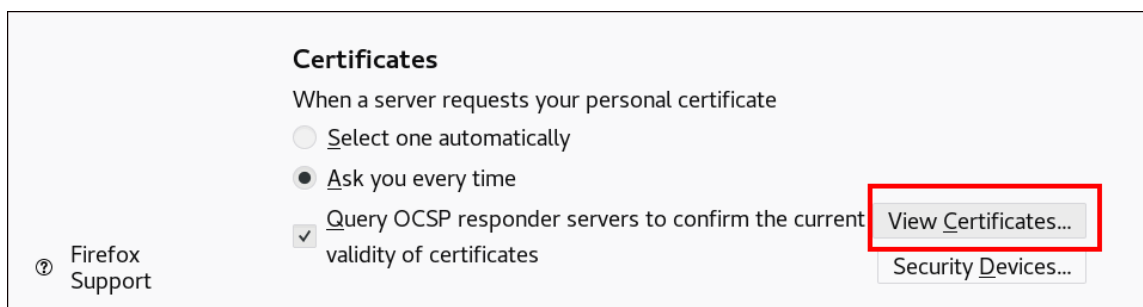
그림 17.10. 기본 설정의 개인 정보 및 보안 섹션



b.

View Certificates (인증서 보기)를 클릭합니다.

그림 17.11. 개인 정보 보호 및 보안의 인증서 보기



c.

Authorities (권한) 탭에서 Import (가져오기)를 클릭합니다. 하위 CA 인증서를 찾아서 엽니다. 인증서를 신뢰하여 웹 사이트를 식별한 다음 확인을 클릭하고 확인을 클릭합니다.

## 18장. 특정 관련 인증서 그룹 무효화를 신속하게

시스템 관리자는 특정 관련 인증서 그룹을 신속하게 무효화하려면 다음을 수행하십시오.

- 특정 **IdM(Lightweight Identity Management)** 하위 **CA**에서 발급한 인증서만 신뢰하도록 애플리케이션을 설계합니다. 이후 이러한 인증서를 발급한 **IdM(Identity Management)** 하위 **CA** 인증서만 취소하여 이러한 모든 인증서를 무효화할 수 있습니다. **IdM**에서 경량 하위 **CA**를 생성하고 사용하는 방법에 대한 자세한 내용은 **특정 관련 인증서 그룹의 유효성을 신속하게** 평가할 수 있도록 참조하십시오.
- **To-be-revoked IdM** 하위 **CA**에서 발행한 모든 인증서가 즉시 유효하지 않게 하려면 **IdM OCSP** 응답자를 사용하도록 이러한 인증서를 사용하는 애플리케이션을 구성합니다. 예를 들어 **OCSP** 응답자를 사용하도록 **Firefox** 브라우저를 구성하려면 쿼리 **OCSP** 응답기 서버가 **Firefox** 기본 설정에서 인증서의 현재 유효성이 선택되었는지 확인합니다.

**IdM**에서 **CRL**(인증서 취소 목록)은 4시간마다 업데이트됩니다. **IdM** 하위 프로세스에서 발행한 모든 인증서를 무효화하려면 **IdM 하위 CA 인증서**를 취소해야 합니다. 또한 **관련 CA ACL**을 **비활성화**하고 **IdM 하위 CA 비활성화**를 고려하십시오. 하위 **CA**를 비활성화하면 하위 **CA**에서 새 인증서를 발행할 수 없지만 하위 **CA**의 서명 키가 유지되므로 이전에 발급한 인증서에 대해 **OCSP(Online Certificate Status Protocol)** 응답이 생성됩니다.

### 중요

환경에서 **OCSP**를 사용하는 경우 하위 **CA**를 삭제하지 마십시오. 하위 **CA**를 삭제하면 하위 **CA**의 서명 키가 삭제되어 해당 하위 **CA**에서 발급한 인증서에 대해 **OCSP** 응답을 프로덕션할 수 없습니다.

하위 **CA**를 삭제할 때 유일한 시나리오는 동일한 주체 고유 이름(**DN**)으로 새 하위 **CA**를 생성하되 새 서명 키를 생성하려는 경우입니다.

### 18.1. IdM CLI에서 CA ACL 비활성화

**IdM** 서비스 또는 **IdM** 서비스 그룹을 폐기하려면 기존 해당 **CA ACL**을 비활성화하는 것이 좋습니다.

이 섹션을 완료하여 **IdM** 클라이언트에서 실행 중인 웹 서버가 **webserver-ca IdM** 하위 **CA**에서 발급하도록 인증서를 요청하고, 웹 **client-ca IdM** 하위 프로세스에서 발행하도록 **IdM** 사용자를 제한하는 **TLS\_web\_client\_authentication CA ACL**을 비활성화하는 **TLS\_web\_client\_authentication CA ACL**을 비활성화합니다.

1.

필요한 경우 IdM 환경의 모든 CA ACL을 보려면 **ipa caacl-find** 명령을 입력합니다.

```
$ ipa caacl-find

3 CA ACLs matched

ACL name: hosts_services_calPAserviceCert
Enabled: TRUE

ACL name: TLS_web_server_authentication
Enabled: TRUE

ACL name: TLS_web_client_authentication
Enabled: TRUE
```

2.

필요한 경우 CA ACL의 세부 정보를 보려면 **ipa caacl-show** 명령을 입력하고 CA ACL 이름을 지정합니다.

```
$ ipa caacl-show TLS_web_server_authentication
ACL name: TLS_web_server_authentication
Description: CAACL for web servers authenticating to web clients using certificates
issued by webserver-ca
Enabled: TRUE
CAs: webserver-ca
Profiles: calPAserviceCert
Services: HTTP/rhel8server.idm.example.com@IDM.EXAMPLE.COM
```

3.

CA ACL을 비활성화하려면 **ipa caacl-disable** 명령을 입력하고 CA ACL 이름을 지정합니다.

•

TLS\_web\_server\_authentication CA ACL을 비활성화하려면 다음을 입력합니다.

```
$ ipa caacl-disable TLS_web_server_authentication

Disabled CA ACL "TLS_web_server_authentication"

```

•

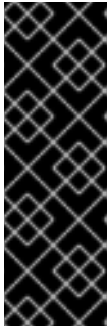
TLS\_web\_client\_authentication CA ACL을 비활성화하려면 다음을 입력합니다.

```
$ ipa caacl-disable TLS_web_client_authentication

Disabled CA ACL "TLS_web_client_authentication"

```

이제 활성화된 유일한 **CA ACL**은 **hosts\_services\_calPAserviceCert CA ACL**입니다.



#### 중요

**hosts\_services\_calPAserviceCert CA ACL** 비활성화에 대해 매우 주의하십시오. **calPAserviceCert** 프로필을 사용하여 **ipa CA**를 사용하는 다른 **CA** 서버를 부여하지 않고 **hosts\_services\_calPAserviceCert** 인증서를 비활성화하면 **IdM HTTP** 및 **LDAP** 인증서가 갱신되지 않습니다. 만료된 **IdM HTTP** 및 **LDAP** 인증서로 인해 **IdM** 시스템이 실패합니다.

## 18.2. IDM 하위 명령 비활성화

해당 하위 **CA**에서 발행한 모든 인증서를 무효화하기 위해 **IdM** 하위 **CA**의 **CA** 인증서를 취소한 후 더 이상 필요하지 않은 경우 **IdM** 하위 캐시를 비활성화하는 것이 좋습니다. 나중에 하위 **CA**를 다시 활성화할 수 있습니다.

하위 **CA**를 비활성화하면 하위 **CA**에서 새 인증서를 발행할 수 없지만 하위 **CA**의 서명 키가 유지되므로 이전에 발급한 인증서에 대해 **OCSP(Online Certificate Status Protocol)** 응답이 생성됩니다.

#### 사전 요구 사항

- **IdM** 관리자로 로그인되어 있습니다.

#### 절차

- **ipa ca-disable** 명령을 입력하고 하위 **CA** 이름을 지정합니다.

```
$ ipa ca-disable webserver-CA
```

```

Disabled CA "webserver-CA"

```

## 19장. IDM 상태 점검을 사용하여 인증서 확인

이 섹션에서는 **IdM(Identity Management)**에서 **Healthcheck** 툴을 이해하고 사용하여 **certmonger**에서 유지 관리하는 **IPA** 인증서의 문제를 식별하는 데 도움이 됩니다.

자세한 내용은 **IdM의 상태 점검**을 참조하십시오.

### 19.1. IDM 인증서 상태 점검 테스트

**Healthcheck** 툴에는 **IdM(Identity Management)**의 **certmonger**에서 유지 관리하는 인증서 상태를 확인하는 여러 테스트가 포함됩니다. **certmonger**에 대한 자세한 내용은 **certmonger**를 **사용하여 서비스에 대한 IdM 인증서 Obtaining an IdM certificate**을 참조하십시오.

이 테스트 검사 모음의 만료, 검증, 신뢰 및 기타 문제. 동일한 기본 문제에 대해 여러 오류가 발생할 수 있습니다.

모든 인증서 테스트를 보려면 **--list-sources** 옵션을 사용하여 **ipa-healthcheck**을 실행합니다.

```
ipa-healthcheck --list-sources
```

**ipahealthcheck.ipa.certs** 소스에서 모든 테스트를 찾을 수 있습니다.

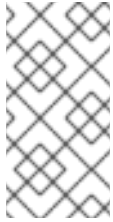
#### IPACertmongerExpirationCheck

이 테스트에서는 **certmonger**의 만료를 확인합니다.

오류가 보고되면 인증서가 만료되었습니다.

경고가 표시되면 인증서가 곧 만료됩니다. 기본적으로 이 테스트는 인증서가 만료되기 전 **28일** 이내에 적용됩니다.

**/etc/ipahealthcheck/ipahealthcheck.conf** 파일에서 일 수를 구성할 수 있습니다. 파일을 연 후 **default** 섹션에 있는 **cert\_expiration\_days** 옵션을 변경합니다.

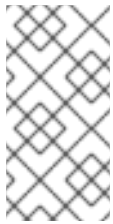


## 참고

**certmonger**는 인증서 만료에 대한 자체 보기를 로드하고 유지합니다. 이 검사에서는 디스크상의 인증서의 유효성을 검사하지 않습니다.

## IPACertfileExpirationCheck

이 테스트에서는 인증서 파일 또는 **NSS** 데이터베이스를 열 수 있는지 확인합니다. 이 테스트는 또한 만료를 확인합니다. 따라서 오류 또는 경고 출력에서 **msg** 속성을 주의 깊게 읽습니다. 메시지는 문제를 지정합니다.



## 참고

이 테스트에서는 디스크상의 인증서를 확인합니다. 인증서가 누락된 경우 읽을 수 없으므로 별도의 오류도 발생할 수 있습니다.

## IPACertNSSTrust

이 테스트에서는 **NSS** 데이터베이스에 저장된 인증서에 대한 신뢰를 비교합니다. **NSS** 데이터베이스에서 예상되는 추적된 인증서의 경우 신뢰는 예상 값과 비교되며 일치하지 않는 항목에서 발생하는 오류입니다.

## IPANSSChainValidation

이 테스트에서는 **NSS** 인증서의 인증서 체인의 유효성을 검사합니다. 테스트 실행: **certutil -V -u V -e -d [dbdir] -n [nickname]**

## IPAOpenSSLChainValidation

이 테스트에서는 **OpenSSL** 인증서의 인증서 체인을 검증합니다. **NSSChain** 검증과 비교하려면 다음을 실행하는 **OpenSSL** 명령을 실행합니다.

```
openssl verify -verbose -show_chain -CAfile /etc/ipa/ca.crt [cert file]
```

## IPARAAGent

이 테스트에서는 디스크의 인증서를 **uid=ipara,ou=People,o=ipaca**의 **LDAP**의 동등한 레코드와 비교합니다.

## IPACertRevocation

이 테스트에서는 **certmonger**를 사용하여 인증서가 취소되지 않았는지 확인합니다. 따라서 이 테스트는 **certmonger**에서만 유지 관리하는 인증서와 관련된 문제를 찾을 수 있습니다.

## IPACertmongerCA

이 테스트는 **certmonger CA**(인증 기관) 구성을 확인합니다. **IdM**은 **CA** 없이 인증서를 발행할 수 없습니다.

**certmonger**는 **CA** 도우미 세트를 유지합니다. **IdM**에는 **IdM**을 통해 인증서를 발행하고 호스트 또는 서비스 인증서에 대한 호스트 또는 사용자 주체로 인증하는 **IPA**라는 **CA**가 있습니다.

**CA** 하위 시스템 인증서를 갱신하는 **dogtag-ipa-ca-renew-agent** 및 **dogtag-ipa-ca-renew-agent-reuse** 도 있습니다.



참고

문제를 확인하려고 할 때 모든 **IdM** 서버에서 이 테스트를 실행합니다.

### 19.2. HEALTHCHECK 도구를 사용하여 인증서 확인

이 섹션에서는 **Healthcheck** 툴을 사용하여 **IdM**(Identity Management) 인증서 상태 점검의 독립 실행형 수동 테스트에 대해 설명합니다.

**Healthcheck** 툴에는 많은 테스트가 포함되어 있으므로 다음과 같이 결과를 단축할 수 있습니다.

- 모든 성공적인 테스트 제외: **--failures-only**
- 인증서 테스트만 포함: **--source=ipahealthcheck.ipa.certs**

#### 사전 요구 사항

- 상태 점검 테스트는 **root** 사용자로 수행해야 합니다.

#### 절차

- 인증서와 관련된 경고, 오류 및 심각한 문제로 **Healthcheck**를 실행하려면 다음을 입력합니다.

```
ipa-healthcheck --source=ipahealthcheck.ipa.certs --failures-only
```

테스트에 빈 대괄호가 표시됩니다.

```
[]
```

실패한 테스트에서는 다음 출력을 보여줍니다.

```
{
 "source": "ipahealthcheck.ipa.certs",
 "check": "IPACertfileExpirationCheck",
 "result": "ERROR",
 "kw": {
 "key": 1234,
 "dbdir": "/path/to/nssdb",
 "error": [error],
 "msg": "Unable to open NSS database '/path/to/nssdb': [error]"
 }
}
```

이 **IPACertfileExpirationCheck** test가 **NSS** 데이터베이스를 여는 데 실패했습니다.

추가 리소스

- **man ipa-healthcheck** 를 참조하십시오.

## 20장. IDM 상태 점검을 사용하여 시스템 인증서 확인

이 섹션에서는 **IdM(Identity Management)**의 상태 점검 툴을 설명하여 시스템 인증서 문제를 식별합니다.

자세한 내용은 **IdM의 상태 점검**을 참조하십시오.

### 20.1. 시스템 인증서 상태 점검 테스트

**Healthcheck** 툴에는 시스템(**DogTag**) 인증서를 확인하기 위한 여러 테스트가 포함되어 있습니다.

모든 테스트를 보려면 **--list-sources** 옵션을 사용하여 **ipa-healthcheck** 을 실행합니다.

```
ipa-healthcheck --list-sources
```

**ipahealthcheck.dogtag.ca** 소스에서 모든 테스트를 찾을 수 있습니다.

#### DogtagCertsConfigCheck

이 테스트에서는 **NSS** 데이터베이스의 **CA(Certificate Authority)** 인증서를 **CS.cfg** 에 저장된 동일한 값과 비교합니다. 일치하지 않는 경우 **CA**가 시작되지 않습니다.

특히 확인을 수행합니다.

- **auditSigningCert cert-pki-ca against ca.audit\_signing.cert**
- **ocspSigningCert cert-pki-ca against ca.ocsp\_signing.cert**
- **caSigningCert cert-pki-ca against ca.signing.cert**
- **subsystemCert cert-pki-ca against ca.subsystem.cert**
- **server-Cert cert-pki-ca against ca.sslserver.cert**

키 복구 기관(KRA)이 설치된 경우:

- `transportCert cert-pki-kra against ca.connector.KRA.transportCert`

### DogtagCertsConnectivityCheck

이 테스트에서는 연결을 확인합니다. 이 테스트는 검사를 수행하는 `ipa cert-show 1` 명령과 동일합니다.

- Apache의 PKI 프록시 구성
- IdM에서 CA를 찾을 수 있음
- RA 에이전트 클라이언트 인증서
- 요청에 대한 CA 응답의 수정

`cert-show` 를 실행할 수 있는지 확인하고 CA(인증서 또는 찾을 수 없음)에서 예상 결과를 반환할 수 있으므로 테스트에서 직렬 #1로 인증서를 확인합니다.



참고

문제를 해결하려고 할 때 모든 IdM 서버에서 이 테스트를 실행합니다.

## 20.2. HEALTHCHECK를 사용하여 시스템 인증서 확인

이 섹션에서는 `Healthcheck` 툴을 사용하여 IdM(Identity Management) 인증서의 독립 실행형 수동 테스트에 대해 설명합니다.

상태 점검 툴에는 많은 테스트가 포함되어 있으므로 `DogTag tests: --source=ipahealthcheck.dogtag.ca`만 포함하면 결과를 좁힐 수 있습니다.

절차

- Healthcheck restricted를 DogTag 인증서로 제한하려면 다음을 입력합니다.

```
ipa-healthcheck --source=ipahealthcheck.dogtag.ca
```

성공적인 테스트의 예:

```
{
 "source: ipahealthcheck.dogtag.ca",
 "check: DogtagCertsConfigCheck",
 "result: SUCCESS",
 "uuid: 9b366200-9ec8-4bd9-bb5e-9a280c803a9c",
 "when: 20191008135826Z",
 "duration: 0.252280",
 "kw:" {
 "key": "Server-Cert cert-pki-ca",
 "configfile": "/var/lib/pki/pki-tomcat/conf/ca/CS.cfg"
 }
}
```

실패한 테스트의 예는 다음과 같습니다.

```
{
 "source: ipahealthcheck.dogtag.ca",
 "check: DogtagCertsConfigCheck",
 "result: CRITICAL",
 "uuid: 59d66200-1447-4b3b-be01-89810c803a98",
 "when: 20191008135912Z",
 "duration: 0.002022",
 "kw:" {
 "exception": "NSDB /etc/pki/pki-tomcat/alias not initialized",
 }
}
```

추가 리소스

- man ipa-healthcheck 를 참조하십시오.