Red Hat Enterprise Linux 8

Integrating RHEL systems directly with Windows Active Directory

Understanding and configuring RHEL systems to connect directly with Active Directory

Last Updated: 2021-05-13

Red Hat Enterprise Linux 8 Integrating RHEL systems directly with Windows Active Directory

Understanding and configuring RHEL systems to connect directly with Active Directory

Legal Notice

Abstract

This documentation collection provides instructions on how to integrate RHEL systems directly with Windows Active Directory using SSSD.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

CHAPTER 1. CONNECTING RHEL SYSTEMS DIRECTLY TO AD USING SSSD 5
1.1. OVERVIEW OF DIRECT INTEGRATION USING SSSD 5
1.2. SUPPORTED WINDOWS PLATFORMS FOR DIRECT INTEGRATION 6
1.3. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL 6
1.4. CONNECTING DIRECTLY TO AD 7
1.4.1. Discovering and joining an AD Domain using SSSD 7
1.4.2. Options for integrating with AD: using ID mapping or POSIX attributes 9
1.4.2.1. Automatically generate new UIDs and GIDs for AD users 9
1.4.2.2. Use POSIX attributes defined in AD 10
1.4.3. Connecting to AD using POSIX attributes defined in Active Directory 10
1.4.4. Connecting to multiple domains in different AD forests with SSSD 12
1.5. HOW THE AD PROVIDER HANDLES DYNAMIC DNS UPDATES 15
1.6. MODIFYING DYNAMIC DNS SETTINGS FOR THE AD PROVIDER 16
1.7. HOW THE AD PROVIDER HANDLES TRUSTED DOMAINS 16
1.8. REALM COMMANDS 17

CHAPTER 2. CONNECTING RHEL SYSTEMS DIRECTLY TO AD USING SAMBA WINBIND 19
2.1. OVERVIEW OF DIRECT INTEGRATION USING SAMBA WINBIND 19
2.2. SUPPORTED WINDOWS PLATFORMS FOR DIRECT INTEGRATION 19
2.3. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL 20
2.4. JOINING A RHEL SYSTEM TO AN AD DOMAIN 21
2.5. REALM COMMANDS 23

CHAPTER 3. MANAGING DIRECT CONNECTIONS TO AD 25
3.1. MODIFYING THE DEFAULT KERBEROS HOST KEYTAB RENEWAL INTERVAL 25
3.2. REMOVING A RHEL SYSTEM FROM AN AD DOMAIN 25
3.3. SETTING THE DOMAIN RESOLUTION ORDER IN SSSD TO RESOLVE SHORT AD USER NAMES 26
3.4. MANAGING LOGIN PERMISSIONS FOR DOMAIN USERS 27
3.4.1. Enabling access to users within a domain 27
3.4.2. Denying access to users within a domain 29
3.5. APPLYING GROUP POLICY OBJECT ACCESS CONTROL IN RHEL 30
3.5.1. How SSSD interprets GPO access control rules 30
3.5.1.1. Limitations on filtering by hosts 30
3.5.1.2. Limitations on filtering by groups 31
3.5.2. List of GPO settings that SSSD supports 31
3.5.3. List of SSSD options to control GPO enforcement 31
3.5.3.1. The ad_gpo_access_control option 31
3.5.3.2. The ad_gpo_implicit_deny option 32
3.5.4. Changing the GPO access control mode 33
3.5.5. Creating and configuring a GPO for a RHEL host in the AD GUI 34
3.5.6. Additional resources 35

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

In Identity Management, planned terminology replacements include:

- ***block list*** replaces *blacklist*
- ***allow list*** replaces *whitelist*
- ***secondary*** replaces *slave*
- The word *master* is going to be replaced with more precise language, depending on the context:
  - ***IdM server*** replaces *IdM master*
  - ***CA renewal server*** replaces *CA renewal master*
  - ***CRL publisher server*** replaces *CRL master*
  - ***multi-supplier*** replaces *multi-master*

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better. To do so:

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the Bugzilla website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.

CHAPTER 1. CONNECTING RHEL SYSTEMS DIRECTLY TO AD USING SSSD

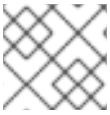
This section describes using the System Security Services Daemon (SSSD) to connect a RHEL system to Active Directory (AD). You need two components to connect a RHEL system to Active Directory (AD). One component, SSSD, interacts with the central identity and authentication source, and the other component, **realmd**, detects available domains and configures the underlying RHEL system services, in this case SSSD, to connect to the domain.

- Overview of direct integration using SSSD
- Supported Windows platforms for direct integration
- Ensuring support for common encryption types in AD and RHEL
- Connecting directly to AD
- How the AD provider handles dynamic DNS updates
- Modifying dynamic DNS settings for the AD provider
- How the AD provider handles trusted domains
- realm commands

1.1. OVERVIEW OF DIRECT INTEGRATION USING SSSD

You use SSSD to access a user directory for authentication and authorization through a common framework with user caching to permit offline logins. SSSD is highly configurable; it provides Pluggable Authentication Modules (PAM) and Name Switch Service (NSS) integration and a database to store local users as well as extended user data retrieved from a central server. SSSD is the recommended component to connect a RHEL system with one of the following types of identity server:

- Active Directory
- Identity Management (IdM) in RHEL
- Any generic LDAP or Kerberos server



NOTE

Direct integration with SSSD works only within a single AD forest by default.

The most convenient way to configure SSSD to directly integrate a Linux system with AD is to use the **realmd** service. It allows callers to configure network authentication and domain membership in a standard way. The **realmd** service automatically discovers information about accessible domains and realms and does not require advanced configuration to join a domain or realm.

You can use SSSD for both direct and indirect integration with AD and it allows you to switch from one integration approach to another. Direct integration is a simple way to introduce RHEL systems to an AD environment. However, as the share of RHEL systems grows, your deployments usually need a better centralized management of the identity-related policies such as host-based access control, sudo, or SELinux user mappings. Initially, you can maintain the configuration of these aspects of the RHEL systems in local configuration files. However, with a growing number of systems, distribution and management of the configuration files is easier with a provisioning system such as Red Hat Satellite.

When direct integration does not scale anymore, you should consider indirect integration. For more information on moving from direct integration (RHEL clients are in the AD domain) to indirect integration (IdM with trust to AD), see Moving RHEL clients from AD domain to IdM Server.

For more information on which type of integration fits your use case, see Deciding between indirect and direct integration.

Additional resources

- The **realm(8)** man page.
- The **sssd-ad(5)** man page.
- The **sssd(8)** man page.

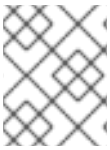
1.2. SUPPORTED WINDOWS PLATFORMS FOR DIRECT INTEGRATION

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:

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2



NOTE

Windows Server 2019 does not introduce a new functional level. The highest functional level Windows Server 2019 uses is Windows Server 2016.

1.3. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL

By default, SSSD supports RC4, AES-128, and AES-256 Kerberos encryption types.

RC4 encryption has been deprecated and disabled by default in RHEL 8, as it is considered less secure than the newer AES-128 and AES-256 encryption types. In contrast, Active Directory (AD) user credentials and trusts between AD domains support RC4 encryption and they might not support AES encryption types.

Without any common encryption types, communication between RHEL hosts and AD domains might not work, or some AD accounts might not be able to authenticate. To remedy this situation, modify one of the following configurations:

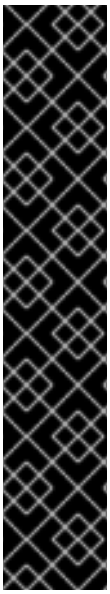
- **Enable AES encryption support in Active Directory (recommended option)** To ensure trusts between AD domains in an AD forest support strong AES encryption types, see the following Microsoft article: AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain

- **Enable RC4 support in RHEL**: On every RHEL host where authentication against AD Domain Controllers takes place:

  1. Use the **update-crypto-policies** command to enable the **AD-SUPPORT** cryptographic subpolicy in addition to the **DEFAULT** cryptographic policy.

     ```
     [root@host ~]# update-crypto-policies --set DEFAULT:AD-SUPPORT
     Setting system policy to DEFAULT:AD-SUPPORT
     Note: System-wide crypto policies are applied on application start-up.
     It is recommended to restart the system for the change of policies
     to fully take place.
     ```

  2. Restart the host.



IMPORTANT

The **AD-SUPPORT** cryptographic subpolicy is only available on RHEL 8.3 and newer.

- To enable support for RC4 in RHEL 8.2, create and enable a custom cryptographic module policy with **cipher = RC4-128+**. For more details, see Customizing system-wide cryptographic policies with policy modifiers .
- To enable support for RC4 in RHEL 8.0 and RHEL 8.1, add **+rc4** to the **permitted_enctypes** option in the **/etc/crypto-policies/back-ends/krb5.config** file:

  ```
  [libdefaults]
  permitted_enctypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-192 camellia256-cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-sha256-128 camellia128-cts-cmac +rc4
  ```

Additional resources

- For more information on working with RHEL cryptographic policies, see Using system-wide cryptographic policies in the Security Hardening guide.

1.4. CONNECTING DIRECTLY TO AD

This section describes how to integrate directly with AD using either ID mapping or POSIX attributes.

- Discovering and joining an AD domain using SSSD
- Options for integrating with AD: using ID mapping or POSIX attributes
- Connecting to AD using POSIX attributes defined in Active Directory
- Connecting to multiple domains in different AD forests with SSSD

1.4.1. Discovering and joining an AD Domain using SSSD

This procedure describes how to discover an AD domain and connect a RHEL system to that domain using SSSD.

Prerequisites

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

Table 1.1. Ports Required for Direct Integration of Linux Systems into AD Using SSSD

Service	Port	Protocol	Notes
DNS	53	UDP and TCP	
LDAP	389	UDP and TCP	
Kerberos	88	UDP and TCP	
Kerberos	464	UDP and TCP	Used by kadmin for setting and changing a password
LDAP Global Catalog	3268	TCP	If the **id_provider = ad** option is being used
NTP	123	UDP	Optional

- Ensure that you are using the AD domain controller server for DNS.
- Verify that the system time on both systems is synchronized. This ensures that Kerberos is able to work correctly.

Procedure

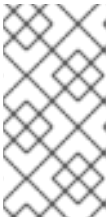
1. Install the following packages:

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

2. To display information for a specific domain, run **realm discover** and add the name of the domain you want to discover:

```
# 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
```

The **realmd** system uses DNS SRV lookups to find the domain controllers in this domain automatically.



NOTE

The **realmd** system can discover both Active Directory and Identity Management domains. If both domains exist in your environment, you can limit the discovery results to a specific type of server using the **--server-software=active-directory** option.

3. Configure the local RHEL system with the **realm join** command. The **realmd** suite edits all required configuration files automatically. For example, for a domain named **ad.example.com**:

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

Verification steps

- Display an AD user details, such as the administrator user:

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

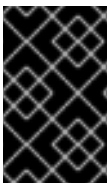
Additional resources

- See the **realm(8)** man page.
- See the **nmcli(1)** man page.

1.4.2. Options for integrating with AD: using ID mapping or POSIX attributes

Linux and Windows systems use different identifiers for users and groups:

- Linux uses *user IDs* (UID) and *group IDs* (GID). See Introduction to managing user and group accounts in *Configuring Basic System Settings*. Linux UIDs and GIDs are compliant with the POSIX standard.
- Windows use *security IDs* (SID).



IMPORTANT

After connecting a RHEL system to AD, you can authenticate with your AD username and password. Do not create a Linux user with the same name as a Windows user, as duplicate names might cause a conflict and interrupt the authentication process.

To authenticate to a RHEL system as an AD user, you must have a UID and GID assigned. SSSD provides the option to integrate with AD either using ID mapping or POSIX attributes. The default is to use ID mapping.

1.4.2.1. Automatically generate new UIDs and GIDs for AD users

SSSD can use the SID of an AD user to algorithmically generate POSIX IDs in a process called *ID mapping*. ID mapping creates a map between SIDs in AD and IDs on Linux.

- When SSSD detects a new AD domain, it assigns a range of available IDs to the new domain.
- When an AD user logs in to an SSSD client machine for the first time, SSSD creates an entry for the user in the SSSD cache, including a UID based on the user's SID and the ID range for that domain.
- Because the IDs for an AD user are generated in a consistent way from the same SID, the user has the same UID and GID when logging in to any Red Hat Enterprise Linux system.

See Discovering and joining an AD domain using SSSD .

NOTE

When all client systems use SSSD to map SIDs to Linux IDs, the mapping is consistent. If some clients use different software, choose one of the following:

- Ensure that the same mapping algorithm is used on all clients.
- Use explicit POSIX attributes defined in AD.

1.4.2.2. Use POSIX attributes defined in AD

AD can create and store POSIX attributes, such as **uidNumber**, **gidNumber**, **unixHomeDirectory**, or **loginShell**.

When using ID mapping described above, SSSD creates new UIDs and GIDs, which overrides the values defined in AD. To keep the AD-defined values, you must disable ID mapping in SSSD.

See Connecting to AD using POSIX attributes defined in Active Directory .

1.4.3. Connecting to AD using POSIX attributes defined in Active Directory

For best performance, publish the POSIX attributes to the AD global catalog. If POSIX attributes are not present in the global catalog, SSSD connects to the individual domain controllers directly on the LDAP port.

Prerequisites

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

Table 1.2. Ports Required for Direct Integration of Linux Systems into AD Using SSSD

Service	Port	Protocol	Notes
DNS	53	UDP and TCP	
LDAP	389	UDP and TCP	
Kerberos	88	UDP and TCP	

Service	Port	Protocol	Notes
Kerberos	464	UDP and TCP	Used by kadmin for setting and changing a password
LDAP Global Catalog	3268	TCP	If the **id_provider = ad** option is being used
NTP	123	UDP	Optional

- Ensure that you are using the AD domain controller server for DNS.
- Verify that the system time on both systems is synchronized. This ensures that Kerberos is able to work correctly.

Procedure

1. Install the following packages:

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

2. Configure the local RHEL system with ID mapping disabled using the **realm join** command with the **--automatic-id-mapping=no** option. The **realmd** suite edits all required configuration files automatically. For example, for a domain named **ad.example.com**:

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

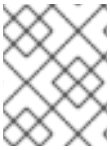
3. If you already joined a domain, you can manually disable ID Mapping in SSSD:
   a. Open the **/etc/sssd/sssd.conf** file.
   b. In the AD domain section, add the **ldap_id_mapping = false** setting.
   c. Remove the SSSD caches:

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

   d. Restart SSSD:

```
systemctl restart sssd
```

SSSD now uses POSIX attributes from AD, instead of creating them locally.



NOTE

You must have the relevant POSIX attributes (**uidNumber**, **gidNumber**, **unixHomeDirectory**, and **loginShell**) configured for the users in AD.

Verification steps

- Display an AD user details, such as the administrator user:

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

Additional resources

- For further details about ID mapping and the **ldap_id_mapping** parameter, see the **sssd-ldap(8)** man page.

1.4.4. Connecting to multiple domains in different AD forests with SSSD

This procedure describes joining and authenticating to multiple Active Directory (AD) domains in different forests where there is no trust between them.

This example describes joining two domains, **addomain1.com** and **addomain2.com**. Use **realmd** to join the first domain and automatically configure SSSD, Kerberos, and other utilities for that domain. Use **adcli** to join additional domains, and manually edit configuration files to include those domains.

Prerequisites

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

Table 1.3. Ports Required for Direct Integration of Linux Systems into AD Using SSSD

Service	Port	Protocol	Notes
DNS	53	UDP and TCP	
LDAP	389	UDP and TCP	
Kerberos	88	UDP and TCP	
Kerberos	464	UDP and TCP	Used by kadmin for setting and changing a password
LDAP Global Catalog	3268	TCP	If the **id_provider = ad** option is being used
NTP	123	UDP	Optional

- Ensure that you are using the AD domain controller server for DNS.
- Verify that the system time on both systems is synchronized. This ensures that Kerberos is able to work correctly.

- Ensure you have credentials for an AD administrator account in each AD domain which has rights to join machines to that domain

Procedure

1. Install required packages.

```
# yum install sssd realmd adcli samba-common-tools oddjob oddjob-mkhomedir
```

2. Use **realmd** to join the first AD domain, **addomain1.com**.

```
# realm join ADDOMAIN1.COM
```

3. Rename the system keytab to a unique name.

```
# mv /etc/krb5.keytab /etc/addomain1.com.krb5.keytab
```

4. Use **adcli** to join the second AD domain, and any additional domains. Use the **-K** option to specify a unique path for the Kerberos keytab where host credentials will be written.

```
# adcli join -D dc2.addomain2.com -K /etc/addomain2.com.krb5.keytab
```

5. Modify **/etc/krb5.conf**.

   - Add the **includedir** option to include SSSD configuration files.
   - Enable DNS lookups for AD Domain Controllers with the **dns_lookup_kdc** option.

```
includedir /var/lib/sss/pubconf/krb5.include.d/

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

[libdefaults]
 default_realm = ADDOMAIN1.COM
 dns_lookup_realm = false
 dns_lookup_kdc = true
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true

 ...
```

6. Modify **/etc/sssd/sssd.conf** to include information about all AD domains in use.

```
[sssd]
services = nss, pam
config_file_version = 2
domains = addomain1.com, addomain2.com

[domain/addomain1.com]
id_provider = ad
```

```
access_provider = ad
krb5_keytab = /etc/addomain1.com.krb5.keytab
ldap_krb5_keytab = /etc/addomain1.com.krb5.keytab
ad_server = dc1.addomain1.com
ad_maximum_machine_account_password_age = 0
use_fully_qualified_names = true
default_shell=/bin/bash
override_homedir=/home/%d/%u

[domain/addomain2.com]
id_provider = ad
access_provider = ad
krb5_keytab = /etc/addomain2.com.krb5.keytab
ldap_krb5_keytab = /etc/addomain2.com.krb5.keytab
ad_server = dc2.addomain2.com
ad_maximum_machine_account_password_age = 0
use_fully_qualified_names = true
default_shell=/bin/bash
override_homedir=/home/%d/%u

[nss]

[pam]
```

- For each domain section, specify the path to the Kerberos keytab that corresponds to each domain with the **krb5_keytab** and **ldap_krb5_keytab** options.
- Set **ad_maximum_machine_account_password_age = 0** to disable renewing host Kerberos keys.
- Set **use_fully_qualified_names = true** to differentiate users from different domains.
- Set **override_homedir = /home/%d/%u** so users (**%u**) from different domains (**%d**) each receive unique home directories. For example, the home directory for user **linuxuser@addomain1.com** is **/home/addomain1.com/linuxuser**.

7. SSH retrieves host keys from the system keytab and provides single sign-on functionality through GSSAPI/Kerberos. If you would like to use single sign-on, copy all current Kerberos host keys to the **/etc/kbr5.keytab** system keytab.

```
# ktutil
ktutil:  rkt /etc/addomain1.com.krb5.keytab
ktutil:  rkt /etc/addomain2.com.krb5.keytab
ktutil:  wkt /etc/krb5.keytab
```

8. Restart and enable the SSSD service.

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

Verification steps

1. Display user details for users from each AD domain:

```
# id administrator@addomain1.com
```

```
uid=1240800500(administrator@addomain1.com) gid=1240800513(domain
users@addomain1.com) groups=1240800513(domain
users@addomain1.com),1240800512(domain
admins@addomain1.com),1240800518(schema
admins@addomain1.com),1240800520(group policy creator
owners@addomain1.com),1240800572(denied rodc password replication
group@addomain1.com),1240800519(enterprise admins@addomain1.com)

# id administrator@addomain2.com
uid=1013800500(administrator@addomain2.com)
gid=1013800500(administrator@addomain2.com)
groups=1013800500(administrator@addomain2.com),1013800513(domain
users@addomain2.com)
```

2. Log in as a user from each domain and verify the correct home directory is created for the user.

```
# ssh administrator@addomain1.com@localhost
administrator@addomain1.com@localhost's password:
Creating directory '/home/addomain1.com/administrator'.

$ pwd
/home/addomain1.com/administrator
```

```
# ssh administrator@addomain2.com@localhost
administrator@addomain2.com@localhost's password:
Creating directory '/home/addomain2.com/administrator'.

$ pwd
/home/addomain2.com/administrator
```

1.5. HOW THE AD PROVIDER HANDLES DYNAMIC DNS UPDATES

Active Directory (AD) actively maintains its DNS records by timing out (*aging*) and removing (*scavenging*) inactive records.

By default, the SSSD service refreshes a RHEL client's DNS record at the following intervals:

- Every time the identity provider comes online.
- Every time the RHEL system reboots.
- At the interval specified by the **dyndns_refresh_interval** option in the **/etc/sssd/sssd.conf** configuration file. The default value is **86400** seconds (24 hours).

NOTE

If you set the **dyndns_refresh_interval** option to the same interval as the DHCP lease, you can update the DNS record after the IP lease is renewed.

SSSD sends dynamic DNS updates to the AD server using Kerberos/GSSAPI for DNS (GSS-TSIG). This means that you only need to enable secure connections to AD.

Additional resources

- The **sssd-ad(5)** man page.

1.6. MODIFYING DYNAMIC DNS SETTINGS FOR THE AD PROVIDER

The following procedure adjusts settings within the SSSD service to affect how it automatically updates the DNS record for a RHEL host joined to an Active Directory environment.

Prerequisites

- You have joined a RHEL host to an Active Directory environment with the SSSD service.
- You need **root** permissions to edit the **/etc/sssd/sssd.conf** configuration file.

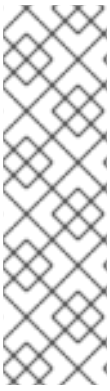
Procedure

1. Open the **/etc/sssd/sssd.conf** configuration file in a text editor.
2. Add the following options to the **[domain]** section for your AD domain to set the DNS record refresh interval to 12 hours, disable updating PTR records, and set the DNS record Time To Live (TTL) to 1 hour.

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

3. Save and close the **/etc/sssd/sssd.conf** configuration file.
4. Restart the SSSD service to load the configuration changes.

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



NOTE

You can disable dynamic DNS updates by setting the **dyndns_update** option in the **sssd.conf** file to **false**:

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

dyndns_update = false
```

Additional resources

- **sssd-ad(5)** man page

1.7. HOW THE AD PROVIDER HANDLES TRUSTED DOMAINS

This section describes how SSSD handles trusted domains if you set the **id_provider = ad** option in the **/etc/sssd/sssd.conf** configuration file.

- SSSD only supports domains in a single AD forest. If SSSD requires access to multiple domains from multiple forests, consider using IPA with trusts (preferred) or the **winbindd** service instead of SSSD.

- By default, SSSD discovers all domains in the forest and, if a request for an object in a trusted domain arrives, SSSD tries to resolve it.
  If the trusted domains are not reachable or geographically distant, which makes them slow, you can set the **ad_enabled_domains** parameter in **/etc/sssd/sssd.conf** to limit from which trusted domains SSSD resolves objects.

- By default, you must use fully-qualified user names to resolve users from trusted domains.

Additional resources

- The **sssd.conf(5)** man page.

1.8. 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 1.4. 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.

For more information about the **realm** commands, see the **realm(8)** man page.

CHAPTER 2. CONNECTING RHEL SYSTEMS DIRECTLY TO AD USING SAMBA WINBIND

This section describes using Samba Winbind to connect a RHEL system to Active Directory (AD). You need two components to connect a RHEL system to AD. One component, Samba Winbind, interacts with the AD identity and authentication source, and the other component, **realmd**, detects available domains and configures the underlying RHEL system services, in this case Samba Winbind, to connect to the AD domain.

- Overview of direct integration using Samba Winbind
- Supported Windows platforms for direct integration
- Ensuring support for common encryption types in AD and RHEL
- Joining a RHEL system to an AD domain
- realm commands

2.1. 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. For further detail, see the section about Using Samba as a server in the Deploying Different Types of Servers Guide .

Additional resources

- See the **realmd** man page.
- See the **windbindd** man page.

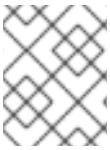
2.2. SUPPORTED WINDOWS PLATFORMS FOR DIRECT INTEGRATION

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:

- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2



NOTE

Windows Server 2019 does not introduce a new functional level. The highest functional level Windows Server 2019 uses is Windows Server 2016.

2.3. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL

By default, Samba Winbind supports RC4, AES-128, and AES-256 Kerberos encryption types.

RC4 encryption has been deprecated and disabled by default in RHEL 8, as it is considered less secure than the newer AES-128 and AES-256 encryption types. In contrast, Active Directory (AD) user credentials and trusts between AD domains support RC4 encryption and they might not support AES encryption types.

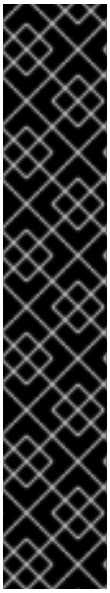
Without any common encryption types, communication between RHEL hosts and AD domains might not work, or some AD accounts might not be able to authenticate. To remedy this situation, modify one of the following configurations:

- **Enable AES encryption support in Active Directory (recommended option)** To ensure trusts between AD domains in an AD forest support strong AES encryption types, see the following Microsoft article: AD DS: Security: Kerberos "Unsupported etype" error when accessing a resource in a trusted domain
- **Enable RC4 support in RHEL**: On every RHEL host where authentication against AD Domain Controllers takes place:

  1. Use the **update-crypto-policies** command to enable the **AD-SUPPORT** cryptographic subpolicy in addition to the **DEFAULT** cryptographic policy.

     ```
     [root@host ~]# update-crypto-policies --set DEFAULT:AD-SUPPORT
     Setting system policy to DEFAULT:AD-SUPPORT
     Note: System-wide crypto policies are applied on application start-up.
     It is recommended to restart the system for the change of policies
     to fully take place.
     ```

  2. Restart the host.



IMPORTANT

The **AD-SUPPORT** cryptographic subpolicy is only available on RHEL 8.3 and newer.

- To enable support for RC4 in RHEL 8.2, create and enable a custom cryptographic module policy with **cipher = RC4-128+**. For more details, see Customizing system-wide cryptographic policies with policy modifiers .
- To enable support for RC4 in RHEL 8.0 and RHEL 8.1, add **+rc4** to the **permitted_enctypes** option in the **/etc/crypto-policies/back-ends/krb5.config** file:

```
[libdefaults]
permitted_enctypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-192 camellia256-cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-sha256-128 camellia128-cts-cmac +rc4
```

Additional resources

- For more information on working with RHEL cryptographic policies, see Using system-wide cryptographic policies in the Security Hardening guide.

2.4. JOINING A RHEL SYSTEM TO AN AD DOMAIN

This section describes how to join a Red Hat Enterprise Linux system to an AD domain by using **realmd** to configure Samba Winbind.

Procedure

1. If your AD requires the deprecated RC4 encryption type for Kerberos authentication, enable support for these ciphers in RHEL:

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

2. Install the following packages:

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

3. To share directories or printers on the domain member, install the **samba** package:

```
# yum install samba
```

4. Backup the existing **/etc/samba/smb.conf** Samba configuration file:

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

5. Join the domain. For example, to join a domain named **ad.example.com**:

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

Using the previous command, the **realm** utility automatically:

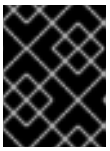
- Creates a **/etc/samba/smb.conf** file for a membership in the **ad.example.com** domain
- Adds the **winbind** module for user and group lookups to the **/etc/nsswitch.conf** file
- Updates the Pluggable Authentication Module (PAM) configuration files in the **/etc/pam.d/** directory
- Starts the **winbind** service and enables the service to start when the system boots

6. Optionally, set an alternative ID mapping back end or customized ID mapping settings in the **/etc/samba/smb.conf** file. For details, see the Understanding and configuring Samba ID mapping section in the **Deploying different types of servers** documentation.
7. Edit the **/etc/krb5.conf** file and add the following section:

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

8. Verify that the **winbind** service is running:

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



IMPORTANT

To enable Samba to query domain user and group information, the **winbind** service must be running before you start **smb**.

9. If you installed the **samba** package to share directories and printers, enable and start the **smb** service:

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

Verification steps

1. Display an AD user's details, such as the AD administrator account in the AD domain:

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

2. Query the members of the domain users group in the AD domain:

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

3. Optionally, verify that you can use domain users and groups when you set permissions on files and directories. For example, to set the owner of the **/srv/samba/example.txt** file to **AD\administrator** and the group to **AD\Domain Users**:

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

4. Verify that Kerberos authentication works as expected:

   a. On the AD domain member, obtain a ticket for the **administrator@AD.EXAMPLE.COM** principal:

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

   b. Display the cached Kerberos ticket:

   ```
   # 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. Display the available domains:

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

Additional resources

- If you do not want to use the deprecated RC4 ciphers, you can enable the AES encryption type in AD. See Enabling the AES encryption type in Active Directory using a GPO in the **Deploying different types of servers** documentation.
- For further details about the **realm** utility, see the **realm(8)** man page.

2.5. 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 2.1. realmd Commands

Command	Description
Realm Commands	

Command	Description
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.

For more information about the **realm** commands, see the **realm(8)** man page.

CHAPTER 3. MANAGING DIRECT CONNECTIONS TO AD

This section describes how to modify and manage your connection to Active Directory.

Prerequisites

- You have connected your RHEL system to the Active Directory domain.

3.1. 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

- The **adcli(8)** man page.
- The **sssd.conf(5)** man page.

3.2. REMOVING A RHEL SYSTEM FROM AN AD DOMAIN

This procedure describes how to remove a RHEL system from an Active Directory (AD) domain.

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, the account is not deleted from AD; the local client configuration is only removed. If you want to delete the AD account, run the command with the **--remove** option. You are prompted for your user password and you must have the 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 steps

- 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

- See the **realm(8)**` man page.

3.3. 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 **[sssd]** 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 Steps

- 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)
```

3.4. 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.

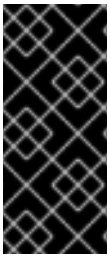
If a domain applies client-side access control, you can use the **realmd** to configure basic allow or deny access rules for users from that domain.

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.

3.4.1. Enabling access to users within a domain

This section describes how to enable access to users within a 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. 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 steps

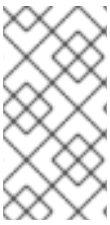
1. Use SSH to log in to the server as the aduser01@example.com user:

```
$ 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** 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

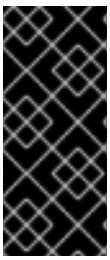
If you set **use_fully_qualified_names** to true in the **sssd.conf** file, all requests must use the fully qualified domain name. However, if you set **use_fully_qualified_names** to false, it is possible to use the fully-qualified name in the requests, but only the simplified version is displayed in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

Additional resources

- See the **realm(8)`** man page.

3.4.2. Denying access to users within a domain

This section describes how to deny access to all users within a 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 steps

- 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

- See the **realm(8)`** man page.

3.5. APPLYING GROUP POLICY OBJECT ACCESS CONTROL IN RHEL

A *Group Policy Object* (GPO) is a collection of access control settings stored in Microsoft Active Directory (AD) that can apply to computers and users in an AD environment. By specifying GPOs in AD, administrators can define login policies honored by both Windows clients and Red Hat Enterprise Linux (RHEL) hosts joined to AD.

The following sections describe how you can manage GPOs in your environment:

- Section 3.5.1, “How SSSD interprets GPO access control rules”
- Section 3.5.2, “List of GPO settings that SSSD supports”
- Section 3.5.3, “List of SSSD options to control GPO enforcement”
- Section 3.5.4, “Changing the GPO access control mode”
- Section 3.5.5, “Creating and configuring a GPO for a RHEL host in the AD GUI”

3.5.1. How SSSD interprets GPO access control rules

By default, SSSD retrieves Group Policy Objects (GPOs) from Active Directory (AD) domain controllers and evaluates them to determine if a user is allowed to log in to a particular RHEL host joined to AD.

SSSD maps AD *Windows Logon Rights* to Pluggable Authentication Module (PAM) service names to enforce those permissions in a GNU/Linux environment.

As an AD Administrator, you can limit the scope of GPO rules to specific users, groups, or hosts by listing them in a *security filter*.

3.5.1.1. Limitations on filtering by hosts

Older versions of SSSD do not evaluate hosts in AD GPO security filters.

- **RHEL 8.3.0 and newer:** SSSD supports users, groups, and hosts in security filters.
- **RHEL versions older than 8.3.0:** SSSD ignores host entries and only supports users and groups in security filters.
  To ensure that SSSD applies GPO-based access control to a specific host, create a new

Organizational Unit (OU) in the AD domain, move the system to the new OU, and then link the GPO to this OU.

3.5.1.2. Limitations on filtering by groups

SSSD currently does not support Active Directory's built-in groups, such as **Administrators** with Security Identifier (SID) **S-1-5-32-544**. Red Hat recommends against using AD built-in groups in AD GPOs targeting RHEL hosts.

Additional resources

- For a list of Windows GPO options and their corresponding SSSD options, see List of GPO settings that SSSD supports.

3.5.2. List of GPO settings that SSSD supports

The following table shows the SSSD options that correspond to Active Directory GPO options as specified in the *Group Policy Management Editor* on Windows.

Table 3.1. GPO access control options retrieved by SSSD

GPO option	Corresponding **sssd.conf** option
Allow log on locally Deny log on locally	**ad_gpo_map_interactive**
Allow log on through Remote Desktop Services Deny log on through Remote Desktop Services	**ad_gpo_map_remote_interactive**
Access this computer from the network Deny access to this computer from the network	**ad_gpo_map_network**
Allow log on as a batch job Deny log on as a batch job	**ad_gpo_map_batch**
Allow log on as a service Deny log on as a service	**ad_gpo_map_service**

- For more information on these **sssd.conf** settings, such as the Pluggable Authentication Module (PAM) services that map to GPO options, see the **sssd-ad(5)** Manual page entry.

3.5.3. List of SSSD options to control GPO enforcement

3.5.3.1. The ad_gpo_access_control option

You can set the **ad_gpo_access_control** option in the **/etc/sssd/sssd.conf** file to choose between three different modes in which GPO-based access control operates.

Table 3.2. Table of ad_gpo_access_control values

Value of ad_gpo_access_control	Behavior
enforcing	GPO-based access control rules are evaluated and enforced. **This is the default setting in RHEL 8.**
permissive	GPO-based access control rules are evaluated but **not** enforced; a **syslog** message is recorded every time access would be denied. This is the default setting in RHEL 7. This mode is ideal for testing policy adjustments while allowing users to continue logging in.
disabled	GPO-based access control rules are neither evaluated nor enforced.

3.5.3.2. The ad_gpo_implicit_deny option

The **ad_gpo_implicit_deny** option is set to **False** by default. In this default state, users are allowed access if applicable GPOs are not found. If you set this option to **True**, you must explicitly allow users access with a GPO rule.

You can use this feature to harden security, but be careful not to deny access unintentionally. Red Hat recommends testing this feature while **ad_gpo_access_control** is set to **permissive**.

The following two tables illustrate when a user is allowed or rejected access based on the allow and deny login rights defined on the AD server-side and the value of **ad_gpo_implicit_deny**.

Table 3.3. Login behavior with ad_gpo_implicit_deny set to False (default)

allow-rules	deny-rules	result
missing	missing	all users are allowed
missing	present	only users not in deny-rules are allowed
present	missing	only users in allow-rules are allowed
present	present	only users in allow-rules and not in deny-rules are allowed

Table 3.4. Login behavior with ad_gpo_implicit_deny set to True

allow-rules	deny-rules	result
missing	missing	no users are allowed
missing	present	no users are allowed
present	missing	only users in allow-rules are allowed

allow-rules	deny-rules	result
present	present	only users in allow-rules and not in deny-rules are allowed

Additional resources

- For the procedure to change the GPO enforcement mode in SSSD, see Changing the GPO access control mode.
- For more details on each of the different GPO modes of operation, see the **ad_gpo_access_control** entry in the **sssd-ad(5)** Manual page.

3.5.4. Changing the GPO access control mode

This procedure changes how GPO-based access control rules are evaluated and enforced on a RHEL host joined to an Active Directory (AD) environment.

In this example, you will change the GPO operation mode from **enforcing** (the default) to **permissive** for testing purposes.

IMPORTANT

If you see the following errors, Active Directory users are unable to log in due to GPO-based access controls:

- In **/var/log/secure**:

```
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]
```

- In **/var/log/sssd/sssd__example.com_.log**:

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

If this is undesired behavior, you can temporarily set **ad_gpo_access_control** to **permissive** as described in this procedure while you troubleshoot proper GPO settings in AD.

Prerequisites

- You have joined a RHEL host to an AD environment using SSSD.

- Editing the **/etc/sssd/sssd.conf** configuration file requires **root** permissions.

Procedure

1. Stop the SSSD service.

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

2. Open the **/etc/sssd/sssd.conf** file in a text editor.

3. Set **ad_gpo_access_control** to **permissive** in the **domain** section for the AD domain.

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

4. Save the **/etc/sssd/sssd.conf** file.

5. Restart the SSSD service to load configuration changes.

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

Additional resources

- For the list of different GPO access control modes, see List of SSSD options to control GPO enforcement.

3.5.5. Creating and configuring a GPO for a RHEL host in the AD GUI

The following procedure creates a Group Policy Object (GPO) in the Active Directory (AD) graphical user interface (GUI) to control logon access to a RHEL host.

Prerequisites

- You have joined a RHEL host to an AD environment using SSSD.
- You have AD Administrator privileges to make changes in AD using the GUI.

Procedure

1. Within **Active Directory Users and Computers**, create an Organizational Unit (OU) to associate with the new GPO:
   a. Right-click on the domain.
   b. Choose **New**.
   c. Choose **Organizational Unit**.
2. Click on the name of the Computer Object that represents the RHEL host (created when it joined Active Directory) and drag it into the new OU. By having the RHEL host in its own OU, the GPO targets this host.
3. Within the **Group Policy Management Editor**, create a new GPO for the OU you created:

a. Expand **Forest**.

b. Expand **Domains**.

c. Expand your domain.

d. Right-click on the new OU.

e. Choose **Create a GPO in this domain**.

4. Specify a name for the new GPO, such as **Allow SSH access** or **Allow Console/GUI access** and click **OK**.

5. Edit the new GPO:

a. Select the OU within the **Group Policy Management** editor.

b. Right-click and choose **Edit**.

c. Select **User Rights Assignment**.

d. Select **Computer Configuration**

e. Select **Policies**.

f. Select **Windows Settings**.

g. Select **Security Settings**.

h. Select **Local Policies**.

i. Select **User Rights Assignment**.

6. Assign login permissions:

a. Double-Click on **Allow log on locally** to grant local console/GUI access.

b. Double-click on **Allow log on through Remote Desktop Services** to grant SSH access.

7. Add the user(s) you would like to access either of these policies to the policies themselves:

a. Click **Add User or Group**.

b. Enter the username within the blank field.

c. Click **OK**.

Additional resources

- For more details on Group Policy Objects, see Group Policy Objects in Microsoft documentation.

3.5.6. Additional resources

- For more information on joining a RHEL host to an Active Directory environment, see Connecting RHEL systems directly to AD using SSSD