Red Hat Enterprise Linux 9.0 Beta

Installing trust between IdM and AD

Configuring a cross-forest trust between an IdM domain and an Active Directory domain.
Configuring a cross-forest trust between an IdM domain and an Active Directory domain.
Abstract

This documentation collection provides instructions on how to create a trust between an Identity Management deployment and an Active Directory (AD) environment.
# Table of Contents

- **MAKING OPEN SOURCE MORE INCLUSIVE** .......................................................... 3
- **PROVIDING FEEDBACK ON RED HAT DOCUMENTATION** .......................... 4
- **CHAPTER 1. PREREQUISITES FOR ESTABLISHING A TRUST** ......................... 5
- **CHAPTER 2. SUPPORTED VERSIONS OF WINDOWS SERVER** ......................... 6
- **CHAPTER 3. HOW THE TRUST WORKS** ...................................................... 7
- **CHAPTER 4. AD ADMINISTRATION RIGHTS** .............................................. 8
- **CHAPTER 5. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL** .................. 9
- **CHAPTER 6. PORTS REQUIRED FOR COMMUNICATION BETWEEN IDM AND AD** .................. 10
- **CHAPTER 7. CONFIGURING DNS AND REALM SETTINGS FOR A TRUST** ............ 13
  - 7.1. UNIQUE PRIMARY DNS DOMAINS .................................................. 13
  - 7.2. CONFIGURING A DNS FORWARD ZONE IN THE IDM WEB UI .................. 14
  - 7.3. CONFIGURING A DNS FORWARD ZONE IN THE CLI ............................. 17
  - 7.4. CONFIGURING DNS FORWARDING IN AD ...................................... 18
  - 7.5. VERIFYING THE DNS CONFIGURATION ...................................... 19
- **CHAPTER 8. CONFIGURING IDM CLIENTS IN AN ACTIVE DIRECTORY DNS DOMAIN** .................. 21
  - 8.1. CONFIGURING AN IDM CLIENT WITHOUT KERBEROS SINGLE SIGN-ON ........... 21
  - 8.2. REQUESTING SSL CERTIFICATES WITHOUT SINGLE SIGN-ON .................. 21
  - 8.3. CONFIGURING AN IDM CLIENT WITH KERBEROS SINGLE SIGN-ON ............ 22
  - 8.4. REQUESTING SSL CERTIFICATES WITH SINGLE SIGN-ON ..................... 22
- **CHAPTER 9. SETTING UP A TRUST** ......................................................... 24
  - 9.1. PREPARING THE IDM SERVER FOR THE TRUST .................................. 24
  - 9.2. SETTING UP A TRUST AGREEMENT USING THE COMMAND LINE .............. 26
  - 9.3. SETTING UP A TRUST AGREEMENT IN THE IDM WEB UI ..................... 27
  - 9.4. VERIFYING THE KERBEROS CONFIGURATION .................................. 30
  - 9.5. VERIFYING THE TRUST CONFIGURATION ON IDM ............................. 30
  - 9.6. VERIFYING THE TRUST CONFIGURATION ON AD .............................. 31
  - 9.7. CREATING A TRUST AGENT ...................................................... 33
  - 9.8. ENABLING AUTOMATIC PRIVATE GROUP MAPPING FOR A POSIX ID RANGE ON THE CLI ........ 33
  - 9.9. ENABLING AUTOMATIC PRIVATE GROUP MAPPING FOR A POSIX ID RANGE IN THE IDM WEB UI ........ 34
- **CHAPTER 10. REMOVING THE TRUST USING THE COMMAND LINE** .................. 37
- **CHAPTER 11. REMOVING THE TRUST USING THE IDM WEB UI** ....................... 38
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 being 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](https://bugzilla.redhat.com) 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. PREREQUISITES FOR ESTABLISHING A TRUST

This documentation aims to help you create a trust between your Identity Management IdM server and Active Directory (AD), where both servers are located in the same forest.

Prerequisites

- First, read the Planning a cross-forest trust between Identity Management and Active Directory document.

- AD is installed with a domain controller on it.

- The IdM server is installed and running. For details, see Installing Identity Management.

- Both the AD server and the IdM server must have their clocks in sync because Kerberos requires max 5 mins delay in communication.

- Unique NetBIOS names for each of the servers placed in the trust because the NetBIOS names are critical for identifying the Active Directory domain. The NetBIOS name of an Active Directory or IdM domain is usually the first part of the corresponding DNS domain. If the DNS domain is ad.example.com, the NetBIOS name is typically AD. However, it is not required. Important is that the NetBIOS name is just one word without periods. The maximum length of a NetBIOS name is 15 characters.

- The IdM system must have the IPv6 protocol enabled in the kernel. If IPv6 is disabled, then the CLDAP plug-in used by the IdM services fails to initialize.
CHAPTER 2. SUPPORTED VERSIONS OF WINDOWS SERVER

Identity Management (IdM) does not support establishing trust to Active Directory with Active Directory domain controllers running Windows Server 2008 R2 or earlier versions. RHEL IdM requires SMB encryption when establishing the trust relationship, which is only supported in Windows Server 2012 or later.

You can establish a trust relationship with Active Directory (AD) forests that use the following forest and domain functional levels:


Identity Management (IdM) supports establishing a trust with Active Directory domain controllers running the following operating systems:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019
CHAPTER 3. HOW THE TRUST WORKS

The trust between Identity Management IdM and Active Directory (AD) is established on the Cross-realm Kerberos trust. This solution uses the Kerberos capability to establish trusts between different identity sources. Therefore, all AD users can:

- Log in to access Linux systems and resources.
- Use single sign-on (SSO).

All IdM objects are managed in IdM in the trust.

All AD objects are managed in AD in the trust.

In complex environments, a single IdM forest can be connected to multiple AD forests. This setup enables better separation of duties for different functions in the organization. AD administrators can focus on users and policies related to users while Linux administrators have full control over the Linux infrastructure. In such a case, the Linux realm controlled by IdM is analogous to an AD resource domain or realm but with Linux systems in it.

From the perspective of AD, Identity Management represents a separate AD forest with a single AD domain. When cross-forest trust between an AD forest root domain and an IdM domain is established, users from the AD forest domains can interact with Linux machines and services from the IdM domain.

NOTE

In trust environments, IdM enables you to use ID views to configure POSIX attributes for AD users on the IdM server.
CHAPTER 4. AD ADMINISTRATION RIGHTS

When you want to build a trust between AD (Active Directory) and IdM (Identity Management), you will need to use an AD administrator account with appropriate AD privileges.

Such an AD administrator must be a member of one of the following groups:

- Enterprise Admin group in the AD forest
- Domain Admins group in the forest root domain for your AD forest

Additional resources

- For details about Enterprise Admins, see Enterprise Admins.
- For details about Domain Admins, see Domain Admins.
- For details about AD trust, see How Domain and Forest Trusts Work.
CHAPTER 5. ENSURING SUPPORT FOR COMMON ENCRYPTION TYPES IN AD AND RHEL

By default, Identity Management establishes a cross-realm trust with support for RC4, AES-128, and AES-256 Kerberos encryption types.

RC4 encryption has been deprecated and disabled by default, 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 IdM and AD child 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 IdM trust controller, trust agent, and client 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.

     ```bash
     [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.

Additional resources

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

- For more information on IdM trust agents and trust controllers, see Trust controllers and trust agents in the Planning Identity Management guide.
CHAPTER 6. PORTS REQUIRED FOR COMMUNICATION BETWEEN IDM AND AD

To enable communication between your Active Directory (AD) and Identity Management (IdM) environments, open the following ports on the firewalls of your AD Domain Controllers and IdM servers.

Table 6.1. Ports required for an AD trust

<table>
<thead>
<tr>
<th>Service</th>
<th>Port</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint resolution portmapper</td>
<td>135</td>
<td>TCP</td>
</tr>
<tr>
<td>NetBIOS-DGM</td>
<td>138</td>
<td>TCP and UDP</td>
</tr>
<tr>
<td>NetBIOS-SSN</td>
<td>139</td>
<td>TCP and UDP</td>
</tr>
<tr>
<td>Microsoft-DS</td>
<td>445</td>
<td>TCP and UDP</td>
</tr>
<tr>
<td>Dynamic RPC</td>
<td>49152-65535</td>
<td>TCP</td>
</tr>
<tr>
<td>AD Global Catalog</td>
<td>3268</td>
<td>TCP</td>
</tr>
<tr>
<td>LDAP</td>
<td>389</td>
<td>TCP and UDP</td>
</tr>
</tbody>
</table>

**NOTE**

The TCP port 389 is not required to be open on IdM servers for trust, but it is necessary for clients communicating with the IdM server.

To open ports, you can use the following methods:

- Firewalld service – you can enable the particular ports or enable the following services which includes the ports:
  - FreeIPA trust setup
  - FreeIPA with LDAP
  - Kerberos
  - DNS

For details, see Controlling ports using CLI.
The `freeipa-trust` Firewalld service currently includes an RPC port range of 1024-1300, but this range has been updated to 49152-65535 in Windows Server 2008 and later. The `freeipa-trust` Firewalld service will be updated to reflect this new range, and this issue is tracked in Bug 1850418 - update freeipa-trust.xml definition to include correct dynamic RPC range.

Until that bug has been resolved, manually open the TCP port range 49152-65535 in addition to enabling the `freeipa-trust` Firewalld service.

- The RHEL web console, which is a UI with firewall settings based on the `firewalld` service.

For details about firewall configuration through the web console, see Enabling services on the firewall using the web console.

### Table 6.2. Ports required by IdM servers in a trust

<table>
<thead>
<tr>
<th>Service</th>
<th>Port</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerberos</td>
<td>88, 464</td>
<td>TCP and UDP</td>
</tr>
<tr>
<td>LDAP</td>
<td>389</td>
<td>TCP</td>
</tr>
</tbody>
</table>
Table 6.3. Ports required by IdM clients in an AD trust

<table>
<thead>
<tr>
<th>Service</th>
<th>Port</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS</td>
<td>53</td>
<td>TCP and UDP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service</th>
<th>Port</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kerberos</td>
<td>88</td>
<td>UDP and TCP</td>
</tr>
</tbody>
</table>

**NOTE**

The **libkrb5** library uses UDP and falls back to the TCP protocol if the data sent from the Key Distribution Centre (KDC) is too large. Active Directory attaches a Privilege Attribute Certificate (PAC) to the Kerberos ticket, which increases the size and requires to use the TCP protocol. To avoid the fall-back and resending the request, by default, SSSD in Red Hat Enterprise Linux 7.4 and later uses TCP for user authentication. If you want to configure the size before libkrb5 uses TCP, set the `udp_preference_limit` in the `/etc/krb5.conf` file. For details, see the `krb5.conf(5)` man page.

**Additional resources**

- For more information on the Dynamic RPC port range in Windows Server 2008 and later, see [The default dynamic port range for TCP/IP has changed since Windows Vista and in Windows Server 2008.](#)
CHAPTER 7. CONFIGURING DNS AND REALM SETTINGS FOR A TRUST

Before you connect Identity Management (IdM) and Active Directory (AD) in a trust, you need to ensure that servers see each other and resolve domain names correctly. This scenario describes configuring DNS to allow using domain names between:

- One primary IdM server using integrated DNS server and Certification Authority.
- One AD Domain Controller.

DNS settings require:

- Configuring DNS zones in the IdM server
- Configuring conditional DNS forwarding in AD
- Verifying correctness of the DNS configuration

7.1. UNIQUE PRIMARY DNS DOMAINS

In Windows, every domain is a Kerberos realm and a DNS domain at the same time. Every domain managed by the domain controller needs to have its own dedicated DNS zone. The same applies when Identity Management (IdM) is trusted by Active Directory (AD) as a forest. AD expects IdM to have its own DNS domain. For the trust setup to work, the DNS domain needs to be dedicated to the Linux environment.

Each system must have its own unique primary DNS domain configured. For example:

- `ad.example.com` for AD and `idm.example.com` for IdM
- `example.com` for AD and `idm.example.com` for IdM
- `ad.example.com` for AD and `example.com` for IdM

The most convenient management solution is an environment where each DNS domain is managed by integrated DNS servers, but it is possible to use any other standard-compliant DNS server as well.

Kerberos realm names as upper-case versions of primary DNS domain names

Kerberos realm names must be the same as the primary DNS domain names, with all letters uppercase. For example, if the domain names are `ad.example.com` for AD and `idm.example.com` for IdM, the Kerberos realm names are required to be `AD.EXAMPLE.COM` and `IDM.EXAMPLE.COM`.

DNS records resolvable from all DNS domains in the trust

All machines must be able to resolve DNS records from all DNS domains involved in the trust relationship.

IdM and AD DNS Domains

Systems joined to IdM can be distributed over multiple DNS domains. Red Hat recommends that you deploy IdM clients in a DNS zone different to the ones owned by Active Directory. The primary IdM DNS domain must have proper SRV records to support AD trusts.
NOTE

In some environments with trusts between IdM and Active Directory, you can install an IdM client on a host that is part of the Active Directory DNS domain. The host can then benefit from the Linux-focused features of IdM. This is not a recommended configuration and has some limitations. See Configuring IdM clients in an Active Directory DNS domain for more details.

You can acquire a list of the required SRV records specific to your system setup by running the following command:

```
$ ipa dns-update-system-records --dry-run
```

The generated list can look for example like this:

```
IPA DNS records:
    _kerberos-master._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
    _kerberos-master._udp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
    _kerberos._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
    _kerberos._tcp.idm.example.com. 86400 IN SRV 0 100 88 server.idm.example.com.
    _kerberos.idm.example.com. 86400 IN TXT "IDM.EXAMPLE.COM"
    _kpasswd._tcp.idm.example.com. 86400 IN SRV 0 100 464 server.idm.example.com.
    _kpasswd._udp.idm.example.com. 86400 IN SRV 0 100 464 server.idm.example.com.
    _ldap._tcp.idm.example.com. 86400 IN SRV 0 100 389 server.idm.example.com.
    _ipa-ca.idm.example.com. 86400 IN A 192.168.122.2
```

For other DNS domains that are part of the same IdM realm, it is not required for the SRV records to be configured when the trust to AD is configured. The reason is that AD domain controllers do not use SRV records to discover KDCs but rather base the KDC discovery on name suffix routing information for the trust.

7.2. CONFIGURING A DNS FORWARD ZONE IN THE IDM WEB UI

This section describes how to add a new DNS forward zone to the Identity Management (IdM) server using the IdM Web UI.

With DNS forward zones, you can forward DNS queries for a specific zone to a different DNS server. For example, you can forward DNS queries for the Active Directory (AD) domain to an AD DNS server.

Prerequisites

- Access to the IdM Web UI with a user account that has administrator rights.
- Correctly configured DNS server.

Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see Accessing the IdM Web UI in a web browser.
2. Click on the Network Services tab.
3. Click on the DNS tab.
4. In the drop down menu, click on the DNS Forward Zones item.

5. Click on the Add button.

6. In the Add DNS forward zone dialog box, add a zone name.

7. In the Zone forwarders item, click on the Add button.

8. In the Zone forwarders field, add the IP address of the server for which you want to create the new forward zone.

9. Click on the Add button.
The forwarded zone has been added to the DNS settings and you can verify it in the DNS Forward Zones settings. The Web UI informs you about success with the following pop-up message: **DNS Forward Zone successfully added.**

**NOTE**

The Web UI might display a warning about a DNSSEC validation failure after adding a new forward zone to the configuration.

DNSSEC (Domain Name System Security Extensions) secures DNS data with a digital signature to protect DNS from attacks. This service is enabled by default in the IdM server. The warning appears because the remote DNS server does not use DNSSEC. Red Hat recommends that you enable DNSSEC on the remote DNS server.

If you cannot enable DNSSEC validation on the remote server, you can disable DNSSEC in the IdM server:

1. Choose the appropriate configuration file to edit:
   - If your IdM server is using RHEL 8.0 or RHEL 8.1, open the `/etc/named.conf` file.
   - If your IdM server is using RHEL 8.2 or later, open the `/etc/named/ipa-options-ext.conf` file.

2. Add the following DNSSEC parameters:
   ```
   dnssec-enable no;
   dnssec-validation no;
   ```

3. Save and close the configuration file.

4. Restart the DNS service:
   ```
   # systemctl restart named-pkcs11
   ```

**Verification steps**

- Use the `nslookup` command with the name of the remote DNS server:
  ```
  $ nslookup ad.example.com
  Server:       192.168.122.2
  Address:      192.168.122.2#53
  ```
If the domain forwarding is configured correctly, the `nslookup` request displays an IP address of the remote DNS server.

7.3. CONFIGURING A DNS FORWARD ZONE IN THE CLI

This section describes how to add a new DNS forward zone to the Identity Management (IdM) server using the command line interface (CLI).

With DNS forward zones, you can forward DNS queries for a specific zone to a different DNS server. For example, you can forward DNS queries for the Active Directory (AD) domain to an AD DNS server.

**Prerequisites**
- Access to the CLI with a user account that has administrator rights.
- Correctly configured DNS server.

**Procedure**
- Create a DNS forward zone for the AD domain, and specify the IP address of the remote DNS server with the `--forwarder` option:
  ```
  # ipa dnsforwardzone-add ad.example.com --forwarder=192.168.122.3 --forward-policy=first
  ```
NOTE

You might see a warning about a DNSSEC validation failure in the /var/log/messages system logs after adding a new forward zone to the configuration:

```
named-pkcs11[2572]: no valid DS resolving 'host.ad.example.com/A/IN': 192.168.100.25#53
```

DNSSEC (Domain Name System Security Extensions) secures DNS data with a digital signature to protect DNS from attacks. This service is enabled by default in the IdM server. The warning appears because the remote DNS server does not use DNSSEC. Red Hat recommends that you enable DNSSEC on the remote DNS server.

If you cannot enable DNSSEC validation on the remote server, you can disable DNSSEC in the IdM server:

1. Choose the appropriate configuration file to edit:
   - If your IdM server is using RHEL 8.0 or RHEL 8.1, open the /etc/named.conf file.
   - If your IdM server is using RHEL 8.2 or later, open the /etc/named/ipa-options-ext.conf file.
2. Add the following DNSSEC parameters:
   ```
   dnssec-enable no;
   dnssec-validation no;
   ```
3. Save and close the configuration file.
4. Restart the DNS service:
   ```
   # systemctl restart named-pkcs11
   ```

Verification steps

- Use the `nslookup` command with the name of the remote DNS server:

```
$ nslookup ad.example.com
Server: 192.168.122.2
Address: 192.168.122.2#53

No-authoritative answer:
Name: ad.example.com
Address: 192.168.122.3
```

If the domain forwarding is configured correctly, the `nslookup` request displays an IP address of the remote DNS server.

7.4. CONFIGURING DNS FORWARDING IN AD

This section describes how to set up a DNS forwarding in Active Directory (AD) for the Identity Management (IdM) server.
Prerequisites

- Windows Server with AD installed.
- DNS port open on both servers.

Procedure

1. Log in to the Windows Server.
2. Open Server Manager.
3. Open DNS Manager.
4. In Conditional Forwarders, add a new conditional forwarder with:
   - The IdM server IP address
   - A fully qualified domain name, for example, server.idm.example.com
5. Save the settings.

7.5. VERIFYING THE DNS CONFIGURATION

Before configuring trust, verify that the Identity Management (IdM) and Active Directory (AD) servers can resolve themselves and each other.

Prerequisites

- You need to be logged in with sudo permissions.

Procedure

1. Run a DNS query for the Kerberos over UDP and LDAP over TCP service records.

   ```bash
   [admin@server ~]# dig +short -t SRV _kerberos._udp.idm.example.com.
   0 100 88 server.idm.example.com.

   [admin@server ~]# dig +short -t SRV _ldap._tcp.idm.example.com.
   0 100 389 server.idm.example.com.
   ```

   The commands are expected to list all IdM servers.

2. Run a DNS query for the TXT record with the IdM Kerberos realm name. The obtained value is expected to match the Kerberos realm you specified when installing IdM.

   ```bash
   [admin@server ~]# dig +short -t TXT _kerberos.idm.example.com.
   "IDM.EXAMPLE.COM"
   ```

   If the previous steps did not return all the expected records, update the DNS configuration with the missing records:

   - If your IdM environment uses an integrated DNS server, enter the `ipa dns-update-system-records` command without any options to update your system records:
If your IdM environment does not use an integrated DNS server:

1. On the IdM server, export the IdM DNS records into a file:

   [admin@server ~]$ ipa dns-update-system-records --dry-run --out dns_records_file.nsupdate

   The command creates a file named `dns_records_file.nsupdate` with the relevant IdM DNS records.

2. Submit a DNS update request to your DNS server using the `nsupdate` utility and the `dns_records_file.nsupdate` file. For more information, see Updating External DNS Records Using `nsupdate` in RHEL 7 documentation. Alternatively, refer to your DNS server documentation for adding DNS records.

3. Verify that IdM is able to resolve service records for AD with a command that runs a DNS query for Kerberos and LDAP over TCP service records:

   [admin@server ~]$ dig +short -t SRV _kerberos._tcp.dc._msdcs.ad.example.com.
   0 100 88 addc1.ad.example.com.

   [admin@server ~]$ dig +short -t SRV _ldap._tcp.dc._msdcs.ad.example.com.
   0 100 389 addc1.ad.example.com.
CHAPTER 8. CONFIGURING IDM CLIENTS IN AN ACTIVE DIRECTORY DNS DOMAIN

If you have client systems in a DNS domain controlled by Active Directory and you require those clients to be able to join the IdM Server to benefit from its RHEL features, you can configure users to access a client using a host name from the Active Directory DNS domain.

IMPORTANT

This is not a recommended configuration and has some limitations. Red Hat recommends to always deploy IdM clients in a DNS zone different from the ones owned by Active Directory and access IdM clients through their IdM host names.

Your IdM client configuration depends on whether you require single sign-on with Kerberos.

8.1. CONFIGURING AN IDM CLIENT WITHOUT KERBEROS SINGLE SIGN-ON

Password authentication is the only authentication method that is available for users to access resources on IdM clients if the IdM clients are in an Active Directory DNS domain. This procedure describes how to configure your client without Kerberos single sign-on.

Procedure

1. Install the IdM client with the --domain=IPA_DNS_Domain option to ensure the System Security Services Daemon (SSSD) can communicate with the IdM servers:

   [root@idm-client.ad.example.com ~]# ipa-client-install --domain=idm.example.com

   This option disables the SRV record auto-detection for the Active Directory DNS domain.

2. Open the /etc/krb5.conf configuration file and locate the existing mapping for the Active Directory domain in the [domain_realm] section.

   .ad.example.com = IDM.EXAMPLE.COM
   ad.example.com = IDM.EXAMPLE.COM

3. Replace both lines with an entry mapping the fully qualified domain name (FQDN) of the Linux clients in the Active Directory DNS zone to the IdM realm:

   idm-client.ad.example.com = IDM.EXAMPLE.COM

   By replacing the default mapping, you prevent Kerberos from sending its requests for the Active Directory domain to the IdM Kerberos Distribution Center (KDC). Instead Kerberos uses auto-discovery through SRV DNS records to locate the KDC.

8.2. REQUESTING SSL CERTIFICATES WITHOUT SINGLE SIGN-ON

SSL-based services require a certificate with dnsName extension records that cover all system host names, because both original (A/AAAA) and CNAME records must be in the certificate. Currently, IdM only issues certificates to host objects in the IdM database.
In the described setup without single sign-on available, IdM already has a host object for the FQDN in the database, and certmonger can request a certificate using this name.

Prerequisites

- Installed and configured the IdM client by following the procedure in Configuring an IdM client without Kerberos single sign-on.

Procedure

- Use certmonger to request a certificate using the FQDN:

```bash
[root@idm-client.ad.example.com ~]# ipa-getcert request -r \
-f /etc/httpd/alias/server.crt \
-k /etc/httpd/alias/server.key \
-N CN=ipa-client.ad.example.com \
-D ipa-client.ad.example.com \
-K host/idm-client.ad.example.com@IDM.EXAMPLE.COM \
-U id-kp-serverAuth
```

The certmonger service uses the default host key stored in the /etc/krb5.keytab file to authenticate to the IdM Certificate Authority (CA).

8.3. CONFIGURING AN IDM CLIENT WITH KERBEROS SINGLE SIGN-ON

If you require Kerberos single sign-on to access resources on the IdM client, the client must be within the IdM DNS domain, for example idm-client.idm.example.com. You must create a CNAME record idm-client.ad.example.com in the Active Directory DNS domain pointing to the A/AAAA record of the IdM client.

For Kerberos-based application servers, MIT Kerberos supports a method to allow the acceptance of any host-based principal available in the application’s keytab.

Procedure

- On the IdM client, disable the strict checks on what Kerberos principal is used to target the Kerberos server by setting the following option in the [libdefaults] section of the /etc/krb5.conf configuration file:

  ```
  ignore_acceptor_hostname = true
  ```

8.4. REQUESTING SSL CERTIFICATES WITH SINGLE SIGN-ON

SSL-based services require a certificate with dNSName extension records that cover all system host names, because both original (A/AAAA) and CNAME records must be in the certificate. Currently, IdM only issues certificates to host objects in the IdM database.

This procedure describes how to create a host object for ipa-client.example.com in IdM and make sure the real IdM machine’s host object is able to manage this host.

Prerequisites
• You have disabled the strict checks on what Kerberos principal is used to target the Kerberos server as outlined in Configuring an IdM client with Kerberos single sign-on.

Procedure

1. Create a new host object on the IdM server:

   ```
   [root@idm-server.idm.example.com ~]# ipa host-add idm-client.ad.example.com --force
   ```

   Use the `--force` option, because the host name is a CNAME and not an A/AAAA record.

2. On the IdM server, allow the IdM DNS host name to manage the Active Directory host entry in the IdM database:

   ```
   [root@idm-server.idm.example.com ~]# ipa host-add-managedby idm-client.ad.example.com
   --hosts=idm-client.idm.example.com
   ```

3. Your can now request an SSL certificate for your IdM client with the `dNSName` extension record for its host name within the Active Directory DNS domain:

   ```
   [root@idm-client.idm.example.com ~]# ipa-getcert request -r
   -f /etc/httpd/alias/server.crt
   -k /etc/httpd/alias/server.key
   -N CN=`hostname --fqdn`
   -D `hostname --fqdn`
   -D idm-client.ad.example.com
   -K host/idm-client.idm.example.com@IDM.EXAMPLE.COM
   -U id-kp-serverAuth
   ```
CHAPTER 9. SETTING UP A TRUST

This section describes how to configure the Identity Management (IdM)/Active Directory (AD) trust on the IdM side using the command line.

Prerequisites

- DNS is correctly configured. Both IdM and AD servers must be able to resolve each other names. For details, see Configuring DNS and realm settings for a trust.
- Supported versions of AD and IdM are deployed. For details, see Supported versions of Windows Server.
- You have obtained a Kerberos ticket. For details, see Using kinit to log in to IdM manually.

9.1. PREPARING THE IDM SERVER FOR THE TRUST

Before you can establish a trust with AD, you must prepare the IdM domain using the `ipa-adtrust-install` utility on an IdM server.

NOTE

Any system where you run the `ipa-adtrust-install` command automatically becomes an AD trust controller. However, you must run `ipa-adtrust-install` only once on an IdM server.

Prerequisites

- IdM server is installed.
- You need root privileges to install packages and restart IdM services.

Procedure

1. Install the required packages:

   ```
   [root@ipaserver ~]# dnf install ipa-server-trust-ad samba-client
   ```

2. Authenticate as the IdM administrative user:

   ```
   [root@ipaserver ~]# kinit admin
   ```

3. Run the `ipa-adtrust-install` utility:

   ```
   [root@ipaserver ~]# ipa-adtrust-install
   ```

   The DNS service records are created automatically if IdM was installed with an integrated DNS server.

   If you installed IdM without an integrated DNS server, `ipa-adtrust-install` prints a list of service records that you must manually add to DNS before you can continue.

4. The script prompts you that the `/etc/samba/smb.conf` already exists and will be rewritten:
WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing Samba configuration.

Do you wish to continue? [no]: yes

5. The script prompts you to configure the slapi-nis plug-in, a compatibility plug-in that allows older Linux clients to work with trusted users:

Do you want to enable support for trusted domains in Schema Compatibility plugin? This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.

Enable trusted domains support in slapi-nis? [no]: yes

6. When prompted, enter the NetBIOS name for the IdM domain or press Enter to accept the name suggested:

Trust is configured but no NetBIOS domain name found, setting it now.

Enter the NetBIOS name for the IPA domain.

Only up to 15 uppercase ASCII letters, digits and dashes are allowed.

Example: EXAMPLE.

NetBIOS domain name [IDM]:

7. You are prompted to run the SID generation task to create a SID for any existing users:

Do you want to run the ipa-sidgen task? [no]: yes

This is a resource-intensive task, so if you have a high number of users, you can run this at another time.

8. (Optional) By default, the Dynamic RPC port range is defined as 49152-65535 for Windows Server 2008 and later. If you need to define a different Dynamic RPC port range for your environment, configure Samba to use different ports and open those ports in your firewall settings. The following example sets the port range to 55000-65000.

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

9. Make sure that DNS is properly configured, as described in Verifying the DNS configuration for a trust.

IMPORTANT

Red Hat strongly recommends you verify the DNS configuration as described in Verifying the DNS configuration for a trust every time after running ipa-adtrust-install, especially if IdM or AD do not use integrated DNS servers.

10. Restart the ipa service:

```
[root@ipaserver ~]# ipactl restart
```
II. Use the `smbclient` utility to verify that Samba responds to Kerberos authentication from the IdM side:

```
[root@ipaserver ~]# smbclient -L server.idm.example.com -k
lp_load_ex: changing to config backend registry
Sharename       Type      Comment
---------       ----      -------
IPC$           IPC       IPC Service (Samba 4.12.3)
...
```

9.2. SETTING UP A TRUST AGREEMENT USING THE COMMAND LINE

This section describes how to set up the trust agreement using the command line. The Identity Management (IdM) server allows you to configure three types of trust agreements:

- **One-way trust** — default option. One-way trust enables Active Directory (AD) users and groups to access resources in IdM, but not the other way around. The IdM domain trusts the AD forest, but the AD forest does not trust the IdM domain.

- **Two-way trust** — Two-way trust enables AD users and groups to access resources in IdM. You must configure a two-way trust for solutions such as Microsoft SQL Server that expect the `S4U2Self` and `S4U2Proxy` Microsoft extensions to the Kerberos protocol to work over a trust boundary. An application on a RHEL IdM host might request `S4U2Self` or `S4U2Proxy` information from an Active Directory domain controller about an AD user, and a two-way trust provides this feature.

  Note that this two-way trust functionality does not allow IdM users to login to Windows systems, and the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD.

  - To create the two-way trust, add the following option to the command: `--two-way=true`

- **External trust** — a trust relationship between IdM and an AD domain in different forests. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, an external trust can be established from IdM to a domain within a forest. This is only recommended if it is not possible to establish a forest trust between forest root domains due to administrative or organizational reasons.

  - To create the external trust, add the following option to the command: `--external=true`

In this section, the steps below shows you how to create a one-way trust agreement.

**Prerequisites**

- User name and password of a Windows administrator.
- You have prepared the IdM server for the trust.

**Procedure**

- Create a trust agreement for the AD domain and the IdM domain by using the `ipa trust-add` command:
To have SSSD automatically generate UIDs and GIDs for AD users based on their SID, create a trust agreement with the the **Active Directory domain** ID range type. This is the most common configuration.

```
[root@server ~]# ipa trust-add --type=ad ad.example.com --admin <ad_admin_username> --password --range-type=ipa-ad-trust
```

If you have configured POSIX attributes for your users in Active Directory (such as `uidNumber` and `gidNumber`) and you want SSSD to process this information, create a trust agreement with the **Active Directory domain with POSIX attributes** ID range type:

```
[root@server ~]# ipa trust-add --type=ad ad.example.com --admin <ad_admin_username> --password --range-type=ipa-ad-trust-posix
```

**WARNING**

If you do not specify an ID Range type when creating a trust, IdM attempts to automatically select the appropriate range type by requesting details from AD domain controllers in the forest root domain. If IdM does not detect any POSIX attributes, the trust installation script selects the **Active Directory domain** ID range.

If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD. If POSIX attributes are not correctly set in AD, you will not be able to resolve AD users.

For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

### 9.3. SETTING UP A TRUST AGREEMENT IN THE IDM WEB UI

This section describes how to configure the Identity Management (IdM)/Active Directory (AD) trust agreement on the IdM side using the IdM Web UI.

**Prerequisites**

- DNS is correctly configured. Both IdM and AD servers must be able to resolve each other names.

- Supported versions of AD and IdM are deployed.

- You have obtained a Kerberos ticket.

- Before creating a trust in the Web UI, prepare the IdM server for the trust as described in:  
  Preparing the IdM server for the trust.
You need to be logged in as an IdM administrator.

**Procedure**

1. Log in to the IdM Web UI with administrator privileges. For details, see [Accessing the IdM Web UI in a web browser](#).

2. In the IdM Web UI, click the **IPA Server** tab.

3. In the **IPA Server** tab, click the **Trusts** tab.

4. In the drop down menu, select the **Trusts** option.

5. Click the **Add** button.

6. In the **Add Trust** dialog box, enter the name of the Active Directory domain.

7. In the **Account** and **Password** fields, add the administrator credentials of the Active Directory administrator.

8. *(Optional)* Select **Two-way trust**, if you want to enable AD users and groups to access resources in IdM. However, the two-way trust in IdM does not give the users any additional rights compared to the one-way trust solution in AD. Both solutions are considered equally secure because of default cross-forest trust SID filtering settings.

9. *(Optional)* Select **External trust** if you are configuring a trust with an AD domain that is not the root domain of an AD forest. While a forest trust always requires establishing a trust between IdM and the root domain of an Active Directory forest, you can establish an external trust from IdM to any domain within an AD forest.
10. *(Optional)* By default, the trust installation script tries to detect the appropriate ID range type. You can also explicitly set the ID range type by choosing one of the following options:

a. To have SSSD automatically generate UIDs and GIDs for AD users based on their SID, select the **Active Directory domain** ID range type. This is the most common configuration.

b. If you have configured POSIX attributes for your users in Active Directory (such as *uidNumber* and *gidNumber*) and you want SSSD to process this information, select the **Active Directory domain with POSIX attributes** ID range type.

![Range type options]

**WARNING**

If you leave the **Range type** setting on the default *Detect* option, IdM attempts to automatically select the appropriate range type by requesting details from AD domain controllers in the forest root domain. If IdM does not detect any POSIX attributes, the trust installation script selects the **Active Directory domain** ID range.

If IdM detects any POSIX attributes in the forest root domain, the trust installation script selects the **Active Directory domain with POSIX attributes** ID range and assumes that UIDs and GIDs are correctly defined in AD. If POSIX attributes are not correctly set in AD, you will not be able to resolve AD users.

For example, if the users and groups that need access to IdM systems are not part of the forest root domain, but instead are located in a child domain of the forest domain, the installation script may not detect the POSIX attributes defined in the child AD domain. In this case, Red Hat recommends that you explicitly choose the POSIX ID range type when establishing the trust.

11. Click **Add**.

**Verification steps**

- If the trust has been successfully added to the IdM server, you can see the green pop-up window in the IdM Web UI. It means that the:
  - Domain name exists
  - User name and password of the Windows Server has been added correctly.
9.4. VERIFYING THE KERBEROS CONFIGURATION

To verify the Kerberos configuration, test if it is possible to obtain a ticket for an Identity Management (IdM) user and if the IdM user can request service tickets.

Procedure

1. Request a ticket for an Active Directory (AD) user:
   
   ```bash
   [root@ipaserver ~]# kinit user@AD.EXAMPLE.COM
   ```

2. Request service tickets for a service within the IdM domain:

   ```bash
   [root@server ~]# kvno -S host server.idm.example.com
   ```

   If the AD service ticket is successfully granted, there is a cross-realm ticket-granting ticket (TGT) listed with all of the other requested tickets. The TGT is named krbtgt/IPA.DOMAIN@AD.DOMAIN.

   ```bash
   [root@server ]# klist
   Ticket cache: KEYRING:persistent:0:krb_cache_hRtox00
   Default principal: user@AD.EXAMPLE.COM
   
   Valid starting       Expires              Service principal
   03.05.2016 18:31:06  04.05.2016 04:31:01  host/server.idm.example.com@IDM.EXAMPLE.COM
   renew until 04.05.2016 18:31:00
   03.05.2016 18:31:06 04.05.2016 04:31:01 krbtgt/IDM.EXAMPLE.COM@AD.EXAMPLE.COM
   renew until 04.05.2016 18:31:00
   03.05.2016 18:31:01  04.05.2016 04:31:01  krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
   renew until 04.05.2016 18:31:00
   ```

The `localauth` plug-in maps Kerberos principals to local System Security Services Daemon (SSSD) user names. This allows AD users to use Kerberos authentication and access Linux services, which support GSSAPI authentication directly.

9.5. VERIFYING THE TRUST CONFIGURATION ON IDM

Before configuring trust, verify that the Identity Management (IdM) and Active Directory (AD) servers can resolve themselves and each other.

Prerequisites
You need to be logged in with administrator privileges.

### Procedure

1. Run a DNS query for the MS DC Kerberos over UDP and LDAP over TCP service records.

   ```
   [root@server ~]# dig +short -t SRV _kerberos._udp.dc._msdcs.idm.example.com.
   0 100 88 server.idm.example.com.
   [root@server ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.idm.example.com.
   0 100 389 server.idm.example.com.
   ```

   These commands list all IdM servers on which `ipa-adtrust-install` has been executed. The output is empty if `ipa-adtrust-install` has not been executed on any IdM server, which is typically before establishing the first trust relationship.

2. Run a DNS query for the Kerberos and LDAP over TCP service records to verify that IdM is able to resolve service records for AD:

   ```
   [root@server ~]# dig +short -t SRV _kerberos._tcp.dc._msdcs.ad.example.com.
   0 100 88 addc1.ad.example.com.
   [root@ipaserver ~]# dig +short -t SRV _ldap._tcp.dc._msdcs.ad.example.com.
   0 100 389 addc1.ad.example.com.
   ```

### 9.6. VERIFYING THE TRUST CONFIGURATION ON AD

After configuring the trust, verify that:

- The Identity Management (IdM)-hosted services are resolvable from the Active Directory (AD) server.

- AD services are resolvable from the AD server.

### Prerequisites

- You need to be logged in with administrator privileges.

### Procedure

1. On the AD server, set the `nslookup.exe` utility to look up service records.

   ```
   C:\> nslookup.exe
   > set type=SRV
   ```

2. Enter the domain name for the Kerberos over UDP and LDAP over TCP service records.

   ```
   > _kerberos._udp.idm.example.com.
   _kerberos._udp.idm.example.com. SRV service location:
   priority = 0
   weight = 100
   port = 88
   ```
svr hostname = server.idm.example.com
> _ldap._tcp.idm.example.com
_idap._tcp.idm.example.com  SRV service location:
  priority = 0
  weight = 100
  port = 389
  svr hostname = server.idm.example.com

3. Change the service type to TXT and run a DNS query for the TXT record with the IdM Kerberos realm name.

C:\>nslookup.exe
> set type=TXT
> _kerberos.idm.example.com.
  _kerberos.idm.example.com.  text = "IDM.EXAMPLE.COM"

4. Run a DNS query for the MS DC Kerberos over UDP and LDAP over TCP service records.

C:\>nslookup.exe
> set type=SRV
> _kerberos._udp.dc._msdcs.idm.example.com.
  _kerberos._udp.dc._msdcs.idm.example.com.  SRV service location:
    priority = 0
    weight = 100
    port = 88
    svr hostname = server.idm.example.com
> _ldap._tcp.dc._msdcs.idm.example.com.
  _ldap._tcp.dc._msdcs.idm.example.com.  SRV service location:
    priority = 0
    weight = 100
    port = 389
    svr hostname = server.idm.example.com

Active Directory only expects to discover domain controllers that can respond to AD-specific protocol requests, such as other AD domain controllers and IdM trust controllers. Use the ipa-adtrust-install tool to promote an IdM server to a trust controller, and you can verify which servers are trust controllers with the ipa server-role-find --role 'AD trust controller' command.

5. Verify that AD services are resolvable from the AD server.

C:\>nslookup.exe
> set type=SRV

6. Enter the domain name for the Kerberos over UDP and LDAP over TCP service records.

> _kerberos._udp.dc._msdcs.ad.example.com.
  _kerberos._udp.dc._msdcs.ad.example.com.  SRV service location:
    priority = 0
    weight = 100
    port = 88
    svr hostname = addc1.ad.example.com
> _ldap._tcp.dc._msdcs.ad.example.com.
  _ldap._tcp.dc._msdcs.ad.example.com.  SRV service location:
9.7. CREATING A TRUST AGENT

A trust agent is an IdM server that can perform identity lookups against AD domain controllers.

For example, if you are creating a replica of an IdM server that has a trust with Active Directory, you can set up the replica as a trust agent. A replica does not automatically have the AD trust agent role installed.

**Prerequisites**

- IdM is installed with an Active Directory trust.
- The `sssd-tools` package is installed.

**Procedure**

1. On an existing trust controller, run the `ipa-adtrust-install --add-agents` command:

   ```
   [root@existing_trust_controller]# ipa-adtrust-install --add-agents
   ```

   The command starts an interactive configuration session and prompts you for the information required to set up the agent.

2. Restart the IdM service on the trust agent.

   ```
   [root@new_trust_agent]# ipactl restart
   ```

3. Remove all entries from the SSSD cache on the trust agent:

   ```
   [root@new_trust_agent]# sssctl cache-remove
   ```

4. Verify that the replica has the AD trust agent role installed:

   ```
   [root@existing_trust_controller]# ipa server-show new_replica.idm.example.com
   ...
   Enabled server roles: CA server, NTP server, AD trust agent
   ```

**Additional resources**

- For further information about the `--add-agents` option, see the `ipa-adtrust-install(1)` man page.
- For more information on trust agents, see Trust controllers and trust agents in the Planning Identity Management guide.

9.8. ENABLING AUTOMATIC PRIVATE GROUP MAPPING FOR A POSIX ID RANGE ON THE CLI
By default, SSSD does not map private groups for Active Directory (AD) users if you have established a POSIX trust that relies on POSIX data stored in AD. If any AD users do not have primary groups configured, IdM is not be able to resolve them.

This procedure explains how to enable automatic private group mapping for an ID range by setting the **hybrid** option for the **auto_private_groups** SSSD parameter on the command line. As a result, IdM is able to resolve AD users that do not have primary groups configured in AD.

**Prerequisites**

- You have successfully established a POSIX cross-forest trust between your IdM and AD environments.

**Procedure**

1. Display all ID ranges and make note of the AD ID range you want to modify.

   ```bash
   [root@server ~]# ipa idrange-find
   2 ranges matched
   
   Range name: IDM.EXAMPLE.COM_id_range
   First Posix ID of the range: 882200000
   Number of IDs in the range: 200000
   Range type: local domain range
   
   Range name: AD.EXAMPLE.COM_id_range
   First Posix ID of the range: 1337000000
   Number of IDs in the range: 200000
   Domain SID of the trusted domain: S-1-5-21-4123312420-990666102-3578675309
   Range type: Active Directory trust range with POSIX attributes
   Number of entries returned 2
   
   Number of entries returned 2
   
   2. Adjust the automatic private group behavior for the AD ID range with the **ipa idrange-mod** command.

     ```bash
     [root@server ~]# ipa idrange-mod --auto-private-groups=hybrid AD.EXAMPLE.COM_id_range
     
     3. Reset the SSSD cache to enable the new setting.

     ```bash
     [root@server ~]# sss_cache -E
     
**Additional resources**

- [Options for automatically mapping private groups for AD users](#)

**9.9. Enabling automatic private group mapping for a P0sIxF Id range in the IDm WebUI**
By default, SSSD does not map private groups for Active Directory (AD) users if you have established a POSIX trust that relies on POSIX data stored in AD. If any AD users do not have primary groups configured, IdM is not be able to resolve them.

This procedure explains how to enable automatic private group mapping for an ID range by setting the **hybrid** option for the `auto_private_groups` SSSD parameter in the Identity Management (IdM) WebUI. As a result, IdM is able to resolve AD users that do not have primary groups configured in AD.

**Prerequisites**

- You have successfully established a POSIX cross-forest trust between your IdM and AD environments.

**Procedure**

1. Log into the IdM Web UI with your user name and password.
2. Open the **IPA Server → ID Ranges** tab.
3. Select the ID range you want to modify, such as `AD.EXAMPLE.COM_id_range`.
4. From the **Auto private groups** drop down menu, select the **hybrid** option.
5. Click the **Save** button to save your changes.
Additional resources

- Options for automatically mapping private groups for AD users
CHAPTER 10. REMOVING THE TRUST USING THE COMMAND LINE

This section describes how to remove the Identity Management (IdM)/Active Directory (AD) trust on the IdM side using the command line interface.

Prerequisites

- You have obtained a Kerberos ticket as an IdM administrator. For details, see Logging in to IdM in the Web UI: Using a Kerberos ticket.

Procedure

1. Use the `ipa trust-del` command to remove the trust configuration from IdM.

   ```bash
   [root@server ~]# ipa trust-del ad_domain_name
   -----------------------------
   Deleted trust "ad_domain_name"
   -----------------------------
   ```

2. Remove the trust object from your Active Directory configuration.

Verification steps

- Use the `ipa trust-show` command to confirm that the trust has been removed.

   ```bash
   [root@server ~]# ipa trust-show ad.example.com
   ipa: ERROR: ad.example.com: trust not found
   ```
CHAPTER 11. REMOVING THE TRUST USING THE IDM WEB UI

This section describes how to remove the Identity Management (IdM)/Active Directory (AD) trust using the IdM Web UI.

Prerequisites

- You have obtained a Kerberos ticket. For details, see Logging in to IdM in the Web UI: Using a Kerberos ticket.

Procedure

1. Log in to the IdM Web UI with administrator privileges. For details, see Accessing the IdM Web UI in a web browser.

2. In the IdM Web UI, click the IPA Server tab.

3. In the IPA Server tab, click the Trusts tab.

4. Select the trust you want to remove.

5. Click the Delete button.

6. In the Remove trusts dialog box, click Delete.

7. Remove the trust object from your Active Directory configuration.

Verification steps
If the trust has been successfully deleted, the Web UI displays a green pop-up with the text:

```
If the trust has been successfully deleted, the Web UI displays a green pop-up with the text:
```

<table>
<thead>
<tr>
<th>Role-Based Access Control</th>
<th>ID Ranges</th>
<th>Realm Domains</th>
<th>Trusts</th>
<th>Topology</th>
<th>API Browser</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>Policy</td>
<td>Authentication</td>
<td>Network Services</td>
<td>1 Item(s) Deleted</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Trusts**

<table>
<thead>
<tr>
<th>Search</th>
<th>Refresh</th>
<th>Delete</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>Realms</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No entries.