



## Red Hat Enterprise Linux 9

# Windows Active Directory와 RHEL 시스템을 직 접 통합

AD에 RHEL 호스트에 가입하고 AD에서 리소스에 액세스



# Red Hat Enterprise Linux 9 Windows Active Directory와 RHEL 시스템을 직접 통합

---

AD에 RHEL 호스트에 가입하고 AD에서 리소스에 액세스

## 법적 공지

Copyright © 2024 Red Hat, Inc.

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

관리자는 SSSD(System Security Services Daemon) 또는 Samba Winbind 서비스를 사용하여 AD 리소스에 액세스하여 Red Hat Enterprise Linux (RHEL) 호스트를 AD(Active Directory) 도메인에 연결할 수 있습니다. 또는 MSA(Managed Service Account)를 사용하여 도메인 통합 없이 AD 리소스에 액세스할 수도 있습니다.

## 차례

<b>RED HAT 문서에 관한 피드백 제공</b> .....	<b>3</b>
<b>1장. SSSD를 사용하여 RHEL 시스템을 AD에 직접 연결</b> .....	<b>4</b>
1.1. SSSD를 사용한 직접 통합 개요	4
1.2. 직접 통합을 위해 지원되는 WINDOWS 플랫폼	5
1.3. AD에 직접 연결	5
1.4. AD 공급자가 동적 DNS 업데이트를 처리하는 방법	10
1.5. AD 공급자의 동적 DNS 설정 수정	11
1.6. AD 공급자가 신뢰할 수 있는 도메인을 처리하는 방법	11
1.7. SSSD를 사용하여 ACTIVE DIRECTORY 사이트 자동 검색 재정의	12
1.8. 영역 명령	13
<b>2장. SAMBA WINBIND를 사용하여 AD에 RHEL 시스템 직접 연결</b> .....	<b>15</b>
2.1. SAMBA WINBIND를 사용한 직접 통합 개요	15
2.2. 직접 통합을 위해 지원되는 WINDOWS 플랫폼	16
2.3. AD 도메인에 RHEL 시스템 연결	17
2.4. 영역 명령	20
<b>3장. RHEL 시스템 역할을 사용하여 RHEL 시스템을 AD에 직접 통합</b> .....	<b>22</b>
3.1. AD_INTEGRATION RHEL 시스템 역할	22
<b>4장. AD에 대한 직접 연결 관리</b> .....	<b>23</b>
4.1. 기본 KERBEROS 호스트 키 탭 갱신 간격 수정	23
4.2. AD 도메인에서 RHEL 시스템 제거	24
4.3. SSSD에서 짧은 AD 사용자 이름을 확인하도록 도메인 확인 순서 설정	25
4.4. 도메인 사용자의 로그인 권한 관리	26
4.5. RHEL에서 그룹 정책 오브젝트 액세스 제어 적용	30
<b>5장. 관리 서비스 계정을 사용하여 AD에 액세스</b> .....	<b>39</b>
5.1. 관리형 서비스 계정의 이점	39
5.2. RHEL 호스트에 대한 관리형 서비스 계정 구성	40
5.3. 관리형 서비스 계정의 암호 업데이트	42
5.4. 관리형 서비스 계정 사양	43
5.5. ADCLI CREATE-MSA 명령의 옵션	44



## RED HAT 문서에 관한 피드백 제공

문서에 대한 피드백에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

### Jira를 통해 피드백 제출 (등록 필요)

1. [Jira](#) 웹 사이트에 로그인합니다.
2. 상단 탐색 모음에서 **생성** 을 클릭합니다.
3. **Summary** (요약) 필드에 설명 제목을 입력합니다.
4. **Description** (설명) 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. 대화 상자 하단에서 **생성** 을 클릭합니다.

## 1장. SSSD를 사용하여 RHEL 시스템을 AD에 직접 연결

RHEL 시스템을 AD(Active Directory)에 연결하려면 두 가지 구성 요소가 필요합니다. SSSD는 중앙 ID 및 인증 소스와 상호 작용하며 다른 구성 요소인 **realmd** 는 사용 가능한 도메인을 감지하고 기본 RHEL 시스템 서비스(이 경우 SSSD)를 구성하여 도메인에 연결합니다.

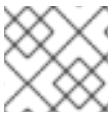
이 섹션에서는 SSSD(System Security Services Daemon)를 사용하여 RHEL 시스템을 AD(Active Directory)에 연결하는 방법을 설명합니다.

- [SSSD를 사용한 직접 통합 개요](#)
- [직접 통합을 위해 지원되는 Windows 플랫폼](#)
- [AD에 직접 연결](#)
- [AD 공급자가 동적 DNS 업데이트를 처리하는 방법](#)
- [AD 공급자의 동적 DNS 설정 수정](#)
- [AD 공급자가 신뢰할 수 있는 도메인을 처리하는 방법](#)
- [SSSD를 사용하여 Active Directory 사이트 자동 검색 재정의](#)
- [영역 명령](#)

### 1.1. SSSD를 사용한 직접 통합 개요

SSSD를 사용하여 오프라인 로그인을 허용하기 위해 사용자 캐싱과 함께 공통 프레임워크를 통해 인증 및 권한 부여를 위해 사용자 디렉터리에 액세스합니다. SSSD는 고도로 구성할 수 있습니다. 이 모듈은 PAM(Pluggable Authentication Modules) 및 NSS(Name Switch Service) 통합과 로컬 사용자를 저장하는 데이터베이스 및 중앙 서버에서 검색된 확장된 사용자 데이터를 제공합니다. SSSD는 다음 유형의 ID 서버 중 하나를 사용하여 RHEL 시스템을 연결하는 권장 구성 요소입니다.

- Active Directory
- RHEL의 IdM(Identity Management)
- 일반 LDAP 또는 Kerberos 서버



#### 참고

SSSD와 직접 통합은 기본적으로 단일 AD forest 내에서만 작동합니다.

AD와 Linux 시스템을 직접 통합하도록 SSSD를 구성하는 가장 편리한 방법은 **realmd** 서비스를 사용하는 것입니다. 호출자는 네트워크 인증 및 도메인 멤버십을 표준 방식으로 구성할 수 있습니다. **realmd** 서비스는 액세스 가능한 도메인 및 영역에 대한 정보를 자동으로 검색하고 도메인 또는 영역에 조인하기 위해 고급 구성이 필요하지 않습니다.

SSSD를 AD와 직접 및 간접 통합에 모두 사용할 수 있으며 하나의 통합 방법에서 다른 방법으로 전환할 수 있습니다. 직접 통합은 RHEL 시스템을 AD 환경에 소개하는 간단한 방법입니다. 그러나 RHEL 시스템의 공유가 증가함에 따라 배포 시 호스트 기반 액세스 제어, sudo 또는 SELinux 사용자 매핑과 같은 ID 관련 정책을 보다 효율적으로 관리해야 합니다. 처음에 RHEL 시스템의 이러한 측면을 로컬 구성 파일에서 유지 관리할 수 있습니다. 그러나 시스템의 수가 증가함에 따라 Red Hat Satellite와 같은 프로비저닝 시스템을 사용하면 구성 파일의 배포 및 관리가 쉬워집니다. 직접 통합이 더 이상 확장되지 않는 경우 간접 통



합을 고려해야 합니다. 직접 통합(RHEL 클라이언트가 AD 도메인에 있음)에서 간접 통합( trust to AD로 IdM)으로 이동하는 방법에 대한 자세한 내용은 [AD 도메인에서 IdM 서버로 RHEL 클라이언트 이동을 참조하십시오.](#)



### 중요

IdM이 FIPS 모드인 경우 IdM-AD 통합은 RC4 또는 AES HMAC-SHA1 암호화만 지원하는 경우에만 AD로 인해 작동하지 않지만 FIPS 모드의 RHEL 9에서는 기본적으로 AES HMAC-SHA2만 허용합니다. RHEL 9에서 AES HMAC-SHA1 사용을 활성화하려면 **# update-crypto-policies --set FIPS:AD-SUPPORT** 를 입력합니다.

IdM은 Common Criteria 평가 시스템에서만 사용해야 하는 보다 제한적인 **FIPS:OSPP** 암호화 정책을 지원하지 않습니다.

사용 사례에 맞는 통합 유형에 대한 자세한 내용은 [간접 통합 및 직접 통합](#) 정의를 참조하십시오.

### 추가 리소스

- **realm(8)** 도움말 페이지.
- **sssd-ad(5)** 도움말 페이지.
- **sssd(8)** 도움말 페이지.

## 1.2. 직접 통합을 위해 지원되는 WINDOWS 플랫폼

다음 Specest 및 도메인 기능 수준을 사용하는 Active Directory와 RHEL 시스템을 직접 통합할 수 있습니다.

- 도메인 기능 수준 범위: Windows Server 2008 - Windows Server 2016
- 도메인 기능 수준 범위: Windows Server 2008 - Windows Server 2016

다음과 같은 지원되는 운영 체제에서 직접 통합되었습니다.

- Windows Server 2022 (RHEL 9.1 이상)
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2



### 참고

Windows Server 2019 및 Windows Server 2022는 새로운 기능 수준을 도입하지 않습니다. 기능 수준 Windows Server 2019 및 Windows Server 2022 용도는 Windows Server 2016입니다.

## 1.3. AD에 직접 연결

SSSD(System Security Services Daemon)는 RHEL(Red Hat Enterprise Linux) 시스템을 AD(Active Directory)에 연결하는 데 권장되는 구성 요소입니다. 이 섹션에서는 SSSD의 기본 ID 매핑 또는 POSIX 속성을 사용하여 AD와 직접 통합하는 방법을 설명합니다.

- AD와 통합하기 위한 옵션: ID 매핑 또는 POSIX 특성 사용
- SSSD를 사용하여 AD 도메인 검색 및 참여
- Active Directory에 정의된 POSIX 특성을 사용하여 AD에 연결
- SSSD를 사용하여 다양한 AD forest의 여러 도메인에 연결

**중요**

시스템에 AD에 가입하기 전에 [Basic Prechecks 단계의 절차에 따라 RHEL이 'adcli', 'realm' 및 'net' 명령을 사용하여 Active Directory와 함께 시스템을 올바르게 구성해야 합니다.](#)

### 1.3.1. AD와 통합하기 위한 옵션: ID 매핑 또는 POSIX 특성 사용

Linux 및 Windows 시스템은 사용자 및 그룹에 다른 식별자를 사용합니다.

- Linux는 UID( *사용자 ID* ) 및 그룹 ID (GID)를 사용합니다. [기본 시스템 설정 구성에서 사용자 및 그룹 계정 관리 소개](#) 를 참조하십시오. Linux UID 및 GID는 POSIX 표준을 준수합니다.
- Windows는 SID( *보안 ID* )를 사용합니다.



**중요**

RHEL 시스템을 AD에 연결한 후 AD 사용자 이름과 암호로 인증할 수 있습니다. 중복 이름으로 인해 충돌이 발생하고 인증 프로세스가 중단될 수 있으므로 Windows 사용자와 동일한 Linux 사용자를 생성하지 마십시오.

AD 사용자로 RHEL 시스템에 인증하려면 UID 및 GID가 할당되어 있어야 합니다. SSSD는 ID 매핑 또는 POSIX 속성을 사용하여 AD와 통합할 수 있는 옵션을 제공합니다. 기본값은 ID 매핑을 사용하는 것입니다.

#### AD 사용자에게 대한 새 UID 및 GID 자동 생성

SSSD는 AD 사용자의 SID를 사용하여 ID 매핑이라는 프로세스에서 POSIX ID를 알고리즘화할 수 있습니다. ID 매핑은 Linux의 AD ID와 AD의 SID 간 맵을 생성합니다.

- SSSD가 새 AD 도메인을 감지하면 사용 가능한 다양한 ID를 새 도메인에 할당합니다.
- AD 사용자가 SSSD 클라이언트 시스템에 처음으로 로그인하면 SSSD 캐시에 사용자 SID 및 해당 도메인의 ID 범위를 기반으로 UID를 포함하여 사용자에게 대한 항목을 만듭니다.
- AD 사용자의 ID는 동일한 SID에서 일관된 방식으로 생성되므로 사용자는 Red Hat Enterprise Linux 시스템에 로그인할 때 동일한 UID 및 GID를 갖습니다.

[SSSD를 사용하여 AD 도메인 검색 및 참여](#)를 참조하십시오.



**참고**

모든 클라이언트 시스템에서 SSSD를 사용하여 SID를 Linux ID에 매핑하면 매핑이 일관되게 유지됩니다. 일부 클라이언트가 다른 소프트웨어를 사용하는 경우 다음 중 하나를 선택합니다.

- 모든 클라이언트에서 동일한 매핑 알고리즘이 사용되는지 확인합니다.
- AD에 정의된 명시적 POSIX 특성을 사용합니다.

## AD에 정의된 POSIX 특성 사용

AD는 `uidNumber`, `gidNumber`, `unixHomeDirectory` 또는 `loginShell` 과 같은 POSIX 특성을 생성하고 저장할 수 있습니다.

위에서 설명한 ID 매핑을 사용하는 경우 SSSD는 새 UID와 GID를 생성하여 AD에 정의된 값을 덮어씁니다. AD 정의 값을 유지하려면 SSSD에서 ID 매핑을 비활성화해야 합니다.

[Active Directory에 정의된 POSIX 특성을 사용하여 AD에 연결을](#) 참조하십시오.

### 1.3.2. SSSD를 사용하여 AD 도메인 검색 및 참여

다음 절차에 따라 AD 도메인을 검색하고 SSSD를 사용하여 RHEL 시스템을 해당 도메인에 연결합니다.

#### 사전 요구 사항

- AD 도메인 컨트롤러에서 다음 포트가 열려 있고 RHEL 호스트에서 액세스할 수 있는지 확인합니다.

표 1.1. SSSD를 사용하여 Linux 시스템을 AD로 직접 통합하는 데 필요한 포트

서비스	포트	프로토콜	참고
DNS	53	UDP 및 TCP	
LDAP	389	UDP 및 TCP	
samba	445	UDP 및 TCP	AD Group Policy Objects(GPO)의 경우
Kerberos	88	UDP 및 TCP	
Kerberos	464	UDP 및 TCP	<b>kadmin</b> 에서 암호 설정 및 변경에 사용
LDAP 글로벌 카탈로그	3268	TCP	<b>id_provider = ad</b> 옵션을 사용하는 경우
NTP	123	UDP	선택 사항

- DNS에 AD 도메인 컨트롤러 서버를 사용하고 있는지 확인합니다.
- 두 시스템의 시스템 시간이 동기화되었는지 확인합니다. 이렇게 하면 Kerberos가 올바르게 작동할 수 있습니다.

#### 절차

- 다음 패키지를 설치합니다.

```
# dnf install samba-common-tools realmd oddjob oddjob-mkhomedir sssd adcli krb5-workstation
```

2. 특정 도메인에 대한 정보를 표시하려면 **영역 검색을 실행하고 검색할 도메인의 이름을 추가**합니다.

```
# realm discover ad.example.com
ad.example.com
type: kerberos
realm-name: AD.EXAMPLE.COM
domain-name: ad.example.com
configured: no
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common
```

**realmd** 시스템은 DNS SRV lookups를 사용하여 이 도메인의 도메인 컨트롤러를 자동으로 찾습니다.



**참고**

**realmd** 시스템은 Active Directory와 Identity Management 도메인을 모두 검색할 수 있습니다. 두 도메인이 모두 사용자 환경에 있는 경우 **--server-software=active-directory** 옵션을 사용하여 검색 결과를 특정 유형의 서버로 제한할 수 있습니다.

3. **realm join** 명령을 사용하여 로컬 RHEL 시스템을 구성합니다. **realmd** 모음은 필요한 모든 구성 파일을 자동으로 편집합니다. 예를 들어 **ad.example.com** 이라는 도메인의 경우:

```
# realm join ad.example.com
```

**검증 단계**

- 관리자와 같은 AD 사용자 세부 정보를 표시합니다.

```
# getent passwd administrator@ad.example.com
administrator@ad.example.com:*:1450400500:1450400513:Administrator:/home/administrator
@ad.example.com:/bin/bash
```

**추가 리소스**

- **realm(8)** 매뉴얼 페이지를 참조하십시오.
- **nmcli(1)** 매뉴얼 페이지를 참조하십시오.

**1.3.3. Active Directory에 정의된 POSIX 특성을 사용하여 AD에 연결**

최상의 성능을 위해 POSIX 특성을 AD 글로벌 카탈로그에 게시합니다. 글로벌 카탈로그에 POSIX 속성이 없는 경우 SSSD는 LDAP 포트의 개별 도메인 컨트롤러에 직접 연결합니다.

**사전 요구 사항**

- RHEL 호스트의 다음 포트가 열려 있고 AD 도메인 컨트롤러에서 액세스할 수 있는지 확인합니다.

표 1.2. SSSD를 사용하여 Linux 시스템을 AD로 직접 통합하는 데 필요한 포트

서비스	포트	프로토콜	참고
DNS	53	UDP 및 TCP	
LDAP	389	UDP 및 TCP	
Kerberos	88	UDP 및 TCP	
Kerberos	464	UDP 및 TCP	kadmin에서 암호 설정 및 변경에 사용
LDAP 글로벌 카탈로그	3268	TCP	<b>id_provider = ad</b> 옵션을 사용하는 경우
NTP	123	UDP	선택 사항

- DNS에 AD 도메인 컨트롤러 서버를 사용하고 있는지 확인합니다.
- 두 시스템의 시스템 시간이 동기화되었는지 확인합니다. 이렇게 하면 Kerberos가 올바르게 작동할 수 있습니다.

## 절차

1. 다음 패키지를 설치합니다.

```
# dnf install realmd oddjob oddjob-mkhomedir sssd adcli krb5-workstation
```

2. zone **join** 명령을 **--automatic-id-mapping=no** 옵션과 함께 사용하여 ID 매핑이 비활성화된 로컬 RHEL 시스템을 구성합니다. **realmd** 모음은 필요한 모든 구성 파일을 자동으로 편집합니다. 예를 들어 **ad.example.com** 이라는 도메인의 경우:

```
# realm join --automatic-id-mapping=no ad.example.com
```

3. 도메인에 이미 가입한 경우 SSSD에서 ID 매핑을 수동으로 비활성화할 수 있습니다.

- a. **/etc/sss/sss.conf** 파일을 엽니다.

- b. AD 도메인 섹션에서 **ldap\_id\_mapping = false** 설정을 추가합니다.

- c. SSSD 캐시를 제거합니다.

```
rm -f /var/lib/sss/db/*
```

- d. SSSD를 다시 시작합니다.

```
systemctl restart sssd
```

SSSD는 이제 로컬에서 생성하는 대신 AD에서 POSIX 속성을 사용합니다.



**참고**

AD의 사용자에 대해 관련 POSIX 속성(**uidNumber, gidNumber, unixHomeDirectory, loginShell**)이 구성되어 있어야 합니다.

**검증 단계**

- 관리자와 같은 AD 사용자 세부 정보를 표시합니다.

```
# getent passwd administrator@ad.example.com
administrator@ad.example.com:*:10000:10000:Administrator:/home/Administrator:/bin/bash
```

**추가 리소스**

- ID 매핑 및 **ldap\_id\_mapping** 매개변수에 대한 자세한 내용은 **sssd-ldap(8)** 매뉴얼 페이지를 참조하십시오.

**1.3.4. SSSD를 사용하여 다양한 AD forest의 여러 도메인에 연결**

AD(Active Directory) 관리 서비스 계정(MSA)을 사용하여 서로 다른 포리스트의 AD 도메인에 액세스할 수 있습니다. You can use an Active Directory (AD) Managed Service Account (MSA) to access AD domains from different forests where there is no trust between them.

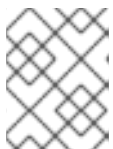
[관리 서비스 계정을 사용하여 AD 액세스 단원을 참조하십시오.](#)

**1.4. AD 공급자가 동적 DNS 업데이트를 처리하는 방법**

Active Directory (AD)는 시간 초과 (조장) 및 비활성 레코드를 제거하여 DNS 레코드를 적극적으로 유지합니다.

기본적으로 SSSD 서비스는 다음과 같은 간격으로 RHEL 클라이언트의 DNS 레코드를 새로 고칩니다.

- ID 공급자가 온라인 상태가 될 때마다
- RHEL 시스템이 재부팅될 때마다.
- **/etc/sss/sss.conf** 설정 파일의 **dyndns\_refresh\_interval** 옵션으로 지정된 간격에서 다음을 수행합니다. 기본값은 **86400** 초(24시간)입니다.



**참고**

**dyndns\_refresh\_interval** 옵션을 DHCP 리스와 동일한 간격으로 설정하면 IP 리스가 갱신된 후 DNS 레코드를 업데이트할 수 있습니다.

SSSD는 DNS(GSS-TSIG)용 Kerberos/GSSAPI를 사용하여 AD 서버로 동적 DNS 업데이트를 보냅니다. 즉, AD에 대한 보안 연결을 사용하도록 설정해야 합니다.

**추가 리소스**

- **sssd-ad(5)** 도움말 페이지.

## 1.5. AD 공급자의 동적 DNS 설정 수정

SSSD(System Security Services Daemon) 서비스는 기본 간격으로 AD 환경에 연결된 RHEL(Red Hat Enterprise Linux) 클라이언트의 DNS 레코드를 새로 고칩니다. 다음 절차에서는 이러한 간격을 조정합니다.

### 사전 요구 사항

- SSSD 서비스를 사용하여 Active Directory 환경에 RHEL 호스트에 참여했습니다.
- `/etc/sss/sss.conf` 설정 파일을 편집하려면 루트 권한이 필요합니다.

### 절차

1. 텍스트 편집기에서 `/etc/sss/sss.conf` 설정 파일을 엽니다.
2. AD 도메인의 **[domain]** 섹션에 DNS 레코드 새로 고침 간격을 12시간으로 설정하고, PTR 레코드 업데이트를 비활성화하며, DNS 레코드 TTL(TTL)을 1시간으로 설정합니다.

```
[domain/ad.example.com]
id_provider = ad
...
dyndns_refresh_interval = 43200
dyndns_update_ptr = false
dyndns_ttl = 3600
```

3. `/etc/sss/sss.conf` 설정 파일을 저장하고 닫습니다.
4. SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

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

### 참고

`sss.conf` 파일에서 `dyndns_update` 옵션을 **false** 로 설정하여 동적 DNS 업데이트를 비활성화할 수 있습니다.

```
[domain/ad.example.com]
id_provider = ad
...
dyndns_update = false
```

### 추가 리소스

- [AD 공급자가 동적 DNS 업데이트를 처리하는 방법](#)
- `sss-ad(5)` 매뉴얼 페이지

## 1.6. AD 공급자가 신뢰할 수 있는 도메인을 처리하는 방법

`/etc/sss/sss.conf` 구성 파일에서 `id_provider = ad` 옵션을 설정하면 SSSD가 신뢰할 수 있는 도메인을 다음과 같이 처리합니다.

- SSSD는 단일 AD forest의 도메인만 지원합니다. SSSD에서 여러 개의 포리스트에서 여러 도메인에 액세스해야 하는 경우 SSSD 대신 신뢰(preferred) 또는 **winbindd** 서비스를 사용하는 것이 좋습니다.
- 기본적으로 SSSD는 버스트에 있는 모든 도메인을 검색하고, 신뢰할 수 있는 도메인의 개체에 대한 요청이 도착하면 SSSD에서 이 도메인을 해결합니다.  
신뢰할 수 있는 도메인에 연결할 수 없거나 지리적으로 멀리 있지 않으면 속도가 느려지면 **/etc/sss/sss.conf**에서 **ad\_enabled\_domains** 매개변수를 설정하여 신뢰할 수 있는 도메인 SSSD에서 개체를 확인할 수 있습니다.
- 기본적으로 정규화된 사용자 이름을 사용하여 신뢰할 수 있는 도메인에서 사용자를 확인해야 합니다.

## 추가 리소스

- **sss.conf(5)** 도움말 페이지.

## 1.7. SSSD를 사용하여 ACTIVE DIRECTORY 사이트 자동 검색 재정의

Active Directory (AD)는 매우 클 수 있으며, 다양한 도메인 컨트롤러, 도메인, 하위 도메인 및 물리적 사이트와 함께 매우 클 수 있습니다. AD는 **사이트** 개념을 사용하여 도메인 컨트롤러의 물리적 위치를 식별합니다. 이를 통해 클라이언트는 지리적으로 가장 가까운 도메인 컨트롤러에 연결할 수 있으므로 클라이언트 성능이 향상됩니다.

이 섹션에서는 SSSD에서 자동 검색을 사용하여 AD 사이트를 찾아 연결하는 방법과 자동 검색을 재정의하고 사이트를 수동으로 지정하는 방법을 설명합니다.

### 1.7.1. SSSD에서 AD 사이트 자동 검색을 처리하는 방법

기본적으로 SSSD 클라이언트는 자동 검색을 사용하여 AD 사이트를 찾고 가장 가까운 도메인 컨트롤러에 연결합니다. 프로세스는 다음 단계로 구성됩니다.

1. SSSD는 SRV 쿼리를 수행하여 도메인에서 도메인 컨트롤러(DC)를 찾습니다. SSSD는 SSSD 구성 파일의 **dns\_discovery\_domain** 또는 **ad\_domain** 옵션에서 검색 도메인을 읽습니다.
2. SSSD는 너무 많은 DC를 ping하고 연결할 수 없는 DC에서 이 DC에 대해 CLDAP(Connection-Less LDAP) ping을 수행합니다. SSSD가 이러한 배치 중 사이트 및 색인 정보를 수신하는 경우 나머지 배치를 건너뛵니다.
3. SSSD는 사이트별 및 백업 서버 목록을 생성하고 저장합니다.

### 1.7.2. AD 사이트 자동 검색 덮어쓰기

자동 검색 프로세스를 재정의하려면 **ad\_site** 옵션을 **/etc/sss/sss.conf** 파일의 **[domain]** 섹션에 추가하여 클라이언트를 연결할 AD 사이트를 지정합니다. 이 예에서는 클라이언트가 **ExampleSite** AD 사이트에 연결하도록 구성합니다.

## 사전 요구 사항

- SSSD 서비스를 사용하여 RHEL 호스트에 Active Directory 환경에 가입했습니다.
- **/etc/sss/sss.conf** 구성 파일을 편집할 수 있도록 **root** 사용자로 인증할 수 있습니다.

## 절차



1. 텍스트 편집기에서 `/etc/sss/sss.conf` 파일을 엽니다.
2. AD 도메인의 `[domain]` 섹션에 `ad_site` 옵션을 추가합니다.

```
[domain/ad.example.com]
id_provider = ad
...
ad_site = ExampleSite
```

3. `/etc/sss/sss.conf` 설정 파일을 저장하고 닫습니다.
4. SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.

```
# systemctl restart sssd
```

## 1.8. 영역 명령

`realmd` 시스템에는 두 가지 주요 작업 영역이 있습니다.

- 도메인의 시스템 등록 관리.
- 로컬 시스템 리소스에 액세스할 수 있는 도메인 사용자를 제어합니다.

`realmd` 에서 명령줄 툴 **영역**을 사용하여 명령을 실행합니다. 대부분의 `realm` 명령을 실행하려면 유틸리티에서 수행해야 하는 작업과 작업을 수행할 도메인 또는 사용자 계정과 같은 엔티티를 지정해야 합니다.

표 1.3. `realmd` Commands

명령	설명
영역 명령	
discover	네트워크에서 도메인에 대한 검색 검사를 실행합니다.
join	지정된 도메인에 시스템을 추가합니다.
자주 묻는 질문	지정된 도메인에서 시스템을 제거합니다.
list	시스템에 대해 구성된 모든 도메인 또는 검색 및 구성된 모든 도메인을 나열합니다.
로그인 명령	
허용	특정 사용자 또는 구성된 도메인 내 모든 사용자에게 대한 액세스를 활성화하여 로컬 시스템에 액세스합니다.
deny	특정 사용자 또는 구성된 도메인 내 모든 사용자에게 대한 액세스 권한을 제한하여 로컬 시스템에 액세스합니다.

추가 리소스

- **realm(8)** 도움말 페이지.

## 2장. SAMBA WINBIND를 사용하여 AD에 RHEL 시스템 직접 연결

RHEL 시스템을 AD에 연결하려면 두 가지 구성 요소가 필요합니다. 하나의 구성 요소인 Samba Winbind는 AD ID 및 인증 소스와 상호 작용하고, 사용 가능한 도메인을 감지하여 기본 RHEL 시스템 서비스를 구성하고 기본 RHEL 시스템 서비스를 구성하여 AD 도메인에 연결합니다.

이 섹션에서는 **Samba Winbind**를 사용하여 **RHEL** 시스템을 **AD(Active Directory)**에 연결하는 방법을 설명합니다.

- [Samba Winbind를 사용한 직접 통합 개요](#)
- [직접 통합을 위해 지원되는 Windows 플랫폼](#)
- [AD 도메인에 RHEL 시스템 연결](#)
- [영역 명령](#)

### 2.1. SAMBA WINBIND를 사용한 직접 통합 개요

**Samba Winbind**는 **Linux** 시스템에서 **Windows** 클라이언트를 에뮬레이션하고 **AD** 서버와 통신합니다.

**realmd** 서비스를 사용하여 다음을 통해 **Samba Winbind**를 구성할 수 있습니다.

- 표준 방식으로 네트워크 인증 및 도메인 멤버십 구성.
- 액세스 가능한 도메인 및 영역에 대한 정보를 자동으로 검색합니다.
- 고급 구성이 도메인 또는 영역에 가입할 필요가 없습니다.

다음 사항에 유의하십시오.

- 다중 **Forest AD** 설정에서 **Winbind**와 직접 통합하려면 양방향 신뢰가 필요합니다.
- **remote forests**는 **local forests**를 신뢰하여 **idmap\_ad** 플러그인이 원격 forest 사용자를 올바르게 처리할 수 있도록 해야 합니다.

**Samba**의 **winbindd** 서비스는 **NSS(Name Service Switch)**에 대한 인터페이스를 제공하고 도메인 사용자가 로컬 시스템에 로그인할 때 **AD**를 인증할 수 있도록 합니다.

**winbindd**를 사용하면 추가 소프트웨어를 설치하지 않고도 디렉터리 및 프린터를 공유하는 구성을 향상시킬 수 있는 이점이 있습니다. 자세한 내용은 [다른 유형의 서버 배포 가이드의 Samba를 서버로 사용하는 방법에 대한 섹션을 참조하십시오.](#)

#### 추가 리소스

- **realmd man** 페이지를 참조하십시오.
- **winbindd man** 페이지를 참조하십시오.

## 2.2. 직접 통합을 위해 지원되는 WINDOWS 플랫폼

다음 **Specest** 및 도메인 기능 수준을 사용하는 **Active Directory**와 **RHEL** 시스템을 직접 통합할 수 있습니다.

- 도메인 기능 수준 범위: **Windows Server 2008 - Windows Server 2016**
- 도메인 기능 수준 범위: **Windows Server 2008 - Windows Server 2016**

다음과 같은 지원되는 운영 체제에서 직접 통합되었습니다.

- **Windows Server 2022 (RHEL 9.1 이상)**
- **Windows Server 2019**

- **Windows Server 2016**
- **Windows Server 2012 R2**



#### 참고

**Windows Server 2019 및 Windows Server 2022**는 새로운 기능 수준을 도입하지 않습니다. 기능 수준 **Windows Server 2019 및 Windows Server 2022** 용도는 **Windows Server 2016**입니다.

### 2.3. AD 도메인에 RHEL 시스템 연결

**Samba Winbind**는 **RHEL(Red Hat Enterprise Linux)** 시스템을 **AD(Active Directory)**와 연결하기 위한 **SSSD(System Security Services Daemon)**의 대안입니다. **realmd** 를 사용하여 **Samba Winbind**를 구성하여 **RHEL** 시스템을 **AD** 도메인에 연결할 수 있습니다.

#### 절차

1. **AD에 Kerberos** 인증을 위한 더 이상 사용되지 않는 **RC4** 암호화 유형이 필요한 경우 **RHEL**에서 이러한 암호에 대한 지원을 활성화합니다.

```
# update-crypto-policies --set DEFAULT:AD-SUPPORT
```

2. 다음 패키지를 설치합니다.

```
# dnf install realmd oddjob-mkhomedir oddjob samba-winbind-clients \
samba-winbind samba-common-tools samba-winbind-krb5-locator
```

3. 도메인 구성원에서 디렉터리 또는 프린터를 공유하려면 **samba** 패키지를 설치합니다.

```
# dnf install samba
```

4. 기존 **/etc/samba/smb.conf** **Samba** 구성 파일을 백업합니다.

```
# mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

- 5.

도메인에 가입합니다. 예를 들어 `ad.example.com`이라는 도메인에 가입하려면 다음을 수행합니다.

```
# realm join --membership-software=samba --client-software=winbind ad.example.com
```

이전 명령을 사용하면 `realm` 유틸리티가 자동으로 수행됩니다.

- `ad.example.com` 도메인 멤버십에 대한 `/etc/samba/smb.conf` 파일을 만듭니다.
  - 사용자 및 그룹 조회에 대한 `winbind` 모듈을 `/etc/nsswitch.conf` 파일에 추가합니다.
  - `/etc/pam.d/` 디렉토리에서 **PAM(Pluggable Authentication Module)** 구성 파일을 업데이트합니다.
  - `winbind` 서비스를 시작하고 시스템이 부팅될 때 서비스가 시작됩니다.
6. 선택적으로 `/etc/samba/smb.conf` 파일에서 대체 ID 매핑 백엔드 또는 사용자 지정된 ID 매핑 설정을 설정합니다.

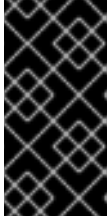
자세한 내용은 [Samba ID 매핑 이해 및 구성](#)을 참조하십시오.

1. `/etc/krb5.conf` 파일을 편집하고 다음 섹션을 추가합니다.

```
[plugins]
localauth = {
    module = winbind:/usr/lib64/samba/krb5/winbind_krb5_localauth.so
    enable_only = winbind
}
```

2. `winbind` 서비스가 실행 중인지 확인합니다.

```
# systemctl status winbind
...
Active: active (running) since Tue 2018-11-06 19:10:40 CET; 15s ago
```



## 중요

**Samba**를 활성화하여 도메인 사용자 및 그룹 정보를 쿼리하려면 **smb** 를 시작하기 전에 **winbind** 서비스를 실행해야 합니다.

3. 디렉터리 및 프린터를 공유하는 **samba** 패키지를 설치한 경우 **smb** 서비스를 활성화하고 시작합니다.

```
# systemctl enable --now smb
```

## 검증 단계

1. AD 도메인의 AD 관리자 계정과 같은 AD 사용자의 세부 정보를 표시합니다.

```
# getent passwd "AD\administrator"
AD\administrator*:10000:10000::/home/administrator@AD:/bin/bash
```

2. AD 도메인에서 도메인 사용자 그룹의 멤버를 쿼리합니다.

```
# getent group "AD\Domain Users"
AD\domain users:x:10000:user1,user2
```

3. 선택적으로 파일 및 디렉터리에 대한 권한을 설정할 때 도메인 사용자와 그룹을 사용할 수 있는지 확인합니다. 예를 들어 `/srv/samba/example.txt` 파일의 소유자를 **AD\administrator** 로 설정하고 그룹을 **AD\Domain Users** 로 설정하려면 다음을 수행합니다.

```
# chown "AD\administrator":"AD\Domain Users" /srv/samba/example.txt
```

4. Kerberos 인증이 예상대로 작동하는지 확인합니다.

- a. AD 도메인 멤버에서 **administrator@AD.EXAMPLE.COM** 주체의 티켓을 받습니다.

```
# kinit administrator@AD.EXAMPLE.COM
```

- b. 캐시된 Kerberos 티켓을 표시합니다.

```
# klist
```

```

Ticket cache: KCM:0
Default principal: administrator@AD.EXAMPLE.COM

Valid starting   Expires   Service principal
01.11.2018 10:00:00 01.11.2018 20:00:00
krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 08.11.2018 05:00:00

```

- 5. 사용 가능한 도메인을 표시합니다.

```

# wbinfo --all-domains
BUILTIN
SAMBA-SERVER
AD

```

추가 리소스

- 더 이상 사용되지 않는 RC4 암호를 사용하지 않으려면 AD에서 AES 암호화 유형을 활성화할 수 있습니다. [see](#)
- [GPO를 사용하여 Active Directory의 AES 암호화 유형 활성화](#)
- [realm\(8\) 도움말 페이지](#)

2.4. 영역 명령

realmd 시스템에는 두 가지 주요 작업 영역이 있습니다.

- 도메인의 시스템 등록 관리.
- 로컬 시스템 리소스에 액세스할 수 있는 도메인 사용자를 제어합니다.

realmd 에서 명령줄 틀 영역을 사용하여 명령을 실행합니다. 대부분의 realm 명령을 실행하려면 유틸리티에서 수행해야 하는 작업과 작업을 수행할 도메인 또는 사용자 계정과 같은 엔터티를 지정해야 합니다.

표 2.1. realmd Commands



명령	설명
영역 명령	
discover	네트워크에서 도메인에 대한 검색 검사를 실행합니다.
join	지정된 도메인에 시스템을 추가합니다.
자주 묻는 질문	지정된 도메인에서 시스템을 제거합니다.
list	시스템에 대해 구성된 모든 도메인 또는 검색 및 구성된 모든 도메인을 나열합니다.
로그인 명령	
허용	특정 사용자 또는 구성된 도메인 내 모든 사용자에게 대한 액세스를 활성화하여 로컬 시스템에 액세스합니다.
deny	특정 사용자 또는 구성된 도메인 내 모든 사용자에게 대한 액세스 권한을 제한하여 로컬 시스템에 액세스합니다.

#### 추가 리소스

- ***realm(8)* 도움말 페이지.**

### 3장. RHEL 시스템 역할을 사용하여 RHEL 시스템을 AD에 직접 통합

**ad\_integration** 시스템 역할을 사용하면 **Red Hat Ansible Automation Platform**을 사용하여 RHEL 시스템을 AD(Active Directory)와 직접 통합할 수 있습니다.

#### 3.1. AD\_INTEGRATION RHEL 시스템 역할

**ad\_integration** 시스템 역할을 사용하여 RHEL 시스템을 AD(Active Directory)에 직접 연결할 수 있습니다.

역할은 다음 구성 요소를 사용합니다.

- 중앙 ID 및 인증 소스와 상호 작용하는 **SSSD**
- 사용 가능한 AD 도메인을 감지하고 기본 RHEL 시스템 서비스(이 경우 **SSSD**)를 구성하여 선택한 AD 도메인에 연결합니다.



#### 참고

**ad\_integration** 역할은 **IdM(Identity Management)** 환경 없이 직접 AD 통합을 사용하는 배포용입니다. IdM 환경의 경우 **ansible-freeipa** 역할을 사용합니다.

#### 추가 리소스

- [/usr/share/ansible/roles/rhel-system-roles.ad\\_integration/README.md](#) 파일
- [/usr/share/doc/rhel-system-roles/ad\\_integration/ directory](#)
- [SSSD를 사용하여 RHEL 시스템을 AD에 직접 연결](#)

## 4장. AD에 대한 직접 연결 관리

**SSSD(System Security Services Daemon)** 또는 **Samba Winbind**를 사용하여 **RHEL(Red Hat Enterprise Linux)** 시스템을 **AD(Active Directory)**에 연결할 수 있습니다. 이 섹션에서는 **RHEL** 시스템이 **AD** 클라이언트로 이미 구성된 경우 **AD** 연결을 수정하고 관리하는 방법을 설명합니다.

### 사전 요구 사항

- **SSSD** 또는 **Samba Winbind**를 사용하여 **RHEL** 시스템을 **Active Directory** 도메인에 연결했습니다.

### 4.1. 기본 **KERBEROS** 호스트 키 탭 갱신 간격 수정

**adcli** 패키지가 설치된 경우 **SSSD**는 **AD** 환경에서 **Kerberos** 호스트 키 탭 파일을 자동으로 갱신합니다. 데몬은 머신 계정 암호가 구성된 값보다 오래되었는지 매일 확인하고 필요한 경우 갱신합니다.

기본 갱신 간격은 **30일**입니다. 기본값을 변경하려면 다음 절차의 단계를 따르십시오.

### 절차

1. `/etc/sss/sss.conf` 파일의 **AD** 공급자에 다음 매개 변수를 추가합니다.

```
ad_maximum_machine_account_password_age = value_in_days
```

2. **SSSD**를 다시 시작합니다.

```
# systemctl restart sssd
```

3. 자동 **Kerberos** 호스트 키 탭 업데이트를 비활성화하려면 `ad_maximum_machine_account_password_age = 0` 을 설정합니다.

### 추가 리소스

- `adcli(8)`

- **sssd.conf(5)**
- **SSSD 서비스가 'MEMORY:/etc/krb5.keytab]: Preauthentication failed를 사용하여 인증 정보를 초기화하지 못한 오류와 함께 실패합니다.**

#### 4.2. AD 도메인에서 RHEL 시스템 제거

AD 도메인에서 직접 AD(Active Directory)에 통합된 RHEL(Red Hat Enterprise Linux) 시스템을 제거하려면 다음 절차를 따르십시오.

##### 사전 요구 사항

- **SSSD(System Security Services Daemon) 또는 Samba Winbind를 사용하여 RHEL 시스템을 AD에 연결했습니다.**

##### 절차

1. **realm leave** 명령을 사용하여 ID 도메인에서 시스템을 제거합니다. 이 명령은 SSSD 및 로컬 시스템에서 도메인 구성을 제거합니다.

```
# realm leave ad.example.com
```



##### 참고

클라이언트가 도메인을 벗어나면 AD는 계정을 삭제하지 않고 로컬 클라이언트 구성만 제거합니다. AD 계정을 삭제하려면 --remove 옵션을 사용하여 명령을 실행합니다. 처음에 인증 정보 없이 연결을 시도하지만 유효한 Kerberos 티켓이 없는 경우 사용자 암호를 입력하라는 메시지가 표시됩니다. Active Directory에서 계정을 삭제할 수 있는 권한이 있어야 합니다.

2. **realm leave** 명령과 함께 -U 옵션을 사용하여 ID 도메인에서 시스템을 제거하려면 다른 사용자를 지정합니다.

기본적으로 realm leave 명령은 기본 관리자로 실행됩니다. AD의 경우 관리자 계정을 Administrator 라고 합니다. 다른 사용자가 도메인에 가입하는 데 사용된 경우 해당 사용자로 제거를 수행해야 할 수 있습니다.

```
# realm leave [ad.example.com] -U [AD.EXAMPLE.COM\user]'
```

명령은 먼저 자격 증명 없이 연결을 시도하지만 필요한 경우 암호를 입력하라는 메시지가 표시됩니다.

#### 검증 단계

- 도메인이 더 이상 구성되지 않았는지 확인합니다.

```
# realm discover [ad.example.com]
ad.example.com
  type: kerberos
  realm-name: EXAMPLE.COM
  domain-name: example.com
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: oddjob
  required-package: oddjob-mkhomedir
  required-package: sssd
  required-package: adcli
  required-package: samba-common-tools
```

#### 추가 리소스

- **realm(8)** 매뉴얼 페이지를 참조하십시오.

### 4.3. SSSD에서 짧은 AD 사용자 이름을 확인하도록 도메인 확인 순서 설정

SSSD 서비스를 사용하여 AD에 연결된 RHEL 호스트의 AD(Active Directory) 사용자 및 그룹을 해결하려면 기본적으로 정규화된 사용자 이름(예: `ad_username@ad.example.com` 및 `group@ad.example.com`)을 지정해야 합니다.

이 절차에서는 SSSD 구성에서 도메인 확인 순서를 설정하여 `ad_username` 과 같은 짧은 이름을 사용하여 AD 사용자 및 그룹을 확인할 수 있도록 합니다. 이 예제 구성은 다음과 같은 순서로 사용자 및 그룹을 검색합니다.

1. **Active Directory (AD) 하위 도메인 `subdomain2.ad.example.com`**
2. **AD 하위 도메인 `subdomain1.ad.example.com`**
3. **AD root 도메인 `ad.example.com`**

## 사전 요구 사항

- **SSSD 서비스를 사용하여 RHEL 호스트를 AD에 직접 연결합니다.**

## 절차

1. 텍스트 편집기에서 `/etc/sss/sss.conf` 파일을 엽니다.
2. 파일의 `[sss]` 섹션에서 `domain_resolution_order` 옵션을 설정합니다.

```
domain_resolution_order = subdomain2.ad.example.com,
subdomain1.ad.example.com, ad.example.com
```

3. 파일을 저장한 후 닫습니다.
4. **SSSD 서비스를 다시 시작하여 새 구성 설정을 로드합니다.**

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

## 검증 단계

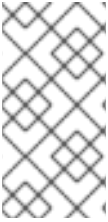
- 짧은 이름만 사용하여 첫 번째 도메인에서 사용자에게 대한 사용자 정보를 검색할 수 있는지 확인합니다.

```
[root@ad-client ~]# id <user_from_subdomain2>
uid=1916901142(user_from_subdomain2) gid=1916900513(domain users)
groups=1916900513(domain users)
```

## 4.4. 도메인 사용자의 로그인 권한 관리

기본적으로 도메인 측 액세스 제어가 적용되므로 **AD(Active Directory)** 사용자에게 대한 로그인 정책이 **AD** 도메인 자체에 정의되어 있습니다. 클라이언트 측 액세스 제어를 사용하도록 이 기본 동작을 재정의할 수 있습니다. 클라이언트 측 액세스 제어를 사용하면 로컬 정책에 의해서만 로그인 권한을 정의합니다.

도메인이 클라이언트 측 액세스 제어를 적용하는 경우 **realmd** 를 사용하여 해당 도메인에서 사용자에게 대한 기본 허용 또는 액세스 규칙을 구성할 수 있습니다.

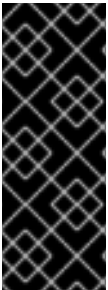


## 참고

액세스 규칙은 시스템의 모든 서비스에 대한 액세스를 허용하거나 거부합니다. 특정 시스템 리소스 또는 도메인에 더 구체적인 액세스 규칙을 설정해야 합니다.

### 4.4.1. 도메인 내 사용자에게 대한 액세스 활성화

기본적으로 **AD(Active Directory)** 사용자에게 대한 로그인 정책은 **AD** 도메인 자체에 정의되어 있습니다. 이 기본 동작을 재정의하고 **AD** 도메인 내의 사용자에게 대한 액세스를 활성화하도록 **RHEL** 호스트를 구성하려면 다음 절차를 따르십시오.



## 중요

**realm permit -x.x**를 사용하는 특정 사용자에게만 거부하면서 기본적으로 모두에 대한 액세스를 허용하지 않는 것이 좋습니다. 대신 모든 사용자에게 기본 액세스 정책 없음 정책을 유지하고 영역 허용 권한을 사용하여 선택한 사용자에게만 액세스 권한을 부여할 것을 권장합니다.

### 사전 요구 사항

- **RHEL** 시스템은 **Active Directory** 도메인의 멤버입니다.

### 절차

1. 모든 사용자에게 액세스 권한을 부여합니다.

```
# realm permit --all
```

2. 특정 사용자에게 액세스 권한을 부여합니다.

```
$ realm permit aduser01@example.com
$ realm permit 'AD.EXAMPLE.COM\aduser01'
```

현재는 신뢰할 수 있는 도메인의 사용자가 아닌 기본 도메인의 사용자에게 대한 액세스만 허용할 수 있습니다. 사용자 로그인에 도메인 이름이 포함되어 있어야 하고 **SSSD**가 현재 사용 가능한 하위 도메인에 대한 정보를 제공할 수 없기 때문입니다.

### 검증 단계

1.

SSH를 사용하여 서버에 `aduser01@example.com` 사용자로 로그인합니다.

```
$ ssh aduser01@example.com@server_name
[aduser01@example.com@server_name ~]$
```

2.

ssh 명령을 두 번째로 사용하여 이번에는 `aduser02@example.com` 사용자와 동일한 서버에 액세스합니다.

```
$ ssh aduser02@example.com@server_name
Authentication failed.
```

`aduser02@example.com` 사용자가 시스템에 대한 액세스를 거부한 방법을 확인합니다. 시스템에 로그인할 수 있는 권한이 부여된 경우 `aduser01@example.com` 사용자만 시스템에 로그인할 수 있습니다. 지정된 로그인 정책으로 인해 Active Directory 도메인의 다른 모든 사용자가 거부됩니다.



참고

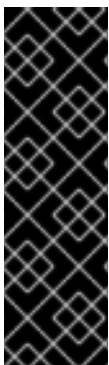
`sssd.conf` 파일에서 `use_fully_qualified_names` 를 `true`로 설정하면 모든 요청이 정규화된 도메인 이름을 사용해야 합니다. 그러나 `use_fully_qualified_names` 를 `false`로 설정하면 요청에 정규화된 이름을 사용할 수 있지만 간소화된 버전만 출력에 표시됩니다.

추가 리소스

- `realm(8)` 매뉴얼 페이지를 참조하십시오.

4.4.2. 도메인 내 사용자에게 대한 액세스 거부

기본적으로 AD(Active Directory) 사용자에게 대한 로그인 정책은 AD 도메인 자체에 정의되어 있습니다. 이 기본 동작을 재정의하고 AD 도메인 내의 사용자에게 대한 액세스를 거부하도록 RHEL 호스트를 구성하려면 다음 절차를 따르십시오.



중요

특정 사용자 또는 그룹에 대한 액세스만 허용하는 것보다 일부에 대한 액세스를 거부하는 것 외에도 다른 사용자 또는 그룹에 대한 액세스만 허용하는 것이 더 안전합니다. 따라서 `realm permit -x` 가 있는 특정 사용자에게만 거부하고 기본적으로 모든 액세스 권한을 허용하지 않는 것이 좋습니다. 대신 모든 사용자에게 기본 액세스 정책 없음 정책을 유지하고 영역 허용 권한을 사용하여 선택한 사용자에게만 액세스 권한을 부여할 것을 권장합니다.



## 사전 요구 사항

- **RHEL 시스템은 Active Directory 도메인의 멤버입니다.**

## 절차

1. 도메인 내 모든 사용자에게 대한 액세스를 거부합니다.

```
# realm deny --all
```

이 명령은 영역 계정이 로컬 시스템에 로그인하지 못하도록 합니다. **realm permit** 을 사용하여 특정 계정에 대한 로그인을 제한합니다.

2. 도메인 사용자의 **login-policy** 가 **deny-any-login** 으로 설정되어 있는지 확인합니다.

```
[root@replica1 ~]# realm list
example.net
type: kerberos
realm-name: EXAMPLE.NET
domain-name: example.net
configured: kerberos-member
server-software: active-directory
client-software: sssd
required-package: oddjob
required-package: oddjob-mkhomedir
required-package: sssd
required-package: adcli
required-package: samba-common-tools
login-formats: %U@example.net
login-policy: deny-any-login
```

3. **-x** 옵션을 사용하여 특정 사용자에게 대한 액세스를 거부합니다.

```
$ realm permit -x 'AD.EXAMPLE.COM\aduser02'
```

## 검증 단계

- **SSH를 사용하여 서버에 aduser01@example.net 사용자로 로그인합니다.**

```
$ ssh aduser01@example.net@server_name
Authentication failed.
```



## 참고

**sssd.conf** 파일에서 **use\_fully\_qualified\_names** 를 **true**로 설정하면 모든 요청이 정규화된 도메인 이름을 사용해야 합니다. 그러나 **use\_fully\_qualified\_names** 를 **false**로 설정하면 요청에 정규화된 이름을 사용할 수 있지만 간소화된 버전만 출력에 표시됩니다.

## 추가 리소스

- **realm(8)** 매뉴얼 페이지를 참조하십시오.

## 4.5. RHEL에서 그룹 정책 오브젝트 액세스 제어 적용

그룹 정책 개체 (GPO)는 AD(Microsoft Active Directory)에 저장된 액세스 제어 설정 컬렉션입니다. **administrators**는 AD에 **EgressIPs**를 지정하면 관리자가 AD에 가입한 Windows 클라이언트와 RHEL(Red Hat Enterprise Linux) 호스트 둘 다에서 준수하는 로그인 정책을 정의할 수 있습니다.

다음 섹션에서는 사용자 환경에서 GPO를 관리하는 방법에 대해 설명합니다.

- [SSSD에서 EgressIP 액세스 제어 규칙을 해석하는 방법](#)
- [SSSD에서 지원하는 EgressIP 설정 목록](#)
- [nmap 적용 제어를 위한 SSSD 옵션 목록](#)
- [EgressIP 액세스 제어 모드 변경](#)
- [RHEL 호스트에 대한 EgressIP 생성 및 구성](#)

### 4.5.1. SSSD에서 EgressIP 액세스 제어 규칙을 해석하는 방법

기본적으로 SSSD는 AD(Active Directory) 도메인 컨트롤러에서 GPG(그룹 정책 개체)를 검색하고 이를 평가하여 사용자가 AD에 결합된 특정 RHEL 호스트에 로그인할 수 있는지 확인합니다.

SSSD는 AD Windows Logon Rights 를 PAM(Pluggable Authentication Module) 서비스 이름에 매

평하여 GNU/Linux 환경에서 이러한 권한을 적용합니다.

**AD Administrator**는 **EgressIP** 규칙의 범위를 보안 필터에 나열하여 특정 사용자, 그룹 또는 호스트로 제한할 수 있습니다.

#### 호스트 필터링에 대한 제한

이전 버전의 **SSSD**는 **AD EgressIP** 보안 필터의 호스트를 평가하지 않습니다.

- **RHEL 8.3.0 이상: SSSD**는 보안 필터의 사용자, 그룹 및 호스트를 지원합니다.
- **8.3.0 이전의 RHEL 버전: SSSD**는 호스트 항목을 무시하고 보안 필터의 사용자 및 그룹만 지원합니다.  
**SSSD가 fsGroup** 기반 액세스 제어를 특정 호스트에 적용하도록 하려면 **AD** 도메인에서 새 **OU(조직 단위)**를 만들고 시스템을 새 **OU**로 이동한 다음 **EgressIP**를 이 **OU**에 연결합니다.

#### 그룹 필터링에 대한 제한 사항

**SSSD**는 현재 **SID(Security Identifier) S-1-5-32-544** 와 같은 **Active Directory**의 기본 제공 그룹을 지원하지 않습니다. **Red Hat**은 **RHEL** 호스트를 대상으로 하는 **AD EgressIPs**에서 **AD** 기본 제공 그룹을 사용하지 않는 것이 좋습니다.

#### 추가 리소스

- **Windows EgressIP** 옵션 목록과 해당 **SSSD** 옵션 목록은 **SSSD**에서 지원하는 **GPO** 설정 목록을 참조하십시오.

#### 4.5.2. SSSD에서 지원하는 EgressIP 설정 목록

다음 표에서는 **Windows**의 그룹 정책 관리 편집기에 지정된 대로 **Active Directory options**에 해당하는 **SSSD** 옵션을 보여줍니다.

표 4.1. SSSD에서 검색한 EgressIP 액세스 제어 옵션

EgressIP 옵션	해당 sssd.conf 옵션
로컬에 대한 로그 허용 로컬 거부 로그	ad_gpo_map_interactive

EgressIP 옵션	해당 sssd.conf 옵션
원격 데스크톱 서비스를 통해 로그인 허용 원격 데스크톱 서비스를 통한 로그인 거부	<b>ad_gpo_map_remote_interactive</b>
네트워크에서 이 컴퓨터에 액세스 네트워크에서 이 컴퓨터에 대한 액세스 거부	<b>ad_gpo_map_network</b>
배치 작업으로 로그인 허용 배치 작업으로 로그인 거부	<b>ad_gpo_map_batch</b>
서비스로 로그인 허용 서비스 거부	<b>ad_gpo_map_service</b>

추가 리소스

- GPO 옵션에 매핑되는 PAM(Pluggable Authentication Module) 서비스와 같은 sssd.conf 설정에 대한 자세한 내용은 sssd-ad(5) 매뉴얼 페이지 항목을 참조하십시오.**

4.5.3. nmap 적용 제어를 위한 SSSD 옵션 목록

다음 SSSD 옵션을 설정하여 EgressIP 규칙의 범위를 제한할 수 있습니다.

**ad\_gpo\_access\_control** 옵션

/etc/sss/sss.conf 파일에서 **ad\_gpo\_access\_control** 옵션을 설정하여 **usually** 기반 액세스 제어가 작동하는 세 가지 모드 중에서 선택할 수 있습니다.

표 4.2. ad\_gpo\_access\_control 값 테이블

값 of ad_gpo_access_control	동작
<b>enforcing</b>	CloudEvent 기반 액세스 제어 규칙이 평가되고 적용됩니다. 이는 RHEL 8의 기본 설정입니다.
허용	EgressIP 기반 액세스 제어 규칙은 평가되지만 적용되지 <b>않습니다</b> . 액세스 거부될 때마다 <b>syslog</b> 메시지가 기록됩니다. 이는 RHEL 7의 기본 설정입니다. 이 모드는 사용자가 계속 로그인할 수 있도록 허용하면서 정책 조정을 테스트하는 데 적합합니다.
<b>disabled</b>	EgressIP 기반 액세스 제어 규칙은 평가되거나 적용되지 <b>않습니다</b> .

**ad\_gpo\_implicit\_deny** 옵션

`ad_gpo_implicit_deny` 옵션은 기본적으로 **False** 로 설정됩니다. 이 기본 상태에서는 해당하는 **EgressIPs**를 찾을 수 없는 경우 사용자가 액세스가 허용됩니다. 이 옵션을 **True** 로 설정하면 **EgressIP** 규칙이 있는 사용자의 액세스를 명시적으로 허용해야 합니다.

이 기능을 사용하여 보안을 강화할 수 있지만 의도하지 않게 액세스를 거부하지 않도록 주의하십시오. `ad_gpo_access_control` 이 허용 으로 설정되어 있는 동안 이 기능을 테스트하는 것이 좋습니다.

다음 두 테이블은 사용자가 **AD** 서버 측에 정의된 허용 및 거부 권한 및 `ad_gpo_implicit_deny` 의 값을 기반으로 사용자가 허용되거나 거부되는 경우를 보여줍니다.

표 4.3. `ad_gpo_implicit_deny` 를 **False** (기본값)로 설정한 로그인 동작

allow-rules	deny-rules	결과
누락됨	누락됨	모든 사용자가 허용됩니다.
누락됨	Exists	거부 규칙에 없는 사용자만 허용됩니다.
Exists	누락됨	allow-rules의 사용자만 허용
Exists	Exists	allow-rules에 있는 사용자와 deny-rules에 포함되지 않은 사용자만 허용됩니다.

표 4.4. `ad_gpo_implicit_deny` 를 **True**로 설정하여 로그인 동작

allow-rules	deny-rules	결과
누락됨	누락됨	사용자가 허용되지 않음
누락됨	Exists	사용자가 허용되지 않음
Exists	누락됨	allow-rules의 사용자만 허용
Exists	Exists	allow-rules에 있는 사용자와 deny-rules에 포함되지 않은 사용자만 허용됩니다.

#### 추가 리소스

- SSSD에서 8.4 적용 모드를 변경하는 절차는 [Changing the GPO 액세스 제어 모드 변경을 참조하십시오.](#)**
- 서로 다른 **task**의 각 **options** 모드에 대한 자세한 내용은 **sssd-ad(5)** 매뉴얼 페이지의

**ad\_gpo\_access\_control** 항목을 참조하십시오.

#### 4.5.4. EgressIP 액세스 제어 모드 변경

이 절차에서는 **EgressIP** 기반 액세스 제어 규칙을 평가하여 **RHEL** 호스트에 **AD(Active Directory)** 환경에 적용하는 방법을 변경합니다.

이 예제에서는 **tests**의 작업 모드를 강제 (기본값)에서 테스트 목적으로 허용으로 변경합니다. **In this example, you will change the EgressIP operation mode from enforcing (the default) to permissive for testing purposes.**

##### 중요

다음과 같은 오류가 표시되면 **Active Directory** 사용자가 **GPO** 기반 액세스 제어로 인해 로그인할 수 없습니다.

- 

**/var/log/secure** 에서:

```
Oct 31 03:00:13 client1 sshd[124914]: pam_sss(sshd:account): Access
denied for user aduser1: 6 (Permission denied)
Oct 31 03:00:13 client1 sshd[124914]: Failed password for aduser1 from
127.0.0.1 port 60509 ssh2
Oct 31 03:00:13 client1 sshd[124914]: fatal: Access denied for user
aduser1 by PAM account configuration [preauth]
```

- 

**/var/log/sss/sss\_\_example.com\_.log** 에서 다음을 수행합니다.

```
(Sat Oct 31 03:00:13 2020) [sss[be[example.com]]]
[ad_gpo_perform_hbac_processing] (0x0040): GPO access check failed:
[1432158236](Host Access Denied)
(Sat Oct 31 03:00:13 2020) [sss[be[example.com]]] [ad_gpo_cse_done]
(0x0040): HBAC processing failed: [1432158236](Host Access Denied)
(Sat Oct 31 03:00:13 2020) [sss[be[example.com]]]
[ad_gpo_access_done] (0x0040): GPO-based access control failed.
```

바람직하지 않은 동작인 경우 **AD**의 적절한 **EgressIP** 설정 문제를 해결하는 동안 이 절차에 설명된 대로 **ad\_gpo\_access\_control** 을 일시적으로 허용으로 설정할 수 있습니다.

사전 요구 사항

- **SSSD를 사용하여 AD 환경에 RHEL 호스트에 참여했습니다.**
- **/etc/sss/sssd.conf 설정 파일을 편집하려면 root 권한이 필요합니다.**

#### 절차

1. **SSSD 서비스를 중지합니다.**

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

2. **텍스트 편집기에서 /etc/sss/sssd.conf 파일을 엽니다.**

3. **AD 도메인의 domain 섹션에서 ad\_gpo\_access\_control 을 허용하도록 설정합니다.**

```
[domain/example.com]
ad_gpo_access_control=permissive
...
```

4. **/etc/sss/sssd.conf 파일을 저장합니다.**

5. **SSSD 서비스를 다시 시작하여 구성 변경 사항을 로드합니다.**

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

#### 추가 리소스

- **다른 EgressIP 액세스 제어 모드의 목록은 [List of SSSD options to controlResourceOverride enforcement](#) 를 참조하십시오.**

#### 4.5.5. AD GUI에서 RHEL 호스트에 대한 EgressIP 생성 및 구성

그룹 정책 개체(GPO)는 AD(Microsoft Active Directory)에 저장된 액세스 제어 설정 컬렉션입니다. 다음 절차에서는 AD GUI(그래픽 사용자 인터페이스)에 EgressIP를 생성하여 AD 도메인에 직접 통합된 RHEL 호스트에 대한 로그온 액세스를 제어합니다.

#### 사전 요구 사항

- **SSSD를 사용하여 AD 환경에 RHEL 호스트에 참여했습니다.**
- **GUI를 사용하여 AD를 변경할 수 있는 AD 관리자 권한이 있어야 합니다.**

### 절차

1. **Active Directory 사용자 및 컴퓨터 내에서 OU(Organizational Unit)를 만들어 새 EgressIP 과 연결합니다.**
  - a. **도메인을 마우스 오른쪽 버튼으로 클릭합니다.**
  - b. **새로 생성을 선택합니다.**
  - c. **조직 단위 를 선택합니다.**
2. **RHEL 호스트(Active Directory에 참여할 때 생성된)를 나타내는 Computer Object의 이름을 클릭하고 새 OU로 끌어 옵니다. 자체 OU에 RHEL 호스트를 보유함으로써 EgressIP는 이 호스트를 대상으로 합니다.**
3. **그룹 정책 관리 편집기 에서 만든 OU에 대한 새 GPO를 만듭니다.**
  - a. **Forest 펼치기입니다.**
  - b. **도메인 을 확장합니다.**
  - c. **도메인을 확장합니다.**
  - d. **새 OU를 마우스 오른쪽 버튼으로 클릭합니다.**
  - e. **이 도메인에서 Create a EgressIP in this domain 을 선택합니다.**



4. **Allow SSH 액세스 또는 Allow Console/GUI 액세스 권한과 같은 새 EgressIP의 이름을 지정하고 확인을 클릭합니다.**
  
5. **새 EgressIP을 편집합니다.**
  - a. **그룹 정책 관리 편집기에서 OU를 선택합니다.**
  
  - b. **마우스 오른쪽 버튼으로 클릭하고 편집 을 선택합니다.**
  
  - c. **사용자 권한 할당 을 선택합니다.**
  
  - d. **컴퓨터 구성선택**
  
  - e. **정책을 선택합니다.**
  
  - f. **Windows 설정을 선택합니다.**
  
  - g. **보안 설정을 선택합니다.**
  
  - h. **로컬 정책을 선택합니다.**
  
  - i. **사용자 권한 할당 을 선택합니다.**
  
6. **로그인 권한을 할당합니다.**
  - a. **로컬 콘솔/GUI에 대한 액세스 권한을 부여하려면 허용 로그 를 두 번 클릭합니다.**
  
  - b. **원격 데스크탑 서비스를 통해 로그를 두 번 클릭하여 SSH 액세스 권한을 부여합니다.**

7. 이러한 정책 중 하나에 액세스하려는 사용자를 정책 자체에 추가합니다.
  - a. 사용자 또는 그룹 추가를 클릭합니다.
  - b. 빈 필드 내에 사용자 이름을 입력합니다.
  - c. **OK**를 클릭합니다.

#### 추가 리소스

- 그룹 정책 개체에 대한 자세한 내용은 **Microsoft** 문서의 [그룹 정책 개체](#) 를 참조하십시오.

#### 4.5.6. 추가 리소스

- **RHEL** 호스트를 **Active Directory** 환경에 연결하는 방법에 대한 자세한 내용은 **SSSD**를 사용하여 **RHEL** 시스템을 **AD**에 직접 연결을 참조하십시오.

## 5장. 관리 서비스 계정을 사용하여 AD에 액세스

**Active Directory (AD) 관리 서비스 계정 (MSAs)을 사용하면 특정 컴퓨터에 해당하는 AD에서 계정을 만들 수 있습니다. RHEL 호스트를 AD 도메인에 가입하지 않고 MSA를 사용하여 특정 사용자 주체로 AD 리소스에 연결할 수 있습니다.**

이 섹션에서는 다음 항목에 대해 설명합니다.

- [관리형 서비스 계정의 이점](#)
- [RHEL 호스트에 대한 관리형 서비스 계정 구성](#)
- [관리형 서비스 계정의 암호 업데이트](#)
- [관리형 서비스 계정 사양](#)
- [adcli create-msa 명령의 옵션](#)

### 5.1. 관리형 서비스 계정의 이점

**RHEL 호스트가 가입하지 않고 AD(Active Directory) 도메인에 액세스하도록 허용하려면 MSA(관리 서비스 계정)를 사용하여 해당 도메인에 액세스할 수 있습니다. MSA는 특정 컴퓨터에 해당하는 AD 계정이며 특정 사용자 보안 주체로 AD 리소스에 연결하는 데 사용할 수 있습니다. An MSA is an account in AD that corresponds to a specific computer, which you can use to connect to AD resources as a specific user principal.**

예를 들어 AD 도메인 `production.example.com` 에 `lab.example.com` AD 도메인과의 단방향 신뢰 관계가 있는 경우 다음 조건이 적용됩니다.

- 랩 도메인은 `production` 도메인에서 사용자 및 호스트를 신뢰합니다.
- `production` 도메인은 랩 도메인에서 사용자 및 호스트를 신뢰하지 않습니다.

즉, **client.lab.example.com** 과 같은 랩 도메인에 연결된 호스트는 신뢰를 통해 **production** 도메인의 리소스에 액세스할 수 없습니다.

**client.lab.example.com** 호스트에 대한 예외를 생성하려면 **adcli** 유틸리티를 사용하여 **production.example.com** 도메인에서 클라이언트 호스트에 대한 **MSA**를 생성할 수 있습니다. **MSA**의 **Kerberos** 주체로 인증하면 클라이언트 호스트에서 프로덕션 도메인에서 보안 **LDAP** 검색을 수행할 수 있습니다.

## 5.2. RHEL 호스트에 대한 관리형 서비스 계정 구성

이 절차에서는 **lab.example.com AD(Active Directory)** 도메인에서 호스트에 대한 **MSA(관리 서비스 계정)**를 생성하고 **production.example.com AD** 도메인에 액세스하고 인증할 수 있도록 **SSSD**를 구성합니다.



### 참고

**RHEL 호스트의 AD 리소스에 액세스해야 하는 경우, realm 명령을 사용하여 RHEL 호스트를 AD 도메인에 참여하는 것이 좋습니다. SSSD를 사용하여 RHEL 시스템을 AD에 직접 연결하십시오.**

다음 조건 중 하나가 적용되는 경우에만 다음 절차를 수행하십시오.

- RHEL 호스트를 AD 도메인에 연결할 수 없으며, AD에서 해당 호스트에 대한 계정을 만들려고 합니다.**
- RHEL 호스트에 AD 도메인에 참여했으며 가입한 도메인의 호스트 인증 정보가 유효하지 않은 다른 AD 도메인에 액세스해야 합니다(예: 일방향 신뢰).**

### 사전 요구 사항

- RHEL 호스트의 다음 포트가 열려 있고 AD 도메인 컨트롤러에서 액세스할 수 있는지 확인합니다.**

서비스	포트	프로토콜
DNS	53	TCP, UDP
LDAP	389	TCP, UDP

서비스	포트	프로토콜
LDAPS(선택 사항)	636	TCP, UDP
Kerberos	88	TCP, UDP

- **production.example.com** 도메인에 MSA를 생성할 수 있는 권한이 있는 **AD Administrator**의 암호가 있습니다.
- **adcli** 명령을 실행하고 **/etc/sss/sss.conf** 구성 파일을 수정하는 데 필요한 루트 권한이 있습니다.
- (선택 사항) **klist** 진단 유틸리티를 포함하는 **krb5-workstation** 패키지가 설치되어 있습니다.

#### 절차

1.

**production.example.com** AD 도메인에 호스트에 대한 MSA를 생성합니다.

```
[root@client ~]# adcli create-msa --domain=production.example.com
```

2.

생성된 Kerberos 키 탭에서 MSA에 대한 정보를 표시합니다. MSA 이름을 기록해 둡니다.

```
[root@client ~]# klist -k /etc/krb5.keytab.production.example.com
Keytab name: FILE:/etc/krb5.keytab.production.example.com
KVNO Principal
```

```
-----
 2 CLIENT!S3A$@PRODUCTION.EXAMPLE.COM (aes256-cts-hmac-sha1-96)
 2 CLIENT!S3A$@PRODUCTION.EXAMPLE.COM (aes128-cts-hmac-sha1-96)
```

3.

**/etc/sss/sss.conf** 파일을 열고 다음을 추가할 적절한 SSSD 도메인 구성을 선택합니다.

- MSA가 다른 포리스트의 AD 도메인에 해당하는 경우 **[domain /<name\_of\_domain>]**이라는 새 도메인 섹션을 만들고 MSA 및 keytab에 대한 정보를 입력합니다. 가장 중요한 옵션은 **ldap\_sasl\_authid, ldap\_krb5\_keytab, krb5\_keytab**입니다.

```
[domain/production.example.com]
ldap_sasl_authid = CLIENT!S3A$@PRODUCTION.EXAMPLE.COM
ldap_krb5_keytab = /etc/krb5.keytab.production.example.com
krb5_keytab = /etc/krb5.keytab.production.example.com
```

```
ad_domain = production.example.com
krb5_realm = PRODUCTION.EXAMPLE.COM
access_provider = ad
...
```

- **MSA가 지역 산포의 AD 도메인에 해당하는 경우 [domain/root.example.com/sub-domain.example.com] 형식으로 새 하위 도메인 섹션을 만들고 MSA 및 키 탭에 대한 정보를 입력합니다. 가장 중요한 옵션은 ldap\_sasl\_authid, ldap\_krb5\_keytab, krb5\_keytab 입니다.**

```
[domain/ad.example.com/production.example.com]
ldap_sasl_authid = CLIENT!S3A$@PRODUCTION.EXAMPLE.COM
ldap_krb5_keytab = /etc/krb5.keytab.production.example.com
krb5_keytab = /etc/krb5.keytab.production.example.com
ad_domain = production.example.com
krb5_realm = PRODUCTION.EXAMPLE.COM
access_provider = ad
...
```

검증 단계

- **Kerberos 티켓 확장 티켓(TGT)을 MSA로 검색할 수 있는지 확인합니다.**

```
[root@client ~]# kinit -k -t /etc/krb5.keytab.production.example.com 'CLIENT!S3A$'
[root@client ~]# klist
Ticket cache: KCM:0:54655
Default principal: CLIENT!S3A$@PRODUCTION.EXAMPLE.COM

Valid starting    Expires          Service principal
11/22/2021 15:48:03 11/23/2021 15:48:03
krbtgt/PRODUCTION.EXAMPLE.COM@PRODUCTION.EXAMPLE.COM
```

- **AD에서 관리 서비스 계정 구성 단위(OU)에 호스트에 대한 MSA가 있는지 확인합니다.**

추가 리소스

- **[SSSD를 사용하여 RHEL 시스템을 AD에 직접 연결](#)**

5.3. 관리형 서비스 계정의 암호 업데이트

**MSA(관리 서비스 계정)에는 AD(Active Directory)에서 자동으로 유지 관리하는 복잡한 암호가 있습니다. 기본적으로 SSSD(System Services Security Daemon)는 30일이 지난 경우 Kerberos 키탭에서 MSA 암호를 자동으로 업데이트하여 AD의 암호를 최신 상태로 유지합니다. 이 절차에서는 MSA의 암호를 수동으로 업데이트하는 방법을 설명합니다.**

## 사전 요구 사항

- 이전에 **production.example.com AD** 도메인에 호스트에 대한 **MSA**를 생성했습니다.
- (선택 사항) **klist** 진단 유틸리티를 포함하는 **krb5-workstation** 패키지가 설치되어 있습니다.

## 절차

1. (선택 사항) Kerberos 키탭에 **MSA**의 현재 키 버전 번호(**KVNO**)를 표시합니다. 현재 **KVNO**는 2입니다.

```
[root@client ~]# klist -k /etc/krb5.keytab.production.example.com
Keytab name: FILE:/etc/krb5.keytab.production.example.com
KVNO Principal
-----
  2 CLIENT!S3A$@PRODUCTION.EXAMPLE.COM (aes256-cts-hmac-sha1-96)
  2 CLIENT!S3A$@PRODUCTION.EXAMPLE.COM (aes128-cts-hmac-sha1-96)
```

2. **production.example.com AD** 도메인에서 **MSA**의 암호를 업데이트합니다.

```
[root@client ~]# adcli update --domain=production.example.com --host-
keytab=/etc/krb5.keytab.production.example.com --computer-password-lifetime=0
```

## 검증 단계

- Kerberos 키 탭에서 **KVNO**가 증가했는지 확인합니다.

```
[root@client ~]# klist -k /etc/krb5.keytab.production.example.com
Keytab name: FILE:/etc/krb5.keytab.production.example.com
KVNO Principal
-----
  3 CLIENT!S3A$@PRODUCTION.EXAMPLE.COM (aes256-cts-hmac-sha1-96)
  3 CLIENT!S3A$@PRODUCTION.EXAMPLE.COM (aes128-cts-hmac-sha1-96)
```

## 5.4. 관리형 서비스 계정 사양

**adcli** 유틸리티에서 생성하는 **MSA**(관리형 서비스 계정)에는 다음과 같은 사양이 있습니다.

- 추가 서비스 주체 이름(**SPN**)을 사용할 수 없습니다.

- 기본적으로 MSA의 Kerberos 주체는 `/etc/krb5.keytab.example.com`과 같이 `<default_keytab_location>.<Active_Directory_domain >`이라는 Kerberos 키 탭에 저장됩니다.
- 이름은 20자로 제한됩니다. **Names are limited to 20 characters.** 마지막 4자의 접미사는 숫자와 대문자 및 소문자 ASCII 범위로, 입력한 짧은 호스트 이름에 추가되는 ASCII 범위이며 구분 기호로 ! 문자를 사용합니다. 예를 들어, 짧은 이름이 `myhost` 인 호스트는 다음 사양으로 MSA를 수신합니다.

사양	값
CN(Common Name) 특성	myhost!A2c
NetBIOS 이름	myhost!A2c\$
sAMAccountName	myhost!A2c\$
production.example.com AD 도메인의 Kerberos 사용자	myhost!A2c\$@PRODUCTION.EXAMPLE.COM

### 5.5. ADCLI CREATE-MSA 명령의 옵션

`adcli` 유틸리티에 전달할 수 있는 글로벌 옵션 외에도 다음 옵션을 지정하여 MSA(관리 서비스 계정)를 구체적으로 제어할 수 있습니다.

#### -N, --computer-name

Active Directory (AD) 도메인에 생성되는 MSA의 짧은 이름. 이름을 지정하지 않으면 `--host-fqdn`의 첫 번째 부분이 임의의 접미사와 함께 사용됩니다.

#### -O, --domain-ou=OU=<path\_to\_OU>

MSA를 만드는 OU(조직 단위)의 전체 고유 이름입니다. 이 값을 지정하지 않으면 MSA가 기본 위치 `OU=CN=Managed Service Accounts,DC=EXAMPLE,DC=COM`에 생성됩니다.

#### -H, --host-fqdn=host

로컬 시스템의 정규화된 DNS 도메인 이름을 재정의합니다. 이 옵션을 지정하지 않으면 로컬 시스템의 호스트 이름이 사용됩니다.

#### -K, --host-keytab=<path\_to\_keytab>

MSA 자격 증명을 저장할 호스트 키탭의 경로입니다. 이 값을 지정하지 않으면 기본 위치 `/etc/krb5.keytab` 이 `/etc/krb5.keytab.com`과 같이 접미사로 추가된 소문자 Active Directory 도메인



이름과 함께 사용됩니다.

**--use-ldaps**

**LDAP(Secure LDAP) 채널을 통해 MSA를 생성합니다.**

**--verbose**

**MSA를 만드는 동안 자세한 정보를 인쇄합니다.**

**--show-details**

**MSA를 생성한 후 MSA에 대한 정보를 출력합니다.**

**--show-password**

**MSA를 만든 후 MSA 암호를 인쇄합니다.**