Red Hat Enterprise Linux 10

Integrating RHEL systems directly with Windows Active Directory

Joining RHEL hosts to AD and accessing resources in AD

Last Updated: 2025-06-11

Red Hat Enterprise Linux 10 Integrating RHEL systems directly with Windows Active Directory

Joining RHEL hosts to AD and accessing resources in AD

Legal Notice

Abstract

You can join Red Hat Enterprise Linux (RHEL) hosts to an Active Directory (AD) domain by using the System Security Services Daemon (SSSD) or the Samba Winbind service to access AD resources. Alternatively, it is also possible to access AD resources without domain integration by using a Managed Service Account (MSA).

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Log in to the Jira website.

2. Click **Create** in the top navigation bar

3. Enter a descriptive title in the **Summary** field.

4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.

5. Click **Create** at the bottom of the dialogue.

CHAPTER 1. CONNECTING RHEL SYSTEMS DIRECTLY TO AD USING SAMBA WINBIND

To connect a RHEL system to Active Directory (AD), use:

- Samba Winbind to interact with the AD identity and authentication source
- **realmd** to detect available domains and configure the underlying RHEL system services.

1.1. OVERVIEW OF DIRECT INTEGRATION USING SAMBA WINBIND

Samba Winbind emulates a Windows client on a Linux system and communicates with AD servers.

You can use the **realmd** service to configure Samba Winbind by:

- Configuring network authentication and domain membership in a standard way.
- Automatically discovering information about accessible domains and realms.
- Not requiring advanced configuration to join a domain or realm.

Note that:

- Direct integration with Winbind in a multi-forest AD setup requires bidirectional trusts.
- Remote forests must trust the local forest to ensure that the **idmap_ad** plug-in handles remote forest users correctly.

Samba's **winbindd** service provides an interface for the Name Service Switch (NSS) and enables domain users to authenticate to AD when logging into the local system.

Using **winbindd** provides the benefit that you can enhance the configuration to share directories and printers without installing additional software.

Additional resources

- **realmd** man page on your system
- **winbindd** man page on your system

1.2. SUPPORTED WINDOWS PLATFORMS FOR DIRECT INTEGRATION

You can directly integrate your RHEL system with Active Directory forests that use the following forest and domain functional levels:

- Forest functional level range: Windows Server 2008 - Windows Server 2016
- Domain functional level range: Windows Server 2008 - Windows Server 2016

Direct integration has been tested on the following supported operating systems:

NOTE

Windows Server 2019 and Windows Server 2022 do not introduce a new functional level. The highest functional level Windows Server 2019 and Windows Server 2022 use is Windows Server 2016.

1.3. REALM COMMANDS

The **realmd** system has two major task areas:

- Managing system enrollment in a domain.
- Controlling which domain users are allowed to access local system resources.

In **realmd** use the command line tool **realm** to run commands. Most **realm** commands require the user to specify the action that the utility should perform, and the entity, such as a domain or user account, for which to perform the action.

Table 1.1. realmd commands

Command	Description
Realm Commands	
discover	Run a discovery scan for domains on the network.
join	Add the system to the specified domain.
leave	Remove the system from the specified domain.
list	List all configured domains for the system or all discovered and configured domains.
Login Commands	
permit	Enable access for specific users or for all users within a configured domain to access the local system.
deny	Restrict access for specific users or for all users within a configured domain to access the local system.

Additional resources

- **realm(8)** man page on your system

CHAPTER 2. MANAGING DIRECT CONNECTIONS TO AD

After you connect your Red Hat Enterprise Linux (RHEL) system to an Active Directory (AD) domain using System Security Services Daemon (SSSD) or Samba Winbind, you can manage key settings such as Kerberos renewals, domain membership, user access permissions, and Group Policy Objects (GPOs).

Prerequisites

- You have connected your RHEL system to the Active Directory domain, either with SSSD or Samba Winbind.

2.1. MODIFYING THE DEFAULT KERBEROS HOST KEYTAB RENEWAL INTERVAL

SSSD automatically renews the Kerberos host keytab file in an AD environment if the **adcli** package is installed. The daemon checks daily if the machine account password is older than the configured value and renews it if necessary.

The default renewal interval is 30 days. To change the default, follow the steps in this procedure.

Procedure

1. Add the following parameter to the AD provider in your **/etc/sssd/sssd.conf** file:

   ```
   ad_maximum_machine_account_password_age = value_in_days
   ```

2. Restart SSSD:

   ```
   # systemctl restart sssd
   ```

3. To disable the automatic Kerberos host keytab renewal, set **ad_maximum_machine_account_password_age = 0**.

Additional resources

- **adcli(8)**
- **sssd.conf(5)**
- SSSD service is failing with an error 'Failed to initialize credentials using keytab [MEMORY:/etc/krb5.keytab]: Preauthentication failed.' (Red Hat Knowledgebase)

2.2. REMOVING A RHEL SYSTEM FROM AN AD DOMAIN

Follow this procedure to remove a Red Hat Enterprise Linux (RHEL) system that is integrated into Active Directory (AD) directly from the AD domain.

Prerequisites

- You have used the System Security Services Daemon (SSSD) or Samba Winbind to connect your RHEL system to AD.

Procedure

1. Remove a system from an identity domain using the **realm leave** command. The command removes the domain configuration from SSSD and the local system.

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

   NOTE

   When a client leaves a domain, AD does not delete the account and only removes the local client configuration. To delete the AD account, run the command with the **--remove** option. Initially, an attempt is made to connect without credentials, but you are prompted for your user password if you do not have a valid Kerberos ticket. You must have rights to remove an account from Active Directory.

2. Use the **-U** option with the **realm leave** command to specify a different user to remove a system from an identity domain.
   By default, the **realm leave** command is executed as the default administrator. For AD, the administrator account is called **Administrator**. If a different user was used to join to the domain, it might be required to perform the removal as that user.

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

   The command first attempts to connect without credentials, but it prompts for a password if required.

Verification

- Verify the domain is no longer configured:

  ```
  # 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
  ```

Additional resources

- **realm(8)** man page on your system

2.3. SETTING THE DOMAIN RESOLUTION ORDER IN SSSD TO RESOLVE SHORT AD USER NAMES

By default, you must specify fully qualified usernames, like **ad_username@ad.example.com** and **group@ad.example.com**, to resolve Active Directory (AD) users and groups on a RHEL host connected to AD with the SSSD service.

This procedure sets the domain resolution order in the SSSD configuration so you can resolve AD users and groups using short names, like **ad_username**. This example configuration searches for users and groups in the following order:

1. Active Directory (AD) child domain **subdomain2.ad.example.com**
2. AD child domain **subdomain1.ad.example.com**
3. AD root domain **ad.example.com**

Prerequisites

- You have used the SSSD service to connect the RHEL host directly to AD.

Procedure

1. Open the **/etc/sssd/sssd.conf** file in a text editor.
2. Set the **domain_resolution_order** option in the `` section of the file.

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

3. Save and close the file.
4. Restart the SSSD service to load the new configuration settings.

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

Verification

- Verify you can retrieve user information for a user from the first domain using only a short name.

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

2.4. MANAGING LOGIN PERMISSIONS FOR DOMAIN USERS

By default, domain-side access control is applied, which means that login policies for Active Directory (AD) users are defined in the AD domain itself. This default behavior can be overridden so that client-side access control is used. With client-side access control, login permission is defined by local policies only.

If a domain applies client-side access control, you can use the **realmd** to configure basic allow or deny access rules for users from that domain.

NOTE

Access rules either allow or deny access to all services on the system. More specific access rules must be set on a specific system resource or in the domain.

2.4.1. Enabling access to users within a domain

By default, login policies for Active Directory (AD) users are defined in the AD domain itself. You can override this default behavior and configure a RHEL host to enable access for users within an AD domain.

IMPORTANT

It is not recommended to allow access to all by default while only denying it to specific users with realm permit **-x**. Instead, Red Hat recommends maintaining a default no access policy for all users and only grant access to selected users using realm permit.

Prerequisites

- Your RHEL system is a member of the Active Directory domain.

Procedure

1. Grant access to all users:

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

2. Grant access to specific users:

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

Currently, you can only allow access to users in primary domains and not to users in trusted domains. This is due to the fact that user login must contain the domain name and SSSD cannot currently provide **realmd** with information about available child domains.

Verification

1. Use SSH to log in to the server as the **aduser01@example.com** user:

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

2. Use the ssh command a second time to access the same server, this time as the **aduser02@example.com** user:

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

Notice how the **aduser02@example.com** user is denied access to the system. You have granted the permission to log in to the system to the **aduser01@example.com** user only. All other users from that Active Directory domain are rejected because of the specified login policy.

NOTE

If you set **use_fully_qualified_names** to true in the **sssd.conf** file, all requests must use the fully qualified domain name. However, if you set **use_fully_qualified_names** to false, it is possible to use the fully-qualified name in the requests, but only the simplified version is displayed in the output.

Additional resources

- **realm(8)** man page on your system

2.4.2. Denying access to users within a domain

By default, login policies for Active Directory (AD) users are defined in the AD domain itself. You can override this default behavior and configure a RHEL host to deny access to users within an AD domain.

IMPORTANT

It is safer to only allow access to specific users or groups than to deny access to some, while enabling it to everyone else. Therefore, it is not recommended to allow access to all by default while only denying it to specific users with realm permit **-x**. Instead, Red Hat recommends maintaining a default no access policy for all users and only grant access to selected users using realm permit.

Prerequisites

- Your RHEL system is a member of the Active Directory domain.

Procedure

1. Deny access to all users within the domain:

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

   This command prevents **realm** accounts from logging into the local machine. Use **realm permit** to restrict login to specific accounts.

2. Verify that the domain user's **login-policy** is set to **deny-any-login**:

```
[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. Deny access to specific users by using the **-x** option:

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

Verification

- Use SSH to log in to the server as the **aduser01@example.net** user.

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

NOTE

If you set **use_fully_qualified_names** to true in the **sssd.conf** file, all requests must use the fully qualified domain name. However, if you set **use_fully_qualified_names** to false, it is possible to use the fully-qualified name in the requests, but only the simplified version is displayed in the output.

Additional resources

- **realm(8)** man page on your system

CHAPTER 3. ACCESSING AD WITH A MANAGED SERVICE ACCOUNT

Active Directory (AD) Managed Service Accounts (MSAs) allow you to create an account in AD that corresponds to a specific computer. You can use an MSA to connect to AD resources as a specific user principal, without joining the RHEL host to the AD domain.

3.1. THE BENEFITS OF A MANAGED SERVICE ACCOUNT

If you want to allow a RHEL host to access an Active Directory (AD) domain without joining it, you can use a Managed Service Account (MSA) to access that domain. 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.

For example, if the AD domain **production.example.com** has a one-way trust relationship with the **lab.example.com** AD domain, the following conditions apply:

- The **lab** domain trusts users and hosts from the **production** domain.
- The **production** domain does **not** trust users and hosts from the **lab** domain.

This means that a host joined to the **lab** domain, such as **client.lab.example.com**, cannot access resources from the **production** domain through the trust.

If you want to create an exception for the **client.lab.example.com** host, you can use the **adcli** utility to create a MSA for the **client** host in the **production.example.com** domain. By authenticating with the Kerberos principal of the MSA, you can perform secure LDAP searches in the **production** domain from the **client** host.

3.2. CONFIGURING A MANAGED SERVICE ACCOUNT FOR A RHEL HOST

This procedure creates a Managed Service Account (MSA) for a host from the **lab.example.com** Active Directory (AD) domain, and configures SSSD so you can access and authenticate to the **production.example.com** AD domain.

NOTE

If you need to access AD resources from a RHEL host, Red Hat recommends that you join the RHEL host to the AD domain with the **realm** command. See Connecting RHEL systems directly to AD using SSSD.

Only perform this procedure if one of the following conditions applies:

- You cannot join the RHEL host to the AD domain, and you want to create an account for that host in AD.
- You have joined the RHEL host to an AD domain, and you need to access another AD domain where the host credentials from the domain you have joined are not valid, such as with a one-way trust.

Prerequisites

- Ensure that the following ports on the RHEL host are open and accessible to the AD domain controllers.

Service	Port	Protocols
DNS	53	TCP, UDP
LDAP	389	TCP, UDP
LDAPS (optional)	636	TCP, UDP
Kerberos	88	TCP, UDP

- You have the password for an AD Administrator that has rights to create MSAs in the **production.example.com** domain.
- You have root permissions that are required to run the **adcli** command, and to modify the **/etc/sssd/sssd.conf** configuration file..
- Optional: You have the **krb5-workstation** package installed, which includes the **klist** diagnostic utility.

Procedure

1. Create an MSA for the host in the **production.example.com** AD domain.

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

2. Display information about the MSA from the Kerberos keytab that was created. Make note of the MSA name:

```
[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. Open the **/etc/sssd/sssd.conf** file and choose the appropriate SSSD domain configuration to add:

   - If the MSA corresponds to an **AD domain from a different forest**, create a new domain section named **[domain/<name_of_domain>]**, and enter information about the MSA and the keytab. The most important options are **ldap_sasl_authid**, **ldap_krb5_keytab**, and **krb5_keytab**:

```
[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
...
```

WARNING

Even with an existing trust relationship, **sssd-ad** requires a MSA in the second forest.

- If the MSA corresponds to an **AD domain from the local forest**, create a new sub-domain section in the format **[domain/root.example.com/sub-domain.example.com]**, and enter information about the MSA and the keytab. The most important options are **ldap_sasl_authid**, **ldap_krb5_keytab**, and **krb5_keytab**:

```
[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
...
```

Verification

- Verify you can retrieve a Kerberos ticket-granting ticket (TGT) as the MSA:

```
[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
```

- In AD, verify you have an MSA for the host in the Managed Service Accounts Organizational Unit (OU).

3.3. UPDATING THE PASSWORD FOR A MANAGED SERVICE ACCOUNT

Managed Service Accounts (MSAs) have a complex password that is maintained automatically by Active Directory (AD). By default, the System Services Security Daemon (SSSD) automatically updates the MSA password in the Kerberos keytab if it is older than 30 days, which keeps it up to date with the password in AD. This procedure explains how to manually update the password for your MSA.

Prerequisites

- You have previously created an MSA for a host in the production.example.com AD domain.
- Optional: You have the **krb5-workstation** package installed, which includes the **klist** diagnostic utility.

Procedure

1. Optional: Display the current Key Version Number (KVNO) for the MSA in the Kerberos keytab. The current KVNO is 2.

```
[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. Update the password for the MSA in the **production.example.com** AD domain.

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

Verification

- Verify that you have incremented the KVNO in the Kerberos keytab:

```
[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)
```

3.4. MANAGED SERVICE ACCOUNT SPECIFICATIONS

The Managed Service Accounts (MSAs) that the **adcli** utility creates have the following specifications:

- They cannot have additional service principal names (SPNs).
- By default, the Kerberos principal for the MSA is stored in a Kerberos keytab named ***<default_keytab_location>.<Active_Directory_domain>***, like ***/etc/krb5.keytab.production.example.com***.
- MSA names are limited to 20 characters or fewer. The last 4 characters are a suffix of 3 random characters from number and upper- and lowercase ASCII ranges appended to the short host name you provide, using a **!** character as a separator. For example, a host with the short name **myhost** receives an MSA with the following specifications:

Specification	Value
Common name (CN) attribute	**myhost!A2c**
NetBIOS name	**myhost!A2c$**

Specification	Value
sAMAccountName	**myhost!A2c$**
Kerberos principal in the **production.example.com** AD domain	**myhost!A2c$@PRODUCTION.EXAMPLE.COM**

3.5. OPTIONS FOR THE ADCLI CREATE-MSA COMMAND

In addition to the global options you can pass to the **adcli** utility, you can specify the following options to specifically control how it handles Managed Service Accounts (MSAs).

-N, --computer-name

The short non-dotted name of the MSA that will be created in the Active Directory (AD) domain. If you do not specify a name, the first portion of the **--host-fqdn** or its default is used with a random suffix.

-O, --domain-ou=OU=*<path_to_OU>*

The full distinguished name of the Organizational Unit (OU) in which to create the MSA. If you do not specify this value, the MSA is created in the default location **OU=CN=Managed Service Accounts,DC=EXAMPLE,DC=COM**.

-H, --host-fqdn=host

Override the local machine’s fully qualified DNS domain name. If you do not specify this option, the host name of the local machine is used.

-K, --host-keytab=*<path_to_keytab>*

The path to the host keytab to store MSA credentials. If you do not specify this value, the default location **/etc/krb5.keytab** is used with the lower-cased Active Directory domain name added as a suffix, such as **/etc/krb5.keytab.domain.example.com**.

--use-ldaps

Create the MSA over a Secure LDAP (LDAPS) channel.

--verbose

Print out detailed information while creating the MSA.

--show-details

Print out information about the MSA after creating it.

--show-password

Print out the MSA password after creating the MSA.