Red Hat Enterprise Linux 8

RHEL에서 인증 및 권한 부여 구성

SSSD, authselect 및 sssctl을 사용하여 인증 및 권한 부여 구성

Last Updated: 2022-07-10

Red Hat Enterprise Linux 8 RHEL에서 인증 및 권한 부여 구성

SSSD, authselect 및 sssctl을 사용하여 인증 및 권한 부여 구성

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

초록

이 문서 컬렉션은 Red Hat Enterprise Linux 8 호스트에서 인증 및 권한 부여를 구성하는 방법에 대한 지침을 제공합니다.

차례

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.

Identity Management에서는 계획된 용어 교체는 다음과 같습니다.

- ***블록 목록이*** *블랙리스트*로 대체
- ***허용 목록이*** *허용 목록*교체
- ***보조*** 로 *슬레이브*교체
- *마스터* 라는 단어가 컨텍스트에 따라 더 정확한 언어로 교체되고 있습니다.
  - ***IdM 서버*** *IdM 마스터*교체
  - ***CA 갱신 서버*** *CA 갱신 마스터*교체
  - ***CRL 게시자 서버가*** *CRL 마스터*교체
  - ***다중*** 마스터 교체

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. 문서를 개선할 수 있는 방법에 관해 알려주십시오.

- 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 이동하십시오.
  2. Component로 **Documentation**을 선택하십시오.
  3. **Description** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
  4. **Submit Bug**를 클릭하십시오.

1장. AUTHSELECT를 사용하여 사용자 인증 구성

Authselect 는 특정 프로필을 선택하여 시스템 ID 및 인증 소스를 구성할 수 있는 유틸리티입니다. 프로필은 결과적으로 PAM(Pluggable Authentication Modules) 및 NSS(Network Security Services) 구성을 설명하는 파일 집합입니다. 기본 프로필을 선택하거나 사용자 지정 프로필을 만들 수 있습니다.

1.1. AUTHSELECT 사용

authselect 유틸리티를 사용하여 Red Hat Enterprise Linux 8 호스트에서 사용자 인증을 구성할 수 있습니다.

준비된 프로필 중 하나를 선택하여 ID 정보 및 인증 소스 및 공급자를 구성할 수 있습니다.

- 기본 **sssd** 프로필은 LDAP 인증을 사용하는 시스템에 대해 SSSD(System Security Services Daemon)를 활성화합니다.
- **winbind** 프로필을 사용하면 Microsoft Active Directory와 직접 통합된 시스템에 대해 Winbind 유틸리티를 사용할 수 있습니다.
- **nis** 프로필은 레거시 NIS(Network Information Service) 시스템과의 호환성을 보장합니다.
- **최소** 프로필은 시스템 파일에서 직접 로컬 사용자와 그룹만 제공하므로 관리자가 더 이상 필요하지 않은 네트워크 인증 서비스를 제거할 수 있습니다.

지정된 호스트에 대해 **authselect** 프로필을 선택하면 이 프로필이 호스트에 로그인하는 모든 사용자에게 적용됩니다.

반중앙화된 ID 관리 환경에서 **authselect** 를 사용하는 것이 좋습니다. 예를 들어 조직에서 LDAP, Winbind 또는 NIS 데이터베이스를 활용하여 도메인의 서비스를 사용하도록 사용자를 인증하는 것이 좋습니다.

주의

*다음과 같은 경우 **authselect** 를 사용할 필요가 없습니다.*

- *호스트는 Red Hat Enterprise Linux IdM(Identity Management)의 일부입니다. **ipa-client-install** 명령을 사용하여 호스트를 IdM 도메인에 연결하면 호스트에서 SSSD 인증을 자동으로 구성합니다.*
- *호스트는 SSSD를 통한 Active Directory의 일부입니다. 호스트를 Active Directory 도메인에 연결하도록 **realm join** 명령 호출은 호스트에서 SSSD 인증을 자동으로 구성합니다.*

***ipa-client-install** 또는 **realm join** 으로 구성된 **authselect** 프로필을 변경하지 않는 것이 좋습니다. 수정이 필요한 경우 수정하기 전에 현재 설정을 표시하여 필요한 경우 되돌릴 수 있습니다.*

```
$ authselect current
Profile ID: sssd
Enabled features:
- with-sudo
- with-mkhomedir
- with-smartcard
```

1.1.1. 파일 및 디렉터리 authselect 수정

*이전 Red Hat Enterprise Linux 버전에서 사용된 **authconfig** 유틸리티는 다양한 구성 파일을 생성 및 수정하여 문제 해결을 어렵게 만듭니다. **Authselect** 는 다음 파일 및 디렉터리만 수정하므로 테스트 및 문제 해결을 간소화합니다.*

/etc/nsswitch.conf	GNU C 라이브러리 및 기타 애플리케이션은 이 NSS(Name Service Switch) 구성 파일을 사용하여 다양한 범주에서 이름 서비스 정보를 가져올 소스와 순서가 무엇인지 확인합니다. 정보의 각 카테고리는 데이터베이스 이름으로 식별됩니다.
/etc/pam.d/* 파일	Linux-PAM(Pluggable Authentication Modules)은 시스템에서 애플리케이션(서비스)의 인증 작업을 처리하는 모듈 시스템입니다. 인증의 특성은 동적으로 구성할 수 있습니다. 시스템 관리자는 개별 서비스 제공 애플리케이션이 사용자를 인증하는 방법을 선택할 수 있습니다. **/etc/pam.d/** 디렉터리의 구성 파일은 서비스에 필요한 인증 작업을 수행할 PAM과 개별 PAM이 실패하는 경우 PAM-API의 적절한 동작을 나열합니다. 특히 이 파일에는 다음에 대한 정보가 포함되어 있습니다. • 사용자 암호 잠금 조건 • 스마트 카드로 인증하는 기능 • 지문 리더로 인증하는 기능

/etc/dconf/db/distro.d/* 파일	이 디렉토리에는 **dconf** 유틸리티의 구성 프로필이 있으며, 이 프로필은 GNOME 데스크탑 GUI(그래픽 사용자 인터페이스)의 설정을 관리하는 데 사용할 수 있습니다.

1.1.2. /etc/nsswitch.conf의 데이터 공급자

*기본 **sssd** 프로필은 **/etc/nsswitch.conf에 sss 항목을 생성하여 SSSD를 정보 소스로 설정합니다.***

```
passwd:     sss files
group:      sss files
netgroup:   sss files
automount:  sss files
services:   sss files
...
```

즉, 시스템은 먼저 이러한 항목 중 하나에 대한 정보가 요청되는 경우 SSSD를 찾습니다.

- *사용자 정보의 **passwd***
- *사용자 **그룹** 정보를 위한 그룹*
- *NIS **netgroup** 정보를 위한 **netgroup***
- *NFS **자동 마운트** 정보를 위한 자동 마운트*
- ***서비스** 관련 정보의 서비스*

sssd** 캐시에 요청한 정보를 찾을 수 없고 인증을 제공하는 서버에서 요청한 정보를 찾을 수 없거나 **sssd** 가 실행 중이 아닌 경우 시스템은 로컬 파일 즉 **/etc/ 를 확인합니다.*

*예를 들어 사용자 ID에 대한 정보가 요청되면 사용자 ID가 **sssd** 캐시에서 먼저 검색됩니다. 여기에서 찾을 수 없는 경우 **/etc/passwd** 파일을 참조합니다. 논리적으로 사용자의 그룹 제휴가 요청되면 **sssd** 캐시에서 먼저 검색되며 찾을 수 없는 경우에만 **/etc/group** 파일이 참조됩니다.*

*실제로는 로컬 **파일** 데이터베이스가 일반적으로 참조되지 않습니다. 가장 중요한 예외는 **sssd** 에 의해 처리되지 않지만 **파일에** 의해 처리되지 않는 **루트** 사용자의 경우입니다.*

1.2. AUTHSELECT 프로파일 선택

*시스템 관리자는 특정 호스트의 **authselect** 유틸리티에 대한 프로필을 선택할 수 있습니다. 프로필은 호스트에 로그인하는 모든 사용자에게 적용됩니다.*

사전 요구 사항

- ***authselect** 명령을 실행하려면 **root** 인증 정보가 필요합니다*

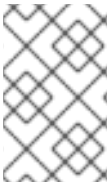
절차

- *인증 공급자에 적합한 **authselect** 프로필을 선택합니다. 예를 들어 LDAP를 사용하는 회사의 네트워크에 로그인하려면 **sssd** 를 선택합니다.*

```
# authselect select sssd
```

- *(선택 사항)* ***authselect select sssd* 또는 *authselect select winbind*** *명령에 다음 옵션을 추가하여 기본 프로필 설정을 수정할 수 있습니다.*
  - ***with-faillock***
  - ***with-smartcard***
  - ***with-fingerprint***

사용 가능한 옵션의 전체 목록을 보려면 스크립트 변환을 authconfig에서 authselect 또는 ***authselect*** *-migration(7) 도움말 페이지로 변환을 참조하십시오.*



참고

authselect 선택 *절차를 완료하기 전에 프로필에 해당하는 구성 파일이 올바르게 구성되었는지 확인합니다. 예를 들어* ***sssd*** *데몬이 올바르게 구성되지 않고 활성 상태가 되면* ***authselect select*****를** *실행하면 로컬 사용자만* ***pam_unix*** *를 사용하여 인증할 수 있습니다.*

검증 단계

1. *SSSD에 대한* ***ssss*** *항목이* ***/etc/nsswitch.conf*** *에 있는지 확인하십시오:*

```
passwd:     sss files
group:      sss files
netgroup:   sss files
automount:  sss files
services:   sss files
...
```

2. ***pam_sss .so 항목의 /etc/pam.d/system-auth 파일의 내용을 검토합니다.***

```
# Generated by authselect on Tue Sep 11 22:59:06 2018
# Do not modify this file manually.

auth        required      pam_env.so
auth        required      pam_faildelay.so delay=2000000
auth        [default=1 ignore=ignore success=ok]    pam_succeed_if.so uid >= 1000 quiet
auth        [default=1 ignore=ignore success=ok]    pam_localuser.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        sufficient    pam_sss.so forward_pass
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
...
```

추가 리소스

- *authselect 사용*
- *준비된 authselect 프로파일 수정*
- *고유한 authselect 프로필 생성 및 배포*

1.3. 준비된 AUTHSELECT 프로파일 수정

시스템 관리자는 요구 사항에 맞게 기본 프로필 중 하나를 수정할 수 있습니다.

다음을 제외하고 **/etc/authselect/user-nsswitch.conf** 파일의 항목을 수정할 수 있습니다.

- **passwd**
- **group**
- **netgroup**
- **자동 마운트**
- **services**

나중에 **authselect select profile_name** 을 실행하면 허용되는 변경 사항이 **/etc/authselect/user-nsswitch.conf** 에서 **/etc/nsswitch.conf** 파일로 전송됩니다. 허용되지 않는 변경 사항은 기본 프로필 구성에서 덮어씁니다.

중요

/etc/nsswitch.conf 파일을 직접 수정하지 마십시오.

절차

1. **authselect** 프로필을 선택합니다. 예를 들면 다음과 같습니다.

```
# authselect select sssd
```

2. 원하는 변경 사항으로 **/etc/authselect/user-nsswitch.conf** 파일을 편집합니다.
3. **/etc/authselect/user-nsswitch.conf 파일의 변경 사항을 적용합니다.**

```
# authselect apply-changes
```

검증 단계

- **/etc/nsswitch.conf** 파일을 검토하여 **/etc/authselect/user-nsswitch.conf** 의 변경 사항이 여기에 전파되었는지 확인합니다.

추가 리소스

- authselect 사용

1.4. 고유한 AUTHSELECT 프로필 생성 및 배포

시스템 관리자는 기본 프로필 중 하나의 사용자 지정 복사본을 만들어 사용자 지정 프로필을 생성하고 배포할 수 있습니다.

이 기능은 준비된 authselect 프로필을 수정하는 것이 요구 사항에 적합하지 않은 경우 특히 유용합니다. 사용자 지정 프로필을 배포할 때 지정된 호스트에 로그인하는 모든 사용자에게 프로필이 적용됩니다.

절차

1. ***authselect create-profile*** *명령을 사용하여 사용자 지정 프로필을 생성합니다. 예를 들어 준비된* ***sssd*** *프로필을 기반으로* ***user-profile이라는 사용자*** *지정 프로필을 생성하려면* ***/etc/nsswitch.conf 파일에서 직접 항목을 구성할 수 있습니다.***

```
# authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
New profile was created at /etc/authselect/custom/user-profile
```

명령에 ***--symlink-pam*** *옵션을 포함하면 PAM 템플릿이 복사 대신 원본 프로필 파일에 대한 심볼릭 링크임을 의미합니다.* ***--symlink-meta*** *옵션을 사용하면 README 및 REQUIREMENTS와 같은 메타 파일이 복사 대신 원본 프로필 파일에 대한 심볼릭 링크입니다. 이렇게 하면 원래 프로필의 PAM 템플릿 및 메타 파일에 대한 향후 모든 업데이트가 사용자 지정 프로필에 반영됩니다.*

이 명령은 ***/etc/ authselect/custom/user-profile/ 디렉터리에 /etc/ nsswitch.conf*** *파일의 복사본을 생성합니다.*

2. ***/etc/authselect/custom/user-profile/nsswitch.conf*** *파일을 구성합니다.*

3. ***authselect select*** *명령을 실행하고* ***custom/name_of_the_profile을 매개 변수로 추가하여 사용자 지정*** *프로필을 선택합니다. 예를 들어* ***사용자 프로필 프로필*** *을 선택하려면 다음을 수행합니다.*

```
# authselect select custom/user-profile
```

시스템에 대한 ***사용자 프로필 프로필*** *을 선택하면 나중에 Red Hat에서* ***sssd*** *프로필이 업데이트되면* ***/etc/nsswitch.conf*** *파일에 대한 업데이트를 제외하고 모든 업데이트를 활용할 수 있습니다.*

예 1.1. 프로파일 생성

다음 절차에서는 ***dns*** *또는* ***myhostname*** *데이터베이스가 아닌* ***/etc/hosts*** *파일의 호스트 이름에 대한 로컬 정적 테이블 조회만 참조하는* ***sssd*** *프로필을 기반으로 프로필을 생성하는 방법을 보여줍니다.*

1. *다음 행을 편집하여* ***/etc/nsswitch.conf*** *파일을 편집합니다.*

```
hosts:      files
```

2. ***/etc/nsswitch.conf에 대한 변경 사항을 제외하는 sssd 를 기반으로 사용자 지정 프로필을 생성합니다.***

```
# authselect create-profile user-profile -b sssd --symlink-meta --symlink-pam
```

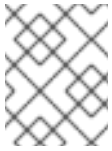
3. *프로필을 선택합니다.*

```
# authselect select custom/user-profile
```

4. *선택적으로 사용자 지정 프로필을 선택할 수 있는지 확인합니다.*

   - *선택한* ***sssd*** *프로필에 따라* ***/etc/pam.d/system-auth*** *파일을 생성*
   - ***/etc/nsswitch.conf에 구성을 변경하지 않은 상태로 둡니다.***

```
hosts:      files
```



참고

***authselect**를 실행하면 **sssd** 가 반대로 하면 **호스트 이름이 dns myhostname**이 됩니다.*

추가 리소스

- *authselect 사용*

1.5. 스크립트를 AUTHCONFIG 에서 AUTHSELECT로 변환

***ipa-client-install** 또는 **realm join** 을 사용하여 도메인에 가입하는 경우 스크립트에서 **authconfig** 호출을 안전하게 제거할 수 있습니다. 이를 수행할 수 없는 경우 각 **authconfig** 호출을 동일한 **authselect** 호출로 바꿉니다. 이렇게 하면 올바른 프로필과 적절한 옵션을 선택합니다. 또한 필요한 구성 파일을 편집합니다.*

- ***/etc/krb5.conf***
- ***/etc/sssd/sssd.conf** (sssd **프로필** 의 경우) 또는 **/etc/samba/smb.conf** (**winbind** 프로필의 경우)*

*authconfig 옵션과 authconfig 옵션 authselect 및 authconfig 옵션 동등한 Authselect 프로필 옵션과의 관계로 authconfig 옵션 의 **authselect** 동등한 기능이 표시됩니다 .*

표 1.1. authselect 프로필에 대한 authconfig 옵션의 관계

authconfig 옵션	Authselect 프로필
--enableldap --enableldapauth	**sssd**
--enablesssd --enablesssdauth	**sssd**
--enablekrb5	**sssd**
--enablewinbind --enablewinbindauth	**winbind**
--enablenis	**NIS**

표 1.2. authconfig 옵션과 동일한 Authselect 프로필 옵션

authconfig 옵션	Authselect 프로필 기능
--enablesmartcard	**with-smartcard**
--enablefingerprint	**with-fingerprint**
--enableecryptfs	**with-ecryptfs**
--enablemkhomedir	**with-mkhomedir**

authconfig 옵션	Authselect 프로필 기능
--enablefaillock	**with-faillock**
--enablepamaccess	**with-pamaccess**
--enablewinbindkrb5	**with-krb5**

*authconfig 명령에 해당하는 authselect 명령의 예는 authconfig에 대한 Kickstart 호출의 예를 **authselect** 호출에 대한 **Kickstart** 호출으로의 변환 예를 보여줍니다.*

표 1.3. authconfig 명령에 해당하는 authselect 명령의 예

authconfig 명령	Authselect 동급
authconfig --enableldap --enableldapauth --enablefaillock --updateall	**Authselect로 sssd with-faillock 선택**
authconfig --enablesssd --enablesssdauth --enablesmartcard --smartcardmodule=sssd --updateall	**Authselect를 사용하여 sssd(스마트카드) 선택**
authconfig --enableecryptfs --enablepamaccess --updateall	**-pamaccess를 사용하여 sssd with-ecryptfs 선택**
authconfig --enablewinbindauth --winbindjoin=Administrator --updateall	**영역 연결 -U Administrator --client-software=winbind WINBINDDOMAIN**

2장. SSSD 및 이점 이해

SSSD(System Security Services Daemon)는 원격 디렉터리 및 인증 메커니즘에 액세스하는 시스템 서비스입니다. 다음 장에서는 SSSD의 작동 방식, 사용 시 얻을 수 있는 이점, 구성 파일의 처리 방법, 구성할 수 있는 ID 및 인증 프로바이더에 대해 간략하게 설명합니다.

2.1. SSSD 작동 방식

SSSD(시스템 보안 서비스 데몬)는 원격 디렉터리 및 인증 메커니즘에 액세스할 수 있는 시스템 서비스입니다. 로컬 시스템인 SSSD 클라이언트를 외부 백엔드 시스템인 프로바이더에 연결할 수 있습니다.

예를 들면 다음과 같습니다.

- LDAP 디렉토리
- IdM(Identity Management) 도메인
- AD(Active Directory) 도메인
- Kerberos 영역

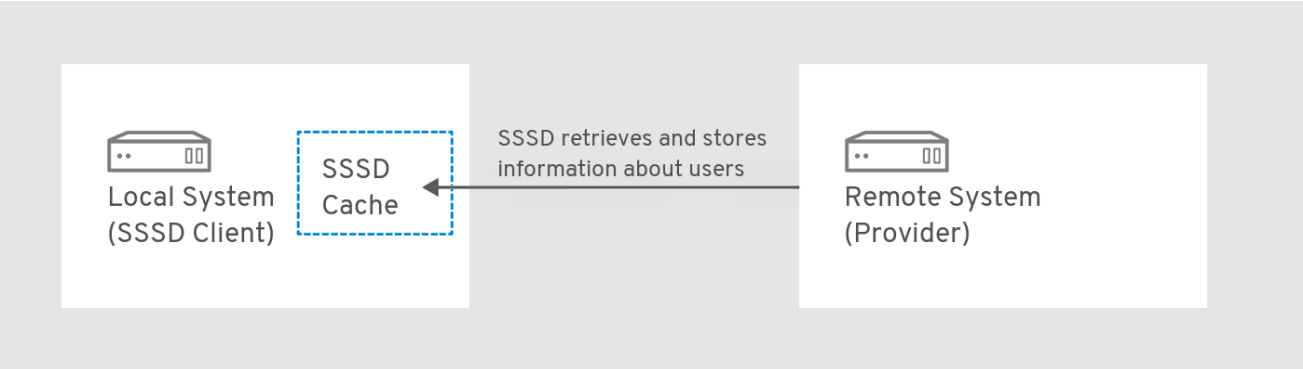
SSSD는 두 단계로 작동합니다.

1. 클라이언트를 원격 프로바이더에 연결하여 ID 및 인증 정보를 검색합니다.
2. 가져온 인증 정보를 사용하여 클라이언트에서 사용자 및 자격 증명의 로컬 캐시를 생성합니다.

그런 다음 로컬 시스템의 사용자는 원격 프로바이더에 저장된 사용자 계정을 사용하여 인증할 수 있습니다.

SSSD는 로컬 시스템에서 사용자 계정을 생성하지 않습니다. 그러나 IdM 사용자의 홈 디렉토리를 생성하도록 SSSD를 구성할 수 있습니다. 생성되면 사용자가 로그아웃하면 IdM 사용자 홈 디렉터리와 클라이언트의 내용이 삭제되지 않습니다.

그림 2.1. SSSD 작동 방식



SSSD는 NSS(Name Service Switch) 또는 PAM(Pluggable Authentication Modules)과 같은 여러 시스템 서비스에 대한 캐시를 제공할 수도 있습니다.

2.2. SSSD 사용의 이점

SSSD(System Security Services Daemon)를 사용하면 사용자 ID 검색 및 사용자 인증과 관련하여 다양한 이점이 있습니다.

오프라인 인증

SSSD는 선택적으로 원격 공급자로부터 검색한 사용자 ID 및 자격 증명의 캐시를 보관합니다. 이 설정에서 세션 시작 시 원격 프로바이더에 대해 한 번 이미 인증한 사용자인 경우 원격 프로바이더 또는 클라이언트가 오프라인 상태인 경우에도 리소스에 대해 성공적으로 인증할 수 있습니다.

단일 사용자 계정: 인증 프로세스의 일관성 향상

SSSD에서는 오프라인 인증을 위해 중앙 계정과 로컬 사용자 계정을 둘 다 유지할 필요가 없습니다. 조건은 다음과 같습니다.

- 특정 세션에서는 사용자가 한 번 이상 로그인해야 합니다. 사용자가 처음 로그인할 때 클라이언트를 원격 프로바이더에 연결해야 합니다.

- SSSD에서 캐싱을 활성화해야 합니다.
  SSSD를 사용하지 않으면 원격 사용자에게 종종 여러 사용자 계정이 있습니다. 예를 들어 VPN(가상 사설 네트워크)에 연결하려면 원격 사용자에게 로컬 시스템용 하나의 계정과 VPN 시스템용 계정이 있습니다. 이 시나리오에서는 사설 네트워크에서 먼저 인증하여 원격 서버에서 사용자를 가져오고 사용자 자격 증명을 로컬로 캐시해야 합니다.

  캐싱 및 오프라인 인증 덕분에 원격 사용자는 로컬 시스템에 인증하기만 하면 네트워크 리소스에 연결할 수 있습니다. SSSD에서 네트워크 자격 증명을 유지 관리합니다.

ID 및 인증 공급자의 로드 감소

정보를 요청할 때 클라이언트는 먼저 로컬 SSSD 캐시를 확인합니다. 캐시에서 정보를 사용할 수 없는 경우에만 SSSD에서 원격 공급자에게 연락합니다.

2.3. 클라이언트별로 여러 SSSD 구성 파일

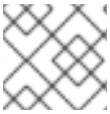
SSSD의 기본 설정 파일은 **/etc/sssd/sssd.conf** 입니다. SSSD는 이 파일 이외에 **/etc/sssd/ conf.d/ 디렉토리에 있는 모든 *.conf** 파일에서 해당 구성을 읽을 수 있습니다.

이 조합을 사용하면 모든 클라이언트에서 기본 **/etc/sssd/sssd.conf** 파일을 사용하고 추가 구성 파일에 설정을 추가하여 클라이언트별로 기능을 개별적으로 확장할 수 있습니다.

SSSD에서 구성 파일을 처리하는 방법
SSSD는 이 순서대로 구성 파일을 읽습니다.

1. 기본 **/etc/sssd/sssd.conf** 파일
2. 기타 ***.conf** 파일은 **/etc/sssd/conf.d/** 에 있는 알파벳순

동일한 매개 변수가 여러 구성 파일에 표시되면 SSSD는 마지막 read 매개 변수를 사용합니다.



참고

SSSD는 **conf.d** 디렉토리에서 숨겨진 파일(**.**로 시작하는 파일)을 읽지 않습니다.

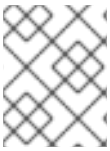
2.4. SSSD의 ID 및 인증 공급자

SSSD 클라이언트를 외부 ID 및 인증 공급자(예: LDAP 디렉터리, IdM(Identity Management), AD(Active Directory) 도메인 또는 Kerberos 영역)에 연결할 수 있습니다. 그러면 SSSD 클라이언트에서 SSSD 공급자를 사용하여 ID 및 원격 서비스에 액세스할 수 있습니다. 다른 ID 및 인증 공급자 또는 조합을 사용하도록 SSSD를 구성할 수 있습니다.

SSSD 도메인으로서의 ID 및 인증 공급자
ID 및 인증 공급자는 SSSD 구성 파일 **/etc/sssd/sssd.conf** 에서 도메인으로 구성됩니다. 프로바이더는 파일의 **[domain/name) 또는 [domain/default]** 섹션에 나열됩니다.

단일 도메인은 다음 공급자 중 하나로 구성할 수 있습니다.

- *UID 및 GID와 같은 사용자 정보를 제공하는 ID 프로바이더 입니다.*
  - ***/etc/sssd /sssd.conf 파일의도메인] 섹션에 있는 id_provider 옵션을 사용하여 도메인** 을 ID 프로바이더 로 지정합니다.*
- *인증 요청을 처리하는 인증 프로바이더 입니다.*
  - ***/etc/sssd /sssd.conf 의 도메인] 섹션에 있는 auth_provider 옵션을 사용하여 도메인** 을 인증 프로바이더 로 지정합니다.*
- *권한 부여 요청을 처리하는 액세스 제어 프로바이더 입니다.*
  - ***/etc/sssd /sssd.conf 의 도메인] 섹션에 있는 access_provider 옵션을 사용하여 도메인** 을 액세스 제어 프로바이더 로 지정합니다. 기본적으로 옵션은 **허용** 으로 설정되어 항상 모든 액세스를 허용합니다. 자세한 내용은 **sssd.conf**(5) 도움말 페이지를 참조하십시오.*
- *이러한 공급업체의 조합입니다(예: 모든 해당 작업이 단일 서버 내에서 수행되는 경우).*
  - *이 경우 **id_provider,auth_provider, access_provider** 옵션은 모두 **/etc/ssd /sssd.conf의 동일한 [domain/name) 또는 [domain/ default]** 섹션에 나열됩니다.*



참고

SSSD에 대해 여러 도메인을 구성할 수 있습니다. 하나 이상의 도메인을 구성해야 합니다. 그렇지 않으면 SSSD가 시작되지 않습니다.

프록시 공급자

프록시 프로바이더는 SSSD와 SSSD가 사용할 수 없는 리소스 간에 중간 릴레이로 작동합니다. 프록시 공급자를 사용하는 경우 SSSD는 프록시 서비스에 연결되고 프록시는 지정된 라이브러리를 로드합니다.

활성화하려면 프록시 공급자를 사용하도록 SSSD를 구성할 수 있습니다.

- *지문 스캐너와 같은 대체 인증 방법*
- *NIS와 같은 레거시 시스템*
- ***/etc/passwd** 파일에 ID 공급자 및 원격 인증 프로바이더로 정의된 로컬 시스템 계정(예: Kerberos)*

ID 및 인증 공급자의 사용 가능한 조합

다음과 같은 ID 및 인증 공급자 조합을 사용하도록 SSSD를 구성할 수 있습니다.

표 2.1. ID 및 인증 공급자의 사용 가능한 조합

ID 공급자	인증 공급자
IdM (Identity Management) [a]	IdM (Identity Management)
Active Directory	Active Directory
LDAP	LDAP

ID 공급자	인증 공급자
LDAP	Kerberos
proxy	proxy
proxy	LDAP
proxy	Kerberos
[a] LDAP 공급자 유형의 확장입니다.	

추가 리소스

- *authselect를 사용하여 사용자 인증 구성*
- *SSSD를 사용하여 도메인 정보 쿼리* [1]
- *SSSD를 사용하여 호스트에서 사용자 액세스 보고*

---

[1] ***sssctl*** *유틸리티를 사용하여 도메인 상태를 나열하고 확인하려면 AD(Active Directory) 포리스트와 신뢰 계약에 있는 IdM(Identity Management)에 호스트를 등록해야 합니다.*

3장. LDAP를 사용하도록 SSSD 구성 및 TLS 인증 필요

SSSD(System Security Services Daemon)는 RHEL 호스트에서 ID 데이터 검색 및 인증을 관리하는 데몬입니다. 시스템 관리자는 독립 실행형 LDAP 서버를 사용자 계정 데이터베이스로 사용하도록 호스트를 구성할 수 있습니다. 관리자는 LDAP 서버와의 연결을 TLS 인증서로 암호화해야 한다는 요구 사항을 지정할 수도 있습니다.

3.1. 암호화된 방식으로 LDAP에서 데이터를 검색하기 위해 SSSD를 사용하는 OPENLDAP 클라이언트

LDAP 개체의 인증 방법은 Kerberos 암호 또는 LDAP 암호일 수 있습니다. LDAP 개체의 인증 및 권한 부여에 대한 질문은 이 장에서 다루지 않습니다.

중요

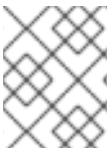
LDAP로 SSSD를 구성하는 절차는 SSSD 및 LDAP에 높은 수준의 전문 지식을 필요로 하는 복잡한 절차입니다. 대신 Active Directory 또는 Red Hat IdM(Identity Management)과 같은 통합되고 자동화된 솔루션을 사용하는 것이 좋습니다. IdM에 대한 자세한 내용은 ID 관리 계획을 참조하십시오.

3.2. LDAP를 사용하도록 SSSD 구성 및 TLS 인증 필요

RHEL(Red Hat Enterprise Linux) 시스템을 OpenLDAP 클라이언트로 구성하려면 다음 절차를 완료합니다.

다음 클라이언트 구성을 사용합니다.

- RHEL 시스템은 OpenLDAP 사용자 계정 데이터베이스에 저장된 사용자를 인증합니다.
- RHEL 시스템은 SSSD(System Security Services Daemon) 서비스를 사용하여 사용자 데이터를 검색합니다.
- RHEL 시스템은 TLS 암호화 연결을 통해 OpenLDAP 서버와 통신합니다.



참고

또는 이 절차를 사용하여 RHEL 시스템을 Red Hat Directory Server의 클라이언트로 구성할 수 있습니다.

사전 요구 사항

- OpenLDAP 서버가 사용자 정보로 설치 및 구성됩니다.
- LDAP 클라이언트로 구성 중인 호스트에 대한 루트 권한이 있습니다.
- LDAP 클라이언트로 구성하는 호스트에서 **/etc/sssd/sssd.conf** 파일이 생성되어 **ldap** 를 **autofs_provider** 및 **id_provider** 로 지정하도록 구성되었습니다.
- **core-dirsrv.ca.pem** 이라는 로컬 파일에 저장된 OpenLDAP 서버 인증서를 발급한 인증 기관에서 루트 CA 서명 인증서 체인의 PEM 형식의 사본이 있습니다.

절차

1. 필수 패키지를 설치합니다.

```
# dnf -y install openldap-clients sssd sssd-ldap oddjob-mkhomedir
```

2. *인증 공급자를* ***sssd*** *로 전환:*

```
# authselect select sssd with-mkhomedir
```

3. *OpenLDAP 서버의 SSL/TLS 인증서를 발급한 인증 기관에서 루트 CA 서명 인증서 체인이 포함된* ***core-dirsrv.ca.pem*** *파일을* ***/etc/openldap/certs*** *폴더에 복사합니다.*

```
# cp core-dirsrv.ca.pem /etc/openldap/certs
```

4. *LDAP 서버의 URL 및 접미사를* ***/etc/openldap/ldap.conf*** *파일에 추가합니다.*

```
URI ldap://ldap-server.example.com/
BASE dc=example,dc=com
```

5. ***/etc/openldap/ldap.conf*** *파일에서* ***TLS_CACERT*** *매개 변수를* ***/etc/openldap/certs/core-dirsrv.ca.pem에 가리키는 행을 추가합니다.***

```
# When no CA certificates are specified the Shared System Certificates
# are in use. In order to have these available along with the ones specified
# by TLS_CACERTDIR one has to include them explicitly:
TLS_CACERT /etc/openldap/certs/core-dirsrv.ca.pem
```

6. ***/etc/sssd/sssd.conf*** *파일에서 환경 값을* ***ldap_uri 및 ldap_ search_base*** *매개변수에 추가합니다.*

```
[domain/default]
id_provider = ldap
autofs_provider = ldap
auth_provider = ldap
chpass_provider = ldap
ldap_uri = ldap://ldap-server.example.com/
ldap_search_base = dc=example,dc=com
ldap_id_use_start_tls = True
cache_credentials = True
ldap_tls_cacertdir = /etc/openldap/certs
ldap_tls_reqcert = allow

[sssd]
services = nss, pam, autofs
domains = default

[nss]
homedir_substring = /home
…
```

7. ***/etc/sssd/sssd.conf*** *에서* ***[domain]*** *섹션의* ***ldap_tls_cacert*** *및* ***ldap_tls_reqcert*** *값을 수정하여 TLS 인증 요구 사항을 지정합니다.*

```
…
cache_credentials = True
ldap_tls_cacert = /etc/openldap/certs/core-dirsrv.ca.pem
```

```
ldap_tls_reqcert = hard
...
```

8. **/etc/sssd/sssd.conf 파일에 대한 권한을 변경합니다.**

```
# chmod 600 /etc/sssd/sssd.conf
```

9. SSSD 서비스와 **oddjobd** 데몬을 다시 시작하고 활성화합니다.

```
# systemctl restart sssd oddjobd
# systemctl enable sssd oddjobd
```

10. (선택 사항) LDAP 서버가 더 이상 사용되지 않는 TLS 1.0 또는 TLS 1.1 프로토콜을 사용하는 경우 클라이언트 시스템의 시스템 전체 암호화 정책을 LEGACY 수준으로 전환하여 RHEL 이 이러한 프로토콜을 사용하여 통신할 수 있도록 합니다.

```
# update-crypto-policies --set LEGACY
```

자세한 내용은 RHEL 8의 Strong crypto defaults 및 Red Hat Customer Portal의 약한 암호화 알고리즘 지식 베이스 문서 및 **update-crypto-policies(8)** 도움말 페이지를 참조하십시오.

검증 단계

- **id** 명령을 사용하고 LDAP 사용자를 지정하여 LDAP 서버에서 사용자 데이터를 검색할 수 있는지 확인합니다.

```
# id ldap_user
uid=17388(ldap_user) gid=45367(sysadmins)
groups=45367(sysadmins),25395(engineers),10(wheel),1202200000(admins)
```

시스템 관리자는 이제 **id** 명령을 사용하여 LDAP의 사용자를 쿼리할 수 있습니다. 명령은 올바른 사용자 ID와 그룹 멤버십을 반환합니다.

4장. ID 및 인증 공급자를 위한 추가 구성

SSSD(System Security Services Daemon)는 원격 디렉터리 및 인증 메커니즘에 액세스하는 시스템 서비스입니다. SSSD의 기본 설정 파일은 **/etc/sssd/sssd.conf** 입니다. 다음 장에서는 **/etc/sssd/sssd.conf 파일을 다음과 같이 수정하여 SSSD 서비스 및 도메인을 구성하는 방법을 간략하게 설명합니다.**

- SSSD에서 전체 사용자 이름을 해석하고 출력하는 방법을 조정하여 오프라인 인증을 활성화합니다.
- LDAP 액세스 필터를 적용하기 위해 DNS 서비스 검색, 간단한 액세스 공급자 규칙 및 SSSD를 구성합니다.

4.1. SSSD에서 전체 사용자 이름을 해석하는 방법 조정

SSSD는 전체 사용자 이름 문자열을 사용자 이름 및 도메인 구성 요소로 구문 분석합니다. 기본적으로 SSSD는 Python 구문의 다음 정규 표현식에 따라 **usern_ame@domain_name** 형식의 전체 사용자 이름을 해석합니다.

```
(?P<name>[^@]+)@?(?P<domain>[^@]*$)
```

참고

ID 관리 및 Active Directory 공급자의 경우 기본 사용자 이름 형식은 **user_name@domain_name** 또는 **NetBIOS_name\user_name** 입니다.

SSSD에서 **re_expression** 옵션을 **/etc/sssd/sssd.conf** 파일에 추가하고 사용자 지정 정규식을 정의하여 전체 사용자 이름을 해석하는 방법을 조정할 수 있습니다.

- 정규 표현식을 전역적으로 정의하려면 *전역적으로 정규 표현식 정의와 같이* **sssd.conf** *파일의* **[sssd]** *섹션에 정규 표현식* 을 추가합니다.
- 특정 도메인의 정규 표현식을 정의하려면 정규 표현식을 *특정 도메인 예에 표시된 대로* **sssd.conf** *파일의 해당 도메인 섹션(예:* **[domain/LDAP]**)에 추가합니다.

사전 요구 사항

- **루트** 액세스

절차

1. **/etc/sssd/sssd.conf** 파일을 엽니다.
2. **re_expression** 옵션을 사용하여 사용자 지정 정규식을 정의합니다.

예 4.1. 전역 정규식 정의

모든 도메인에 대해 정규 표현식을 전역적으로 정의하려면 **sssd.conf** 파일의 **[sssd]** 섹션에 **re_expression** 을 추가합니다.

다음 전역 표현식을 사용하여 **도메인\\username 또는 domain@username** 형식으로 사용자 이름을 정의할 수 있습니다:

```
[sssd]
[... file truncated ...]
re_expression = (?P<domain>[^\\]*?)\\?(?P<name>[^\\]+$)
```

예 4.2. 특정 도메인의 정규 표현식 정의

특정 도메인에 대한 정규 표현식을 개별적으로 정의하려면 **sssd.conf** 파일의 해당 도메인 섹션에 **re_expression** 을 추가합니다.

다음 전역 표현식을 사용하여 LDAP **도메인의 domain\\username 또는 domain@username** 형식으로 사용자 이름을 정의할 수 있습니다.

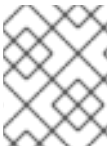
```
[domain/LDAP]
[... file truncated ...]
re_expression = (?P<domain>[^\\]*?)\\?(?P<name>[^\\]+$)
```

자세한 내용은 **sssd.conf(5) 도움말** 페이지의 **SPECIAL SECTIONS** 및 **DOMAIN SECTIONS** 부분의 **re_expression** 에 대한 설명을 참조하십시오.

4.2. SSSD가 전체 사용자 이름을 인쇄하는 방법 조정

/etc/sssd/sssd.conf 파일에서 **use_fully_qualified_names** 옵션을 활성화하면 SSSD는 기본적으로 다음 확장을 기반으로 **이름@domain** 형식의 전체 사용자 이름을 출력합니다.

```
%1$s@%2$s
```



참고

신뢰할 수 있는 도메인에 대해 **use_fully_qualified_names** 를 설정하지 않거나 명시적으로 **false** 로 설정된 경우 도메인 구성 요소 없이 사용자 이름만 출력합니다.

/etc/sssd/sssd.conf 파일에 **full_name_format** 옵션을 추가하고 사용자 지정 확장을 정의하여 SSSD에서 전체 사용자 이름을 출력하는 형식을 조정할 수 있습니다.

사전 요구 사항

- **루트** 액세스

절차

1. **루트로 /etc/sssd/sssd.conf** 파일을 엽니다.
2. **full_name_format** 옵션을 사용하여 전체 사용자 이름 형식의 사용자 지정 확장을 정의합니다.

   예 4.3. 전역적으로 사용자 이름 인쇄 형식을 정의합니다.

   모든 도메인의 전역적으로 확장을 정의하려면 **sssd.conf** 의 **[sssd]** 섹션에 **full_name_format** 을 추가합니다.

```
[sssd]
[... file truncated ...]
full_name_format = %1$s
```

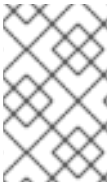
예 4.4. 특정 도메인의 사용자 이름 인쇄 형식 정의

특정 도메인에 대한 확장을 개별적으로 정의하려면 **full_name_format** 을 **sssd.conf** 의 해당 도메인 섹션에 추가합니다.

예를 들어, AD(Active Directory) 도메인의 확장을 구성하려면 다음을 사용합니다.

```
[domain/AD]
[... file truncated ...]
full_name_format = %3$s
```

자세한 내용은 **SPECIAL SECTIONS 및 sssd.conf(5) 도움말 페이지의 DOMAIN SECTIONS** 부분에 있는 **full_name_format** 에 대한 설명을 참조하십시오.



참고

SSSD는 일부 이름 구성에서 이름의 도메인 구성 요소를 제거하여 인증 오류를 일으킬 수 있습니다. **full_name_format** 을 비표준 값으로 설정하면 표준 형식으로 변경하라는 경고 메시지가 표시됩니다.

4.3. 오프라인 인증 활성화

SSSD는 기본적으로 사용자 자격 증명을 캐시하지 않습니다. 인증 요청을 처리할 때 SSSD는 항상 ID 공급자에게 연락합니다. 공급자를 사용할 수 없는 경우 사용자 인증이 실패합니다.

중요

SSSD는 일반 텍스트로 암호를 캐시하지 않습니다. 암호의 해시만 저장합니다.

ID 공급자를 사용할 수 없는 경우에도 사용자가 인증할 수 있도록 **/etc/sssd/sssd.conf** 파일에서 **cache_credentials** 를 **true** 로 설정하여 인증 정보 캐싱을 활성화할 수 있습니다.

사전 요구 사항

- **루트** 액세스

절차

1. **/etc/sssd/sssd.conf** 파일을 엽니다.
2. domain 섹션에서 **cache_credentials = true** 설정을 추가합니다.

```
[domain/your-domain-name]
cache_credentials = true
```

3. *선택 사항이지만 권장 사항: ID 공급자를 사용할 수 없는 경우 SSSD에서 오프라인 인증을 허용하는 기간에 대한 시간 제한을 구성합니다.*

   a. *SSSD에서 작동하도록 PAM 서비스를 구성합니다.*
   자세한 내용은 authselect를 사용하여 사용자 인증 구성 을 참조하십시오.

   b. ***offline_credentials_expiration** 옵션을 사용하여 시간 제한을 지정합니다.*
   제한은 일 단위로 설정됩니다.

   예를 들어 사용자가 마지막으로 로그인한 후 3일 동안 오프라인으로 인증할 수 있도록 지정하려면 다음을 사용합니다.

```
[pam]
offline_credentials_expiration = 3
```

추가 리소스

- ***sssd.conf(5)** 도움말 페이지*

4.4. DNS 서비스 검색 구성

*DNS 서비스 검색을 사용하면 애플리케이션에서 특정 유형의 특정 서비스에 대해 지정된 도메인의 SRV 레코드를 확인한 다음 필수 유형과 일치하는 서버를 모두 반환할 수 있습니다. ID 또는 인증 서버가 **/etc/sssd/sssd.conf** 파일에 명시적으로 정의되어 있지 않은 경우 SSSD는 DNS 서비스 검색을 사용하여 서버를 동적으로 검색할 수 있습니다.*

*예를 들어 **sssd.conf** 에 **id_provider = ldap** 설정이 포함되지만 **ldap_uri** 옵션이 호스트 이름 또는 IP 주소를 지정하지 않는 경우 SSSD는 DNS 서비스 검색을 사용하여 서버를 동적으로 검색합니다.*

참고

SSSD는 기본 서버만 동적으로 백업 서버를 검색할 수 없습니다.

사전 요구 사항

- ***루트** 액세스*

절차

1. ***/etc/sssd/sssd.conf** 파일을 엽니다.*

2. *기본 서버 값을 **_srv_** 로 설정합니다.*
   *LDAP 공급자의 경우 기본 서버는 **ldap_uri** 옵션을 사용하여 설정됩니다.*

```
[domain/your-domain-name]
id_provider = ldap
ldap_uri = _srv_
```

3. *서비스 유형을 설정하여 암호 변경 공급자의 서비스 검색을 활성화합니다.*

```
[domain/your-domain-name]
id_provider = ldap
ldap_uri = _srv_
```

```
chpass_provider = ldap
ldap_chpass_dns_service_name = ldap
```

4. *선택 사항: 기본적으로 서비스 검색에서는 시스템 호스트 이름의 도메인 부분을 도메인 이름으로 사용합니다. 다른 DNS 도메인을 사용하려면* ***dns_discovery_domain*** *옵션을 사용하여 도메인 이름을 지정합니다.*

5. *선택 사항: 기본적으로 서비스 검색은 LDAP 서비스 유형을 검사합니다. 다른 서비스 유형을 사용하려면* ***ldap_dns_service_name*** *옵션을 사용하여 유형을 지정합니다.*

6. *선택 사항: 기본적으로 SSSD는 IPv4 주소를 조회합니다. 시도에 실패하면 SSSD에서 IPv6 주소를 조회하려고 시도합니다. 이 동작을 사용자 지정하려면* ***lookup_family_order*** *옵션을 사용합니다.*

7. *서비스 검색을 사용할 모든 서비스에 대해 DNS 레코드를 DNS 서버에 추가합니다.*

```
_service._protocol._domain TTL priority weight port host_name
```

추가 리소스

- *DNS 서비스 검색의 RFC 2782*
- ***sssd.conf(5) 도움말*** *페이지*

4.5. 간단한 액세스 공급자 규칙 구성

간단한 *액세스 프로바이더는 사용자 이름 또는 그룹 목록에 따라 액세스를 허용하거나 거부합니다. 이를 통해 특정 시스템에 대한 액세스를 제한할 수 있습니다.*

예를 들어 ***간단한*** *액세스 공급자를 사용하여 특정 사용자 또는 그룹에 대한 액세스를 제한할 수 있습니다. 다른 사용자 또는 그룹은 구성된 인증 프로바이더에 대해 성공적으로 인증하더라도 로그인할 수 없습니다.*

사전 요구 사항

- ***루트*** *액세스*

절차

1. ***/etc/sssd/sssd.conf*** *파일을 엽니다.*

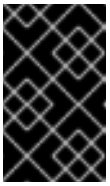
2. ***access_provider*** *옵션을* ***simple*** *로 설정합니다.*

```
[domain/your-domain-name]
access_provider = simple
```

3. *사용자에 대한 액세스 제어 규칙을 정의합니다.*

   a. *사용자에 대한 액세스를 허용하려면* ***simple_allow_users*** *옵션을 사용합니다.*

   b. *사용자에 대한 액세스를 거부하려면* ***simple_deny_users*** *옵션을 사용합니다.*



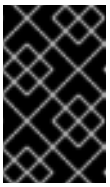
중요

특정 사용자에 대한 액세스를 거부하는 경우 다른 모든 사용자에 대한 액세스를 자동으로 허용합니다. 특정 사용자에 대한 액세스 허용은 거부하는 것보다 안전한 것으로 간주됩니다.

4. *그룹에 대한 액세스 제어 규칙을 정의합니다. 다음 중 하나를 선택합니다.*

   a. *그룹에 대한 액세스를 허용하려면* ***simple_allow_groups*** *옵션을 사용합니다.*

   b. *그룹에 대한 액세스를 거부하려면* ***simple_deny_groups*** *옵션을 사용합니다.*



중요

특정 그룹에 대한 액세스를 거부하면 다른 모든 그룹에 대한 액세스를 자동으로 허용합니다. 특정 그룹에 대한 액세스 허용은 거부하는 것보다 안전한 것으로 간주됩니다.

> ***예 4.5. 특정 사용자 및 그룹에 대한 액세스 허용***
>
> *다음 예제에서는 user1, user2 및 group1의 액세스를 허용하는 동시에 다른 모든 사용자에 대한 액세스를 거부합니다.*
>
> ```
> [domain/your-domain-name]
> access_provider = simple
> simple_allow_users = user1, user2
> simple_allow_groups = group1
> ```

중요

거부 목록을 비워 두면 모든 사용자에 대한 액세스를 허용할 수 있습니다.

추가 리소스

- ***sssd-simple5*** *도움말 페이지*

4.6. LDAP 액세스 필터를 적용하도록 SSSD 구성

access_provider *옵션이* ***/etc/sssd/sssd.conf*** *에 설정된 경우 SSSD는 지정된 액세스 공급자를 사용하여 시스템에 대한 액세스 권한이 부여된 사용자를 평가합니다. 사용 중인 액세스 공급자가 LDAP 공급자 유형의 확장 기능인 경우 시스템에 대한 액세스를 허용하기 위해 사용자가 일치해야 하는 LDAP 액세스 제어 필터를 지정할 수도 있습니다.*

예를 들어 AD(Active Directory) 서버를 액세스 공급자로 사용하는 경우 지정된 AD 사용자로만 Linux 시스템에 대한 액세스를 제한할 수 있습니다. 지정된 필터와 일치하지 않는 다른 모든 사용자에게는 액세스 권한이 거부됩니다.

참고

액세스 필터는 LDAP 사용자 항목에만 적용됩니다. 따라서 중첩된 그룹에서 이러한 유형의 액세스 제어를 사용하는 것이 작동하지 않을 수 있습니다. 중첩 그룹에 액세스 제어를 적용하려면 ***간단한*** *액세스 공급자 규칙 구성 을 참조하십시오.*

중요

오프라인 캐싱을 사용할 때 SSSD는 사용자의 최신 온라인 로그인 시도가 성공했는지 확인합니다. 가장 최근의 온라인 로그인 중에 성공적으로 로그인한 사용자는 액세스 필터와 일치하지 않더라도 오프라인에 로그인할 수 있습니다.

사전 요구 사항

- *루트 액세스*

절차

1. ***/etc/sssd/sssd.conf** 파일을 엽니다.*
2. ***[domain]** 섹션에서 LDAP 액세스 제어 필터를 지정합니다.*
   - *LDAP 액세스 공급자의 경우 **ldap_access_filter** 옵션을 사용합니다. 자세한 내용은 **sssd-ldap(5) 도움말** 페이지를 참조하십시오.*
   - *AD 액세스 공급자의 경우 **ad_access_filter** 옵션을 사용합니다. 자세한 내용은 **sssd-ad(5) 도움말** 페이지를 참조하십시오.*

     예 4.6. 특정 AD 사용자에게 액세스 허용

     *예를 들어 **admins** 사용자 그룹에 속하고 **unixHomeDirectory** 특성이 설정된 AD 사용자에게만 액세스를 허용하려면 다음을 사용합니다.*

     ```
     [domain/your-AD-domain-name]
     access provider = ad
     [... file truncated ...]
     ad_access_filter = (&(memberOf=cn=admins,ou=groups,dc=example,dc=com)
     (unixHomeDirectory=*))
     ```

*SSSD는 항목의 **authorizedService** 또는 **host** 속성에서 결과를 확인할 수도 있습니다. 실제로 사용자 항목 및 구성에 따라 **MDASH LDAP** 필터, 인증 서비스 및 호스트 **MDASH** 옵션을 모두 평가할 수 있습니다. **ldap_access_order** 매개 변수는 사용할 모든 액세스 제어 메서드를 나열합니다.*

```
[domain/example.com]
access_provider = ldap
ldap_access_filter = memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com
ldap_access_order = filter, host, authorized_service
```

추가 리소스

- ***sssd-ldap(5) 도움말 페이지***

5장. SSSD 클라이언트 측 보기

SSSD는 sss_override 유틸리티를 제공하여 로컬 시스템에 고유한 POSIX 사용자 또는 그룹 속성의 값을 표시하는 로컬 보기를 만들 수 있습니다. ipa 를 제외한 모든 id_provider 값에 대한 재정의를 구성할 수 있습니다.

ipa 프로바이더를 사용하는 경우 IPA에서 ID 뷰를 중앙에서 정의합니다. 자세한 내용은 ID 보기 섹션을 참조하십시오.

SSSD 성능에 미치는 잠재적인 부정적인 영향에 대한 자세한 내용은 SSSD 성능에 미치는 잠재적 영향 섹션을 참조하십시오.

5.1. LDAP USERNAME 속성 덮어쓰기

관리자는 LDAP의 계정을 사용하도록 기존 호스트를 구성할 수 있습니다. 그러나 LDAP에서 사용자(name, UID, GID, 홈 디렉터리, 쉘)의 값은 로컬 시스템의 값과 다릅니다. 다음 절차에 따라 보조 사용자 이름을 정의하여 LDAP 사용자 이름 속성 을 재정의할 수 있습니다.

사전 요구 사항

- 루트 액세스
- 설치된 sssd-tools

절차

1. 사용자의 현재 정보를 표시합니다.

```
# id username
```

username 을 사용자 이름으로 바꿉니다.

2. 보조 사용자 이름을 추가합니다:

```
# sss_override user-add username -n secondary-username
```

username을 사용자 이름으로 바꾸고 secondary-username 을 새 사용자 이름으로 교체합니다.

3. *sss_override user-add 명령을 사용하여 첫 번째 재정의를 생성한 후 SSSD를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

검증 단계

- *새 사용자 이름이 추가되었는지 확인합니다.*

```
# id secondary-username
```

- *선택사항입니다. 사용자의 재정의를 표시합니다.*

```
# sss_override user-show user-name
user@ldap.example.com:secondary-username::::::
```

예 5.1. 보조 사용자 이름 정의

사용자 sjones 에 대한 보조 사용자 이름을 추가하려면 다음을 수행합니다.

1. *sjones: 사용자의 현재 정보를 표시합니다.*

```
# id sjones
uid=1001(sjones) gid=6003 groups=6003,10(wheel)
```

2. *보조 사용자 이름을 추가합니다:*

```
# sss_override user-add sjones -n sarah
```

3. *새 사용자 이름이 추가되고 사용자의 재정의가 올바르게 표시되는지 확인합니다.*

```
# id sarah
uid=1001(sjones) gid=6003(sjones) groups=6003(sjones),10(wheel)
```

```
# sss_override user-show sjones
user@ldap.example.com:sarah::::::
```

추가 리소스

- **sss_override** 도움말 페이지

5.2. LDAP UID 속성 덮어쓰기

관리자는 **LDAP**의 계정을 사용하도록 기존 호스트를 구성할 수 있습니다. 그러나 **LDAP**에서 사용자 **(name, UID, GID,** 홈 디렉터리, 쉘)의 값은 로컬 시스템의 값과 다릅니다. 다음 절차에 따라 다른 **UID**를 정의하여 **LDAP UID** 속성을 재정의할 수 있습니다.

사전 요구 사항

- 루트 액세스
- 설치된 **sssd-tools**

절차

1. 사용자의 현재 **UID**를 표시합니다.

```
# id -u user-name
```

user-name 을 사용자 이름으로 변경합니다.

2. 사용자 계정의 **UID**를 재정의합니다.

```
# sss_override user-add user-name -u new-UID
```

user-name 을 사용자 이름으로 바꾸고 **new-UID** 를 새 **UID** 번호로 바꿉니다.

3. 메모리 내 캐시를 만료합니다.

```
# sss_cache --users
```

4. ***sss_override user-add** 명령을 사용하여 첫 번째 재정의를 생성한 후 **SSSD**를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

검증 단계

- *새 **UID**가 적용되었는지 확인합니다.*

```
# id -u user-name
```

- *선택사항입니다. 사용자의 재정의를 표시합니다.*

```
# sss_override user-show user-name
user@ldap.example.com::new-UID:::::
```

*예 **5.2.** 사용자의 **UID** 덮어쓰기*

*사용자의 **UID**를 **UID 6666** 로 재정의하려면 다음을 수행합니다.*

1. ***userarah**의 현재 **UID**를 표시하십시오:*

```
# id -u sarah
1001
```

2. ***UID 6666** 으로 사용자 계정의 **UID**를 재정의합니다:*

```
# sss_override user-add sarah -u 6666
```

3. *메모리 내 캐시를 수동으로 만료합니다.*

```
# sss_cache --users
```

4. ***SSSD**를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

5. 새 UID가 적용되고 사용자 재정의가 올바르게 표시되는지 확인합니다.

```
# id sarah
6666

# sss_override user-show sarah
user@ldap.example.com::6666:::::
```

추가 리소스

- sss_override 도움말 페이지

5.3. LDAP GID 속성 덮어쓰기

관리자는 LDAP의 계정을 사용하도록 기존 호스트를 구성할 수 있습니다. 그러나 LDAP에서 사용자(name, UID, GID, 홈 디렉터리, 쉘)의 값은 로컬 시스템의 값과 다릅니다. 다음 절차에 따라 다른 GID를 정의하여 LDAP GID 속성을 재정의할 수 있습니다.

사전 요구 사항

- 루트 액세스
- 설치된 sssd-tools

절차

1. 사용자의 현재 GID를 표시합니다.

```
# id -g user-name
```

user-name 을 사용자 이름으로 변경합니다.

2. 사용자 계정의 GID를 재정의합니다.

```
# sss_override user-add user-name -u new-GID
```

user-name 을 사용자 이름으로 바꾸고 new-GID 를 새 GID 번호로 바꿉니다.

3. *메모리 내 캐시를 만료합니다.*

```
# sss_cache --users
```

4. *sss_override user-add 명령을 사용하여 첫 번째 재정의를 생성한 후 SSSD를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

검증 단계

- *새 GID가 적용되었는지 확인합니다.*

```
# id -g user-name
```

- *선택사항입니다. 사용자의 재정의를 표시합니다.*

```
# sss_override user-show user-name
user@ldap.example.com:::6666::::
```

예 5.3. 사용자의 GID 덮어쓰기

GID 6666 으로 사용자의 GID를 재정의하려면 다음을 실행합니다:

1. *userarah의 현재 GID를 표시합니다.*

```
# id -g sarah
6003
```

2. *GID 6666 으로 사용자 계정의 GID를 재정의합니다:*

```
# sss_override user-add sarah -g 6666
```

3. *메모리 내 캐시를 수동으로 만료합니다.*

```
# sss_cache --users
```

4. *첫 번째 재정의인 경우 **SSSD**를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

5. *새 **GID**가 적용되고 사용자의 재정의가 올바르게 표시되는지 확인합니다.*

```
# id -g sarah
6666

# sss_override user-show sarah
user@ldap.example.com::6666:::::
```

추가 리소스

- ***sss_override** 도움말 페이지*

5.4. LDAP 홈 디렉터리 속성 덮어쓰기

*관리자는 **LDAP**의 계정을 사용하도록 기존 호스트를 구성할 수 있습니다. 그러나 **LDAP**에서 사용자 **(name, UID, GID,** 홈 디렉터리, 쉘**)**의 값은 로컬 시스템의 값과 다릅니다. 다음 절차에 따라 다른 홈 디렉터리를 정의하여 **LDAP** 홈 디렉터리 속성을 재정의할 수 있습니다.*

사전 요구 사항

- *루트 액세스*
- *설치된 **sssd-tools***

절차

1. *사용자의 현재 홈 디렉토리를 표시합니다.*

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:/bin/bash
```

user-name 을 사용자 이름으로 변경합니다.

2. *사용자의 홈 디렉터리를 재정의합니다.*

```
# sss_override user-add user-name -h new-home-directory
```

user-name 을 사용자 이름으로 바꾸고 new-home-directory 를 새 홈 디렉터리로 교체합니다.

3. ***SSSD**를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

검증 단계

- *새 홈 디렉터리가 정의되어 있는지 확인합니다.*

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/new-home-directory:/bin/bash
```

- *선택사항입니다. 사용자의 재정의를 표시합니다.*

```
# sss_override user-show user-name
user@ldap.example.com:::::::new-home-directory::
```

예 5.4. 사용자의 홈 디렉터리 덮어쓰기

admin 을 사용하여 사용자의 홈 디렉토리를 재정의하려면 다음을 수행합니다.

1. *user sarah 의 현재 홈 디렉토리 표시 :*

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/bin/bash
```

2. *사용자의 홈 디렉토리를 새 홈 디렉토리 **admin** 으로 재정의합니다:*

```
# sss_override user-add sarah -h admin
```

3. ***SSSD**를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

4. *새 홈 디렉터리가 정의되었으며 사용자 가 올바르게 표시되는지 확인합니다.*

```
# getent passwd sarah
sarah:x:1001:6003::admin:/bin/bash

# sss_override user-show user-name
user@ldap.example.com:::::::admin::
```

추가 리소스

- ***sss_override** 도움말 페이지*

5.5. LDAP 쉘 속성 덮어쓰기

*관리자는 **LDAP**의 계정을 사용하도록 기존 호스트를 구성할 수 있습니다. 그러나 **LDAP**에서 사용자 **(name, UID, GID,** 홈 디렉터리, 쉘**)**의 값은 로컬 시스템의 값과 다릅니다. 다음 절차에 따라 다른 쉘을 정의하여 **LDAP** 쉘 속성을 재정의할 수 있습니다.*

사전 요구 사항

- *루트 액세스*
- *설치된 **sssd-tools***

절차

1. *사용자의 현재 쉘을 표시합니다.*

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:/bin/bash
```

user-name 을 사용자 이름으로 변경합니다.

2. 사용자의 쉘을 재정의합니다.

```
# sss_override user-add user-name -s new-shell
```

user-name 을 사용자 이름으로 바꾸고 **new-shell** 을 새 쉘로 바꿉니다.

3. **SSSD**를 재시작하여 변경 사항을 적용합니다.

```
# systemctl restart sssd
```

검증 단계

- 새 쉘이 정의되어 있는지 확인합니다.

```
# getent passwd user-name
user-name:x:XXXX:XXXX::/home/home-directory:new-shell
```

- 선택사항입니다. 사용자의 재정의를 표시합니다.

```
# sss_override user-show user-name
user@ldap.example.com::::::new-shell:
```

예 5.5. 사용자의 쉘 덮어쓰기

user sarah 쉘을 **/bin/bash**에서 **sbin/ nologin**으로 변경하려면:

1. **userarah**의 현재 쉘을 표시합니다 :

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/bin/bash
```

2. *새로운 /sbin/nologin 쉘 을 사용하여 usersarah 쉘을 재정의합니다.*

```
# sss_override user-add sarah -s /sbin/nologin
```

3. *SSSD를 재시작하여 변경 사항을 적용합니다.*

```
# systemctl restart sssd
```

4. *새 쉘이 정의되고 사용자 가 올바르게 표시되는지 확인합니다.*

```
# getent passwd sarah
sarah:x:1001:6003::sarah:/sbin/nologin

# sss_override user-show user-name
user@ldap.example.com::::::/sbin/nologin:
```

추가 리소스

- *sss_override 도움말 페이지*

5.6. 호스트의 덮어쓰기 나열

관리자는 호스트의 모든 사용자 및 그룹 재정의를 나열하여 올바른 특성이 재정의되었는지 확인할 수 있습니다.

사전 요구 사항

- *루트 액세스*
- *설치된 sssd-tools*

절차

- *모든 사용자 재정의를 나열합니다.*

```
# sss_override user-find
```

```
user1@ldap.example.com::8000::::/bin/zsh:
user2@ldap.example.com::8001::::/bin/bash:
...
```

- 모든 그룹 덮어쓰기를 나열합니다.

```
# sss_override group-find
group1@ldap.example.com::7000
group2@ldap.example.com::7001
...
```

5.7. 로컬 덮어쓰기 제거

글로벌 **LDAP** 디렉터리에 정의된 로컬 오버라이드를 제거하려면 다음 절차를 사용하십시오.

사전 요구 사항

- 루트 액세스
- 설치된 **sssd-tools**

절차

- 사용자 계정의 재정의를 제거하려면 다음을 사용합니다.

```
# sss_override user-del user-name
```

user-name 을 사용자 이름으로 변경합니다. 변경 사항이 즉시 적용됩니다.

- 그룹의 재정의를 제거하려면 다음을 사용합니다.

```
# sss_override group-del group-name
```

- **sss _override user-del** 또는 **sss_override group-del** 명령을 사용하여 첫 번째 재정의를 제거한 후 **SSSD**를 재시작하여 변경 사항을 적용합니다.

```
# systemctl restart sssd
```

사용자 또는 그룹의 재정의를 제거하면 이 오브젝트에 대한 모든 재정의가 제거됩니다.

5.8. 로컬 보기 내보내기 및 가져오기

로컬 재정의는 로컬 **SSSD** 캐시에 저장됩니다. 이 캐시에서 파일로 사용자 및 그룹 재정의를 내보내 백업을 생성할 수 있습니다. 이렇게 하면 캐시가 지워지더라도 나중에 구성을 복원할 수 있습니다.

사전 요구 사항

- 루트 액세스
- 설치된 **sssd-tools**

절차

- 사용자 및 그룹 보기를 백업하려면 다음을 사용합니다.

```
# sss_override user-export /var/lib/sss/backup/sssd_user_overrides.bak
# sss_override group-export /var/lib/sss/backup/sssd_group_overrides.bak
```

- 사용자 및 그룹 보기를 복원하려면 다음을 사용합니다.

```
# sss_override user-import /var/lib/sss/backup/sssd_user_overrides.bak
# sss_override group-import /var/lib/sss/backup/sssd_group_overrides.bak
```

6장. 인증 공급자로 AD를 사용하도록 RHEL 호스트 구성

시스템 관리자는 호스트를 AD에 연결하지 않고 RHEL(Red Hat Enterprise Linux) 호스트의 인증 공급자로 AD(Active Directory)를 사용할 수 있습니다.

예를 들면 다음과 같은 경우 이 작업을 수행할 수 있습니다.

- AD 관리자에게 호스트 활성화 및 비활성화에 대한 제어 권한을 부여하지는 않습니다.
- 회사 PC가 될 수 있는 호스트는 회사의 한 명의 사용자만 사용해야 합니다.

중요

이 접근 방식이 선호되는 드물지만 이 절차를 구현하십시오.

시스템을 AD 또는 Red Hat IdM(Identity Management)에 완전히 참여시키는 것이 좋습니다. RHEL 호스트를 도메인에 연결하면 설정을 쉽게 관리할 수 있습니다. 클라이언트를 AD에 직접 연결하는 것과 관련된 클라이언트 액세스 라이센스에 관심이 있는 경우 AD와의 신뢰 계약에 있는 IdM 서버를 활용하는 것이 좋습니다. IdM-AD 신뢰에 대한 자세한 내용은 IdM과 AD 사이의 교차 신뢰 계획 및 IdM과 AD 간의 신뢰 설치를 참조하십시오.

이 절차를 통해 AD_user 라는 사용자가 example.com 도메인의 AD(Active Directory) 사용자 데이터베이스에 설정된 암호를 사용하여 rhel_host 시스템에 로그인할 수 있습니다. 이 예제에서 EXAMPLE.COM Kerberos 영역은 example.com 도메인에 해당합니다.

사전 요구 사항

- rhel_host 에 대한 루트 액세스 권한이 있습니다.
- AD_user 사용자 계정은 example.com 도메인에 있습니다.
- Kerberos 영역은 EXAMPLE.COM 입니다.

- *rhel_host 가 realm join 명령을 사용하여 AD에 가입되지 않았습니다.*

절차

1. *암호를 할당하지 않고 AD_user 사용자 계정을 로컬로 생성합니다.*

```
# useradd AD_user
```

2. *편집을 위해 /etc/nsswitch.conf 파일을 열고 다음 행을 포함하는지 확인합니다.*

```
passwd:     sss files systemd
group:      sss files systemd
shadow:     files sss
```

3. *편집을 위해 /etc/krb5.conf 파일을 열고 다음 섹션 및 항목이 포함되어 있는지 확인합니다.*

```
# To opt out of the system crypto-policies configuration of krb5, remove the
# symlink at /etc/krb5.conf.d/crypto-policies which will not be recreated.
includedir /etc/krb5.conf.d/

[logging]
    default = FILE:/var/log/krb5libs.log
    kdc = FILE:/var/log/krb5kdc.log
    admin_server = FILE:/var/log/kadmind.log

[libdefaults]
    dns_lookup_realm = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true
    rdns = false
    pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
    spake_preauth_groups = edwards25519
    default_realm = EXAMPLE.COM
    default_ccache_name = KEYRING:persistent:%{uid}

[realms]
 EXAMPLE.COM = {
     kdc = ad.example.com
     admin_server = ad.example.com
 }

[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

4.

/etc/sssd/sssd.conf 파일을 생성하고 여기에 다음 섹션과 행을 삽입합니다.

```
[sssd]
   services = nss, pam
   domains = EXAMPLE.COM

[domain/EXAMPLE.COM]
   id_provider = files
   auth_provider = krb5
   krb5_realm = EXAMPLE.COM
   krb5_server = ad.example.com
```

5.

/etc/sssd/sssd.conf 파일에 대한 권한을 변경합니다.

```
# chmod 600 /etc/sssd/sssd.conf
```

6.

SSSD(보안 시스템 서비스 데몬)를 시작합니다.

```
# systemctl start sssd
```

7.

SSSD를 활성화합니다.

```
# systemctl enable sssd
```

8.

/etc/pam.d/system-auth 파일을 열고 다음 섹션과 행을 포함하도록 수정합니다.

```
# Generated by authselect on Wed May  8 08:55:04 2019
# Do not modify this file manually.

auth        required                                pam_env.so
auth        required                                pam_faildelay.so delay=2000000
auth        [default=1 ignore=ignore success=ok]        pam_succeed_if.so uid >= 1000
quiet
auth        [default=1 ignore=ignore success=ok]        pam_localuser.so
auth        sufficient                              pam_unix.so nullok try_first_pass
auth        requisite                               pam_succeed_if.so uid >= 1000 quiet_success
auth        sufficient                              pam_sss.so forward_pass
auth        required                                pam_deny.so

account     required                                  pam_unix.so
account     sufficient                                pam_localuser.so
account     sufficient                                pam_succeed_if.so uid < 1000 quiet
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required                                  pam_permit.so
```

```
password    requisite                     pam_pwquality.so try_first_pass
local_users_only
password    sufficient                    pam_unix.so sha512 shadow nullok
try_first_pass use_authtok
password    sufficient                    pam_sss.so use_authtok
password    required                      pam_deny.so

session     optional                      pam_keyinit.so revoke
session     required                      pam_limits.so
-session    optional                      pam_systemd.so
session     [success=1 default=ignore]    pam_succeed_if.so service in crond
quiet use_uid
session     required                      pam_unix.so
session     optional                      pam_sss.so
```

9. */etc/pam.d/system-auth* 파일의 콘텐츠를 */etc/pam.d/password-auth* 파일에 복사합니다. **yes** 를 입력하여 파일의 현재 내용을 덮어쓰는 것을 확인합니다.

```
# cp /etc/pam.d/system-auth /etc/pam.d/password-auth
cp: overwrite '/etc/pam.d/password-auth'? yes
```

검증 단계

1. **AD_user** 의 **Kerberos TGT(Tross-granting** 티켓) 요청. 요청한 대로 **AD_user** 의 암호를 입력합니다.

```
# kinit AD_user
Password for AD_user@EXAMPLE.COM:
```

2. 가져온 **TGT**를 표시합니다.

```
# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: AD_user@EXAMPLE.COM

Valid starting     Expires           Service principal
11/02/20 04:16:38  11/02/20 14:16:38  krbtgt/EXAMPLE.COM@EXAMPLE.COM
 renew until 18/02/20 04:16:34
```

AD_user 가 **EXAMPLE.COM Kerberos** 도메인의 자격 증명을 사용하여 **rhel_host** 에 성공적으로 로그인했습니다.

7장. SSSD를 사용하여 호스트에서 사용자 액세스 보고

SSSD(보안 시스템 서비스 데몬)는 사용자가 클라이언트에 액세스할 수 있거나 액세스할 수 없는 항목을 추적합니다. 이 장에서는 액세스 제어 보고서 생성과 sssctl 도구를 사용하여 사용자 데이터를 표시하는 방법에 대해 설명합니다.

사전 요구 사항

- SSSD 패키지가 네트워크 환경에 설치되어 있습니다

7.1. SSSCTL 명령

sssctl 은 SSSD(보안 시스템 서비스 데몬) 상태에 대한 정보를 가져오는 통합된 방법을 제공하는 명령줄 도구입니다.

sssctl 유틸리티를 사용하여 다음에 대한 정보를 수집할 수 있습니다.

- 도메인 상태
- 클라이언트 사용자 인증
- 특정 도메인의 클라이언트에서 사용자 액세스
- 캐시된 컨텐츠에 대한 정보

sssctl 도구를 사용하면 다음을 수행할 수 있습니다.

- SSSD 캐시 관리
- 로그 관리

- *설정 파일 확인*

참고

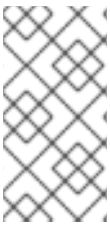
***sssctl** 툴은 **sss_cache** 및 **sss_debuglevel** 툴을 대체합니다.*

추가 리소스

- ***sssctl --help***

7.2. SSSCTL을 사용하여 액세스 제어 보고서 생성

***SSSD**는 클라이언트에 로그인할 수 있는 사용자를 제어하므로 보고서를 실행하는 시스템에 적용되는 액세스 제어 규칙을 나열할 수 있습니다.*



참고

*이 툴이 **KDC(**키 배포 센터**)**에 의해 잠긴 사용자를 추적하지 않기 때문에 액세스 보고서는 정확하지 않습니다.*

사전 요구 사항

- *관리자 권한으로 로그인해야 합니다*
- ***sssctl** 도구는 **RHEL 7** 및 **RHEL 8** 시스템에서 사용할 수 있습니다.*

절차

- ***idm.example.com** 도메인에 대한 보고서를 생성하려면 다음을 입력합니다.*

```
[root@client1 ~]# sssctl access-report idm.example.com
1 rule cached

Rule name: example.user
 Member users: example.user
 Member services: sshd
```

7.3. SSSCTL을 사용하여 사용자 권한 부여 세부 정보 표시

sssctl user-checks 명령을 사용하면 사용자 조회, 인증 및 권한 부여에 대해 **SSSD(System Security Services Daemon)**를 사용하는 애플리케이션의 문제를 디버깅할 수 있습니다.

sssctl user-checks [USER_NAME] 명령은 **NSS(Name Service Switch)** 및 **D-Bus** 인터페이스에 대한 **InfoPipe** 응답자를 통해 사용 가능한 사용자 데이터를 표시합니다. 표시된 데이터는 사용자가 **PAM(system -auth Pluggable Authentication Module)** 서비스를 사용하여 로그인할 권한이 있는지 여부를 보여줍니다.

명령에는 다음 두 가지 옵션이 있습니다.

- **-a PAM** 작업
- **- PAM** 서비스의 경우

a 및 **- s** 옵션을 정의하지 않으면 **sssctl** 도구는 기본 옵션인 **-a acct -s system-auth** 를 사용합니다.

사전 요구 사항

- 관리자 권한으로 로그인해야 합니다
- **sssctl** 도구는 **RHEL 7** 및 **RHEL 8** 시스템에서 사용할 수 있습니다.

절차

- 특정 사용자에 대한 사용자 데이터를 표시하려면 다음을 입력합니다.

```
[root@client1 ~]# sssctl user-checks -a acct -s sshd example.user
user: example.user
action: acct
service: sshd
....
```

추가 리소스

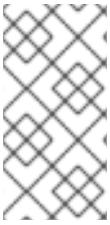
- ***sssctl user-checks --help***

8장. SSSD를 사용하여 도메인 정보 쿼리

SSSD(Security System Services Daemon)는 IdM(Identity Management) 및 IdM에 연결된 도메인(cross-forest trust)의 도메인을 나열할 수 있습니다.

8.1. SSSCTL을 사용하여 도메인 나열

sssctl domain-list 명령을 사용하여 도메인 토폴로지 문제를 디버깅할 수 있습니다.



참고

상태를 즉시 사용할 수 없을 수도 있습니다. 도메인이 표시되지 않으면 명령을 반복합니다.

사전 요구 사항

- 관리자 권한으로 로그인해야 합니다
- sssctl 도구는 RHEL 7 및 RHEL 8 시스템에서 사용할 수 있습니다.

절차

1. sssctl 명령에 대한 도움말을 표시하려면 다음을 입력합니다.

```
[root@client1 ~]# sssctl --help
....
```

2. 사용 가능한 도메인 목록을 표시하려면 다음을 입력합니다.

```
[root@client1 ~]# sssctl domain-list
implicit_files
idm.example.com
ad.example.com
sub1.ad.example.com
```

목록에는 Active Directory와 Identity Management 간의 교차 포트 간 신뢰 도메인이 포함되어 있습니다.

8.2. SSSCTL을 사용하여 도메인 상태 확인

sssctl domain-status 명령을 사용하여 도메인 토폴로지 문제를 디버깅할 수 있습니다.

참고

상태를 즉시 사용할 수 없을 수도 있습니다. 도메인이 표시되지 않으면 명령을 반복합니다.

사전 요구 사항

- 관리자 권한으로 로그인해야 합니다
- **sssctl** 도구는 **RHEL 7** 및 **RHEL 8** 시스템에서 사용할 수 있습니다.

절차

1. **sssctl** 명령에 대한 도움말을 표시하려면 다음을 입력합니다.

```
[root@client1 ~]# sssctl --help
```

2. 특정 도메인에 대한 사용자 데이터를 표시하려면 다음을 입력합니다.

```
[root@client1 ~]# sssctl domain-status idm.example.com
Online status: Online

Active servers:
IPA: server.idm.example.com

Discovered IPA servers:
- server.idm.example.com
```

도메인 **idm.example.com** 은 온라인이며 명령을 적용한 클라이언트에서 볼 수 있습니다.

도메인을 사용할 수 없는 경우 결과는 다음과 같습니다.

```
[root@client1 ~]# sssctl domain-status ad.example.com
Unable to get online status
```

9장. SSSD를 사용하여 PAM 서비스의 도메인 제한

PAM(장착형 인증 모듈)은 인증 및 권한 부여를 위한 공통 프레임워크입니다. Red Hat Enterprise Linux에서 대부분의 시스템 애플리케이션은 인증 및 권한 부여를 위한 기본 PAM 구성에 의존합니다.

SSSD(System Security Services Daemon)를 사용하면 PAM 서비스가 액세스할 수 있는 도메인을 제한할 수 있습니다. SSSD는 특정 PAM 서비스를 실행하는 사용자를 기반으로 PAM 서비스의 인증 요청을 평가합니다. 즉, PAM 서비스 사용자가 SSSD 도메인에 액세스할 수 있는 경우 PAM 서비스도 해당 도메인에 액세스할 수 있습니다.

9.1. PAM 정보

PAM(Pluggable Authentication Modules)은 시스템 애플리케이션이 중앙 집중식으로 구성된 프레임워크에 인증을 릴레이하는 데 사용할 수 있는 중앙 집중식 인증 메커니즘을 제공합니다.

PAM은 Kerberos, SSSD, NIS 또는 로컬 파일 시스템과 같은 다양한 인증 소스에 대해 PAM 모듈이 있으므로 플러그형입니다. 다양한 인증 소스의 우선 순위를 지정할 수 있습니다.

이 모듈식 아키텍처는 관리자가 시스템에 대한 인증 정책을 설정할 때 상당한 유연성을 제공합니다. PAM은 다음과 같은 여러 가지 이유로 개발자와 관리자에게 유용한 시스템입니다.

- PAM은 공통 인증 스키마를 제공하며 다양한 애플리케이션과 함께 사용할 수 있습니다.
- PAM은 시스템 관리자의 인증에 대한 상당한 유연성과 제어를 제공합니다.
- PAM은 개발자가 자체 인증 스키마를 만들지 않고도 프로그램을 작성할 수 있는 완전한 단일 라이브러리를 제공합니다.

9.2. 도메인 액세스 제한 옵션

선택한 도메인에 대한 액세스를 제한하는 다음 옵션을 사용할 수 있습니다.

pam_trusted_users in /etc/sssd/sssd.conf

이 옵션은 SSSD가 신뢰하는 PAM 서비스를 나타내는 숫자 UID 또는 사용자 이름 목록을 허용합니다. 기본 설정은 all 입니다. 즉, 모든 서비스 사용자가 신뢰할 수 있으며 모든 도메인에 액세스할 수

있습니다.

pam_public_domains in /etc/sssd/sssd.conf

이 옵션은 공용 **SSSD** 도메인 목록을 허용합니다. 퍼블릭 도메인은 신뢰할 수 없는 **PAM** 서비스 사용자가도 액세스할 수 있는 도메인입니다. 옵션은 모든 값과 **none** 값도 허용합니다. 기본값은 **none** 입니다. 즉, 도메인이 공용이고 신뢰할 수 없는 서비스 사용자가 어떤 도메인에도 액세스할 수 없습니다.

PAM 구성 파일의 도메인

이 옵션은 **PAM** 서비스가 인증할 수 있는 도메인 목록을 지정합니다. 도메인을 지정하지 않고 도메인 을 사용하는 경우 **PAM** 서비스는 모든 도메인에 대해 인증할 수 없습니다. 예를 들면 다음과 같습니다.

```
auth     required   pam_sss.so domains=
```

PAM 구성 파일에서 도메인을 사용하는 경우 **PAM** 서비스는 신뢰할 수 있는 사용자로 서비스가 실행 중일 때 모든 도메인에 대해 인증할 수 있습니다.

/etc/sssd/sssd.conf SSSD 구성 파일의 **domain** 옵션 도 **SSSD**에서 인증을 시도하는 도메인 목록을 지정합니다. **PAM** 구성 파일의 **domain** 옵션은 **sssd.conf** 의 도메인 목록을 확장할 수 없으며 짧은 목록을 지정하여 **sssd.conf** 도메인 목록만 제한할 수 있습니다. 따라서 도메인이 **PAM** 파일에 지정되지만 **sssd.conf** 에 지정되지 않은 경우 **PAM** 서비스는 도메인에 대해 인증할 수 없습니다.

기본 설정 **pam_trusted_users = all** 및 **pam_public_domains = none** 은 모든 **PAM** 서비스 사용자가 신뢰할 수 있고 모든 도메인에 액세스할 수 있음을 지정합니다. **PAM** 구성 파일의 **domain** 옵션을 사용하면 도메인에 대한 액세스가 제한됩니다.

PAM 구성 파일에서 도메인을 사용하여 도메인 을 지정하는 반면 **sssd.conf** 에는 **pam_public_domains** 에도 도메인을 지정해야 합니다 . 필수 도메인을 포함하지 않고 **pam_public_domains** 옵션을 사용하면 이 서비스가 신뢰할 수 없는 사용자로 실행되는 경우 **PAM** 서비스가 도메인에 대한 인증 실패로 이어집니다.

참고

PAM 구성 파일에 정의된 도메인 제한은 사용자 조회가 아닌 인증 작업에만 적용됩니다.

추가 리소스

-

pam_trusted_users 및 pam_ public_domains 옵션에 대한 자세한 내용은 sssd.conf(5) 도움말 페이지를 참조하십시오.

- *PAM 구성 파일에 사용된 domain 옵션에 대한 자세한 내용은 pam_sss(8) 도움말 페이지를 참조하십시오.*

9.3. PAM 서비스의 도메인 제한

다음 절차에서는 도메인에 대한 PAM 서비스 인증을 제한하는 방법을 설명합니다.

사전 요구 사항

- *SSSD 설치 및 실행.*

절차

1. *필요한 도메인 또는 도메인에 액세스하도록 SSSD를 구성합니다. /etc/sssd/sssd.conf 파일의 domain 옵션에서 SSSD를 인증할 수 있는 도메인 을 정의합니다.*

```
[sssd]
domains = domain1, domain2, domain3
```

2. *PAM 구성 파일에서 domain 옵션을 설정하여 인증할 수 있는 도메인 또는 도메인을 지정합니다. 예를 들면 다음과 같습니다.*

```
auth        sufficient    pam_sss.so forward_pass domains=domain1
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
password    sufficient    pam_sss.so use_authtok
```

이 예에서는 PAM 서비스가 domain1 에 대해서만 인증할 수 있습니다.

검증 단계

- *domain1 에 대해 인증합니다. 성공해야 합니다.*

10장. NSLCD에서 SSSD로 인증 마이그레이션

10.1. RHEL 클라이언트를 NSLCD에서 SSSD로 마이그레이션

RHEL에서 ns-pam- ldap d 패키지가 제거되었으므로 Red Hat은 SSSD 및 해당 ldap 공급자로 마이그레이션할 것을 권장합니다. 이 공급자는 nslcd 서비스의 기능을 대체합니다. 다음 절차에서는 nss-pam-ldap 인증 구성을 사용하도록 이전에 구성한 클라이언트에서 LDAP 사용자를 인증하도록 SSSD 를 구성하는 방법을 설명합니다.

사전 요구 사항

- RHEL 클라이언트는 RHEL 8 또는 RHEL 9에 있습니다.
- 이전에는 nslcd 서비스를 사용하여 LDAP 디렉터리 서버에 인증하도록 RHEL 클라이언트를 구성했습니다.
- LDAP 디렉터리 서비스는 RFC-2307에 정의된 스키마를 사용합니다.

절차

1. 현재 인증 구성을 백업합니다.

```
# authselect apply-changes -b --backup=ldap-configuration-backup
```

2. SSSD 패키지를 설치합니다.

```
# yum install sssd-ldap sssd-ad sssd-client \
        sssd-common sssd-common-pac \
        sssd-krb5 sssd-krb5-common
```

3. nslcd 및 nscd 서비스를 중지하고 비활성화합니다.

```
# systemctl stop nslcd nscd
# systemctl disable nslcd nscd
```

4. SSSD 를 사용하여 인증 설정:

```
# authselect select sssd with-mkhomedir --force
```

5. *SSSD 구성 파일에 필요한 소유권 및 권한을 설정합니다.*

```
# chown root:root /etc/sssd/sssd.conf
# chmod 600 /etc/sssd/sssd.conf
```

6. *편집을 위해 /etc/sssd/sssd.conf 파일을 엽니다.*

7. *다음 구성을 입력하고 example.com 및 dc=example,dc=com 과 같은 값을 환경에 적합한 값으로 바꿉니다.*

```
[sssd]
config_file_version = 2
services = nss, pam
domains = EXAMPLE.COM
debug_level = 6

[domain/EXAMPLE.COM]
id_provider = ldap
auth_provider = ldap
ldap_uri = ldap://server.example.com/
ldap_search_base = dc=example,dc=com
ldap_default_bind_dn = CN=binddn,DC=example,DC=com
ldap_default_authtok_type = password
ldap_default_authtok = <bind_account_password>
cache_credentials = True
```

참고

SSSD 구성에서 LDAP 스키마를 지정해야 할 수 있습니다.

디렉터리 서버에서 RFC-2307bis 스키마를 사용하는 경우 [domain/EXAMPLE.COM] 섹션에 다음 행을 추가하십시오.

```
ldap_schema = rfc2307bis
```

Microsoft Active Directory 서버를 사용하는 경우 [domain/EXAMPLE.COM] 섹션에 다음 행을 추가하여 LDAP 기반 인증을 활성화합니다.

```
ldap_schema = ad
```

Kerberos 인증이 필요한 경우 Red Hat은 SSSD 서비스를 자동으로 구성하는 realm 명령을 사용하여 RHEL 클라이언트를 AD 도메인에 가입하는 것이 좋습니다.

8. SSSD 서비스를 활성화하고 시작합니다.

```
# systemctl enable sssd
# systemctl start sssd
```

검증 단계

1. LDAP 사용자에 대한 정보를 검색할 수 있는지 확인합니다.

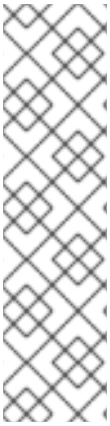
```
# id ldapuser
 uid=100424(ldapuser) gid=100424(ldapuser) groups=100424(ldapuser)

# getent passwd ldapuser
ldapuser:*: 100424: 100424:User, LDAP:/home/ldapuser:/bin/bash
```

2. LDAP 사용자로 로그인할 수 있는지 확인합니다.

```
# ssh -l ldapuser localhost
ldapuser@localhost's password:
Last login: Tue Dec 07 19:34:35 2021 from localhost
-sh-4.2$
```

-



참고

***nslcd* 및 *nscd* 를 사용하여 원래 *LDAP* 구성을 복원해야 하는 경우 다음 명령을 사용하십시오.**

```
# authselect backup-restore=ldap-configuration-backup
# systemctl stop sssd && systemctl disable sssd
# systemctl start nslcd nscd
# systemctl enable nslcd nscd
```

추가 리소스

- ***RFC-2307: LDAP*를 네트워크 정보 서비스로 사용하기 위한 접근 방식**
- ***RFC-2307bis* 스키마의 인터넷 개발**

10.2. SSSD.CONF 옵션 동등한 NSLCD.CONF 옵션

*다음 표에서는 **nslcd** 에서 **SSSD** 로 마이그레이션하는 데 도움이 되도록 **sssd.conf** 구성 파일의 **nslcd.conf** 구성 파일과 해당 옵션의 공통 옵션을 보여줍니다.*

표 10.1. sssd.conf 옵션 동등한 nslcd.conf 옵션

nslcd.conf 옵션	sssd.conf 옵션	설명
uid	*동등한되지 않음*	데몬을 실행해야 하는 사용자 ID입니다. 기본적으로 SSSD는 **sssd** 사용자로 실행됩니다.
gid	*동등한되지 않음*	데몬을 실행해야 하는 그룹 ID입니다. 기본적으로 SSSD는 **sssd** 개인 그룹으로 실행됩니다.
uri	**ldap_uri**	다음 형식의 LDAP 서버의 URI입니다. ldap **[s]://<host>[:port]**
base	**ldap_search_base**	검색 기반의 고유 이름입니다.
binddn	**ldap_default_bind_dn**	LDAP 작업을 수행하는 데 사용할 기본 바인딩 DN

nslcd.conf 옵션	sssd.conf 옵션	설명
bindpw	**ldap_default_authtok**	기본 바인딩 DN의 인증 토큰입니다. 현재 일반 텍스트 암호만 지원됩니다.
SSL start_tls	**ldap_id_use_start_tls = true**	기본 바인딩 DN의 인증 토큰입니다. 현재 일반 텍스트 암호만 지원됩니다.
tls_reqcert	**ldap_tls_reqcert**	서버 제공 인증서에서 수행할 검사를 지정합니다.
tls_cacertfile	**ldap_tls_cacert**	모든 인증 기관의 인증서가 포함된 파일
tls_cacertdir	**ldap_tls_cacertdir**	개별 파일에 인증 기관 인증서가 포함된 디렉터리의 경로입니다.
기본 passwd	**ldap_user_search_base**	사용자에 대한 LDAP 검색을 제한하는 선택적 기본 DN, 검색 범위 및 LDAP 필터입니다.
기본 그룹	**ldap_group_search_base**	그룹 검색을 제한하는 선택적 기본 DN, 검색 범위 및 LDAP 필터입니다.

추가 리소스

- ***nslcd.conf(5) 매뉴얼 페이지***
- ***sssd-ldap(5) man page***

11장. 로컬 SSSD 설정에서 오타 오류 제거

호스트의 /etc/sssd/sssd.conf 파일에 sssctl config-check 명령을 사용하여 오타 오류가 있는지 테스트할 수 있습니다.

사전 요구 사항

- 루트로 로그인되어 있습니다.
- sssd-tools 패키지가 설치됩니다.

절차

1. sssctl config-check 명령을 입력합니다.

```
# sssctl config-check

Issues identified by validators: 1
[rule/allowed_domain_options]: Attribute 'ldap_search' is not allowed in section
'domain/example1'. Check for typos.

Messages generated during configuration merging: 0

Used configuration snippet files: 0
```

2. /etc/sssd/sssd.conf 파일을 열고 오타를 수정합니다. 예를 들어 이전 단계에서 오류 메시지가 수신되면 ldap_search를 ldap_ search_ base:로 바꿉니다.

```
[...]
[domain/example1]
ldap_search_base = dc=example,dc=com
[...]
```

3. 파일을 저장합니다.
4. SSSD를 다시 시작:

```
# systemctl restart sssd
```

검증 단계

- *sssctl config-check 명령을 입력합니다.*

```
# sssctl config-check

Issues identified by validators: 0

Messages generated during configuration merging: 0

Used configuration snippet files: 0
```

/etc/sssd/sssd.conf 파일에 오타 오류가 없습니다.

12장. IDM에서 SSSD를 사용한 인증 문제 해결

IdM(Identity Management) 환경의 인증에는 다음과 같은 많은 구성 요소가 포함됩니다.

IdM 클라이언트의 경우:

- SSSD 서비스.
- NSS(이름 서비스 스위치).
- PAM(장착형 인증 모듈).

IdM 서버에서 다음을 수행합니다.

- SSSD 서비스.
- IdM 디렉터리 서버.
- IdM Kerberos KDC(키 배포 센터).

AD(Active Directory) 사용자로 인증하는 경우:

- AD 도메인 컨트롤러의 디렉터리 서버.
- AD 도메인 컨트롤러의 Kerberos 서버.

사용자를 인증하려면 SSSD 서비스를 사용하여 다음 기능을 수행할 수 있어야 합니다.

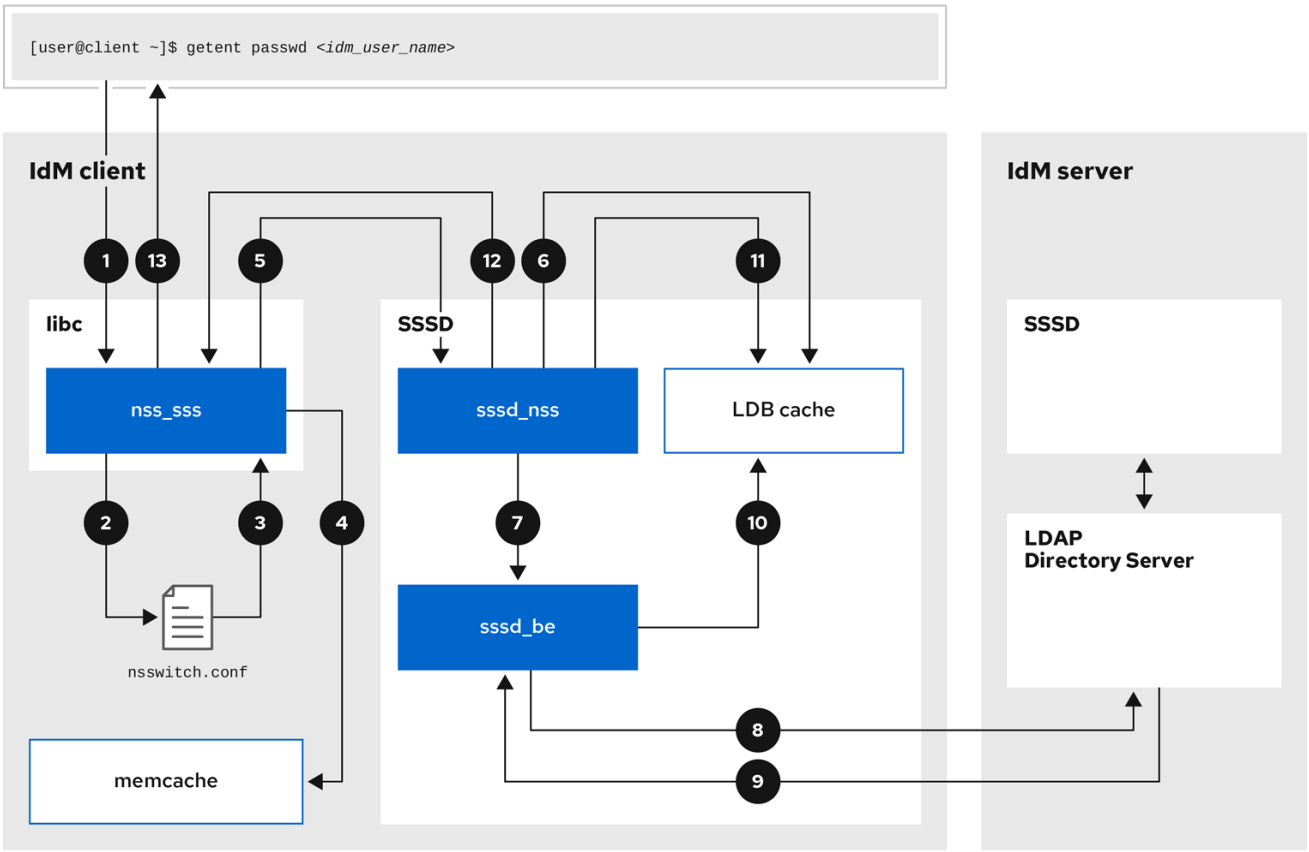
- 인증 서버에서 사용자 정보를 검색합니다.
- 사용자에게 자격 증명을 요청하고 해당 자격 증명을 인증 서버에 전달한 다음 결과를 처리합니다.

다음 섹션에서는 사용자 정보를 저장하는 SSSD 서비스와 서버 간 정보 흐름을 설명하므로 환경에서 실패한 인증 시도의 문제를 해결할 수 있습니다.

1. SSSD로 IdM 사용자 정보를 검색할 때 데이터 흐름
2. SSSD로 AD 사용자 정보를 검색할 때 데이터 흐름
3. IdM에서 SSSD를 사용하여 사용자로 인증할 때 데이터 흐름
4. 인증 문제 범위 좁음
5. SSSD 로그 파일 및 로깅 수준
6. sssd.conf 파일에서 SSSD에 대한 자세한 로깅 활성화
7. sssctl 명령을 사용하여 SSSD에 대한 자세한 로깅 활성화
8. IdM 서버의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집
9. IdM 클라이언트의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집
10. SSSD 백엔드에서 클라이언트 요청 추적
11. 로그 분석기 툴을 사용하여 클라이언트 요청 추적

12.1. SSSD로 IDM 사용자 정보를 검색할 때 데이터 흐름

다음 다이어그램은 getent passwd <idm_user_name> 명령을 사용하여 IdM 사용자 정보를 요청하는 동안 IdM 클라이언트와 IdM 서버 간의 정보 흐름을 간소화한 것입니다.



1. *getent 명령은 libc 라이브러리에서 getpwnam 호출을 트리거합니다.*

2. *libc 라이브러리는 /etc/nsswitch.conf 구성 파일을 참조하여 사용자 정보 제공을 담당하는 서비스를 확인하고 SSSD 서비스에 대한 항목을 검색합니다.*

3. *libc 라이브러리는 nss_sss 모듈을 엽니다.*

4. *nss_sss 모듈은 사용자 정보에 대한 메모리 매핑 캐시를 확인합니다. 데이터가 캐시에 있는 경우 nss_sss 모듈에서 해당 데이터를 반환합니다.*

5. *사용자 정보가 메모리 매핑 캐시에 없는 경우 요청이 SSSD sssd_nss 응답자 프로세스에 전달됩니다.*

6. *SSSD 서비스는 해당 캐시를 확인합니다. 데이터가 캐시에 있고 유효한 경우 sssd_nss 응답자가 캐시에서 데이터를 읽고 애플리케이션에 반환합니다.*

7. *데이터가 캐시에 없거나 만료된 경우 sssd_nss 응답자가 적절한 백엔드 프로세스를 쿼리하고 응답을 기다립니다. SSSD 서비스는 sssd.conf 구성 파일에서 id_provider=ipa 를 설정하여 활성화되는 IdM 환경에서 IPA 백엔드를 사용합니다.*

8. *sssd_be 백엔드 프로세스는 IdM 서버에 연결하고 IdM LDAP 디렉터리 서버에서 정보를 요청합니다.*

9. *IdM 서버의 SSSD 백엔드는 IdM 클라이언트의 SSSD 백엔드 프로세스에 응답합니다.*

10. *클라이언트의 SSSD 백엔드는 결과 데이터를 SSSD 캐시에 저장하고 캐시가 업데이트되었음을 응답 프로세스를 경고합니다.*

11. *sssd_nss 프론트엔드 응답자 프로세스는 SSSD 캐시에서 정보를 검색합니다.*

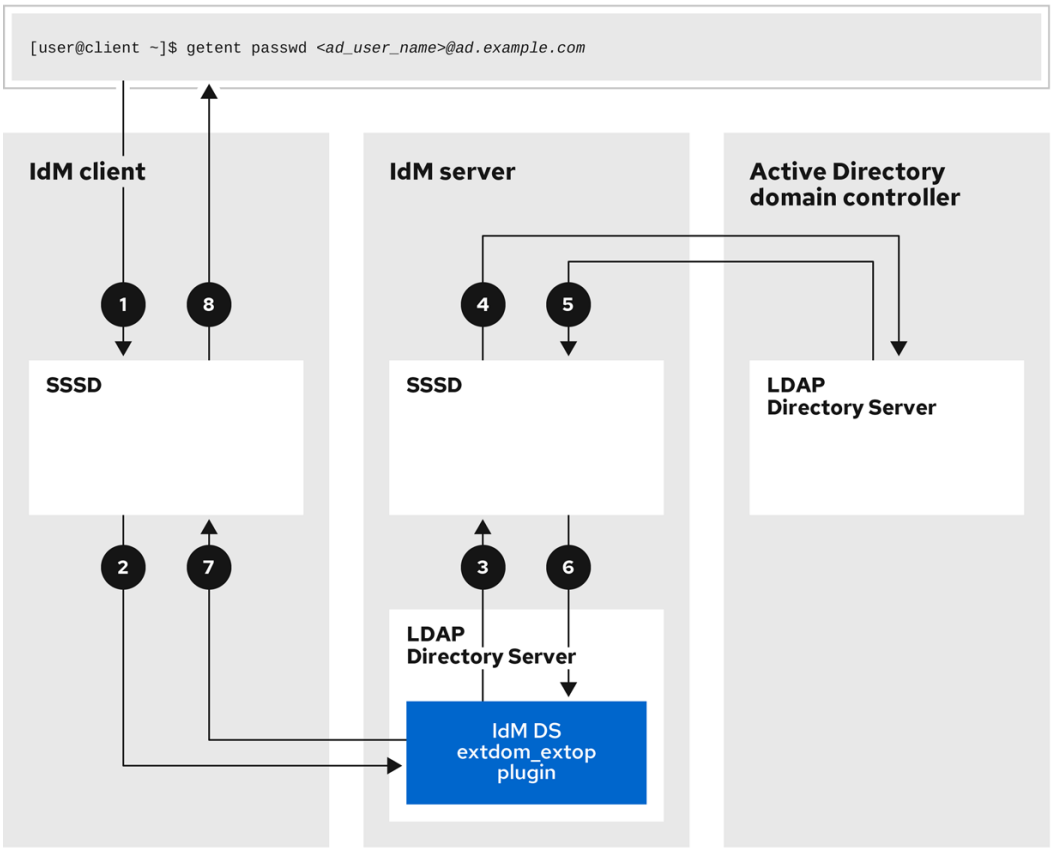
12. *sssd_nss 응답자가 사용자 정보를 nss_sss 응답자에게 전송하여 요청을 완료합니다.*

13. *libc 라이브러리는 사용자 정보를 요청한 애플리케이션에 반환합니다.*

12.2. SSSD로 AD 사용자 정보를 검색할 때 데이터 흐름

IdM 환경과 AD(Active Directory) 도메인 간에 일방적 신뢰를 수립한 경우 IdM 클라이언트의 AD 사용자 정보를 검색할 때 정보 흐름이 IdM 사용자 정보를 검색할 때 정보 흐름과 매우 유사하며, IdM 사용자 정보를 검색하는 추가 단계와 AD 사용자 데이터베이스에 연결하는 추가 단계가 있습니다.

다음 다이어그램은 사용자가 getent passwd <ad_user_name@ad.example.com> 명령을 사용하여 AD 사용자에 대한 정보를 요청할 때 정보 흐름을 간소화한 것입니다. 이 다이어그램에는 SSSD 섹션을 사용하여 IdM 사용자 정보를 검색할 때 데이터 흐름에서 설명한 내부 세부 정보가 포함되지 않습니다. IdM 클라이언트의 SSSD 서비스, IdM 서버의 SSSD 서비스 및 AD 도메인 컨트롤러의 LDAP 데이터베이스 간의 통신에 중점을 둡니다.



1. *IdM 클라이언트는 AD 사용자 정보를 위해 로컬 SSSD 캐시를 찾습니다.*

2. *IdM 클라이언트에 사용자 정보가 없거나 정보가 오래된 경우 클라이언트의 SSSD 서비스는 IdM 서버의 extdom_extop 플러그인에 연결하여 LDAP 확장 작업을 수행하고 정보를 요청합니다.*

3. *IdM 서버의 SSSD 서비스는 로컬 캐시에서 AD 사용자 정보를 찾습니다.*

4. *IdM 서버에 SSSD 캐시에 사용자 정보가 없거나 정보가 오래된 경우 LDAP 검색을 수행하여 AD 도메인 컨트롤러에서 사용자 정보를 요청합니다.*

5. *IdM 서버의 SSSD 서비스는 AD 도메인 컨트롤러에서 AD 사용자 정보를 수신하여 캐시에 저장합니다.*

6. *extdom_extop 플러그인은 IdM 서버의 SSSD 서비스에서 LDAP 확장 작업을 완료하는 정보를 받습니다.*

7. *IdM 클라이언트의 SSSD 서비스는 LDAP 확장 작업에서 AD 사용자 정보를 받습니다.*

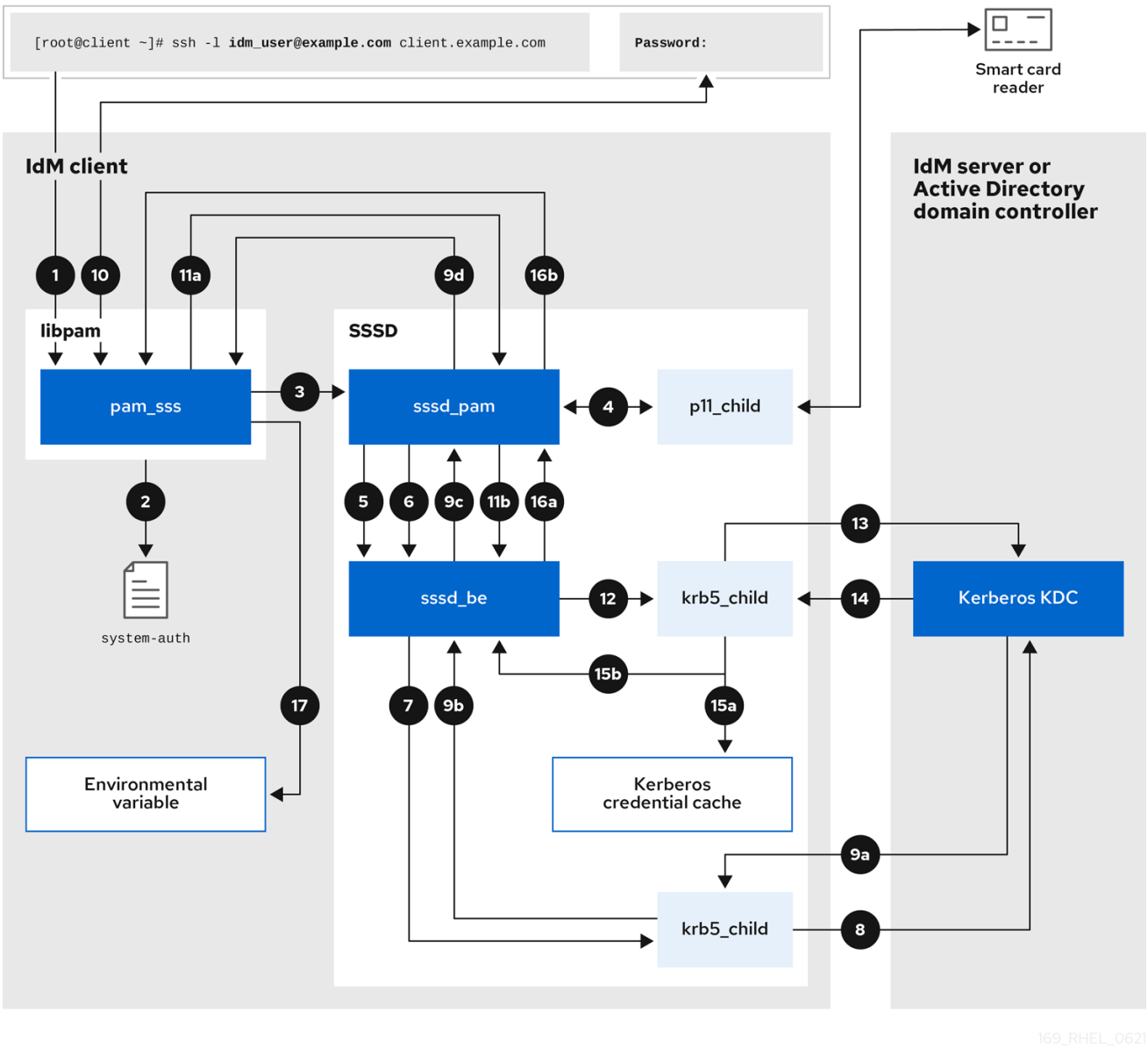
8. *IdM 클라이언트는 AD 사용자 정보를 SSSD 캐시에 저장하고 요청한 애플리케이션에 정보를 반환합니다.*

12.3. IDM에서 SSSD를 사용하여 사용자로 인증할 때 데이터 흐름

IdM 서버 또는 클라이언트에서 사용자로 인증하는 데 다음 구성 요소가 포함됩니다.

- *인증 요청을 시작하는 서비스(예: sshd 서비스).*
- *PAM(Pluggable Authentication Module) 라이브러리 및 해당 모듈.*
- *SSSD 서비스, 응답자 및 백엔드.*
- *스마트 카드 인증이 구성된 경우 스마트 카드 리더입니다.*
- *인증 서버:*
  - *IdM 사용자는 IdM Kerberos KDC(키 배포 센터)에 대해 인증됩니다.*
  - *AD(Active Directory) 사용자는 AD DC(Domain Controller)에 대해 인증됩니다.*

다음 다이어그램은 사용자가 명령줄에서 SSH 서비스를 통해 호스트에 로컬로 로그인하는 동안 인증해야 하는 경우 정보 흐름을 간소화한 것입니다.



1. *ssh 명령을 사용한 인증 시도는 libpam 라이브러리를 트리거합니다.*

2. *libpam 라이브러리는 인증 시도를 요청하는 서비스에 해당하는 /etc/pam.d/ 디렉터리에서 PAM 파일을 참조합니다. 이 예에서는 로컬 호스트의 SSH 서비스를 통한 인증과 함께 libpam 라이브러리에서 /etc/pam.d/system-auth 구성 파일을 확인하고 SSSD PAM에 대한 pam_sss.so 항목을 검색합니다.*

```
auth    sufficient    pam_sss.so
```

3. *사용 가능한 인증 방법을 결정하기 위해 libpam 라이브러리는 pam_sss 모듈을 열고 SSSD 서비스의 sssd _pam PAM 응답자에게 SSS_PAM_ PREAUTH 요청을 전송합니다.*

4. *스마트 카드 인증이 구성된 경우 **SSSD** 서비스는 스마트 카드를 확인하고 여기에서 인증서를 검색하는 임시 **p11_child** 프로세스를 생성합니다.*

5. *사용자에 대해 스마트 카드 인증이 구성된 경우 **sssd_pam responder**는 사용자와 함께 스마트 카드의 인증서 일치를 시도합니다. **sssd_pam responder**는 그룹 멤버십이 액세스 제어에 영향을 미칠 수 있으므로 사용자가 속한 그룹에 대한 검색도 수행합니다.*

6. ***sssd_pam responder**는 **SSS_PAM_PREAUTH** 요청을 **sssd_be** 백엔드 응답자에게 전송하여 서버가 지원하는 인증 방법(예: 암호 또는 **2** 단계 인증)을 확인합니다. **SSSD** 서비스가 **IPA** 응답자를 사용하는 **IdM** 환경에서 기본 인증 방법은 **Kerberos**입니다. 이 예에서는 사용자가 간단한 **Kerberos** 암호를 사용하여 인증합니다.*

7. ***sssd_be** 응답자는 **temporary krb5_child** 프로세스를 생성합니다.*

8. ***The krb5_child** 프로세스는 **IdM** 서버의 **KDC**에 접속하고 사용 가능한 인증 방법을 확인합니다.*

9. ***KDC**는 요청에 응답합니다.*

   a. ***The krb5_child** 프로세스는 응답을 평가하고 결과를 **sssd_be** 백엔드 프로세스로 다시 보냅니다.*

   b. ***sssd_be** 백엔드 프로세스는 결과를 수신합니다.*

   c. ***sssd_pam responder**는 결과를 수신합니다.*

   d. ***pam_sss** 모듈은 결과를 수신합니다.*

10. *사용자에 대해 암호 인증이 구성된 경우 **pam_sss** 모듈은 사용자에게 암호를 요청합니다. 스마트 카드 인증이 구성된 경우 **pam_sss** 모듈은 사용자에게 스마트 카드 **PIN**을 묻는 메시지를 표시합니다.*

11. *이 모듈은 **SSS_PAM_AUTHENTICATE** 요청을 보냅니다. 이 요청은 다음과 같이 이동합니다.*

a. *sssd_pam 응답자.*

b. *sssd_be 백엔드 프로세스.*

12. *sssd_be 프로세스는 KDC에 연결하기 위해 temporary krb5_child 프로세스를 생성합니다.*

13. *The krb5_child 프로세스는 사용자가 제공한 사용자 이름과 암호를 사용하여 KDC에서 Kerberos Ticket Granting Ticket(TGT)을 검색하려고 시도합니다.*

14. *The krb5_child 프로세스는 인증 시도의 결과를 수신합니다.*

15. *The krb5_child 프로세스:*

a. *TGT를 자격 증명 캐시에 저장합니다.*

b. *sssd_be 백엔드 프로세스에 인증 결과를 반환합니다.*

16. *인증 결과는 sssd_be 프로세스에서 다음과 같이 이동합니다.*

a. *sssd_pam 응답자.*

b. *pam_sss 모듈.*

17. *pam_sss 모듈은 다른 애플리케이션에서 참조할 수 있도록 사용자 TGT의 위치로 환경 변수를 설정합니다.*

12.4. 인증 문제 범위 좁음

사용자를 성공적으로 인증하려면 사용자 정보를 저장하는 데이터베이스에서 SSSD 서비스를 사용하여 사용자 정보를 검색할 수 있어야 합니다. 다음 절차에서는 사용자가 로그인할 수 없는 경우 인증 문제의

범위를 좁힐 수 있도록 인증 프로세스의 다양한 구성 요소를 테스트하는 단계를 설명합니다.

절차

1. SSSD 서비스 및 해당 프로세스가 실행 중인지 확인합니다.

```
[root@client ~]# pstree -a | grep sssd
  |-sssd -i --logger=files
  |   |-sssd_be --domain implicit_files --uid 0 --gid 0 --logger=files
  |   |-sssd_be --domain example.com --uid 0 --gid 0 --logger=files
  |   |-sssd_ifp --uid 0 --gid 0 --logger=files
  |   |-sssd_nss --uid 0 --gid 0 --logger=files
  |   |-sssd_pac --uid 0 --gid 0 --logger=files
  |   |-sssd_pam --uid 0 --gid 0 --logger=files
  |   |-sssd_ssh --uid 0 --gid 0 --logger=files
  |   `-sssd_sudo --uid 0 --gid 0 --logger=files
  |-sssd_kcm --uid 0 --gid 0 --logger=files
```

2. 클라이언트가 IP 주소를 통해 사용자 데이터베이스 서버에 연결할 수 있는지 확인합니다.

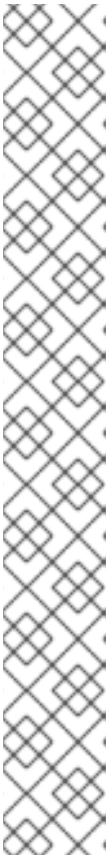
```
[user@client ~]$ ping <IP_address_of_the_database_server>
```

이 단계가 실패하면 네트워크 및 방화벽 설정에서 IdM 클라이언트와 서버 간의 직접 통신을 허용하는지 확인합니다. firewalld 사용 및 구성을 참조하십시오.

3. 클라이언트가 정규화된 호스트 이름을 통해 IdM LDAP 서버(Id 사용자용) 또는 AD 도메인 컨트롤러(AD 사용자의 경우)를 검색하고 연결할 수 있는지 확인합니다.

```
[user@client ~]$ dig -t SRV _ldap._tcp.example.com @<name_server>
[user@client ~]$ ping <fully_qualified_host_name_of_the_server>
```

이 단계가 실패하면 /etc/resolv.conf 파일을 포함하여 DNS(Dynamic Name Service) 설정을 확인합니다. DNS 서버 순서 구성을 참조하십시오.



참고

기본적으로 SSSD 서비스는 DNS 서비스(SRV) 레코드를 통해 LDAP 서버와 AD DC를 자동으로 검색합니다. 또는 sssd.conf 구성 파일에서 다음 옵션을 설정하여 특정 서버를 사용하도록 SSSD 서비스를 제한할 수 있습니다.

- ipa_server = <fully_qualified_host_name_of_the_server>
- ad_server = <fully_qualified_host_name_of_the_server>
- ldap_uri = <fully_qualified_host_name_of_the_server>

이러한 옵션을 사용하는 경우 목록에 나열된 서버에 연결할 수 있는지 확인합니다.

4. 클라이언트가 LDAP 서버에 인증하고 ldapsearch 명령으로 사용자 정보를 검색할 수 있는지 확인합니다.

a. LDAP 서버가 server.example.com 과 같이 IdM 서버인 경우 호스트의 Kerberos 티켓을 검색하고 호스트 Kerberos 주체로 데이터베이스 검색을 수행합니다.

```
[user@client ~]$ kinit -t 'host/client.example.com@EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.example.com -b
“dc=example,dc=com” uid=<user_name>
```

b. LDAP 서버가 server.ad.example.com 과 같이 AD(Active Directory) DC(Domain Controller)인 경우 호스트에 대한 Kerberos 티켓을 검색하고 호스트 Kerberos 주체로 데이터베이스 검색을 수행합니다.

```
[user@client ~]$ kinit -t 'CLIENT$@AD.EXAMPLE.COM'
[user@client ~]$ ldapsearch -LLL -Y GSSAPI -h server.ad.example.com -b
“dc=example,dc=com” sAMAccountname=<user_name>
```

c. LDAP 서버가 일반 LDAP 서버이고 sssd.conf 파일에서 ldap_default_bind_dn 및 ldap_default_authtok 옵션을 설정한 경우 동일한 ldap_default_bind_dn 계정으로 인증합니다.

```
[user@client ~]$ ldapsearch -xLLL -D "cn=ldap_default_bind_dn_value" -W -h
ldapserver.example.com -b “dc=example,dc=com” uid=<user_name>
```

*이 단계가 실패하면 데이터베이스 설정을 통해 호스트에서 **LDAP** 서버를 검색할 수 있는지 확인합니다.*

5. *SSSD 서비스는 **Kerberos** 암호화를 사용하므로 로그인할 수 없는 사용자로 **Kerberos** 티켓을 얻을 수 있는지 확인합니다.*

   a. ***LDAP** 서버가 **IdM** 서버인 경우:*

   ```
   [user@client ~]$ kinit <user_name>
   ```

   b. ***LDAP** 서버 데이터베이스가 **AD** 서버인 경우:*

   ```
   [user@client ~]$ kinit <user_name@AD.EXAMPLE.COM>
   ```

   *이 단계가 실패하면 **Kerberos** 서버가 올바르게 작동하고 모든 서버의 시간이 동기화되며 사용자 계정이 잠겨 있지 않은지 확인합니다.*

6. *명령줄에서 사용자 정보를 검색할 수 있는지 확인합니다.*

   ```
   [user@client ~]$ getent passwd <user_name>
   [user@client ~]$ id <user_name>
   ```

   *이 단계가 실패하면 클라이언트의 **SSSD** 서비스가 사용자 데이터베이스에서 정보를 수신할 수 있는지 확인합니다.*

   a. ***/var/log/messages** 로그 파일의 오류를 검토합니다.*

   b. ***SSSD** 서비스에서 자세한 로깅을 활성화하고 디버깅 로그를 수집한 다음 로그에서 문제 소스에 대한 표시를 확인합니다.*

   c. *(선택 사항) **Red Hat** 기술 지원 케이스를 열고 수집한 문제 해결 정보를 제공합니다.*

7. 호스트에서 **sudo** 를 실행할 수 있는 경우 **sssctl** 유틸리티를 사용하여 사용자가 로그인할 수 있는지 확인합니다.

```
[user@client ~]$ sudo sssctl user-checks -a auth -s ssh <user_name>
```

이 단계가 실패하면 **PAM** 구성, **IdM HBAC** 규칙 및 **IdM RBAC** 규칙과 같은 인증 설정을 확인합니다.

a. 사용자의 **UID**가 **/etc/login.defs** 파일에 정의되어 있는 **UID_MIN** 보다 크거나 같은지 확인합니다.

b. **/var/log/secure** 및 **/var/log /messages** 로그 파일에서 인증 오류를 검토합니다.

c. **SSSD** 서비스에서 자세한 로깅을 활성화하고 디버깅 로그를 수집한 다음 로그에서 문제 소스에 대한 표시를 확인합니다.

d. (선택 사항) **Red Hat** 기술 지원 케이스를 열고 수집한 문제 해결 정보를 제공합니다.

추가 리소스

- **sssd.conf** 파일에서 **SSSD**에 대한 자세한 로깅 활성화
- **sssctl** 명령을 사용하여 **SSSD**에 대한 자세한 로깅 활성화
- **IdM** 서버의 인증 문제 해결을 위해 **SSSD** 서비스에서 디버깅 로그 수집
- **IdM** 클라이언트의 인증 문제 해결을 위해 **SSSD** 서비스에서 디버깅 로그 수집

12.5. SSSD 로그 파일 및 로깅 수준

각 **SSSD** 서비스는 **/var/log/sssd/** 디렉토리에 있는 자체 로그 파일에 로그인합니다. **example.com IdM** 도메인에 있는 **IdM** 서버의 경우 로그 파일은 다음과 같을 수 있습니다.

```
[root@server ~]# ls -l /var/log/sssd/
total 620
-rw-------. 1 root root      0 Mar 29 09:21 krb5_child.log
-rw-------. 1 root root  14324 Mar 29 09:50 ldap_child.log
-rw-------. 1 root root 212870 Mar 29 09:50 sssd_example.com.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_ifp.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_implicit_files.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd.log
-rw-------. 1 root root 219873 Mar 29 10:03 sssd_nss.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_pac.log
-rw-------. 1 root root  13105 Mar 29 09:21 sssd_pam.log
-rw-------. 1 root root   9390 Mar 29 09:21 sssd_ssh.log
-rw-------. 1 root root      0 Mar 29 09:21 sssd_sudo.log
```

12.5.1. SSSD 로그 파일 용도

krb5_child.log

Kerberos 인증과 관련된 수명이 짧은 도우미 프로세스의 로그 파일입니다.

ldap_child.log

LDAP 서버와의 통신을 위한 Kerberos 티켓을 가져오는 데 관련된 짧은 도우미 프로세스의 로그 파일입니다.

sssd_<example.com>.log

sssd.conf 파일의 각 도메인 섹션에 대해 SSSD 서비스는 LDAP 서버와의 통신에 대한 정보를 별도의 로그 파일에 기록합니다. 예를 들어, example.com 이라는 IdM 도메인이 있는 환경에서 SSSD 서비스는 sssd_example.com.log 라는 파일에 해당 정보를 기록합니다. 호스트가 ad.example.com 이라는 AD 도메인과 직접 통합되는 경우 정보는 sssd_ad.example.com.log 라는 파일에 기록됩니다.

참고

IdM 환경과 AD 도메인에 대한 교차 신뢰가 있는 경우 AD 도메인에 대한 정보는 여전히 IdM 도메인의 로그 파일에 기록됩니다.

마찬가지로 호스트가 AD 도메인에 직접 통합된 경우 하위 도메인에 대한 정보가 기본 도메인의 로그 파일에 기록됩니다.

selinux_child.log

SELinux 정보를 검색하고 설정하는 수명이 짧은 도우미 프로세스의 로그 파일입니다.

sssd.log

SSSD 모니터링 및 해당 응답자 및 백엔드 프로세스와의 통신을 위한 로그 파일.

sssd_ifp.log

시스템 버스를 통해 액세스할 수 있는 공용 D-Bus 인터페이스를 제공하는 InfoPipe 응답자의 로그 파일입니다.

sssd_nss.log

사용자 및 그룹 정보를 검색하는 NSS(Name Services Switch) 응답자의 로그 파일입니다.

sssd_pac.log

Microsoft 권한 속성 인증서(PAC) 응답자의 로그 파일은 AD Kerberos 티켓에서 PAC를 수집하고 PAC에서 AD 사용자에 대한 정보를 파생하여 AD로부터 직접 요청하지 않도록 합니다.

sssd_pam.log

PAM(Pluggable Authentication Module) 응답자의 로그 파일.

sssd_ssh.log

SSH 응답자 프로세스의 로그 파일.

12.5.2. SSSD 로깅 수준

디버그 수준을 설정하면 그 아래의 모든 디버그 수준도 활성화됩니다. 예를 들어 6에서 디버그 수준을 설정하면 디버그 수준 0~5도 활성화됩니다.

표 12.1. SSSD 로깅 수준

level	설명
0	**치명적인 오류**. SSSD 서비스가 시작되지 않거나 종료되지 않도록 하는 오류입니다. 이는 RHEL 8.3 및 이전 버전의 기본 디버그 로그 수준입니다.
1	**심각한 오류**. SSSD 서비스를 종료하지 않지만 하나 이상의 주요 기능이 제대로 작동하지 않는 오류.
2	**심각한 오류**. 특정 요청 또는 작업이 실패했음을 알리는 오류가 발생했습니다. 이는 RHEL 8.4 이상의 기본 디버그 로그 수준입니다.
3	**사소한 오류**. 수준 2에서 작업 실패를 발생시키는 오류.

level	설명
4	**구성** 설정.
5	**기능** 데이터.
6	**작업** 기능에 대한 메시지 추적.
7	**내부 제어** 기능에 대한 메시지 추적.
8	**함수 내부 변수의** 내용.
9	매우 **낮은 수준의 추적** 정보.

12.6. SSSD.CONF 파일에서 SSSD에 대한 자세한 로깅 활성화

*기본적으로 **RHEL 8.4** 이상의 **SSSD** 서비스는 심각한 오류(디버그 수준 **2**)만 기록하지만 인증 문제를 해결하는 데 필요한 세부 수준까지 기록하지는 않습니다.*

***SSSD** 서비스를 다시 시작할 때 상세한 로깅을 영구적으로 활성화하려면 **/etc/sssd/sssd.conf** 설정 파일의 각 섹션에서 **debug_level=<integer>** 옵션을 추가합니다. 여기서 **<integer>** 값은 **0**에서 **9** 사이의 숫자입니다. 디버그 수준은 최대 **3** 로그 오류, 수준 **8** 이상에서는 많은 양의 자세한 로그 메시지를 제공합니다. 레벨 **6** 은 인증 문제를 디버깅하기 위한 시작점입니다.*

사전 요구 사항

- ***sssd.conf** 설정 파일을 편집하고 **SSSD** 서비스를 다시 시작하려면 루트 암호가 필요합니다.*

절차

1. *텍스트 편집기에서 **/etc/sssd/sssd.conf** 파일을 엽니다.*

2. *파일의 모든 섹션에 **debug_level** 옵션을 추가하고 **debug** 수준을 선택한 상세 정보로 설정합니다.*

```
[domain/example.com]
debug_level = 6
id_provider = ipa
...

[sssd]
```

```
debug_level = 6
services = nss, pam, ifp, ssh, sudo
domains = example.com

[nss]
debug_level = 6

[pam]
debug_level = 6

[sudo]
debug_level = 6

[ssh]
debug_level = 6

[pac]
debug_level = 6

[ifp]
debug_level = 6
```

3. *sssd.conf 파일을 저장하고 닫습니다.*

4. *SSSD 서비스를 다시 시작하여 새 구성 설정을 로드합니다.*

```
[root@server ~]# systemctl restart sssd
```

추가 리소스

- *SSSD 로그 파일 및 로깅 수준*

12.7. SSSCTL 명령을 사용하여 SSSD에 대한 자세한 로깅 활성화

기본적으로 RHEL 8.4 이상의 SSSD 서비스는 심각한 오류(디버그 수준 2)만 기록하지만 인증 문제를 해결하는 데 필요한 세부 수준까지 기록하지는 않습니다.

명령줄에서 sssctl debug-level <integer> 명령을 사용하여 SSSD 서비스의 디버그 수준을 변경할 수 있습니다. 여기서 <integer> 값은 0에서 9 사이의 숫자입니다. 디버그 수준은 최대 3 로그 오류, 수준 8 이상에서는 많은 양의 자세한 로그 메시지를 제공합니다. 레벨 6은 인증 문제를 디버깅하기 위한 시작점입니다.

사전 요구 사항

- *sssctl 명령을 실행하려면 루트 암호가 필요합니다.*

절차

- *sssctl debug-level 명령을 사용하여 선택한 디버그 수준을 원하는 세부 정보 표시로 설정합니다.*

```
[root@server ~]# sssctl debug-level 6
```

추가 리소스

- *SSSD 로그 파일 및 로깅 수준*

12.8. IDM 서버의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집

IdM 사용자로 IdM 서버를 인증할 때 문제가 발생하는 경우 서버의 SSSD 서비스에 대한 자세한 디버그 로그인을 활성화하고 사용자에 대한 정보를 검색하려는 시도의 로그를 수집합니다.

사전 요구 사항

- *sssctl 명령을 실행하고 SSSD 서비스를 다시 시작하려면 루트 암호가 필요합니다.*

절차

1. *IdM 서버에서 자세한 SSSD 디버그 로깅을 활성화합니다.*

```
[root@server ~]# sssctl debug-level 6
```

2. *인증 문제가 발생한 사용자에 대해 SSSD 캐시에 있는 개체를 무효화하므로 LDAP 서버를 우회하지 않고 SSSD에 이미 캐시된 정보를 검색하지 않습니다.*

```
[root@server ~]# sssctl cache-expire -u idmuser
```

3. *이전 SSSD 로그를 제거하여 문제 해결 최소화.*

```
[root@server ~]# sssctl logs-remove
```

4. *시도 전후에 타임 스탬프를 수집하는 동안 인증 문제가 발생한 사용자로 전환하십시오. 이러한 타임스탬프는 데이터 공간의 범위를 더 좁힙니다.*

```
[root@server sssd]# date; su idmuser; date
Mon Mar 29 15:33:48 EDT 2021
su: user idmuser does not exist
Mon Mar 29 15:33:49 EDT 2021
```

5. *(선택 사항) 자세한 **SSSD** 로그를 계속 수집하지 않으려면 디버그 수준을 낮춥니다.*

```
[root@server ~]# sssctl debug-level 2
```

6. *실패한 요청에 대한 정보는 **SSSD** 로그를 검토합니다. 예를 들어, **/var/log/sssd/sssd_example.com.log** 파일을 검토하면 **SSSD** 서비스에서 **cn=accounts,dc=example,dc=com LDAP** 하위 트리에서 사용자를 찾지 못했음을 확인할 수 있습니다. 이는 사용자가 없거나 다른 위치에 있음을 나타낼 수 있습니다.*

```
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [dp_get_account_info_send]
(0x0200): Got request for [0x1][BE_REQ_USER][name=idmuser@example.com]
...
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_get_generic_ext_step]
(0x0400): calling ldap_search_ext with [(&(uid=idmuser)(objectclass=posixAccount)
(uid=)(&(uidNumber=)(!(uidNumber=0))))][cn=accounts,dc=example,dc=com].
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_get_generic_op_finished]
(0x0400): Search result: Success(0), no errmsg set
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sdap_search_user_process]
(0x0400): Search for users, returned 0 results.
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_search_by_name]
(0x0400): No such entry
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_delete_user] (0x0400):
Error: 2 (No such file or directory)
(Mon Mar 29 15:33:48 2021) [sssd[be[example.com]]] [sysdb_search_by_name]
(0x0400): No such entry
(Mon Mar 29 15:33:49 2021) [sssd[be[example.com]]]
[ipa_id_get_account_info_orig_done] (0x0080): Object not found, ending request
```

7. *인증 문제의 원인을 확인할 수 없는 경우:*

   a. *최근에 생성한 **SSSD** 로그를 수집합니다.*

   ```
   [root@server ~]# sssctl logs-fetch sssd-logs-Mar29.tar
   ```

   b. ***Red Hat** 기술 지원 케이스를 열고 다음을 제공합니다.*

i.

SSSD 로그: sssd-logs-Mar29.tar

ii.

로그에 해당하는 요청의 타임스탬프 및 사용자 이름을 포함한 콘솔 출력입니다.

```
[root@server sssd]# date; id idmuser; date
Mon Mar 29 15:33:48 EDT 2021
id: 'idmuser': no such user
Mon Mar 29 15:33:49 EDT 2021
```

12.9. IDM 클라이언트의 인증 문제 해결을 위해 SSSD 서비스에서 디버깅 로그 수집

IdM 클라이언트로 인증하려고 할 때 문제가 발생하는 경우 IdM 서버에서 사용자 정보를 검색할 수 있는지 확인합니다. IdM 서버에서 사용자 정보를 검색할 수 없는 경우 IdM 서버에서 정보를 검색하는 IdM 클라이언트에서 검색할 수 없습니다.

IdM 서버에서 인증 문제가 발생하지 않았다는 것을 확인한 후 IdM 서버와 IdM 클라이언트 모두에서 SSSD 디버깅 로그를 수집합니다.

사전 요구 사항

- *사용자는 IdM 서버가 아닌 IdM 클라이언트에 대한 인증 문제만 있습니다.*
- *sssctl 명령을 실행하고 SSSD 서비스를 다시 시작하려면 루트 암호가 필요합니다.*

절차

1. *클라이언트에서 다음을 수행합니다. 텍스트 편집기에서 /etc/sssd/sssd.conf 파일을 엽니다.*
2. *클라이언트에서 다음을 수행합니다. ipa_server 옵션을 파일의 [domain] 섹션에 추가하고 IdM 서버로 설정합니다. 이렇게 하면 IdM 클라이언트가 다른 IdM 서버를 자동으로 검색하지 않으므로 이 테스트가 하나의 클라이언트와 서버로만 제한됩니다.*

```
[domain/example.com]
ipa_server = server.example.com
...
```

3. *클라이언트에서 다음을 수행합니다. **sssd.conf** 파일을 저장하고 닫습니다.*

4. *클라이언트에서 다음을 수행합니다. **SSSD** 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.*

```
[root@client ~]# systemctl restart sssd
```

5. *서버 및 클라이언트에서 다음을 수행합니다. 자세한 **SSSD** 디버그 로깅을 활성화합니다.*

```
[root@server ~]# sssctl debug-level 6
```

```
[root@client ~]# sssctl debug-level 6
```

6. *서버 및 클라이언트에서 다음을 수행합니다. 인증 문제가 발생한 사용자에게 **SSSD** 캐시에 있는 개체를 무효화하므로 **LDAP** 데이터베이스를 우회하지 않고 **SSSD**에 이미 캐시된 정보를 검색하지 않습니다.*

```
[root@server ~]# sssctl cache-expire -u idmuser
```

```
[root@client ~]# sssctl cache-expire -u idmuser
```

7. *서버 및 클라이언트에서 다음을 수행합니다. 이전 **SSSD** 로그를 제거하여 문제 해결 최소화.*

```
[root@server ~]# sssctl logs-remove
```

```
[root@server ~]# sssctl logs-remove
```

8. *클라이언트에서 다음을 수행합니다. 시도 전후에 타임스탬프를 수집하는 동안 인증 문제가 발생한 사용자로 전환합니다. 이러한 타임스탬프는 데이터 공간의 범위를 더 좁힙니다.*

```
[root@client sssd]# date; su idmuser; date
Mon Mar 29 16:20:13 EDT 2021
su: user idmuser does not exist
Mon Mar 29 16:20:14 EDT 2021
```

9. *(선택 사항) 서버 및 클라이언트에서 다음을 수행합니다. 자세한 **SSSD** 로그를 계속 수집하지 않으려면 디버그 수준을 낮추십시오.*

```
[root@server ~]# sssctl debug-level 0
```

```
[root@client ~]# sssctl debug-level 0
```

10. *서버 및 클라이언트에서 다음을 수행합니다. 실패한 요청에 대한 정보는 **SSSD** 로그를 검토합니다.*

    a. *클라이언트 로그에서 클라이언트의 요청을 검토합니다.*

    b. *서버 로그에서 클라이언트의 요청을 검토합니다.*

    c. *서버 로그에서 요청 결과를 검토합니다.*

    d. *서버에서 요청의 결과를 수신한 클라이언트의 결과를 검토합니다.*

11. *인증 문제의 원인을 확인할 수 없는 경우:*

    a. ***IdM** 서버 및 **IdM** 클라이언트에서 최근에 생성한 **SSSD** 로그를 수집합니다. 호스트 이름 또는 역할에 따라 레이블을 지정합니다.*

    ```
    [root@server ~]# sssctl logs-fetch sssd-logs-server-Mar29.tar
    ```

    ```
    [root@client ~]# sssctl logs-fetch sssd-logs-client-Mar29.tar
    ```

    b. ***Red Hat** 기술 지원 케이스를 열고 다음을 제공합니다.*

        i. ***SSSD** 디버그 로그:*

            A. ***sssd-logs-server-Mar29.tar** 서버에서*

            B. *클라이언트에서 **sssd-logs-client-Mar29.tar***

        ii. *로그에 해당하는 요청의 타임스탬프 및 사용자 이름을 포함한 콘솔 출력입니다.*

```
[root@client sssd]# date; su idmuser; date
Mon Mar 29 16:20:13 EDT 2021
su: user idmuser does not exist
Mon Mar 29 16:20:14 EDT 2021
```

12.10. SSSD 백엔드에서 클라이언트 요청 추적

SSSD는 요청을 비동기적으로 처리하고 다른 요청의 메시지가 동일한 로그 파일에 추가되므로 고유한 요청 식별자와 클라이언트 ID를 사용하여 백엔드 로그에서 클라이언트 요청을 추적할 수 있습니다. 고유한 요청 식별자는 RID#<integer> 형식의 디버그 로그에 추가되고 [CID #< integer] 형식의 클라이언트 ID가 추가됩니다. 이를 통해 개별 요청과 관련된 로그를 격리하고 요청을 여러 SSSD 구성 요소의 로그 파일에서 처음부터 끝까지 추적할 수 있습니다.

사전 요구 사항

- 디버그 로깅을 활성화했으며 IdM 클라이언트에서 요청이 제출되었습니다.
- SSSD 로그 파일의 내용을 표시하려면 루트 권한이 있어야 합니다.

절차

1. SSSD 로그 파일을 검토하려면 less 유틸리티를 사용하여 로그 파일을 엽니다. 예를 들어 /var/log/sssd/sssd_example.com.log:

```
[root@server ~]# less /var/log/sssd/sssd_example.com.log
```

2. 클라이언트 요청에 대한 정보는 SSSD 로그를 검토합니다.

```
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_destructor] (0x0400): [RID#3]
Number of active DP request: 0
(2021-07-26 18:26:37): [be[testidm.com]] [dp_req_reply_std] (0x1000): [RID#3] DP
Request AccountDomain #3: Returning [Internal Error]:
3,1432158301,GetAccountDomain() not supported
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): [RID#4] DP Request
Account #4: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-07-26 18:26:37): [be[testidm.com]] [dp_attach_req] (0x0400): [RID#4] Number of
active DP request: 1
```

SSSD 로그 파일의 이 샘플 출력은 두 개의 다른 요청에 대한 고유 식별자 RID#3 및 RID#4 를 보여줍니다.

그러나 SSSD 클라이언트 인터페이스에 대한 단일 클라이언트 요청은 백엔드에서 여러 요청을 트리거하는 경우가 많으며 이로 인해 백엔드의 클라이언트 요청과 요청 간에 1-to-1의 상관 관계가 아닙니다. 백엔드의 여러 요청에는 RID 번호가 다르지만 각 초기 백엔드 요청에는 고유한 클라이언트 ID가 포함되어 있으므로 관리자는 여러 RID 번호를 단일 클라이언트 요청에 대해 추적할 수 있습니다.

다음 예제에서는 하나의 클라이언트 요청 [sssd.nss CID #1] 및 백엔드에서 생성된 여러 요청 [RID#5]을 [RID#13]으로 보여줍니다.

```
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#5] DP Request [Account #5]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#6] DP Request
[AccountDomain #6]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:16): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#7] DP Request [Account #7]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#8] DP Request [Initgroups
#8]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#9] DP Request [Account #9]:
REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#10] DP Request [Account
#10]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#11] DP Request [Account
#11]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#12] DP Request [Account
#12]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
(2021-10-29 13:24:17): [be[ad.vm]] [dp_attach_req] (0x0400): [RID#13] DP Request [Account
#13]: REQ_TRACE: New request. [sssd.nss CID #1] Flags [0x0001].
```

12.11. 로그 분석기 툴을 사용하여 클라이언트 요청 추적

SSSD(System Security Services Daemon)에는 여러 SSSD 구성 요소의 로그 파일에서 처음부터 끝까지 요청을 추적하는 데 사용할 수 있는 로그 구문 분석 도구가 포함되어 있습니다.

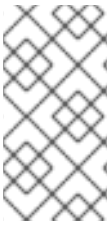
12.11.1. 로그 분석기 툴의 작동 방식

로그 구문 분석 툴을 사용하면 여러 SSSD 구성 요소의 로그 파일에서 SSSD 요청을 처음부터 끝까지 추적할 수 있습니다. sssctl analyze 명령을 사용하여 Analyzer 툴을 실행합니다.

로그 분석기 툴을 사용하면 SSSD에서 NSS 및 PAM 문제를 해결하고 SSSD 디버그 로그를 보다 쉽게 검토할 수 있습니다. SSSD 프로세스에서 특정 클라이언트 요청과 관련된 SSSD 로그를 추출하고 출력할 수 있습니다.

SSSD는 사용자 인증(su,ssh) 정보와 별도로 사용자 및 그룹 ID 정보(id,getent)를 추적합니다. NSS 응답자의 클라이언트 ID(CID)는 PAM 응답자의 CID와 독립적이며 NSS 및 PAM 요청을 분석할 때 중복되

는 숫자가 표시됩니다. **sssctl analyze** 명령과 함께 **--pam** 옵션을 사용하여 **PAM** 요청을 검토합니다.



참고

SSSD 메모리 캐시에서 반환된 요청은 기록되지 않으며 로그 분석기 툴에서 추적할 수 없습니다.

추가 리소스

- **sudo sssctl analyze request --help**
- **sudo sssctl analyze --help**
- **sssd.conf** 도움말 페이지
- **sssctl** 매뉴얼 페이지

12.11.2. 로그 분석기 툴 실행

다음 절차에서는 로그 분석기 툴을 사용하여 **SSSD**에서 클라이언트 요청을 추적하는 방법을 설명합니다.

사전 요구 사항

- 로그 구문 분석 기능을 활성화하려면 **/etc/sssd/sssd.conf** 파일의 **[$responder]** 섹션과 **[domain/$domain]** 섹션에서 **debug_level** 을 **7** 이상으로 설정해야 합니다.
- 분석할 로그는 **RHEL 8.5** 이상에서 **SSSD**인 **libtevent** 체인 **ID** 지원을 사용하여 구축된 **SSSD**의 호환 버전이어야 합니다.

절차

1. 목록 모드에서 로그 분석기 툴을 실행하여 추적 중인 요청의 클라이언트 **ID**를 확인하고 **-v** 옵션을 추가하여 상세 출력을 표시합니다.

```
# sssctl analyze request list -v
```

-

***SSSD*에 수행된 최근 클라이언트 요청의 상세 목록이 표시됩니다.**

참고

***PAM** 요청을 분석하는 경우 **--pam** 옵션을 사용하여 **sssctl analyze request list** 명령을 실행합니다.*

2.

***show [unique** 클라이언트 **ID]** 옵션과 함께 로그 **Analyzer** 툴을 실행하여 지정된 클라이언트 **ID** 번호와 관련된 로그를 표시합니다.*

```
# sssctl analyze request show 20
```

3.

*필요한 경우 로그 파일에 대해 로그 **Analyzer** 툴을 실행할 수 있습니다. 예를 들면 다음과 같습니다.*

```
# sssctl analyze request --logdir=/tmp/var/log/sssd
```

추가 리소스

- ***sssctl analyze request list --help***
- ***sssctl analyze request show --help***
- ***sssctl** 도움말 페이지.*

13장. SINGLE SIGN-ON의 애플리케이션 구성

SSO(Single Sign-On)는 단일 로그인 절차를 통해 여러 시스템에 로그인할 수 있는 인증 스키마입니다. Kerberos 티켓, SSL 인증 또는 토큰을 사용자 인증 수단으로 사용하도록 브라우저 및 이메일 클라이언트를 구성할 수 있습니다.

서로 다른 애플리케이션의 구성은 다를 수 있습니다. 이 장에서는 Mozilla Thunderbird 이메일 클라이언트와 Mozilla Firefox 웹 브라우저에 대한 SSO 인증 스키마를 예로 구성하는 방법을 보여줍니다.

13.1. 사전 요구 사항

- 다음 애플리케이션을 설치했습니다.
  - Mozilla Firefox 버전 88
  - Mozilla Thunderbird 버전 78

13.2. SINGLE SIGN-ON에 KERBEROS를 사용하도록 FIREFOX 구성

인트라넷 사이트 및 기타 보호 웹 사이트에 대해 SSO(Single Sign-On)에 Kerberos를 사용하도록 Firefox를 구성할 수 있습니다. 이렇게 하려면 먼저 Kerberos 자격 증명을 해당 KDC(키 배포 센터)에 보내도록 Firefox를 구성해야 합니다.

참고

Firefox가 Kerberos 자격 증명을 전달하도록 구성된 후에도 사용할 수 있는 유효한 Kerberos 티켓이 필요합니다. Kerberos 티켓을 생성하려면 kinit 명령을 사용하고 KDC에서 사용자에게 사용자 암호를 제공합니다.

```
[jsmith@host ~] $ kinit
Password for jsmith@EXAMPLE.COM:
```

절차

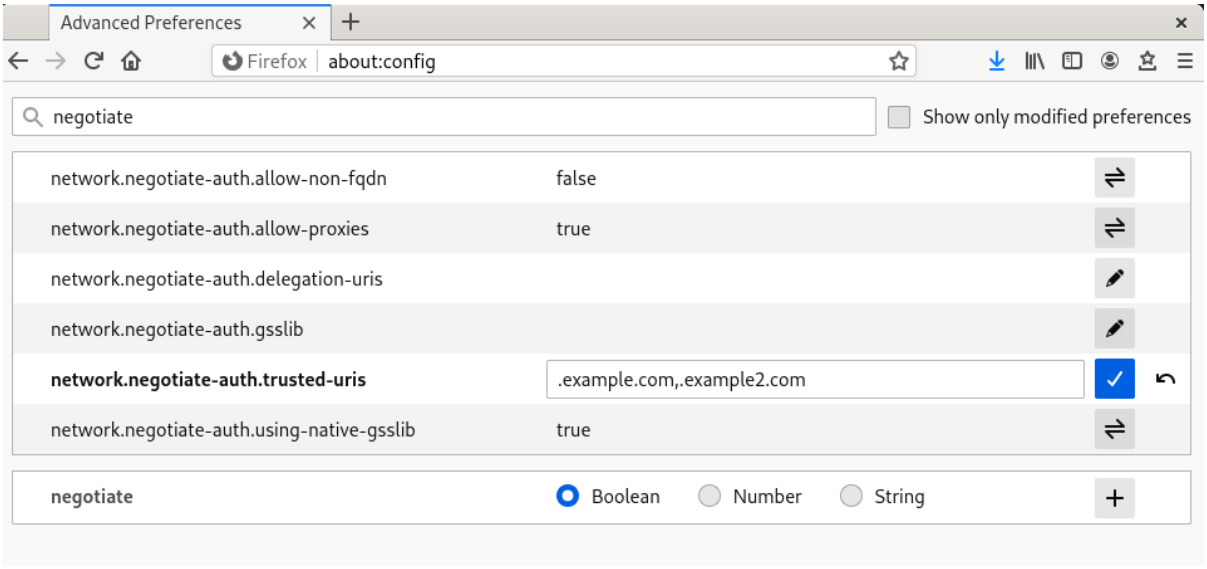
1. Firefox의 주소 표시줄에서 about:config 를 입력하여 현재 구성 옵션 목록을 표시합니다.

2. *Filter(필터) 필드에 negotiate 를 입력하여 옵션 목록을 제한합니다.*

3. *network.negotiate- auth.trusted-uris 항목을 두 번 클릭합니다.*

4. *앞의 기간(.)을 포함하여 인증할 도메인의 이름을 입력합니다. 여러 도메인을 추가하려면 쉼표로 구분된 목록에 입력합니다.*

그림 13.1. 수동 Firefox 설정



추가 리소스

- *ID 관리에서 Kerberos를 사용하도록 Firefox 구성에 대한 자세한 내용은 Linux 도메인 ID, 인증 및 정책 가이드의 해당 섹션을 참조하십시오.*

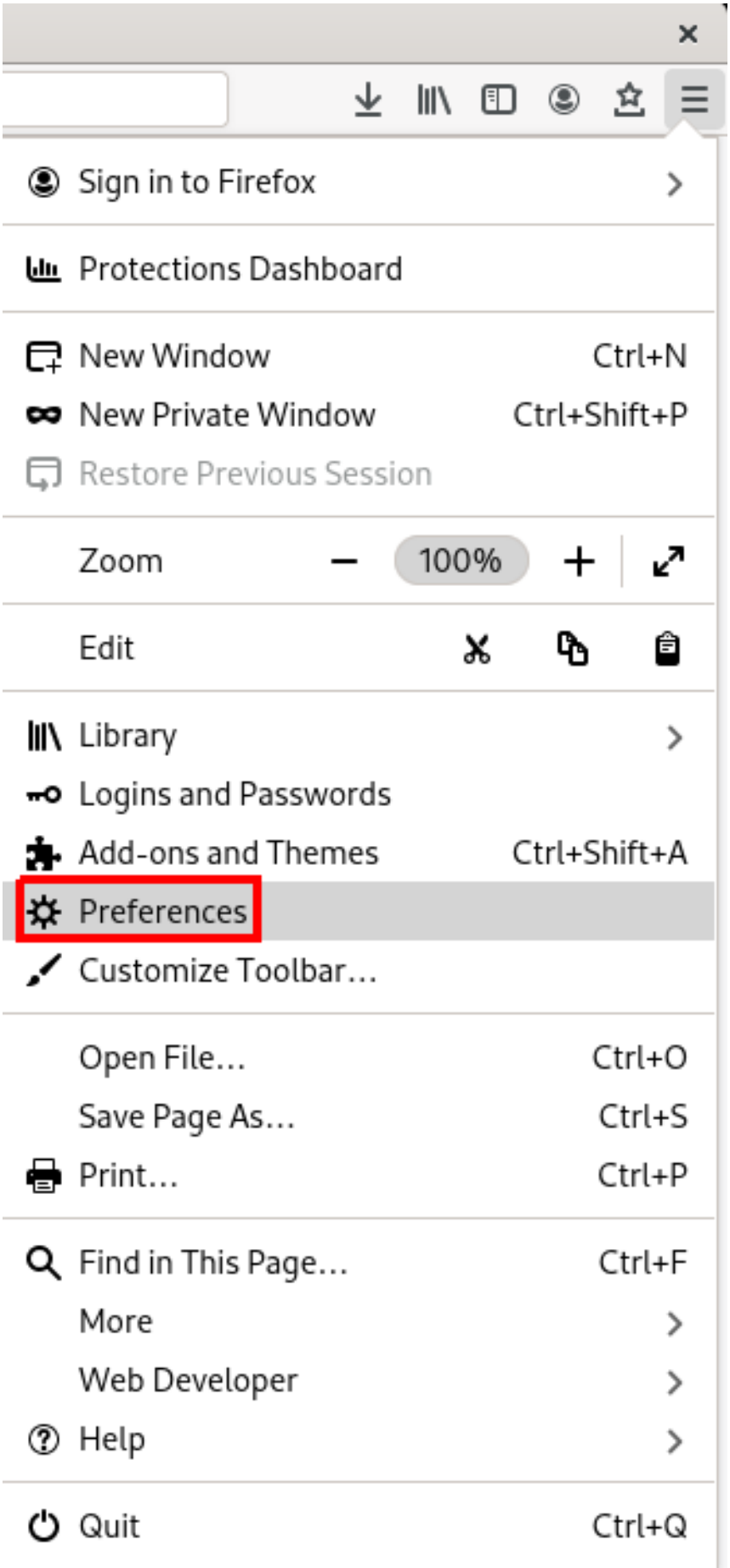
13.3. FIREFOX에서 인증서 보기

다음 예제에서는 Mozilla Firefox에서 인증서를 보는 방법을 보여줍니다.

Firefox에서 인증서를 보려면 인증서 관리자를 열어야 합니다 .

절차

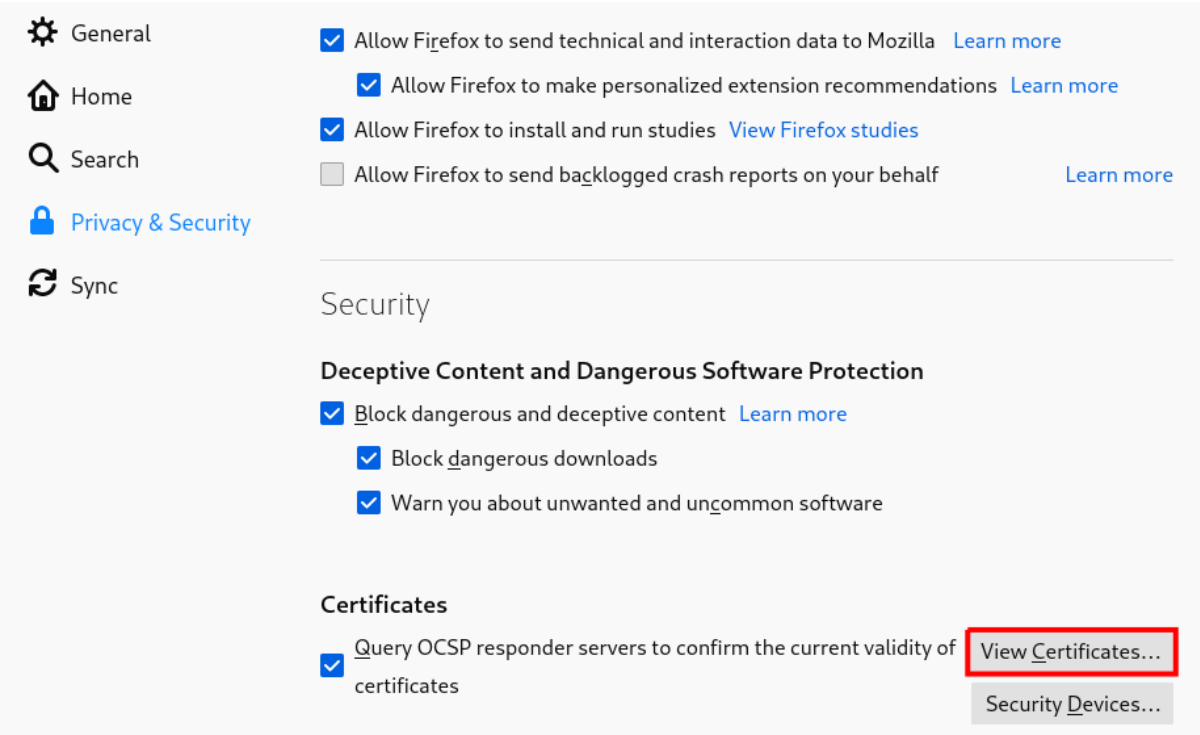
1. *Mozilla Firefox에서 Firefox 메뉴를 열고 Preferences(기본 설정)를 선택합니다.*



2. *왼쪽 패널에서 **Privacy & Security** (개인 정보 보호 및 보안) 섹션을 선택합니다.*

3. ***Certificates** (인증서) 섹션까지 아래로 스크롤합니다.*

4. ***View Certificates** (인증서 보기)를 클릭하여 인증서 관리자를 엽니다.*



13.4. FIREFOX에서 CA 인증서 가져오기

*다음 예제에서는 **Mozilla Firefox**에서 인증서를 가져오는 방법을 보여줍니다.*

사전 요구 사항

- *장치에 **CA** 인증서가 있습니다.*

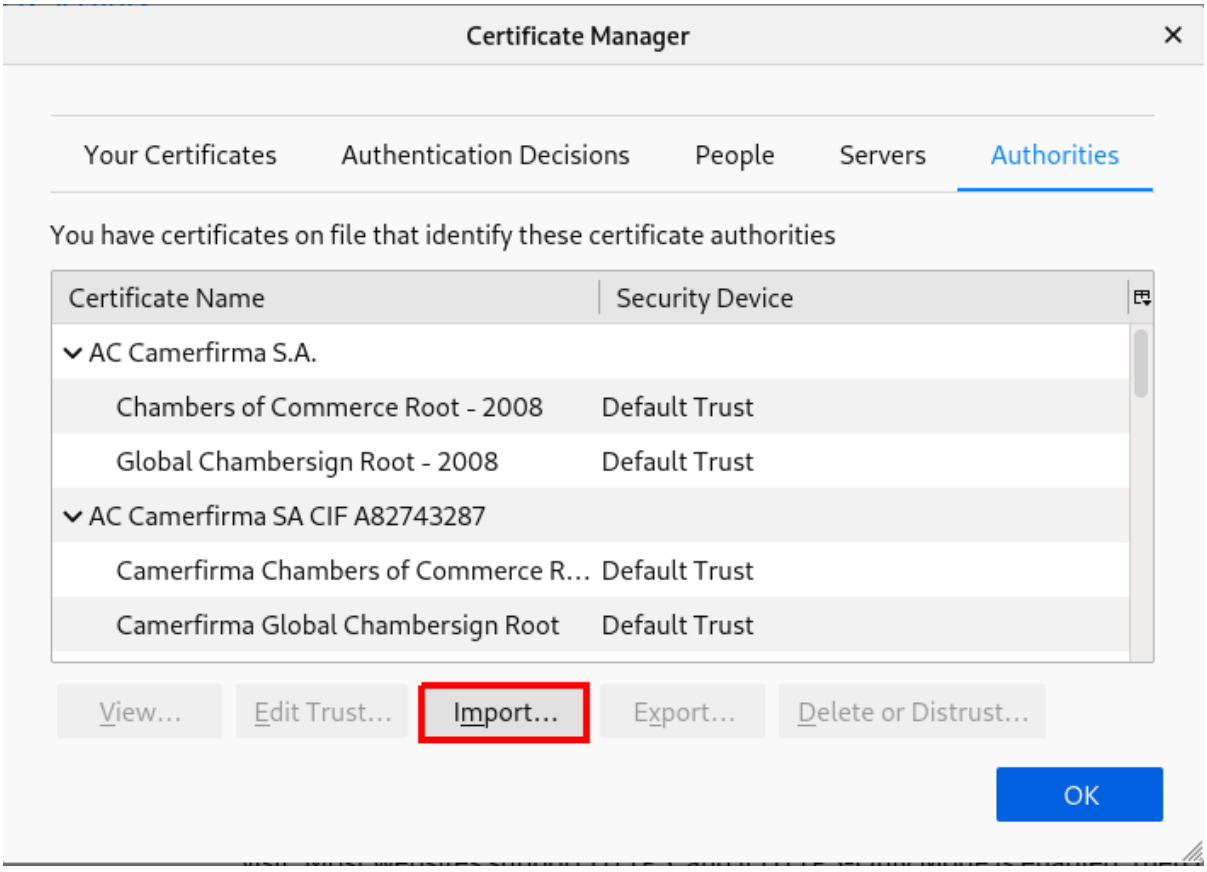
***CA** 인증서를 가져오려면 다음을 수행합니다.*

절차

1. ***Open Certificate Manager** (인증서 관리자 열기).*

2. ***Authorities**(권한) 탭을 선택하고 **Import**(가져오기)를 클릭합니다.*

*그림 **13.2. Firefox**에서 **CA** 인증서 가져오기*



3. *장치에서 다운로드한 **CA** 인증서를 선택합니다.*

13.5. FIREFOX에서 인증서 신뢰 설정 편집

*다음 예제에서는 **Mozilla Firefox**에서 인증서 설정을 편집하는 방법을 보여줍니다.*

사전 요구 사항

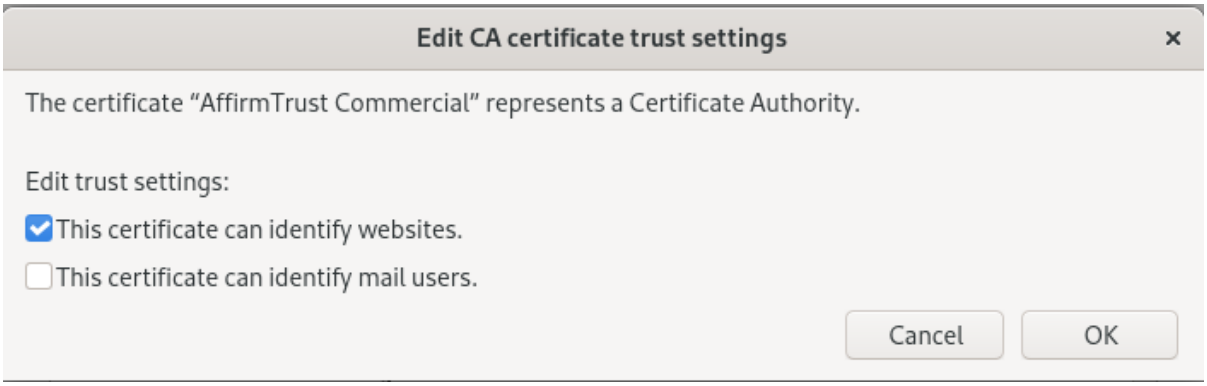
1. *인증서를 가져왔습니다.*

인증서 신뢰 설정을 설정하려면 다음을 수행합니다.

절차

1. ***Open Certificate Manager** (인증서 관리자 열기).*
2. ***Authorities**(권한) 탭에서 적절한 인증서를 선택하고 **Edit Trust**(신뢰 편집)를 클릭합니다.*
3. *인증서 신뢰 설정을 편집합니다.*

*그림 **13.3. Firefox**에서 인증서 신뢰 설정 편집*



13.6. FIREFOX에서 인증을 위한 개인 인증서 가져오기

*다음 예제에서는 **Mozilla Firefox**에서 인증을 위한 개인 인증서를 가져오는 방법을 보여줍니다.*

사전 요구 사항

1. *개인 인증서가 장치에 저장되어 있습니다.*

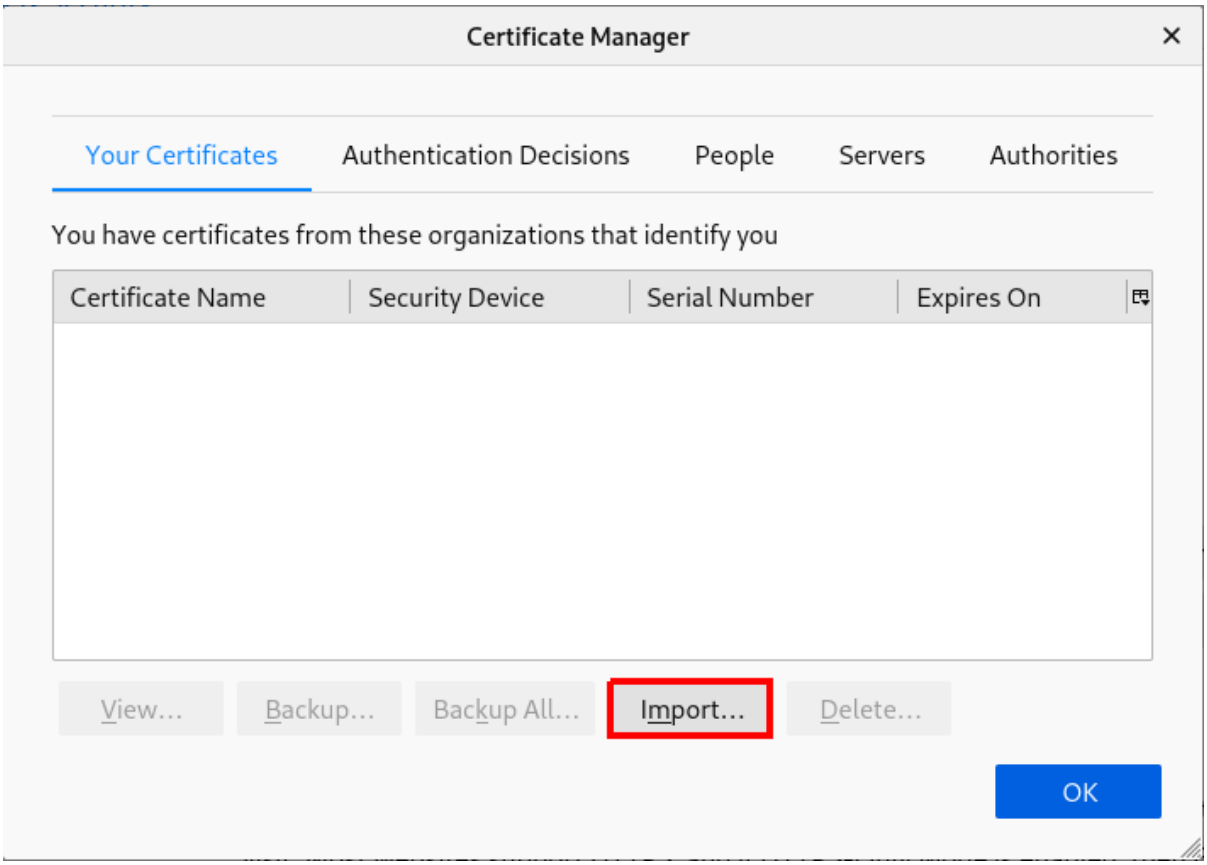
인증을 위해 개인 인증서를 사용하려면 다음을 수행합니다.

절차

1. ***Open Certificate Manager (인증서 관리자 열기).***

2. ***Certificates(인증서) 탭을 선택하고 Import(가져오기)를 클릭합니다.***

그림 13.4. Firefox에서 인증용 개인 인증서 가져오기



3. *컴퓨터에서 적절한 인증서를 선택합니다.*

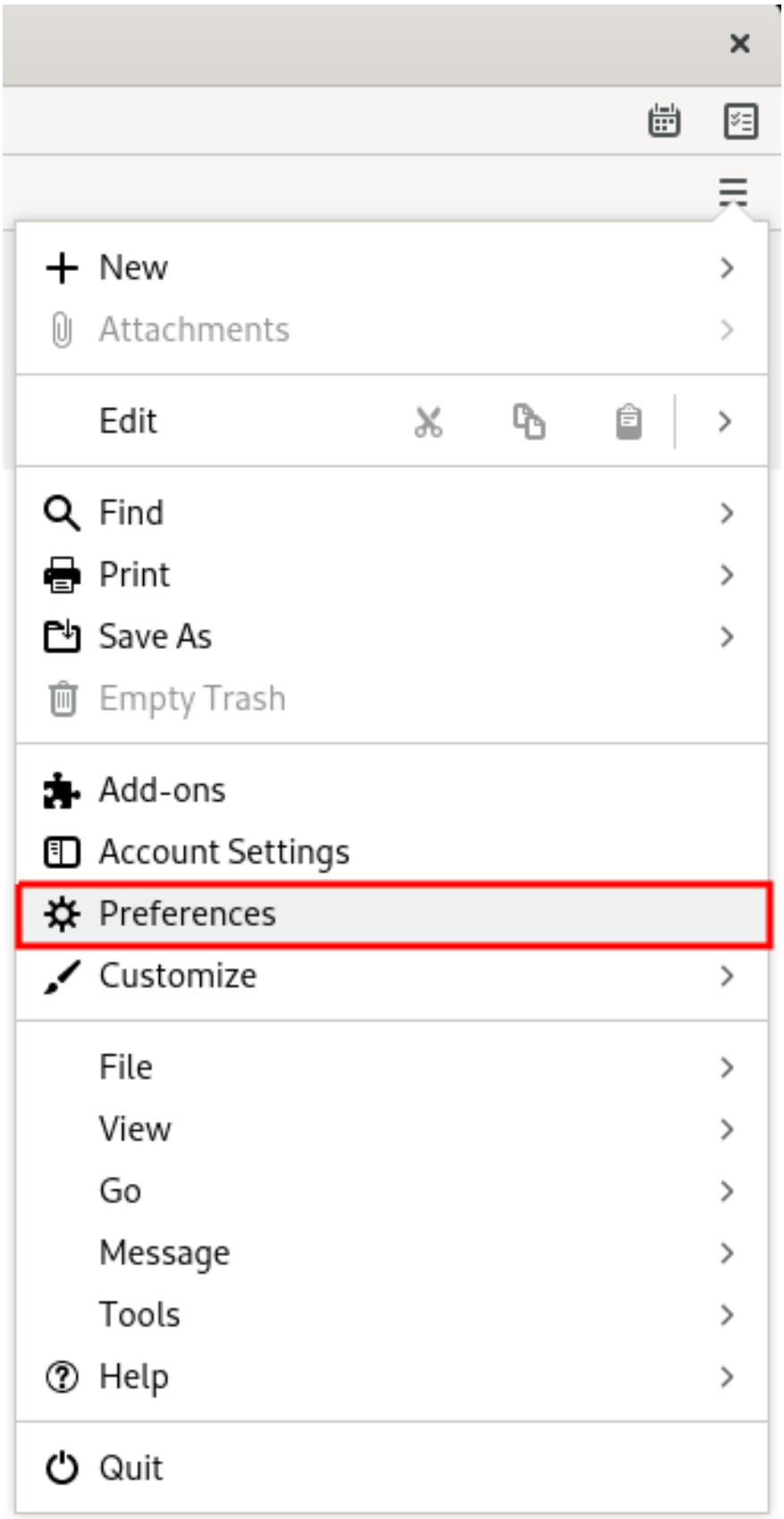
13.7. THUNDERBIRD에서 인증서 보기

다음 예제에서는 Mozilla Thunderbird 이메일 클라이언트에서 인증서를 보는 방법을 보여줍니다.

절차

1. ***Mozilla Thunderbird에서 주 메뉴를 열고 Preferences(기본 설정)를 선택합니다.***

그림 13.5. 메뉴에서 기본 설정 선택



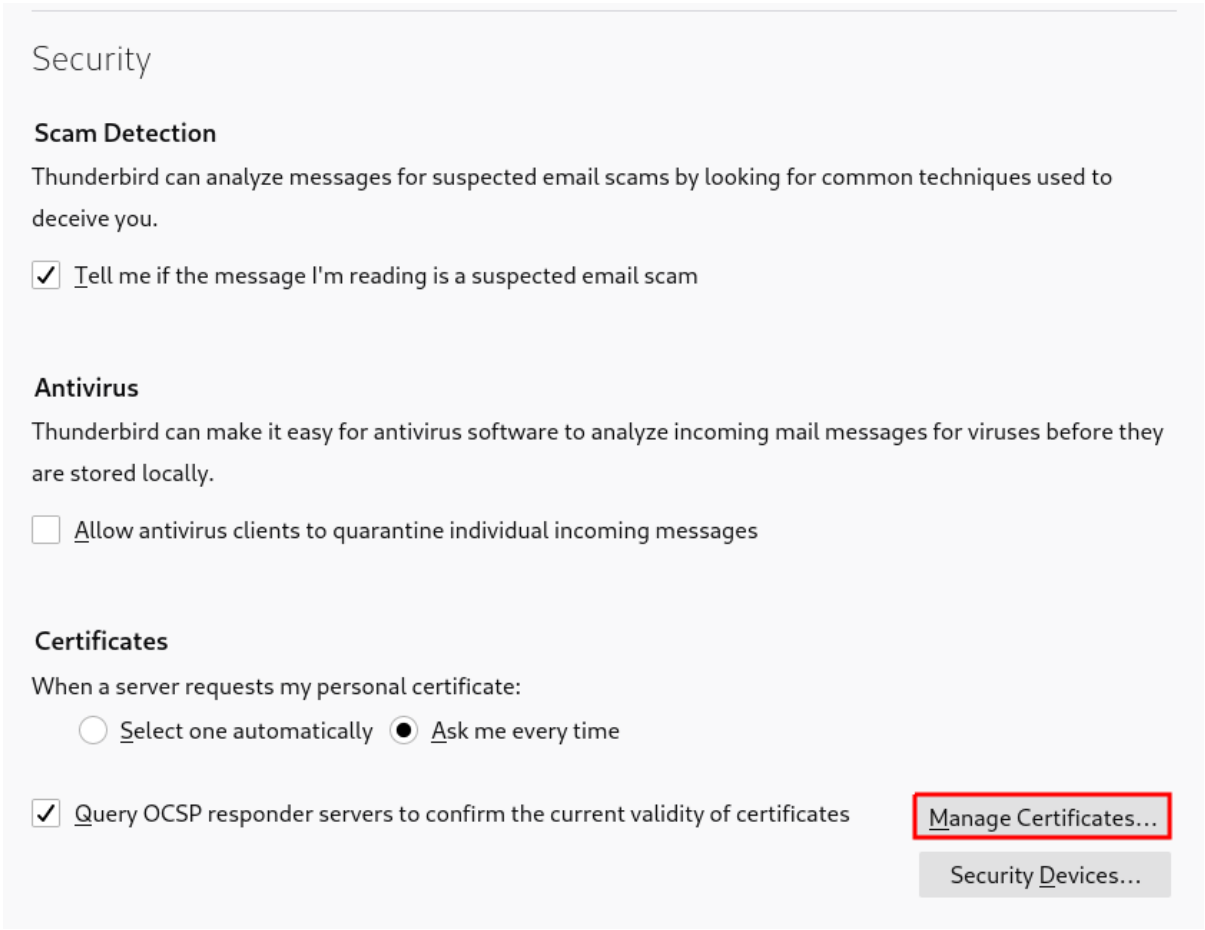
2. *왼쪽 패널에서 **Privacy & Security** (개인 정보 보호 및 보안) 섹션을 선택합니다.*

그림 13.6. 보안 섹션 선택

3. ***Certificates** (인증서) 섹션까지 아래로 스크롤합니다.*

4. ***Manage Certificates** (인증서 관리)를 클릭하여 인증서 관리자를 엽니다.*

그림 13.7. 인증서 관리자 열기



13.8. THUNDERBIRD에서 인증서 가져오기

*다음 예제에서는 **Mozilla Thunderbird** 이메일 클라이언트에서 인증서를 가져오는 방법을 보여줍니다.*

사전 요구 사항

- *장치에 **CA** 인증서가 저장되어 있습니다.*

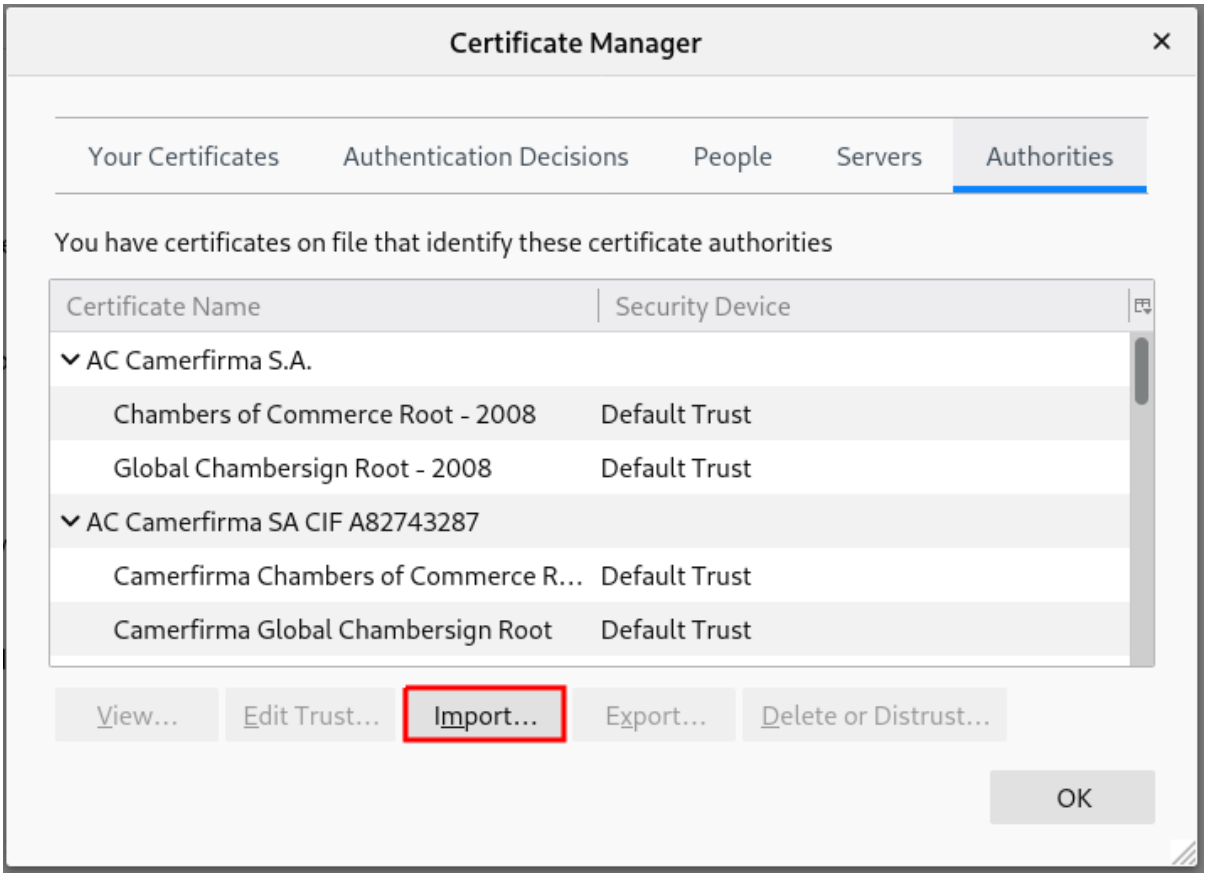
***CA** 인증서를 가져오려면 다음을 수행합니다.*

절차

1. ***Open Certificate Manager** (인증서 관리자 열기).*

2. ***Authorities**(권한) 탭을 선택하고 **Import**(가져오기)를 클릭합니다.*

*그림 **13.8. Thunderbird**에서 **CA** 인증서 가져오기*



3. *다운로드한 **CA** 인증서를 선택합니다.*

13.9. THUNDERBIRD에서 인증서 신뢰 설정 편집

다음 예제에서는 **Mozilla Thunderbird** 이메일 클라이언트에서 인증서 설정을 편집하는 방법을 보여줍니다.

사전 요구 사항

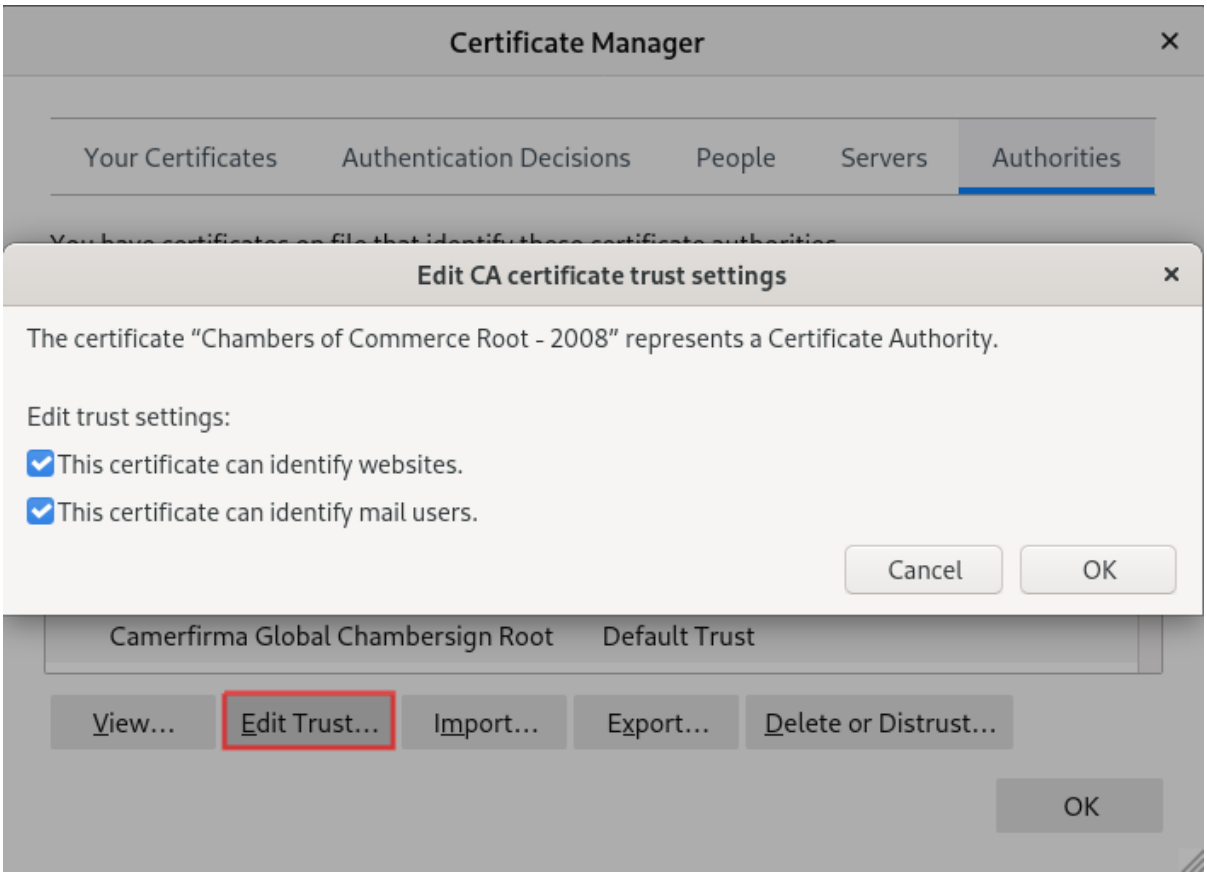
- 인증서를 가져왔습니다.

인증서 신뢰 관계를 설정하려면 다음을 수행합니다.

절차

1. **Open Certificate Manager (**인증서 관리자 열기**).**
2. **Authorities(**권한 **)** 탭에서 적절한 인증서를 선택하고 **Edit Trust(**신뢰 편집 **)**를 클릭합니다.
3. 인증서 신뢰 설정을 편집합니다.

그림 13.9. Thunderbird의 인증서 신뢰 설정 편집



13.10. THUNDERBIRD에서 개인 인증서 가져오기

*다음 예제에서는 **Mozilla Thunderbird** 이메일 클라이언트에서 개인 인증을 위한 인증서를 가져오는 방법을 보여줍니다.*

사전 요구 사항

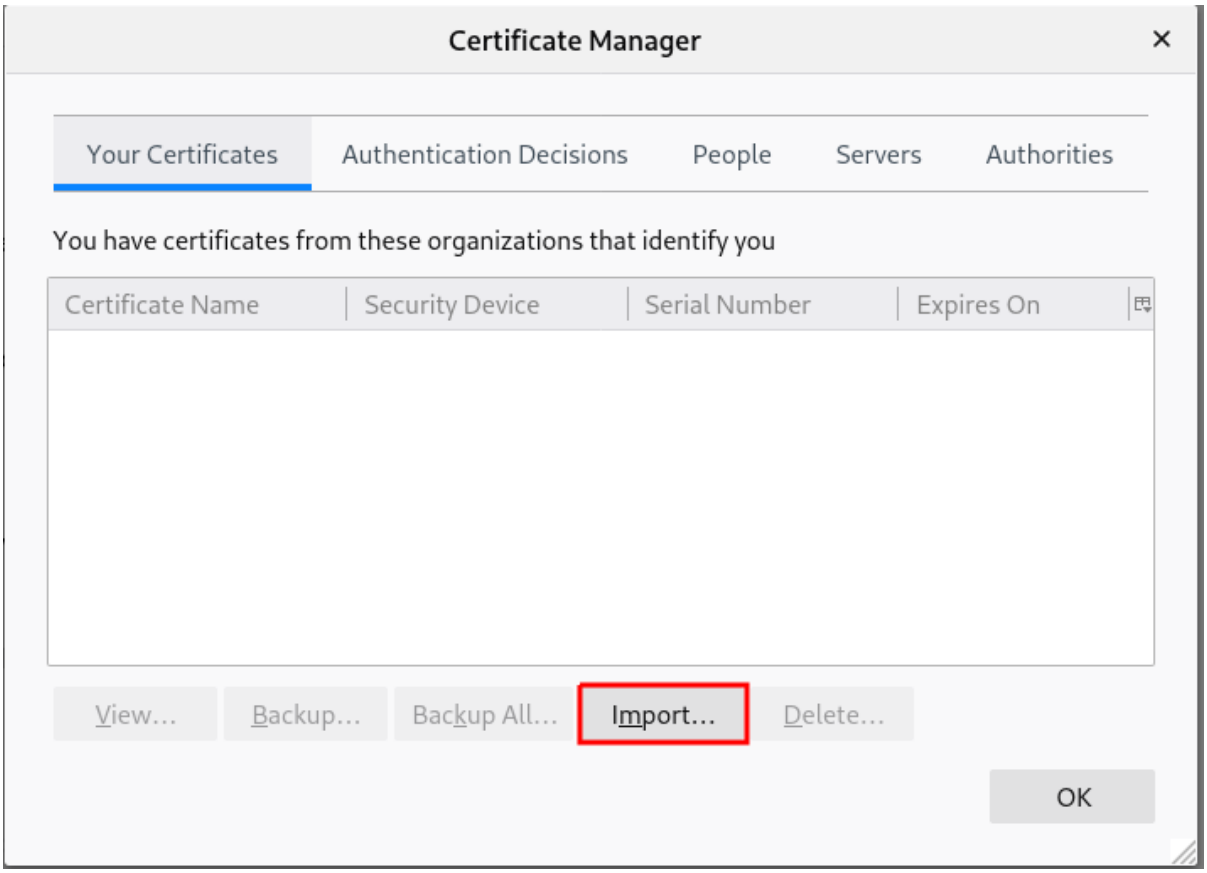
1. *개인 인증서가 장치에 저장되어 있습니다.*

인증을 위해 개인 인증서를 사용하려면 다음을 수행합니다.

절차

1. ***Open Certificate Manager** (인증서 관리자 열기).*
2. ***Your Certificates**(인증서) 탭에서 **Import**(가져오기)를 클릭합니다.*

그림 13.10. Thunderbird에서 인증을 위한 개인 인증서 가져오기

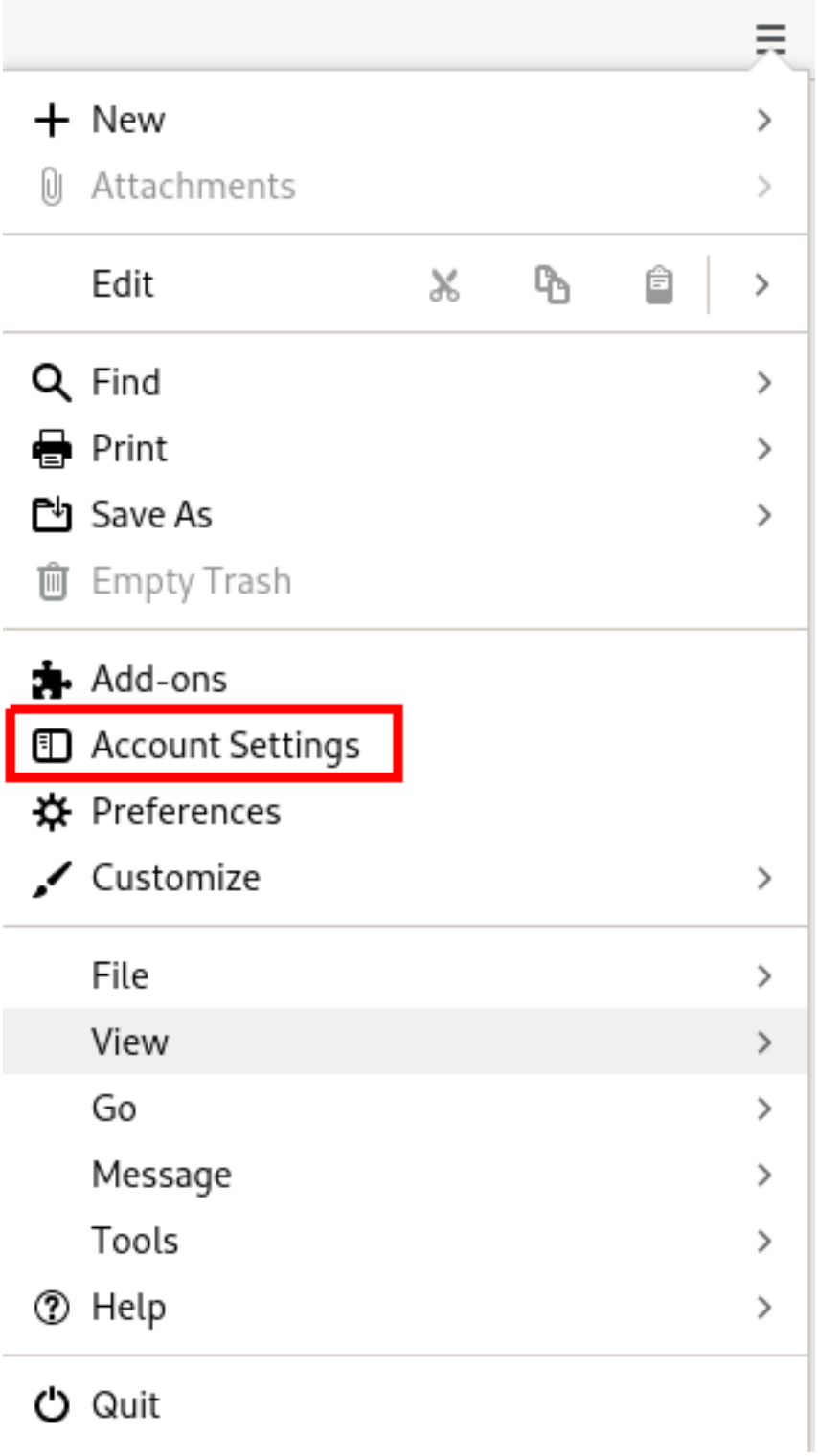


3. *컴퓨터에서 필요한 인증서를 선택합니다.*

4. ***Certificate Manager** 를 닫습니다.*

5. *메인 메뉴를 열고 **Account Settings** (계정 설정)를 선택합니다.*

그림 13.11. 메뉴에서 계정 설정 선택



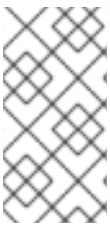
6. *계정 이메일 주소의 왼쪽 패널에서 **End-To-End Encryption** 을 선택합니다.*

   엔드 투 엔드 암호화 섹션 선택.

7. ***S/유럽 섹션에서 첫 번째 선택 버튼을 클릭하여 메시지에 서명하는 데 사용할 개인 인증서를 선택합니다.***

8. ***S/유럽 섹션에서 두 번째 선택 버튼을 클릭하여 메시지를 암호화하고 해독할 개인 인증서를 선택합니다.***

서명 및 암호화/암호 해독을 위한 인증서 선택.



참고

*유효한 인증서 가져오기를 잊은 경우 **Manage S/✓** 인증서를 사용하여 직접 **Certificate Manager** 를 열 수 있습니다.*