Red Hat Enterprise Linux 8

Planning Identity Management

Documentation for planning Identity Management and setting up access control

Last Updated: 2019-06-17

Red Hat Enterprise Linux 8 Planning Identity Management

Documentation for planning Identity Management and setting up access control

Legal Notice

Abstract

This document describes the planning of Identity Management services on Red Hat Enterprise Linux 8. The current version of the document contains only selected preview user stories.

Table of Contents

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, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and 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. OVERVIEW OF PLANNING FOR IDENTITY MANAGEMENT AND ACCESS CONTROL IN RHEL

The following sections provide an overview of the options for identity management and access control in Red Hat Enterprise Linux. After reading these sections, you will be able to approach the planning stage for your environment.

1.1. INTRODUCTION TO IDENTITY MANAGEMENT

This module explains the purpose of Identity Management in Red Hat Enterprise Linux. It also provides basic information about the Identity Management domain, including the client and server machines that are part of the domain.

The goal of Identity Management in Red Hat Enterprise Linux

Identity Management in Red Hat Enterprise Linux (IdM) provides a centralized and unified way to manage identity stores, authentication, policies, and authorization policies in a Linux-based domain. IdM significantly reduces the administrative overhead of managing different services individually and using different tools on different machines.

IdM is one of the few centralized identity, policy, and authorization software solutions that support:

- Advanced features of Linux operating system environments
- Unifying large groups of Linux machines
- Native integration with Active Directory

IdM creates a Linux-based and Linux-controlled domain:

- IdM builds on existing, native Linux tools and protocols. It has its own processes and configuration, but its underlying technologies are well-established on Linux systems and trusted by Linux administrators.
- IdM servers and clients are Red Hat Enterprise Linux machines. IdM clients can also be other Linux and UNIX distributions if they support standard protocols. Windows client cannot be a member of the IdM domain but user logged into Windows systems managed by Active Directory (AD) can connect to Linux clients or access services managed by IdM. This is accomplished by establishing cross forest trust between AD and IdM domains.

Managing identities and policies on multiple Linux servers

Without IdM: Each server is administered separately. All passwords are saved on the local machines. The IT administrator manages users on every machine, sets authentication and authorization policies separately, and maintains local passwords. However, more often the users rely on other centralized solution, for example direct integration with Active Directory (AD). Systems can be directly integrated with AD using several different solutions:

- Legacy Linux tools (not recommended to use)
- Solution based on Samba winbind (recommended for specific use cases)
- Solution based on a third-party software (usually require a license from another vendor)
- Solution based on SSSD (native Linux and recommended for the majority of use cases)

With IdM: The IT administrator can:

- Maintain the identities in one central place: the IdM server
- Apply policies uniformly to multiples of machines at the same time
- Set different access levels for users by using host-based access control, delegation, and other rules
- Centrally manage privilege escalation rules
- Define how home directories are mounted

Enterprise single sign-on

In case of Identity Management Enterprise, SSO (single sign-on) is implemented leveraging the Kerberos protocol. This protocol is popular in the infrastructure level and enables SSO with services such as SSH, LDAP, NFS, CUPS, or DNS. Web services using different web stacks (Apache, EAP, Django, and others) can also be enabled to use Kerberos for SSO. However, practice shows that using OpenID Connect or SAML based on SSO is more convenient for web applications. To bridge the two layers, it is recommended to deploy an Identity Provider solution (IdP) that would be able to convert Kerberos authentication into a OpenID Connect ticket or SAML assertion. Red Hat SSO technology based on the Keycloak open source project in an example of such IdP

Without IdM: Users log in to the system and are prompted for a password every single time they access a service or application. These passwords might be different, and the users have to remember which credential to use for which application.

With Idm: After users log in to the system, they can access multiple services and applications without being repeatedly asked for their credentials. This helps to:

- Improve usability
- Reduce the security risk of passwords being written down or stored insecurely
- Boost user productivity

Managing a mixed Linux and Windows environment

Without IdM: Windows systems are managed in an Active Directory forest, but development, production, and other teams have many Linux systems. The Linux systems are excluded from the Active Directory environment.

With IdM: The IT administrator can:

- Manage the Linux systems using native Linux tools
- Integrate the Linux systems into the environments centrally managed by Active Directory, thus preserving a centralized user store.
- Easily deploy new Linux systems at scale or as needed.
- Quickly react to business needs and make decisions related to management of the Linux infrastructure without dependency on other teams avoiding delays.

Contrasting Identity Management with a Standard LDAP Directory

A standard LDAP directory, such as Red Hat Directory Server, is a general-purpose directory: it can be customized to fit a broad range of use cases.

- Schema: a flexible schema that can be customized for a vast array of entries, such as users, machines, network entities, physical equipment, or buildings.
- Typically used as: a back-end directory to store data for other applications, such as business applications that provide services on the Internet.

Identity Management (IdM) has a specific purpose: managing internal, inside-the-enterprise identities as well as authentication and authorization policies that relate to these identities.

- Schema: a specific schema that defines a particular set of entries relevant to its purpose, such as entries for user or machine identities.
- Typically used as: the identity and authentication server to manage identities within the boundaries of an enterprise or a project.

The underlying directory server technology is the same for both Red Hat Directory Server and IdM. However, IdM is optimized to manage identities inside the enterprise. This limits its general extensibility, but also brings certain benefits: simpler configuration, better automation of resource management, and increased efficiency in managing enterprise identities.

Additional Resources

- Identity Management or Red Hat Directory Server – Which One Should I Use? on the Red Hat Enterprise Linux Blog.
- Knowledge Base article about Standard protocols.
- Red Hat Enterprise Linux 8 Beta Release Notes

1.2. INTRODUCTION TO IDENTITY MANAGEMENT SERVERS AND CLIENTS

The Identity Management domain includes the following types of systems:

Identity Management servers

Identity Management servers are Red Hat Enterprise Linux systems that work as domain controllers (DCs). In most deployments, an integrated certificate authority (CA) is also installed with the IdM server.
Servers are the central repositories for identity and policy information. They also host the services used by domain members.

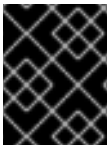
Identity Management clients

Identity Management clients are Red Hat Enterprise Linux systems enrolled with the servers and configured to use the Identity Management services on these servers.
Clients interact with the Identity Management servers to access services provided by them. For example, clients use the Kerberos protocol to perform authentication and acquire tickets for enterprise SSO, use LDAP to get identity and policy infromation, use DNS to detect where the servers and services are located and how to connect to them.

Identity Management servers are also embedded Identity Management clients. As clients enrolled with themselves, the servers provide the same functionality as other clients.

To provide services for large numbers of clients, as well as for redundancy and availability, Identity Management allows deployment on multiple IdM servers in a single domain. It is possible to deploy up to 60 servers. This is the maximum number of IdM servers, also called replicas, that is currently supported in the IdM domain. Identity Management servers provide different services for the client. Not all the servers need to provide all the possible services. Some server components like Kerberos and LDAP are always available on every server. Other services like Certificate authority (CA), DNS, Trust Controller or Vault are optional. This means that different servers in general play different roles in the deployment.

The first server installed to create the domain is the *master server*. If your Identity Management topology contains an integrated Certificate Authority (CA), this server is the *CRL generation master* and the *CA renewal master*: the only system in the domain responsible for tracking CA subsystem certificates and keys and for generating the certificate revocation list (CRL).



IMPORTANT

The CRL generation master role is critical because it is performed by only one server in the topology.

For redundancy and load balancing, administrators create additional servers by creating a *replica* of any existing server, either the master server or another replica. When creating a replica, Identity Management clones the configuration of the existing server. A replica shares with the initial server its core configuration, including internal information about users, systems, certificates, and configured policies.

NOTE

A replica and the server it was created from are functionally identical except for the role of the CRL generation master. Therefore, the term *server* and *replica* are used interchangeably here depending on the context.

1.3. IDENTITY MANAGEMENT AND ACCESS CONTROL IN RHEL: CENTRAL VS. LOCAL

In Red Hat Enterprise Linux, you can manage identities and access control policies using centralized tools for a whole domain of systems, or using local tools for a single system.

Managing identities and policies on multiple Red Hat Enterprise Linux servers: With and without Identity Management

With Identity Management, the IT administrator can:

- Maintain the identities and grouping mechanisms in one central place: the Identity Management server
- Centrally manage different types of credentials such as passwords, PKI certificates, OTP tokens, or SSH keys
- Apply policies uniformly to multiples of machines at the same time
- Manage POSIX and other attributes for external Active Directory users
- Set different access levels for users by using host-based access control, delegation, and other rules

- Centrally manage privilege escalation rules (sudo) and mandatory access control (SELinux user mapping)
- Maintain central PKI infrastructure and secrets store
- Define how home directories are mounted

Without Identity Management:

- Each server is administered separately.
- All passwords are saved on the local machines.
- The IT administrator manages users on every machine, sets authentication and authorization policies separately, and maintains local passwords.

1.4. ADDITIONAL RESOURCES

- For general information on Red Hat Identity Management, see the Red Hat Identity Management product page on the Red Hat Customer Portal.

CHAPTER 2. PLANNING THE REPLICA TOPOLOGY

The following sections provide advice on determining the appropriate replica topology for your use case.

2.1. MULTIPLE REPLICA SERVERS AS A SOLUTION FOR HIGH PERFORMANCE AND DISASTER RECOVERY

Continuous functionality and high availability of Identity Management services is vital for users who access resources. One of the built-in solutions for accomplishing continuous functionality and high availability of the Identity Management infrastructure through load balancing is the replication of the central directory by creating replica servers of the master server.

Identity Management allows placing additional servers in geographically dispersed data centers to reflect your enterprise organizational structure. In this way, the path between Identity Management clients and the nearest accessible server is shortened. In addition, having multiple servers allows spreading the load and scaling for more clients.

Maintaining multiple redundant Identity Management servers and letting them replicate with each other is also a common backup mechanism to mitigate or prevent server loss. For example, if one server fails, the other servers keep providing services to the domain. You can also recover the lost server by creating a new replica based on one of the remaining servers.

2.2. IDENTITY MANAGEMENT SERVERS AND CLIENTS

The Identity Management domain includes the following types of systems:

Identity Management servers

Identity Management servers are Red Hat Enterprise Linux systems that work as domain controllers (DCs). In most deployments, an integrated certificate authority (CA) is also installed with the IdM server.
Servers are the central repositories for identity and policy information. They also host the services used by domain members.

Identity Management servers are also embedded Identity Management clients. As clients enrolled with themselves, the servers provide the same functionality as other clients.

Identity Management clients

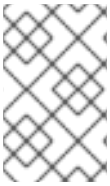
Identity Management clients are Red Hat Enterprise Linux systems enrolled with the servers and configured to use the Identity Management services on these servers.
Clients interact with the Identity Management servers to access domain resources. For example, clients belong to the Kerberos domain configured on the servers, receive certificates and tickets issued by the servers, and use other centralized services for authentication and authorization.

The first server installed to create the domain is the *master server*. If your Identity Management topology contains an integrated Certificate Authority (CA), this server is the *CRL generation master* and the *CA_renewal_master*: the only system in the domain responsible for tracking CA subsystem certificates and keys and for generating the certificate revocation list (CRL).

IMPORTANT

The CRL generation master role is critical because it is performed by only one server in the topology.

For redundancy and load balancing, administrators create additional servers by creating a *replica* of any existing server, either the master server or another replica. When creating a replica, Identity Management clones the configuration of the existing server. A replica shares with the initial server its core configuration, including internal information about users, systems, certificates, and configured policies.



NOTE

A replica and the server it was created from are functionally identical except for the role of the CRL generation master. Therefore, the term *server* and *replica* are used interchangeably here depending on the context.

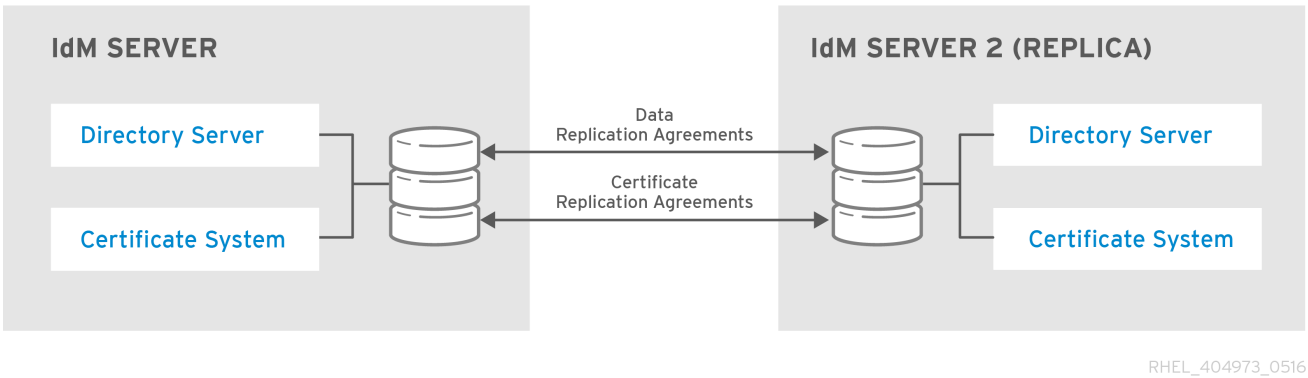
2.3. REPLICATION AGREEMENTS

When an administrator creates a replica based on an existing server, Identity Management creates a *replication agreement* between the initial server and the replica. The replication agreement ensures that the data and configuration is continuously replicated between the two servers.

Replication agreements are always bilateral: the data is replicated from one server to the other as well as from the other server to the first server.

Identity Management uses *multi-master replication*. In multi-master replication, all replicas joined in a replication agreement receive updates, and are therefore considered data masters.

Figure 2.1. Server and replica agreements



Identity Management uses two types of replication agreements:

Domain replication agreements

These agreements replicate the identity information.

Certificate replication agreements

These agreements replicate the certificate information.

Both replication channels are independent. Two servers can have one or both types of replication agreements configured between them. For example, when server A and server B have only domain replication agreement configured, only identity information is replicated between them, not the certificate information.

2.4. DETERMINING THE APPROPRIATE NUMBER OF REPLICAS

Set up at least two replicas in each data center (not a hard requirement)

A data center can be, for example, a main office or a geographical location.

Set up a sufficient number of servers to serve your clients

One Identity Management server can provide services to 2000 - 3000 clients. This assumes the clients query the servers multiple times a day, but not, for example, every minute. If you expect more frequent queries, plan for more servers.

Set up a maximum of 60 replicas in a single Identity Management domain

Red Hat guarantees to support environments with 60 replicas or fewer.

2.5. CONNECTING THE REPLICAS IN A TOPOLOGY

Connect each replica to at least two other replicas

Configuring additional replication agreements ensures that information is replicated not just between the initial replica and the master server, but between other replicas as well.

Connect a replica to a maximum of four other replicas (not a hard requirement)

A large number of replication agreements per server does not bring significant additional benefits. One consumer replica can only be updated by one other replica at a time. Meanwhile, the other replication agreements are idle.

Configuring too many replication agreements can also have a negative impact on overall performance.

Connect the replicas in a data center with each other

This ensures domain replication within the data center.

Connect each data center to at least two other data centers

This ensures domain replication between data centers.

Connect data centers using at least a pair of replication agreements

If data centers A and B have a replication agreement from A1 to B1, having a replication agreement from A2 to B2 ensures that if one of the servers is down, the replication can continue between the two data centers.

2.6. REPLICA TOPOLOGY EXAMPLES

The figures below show examples of Identity Management topologies based on the guidelines for creating a reliable topology.

Figure 2.2, “Replica Topology Example 1” shows four data centers, each with four servers. The servers are connected with replication agreements.

Figure 2.2. Replica Topology Example 1

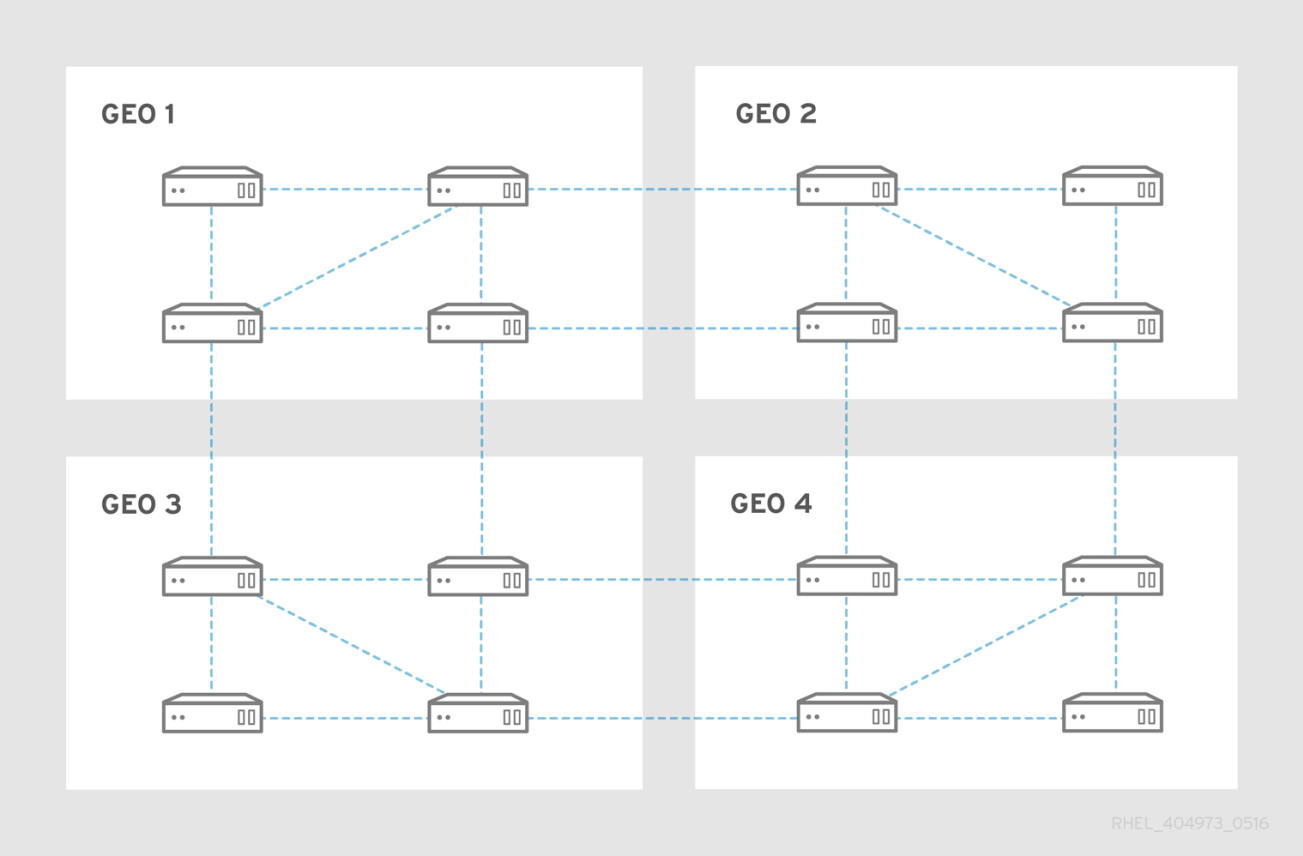
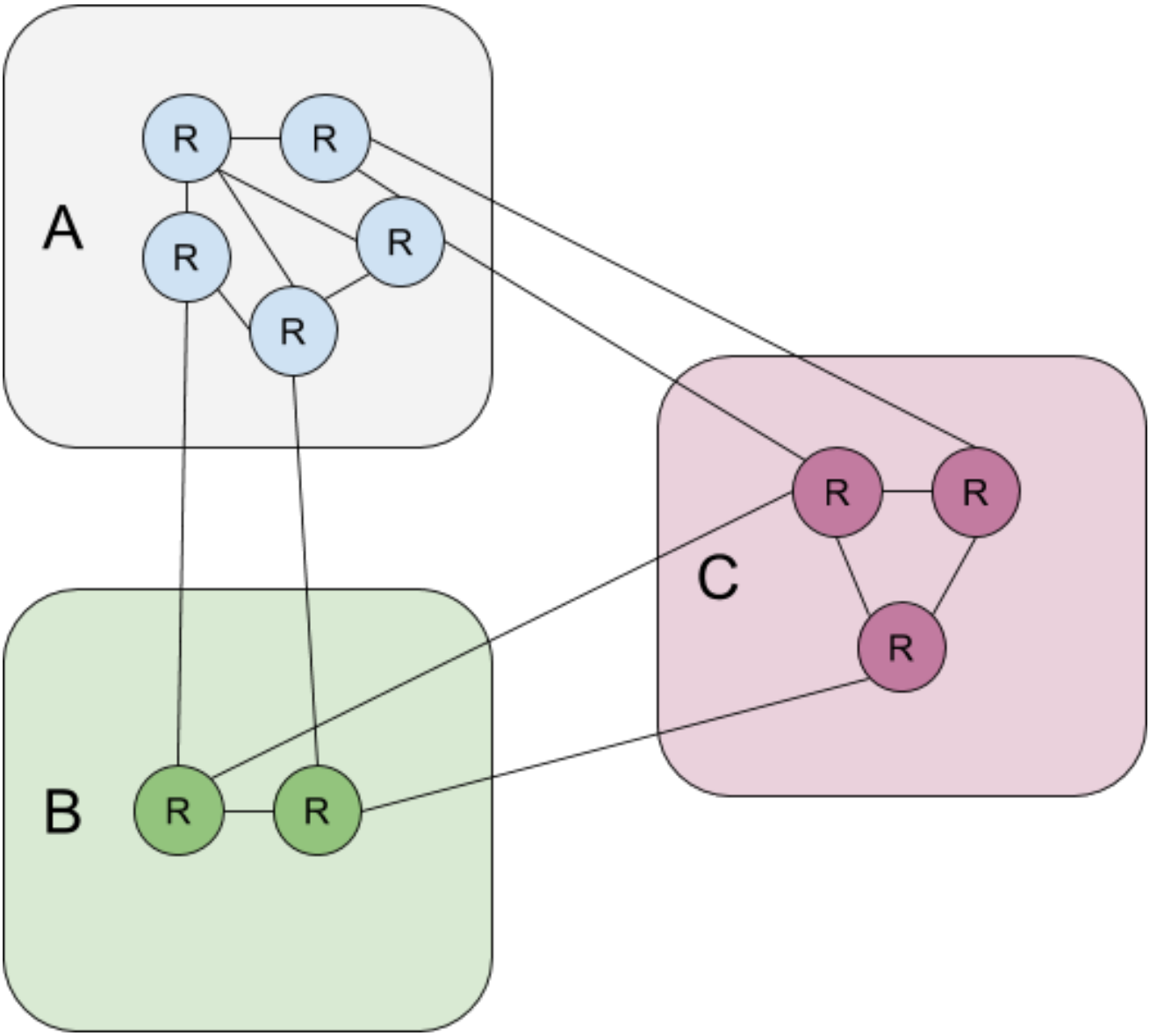


Figure 2.3, "Replica Topology Example 2" shows three data centers, each with a different number of servers. The servers are connected with replication agreements.

Figure 2.3. Replica Topology Example 2



CHAPTER 3. PLANNING YOUR DNS SERVICES AND HOST NAMES

Identity Management provides different types of DNS configurations in the Identity Management server. The following sections describe them and provide advice on how to determine which is the best for your use case.

3.1. DNS SERVICES AVAILABLE IN AN IDENTITY MANAGEMENT SERVER

You can install an Identity Management server with or without integrated DNS.

Table 3.1. Comparing Identity Management with integrated DNS and without integrated DNS

	With integrated DNS	Without integrated DNS
Overview:	Identity Management runs its own DNS service for the Identity Management domain.	Identity Management uses DNS services provided by an external DNS server.
Limitations:	The integrated DNS server provided by Identity Management only supports features related to Identity Management deployment and maintenance. It does not support some of the advanced DNS features. It is not designed to be used as a general-purpose DNS server.	DNS is not integrated with native Identity Management tools. For example, Identity Management does not update the DNS records automatically after a change in the topology.
Works best for:	Basic usage within the Identity Management deployment. When the Identity Management server manages DNS, DNS is tightly integrated with native Identity Management tools, which enables automating some of the DNS record management tasks.	Environments where advanced DNS features beyond the scope of the Identity Management DNS are needed. Environments with a well-established DNS infrastructure where you want to keep using an external DNS server.

Even if an Identity Management server is used as a master DNS server, other external DNS servers can still be used as slave servers. For example, if your environment is already using another DNS server, such as a DNS server integrated with Active Directory, you can delegate only the Identity Management primary domain to the DNS integrated with Identity Management. It is not necessary to migrate DNS zones to the Identity Management DNS.

3.2. GUIDELINES FOR PLANNING THE DNS DOMAIN NAME AND KERBEROS REALM NAME

When installing the first Identity Management server, the installation prompts for a primary DNS name of the Identity Management domain and Kerberos realm name. The guidelines in this section can help you set the names correctly.

WARNING

You will not be able to change the Identity Management primary domain name and Kerberos realm name after the server is already installed. Do not expect to be able to move from a testing environment to a production environment by changing the names, for example from *lab.example.com* to *production.example.com*.

A separate DNS domain for service records

Ensure that the *primary DNS domain* used for Identity Management is not shared with any other system. This helps avoid conflicts on the DNS level.

Proper DNS domain name delegation

Ensure you have valid delegation in the public DNS tree for the DNS domain. Do not use a domain name that is not delegated to you, not even on a private network.

A unique Kerberos realm name

Ensure the realm name is not in conflict with any other existing Kerberos realm name, such as a name used by Active Directory.

Kerberos realm name as an upper-case version of the primary DNS name

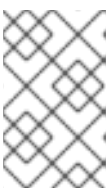
Consider setting the realm name to an upper-case (*EXAMPLE.COM*) version of the primary DNS domain name (*example.com*).

WARNING

If you do not set the Kerberos realm name to be the upper-case version of the primary DNS name, you will not be able to use Active Directory trusts.

Additional notes on planning the DNS domain name and Kerberos realm name

- One Identity Management deployment always represents one Kerberos realm.
- You can join Identity Management clients from multiple distinct DNS domains (*example.com*, *example.net*, *example.org*) to a single Kerberos realm (*EXAMPLE.COM*).
- Identity Management clients do not need to be in the primary DNS domain. For example, if the Identity Management domain is *idm.example.com*, the clients can be in the *clients.example.com* domain, but clear mapping must be configured between the DNS domain and the Kerberos realm.



NOTE

The standard method to create the mapping is using the **_kerberos** TXT DNS records. The Identity Management integrated DNS adds these records automatically.

CHAPTER 4. PLANNING YOUR CA SERVICES

Identity Management in Red Hat Enterprise Linux provides different types of certificate authority (CA) configurations. The following sections describe different scenarios and provide advice to help you determine which configuration is best for your use case.

4.1. CA SERVICES AVAILABLE IN AN IDENTITY MANAGEMENT SERVER

You can install an Identity Management server with an integrated Identity Management certificate authority (CA) or without a CA.

Table 4.1. Comparing Identity Management with integrated CA and without a CA

	Integrated CA	Without a CA
Overview:	Identity Management uses its own public key infrastructure (PKI) service with a *CA signing certificate* to create and sign the certificates in the Identity Management domain. • If the root CA is the integrated CA, Identity Management uses a self-signed CA certificate. • If the root CA is an external CA, the integrated Identity Management CA is subordinate to the external CA. The CA certificate used by Identity Management is signed by the external CA, but all certificates for the Identity Management domain are issued by the integrated Certificate System instance. • Integrated CA is also able to issue certificates for users, hosts, or services. The external CA can be a corporate CA or a third-party CA.	Identity Management does not set up its own CA, but uses signed host certificates from an external CA. Installing a server without a CA requires you to request the following certificates from a third-party authority: • An LDAP server certificate • An Apache server certificate • A PKINIT certificate • Full CA certificate chain of the CA that issued the LDAP and Apache server certificates

	Integrated CA	Without a CA
Limitations:	If the integrated CA is subordinate to an external CA, the certificates issued within the IdM domain are potentially subject to restrictions set by the external CA for various certificate attributes, such as: • The validity period. • Constraints on what subject names can appear on certificates issued by the IDM CA or its subordinates.. • Constraints on whether the IDM CA can itself, issue subordinate CA certificates, or how "deep" the chain of subordinate certificates can go.	Managing certificates outside of Identity Management causes a lot of additional activities, such as : • Creating, uploading, and renewing certificates is a manual process. • The **certmonger** service does not track the IPA certificates (LDAP server, Apache server, and PKINIT certificates) and does not notify you when the certificates are about to expire. The administrators must manually set up notifications for externally issued certificates, or set tracking requests for those certificates if they want **certmonger** to track them.
Works best for:	Environments that allow you to create and use your own certificate infrastructure.	Very rare cases when restrictions within the infrastructure do not allow you to install certificate services integrated with the server.

NOTE

Switching from the self-signed CA to an externally-signed CA, or the other way around, as well as changing which external CA issues the IdM CA certificate, is possible even after the installation. It is also possible to configure an integrated CA even after an installation without a CA.

4.2. CA SUBJECT DISTINGUISHED NAME

The CA subject distinguished name (DN) is the name of the CA. It must be globally unique in the Identity Management CA infrastructure and cannot be changed after the installation. In case you need the IDM CA to be externally signed, you might need to consult the administrator of the external CA about the form your IDM CA Subject DN should take.

4.3. GUIDELINES FOR DISTRIBUTION OF CA SERVICES

Following steps provide guidlines for the distribution of your CA services.

- Install the CA services on more than one server in the topology

Replicas configured without a CA forward all certificate operations requests to the CA servers in your topology.

WARNING

If you lose all servers with a CA, you will lose all the CA configuration without any chance of recovery. In such case you need to set up new CA and issue and install new certificates.

- Maintain a sufficient number of CA servers to handle the CA requests in your deployment

For recommendation see the following table:

Table 4.2. Guidelines for setting up appropriate number of CA servers

Description of the deployment	Suggested number of CA servers
A deployment with a very large number of certificates issued	Three or four CA servers
A deployment with bandwidth or availability problems between multiple regions	One CA server per region, with a minimum of three servers total for the deployment
All other deployments	Two CA servers

CHAPTER 5. PLANNING INTEGRATION WITH ACTIVE DIRECTORY

The following sections introduce the options for integrating Red Hat Enterprise Linux with Active Directory.

- For an overview of direct integration, see Section 5.1, “Direct integration”.
- For an overview of indirect integration, see Section 5.2, “Indirect integration”.
- For advice on how to decide between them, see Section 5.3, “Deciding between indirect and direct integration”.

5.1. DIRECT INTEGRATION

In direct integration, Linux systems are connected directly to Active Directory. The following types of integration are possible:

Integration with the System Security Services Daemon (SSSD)

SSSD can connect a Linux system with various identity and authentication stores: Active Directory, Identity Management, or a generic LDAP or Kerberos server.
Notable requirements for integration with SSSD:

- When integrating with Active Directory, SSSD works only within a single AD forest by default. For multi-forest setup, configure manual domain enumeration.
- Remote Active Directory forests must trust the local forest to ensure that the **idmap_ad** plug-in handles remote forest users correctly.

SSSD supports both direct and indirect integration. It also enables switching from one integration approach to the other without significant migration costs.

Integration with Samba Winbind

The Winbind component of the Samba suite emulates a Windows client on a Linux system and communicates with Active Directory servers.
Notable requirements for integration with Samba Winbind:

- Direct integration with Winbind in a multi-forest Active Directory setup requires bidirectional trusts.
- A bidirectional path from the local domain of a Linux system must exist to the domain of a user in a remote Active Directory forest to allow full information about the user from the remote Active Directory domain to be available to the **idmap_ad** plug-in.

Recommendations

- SSSD satisfies most of the use cases for AD integration and provides a robust solution as a generic gateway between a client system and different types of identity and authentication providers - AD, IdM, Kerberos, and LDAP.
- Winbind is recommended for deployment on those AD domain member servers on which you plan to deploy Samba FS.

5.2. INDIRECT INTEGRATION

In indirect integration, Linux systems are first connected to a central server which is then connected to Active Directory. Indirect integration enables the administrator to manage Linux systems and policies centrally, while users from Active Directory can transparently access Linux systems and services.

Integration based on cross-forest trust with Active Directory

The Identity Management server acts as the central server to control Linux systems. A cross-realm Kerberos trust with Active Directory is established, enabling users from Active Directory to log on to access Linux systems and resources. Identity Management presents itself to Active Directory as a separate forest and takes advantage of the forest-level trusts supported by Active Directory.
When using a trust:

- Active Directory users can access Identity Management resources.
- Identity Management servers and clients can resolve the identities of Active Directory users and groups.
- Active Directory users and groups access Identity Management under the conditions defined by Identity Management, such as host-based access control.
- Active Directory users and groups continue being managed on the Active Directory side.

Integration based on synchronization

This approach is based on the WinSync tool. A WinSync replication agreement synchronizes user accounts from Active Directory to Identity Management.

WARNING

WinSync is no longer actively developed in Red Hat Enterprise Linux 8. The preferred solution for indirect integration is cross-forest trust.

The limitations of integration based on synchronization include:

- Groups are not synchronized from Identity Management to Active Directory.
- Users are duplicated in Active Directory and Identity Management.
- WinSync supports only a single Active Directory domain.
- Only one domain controller in Active Directory can be used to synchronize data to one instance of Identity Management
- User passwords must be synchronized, which requires the PassSync component to be installed on all domain controllers in the Active Directory domain.
- After configuring the synchronization, all Active Directory users must manually change passwords before PassSync can synchronize them.

5.3. DECIDING BETWEEN INDIRECT AND DIRECT INTEGRATION

The guidelines in this section can help decide which type of integration fits your use case.

Number of systems to be connected to Active Directory

Connecting less than 30-50 systems (not a hard limit)

If you connect less than 30-50 systems, consider direct integration. Indirect integration might introduce unnecessary overhead.

Connecting more than 30-50 systems (not a hard limit)

If you connect more than 30-50 systems, consider indirect integration with Identity Management. With this approach, you can benefit from the centralized management for Linux systems.

Managing a small number of Linux systems, but expecting the number to grow rapidly

In this scenario, consider indirect integration to avoid having to migrate the environment later.

Frequency of deploying new systems and their type

Deploying bare metal systems on an irregular basis

If you deploy new systems rarely and they are usually bare metal systems, consider direct integration. In such cases, direct integration is usually simplest and easiest.

Deploying virtual systems frequently

If you deploy new systems often and they are usually virtual systems provisioned on demand, consider indirect integration. With indirect integration, you can use a central server to manage the new systems dynamically and integrate with orchestration tools, such as Red Hat Satellite.

Active Directory is the required authentication provider

Do your internal policies state that all users must authenticate against Active Directory?

You can choose either direct or indirect integration. If you use indirect integration with a trust between Identity Management and Active Directory, the users that access Linux systems authenticate against Active Directory. Policies that exist in Active Directory are executed and enforced during authentication.

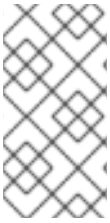
CHAPTER 6. PLANNING A CROSS-FOREST TRUST BETWEEN IDENTITY MANAGEMENT AND ACTIVE DIRECTORY

Active Directory and Identity Management are two alternative environments managing a variety of core services, such as Kerberos, LDAP, DNS, and certificate services. A *cross-forest trust* relationship transparently integrates these two diverse environments by enabling all core services to interact seamlessly. The following sections provide advice on how to plan and design a cross-forest trust deployment.

6.1. CROSS-FOREST TRUSTS BETWEEN IDENTITY MANAGEMENT AND ACTIVE DIRECTORY

In a pure Active Directory environment, a cross-forest trust connects two separate Active Directory forest root domains. When you create a cross-forest trust between Active Directory and Identity Management, the Identity Management domain presents itself to Active Directory as a separate forest with a single domain. A trust relationship is then established between the Active Directory forest root domain and the Identity Management domain. As a result, users from the Active Directory forest can access the resources in the Identity Management domain.

Identity Management can establish a trust with one Active Directory forest or multiple unrelated forests.



NOTE

Two separate Kerberos realms can be connected in a *cross-realm trust*. However, a Kerberos realm only concerns authentication, not other services and protocols involved in identity and authorization operations. Therefore, establishing a Kerberos cross-realm trust is not enough to enable users from one realm to access resources in another realm.

An external trust to an Active Directory domain
An external trust is a trust relationship between Identity Management and an Active Directory domain. While a forest trust always requires establishing a trust between Identity Management and the root domain of an Active Directory forest, an external trust can be established from Identity Management to any domain within a forest.

6.2. TRUST CONTROLLERS AND TRUST AGENTS

Identity Management provides the following types of Identity Management servers that support trust to Active Directory:

Trust agents

Identity Management servers that can perform identity lookups against Active Directory domain controllers.

Trust controllers

Trust agents that also run the Samba suite. Active Directory domain controllers contact trust controllers when establishing and verifying the trust to Active Directory.
The first trust controller is created when you configure the trust.

Trust controllers run more network-facing services than trust agents, and thus present a greater attack surface for potential intruders.

In addition to trust agents and controllers, the Identity Management domain can also include standard

Identity Management servers. However, these servers do not communicate with Active Directory. Therefore, clients that communicate with the standard servers cannot resolve Active Directory users and groups or authenticate and authorize Active Directory users.

Table 6.1. Comparing the capabilities supported by trust controllers and trust agents

Capability	Trust agent	Trust controller
Resolve Active Directory users and groups	Yes	Yes
Enroll Identity Management clients that run services accessible by users from trusted Active Directory forests	Yes	Yes
Manage the trust (for example, add trust agreements)	No	Yes

When planning the deployment of trust controllers and trust agents, consider these guidelines:

- Configure at least two trust controllers per Identity Management deployment.
- Configure at least two trust controllers in each data center.

If you ever want to create additional trust controllers or if an existing trust controller fails, create a new trust controller by promoting a trust agent or a standard server. To do this, use the **ipa-adtrust-install** utility on the Identity Management server.

IMPORTANT

You cannot downgrade an existing trust controller to a trust agent.

6.3. ONE-WAY TRUSTS AND TWO-WAY TRUSTS

In one way trusts, Identity Management (IdM) trusts Active Directory (AD) but AD does not trust IdM. AD users can access resources in the IdM domain but users from IdM cannot access resources within the AD domain. The IdM server connects to AD using a special account, and reads identity information that is then delivered to IdM clients over LDAP.

In two way trusts, IdM users can authenticate to AD, and AD users can authenticate to IdM. AD users can authenticate to and access resources in the IdM domain as in the one way trust case. IdM users can authenticate but cannot access most of the resources in AD. They can only access those Kerberized services in AD forests that do not require any access control check.

To be able to grant access to the AD resources, IdM needs to implement the Global Catalog service. This service does not yet exist in the current version of the IdM server. Because of that, a two-way trust between IdM and AD is nearly functionally equivalent to a one-way trust between IdM and AD.

6.4. NON-POSIX EXTERNAL GROUPS AND SECURITY ID MAPPING

Identity Management uses LDAP for managing groups. Active Directory entries are not synchronized or copied over to Identity Management, which means that Active Directory users and groups have no LDAP objects in the LDAP server, so they cannot be directly used to express group membership in the Identity Management LDAP. For this reason, administrators in Identity Management need to create non-POSIX external groups, referenced as normal Identity Management LDAP objects to signify group membership for Active Directory users and groups in Identity Management.

Security IDs (SIDs) for non-POSIX external groups are processed by SSSD, which maps the SIDs of groups in Active Directory to POSIX groups in Identity Management. In Active Directory, SIDs are associated with user names. When an Active Directory user name is used to access Identity Management resources, SSSD uses the user's SID to build up a full group membership information for the user in the Identity Management domain.

6.5. SETTING UP DNS

These guidelines can help you achieve the right DNS configuration for establishing a cross-forest trust between Identity Management and Active Directory.

Unique primary DNS domains

Ensure both Active Directory and Identity Management have their own unique primary DNS domains configured. For example:

- **ad.example.com** for Active Directory and **idm.example.com** for Identity Management
- **example.com** for Active Directory and **idm.example.com** for Identity Management

The most convenient management solution is an environment where each DNS domain is managed by integrated DNS servers, but you can also use any other standard-compliant DNS server.

No overlap between Identity Management and Active Directory DNS Domains

Systems joined to Identity Management can be distributed over multiple DNS domains. Ensure the DNS domains that contain Identity Management clients do not overlap with DNS domains that contain systems joined to Active Directory.

Proper SRV records

Ensure the primary Identity Management DNS domain has proper SRV records to support Active Directory trusts.
For other DNS domains that are part of the same Identity Management realm, the SRV records do not have to be configured when the trust to Active Directory is established. The reason is that Active Directory domain controllers do not use SRV records to discover Kerberos key distribution centers (KDCs) but rather base the KDC discovery on name suffix routing information for the trust.

DNS records resolvable from all DNS domains in the trust

Ensure all machines can resolve DNS records from all DNS domains involved in the trust relationship:

- When configuring the Identity Management DNS, follow the instructions described in Installing an IdM server with an external CA .
- If you are using Identity Management without integrated DNS, follow the instructions described in Installing an IdM server without integrated DNS .

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

Ensure Kerberos realm names are the same as the primary DNS domain names, with all letters uppercase. For example, if the domain names are **ad.example.com** for Active Directory and **idm.example.com** for Identity Management, the Kerberos realm names must be **AD.EXAMPLE.COM** and **IDM.EXAMPLE.COM**.

6.6. NETBIOS NAMES

The NetBIOS name is usually the far-left component of the domain name. For example:

- In the domain name **linux.example.com**, the NetBIOS name is **linux**.
- In the domain name **example.com**, the NetBIOS name is **example**.

Different NetBIOS names for the Identity Management and Active Directory domains

Ensure the Identity Management and Active Directory domains have different NetBIOS names.
The NetBIOS name is critical for identifying the Active Directory domain. If the Identity Management domain is within a subdomain of the Active Directory DNS, the NetBIOS name is also critical for identifying the Identity Management domain and services.

Character limit for NetBIOS names

The maximum length of a NetBIOS name is 15 characters.

6.7. CONFIGURING ACTIVE DIRECTORY SERVER DISCOVERY AND AFFINITY

Server discovery and affinity configuration affects which Active Directory servers an Identity Management client communicates with. This section provides an overview of how discovery and affinity work in an environment with a cross-forest trust between Identity Management and Active Directory.

Configuring clients to prefer servers in the same geographical location helps prevent time lags and other problems that occur when clients contact servers from another, remote datacenter. To make sure clients communicate with local servers, you must ensure that:

- Clients communicate with local Identity Management servers over LDAP and over Kerberos
- Clients communicate with local Active Directory servers over Kerberos
- Embedded clients on Identity Management servers communicate with local Active Directory servers over LDAP and over Kerberos

Options for configuring LDAP and Kerberos on the Identity Management client for communication with local Identity Management servers

When using Identity Management with integrated DNS

By default, clients use automatic service lookup based on the DNS records. In this setup, you can also use the *DNS locations* feature to configure DNS-based service discovery.
To override the automatic lookup, you can disable the DNS discovery in one of the following ways:

- During the Identity Management client installation by providing failover parameters from the command line
- After the client installation by modifying the System Security Services Daemon configuration

When using Identity Management without integrated DNS

You must explicitly configure clients in one of the following ways:

- During the Identity Management client installation by providing failover parameters from the command line
- After the client installation by modifying the System Security Services Daemon configuration

Options for configuring Kerberos on the Identity Management client for communication with local Active Directory servers

Identity Management clients are unable to automatically discover which Active Directory servers to communicate with. To specify the Active Directory servers manually, modify the **krb5.conf** file:

- Add the Active Directory realm information
- Explicitly list the Active Directory servers to communicate with

For example:

```
[realms]
AD.EXAMPLE.COM = {
kdc = server1.ad.example.com
kdc = server2.ad.example.com
}
```

Options for configuring embedded clients on Identity Management servers for communication with local Active Directory servers over Kerberos and LDAP

The embedded client on an Identity Management server works also as a client of the Active Directory server. It can automatically discover and use the appropriate Active Directory site.

When the embedded client performs the discovery, it might first discover an Active Directory server in a remote location. If the attempt to contact the remote server takes too long, the client might stop the operation without establishing the connection. Use the **dns_resolver_timeout** option in the **sssd.conf** file on the client to increase the amount of time for which the client waits for a reply from the DNS resolver. See the *sssd.conf(5)* man page for details.

Once the embedded client has been configured to communicate with the local Active Directory servers, the System Security Services Daemon (SSSD) remembers the Active Directory site the embedded client belongs to. Thanks to this, SSSD normally sends an LDAP ping directly to a local domain controller to refresh its site information. If the site no longer exists or the client has meanwhile been assigned to a different site, SSSD starts querying for SRV records in the forest and goes through a whole process of autodiscovery.

Using *trusted domain sections* in **sssd.conf**, you can also expicitly override some of the information that is discovered automatically by default.

6.8. OPERATIONS PERFORMED DURING INDIRECT INTEGRATION OF IDENTITY MANAGEMENT TO ACTIVE DIRECTORY

Table 6.2, "Operations performed from an Identity Management trust controller towards Active Directory domain controllers" shows which operations and requests are performed during the creation of an Identity Management to Active Directory trust from the Identity Management trust controller towards Active Directory domain controllers.

Table 6.2. Operations performed from an Identity Management trust controller towards Active Directory domain controllers

Operation	Protocol used	Purpose
DNS resolution against the Active Directory DNS resolvers configured on an Identity Management trust controller	DNS	To discover the IP addresses of Active Directory domain controllers
Requests to UDP/UDP6 port 389 on an Active Directory DC	Connectionless LDAP (CLDAP)	To perform Active Directory DC discovery
Requests to TCP/TCP6 ports 389 and 3268 on an Active Directory DC	LDAP	To query Active Directory user and group information
Requests to TCP/TCP6 ports 389 and 3268 on an Active Directory DC	DCE RPC and SMB	To set up and support cross-forest trust to Active Directory
Requests to TCP/TCP6 ports 135, 139, 445 on an Active Directory DC	DCE RPC and SMB	To set up and support cross-forest trust to Active Directory
Requests to dynamically opened ports on an Active Directory DC as directed by the Active Directory domain controller, likely in the range of 49152-65535 (TCP/TCP6)	DCE RPC and SMB	To respond to requests by DCE RPC End-point mapper (port 135 TCP/TCP6)
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Active Directory DC	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; administer Kerberos remotely

Table 6.3, "Operations performed from an Active Directory domain controller towards Identity Management trust controllers" shows which operations and requests are performed during the creation of an Identity Management to Active Directory trust from the Active Directory domain controller towards Identity Management trust controllers.

Table 6.3. Operations performed from an Active Directory domain controller towards Identity Management trust controllers

Operation	Protocol used	Purpose
DNS resolution against the Identity Management DNS resolvers configured on an Active Directory domain controller	DNS	To discover the IP addresses of Identity Management trust controllers
Requests to UDP/UDP6 port 389 on an Identity Management trust controller	Connectionless LDAP (CLDAP)	To perform Identity Management trust controller discovery

Operation	Protocol used	Purpose
Requests to TCP/TCP6 ports 135, 139, 445 on an Identity Management trust controller	DCE RPC and SMB	To verify the cross-forest trust to Active Directory
Requests to dynamically opened ports on an Identity Management trust controller as directed by the Identity Management trust controller, likely in the range of 49152-65535 (TCP/TCP6)	DCE RPC and SMB	To respond to requests by DCE RPC End-point mapper (port 135 TCP/TCP6)
Requests to ports 88 (TCP/TCP6 and UDP/UDP6), 464 (TCP/TCP6 and UDP/UDP6), and 749 (TCP/TCP6) on an Identity Management trust controller	Kerberos	To obtain a Kerberos ticket; change a Kerberos password; administer Kerberos remotely