Red Hat Reference Architecture Series

Integrating Red Hat Enterprise Linux 6 with Active Directory

Mark Heslin

Principal Software Engineer

Version 1.1

April 2012

redhat.

1801 Varsity Drive™
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Send feedback to refarch-feedback@redhat.com

Table of Contents

1 Executive Summary

In many organizations, system administrators encounter the need to integrate Linux systems into their existing Microsoft Windows Active Directory domain environments. There is a vast array of published material available. How does one begin to sort through this material to better understand and determine the best solution to deploy for their specific environment?

On the surface, the world of Linux and Windows interoperability appears deceptively simple. However, after closer examination, initial optimism gives way to the realization that there is an overwhelming number of components, configurations and integration options available. The intent of this reference architecture is to provide guidelines to simplify and assist in the selection, deployment and integration process.

This paper details the components, considerations and configurations available for selecting, deploying and integrating Red Hat Enterprise Linux 6 into Windows Active Directory domains. Basic concepts are introduced, deployment and integration tasks outlined, best practices and guidelines provided throughout.

To facilitate the selection process, a decision tree has been provided to guide the reader towards one of four recommended configurations. All deployment prerequisites must be completed before proceeding with the integration tasks.

Red Hat Enterprise Linux is a high-performing operating system that has delivered outstanding value to IT environments for nearly a decade. As the world's most trusted IT platform, Red Hat Enterprise Linux has been deployed in mission-critical applications at global stock exchanges, financial institutions, leading telcos, and animation studios. It also powers the websites of some of the most recognizable global retail brands.

Red Hat Enterprise Linux 6 offers unmatched reliability, performance, security, simplified management capabilities and costs savings. The included interoperability features are based on industry-proven standards and capabilities. For organizations looking to integrate Linux systems into Windows Active Directory domains, Red Hat Enterprise Linux 6 remains the platform of choice.

This document does not require extensive Red Hat Enterprise Linux experience but the reader is expected to have a working knowledge of Windows 2008 Server administration concepts. As a convenience, a glossary is provided in **Appendix B: Glossary** and can be consulted for unfamiliar terms or concepts.

2 Component Overview

This section provides detailed descriptions on the various components. A solid understanding of each component and it's relevance is essential to deploying a successful integration project. Depending on which of the actual configurations is selected, some components may or may not be implemented.

2.1 Red Hat Enterprise Linux 6

Red Hat Enterprise Linux 6, the latest release of Red Hat's trusted datacenter platform, delivers advances in application performance, scalability, and security. With Red Hat Enterprise Linux 6, physical, virtual and cloud computing resources can be deployed within the data center. Red Hat Enterprise Linux 6.2 provides the following features and capabilities:

Reliability, Availability, and Security (RAS):

- More sockets, more cores, more threads, and more memory
- RAS hardware-based hot add of CPUs and memory is enabled
- Memory pages with errors can be declared as “poisoned” and can be avoided

File Systems:

- ext4 is the default file system and scales to 16TB
- XFS is available as an add-on and can scale to 100TB
- Fuse allows file systems to run in user space allowing testing and development on newer fuse-based file systems (such as cloud file systems)

High Availability:

- Extends the current clustering solution to the virtual environment allowing for high availability of virtual machines and applications running inside those virtual machines
- Enables NFSv4 resource agent monitoring
- Introduction of Cluster Configuration System (CCS). CCS is a command line tool that allows for complete CLI administration of Red Hat's High Availability Add-On

Resource Management:

- cgroups organize system tasks so that they can be tracked and other system services can control the resources that cgroup tasks may consume
- cpuset applies CPU resource limits to cgroups, allowing processing performance to be allocated to tasks

There are many other feature enhancements to Red Hat Enterprise Linux 6. Please see the Red Hat website for more information.

2.2 Windows Server 2008 R2

Windows Server 2008 R2 is Microsoft's enterprise operating system for businesses and provides features for virtualization, power savings, manageability and mobile access.

Windows Server 2008 R2 is available in several editions – Foundation, Standard, Enterprise, Datacenter, Web and HPC (High Performance Computing). Windows Server 2008 R2 Enterprise Edition is used for the configurations described in this reference architecture.

2.3 Active Directory Domain Services (AD DS)

Active Directory Domain Services is a suite of directory services developed by Microsoft. Active Directory utilizes customized versions of industry standard protocols including:

- Kerberos
- Domain Name System (DNS)
- Lightweight Directory Access Protocol (LDAP)

Active Directory allows Windows system administrators to securely manage directory objects from a scalable, centralized database infrastructure. Directory objects (users, systems, groups, printers, applications) are stored in a hierarchy consisting of nodes, trees, forests and domains.

Prior to Windows Server 2008 R2, Active Directory Domain Services was known as Active Directory. Active Directory Domain Services is included with Windows Server 2008 R2.

2.4 Identity Management (IdM) in Red Hat Enterprise Linux (RHEL)

Red Hat Identity Management (IdM) in RHEL is a domain controller for Linux and UNIX servers that uses native Linux tools. Similar to Active Directory, Identity Management provides centralized management of identity stores, authentication and authorization policies. Identity Management defines a domain, with servers and clients who share centrally-managed services, like Kerberos and DNS. Although centralized applications to manage identity, policy and authorization are not new, Identity Management is one of the only options that supports Linux/Unix domains.

Identity Management provides a unifying interface for standards-based, common network services, including PAM, LDAP, Kerberos, DNS, NTP, and certificate services, and allows Red Hat Enterprise Linux systems to serve as domain controllers.

Currently, Red Hat Identity Management in RHEL does not provide support for full Active Directory domain trusts, therefore it's use is considered out of scope for the configurations detailed within this document. For further information on Identity Management please consult the references found in **Appendix A: References**.

2.5 Samba

Samba is an open source suite of programs that can be installed on Red Hat Enterprise Linux 6 systems to provide file and print services to Microsoft Windows clients.

Samba provides two daemons that run on a Red Hat Enterprise Linux 6 system:

- ***smbd*** (primary daemon providing file and print services to clients via SMB)
- ***nmbd*** (NetBIOS name server - not required for integration purposes)

When combined with the reliability and simplified management capabilities of Red Hat Enterprise Linux 6, Samba is the application of choice for providing file and print sharing to Windows clients. Samba version 3.5 is used in the Samba based configurations detailed within this reference architecture.

2.6 SMB/CIFS

Both Server Message Block (**SMB**) and Common Internet File System (**CIFS**) are network protocols developed to facilitate client to server communications for file and print services. The SMB protocol was originally developed by IBM and later extended by Microsoft as the CIFS protocol.

Samba supports both the SMB and CIFS protocols with SMB provided for client connections to older, legacy Windows servers (Windows 2000 or earlier). The terms SMB and CIFS are often interchanged but from a functional perspective, both are protocols used by Samba.

2.7 Winbind

Winbind is a component of the Samba suite of programs that allows for unified user logon. ***winbind*** uses an implementation of Microsoft RPC (Remote Procedure Calls), PAM (Pluggable Authentication Modules), and Red Hat Enterprise Linux 6 *nsswitch* (Name Service Switch) to allow Windows Active Directory Domain Services users to appear and operate as local users on a Red Hat Enterprise Linux machine. *Winbind* minimizes the need for system administrators to manage separate user accounts on both the Red Hat Enterprise Linux 6 and Windows Server 2008 R2 environments. *winbind* provides three separate functions:

- Authentication of user credentials (via PAM). This makes it possible to log onto a Red Hat Enterprise Linux 6 system using Active Directory user accounts. Authentication is responsible for identifying “Who” a user claims to be.
- ID Tracking/Name Resolution via nsswitch (NSS). The nsswitch service allows user and system information to be obtained from different database services such as LDAP or NIS. ID Tracking/Name Resolution is responsible for determining “Where” user identities are found.
- ID Mapping represents the mapping between Red Hat Enterprise Linux 6 user (UID), group (GID), and Windows Server 2008 R2 security (SID) IDs. ID Mappings are handled through an idmap “backend” that is responsible for tracking “What” ID's users are known by in both operating system environments.

Figure 2.6: Winbind Authentication, ID Components and Backends represents the relationship between Winbind and Active Directory:

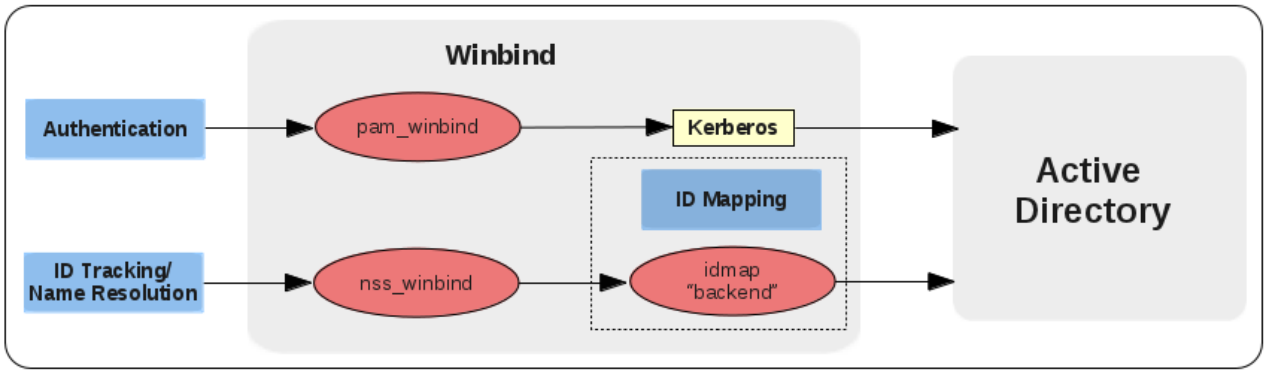

Figure 2.6: Winbind Authentication, ID Components and Backends

Winbind idmap “backends” are one of the most commonly misunderstood components in Samba. Since Winbind provides a number of different “backends” and each manages ID Mappings differently, it is useful to classify them as follows:

- *Allocating* - “Read-Writeable” backends that store ID Mappings in a local database file on the Red Hat Enterprise Linux 6 system(s).
- *Algorithmic* - “Read-Only” backends that are calculate ID Mappings on demand and provide consistent ID Mappings across each Red Hat Enterprise Linux 6 system.
- *Assigned* - “Read-Only” backends that use ID Mappings pre-configured within Active Directory.

Selecting a Winbind “backend” is also dependent on factors such as:

- Whether or not Active Directory schema modifications are permitted
- Preferred location of ID Mappings
- Number of Red Hat Enterprise Linux 6 systems
- Number of nodes in the Active Directory forest
- Use of LDAP

Understanding Winbind backends is essential when selecting a Samba based configuration. **Section 3.2.6 Winbind Backends** provides a comparative overview of each of the Winbind backends currently available under Red Hat Enterprise Linux 6.

2.8 Kerberos

Developed at the Massachusetts Institute of Technology (MIT), **Kerberos** is a network authentication protocol that uses symmetric key cryptography to provide highly secure authentication between client and server applications. Both Red Hat Enterprise Linux 6 and Windows server 2008 R2 are based on the current release of Kerberos - version 5. The configurations described within this paper are based on Kerberos version 5.

Kerberos operates on the basis of “tickets” that are granted by a trusted third-party called a key distribution center (KDC). The KDC maintains a secure database of secret keys that are known only to the KDC itself and the client requesting a ticket. Tickets have a configurable expiration date and must be refreshed by the client on a regular basis.

Kerberos authentication is significantly safer than normal password-based authentication because passwords are never sent over the network - even when services are accessed on other machines.

2.9 Lightweight Directory Access Protocol (LDAP)

The **Lightweight Directory Access Protocol (LDAP)** is a set of open protocols used to access centrally stored information over a network. It is based on the *X.500* standard for directory sharing, but is less complex and resource-intensive. For this reason, LDAP is sometimes referred to as "*X.500 Lite*." The X.500 standard is a directory that contains hierarchical and categorized information, which could include information such as names, addresses, and phone numbers.

Like X.500, LDAP organizes information in a hierarchy based on the use of directories. These directories can store a variety of information and can even be used in a manner similar to the Network Information Service (NIS), enabling anyone to access their account from any machine on the LDAP enabled network.

In many cases, LDAP is used as a virtual phone directory, allowing users to easily access contact information for other users. However, LDAP is much more flexible and capable of referring a query to other LDAP servers throughout the world, providing an ad-hoc global repository of information. Currently, LDAP is more commonly used within individual organizations, like universities, government departments, and private companies.

LDAP is a client/server system. The server can use a variety of databases to store a directory, each optimized for quick and copious read operations. When an LDAP client application connects to an LDAP server, it can either query a directory or attempt to modify it. In the event of a query, the server either answers the query locally, or it can refer the query to an LDAP server which does have the answer. If the client application is attempting to modify information within an LDAP directory, the server verifies that the user has permission to make the change and then adds or updates the information.

The main benefit of using LDAP is that information for an entire organization can be consolidated into a central repository. For example, rather than managing user lists for each group within an organization, LDAP can be used as a central directory accessible from anywhere on the network. Since LDAP supports Secure Sockets Layer (SSL) and Transport Layer Security (TLS), sensitive data can be protected from prying eyes.

2.10 System Security Services Daemon (SSSD)

The **System Security Services Daemon (SSSD)** provides access to different identity and authentication providers. SSSD is an intermediary between local clients and any configured data store. The local clients connect to SSSD and then SSSD contacts the external providers. This brings a number of benefits for administrators:

- *Offline authentication.* SSSD can optionally keep a cache of user identities and credentials that it retrieves from remote authentication/identification services. This allows users to authenticate to resources successfully, even if the remote identification server is offline or the local machine is offline.
- *Reduced load on authentication/identification servers*. Rather than having every client contact the identification server directly, all local clients can contact SSSD which can connect to the identification server or check its cache.
- *Single user account*. Remote users frequently have multiple user accounts, such as one for their local system and one for the organizational system. Since SSSD supports caching and offline authentication, remote users can connect to network resources simply by authenticating to their local machine and then SSSD maintains their network credentials.

SSSD recognizes domains, which are associated with different identity servers. Domains are a combination of an identity provider and an authentication method. SSSD works with LDAP identity providers (OpenLDAP, Red Hat Directory Server, IdM in RHEL, Microsoft Active Directory) and native LDAP authentication or Kerberos authentication.

2.11 Domain Name System (DNS)

Domain Name System (DNS) is a hierarchical, distributed naming system for managing the mappings between human-friendly domain, host and service names to IP addresses. DNS also defines the protocol for DNS communication exchanges as part of the Internet Protocol (IP) suite. On Red Hat Enterprise Linux 6, DNS is configured in the file */etc/resolv.conf*.

2.12 Network Time Protocol (NTP)

Network Time Protocol (NTP) is an Internet protocol used to synchronize computer system clocks to a reference time source. On Red Hat Enterprise Linux 6, the *ntpd* daemon handles synchronization. NTP parameters are configured in the file */etc/ntp.conf*.

2.13 Name Service Switch (NSS)

Name Service Switch (NSS) service allows user and system information (*passwd*, *shadow*, *group*, *hosts*, etc.) to be obtained from different database services such as DNS, LDAP, NIS or local files. On Red Hat Enterprise Linux 6, NSS parameters are configured in the file */etc/nsswitch.conf*.

3 Considerations

There are many reasons why organizations choose to integrate Red Hat Enterprise Linux 6 systems into a Windows Active Directory domain. Some of the most common include:

- Simplify, consolidate the administration of user accounts
- Greater reliability, stability
- Cost savings
- Flexibility
- Customization
- Source code access
- Greater security
- Leverage Red Hat Enterprise Linux 6 benefits

The sections that follow describe the most common technical and non-technical areas for consideration when deploying and integrating Red Hat Enterprise Linux 6 into Windows Active Directory environments.

3.1 Non-technical Considerations

3.1.1 Organizational Alignment

In many organizations, IT roles and responsibilities are separated across different groups for geographical, political or functional purposes. This separation often results in the vertical segmenting of duties. One organization, group or team may be exclusively assigned to managing the Windows Active Directory domain resources while another is responsible for managing the Red Hat Enterprise Linux environment.

When deploying a cross-functional project of any type, it is important not to underestimate the potential impact that may result from segmentation. Resistance can be encountered or misconceptions seen regarding various aspects of the components, technologies, security or project itself. It is important to address these issues by acknowledging and scoping the extent of them early in order to properly engage, communicate and align the project with the stakeholders and technical staff.

3.1.2 Expertise Levels

Depending on the existing skill sets and familiarity of the technical staff managing the project integration tasks, it may be necessary to provide training in areas where skills sets are not as strong. Targeting these areas early into the project and close to the prototyping phase is preferable. In some organizations, cross-functional training of teams may work best.

3.1.3 Scope/Complexity

The scope of a project to integrate five Red Hat Enterprise Linux systems into an Active Directory domain is entirely different than a project to integrate hundred of systems. What about complexity? Are the systems being integrated geographically disperse, managed by different organizations? Are there technical complexities to factor in as well? Good project planning acknowledges and factors in both the scope and the depth of the integration project.

3.1.4 Prototype

Regardless of the scope and complexity of any integration project, the implementation details should be thoroughly tested and verified in advance of the project deployment. Prototyping is an essential step that demonstrates not only the feasibility of the project, but also develops expertise and confidence across the project and its stakeholders.

3.1.5 Project Deployment

For most integration projects, a phased deployment works best since the Red Hat Enterprise Linux systems are being integrated into an existing Active Directory domain environment. In some situations (e.g. - a new office location or recently formed team) a new deployment may be a viable option. Regardless of the deployment method used, be certain to factor in each of the previously discussed considerations before deploying the project.

3.2 Technical Considerations

3.2.1 File Sharing

Depending on the requirements of the integration project, file sharing may or may not be a core requirement. For projects that need file sharing capabilities, a Samba/Winbind based configuration is required.

Since Samba supports both client side and server side file sharing, it is important to identify the location and access methods to the target file share(s). A Red Hat Enterprise Linux 6 system running Samba can function as both a server that manages local file shares to other clients or it can also act as a client that maps file shares from other servers such as Windows 2008 Server R2 or Linux based servers running the Samba client. For some environments a combination of both functions may be necessary.

Regardless of location and access methods, all of the configurations detailed within this reference architecture authenticate file sharing through the use of Active Directory user credentials.

3.2.2 Login Access

For most integration projects, interactive login access to the Red Hat Enterprise Linux 6 systems using Active Directory user credentials is a core requirement. All configurations detailed within this reference architecture support interactive login access through both the command line interface (CLI) and the graphical display manager (GDM) on the Red Hat Enterprise Linux 6 systems.

3.2.3 Active Directory ID Attributes

From an organizational perspective, it is important to identify whether or not Active Directory schema enhancements are permitted. As a matter of policy, some environments do not permit updates to existing Active Directory information. From an integration perspective this is important as organizational policies may restrict the selection of available configurations.

The schema extensions outlined in RFC 2307 define a method for exchanging UNIX ID attributes using LDAP. These extensions are required to allow Red Hat Enterprise Linux 6 systems to integrate with Active Directory domains using LDAP.

Prior to Windows Server 2003 R2, schema extensions are configured by enabling Windows Services for UNIX (SFU). In Windows Server 2003 R2 and Windows Server 2008, the schema extensions are enabled as part of the Identity Management for UNIX (IMU) service. Under Windows Server 2008 R2, Identity Management for UNIX (IMU) is enabled as a server role.

3.2.4 Enumeration

Enumeration refers to the process of listing the users, groups in an Active Directory domain. When a user logs in and authenticates into a domain with a large number of users, Active Directory enumerates the entire list of users before completing the login process. The larger the number of users in the domain, the longer the login can take. For environments with 20,000 or more users it can be advantageous to disable enumeration. Under Samba, this is done by changing the ***enum*** directives in the */etc/samba/smb.conf* file:

```
[global]
   winbind enum users = no
   winbind enum groups = no
```

3.2.5 LDAP Referrals

LDAP referrals occur when a requested object can not be found in the responding domain controller's portion of the Active Directory domain or forest. Clients are then notified of other domain controllers that are more likely to contain the directory tree where the object is stored.

Similar to enumeration, when LDAP referrals occur, a performance penalty is incurred. In larger environments, it may be advantageous to disable LDAP referrals. Under SSSD, this is done by changing the ***ldap_disable_referrals*** directive in the */etc/sssd/sssd.conf* file:

```
ldap_disable_referrals = true
```

Disabling LDAP referrals will realize significant performance improvements. However in environments that employ partial replication, access to users and groups that are not replicated to the local AD server may be unavailable to the RHEL client.

3.2.6 Winbind Backends

When selecting a Samba based configuration, it is essential to understand the function and capabilities of the Winbind backend being used. **Table 3.2.6: Comparison of Winbind Backends** below provides a comparative overview of each of the Windbind backends currently available under Red Hat Enterprise Linux 6:

Backend	Type	ID Mappings	Advantages	Disadvantages
idmap_tdb	Read/Write	Allocating	• Simplest to implement • Default winbind backend	• Limited scalability – not intended for consistent ID mappings across multiple RHEL servers • Cache corruption requires manual intervention to correct file ownership • Static - 1 tdb entry for each SID (slower)
idmap_rid	Read-only	Algorithmic	• Uses algorithmic ID mappings across multiple servers (faster)	• Requires additional configuration work to support a forest of AD domains or multiple domain trees
idmap_ad	Read-only	Assigned (by admin)	• Standardized user configuration (shell, home directory) • Centralized user account management • Compatible with SFU and RFC2307 mappings	• Requires additional configuration work to support a forest of AD domains or multiple domain trees • Requires additional user management tasks – user/group ID attributes must be specified within AD
idmap_ldap	Read/Write	Allocating	• ID mappings stored in centralized, non-AD server (RHDS, OpenLDAP, etc.)	• Requires external LDAP server • Most complex configuration to implement due to Samba LDAP mapping limitations (UID/GID not stored at POSIX level)

Table 3.2.6: Comparison of Winbind Backends

Backend	Type	ID Mappings	Advantages	Disadvantages
idmap_adex	Read-only	Assigned (by admin)	• Supports ID mappings using RFC2307 attributes	• Not recommended for new deployments (deprecated by latest versions of Samba) • Requires Identity Management for UNIX (IMU)
idmap_hash	Read-only	Algorithmic	• Similar to idmap_rid but generates UID/GID from full domain SID • Mappings consistent across RHEL systems	• No additional configuration but potential risk of ID collisions
idmap_tdb2	Read/Write	Allocating	• Script option available for performing ID mappings via an external program	• For Samba clusters (CTDB) only
idmap_nss	Read-only	Pre-existing	• Uses existing UID/GID mappings	• No support for trusted domains • Can't resolve mappings unless SID is available

Table 3.2.6: Comparison of Winbind Backends (continued)

Further details on each of the Winbind backends can be found in **Appendix C: Winbind Backend Reference**.

3.2.7 Services Integration

Improperly resolved hostnames (DNS) and Kerberos authentication failures due to time synchronization (NTP) delays are among the most common causes for integration failures. In environments where DNS or NTP time services are not reliable, best practice is to configure the Red Hat Enterprise Linux 6 systems to perform DNS lookups and NTP time synchronizations from the Windows Server 2008 R2 Active Directory server. Further details on configuring these services is provided in **Sections 5.6 Synchronize Time Services** and **5.7 Configure DNS**.

3.2.8 Log Files

Log files are essential for monitoring application activity and troubleshooting. **Table 3.2.8 Component Log Files** provides a summary of the default log file locations on Red Hat Enterprise Linux 6 systems:

Component	Log File(s)	Description
Red Hat Enterprise Linux 6	/var/log/messages	System events
Red Hat Enterprise Linux 6	/var/log/dmesg	Boot messages
Red Hat Enterprise Linux 6	/var/log/secure	Security, authentication messages
Samba	/var/log/samba/log.smbd	SMB daemon (**smbd**)
Samba	/var/log/samba/log.winbindd	Winbind daemon (**winbindd**)
Samba	/var/log/samba/log.wb-{client}	{Client} specific connections
Kerberos	Standard out (terminal screen)	Kerberos client messages
Kerberos	/var/log/krb5libs.log	Kerberos library messages (*optional*)
Kerberos	/var/log/krb5kdc.log	Kerberos KDC messages (*optional*)
LDAP	/var/log/messages	LDAP daemon (**nslcd**)
SSSD	/var/log/sssd/sssd.log	SSSD daemon (**sssd**)
SSSD	/var/log/sssd/sssd_pam.log	PAM events
DNS	/var/log/messages	DNS messages
NTP	/var/log/messages	NTP messages
NSS	/var/log/messages	NSS caching daemon (**nscd**)

Table 3.2.8: Component Log Files

By default, most events are sent via **syslog** to the default */var/log/messages*. Most daemons have debug mode capabilities that can be enabled through configuration files (*e.g. - /etc/krb5.conf*) or via command line flags (*e.g. - nslcd -d*). Consult the on-line man pages and Red Hat documentation for further details.

4 Configurations

The previous sections provided detailed descriptions of the various components and the relevance of each for deploying a successful integration project. The most common technical and non-technical areas for consideration were also discussed. In this section, the focus shifts to comparing and selecting from the available configurations.

4.1 Overview

The range of options for integrating Red Hat Enterprise Linux 6 systems into an Active Directory domain environment is extensive and each has its advantages and disadvantages. In order to select the configuration that best matches the needs of a given environment, a common set of comparative criteria are needed to answer fundamental questions such as:

- What use case(s) are being addressed?
- What services are provided?
- What are the advantages?
- What are the disadvantages?

In the sections that follow, each of the available configurations are compared at a high level and then narrowed down to a smaller set of recommended configurations through the use of a decision tree.

4.2 Configuration Feature Comparisons

Table 4.2: Configuration Feature Comparisons below provides a concise, comparative overview of each of the configurations available for integrating Red Hat Enterprise Linux 6 systems into an Active Directory domain:

Use Case	Services Provided	Advantages	Disadvantages
Configuration 1 – Samba/Windbind "winbind - idmap_tdb"			
• Single RHEL server integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• Simplest to implement • Default winbind backend	• Limited scalability – not intended for consistent ID mappings across multiple RHEL servers • Cache corruption requires manual intervention to correct file ownership • Static - 1 tdb entry for each SID (slower)
Configuration 2 – Samba/Windbind "winbind - idmap_rid"			
• Multiple RHEL servers integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• Uses algorithmic ID mappings across multiple servers (faster) • Mappings consistent across RHEL systems	• Requires additional configuration work to support a forest of AD domains or multiple domain trees
Configuration 3 – Samba/Windbind "winbind - idmap_ad"			
• Multiple RHEL servers integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• Standardized user configuration (shell, home directory) • Centralized user account management • SFU, RFC2307 compatible mappings • Mappings consistent across RHEL systems	• Requires additional configuration work to support a forest of AD domains or multiple domain trees • Requires additional user management tasks – user/group ID attributes must be set within AD

Table 4.2: Configuration Feature Comparisons

Use Case	Services Provided	Advantages	Disadvantages
Configuration 4 – Samba/Windbind “winbind - idmap_ldap”			
• Multiple RHEL servers integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• ID mappings stored in centralized, non-AD server (RHDS, OpenLDAP, etc.) • Mappings consistent across RHEL systems	• Requires external LDAP server • Most complex configuration to implement due to Samba LDAP mapping limitations (UID/GID not stored at POSIX level)
Configuration 5 – Samba/Winbind “winbind - idmap_adex”			
• Multiple RHEL servers integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• Supports ID mappings using RFC2307 attributes • Mappings consistent across RHEL systems	• Not recommended for new deployments (deprecated by latest versions of Samba) • Requires Identity Management for UNIX (IMU)
Configuration 6 – Samba/Winbind “winbind - idmap_hash”			
• Multiple RHEL servers integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• Similar to idmap_rid but generates UID/GID from full domain SID • Mappings consistent across RHEL systems	• No additional configuration but potential risk of ID collisions
Configuration 7 – Samba/Winbind “winbind - idmap_tdb2”			
• Samba clustered RHEL servers integrated to an AD domain or forest	• File sharing • Login access (RHEL CLI, GUI)	• Script option available for performing ID mappings via an external program	• For Samba clusters (CTDB) only • Scalability - 4-node cluster limitation • Not manageable via cluster management

Table 4.2: Configuration Feature Comparisons (continued)

Use Case	Services Provided	Advantages	Disadvantages
Configuration 8 – Samba/Winbind "winbind - idmap_nss"			
• Multiple RHEL servers integrated to an AD domain or forest • Single RHEL server integrated to an AD domain or forest (local files – passwd, group)	• File sharing • Login access (RHEL CLI, GUI)	• Uses existing UID/GID mappings • Mappings consistent across RHEL systems	• No support for trusted domains • Can't resolve mappings unless SID is available
Configuration 9 – SSSD/Kerberos/LDAP			
• Multiple RHEL servers integrated to an AD domain or forest	• Login access (RHEL CLI, GUI)	• Kerberos SSO capable • Supports SASL/GSSAPI binds of LDAP lookups • Enforces encrypted authentication only • Client side caching • Off-line caching • Reduces client queries to server • Graceful management of ID collisions	• File sharing requires additional components (Samba/Winbind)
Configuration 10 – Kerberos/LDAP Native			
• Multiple RHEL servers integrated to an AD domain or forest	• Login access (RHEL CLI, GUI)	• Kerberos SSO capable • Client side caching (via nscd)	• File sharing requires additional components (Samba/Winbind) • No off-line caching of user credentials • Poor management of ID collisions

Table 4.2: Configuration Feature Comparisons (continued)

4.3 Selecting a Configuration

As demonstrated by the previous sections, there are many criteria to consider when selecting a configuration that best meets the integration the needs for a specific environment. **Figure 4.3.1: Decision Tree - Recommended Configurations** below is provided to simplify and assist in the configuration selection process:

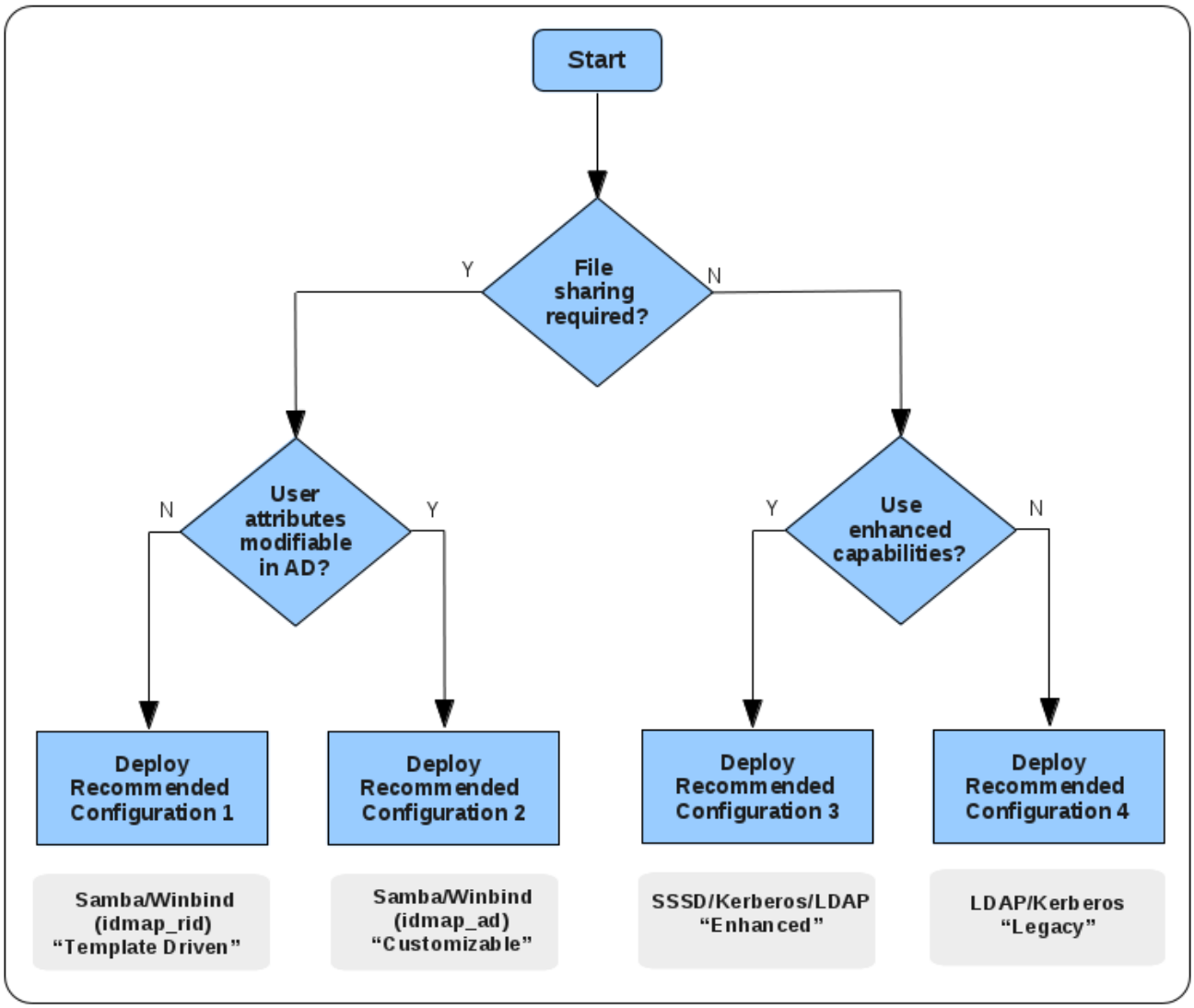

Figure 4.3.1: Decision Tree - Recommended Configurations

The recommended configurations shown here are considered the most practical to deploy and represent those that best match the integration requirements of the majority of Red Hat customer environments. Integration details for each of the four recommended configurations is discussed in **Section 6 Recommended Configurations**.

5 Deployment Prerequisites

In this section, the deployment prerequisites for integrating Red Hat Enterprise Linux 6 systems into an Active Directory domain are provided. Prior to proceeding with the integration tasks described in **Section 6 Recommended Configurations**, the following deployment prerequisites must first be completed:

Windows Server 2008 R2 server:

- Deploy Windows Server 2008 R2
- Configure Active Directory Domain Services

Red Hat Enterprise Linux 6 systems:

- Deploy Red Hat Enterprise Linux 6
- Configure SELinux Security Parameters
- Install/Configure Samba (*for recommended configurations 1, 2 only*)
- Synchronize Time Services
- Configure DNS
- Install/Configure Kerberos Client

Do not proceed with the integration tasks until each of these components have been fully configured and deployed. The sections that follow provide a guide to each of the deployment prerequisites.

5.1 Deploy Windows 2008 Server R2

The following Microsoft TechNet article contains the most current and comprehensive details on installing and deploying Windows Server 2008 R2:

http://technet.microsoft.com/en-us/library/dd283085.aspx[5]

For this reference architecture, an HP ProLiant BL460c G6 blade server was deployed as the Windows 2008 Server R2 server. This server contains 4, Quad core processors (16 cores), 48 GB memory and two mirrored (RAID 1), internal 146 GB SATA drives.

5.2 Configure Active Directory Domain Services

If Active Directory Domain Services is already deployed and configured, proceed to **Section 5.3 Deploy Red Hat Enterprise Linux 6** below. If Active Directory Domain Services is not deployed, please consult the following Microsoft TechNet article for the most current and comprehensive details:

http://technet.microsoft.com/en-us/library/cc770946.aspx[11]

Appendix D: Active Directory Domain Services – Configuration Summary has been provided as a convenience to assist in the installation and configuration of Active Directory.

5.3 Deploy Red Hat Enterprise Linux 6

The Red Hat Enterprise Linux 6 Installation Guide[1] provides complete details on the installation of Red Hat Enterprise Linux 6 for Intel, AMD, and IBM architectures. Red Hat Enterprise Linux 6 systems can be deployed as either physical or virtual machines.

Regardless of whether physical or virtual machines are used, a Red Hat Enterprise Linux 6 installation involves the following series of stages:

1. Install Red Hat Enterprise Linux 6
2. FirstBoot
3. Apply updates

After the operating system has been installed the system reboots and enters what is referred to as *FirstBoot*. During *FirstBoot*, administrators are guided through the process of setting date and time, configuring software updates, registering with Red Hat Network (RHN), initial user account creation and options for Kernel (*Kdump*) crash dumps. The system then reboots to activate the changes. After login has been completed under the newly created user account, updates to the system are applied to bring the Red Hat Enterprise Linux 6 system to the latest versions of all software. Updates to apply the most recent patches and security updates for Red Hat Enterprise Linux 6 can be performed at any time by running:

```
# yum update
```

The Red Hat Enterprise Linux 6 Installation Guide[1] provides complete instructions on each of these stages. Please consult the guide for further installation details.

5.4 Configure SELinux Security Parameters

By default, SELinux is enabled during the Red Hat Enterprise Linux 6 installation process. For maximum security, Red Hat recommends running Red Hat Enterprise Linux 6 with SELinux enabled. In this section, verification is done to ensure that SELinux is enabled and configured on system boot.

1. Verify whether or not SELinux is enabled using the getenforce utlity:

```
# getenforce
Enforcing
```

2. If getenforce returns “Permissive” then set to “Enforcing” and verify:

```
# getenforce
Permissive
# setenforce 1
# getenforce
Enforcing
```

3. Edit the file */etc/selinux/config* and set SELinux to be persistent across reboots:

```
SELINUX=enforcing
```

5.5 Install/Configure Samba

For Samba based configurations, the Samba packages must be installed and the services configured to start on system boot.

1. Install the Samba packages:

```
# yum -y install samba samba-client samba-common samba-winbind \
samba-winbind-clients

                          ...output abbreviated...

Installed:
  samba.x86_64 0:3.5.10-114.el6_0.2
  samba-client.x86_64 0:3.5.10-114.el6_0.2
  samba-common.x86_64 0:3.5.10-114.el6_0.2
  samba-winbind.x86_64 0:3.5.10-114.el6_0.2
  samba-winbind-clients.x86_64 0:3.5.10-114.el6_0.2
```

2. Start and verify Samba services are running:

```
# service smb start
Starting SMB services:                     [  OK  ]
# service smb status
smbd (pid  15123) is running...
# ps -aef | grep smb
root     15123     1  0 18:06 ?        00:00:00 smbd -D
root     15125 15123  0 18:06 ?        00:00:00 smbd -D
root     15136  2327  0 18:06 pts/0    00:00:00 grep smb
```

A base installation of Samba with no local file shares configured runs two instances of the Samba (***smbd***) daemon. The ***smbd*** daemon handles all connection requests and spawns a new process for each client connection made. It is not unusual for a large number of processes to be seen if there are many clients mapping locally served file shares.

3. Configure the Samba daemon to start on server boot:

```
# chkconfig smb on
# chkconfig --list smb
smb             0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

5.6 Synchronize Time Services

It is essential that the time service on the Red Hat Enterprise Linux 6 systems and Active Directory (Windows 2008) server are synchronized, otherwise Kerberos authentication may fail due to clock skew. In environments where time services are not reliable, best practice is to configure the Red Hat Enterprise Linux 6 systems to synchronize time from the Windows Server 2008 R2 server.

1. Edit the file */etc/ntp.conf* so that the Red Hat Enterprise Linux 6 system time is synchronized from a known, reliable time service:

```
# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats
server ns1.bos.redhat.com
server 10.5.26.10
```

2. Activate the change on the Red Hat Enterprise Linux 6 systems by stopping the ***ntp*** daemon, updating the time, then starting the ***ntp*** daemon. Verify the change on both servers:

Red Hat Enterprise Linux 6 system:

```
# service ntpd stop
Shutting down ntpd:                 [  OK  ]

# ntpdate 10.16.255.2
22 Mar 20:17:00 ntpdate[14784]: adjust time server 10.16.255.2 offset
-0.002933 sec
# service ntpd start
Starting ntpd:                      [  OK  ]
```

Windows Server 2008 R2 server:

```
C:\Users\Administrator> w32tm /query /status | find "Source"
Source: ns1.xxx.xxx.com

C:\Users\Administrator> w32tm /query /status | find "source"
Reference Id: 0x0A10FF02 (source IP:  10.nn.nnn.2)
```

3. Configure the ntpd daemon to start on server boot:

```
# chkconfig ntpd on
# chkconfig --list ntpd
smb             0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

5.7 Configure DNS

Proper resolution of DNS hostnames from both the Red Hat Enterprise Linux 6 systems and the Active Directory (Windows 2008) server are an essential prerequisite. Improperly resolved hostnames are one of the leading causes for integration failures. In environments where DNS lookups are not reliable, best practice is to configure the Red Hat Enterprise Linux 6 systems to perform DNS lookups from the Windows Server 2008 R2 Active Directory server.

1. Edit the file */etc/resolv.conf* so that the fully qualified domain name (FQDN) of the DNS servers is specified:

```
domain cloud.lab.eng.bos.redhat.com
search cloud.lab.eng.bos.redhat.com
nameserver 10.nn.nnn.3
nameserver 10.nn.nnn.247
nameserver 10.nn.nnn.2
```

2. Similarly, the hostname of the Red Hat Enterprise Linux 6 system should be set to the FQDN. Edit the file */etc/sysconfig/network* and set the hostname to use the FQDN:

```
HOSTNAME=rhel-srv11.cloud.lab.eng.bos.redhat.com
```

5.8 Install/Configure Kerberos Client

Best practice is to install and configure the Kerberos client (**krb5-workstation**) to insure Kerberos is able to properly authenticate to Active Directory on the Windows Server 2008 R2 server. This step is optional but highly recommended as it is useful for troubleshooting Kerberos authentication issues.

1. Verify the Kerberos client is installed:

```
# yum list installed | grep krb5
Complete!
# yum list installed | grep krb5
krb5-libs.x86_64            1.9-22.el6_2.1             @rhel-6-server-rpms
krb5-workstation.x86_64     1.9-22.el6_2.1             @rhel-6-server-rpms
pam_krb5.x86_64             2.3.11-9.el6               @anaconda-
RedHatEnterpriseLinux-201111171049.x86_64/6.2
```

2. If not, install it as follows:

```
# yum -y install krb5-workstation

                          ...output abbreviated...
Installed:
  krb5-workstation.x86_64 0:1.8.2-3.el6_0.6

Complete!
```

If Kerberos has not been previously configured, modify the Kerberos configuration file (*/etc/krb5.conf*) by adding entries for the new Kerberos and Active Directory realms. Note the differences in the Kerberos `[realms]` and Active Directory `[domain_realm]` realm entries.

1. Create a safety copy of the Kerberos configuration file:

```
# cp -p /etc/krb5.conf /etc/krb5.conf.orig
```

2. Edit the file */etc/krb5.conf* as follows – changes are highlighted bold:

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

[libdefaults]
  default_realm = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
  dns_lookup_realm = false
  dns_lookup_kdc = false
  ticket_lifetime = 24h
  renew_lifetime = 7d
  forwardable = true

[realms]
  REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM  = {
    kdc = WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
    admin_server = WIN-SRV1.REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
 }
```

```
[domain_realm]
  .demo = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
  demo = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
```

Under Kerberos, `[realms]` is set to the Kerberos server definitions and `[domain_realm]` defines the Active Directory server. Both are in the Active Directory REFARCH-AD domain.

3. Verify the Kerberos configuration. First, clear out any existing tickets:

```
# kdestroy
# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

4. Obtain a new Kerberos ticket:

```
# kinit administrator@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
Password for administrator@REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM: ********
```

5. Verify a new Kerberos ticket was granted:

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM

Valid starting     Expires            Service principal
03/22/12 19:31:49  03/23/12 05:31:52  krbtgt/REFARCH-
AD.CLOUD.LAB.ENG.BOS.REDHAT.COM@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
      renew until 03/29/12 19:31:49
```

At this point Kerberos is fully functional and the client utilities (*kinit, klist, kdestroy*) can be used for testing and verifying Kerberos functionality.

5.9 Install oddjob-mkhomedir

Install the ***oddjob-mkhomedir*** package to ensure that user home directories are created with the proper SELinux file and directory contexts:

```
# yum install oddjob-mkhomedir.x86_64
Loaded plugins: product-id, refresh-packagekit, rhnplugin, security,
subscription-manager
Updating certificate-based repositories.

Running Transaction
  Installing : oddjob-mkhomedir-0.30-5.el6.x86_64
1/1
Installed products updated.

Installed:
  oddjob-mkhomedir.x86_64 0:0.30-5.el6

                          ...output abbreviated...
Complete!
```

6 Recommended Configurations

In this section, the tasks necessary for integrating Red Hat Enterprise Linux 6 with Active Directory are provided for each of the four recommended configurations. Prior to proceeding, each of the following deployment prerequisites must first be completed:

Windows Server 2008 R2 server:

- Deploy Windows Server 2008 R2
- Configure Active Directory Domain Services

Red Hat Enterprise Linux 6 systems:

- Deploy Red Hat Enterprise Linux 6
- Configure SELinux Security Parameters
- Install/Configure Samba (*Recommended Configurations 1, 2 only*)
- Synchronize Time Services
- Configure DNS
- Install/Configure Kerberos Client

Do not proceed with the integration tasks for any configuration until each of the deployment prerequisites have been met. Refer to **Section 5 Deployment Prerequisites** for details on any deployment prerequisites that have not been completed.

For each of the four recommended configurations, a configuration summary table and figures depicting the systems overview and authentication and ID components is provided in addition to the integration steps.

Proceed to the appropriate section for the recommended configuration that was previously selected from **Section 4.3 Selecting a Configuration**.

6.1 Configuration 1 - Samba/Winbind (idmap_rid)

This configuration is for environments looking to integrate one or more Red Hat Enterprise Linux 6 systems into an Active Directory domain or forest with standardized, template-driven user configurations. Login access and file sharing services are provided.

6.1.1 Configuration Summary

Configuration 1 Samba/Winbind – idmap_rid “Template Driven”		
Components		
RHEL 6:	• Samba/Winbind	
Windows 2008 Server R2:	• Active Directory	
Services Provided	• File Sharing • Login access (RHEL command line, GUI access via AD credentials)	
Use Cases	• Environments looking to integrate one or more RHEL systems into an AD domain or forest with standardized, template-driven RHEL user configurations (shell, home directory).	
Authentication (pam)	• Windbind (pam_winbind)	
ID Tracking/ Name Resolution (nss)	• Windbind (nss_winbind)	
ID Mapping (“back-end”)	• Windbind (idmap_rid)	
Configuration Files	• /etc/krb5.conf • /etc/samba/smb.conf	• /etc/pam.d/passwd-auth • /etc/pam.d/system-auth
Advantages	• Least intrusive to AD environments • Template driven user configurations (shell, home directory) • SID mappings homogeneous across multiple RHEL servers • Uses algorithmic ID mappings across multiple servers (faster)	
Disadvantages	• Does not work with a forest of AD domains, multiple domain trees without additional configuration	
Notes	• Does not require modifications to user attributes within AD	

Table 6.1.1: Configuration Summary - Configuration 1

6.1.2 Systems Overview

Figure 6.1.2 provides a overview of the systems and services utilized by Configuration 1:

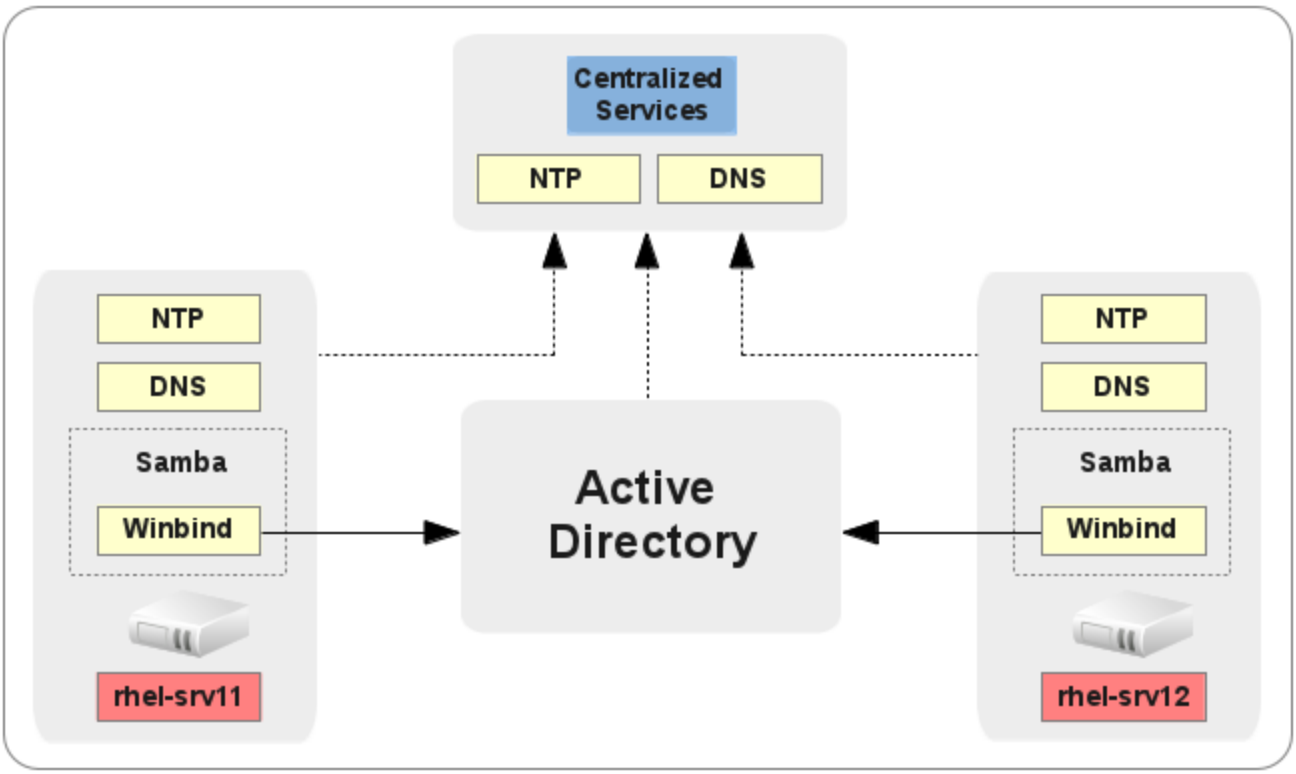

Figure 6.1.2: Systems Overview - Configuration 1

6.1.3 Authentication and ID Components

Figure 6.1.3 depicts the Authentication, ID Tracking, and ID Mapping for Configuration 1:

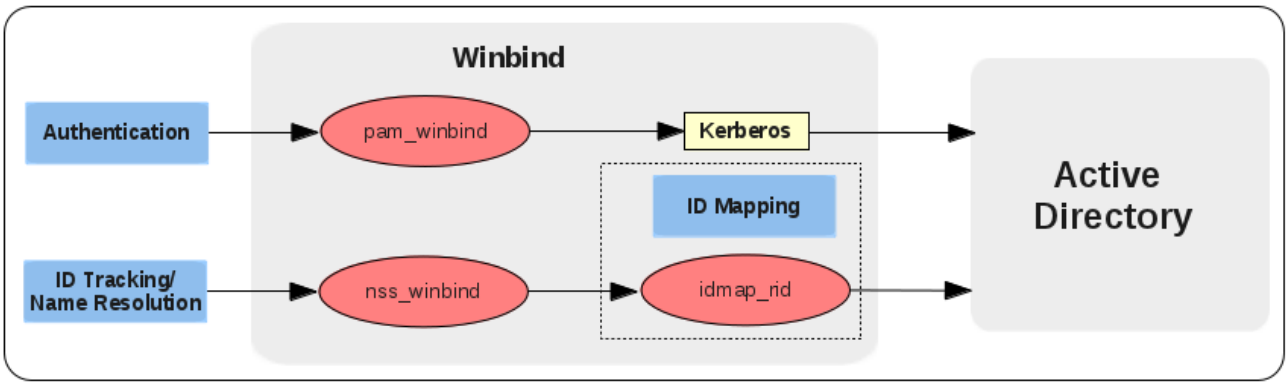

Figure 6.1.3: Authentication and ID Components – Configuration 1

6.1.4 Integration Tasks

Integrating Red Hat Enterprise Linux 6 into an Active Directory domain for Configuration 1 involves the following series of steps:

1. Configure Authentication
2. Verify/Test Active Directory
3. Modify Samba Configuration

The following provides a step-by-step guide to the integration process:

1. Configure Authentication

The **`system-config-authentication`** tool simplifies configuring the Samba, Kerberos, security, and authentication files for Active Directory integration. Invoke the tool as follows:

```
# system-config-authentication
```

Figure 6.1.4-1: User Account Database

On the **Identity & Authentication** tab, select the **User Account Database** drop-down then select **Winbind**.

A new set of fields is displayed. Selecting the **Winbind** option configures the system to connect to a Windows Active Directory domain. User information from a domain can then be accessed, and the following server authentication options can be configured:

- **Winbind Domain**: Windows Active Directory domain
- **Security Model**: The Samba client mode of operation. The drop-down list allows selection of the following options:

  ads - This mode instructs Samba to act as a domain member in an Active Directory Server (ADS) realm. To operate in this mode, the `krb5-server` package must be installed, and Kerberos must be configured properly.

  domain - In this mode, Samba attempts to validate the username/password by authenticating it through a Windows Active Directory domain server, similar to how a Windows Server would.

  server - In this mode, Samba attempts to validate the username/password by authenticating it through another SMB server. If the attempt fails, the user mode takes effect instead.

  user - This is the default mode. With this level of security, a client must first log in with a valid username and password. Encrypted passwords can also be used in this security mode.

- **Winbind ADS Realm**: When the ***ads*** Security Model is selected, this allows you to specify the ADS Realm the Samba server should act as a domain member of.
- **Winbind Domain Controllers**: Use this option to specify which domain server winbind should use.
- **Template Shell**: When filling out the user information for a Windows user, the `winbindd` daemon uses the value chosen here to specify the login shell for that user.
- **Allow offline login**: By checking this option, authentication information is stored in a local cache. This information is then used when a user attempts to authenticate while offline.

Populate the fields as follows:

User Account Database:	**Winbind**
Winbind Domain:	**REFARCH-AD**
Security Model:	**ads**
Winbind ADS Realm:	**REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Winbind Domain Controllers:	**WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Template Shell:	**/sbin/bash**

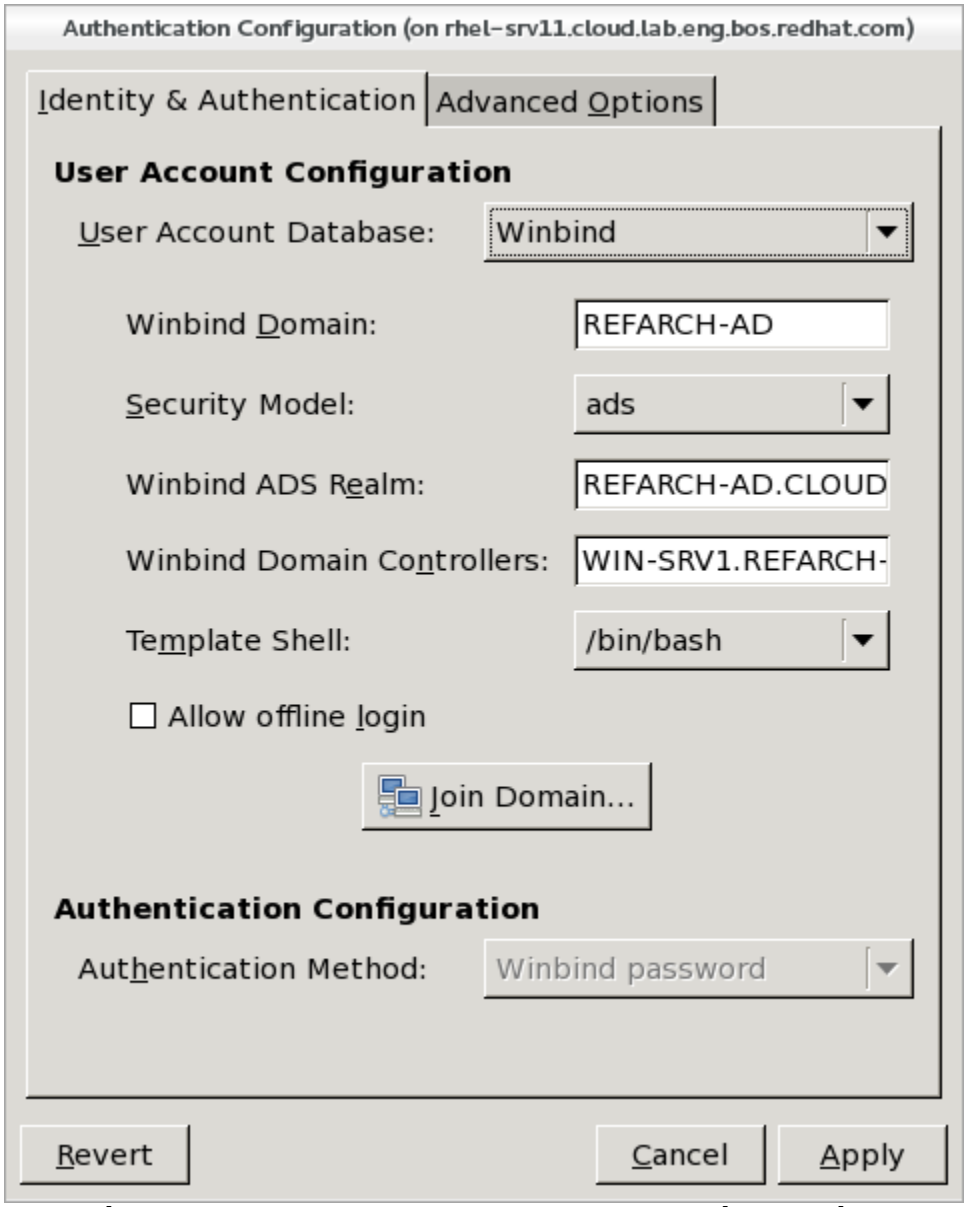

Figure 6.1.4-2: User Account Configuration

Select the **Advanced Options** tab when done.

Figure 6.1.4-3: Advanced Options

Select **Apply**. The terminal window indicated that the **oddjobd** daemon was started:

```
Starting oddjobd:
```

On first successful login to Active Directory, the **oddjobd** daemon calls a method to create a new home directory for a user.

Return to the **Identity & Authentication** tab, select **Join Domain**. An alert indicates the need to save the configuration changes to disk before continuing:

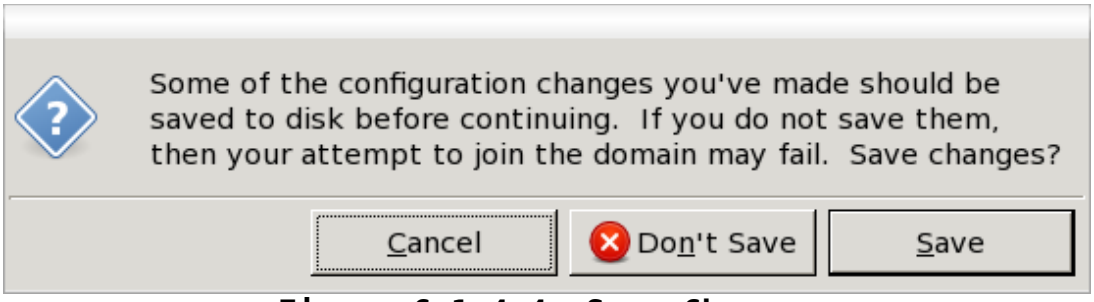

Figure 6.1.4-4: Save Changes

Select **Save**. A new window prompts for the Domain administrator password:

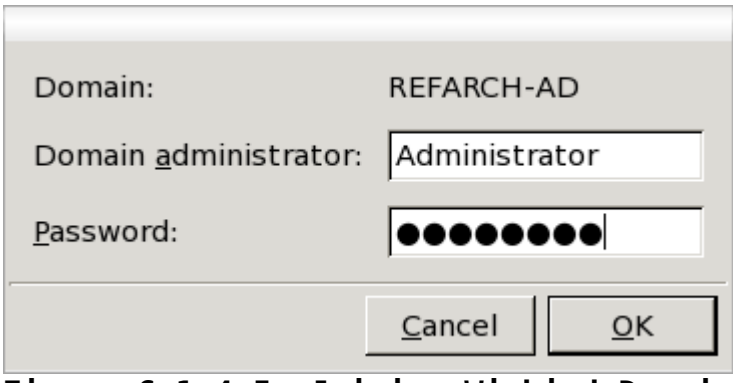

Figure 6.1.4-5: Joining Winbind Domain

Select **OK**. The terminal window displays the status of the domain join:

```
# Starting Winbind services:                    [  OK  ]
[/usr/bin/net join -w REFARCH-AD -S WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.
REDHAT.COM -U Administrator]
Enter Administrator's password:<...>

Using short domain name -- REFARCH-AD
Joined 'RHEL-SRV11' to realm 'refarch-ad.cloud.lab.eng.bos.redhat.com'
```

Select **Apply** then proceed to the next section.

2. Verify/Test Active Directory

The join to the Active Directory domain is complete. Verify access by performing each of the following tasks.

Test Connection to AD:

```
# net ads testjoin
Join is OK

# net ads info
LDAP server: 10.16.142.3
LDAP server name: WIN-SRV1.refarch-ad.cloud.lab.eng.bos.redhat.com
Realm: REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
Bind Path: dc=REFARCH-AD,dc=CLOUD,dc=LAB,dc=ENG,dc=BOS,dc=REDHAT,dc=COM
LDAP port: 389
Server time: Sun, 25 Mar 2012 18:20:00 EDT
KDC server: 10.16.142.3
Server time offset: -2
```

List members in domain:

```
# wbinfo --domain-users
REFARCH-AD\administrator
REFARCH-AD\guest
REFARCH-AD\krbtgt
REFARCH-AD\ad-user11
REFARCH-AD\ad-user12
REFARCH-AD\ad-user21
REFARCH-AD\ad-user22
REFARCH-AD\ad-user31
REFARCH-AD\ad-user32
REFARCH-AD\ad-user41
REFARCH-AD\ad-user42
```

List groups in domain:

```
# wbinfo --domain-groups
REFARCH-AD\domain computers
REFARCH-AD\domain controllers
REFARCH-AD\schema admins
REFARCH-AD\enterprise admins
REFARCH-AD\cert publishers
REFARCH-AD\domain admins
REFARCH-AD\domain users
                          ...output abbreviated...
```

Note: If either of these fail to return all users or groups in the domain, the idmap UID, GUI upper boundaries in the Samba configuration file need to be increased and the **winbind** and **smb** daemons restarted. These tasks are discussed in the next section.

3. Modify Samba Configuration

The previous sections configured Winbind by using the default backend to verify Active Directory domain access. Next, the Samba configuration file is modified to use the **`idmap_rid`** back-end and several other parameters are configured for convenience. **Table 6.1.4: Summary of Changes – Configuration 1** provides a summary of the configuration file parameter changes:

Configuration 1 Samba Configuration File Parameters	
Parameter	**Description**
idmap uid = 10000-19999	Set user id range for default backend (tdb)
idmap gid = 10000-19999	Set group id range for default backend (tdb)
idmap config REFARCH-AD:backend = rid	Configure winbind to use idmap_rid backend
idmap config REFARCH-AD:range = 10000000-19999999	Set range for idmap_rid backend
winbind enum users = no	Disable enumeration of users
winbind enum groups = no	Disable enumeration of groups
winbind separator = +	Change default separator from '\' to '+'
winbind use default domain = yes	Remove need to specify domain in commands
template homedir = /home/%D/%U	Set home directory to /home/REFARCH-AD/user
template shell = /bin/bash	Set login shell to /bin/bash

Table 6.1.4: Summary of Changes – Configuration 1

Make a safety copy of the Samba configuration file:

```
# cp -p /etc/samba/smb.conf /etc/samba/smb.conf.back
```

Edit and save the Samba configuration file as follows – changes are highlighted in bold:

```
[global]
   workgroup = REFARCH-AD
   password server = WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
   realm = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
   security = ads
   idmap uid = 10000-19999
   idmap gid = 10000-19999
   idmap config REFARCH-AD:backend = rid
   idmap config REFARCH-AD:range = 10000000-19999999
   winbind enum users = no
   winbind enum users = no
   winbind separator = +
   winbind use default domain = yes
   template homedir = /home/%D/%U
   template shell = /bin/bash
```

Test the new configuration file:

```
# testparm

Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
'winbind separator = +' might cause problems with group membership.
Server role: ROLE_DOMAIN_MEMBER
Press enter to see a dump of your service definitions

[global]
      workgroup = REFARCH-AD
      realm = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
      server string = Samba Server Version %v
      security = ADS
      allow trusted domains = No
      password server = WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
      log file = /var/log/samba/log.%m
      max log size = 50
      idmap uid = 10000-19999
      idmap gid = 10000-19999
      template shell = /bin/bash
      winbind separator = +
      winbind use default domain = Yes
      idmap config REFARCH-AD:range = 10000000-19999999
      idmap config REFARCH-AD:backend = rid
      cups options = raw
                          ...output abbreviated...
```

Backup and clear out the existing Samba cache files - requires services to be stopped:

```
# service smb stop
Shutting down SMB services:                [  OK  ]
# service winbind stop
Shutting down Winbind services:            [  OK  ]

# tar -cvf /var/tmp/samba-cache-backup.tar /var/lib/samba
tar: Removing leading `/' from member names
/var/lib/samba/
/var/lib/samba/gencache_notrans.tdb
/var/lib/samba/group_mapping.ldb
                          ...output abbreviated...
/var/lib/samba/winbindd_idmap.tdb
/var/lib/samba/smb_krb5/
/var/lib/samba/smb_krb5/krb5.conf.REFARCH-AD
# ls -la /var/tmp/samba-cache-backup.tar
-rw-r--r--. 1 root root 798720 Mar 28 14:32 /var/tmp/samba-cache-backup.tar

# rm -f /var/lib/samba/*
```

Verify no Kerberos tickets are in use:

```
# kdestroy
kdestroy: No credentials cache found while destroying cache
# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

Join the Active Directory domain:

```
# net join -S win-srv1 -U administrator
Enter Administrator's password:
Using short domain name -- REFARCH-AD
Joined 'RHEL-SRV11' to realm 'refarch-ad.cloud.lab.eng.bos.redhat.com'
```

Test connection to the Active Directory domain:

```
# net ads testjoin
Join is OK

# net ads info
LDAP server: 10.16.142.3
LDAP server name: WIN-SRV1.refarch-ad.cloud.lab.eng.bos.redhat.com
Realm: REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
Bind Path: dc=REFARCH-AD,dc=CLOUD,dc=LAB,dc=ENG,dc=BOS,dc=REDHAT,dc=COM
LDAP port: 389
Server time: Wed, 28 Mar 2012 14:45:54 EDT
KDC server: 10.16.142.3
Server time offset: 0
```

Start Winbind and Samba to activate the new configuration changes:

```
# service winbind start
Starting Winbind services:                 [  OK  ]
# service winbind status
winbindd (pid  11533) is running...
# ps -aef | grep winbind
root     11533     1  0 14:36 ?        00:00:00 winbindd

# service smb start
Starting SMB services:                                    [  OK  ]
# service smb status
smbd (pid  11508) is running...
# ps -aef | grep smb
root     11508     1  0 14:36 ?        00:00:00 smbd -D

root     11510 11508  0 14:36 ?        00:00:00 smbd -D
```

List members in domain:

```
# wbinfo --domain-users
administrator
guest
krbtgt
ad-user11
ad-user12
ad-user21
ad-user22
ad-user31
ad-user32
ad-user41
ad-user42
```

List groups in domain:

```
# wbinfo --domain-groups
domain computers
domain controllers
schema admins
enterprise admins
cert publishers
domain admins
domain users
group policy creator owners
ras and ias servers
allowed rodc password replication group
read-only domain controllers
enterprise read-only domain controllers
dnsadmins
dnsupdateproxy
```

6.1.5 Verification of Services

Verify the services provided by Configuration 1 by performing the tasks outlined in the following sections:

1. Login Access

```
# ssh ad-user11@rhel-srv11
ad-user11@rhel-srv11's password:
Creating home directory for ad-user11.

$ hostname
rhel-srv11.cloud.lab.eng.bos.redhat.com

$ id
uid=10001103(ad-user11) gid=10001115(rhel-users) groups=10001115(rhel-
users),10001(BUILTIN+users),10000513(domain users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user11
$ ls -ld
drwxr-xr-x. 4 ad-user11 domain users 4096 Mar 26 20:16 .

$ echo $SHELL
/bin/bash
```

Verify access from another Red Hat Enterprise Linux 6 system, using a different Active Directory user account:

```
$ hostname
rhel-srv12.cloud.lab.eng.bos.redhat.com

$ ssh ad-user12@rhel-srv11
ad-user12@rhel-srv11's password:
Creating home directory for ad-user12.

$ hostname
rhel-srv11.cloud.lab.eng.bos.redhat.com

$ id
uid=10001104(ad-user12) gid=10001115(rhel-users) groups=10001115(rhel-
users),10001(BUILTIN+users),10000513(domain users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user12
$ ls -ld
drwxr-xr-x. 4 ad-user12 domain users 4096 Mar 26 20:19 .

$ echo $SHELL
/bin/bash
```

2. File Share

Use the **smbclient** utility to determine what file shares are available on *win-srv1*:

```
$ id
uid=10001103(ad-user11) gid=10001115(rhel-users) groups=10001115(rhel-
users),10001(BUILTIN+users),10000513(domain users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ kinit
Password for ad-user11@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM:

$ smbclient -L win-srv1 -k
OS=[Windows Server 2008 R2 Enterprise 7601 Service Pack 1] Server=[Windows
Server 2008 R2 Enterprise 6.1]

      Sharename       Type      Comment
      ---------       ----      -------
      ADMIN$          Disk      Remote Admin
      C$              Disk      Default share
      IPC$            IPC       Remote IPC
      NETLOGON        Disk      Logon server share
      SYSVOL          Disk      Logon server share
      Win-Data        Disk
OS=[Windows Server 2008 R2 Enterprise 7601 Service Pack 1] Server=[Windows
Server 2008 R2 Enterprise 6.1]

      Server               Comment
      ---------            -------

      Workgroup            Master
      ---------            -------
```

Use the **smbclient** utility to view what files are available on the *Win-Data* file share:

```
$ smbclient //win-srv1/Win-Data -k
Domain=[REFARCH-AD] OS=[Windows Server 2008 R2 Enterprise 7601 Service Pack
1] Server=[Windows Server 2008 R2 Enterprise 6.1]

smb: \> showconnect
//win-srv1/Win-Data

smb: \> listconnect
0:    server=win-srv1, share=Win-Data

smb: \> ls
  .                                   D        0  Tue Mar 27 23:08:07 2012
  ..                                  D        0  Tue Mar 27 23:08:07 2012
  Win-Srv1.txt                        A      292  Tue Mar 27 23:08:07 2012

                34969 blocks of size 4194304. 19124 blocks available
smb: \> quit
```

Create a mount point, mount the file share and access a file:

```
# mkdir /mnt/Win-Data

# mount -t cifs //win-srv1/Win-Data /mnt/Win-Data -o username=ad-user11
Password:

# df -k -t cifs
Filesystem           1K-blocks      Used Available Use% Mounted on
//win-srv1/Win-Data  143234044  64930604  78303440  46% /mnt/Win-Data

# mount -t cifs
//win-srv1/Win-Data on /mnt/Win-Data type cifs (rw)

# ssh ad-user11@rhel-srv11
ad-user11@rhel-srv11's password:
Last login: Wed Mar 28 14:49:10 2012 from rhel-srv11.refarch-
ad.cloud.lab.eng.bos.redhat.com

$ cat /mnt/Win-Data/Win-Srv1.txt

+-------------------------------------------------------+
+  This file is located on the Windows Server 2008 R2   +
+  server named 'win-srv1.cloud.lab.eng.bos.redhat.com' +
+  located in the Active Directory domain 'REFARCH-AD'  +
+-------------------------------------------------------+
```

This completes the process of integrating a Red Hat Enterprise Linux 6 system into an Active Directory domain using Samba/Winbind and the **idmap_rid** backend. If there are multiple Red Hat Enterprise Linux 6 systems to be integrated, repeat the integration tasks for each system and verify the services provided.

6.2 Configuration 2 – Samba/Winbind (idmap_ad)

This configuration is for environments looking to integrate one or more Red Hat Enterprise Linux 6 systems into an Active Directory domain or forest with the capability to customize user configurations. Login access and file sharing services are provided.

6.2.1 Configuration Summary

Configuration 2 Samba/Winbind – idmap_ad "Customizable"		
Components		
RHEL 6:	• Samba/Winbind	
Windows 2008 Server R2:	• Active Directory • Identity Management for UNIX (IMU)	
Services Provided	• File Sharing • Login access (RHEL command line, GUI access via AD credentials)	
Use Cases	• Environments looking to integrate one or more RHEL systems into an AD domain or forest with the capability to customize RHEL user configurations (shell, home directory).	
Authentication (pam)	• Windbind (pam_winbind)	
ID Tracking/ Name Resolution (nss)	• Windbind (nss_winbind)	
ID Mapping ("back-end")	• Windbind (idmap_ad)	
Configuration Files	• /etc/krb5.conf • /etc/samba/smb.conf	• /etc/pam.d/passwd-auth • /etc/pam.d/system-auth
Advantages	• SID mappings homogeneous across multiple RHEL servers • Customizeable user configurations (shell, home directory) (configured within AD) • Centralized user account management • SFU, RFC2307 compatible mappings	
Disadvantages	• Requires additional configuration work to support a forest of AD domains or multiple domain trees • Requires additional user management tasks – user/group ID attributes must be set within AD	
Notes	• Requires the ability to modify user attributes within AD (via IMU)	

Table 6.2.1: Configuration Summary - Configuration 2

6.2.2 Systems Overview

Figure 6.2.2 provides a overview of the systems and services utilized by Configuration 2:

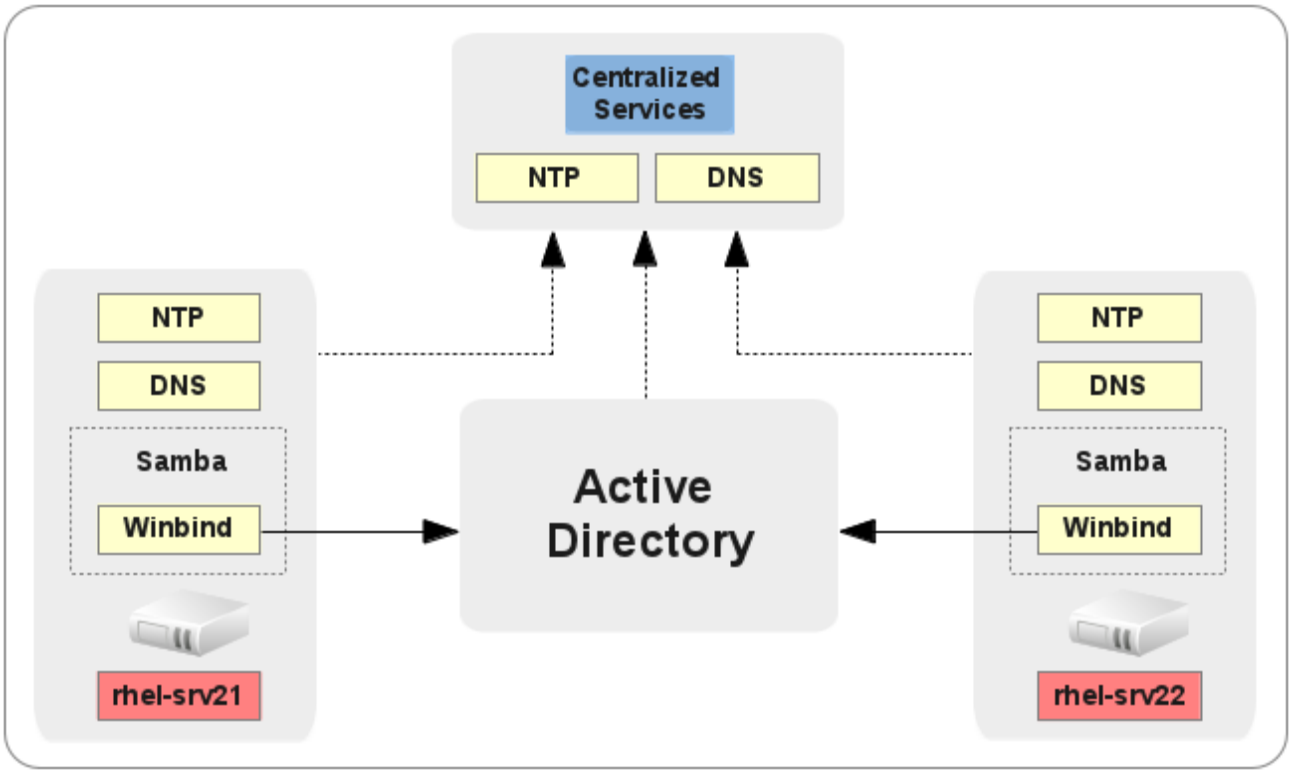

Figure 6.2.2: Systems Overview - Configuration 2

6.2.3 Authentication and ID Components

Figure 6.2.3 depicts the Authentication, ID Tracking and ID Mapping for Configuration 2:

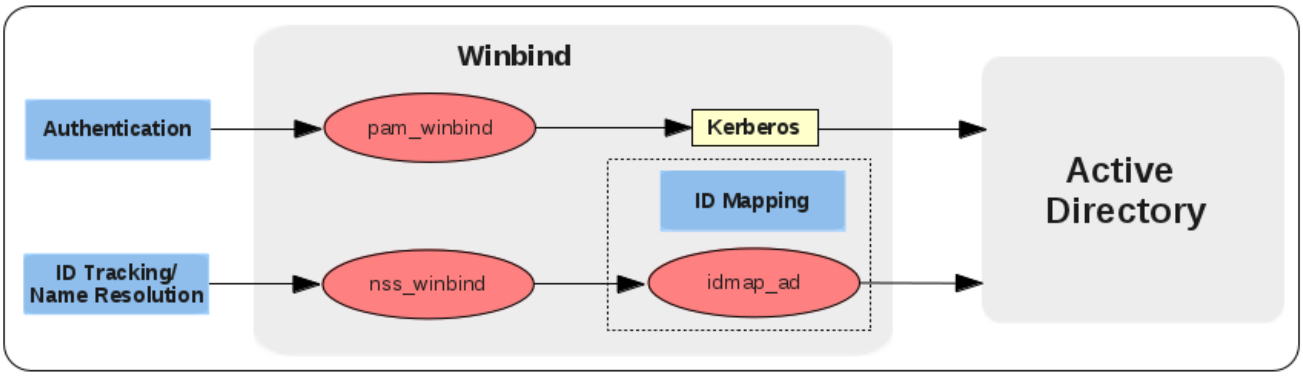

Figure 6.2.3: Authentication and ID Component - Configuration 2

6.2.4 Integration Tasks

Integrating Red Hat Enterprise Linux 6 into an Active Directory domain for Configuration 2 involves the following series of steps:

1. Configure Authentication
2. Verify/Test Active Directory
3. Modify Samba Configuration

The following provides a step-by-step guide to the integration process:

1. Configure Authentication

The **system-config-authentication** tool simplifies configuring the Samba, Kerberos, security and authentication files for Active Directory integration. Invoke the tool as follows:

```
# system-config-authentication
```

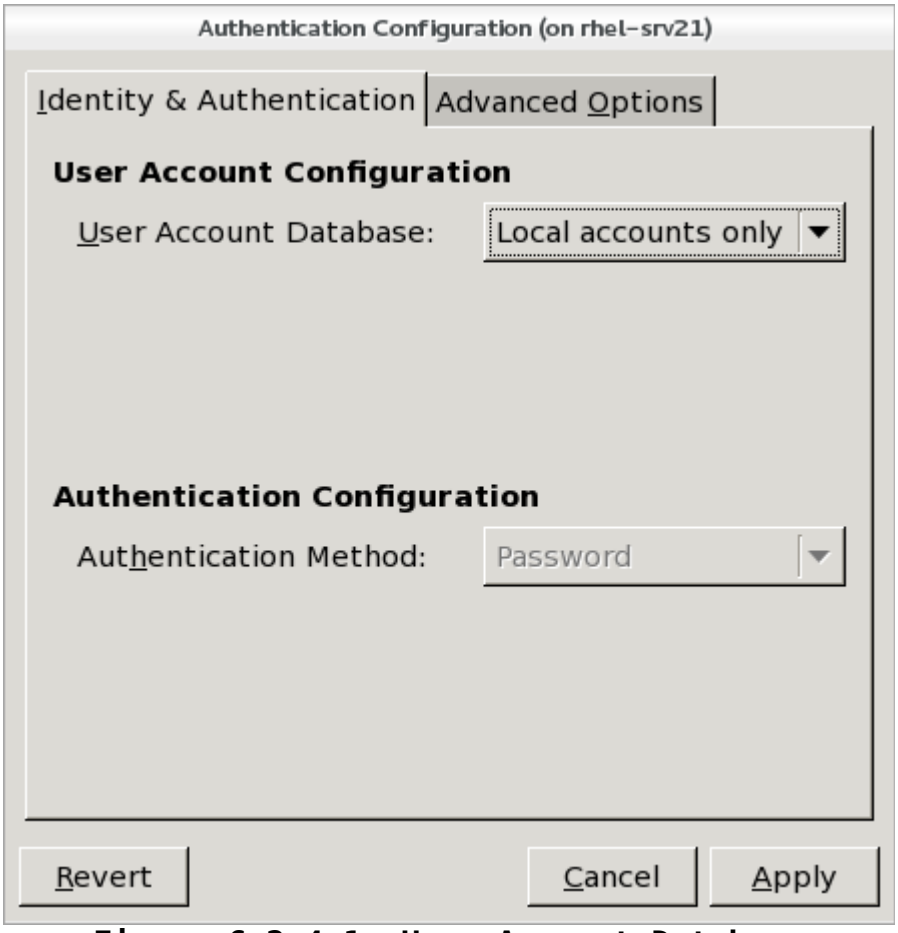

Figure 6.2.4-1: User Account Database

On the **Identity & Authentication** tab, select the **User Account Database** drop-down then select **Winbind**.

A new set of fields is displayed. Selecting the **Winbind** option configures the system to connect to a Windows Active Directory domain. User information from a domain can then be accessed, and the following server authentication options can be configured:

- **Winbind Domain**: Windows Active Directory domain
- **Security Model**: The Samba client mode of operation. The drop-down list allows selection of the following options:

  ads - This mode instructs Samba to act as a domain member in an Active Directory Server (ADS) realm. To operate in this mode, the `krb5-server` package must be installed, and Kerberos must be configured properly.

  domain - In this mode, Samba attempts to validate the username/password by authenticating it through a Windows Active Directory domain server, similar to how a Windows Server would.

  server - In this mode, Samba attempts to validate the username/password by authenticating it through another SMB server. If the attempt fails, the user mode takes effect instead.

  user - This is the default mode. With this level of security, a client must first log in with a valid username and password. Encrypted passwords can also be used in this security mode.

- **Winbind ADS Realm**: When the ***ads*** Security Model is selected, this allows you to specify the ADS Realm the Samba server should act as a domain member of.
- **Winbind Domain Controllers**: Use this option to specify which domain server winbind should use.
- **Template Shell**: When filling out the user information for a Windows user, the `winbindd` daemon uses the value chosen here to specify the login shell for that user.
- **Allow offline login**: By checking this option, authentication information is stored in a local cache. This information is then used when a user attempts to authenticate while offline.

Populate the fields as follows:

User Account Database:	**Winbind**
Winbind Domain:	**REFARCH-AD**
Security Model:	**ads**
Winbind ADS Realm:	**REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Winbind Domain Controllers:	**WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Template Shell:	**/sbin/false**

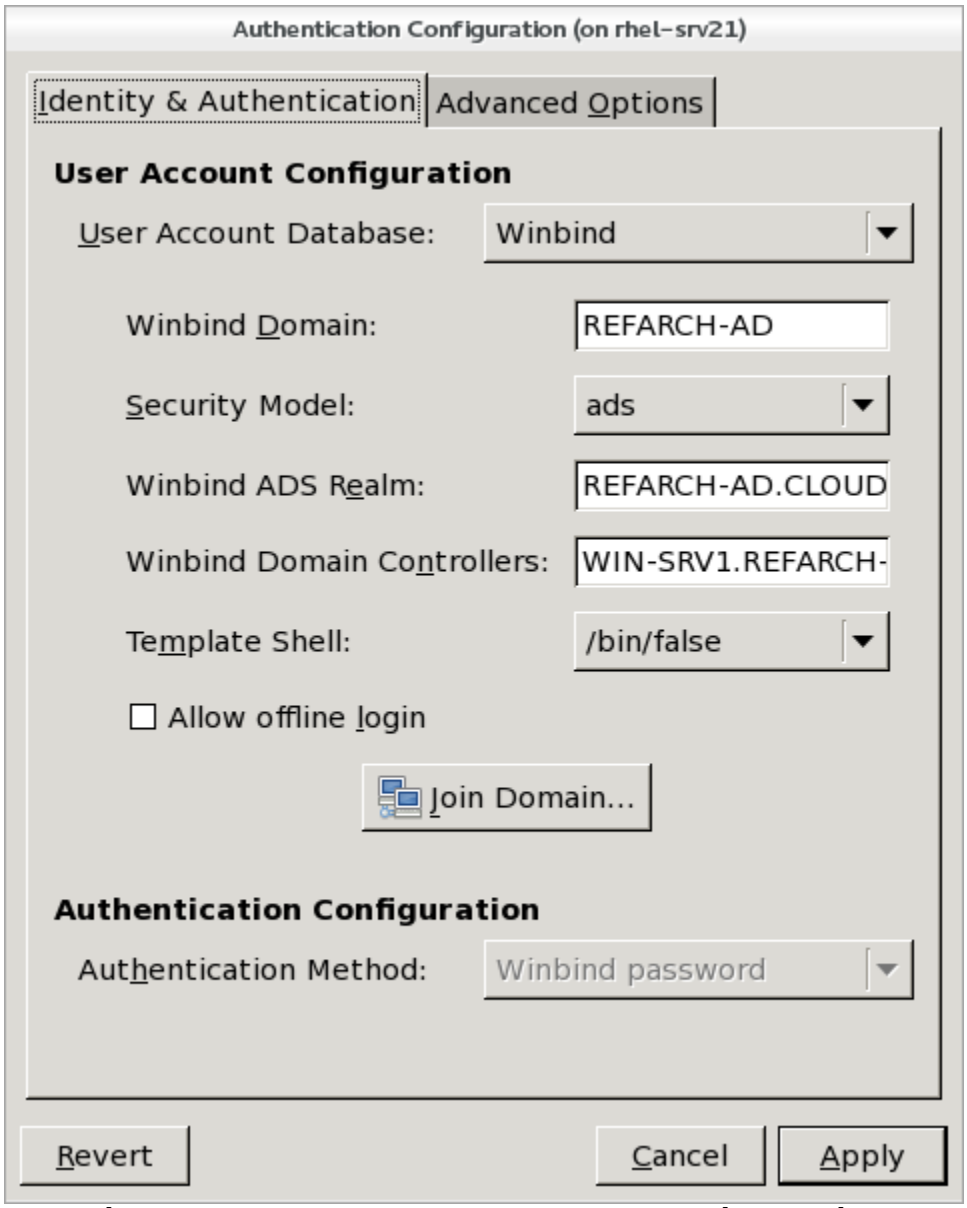

Figure 6.2.4-2: User Account Configuration

Select the **Advanced Options** tab when done.

Under **Other Authentication Options**, select **Create home directories on the first login**.

Figure 6.2.4-3: Advanced Options

Select **Apply**. The terminal window indicated that the **oddjobd** daemon was started:

```
Starting oddjobd:
```

On first successful login to Active Directory, the **oddjobd** daemon calls a method to create a new home directory for a user.

Return to the **Identity & Authentication** tab, select **Join Domain**. An alert indicates the need to save the configuration changes to disk before continuing:

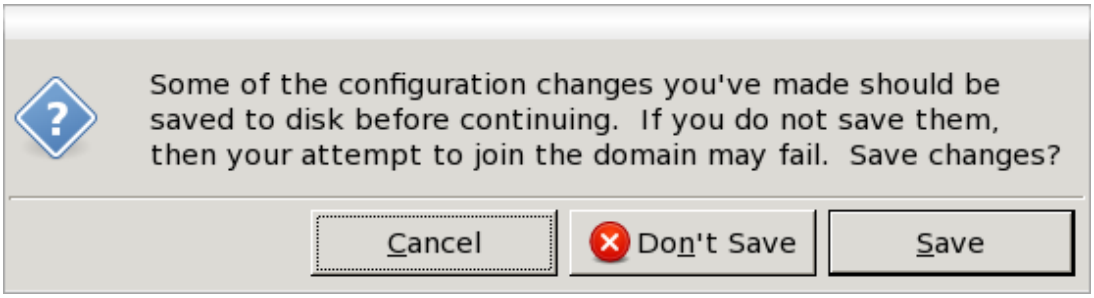

Figure 6.2.4-4: Save Changes

Select **Save**. A new window prompts for the Domain administrator password:

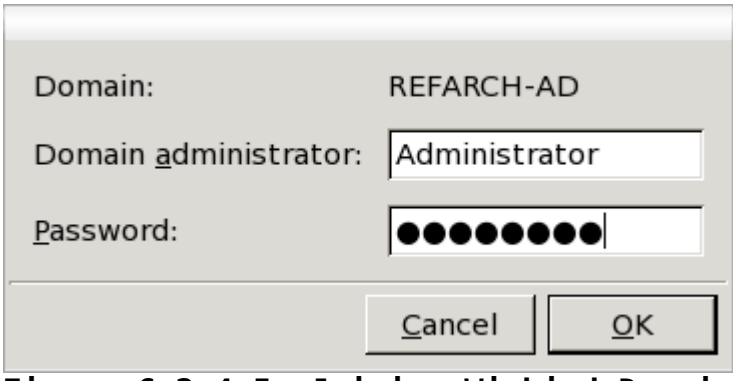

Figure 6.2.4-5: Joining Winbind Domain

Select **OK**. The terminal window displays the status of the domain join:

```
# Starting Winbind services:                    [  OK  ]
[/usr/bin/net join -w REFARCH-AD -S WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.
REDHAT.COM -U Administrator]
Enter Administrator's password:<...>

Using short domain name -- REFARCH-AD
Joined 'RHEL-SRV21' to realm 'refarch-ad.cloud.lab.eng.bos.redhat.com'
```

Select **Apply** then proceed to the next section.

2. Verify/Test Active Directory

The join to the Active Directory domain is complete. Verify access by performing each of the following tasks.

Test Connection to AD:

```
# net ads testjoin
Join is OK
```

List members in domain:

```
# wbinfo -u
REFARCH-AD\administrator
REFARCH-AD\guest
REFARCH-AD\krbtgt
REFARCH-AD\ad-user11
REFARCH-AD\ad-user12
REFARCH-AD\ad-user21
REFARCH-AD\ad-user22
REFARCH-AD\ad-user31
REFARCH-AD\ad-user32
REFARCH-AD\ad-user41
REFARCH-AD\ad-user42
```

List groups in domain:

```
# wbinfo -g
REFARCH-AD\domain computers
REFARCH-AD\domain controllers
REFARCH-AD\schema admins
REFARCH-AD\enterprise admins
REFARCH-AD\cert publishers
REFARCH-AD\domain admins
REFARCH-AD\domain users
REFARCH-AD+group policy creator owners
REFARCH-AD+ras and ias servers
REFARCH-AD+allowed rodc password replication group

                          ...output abbreviated...
```

Note: If either of these fail to return all users or groups in the domain, the idmap UID, GUI upper boundaries in the Samba configuration file need to be increased and the **winbind** and **smb** daemons restarted. These tasks are discussed in the next section.

3. Modify Samba Configuration

The previous sections configured Winbind by using the default backend to verify Active Directory domain access. Next, the Samba configuration file is modified to use the **`idmap_ad`** back-end and several other parameters are configured for convenience. **Table 6.2.4: Summary of Changes – Configuration 2** provides a summary of the configuration file parameter changes:

Configuration 2 Samba Configuration File Parameters	
Parameter	**Description**
idmap uid = 20000-29999	Set user id range for default backend (tdb)
idmap gid = 20000-29999	Set group id range for default backend (tdb)
idmap config REFARCH-AD:backend = ad	Configure winbind to use idmap_ad backend
idmap config REFARCH-AD:default = yes	Configure REFARCH-AD as default domain
idmap config REFARCH-AD:range = 10000000-29999999	Set range for idmap_ad backend
idmap config REFARCH-AD: schema_mode = rfc2307	Enable support for rfc2307 UNIX attributes
winbind nss_info = rfc2307	Obtain user home directory and shell from AD
winbind enum users = no	Disable enumeration of users
winbind enum groups = no	Disable enumeration of groups
winbind separator = +	Change default separator from '\' to '+'
winbind use default domain = yes	Remove need to specify domain in commands
winbind nested groups = yes	Enable nesting of groups in Active Directory

Table 6.2.4: Summary of Changes – Configuration 2

Make a safety copy of the Samba configuration file:

```
# cp -p /etc/samba/smb.conf /etc/samba/smb.conf.back
```

Edit and save the Samba configuration file as follows – changes are highlighted in bold:

```
[global]
   workgroup = REFARCH-AD
   password server = WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
   realm = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
   security = ads
   idmap uid = 20000-29999
   idmap gid = 20000-29999
   idmap config REFARCH-AD:backend = ad
   idmap config REFARCH-AD:default = yes
   idmap config REFARCH-AD:range = 10000000-29999999
   idmap config REFARCH-AD:schema_mode = rfc2307
   winbind nss info = rfc2307
```

```
   winbind enum users = no
   winbind enum groups = no
   winbind separator = +
   winbind use default domain = yes
   winbind nested groups = yes
```

Test the new configuration file:

```
# testparm

Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
'winbind separator = +' might cause problems with group membership.
Server role: ROLE_DOMAIN_MEMBER
Press enter to see a dump of your service definitions

[global]
      workgroup = REFARCH-AD
      realm = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
      server string = Samba Server Version %v
      security = ADS
      password server = WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
      log file = /var/log/samba/log.%m
      max log size = 50
      idmap uid = 20000-29999
      idmap gid = 20000-29999
      winbind separator = +
      winbind use default domain = Yes
      winbind nss info = rfc2307
      idmap config REFARCH-AD:schema_mode = rfc2307
      idmap config REFARCH-AD:range = 10000000-29999999
      idmap config REFARCH-AD:default = false
      idmap config REFARCH-AD:backend = ad
      cups options = raw
                          ...output abbreviated...
```

Backup and clear out the existing Samba cache files - requires services to be stopped:

```
# service smb stop
Shutting down SMB services:                [  OK  ]
# service winbind stop
Shutting down Winbind services:            [  OK  ]

# tar -cvf /var/tmp/samba-cache-backup.tar /var/lib/samba
tar: Removing leading `/' from member names
/var/lib/samba/
/var/lib/samba/smb_krb5/
/var/lib/samba/smb_krb5/krb5.conf.REFARCH-AD
                          ...output abbreviated...
/var/lib/samba/registry.tdb
```

```
/var/lib/samba/perfmon/
/var/lib/samba/winbindd_idmap.tdb

# ls -la /var/tmp/samba-cache-backup.tar
-rw-r--r--. 1 root root 788480 Mar 28 15:34 /var/tmp/samba-cache-backup.tar
# rm -f /var/lib/samba/*
```

Verify no Kerberos tickets are in use:

```
# kdestroy
kdestroy: No credentials cache found while destroying cache
# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

Join the Active Directory domain:

```
# net join -S win-srv1 -U administrator
Enter Administrator's password:
Using short domain name -- REFARCH-AD
Joined 'RHEL-SRV21' to realm 'refarch-ad.cloud.lab.eng.bos.redhat.com'
```

Test connection to the Active Directory domain:

```
# net ads testjoin
Join is OK

# net ads info
LDAP server: 10.16.142.3
LDAP server name: WIN-SRV1.refarch-ad.cloud.lab.eng.bos.redhat.com
Realm: REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
Bind Path: dc=REFARCH-AD,dc=CLOUD,dc=LAB,dc=ENG,dc=BOS,dc=REDHAT,dc=COM
LDAP port: 389
Server time: Wed, 28 Mar 2012 15:45:37 EDT
KDC server: 10.16.142.3
Server time offset: -1
```

Start Winbind and Samba to activate the new configuration changes:

```
# service winbind start
Starting Winbind services:                 [  OK  ]
# service winbind status
winbindd (pid  9582) is running...
# ps -aef | grep winbind
root      9582     1  0 15:40 ?        00:00:00 winbindd
root      9584  9582  0 15:40 ?        00:00:00 winbindd

# service smb start
Starting SMB services:                     [  OK  ]
# service smb status
smbd (pid  9553) is running...
# ps -aef | grep smb
root      9553     1  0 15:39 ?        00:00:00 smbd -D

root      9555  9553  0 15:39 ?        00:00:00 smbd -D
```

List members in domain:

```
# wbinfo --domain-users
administrator
guest
krbtgt
ad-user11
ad-user12
ad-user21
ad-user22
ad-user31
ad-user32
ad-user41
ad-user42
```

List groups in domain:

```
# wbinfo --domain-groups
domain computers
domain controllers
schema admins
enterprise admins
cert publishers
domain admins
domain users
group policy creator owners
ras and ias servers
allowed rodc password replication group
read-only domain controllers
enterprise read-only domain controllers
dnsadmins
dnsupdateproxy
```

6.2.5 Verification of Services

Verify the services provided by Configuration 2 by performing the tasks outlined in the following sections:

1. Login Access

```
$ ssh ad-user21@rhel-srv21
ad-user21@rhel-srv21's password:
Creating home directory for ad-user21.

$ hostname
rhel-srv21.cloud.lab.eng.bos.redhat.com

$ id
uid=20000021(ad-user21) gid=10000002(rhel-users) groups=10000002(rhel-
users),20001(BUILTIN+users),10000003(domain users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user21
$ ls -ld
drwxr-xr-x. 4 ad-user21 domain users 4096 Mar 26 20:33 .

$ echo $SHELL
/bin/bash
```

Verify access from another Red Hat Enterprise Linux 6 system, using a different Active Directory user account:

```
$ hostname
rhel-srv22.cloud.lab.eng.bos.redhat.com

$ ssh ad-user22@rhel-srv21
ad-user22@rhel-srv21's password:
Creating home directory for ad-user22.

$ hostname
rhel-srv21.cloud.lab.eng.bos.redhat.com

$ id
uid=20000022(ad-user22) gid=10000002(rhel-users) groups=10000002(rhel-
users),20001(BUILTIN+users),10000003(domain users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user22
$ ls -ld
drwxr-xr-x. 4 ad-user22 domain users 4096 Mar 26 20:49 .

$ echo $SHELL
/bin/bash
```

2. File Share

Use the **smbclient** utility to determine what file shares are available on *win-srv1*:

```
$ id
uid=20000021(ad-user21) gid=10000002(rhel-users) groups=10000002(rhel-
users),20001(BUILTIN+users),10000003(domain users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ kinit
Password for ad-user21@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM:

$ smbclient -L win-srv1 -k
Enter ad-user21's password:
Domain=[REFARCH-AD] OS=[Windows Server 2008 R2 Enterprise 7601 Service Pack
1] Server=[Windows Server 2008 R2 Enterprise 6.1]

      Sharename       Type      Comment
      ---------       ----      -------
      ADMIN$          Disk      Remote Admin
      C$              Disk      Default share
      IPC$            IPC       Remote IPC
      NETLOGON        Disk      Logon server share
      SYSVOL          Disk      Logon server share
      Win-Data        Disk
Domain=[REFARCH-AD] OS=[Windows Server 2008 R2 Enterprise 7601 Service Pack
1] Server=[Windows Server 2008 R2 Enterprise 6.1]

      Server               Comment
      ---------            -------

      Workgroup            Master
      ---------            -------
```

Use the **smbclient** utility to view what files are available on the *Win-Data* file share:

```
$ smbclient //win-srv1/Win-Data -k
Enter ad-user11's password:
Domain=[REFARCH-AD] OS=[Windows Server 2008 R2 Enterprise 7601 Service Pack
1] Server=[Windows Server 2008 R2 Enterprise 6.1]

smb: \> showconnect
//win-srv1/Win-Data

smb: \> listconnect
0:    server=win-srv1, share=Win-Data

smb: \> ls
  .                                   D        0  Tue Mar 27 23:08:07 2012
  ..                                  D        0  Tue Mar 27 23:08:07 2012
  Win-Srv1.txt                        A      292  Tue Mar 27 23:08:07 2012

                34969 blocks of size 4194304. 19124 blocks available
smb: \> quit
```

Create a mount point, mount the file share and access a file:

```
# mkdir /mnt/Win-Data

# mount -t cifs //win-srv1/Win-Data /mnt/Win-Data -o username=ad-user21
Password:

# df -k -t cifs
Filesystem           1K-blocks      Used Available Use% Mounted on
//win-srv1/Win-Data  143234044  64930604  78303440  46% /mnt/Win-Data

# mount -t cifs
//win-srv1/Win-Data on /mnt/Win-Data type cifs (rw)

# ssh ad-user11@rhel-srv21
ad-user11@rhel-srv11's password:
Last login: Wed Mar 28 14:49:10 2012 from rhel-srv21.refarch-
ad.cloud.lab.eng.bos.redhat.com

$ cat /mnt/Win-Data/Win-Srv1.txt

+-------------------------------------------------------+
+  This file is located on the Windows Server 2008 R2   +
+  server named 'win-srv1.cloud.lab.eng.bos.redhat.com' +
+  located in the Active Directory domain 'REFARCH-AD'  +
+-------------------------------------------------------+
```

This completes the process of integrating a Red Hat Enterprise Linux 6 system into an Active Directory domain using Samba/Winbind and the **idmap_ad** backend. If there are multiple Red Hat Enterprise Linux 6 systems to be integrated, repeat the integration tasks for each system and verify the services provided.

6.3 Configuration 3 – SSSD/Kerberos/LDAP

This configuration is for environments looking to integrate one or more Red Hat Enterprise Linux 6 systems into an Active Directory domain or forest with the enhanced authentication and caching capabilities offered by SSSD. Login access is the only service provided.

6.3.1 Configuration Summary

Configuration 3 SSSD/Kerberos/LDAP "Enhanced"		
Components		
RHEL 6:	• SSSD • Kerberos • LDAP	
Windows 2008 Server R2:	• Active Directory • Identity Management for UNIX (IMU)	
Services Provided	• Login access (RHEL command line, GUI access via AD credentials)	
Use Cases	• Environments looking to integrate one or more RHEL systems into an AD domain or forest with enhanced authentication and caching capabilities.	
Authentication (pam)	• Kerberos (pam_sss)	
ID Tracking/ Name Resolution (nss)	• LDAP (nss_sss)	
ID Mapping ("back-end")	• n/a	
Configuration Files	• /etc/krb5.conf • /etc/sssd.conf	• /etc/pam.d/passwd-auth • /etc/pam.d/system-auth
Advantages	• Kerberos SSO capable • Supports SASL/GSSAPI binds for LDAP queries (optional) • Enforces encrypted authentication only • Client side caching of user information • Off-line caching of previously authenticated user credentials • Reduces number of client queries to server • Graceful ID collision management	
Disadvantages	• No file sharing (can be enabled by configuring Samba)	
Notes	• Newer configuration with enhanced features, flexible capabilities • Requires the ability to modify user attributes within AD	

Table 6.3.1: Configuration Summary - Configuration 3

6.3.2 Systems Overview

Figure 6.3.2 provides a overview of the systems and services utilized by Configuration 3:

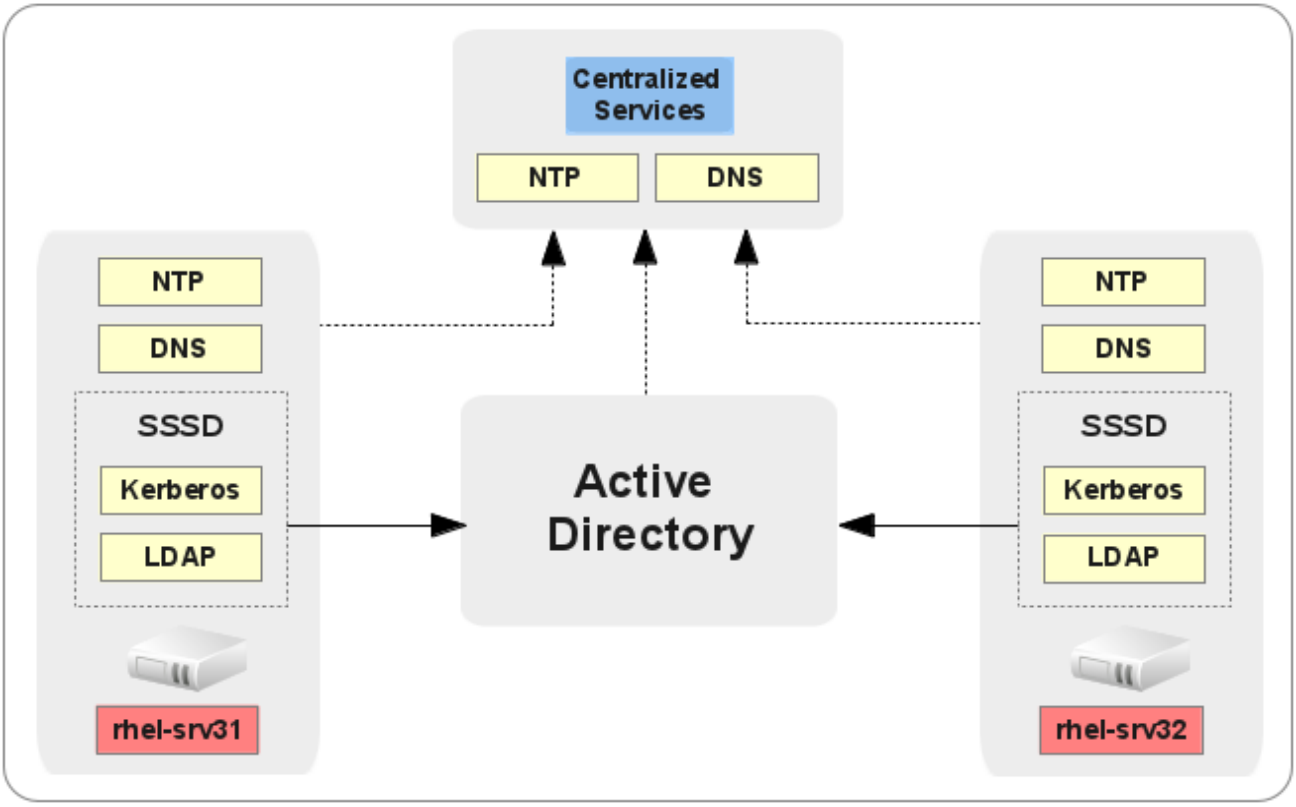

Figure 6.3.2: Systems Overview - Configuration 3

6.3.3 Authentication and ID Components

Figure 6.3.3 depicts the Authentication and ID Tracking for Configuration 3:

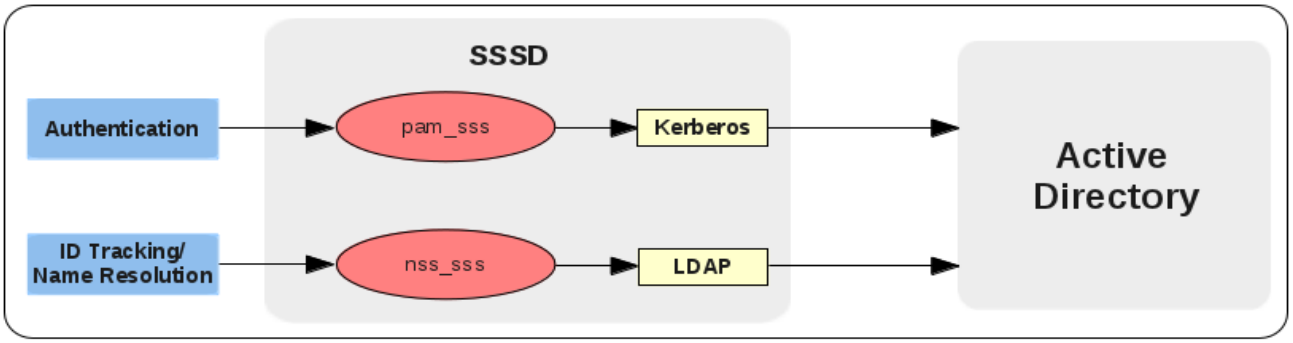

Figure 6.3.3: Authentication and ID Components - Configuration 3

6.3.4 Integration Tasks

Integrating Red Hat Enterprise Linux 6 into an Active Directory domain for Configuration 3 involves the following series of steps:

1. Configure Authentication
2. Enable LDAP Searches
3. Modify SSSD Configuration

The following provides a step-by-step guide to the integration process:

1. Configure Authentication

The **`system-config-authentication`** tool simplifies configuring the SSSD, Kerberos, LDAP security and authentication files for Active Directory integration. Invoke the tool as follows:

```
# system-config-authentication
```

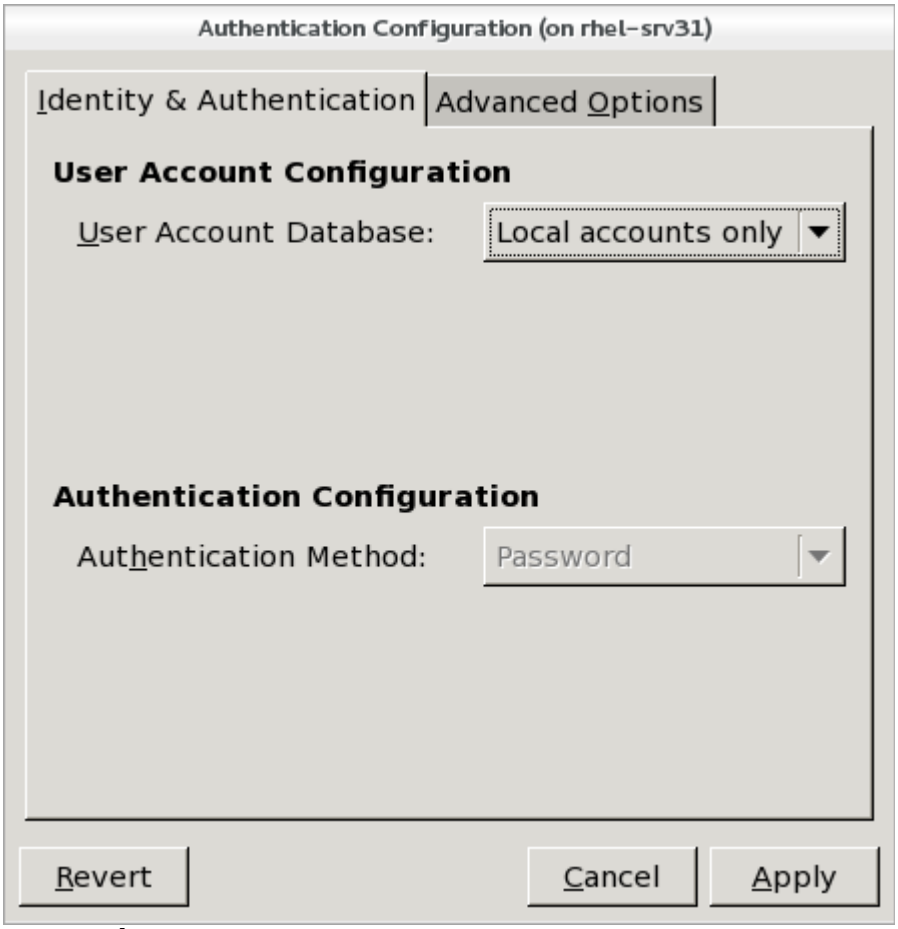

Figure 6.3.4-1: User Account Database

On the **Identity & Authentication** tab, select the **User Account Database** drop-down then select **LDAP**.

A new set of fields is displayed. Selecting the **LDAP** option allows the system to be configured to connect to the Windows Active Directory domain using LDAP with Kerberos authentication. User information from a domain can then be accessed and authenticated. The following server user account and authentication options can be configured:

- **LDAP Search Base DN**: Specifies that user information should be retrieved using the listed Distinguished Name (DN).
- **LDAP Server**: Specifies the address of the LDAP server.
- **Use TLS to encrypt connections**: When enabled, transport *Layer Security* (TLS) is used to encrypt passwords sent to the LDAP server.
    - The **Download CA Certificate** option allows a URL to be specified from which to download a valid *Certificate Authority* (CA) certificate. Valid CA certificates must be in the *Privacy Enhanced Mail* (PEM) format.
- **Authentication method**: Kerberos password - this option enables Kerberos authentication. The following options are available:
    - **Realm**: Configures the realm for the Kerberos server. The realm is the network that users Kerberos, comprised of one or more KDCs and Kerberos clients.
    - **KDCs**: Specifies the *Key Distribution Center* (KDC) for issuing Kerberos tickets.
    - **Admin Servers**: Specifies the administration server(s) running **kadmind**.
- **Use DNS to resolve hosts to realms**: Allows the use of DNS to find Kerberos realms.
- **Use DNS to locate KDCs for realms**: Allows the use of DNS to find Kerberos KDCs.

If Kerberos has been properly configured and verified as per the deployment prerequisites outlined in **Section 5.8 Install/Configure Kerberos Client** then the screen appears as seen on the following page.

Populate the fields as follows:

User Account Database:	**LDAP**
Realm:	**REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
KDCs:	**WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Admin Servers:	**WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**

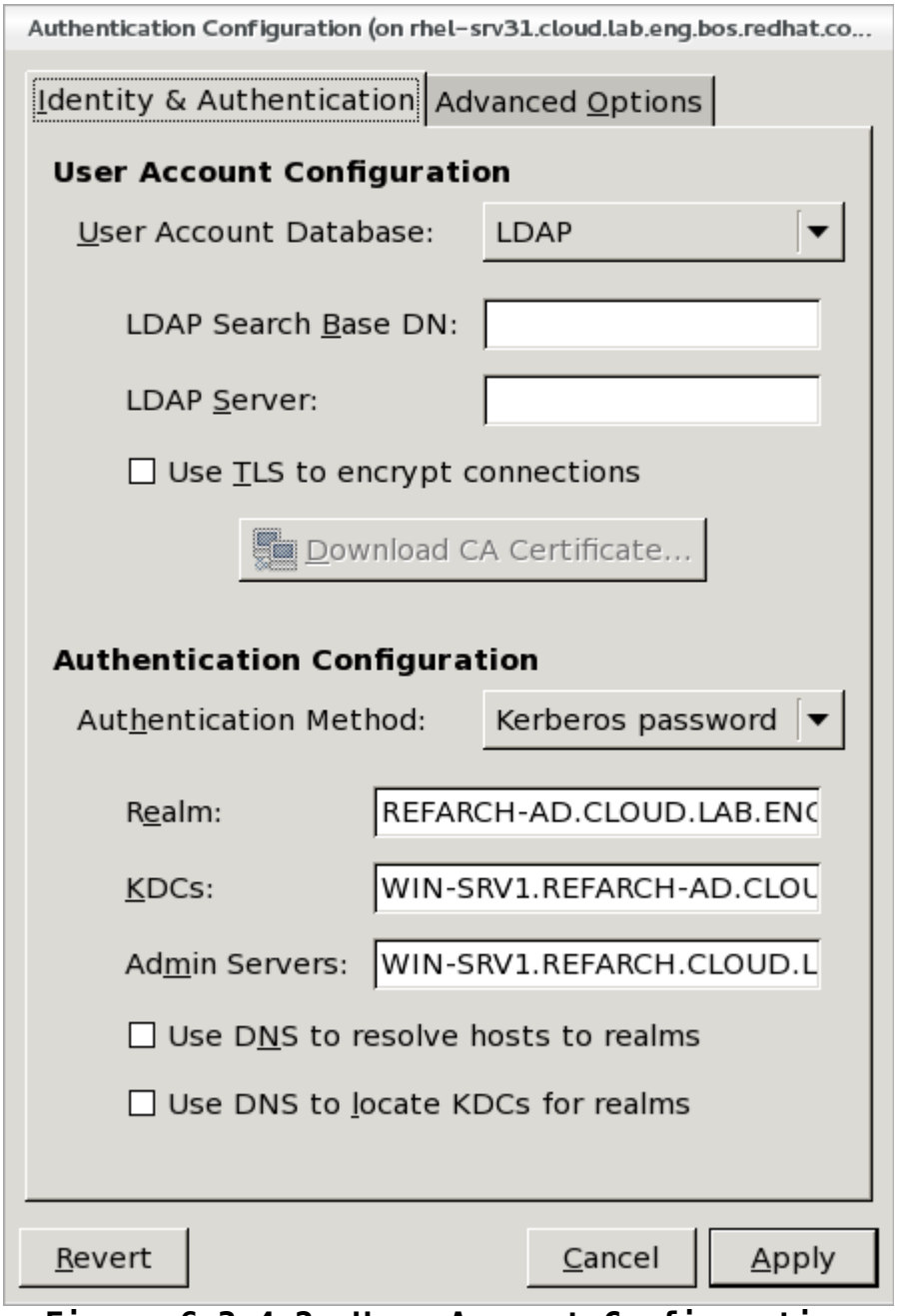

Figure 6.3.4-2: User Account Configuration

Select the **Advanced Options** tab when done.

Figure 6.3.4-3: Advanced Options

Select **Apply**. The terminal window indicated that the **oddjobd** daemon was started:

```
Starting oddjobd:
```

On first successful login to Active Directory, the **oddjobd** daemon calls a method to create a new home directory for a user.

2. Enable LDAP Searches

SSSD can do LDAP queries of Active Directory user information through either of the following methods:

- Anonymous LDAP binds
- SASL/GSSAPI binds with a service keytab acquired using Samba
- SASL/GSSAPI binds with a service keytab generated on a server in the AD domain

By default, anonymous binds are not enabled in Active Directory and often prohibited in many environments. The second approach requires the Samba client package and a properly configured Samba configuration. For these reasons, the third approach is utilized here.

Create a new computer object in Windows Active directory by performing the following steps on the Windows Server 2008 R2 server:

- Open the ***Active Directory Users and Computers*** snap-in:

  Start -> Administrative Tools -> Active Directory Users and Computers

- Create a new computer object:

  Expand '*refarch-ad.cloud.lab.eng.bos.redhat.com*'

  Right-click **Computers**, select **New** -> **Computer**

  Computer name: ***rhel-srv31***

- Select **OK**

Specify the NIS Domain and IP address for the new computer object:

- Select **Computers**
- Right-click '*rhel-srv31*', select **Properties**
- Under the **UNIX Attributes** tab:
  - NIS Domain: ***refarch-ad***
  - IP Address: ***10.16.142.31***
- Select **OK**

Next, open a command prompt to create and view a Kerberos service principal (*keytab*) for the newly created computer object with a randomly generated password:

```
C:\>setspn -A host/rhel-srv31.refarch-ad.cloud.lab.eng.bos.redhat.com
@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM rhel-srv31

Registering ServicePrincipalNames for CN=rhel-srv31,CN=Computers,DC=refarch
-ad,DC=cloud,DC=lab,DC=eng,DC=bos,DC=redhat,DC=com
        host/rhel-srv31.refarch-ad.cloud.lab.eng.bos.redhat.com@REFARCH-
AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
Updated object
```

```
C:\>setspn -L rhel-srv31
Registered ServicePrincipalNames for CN=rhel-srv31,CN=Computers,DC=refarch-
ad,DC=cloud,DC=lab,DC=eng,DC=bos,DC=redhat,DC
=com:
        host/rhel-srv31.refarch-ad.cloud.lab.eng.bos.redhat.com@REFARCH-
AD.CLOUD.LAB.ENG.BOS.REDHAT.COM

C:\>ktpass /princ host/rhel-srv31.refarch-ad.cloud.lab.eng.bos.redhat.com\
   @REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM /out rhel-srv31.keytab       \
   /crypto all /ptype KRB5_NT_PRINCIPAL -desonly                         \
   /mapuser REFARCH-AD\rhel-srv31$ +rndPass
Targeting domain controller:
    WIN-SRV1.refarch-ad.cloud.lab.eng.bos.redhat. com
Using legacy password setting method
Successfully mapped host/rhel-srv31.refarch-ad.cloud.lab.eng.bos.redhat.com
to RHEL-SRV31$.
WARNING: Account RHEL-SRV31$ is not a user account (uacflags=0x1021).
WARNING: Resetting RHEL-SRV31$'s password may cause authentication problems
if RHEL-SRV31$ is being used as a server.

Reset RHEL-SRV31$'s password [y/n]?  y
WARNING: pType and account type do not match. This might cause problems.
Key created.
                          ...output abbreviated...

keysize 130 host/rhel-srv31.refarch-
ad.cloud.lab.eng.bos.redhat.com@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
ptype 1 (KRB
5_NT_PRINCIPAL) vno 5 etype 0x11 (AES128-SHA1) keylength 16
(0x9b1d866a310e048961e1101ab5b381bc)

C:\>dir rhel-srv31.keytab
 Volume in drive C has no label.
 Volume Serial Number is A44B-73F4

 Directory of C:\

04/04/2012  11:33 AM               672 rhel-srv31.keytab
               1 File(s)            672 bytes
               0 Dir(s)  80,131,145,728 bytes free

C:\>
```

Securely transfer the *rhel-srv31.keytab* file over to the ***/root*** directory on the Red Hat Enterprise Linux 6 system using the secure method of choice (WinSCP, PSCP, etc.). Configure the keytab file ownership, permissions and move it into place:

Red Hat Enterprise Linux 6 system:

```
# chown root:root /root/rhel-srv31.keytab
# chmod 0600 /root/rhel-srv31.keytab
# mv /root/rhel-srv31.keytab /etc/krb5.keytab
```

After moving the file, restore the SELinux file context and confirm it:

```
# restorecon /etc/krb5.keytab
# ls -lZ /etc/krb5.keytab
-rw-------. root root unconfined_u:object_r:krb5_keytab_t:s0
/etc/krb5.keytab
```

Verify the keytab file can be read as follows:

```
# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)

# kinit -k -t /etc/krb5.keytab host/rhel-srv31.refarch-ad.cloud.lab.eng. \
bos.redhat.com@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM

# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: host/rhel-srv31.refarch-
ad.cloud.lab.eng.bos.redhat.com@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM

Valid starting     Expires            Service principal
04/04/12 11:55:58  04/04/12 21:55:52  krbtgt/REFARCH-
AD.CLOUD.LAB.ENG.BOS.REDHAT.COM@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
      renew until 04/11/12 11:55:58
```

Verify LDAP searches using `ldapsearch`:

```
# /usr/bin/ldapsearch -H                                          \
  ldap://win-srv1.REFARCH-AD.cloud.lab.eng.bos.redhat.com         \
  -Y GSSAPI -N -b                                                 \
  DC=refarch-ad,DC=cloud,DC=lab,DC=eng,DC=bos,DC=redhat,DC=com    \
  "(&(objectClass=user)(sAMAccountName=ad-user31))"               \
SASL/GSSAPI authentication started
SASL username: host/rhel-srv32.refarch-
ad.cloud.lab.eng.bos.redhat.com@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
SASL SSF: 56
SASL data security layer installed.
# extended LDIF
#
# LDAPv3
# base <DC=refarch-ad,DC=cloud,DC=lab,DC=eng,DC=bos,DC=redhat,DC=com> with
scope subtree
# filter: (&(objectClass=user)(sAMAccountName=ad-user31))
```

```
# requesting: ALL
#

# AD AD. User 31, Users, refarch-ad.cloud.lab.eng.bos.redhat.com

                          ...output abbreviated...

uid: ad-user31
msSFU30Name: ad-user31
msSFU30NisDomain: refarch-ad
uidNumber: 30000031
gidNumber: 10000002
unixHomeDirectory: /home/REFARCH-AD/ad-user31
loginShell: /bin/bash

                          ...output abbreviated...

# search result
search: 4
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

If `ldapsearch` is not available, install the **`openldap-clients`** package:

```
# yum install openldap-clients
Loaded plugins: product-id, refresh-packagekit, security, subscription-
manager
Updating certificate-based repositories.

Installed:
  openldap-clients.x86_64 0:2.4.23-20.el6
```

3. Modify SSSD Configuration

Next, LDAP parameters are modified within the SSSD configuration file. **Table 6.3.4: Summary of Changes – Configuration 3** provides a summary of the configuration file parameter changes:

Configuration 3 SSSD Configuration File Parameters	
Parameter	**Description**
access_provider = ldap	Set access control to LDAP
ldap_sasl_mech = GSSAPI	Set SASL mechanism to GSSAPI
ldap_sasl_authid = host/rhel-srv31.{FQDN} \ @{domain}.{FQDN}	Set SASL to use Kerberos principal
ldap_schema = rfc2307bis	Enable support for nested groups
ldap_user_object_class = user	Set user object class type
ldap_user_home_directory = unixHomeDirectory	Set user home directory to match directory specified in UNIX Attributes – (*e.g. - /home/REFARCH-AD/ad-user31)*
ldap_user_principal = userPrincipalName	Set LDAP user principal to Kerberos User Principal Name (UPN)
ldap_user_name = sAMAccountName	Enable user name compatibility with older Windows versions
ldap_group_object_class = group	Set group object class type
ldap_access_order = expire	Set access order to use value specified by ldap_account_expire_policy
ldap_account_expire_policy = ad	Set account expiration policy to AD
ldap_force_upper_case_realm = true	Set Kerberos realm to uppercase
ldap_disable_referrals = true	Disable LDAP referrals[1]

Table 6.3.4: Summary of Changes – Configuration 3

Make a safety copy of the SSSD configuration file:

```
# cp -p /etc/sssd/sssd.conf /etc/sssd/sssd.conf.back
```

[1]See Section **3.2.5 LDAP Referrals** for further details

Edit and save the SSSD configuration as follows – changes are highlighted in bold:

```
[sssd]
config_file_version = 2
domains = default
services = nss, pam
debug level = 0

[nss]

[pam]

[domain/default]
cache_credentials = true
enumerate = false

id_provider = ldap
auth_provider = krb5
chpass_provider = krb5
access_provider = ldap

ldap_sasl_mech = GSSAPI
ldap_sasl_authid = host/rhel-srv31.refarch-ad.cloud.lab.eng.bos.redhat  \
                   .com@REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM

ldap_schema = rfc2307bis

ldap_user_object_class = user
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_user_name = sAMAccountName

ldap_group_object_class = group

ldap_access_order = expire
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_disable_referrals = true

krb5_realm = REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM
```

Restart the SSSD daemon:

```
# service sssd restart
Stopping sssd:                                     [  OK  ]
Starting sssd:                                     [  OK  ]
```

6.3.5 Verification of Services

Verify the services provided by Configuration 3 by performing the tasks outlined in the following sections:

1. Login Access

```
# ssh ad-user31@rhel-srv31
ad-user31@rhel-srv31's password:
Creating home directory for ad-user31.

$ hostname
rhel-srv31.cloud.lab.eng.bos.redhat.com

$ id
uid=30000031(ad-user31) gid=10000002(rhel-users) groups=10000002(rhel-
users),10000003(Domain Users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user31
$ ls -ld
drwxr-xr-x. 4 ad-user31 rhel-users 4096 Apr  4 13:29 .

$ echo $SHELL
/bin/bash
```

Verify access from another Red Hat Enterprise Linux 6 system, using a different Active Directory user account:

```
# hostname
rhel-srv32.cloud.lab.eng.bos.redhat.com

# ssh ad-user32@rhel-srv31
ad-user32@rhel-srv31's password:
Creating home directory for ad-user32.

$ hostname
rhel-srv31.cloud.lab.eng.bos.redhat.com

$ id
uid=30000032(ad-user32) gid=10000002(rhel-users) groups=10000002(rhel-
users),10000003(Domain Users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user32
$ ls -ld
drwxr-xr-x. 4 ad-user32 rhel-users 4096 Apr  4 13:42 .

$ echo $SHELL
/bin/bash
```

This completes the process of integrating a Red Hat Enterprise Linux 6 system into an Active Directory domain using SSSD/Kerberos/LDAP. If there are multiple Red Hat Enterprise Linux 6 systems to be integrated, repeat the integration tasks for each system and verify the services provided.

6.4 Configuration 4 – Kerberos/LDAP

This is a legacy configuration for use in environments looking to integrate one or more Red Hat Enterprise Linux 6 systems into an Active Directory domain or forest without need or interest in the enhanced capabilities offered by SSSD. Login access is the only service provided.

6.4.1 Configuration Summary

Configuration 4 Kerberos/LDAP "Legacy"		
Components		
RHEL 6:	• Kerberos • LDAP	
Windows 2008 Server R2:	• Active Directory	
Services Provided	• Login access (RHEL command line, GUI access via AD credentials)	
Use Cases	• Environments looking to integrate one or more RHEL systems into an AD domain or forest without enhanced capabilities..	
Authentication (pam)	• Kerberos (pam_krb5)	
ID Tracking/ Name Resolution (nss)	• LDAP (nss_ldap)	
ID Mapping ("back-end")	• n/a	
Configuration Files	• /etc/nslcd.conf • /etc/krb5.conf	• /etc/pam.d/passwd-auth • /etc/pam.d/system-auth
Advantages	• Kerberos SSO capable (requires additional configuration work) • Client side caching (via nscd)	
Disadvantages	• "Legacy" approach to integration • No file sharing (but can be enabled by configuring Samba) • No off-line caching of user credentials • Poor management of ID collisions	
Notes	• This is considered a legacy configuration and is primarily used in environments where SSSD is not in use or of interest	

Table 6.4.1: Configuration Summary - Configuration 4

6.4.2 Systems Overview

Figure 6.4.2 provides a overview of the systems and services utilized by Configuration 4:

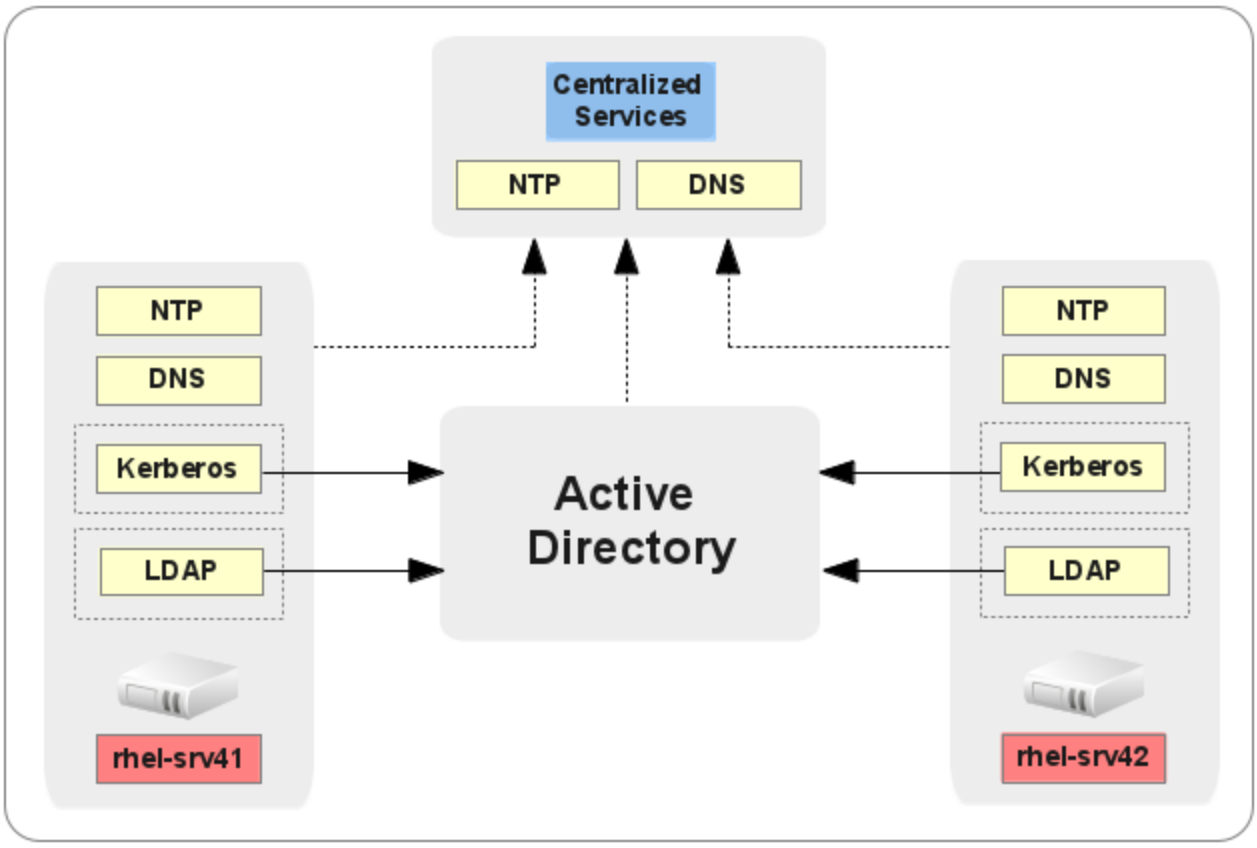

Figure 6.4.2: Systems Overview - Configuration 4

6.4.3 Authentication and ID Components

Figure 6.4.3 depicts the Authentication and ID Tracking for Configuration 4:

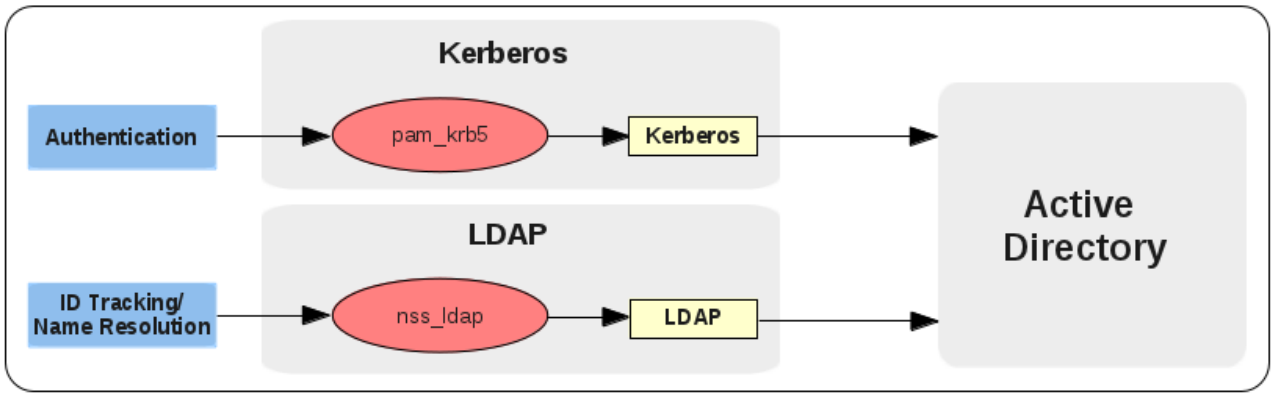

Figure 6.4.3: Authentication and ID Components - Configuration 4

6.4.4 Integration Tasks

Integrating Red Hat Enterprise Linux 6 into an Active Directory domain for Configuration 4 involves the following series of steps:

1. Configure Authentication
2. Modify LDAP Configuration
3. Add PAM Libraries
4. Modify NSS
5. Verify LDAP Queries
6. Enable NSS Caching

The following provides a step-by-step guide to the integration process:

1. Configure Authentication

The **`system-config-authentication`** tool simplifies configuring the Kerberos, LDAP security and authentication files for Active Directory integration. Invoke the tool as follows:

```
# system-config-authentication
```

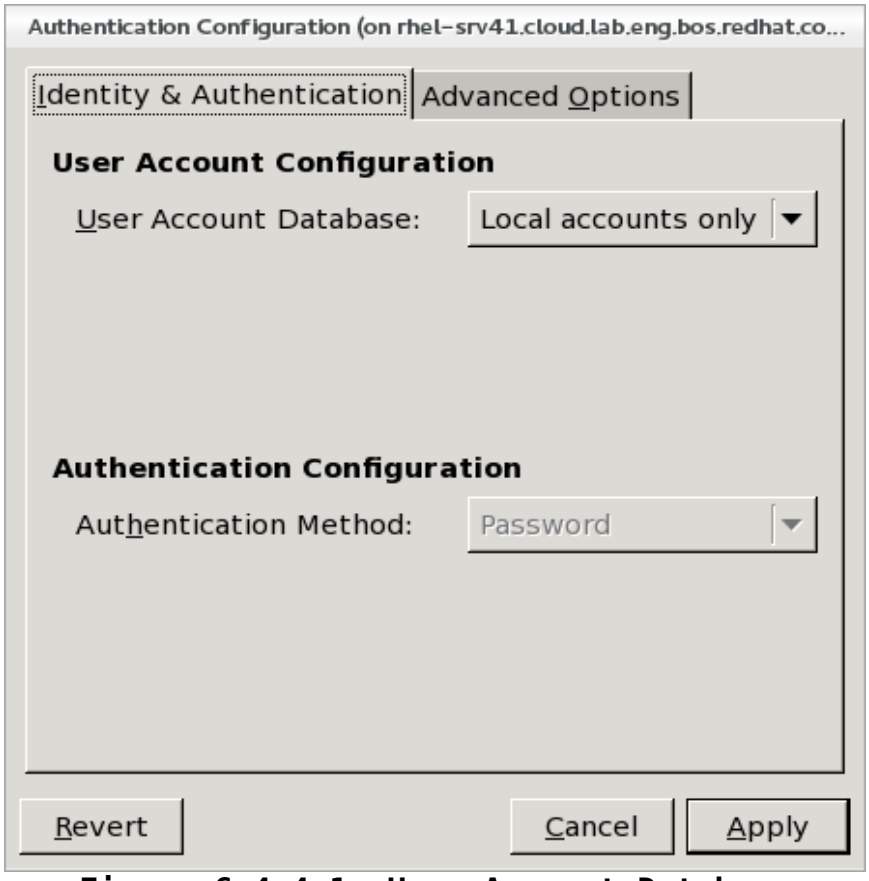

Figure 6.4.4-1: User Account Database

On the **Identity & Authentication** tab, select the **User Account Database** drop-down then select **LDAP**.

A new set of fields is displayed. Selecting the **LDAP** option allows the system to be configured to connect to the Windows Active Directory domain using LDAP with Kerberos authentication. User information from a domain can then be accessed and authenticated. The following server user account and authentication options can be configured:

- **LDAP Search Base DN**: Specifies that user information should be retrieved using the listed Distinguished Name (DN).
- **LDAP Server**: Specifies the address of the LDAP server.
- **Use TLS to encrypt connections**: When enabled, transport *Layer Security* (TLS) is used to encrypt passwords sent to the LDAP server.
    - The **Download CA Certificate** option allows a URL to be specified from which to download a valid *Certificate Authority* (CA) certificate. Valid CA certificates must be in the *Privacy Enhanced Mail* (PEM) format.
- **Authentication method**: Kerberos password - this option enables Kerberos authentication. The following options are available:
    - **Realm**: Configures the realm for the Kerberos server. The realm is the network that users Kerberos, comprised of one or more KDCs and Kerberos clients.
    - **KDCs**: Specifies the *Key Distribution Center* (KDC) for issuing Kerberos tickets.
    - **Admin Servers**: Specifies the administration server(s) running **kadmind**.
- **Use DNS to resolve hosts to realms**: Allows the use of DNS to find Kerberos realms.
- **Use DNS to locate KDCs for realms**: Allows the use of DNS to find Kerberos KDCs.

If Kerberos has been properly configured and verified as per the deployment prerequisites outlined in **Section 5.8 Install/Configure Kerberos Client** the screen appears as seen on the following page.

Populate the fields as follows:

User Account Database:	**LDAP**
LDAP Search DN:	**dc=refarch-ad,dc=cloud,dc=lab,dc=eng,dc=bos,dc=redhat,dc=com**
LDAP Server:	**ldap://WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Realm:	**REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
KDCs:	**WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Admin Servers:	**WIN-SRV1.REFARCH-AD.CLOUD.LAB.ENG.BOS.REDHAT.COM**

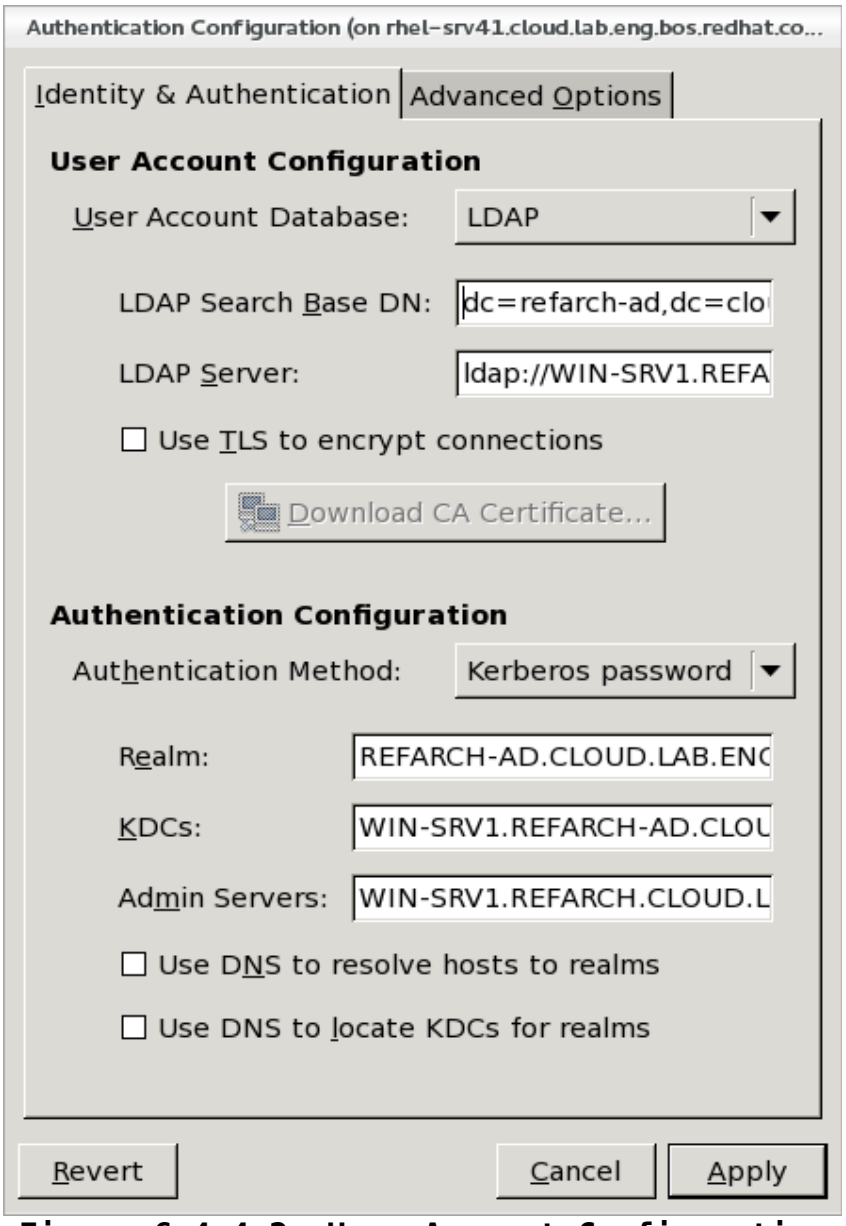

Figure 6.4.4-2: User Account Configuration

Select the **Advanced Options** tab when done.

Figure 6.4.4-3: Advanced Options

Select **Apply**. The terminal window indicated that the **oddjobd** daemon was started:

```
Starting oddjobd:
```

On first successful login to Active Directory, the **oddjobd** daemon calls a method to create a new home directory for a user.

2. Modify LDAP Configuration

Ensure the **nss-pam-ldapd** package is installed:

```
# yum install nss-pam-ldapd
```

Make a safety copy of the LDAP configuration file:

```
# cp -p /etc/nslcd.conf /etc/nslcd.conf.back
```

Edit and save the LDAP configuration as follows – changes are highlighted in bold:

```
#------------------------------------------------------#
# The distinguished name to bind to the AD server with: #
#------------------------------------------------------#
binddn cn=AD LDAP-Bind,cn=users,dc=refarch-ad,dc=cloud,dc=lab, \
dc=eng,dc=bos,dc=redhat,dc=com
bindpw LDAPBind!!

# Mappings for Active Directory
pagesize 1000
referrals off
filter passwd (&(objectClass=user)(!(objectClass=computer))(uidNumber=*)
(unixHomeDirectory=*))
map    passwd uid              sAMAccountName
map    passwd homeDirectory    unixHomeDirectory
map    passwd gecos            displayName
filter shadow (&(objectClass=user)(!(objectClass=computer))(uidNumber=*)
(unixHomeDirectory=*))
map    shadow uid              sAMAccountName
map    shadow shadowLastChange pwdLastSet
filter group  (objectClass=group)
map    group  uniqueMember     member

uid nslcd
gid ldap

uri ldap://win-srv1.refarch-ad.cloud.lab.eng.bos.redhat.com
base dc=refarch-ad,dc=cloud,dc=lab,dc=eng,dc=bos,dc=redhat,dc=com
```

Restart the LDAP daemon and enable it to start on boot:

```
# service nslcd restart
# chkconfig nslcd on
```

3. Add PAM Libraries

Edit */etc/pam.d/system-auth* and add the following **pam_krb5.so** library entries. Changes are highlighted and note that ordering is important:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
#auth        sufficient    pam_sss.so use_first_pass
auth        required      pam_deny.so
auth        sufficient    pam_krb5.so use_first_pass

account     required      pam_unix.so broken_shadow
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
#account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required      pam_permit.so
account     [default=bad success=ok user_unknown=ignore] pam_krb5.so

password    requisite     pam_cracklib.so try_first_pass retry=3 type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass
use_authtok
#password    sufficient    pam_sss.so use_authtok
password    required      pam_deny.so
password    sufficient    pam_krb5.so use_authtok

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     optional      pam_oddjob_mkhomedir.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond
quiet use_uid
session     required      pam_unix.so
#session     optional      pam_sss.so
session     optional      pam_krb5.so
```

Note: Entries for the SSSD libraries (**pam_sss.so**) must be disabled by either removing them or commenting them out as done here.

Edit */etc/pam.d/password-auth* and add the following **pam_krb5.so** library entries. Changes are highlighted and note that ordering is important:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
#auth        sufficient    pam_sss.so use_first_pass
auth        sufficient    pam_krb5.so use_first_pass
auth        required      pam_deny.so

account     required      pam_unix.so broken_shadow
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
#account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     [default=bad success=ok user_unknown=ignore] pam_krb5.so
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3 type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass
use_authtok
#password    sufficient    pam_sss.so use_authtok
password    sufficient    pam_krb5.so use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     optional      pam_oddjob_mkhomedir.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond
quiet use_uid
session     required      pam_unix.so
#session     optional      pam_sss.so
session     optional      pam_krb5.so
```

The entries for the SSSD libraries (**pam_sss.so**) must be disabled here as well by either removing them or commenting them out.

4. Modify NSS

Edit */etc/nsswitch.conf* and change the **passwd**, **shadow,** and **group** entries from SSSD (***sss***) to LDAP (***ldap***):

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
```

5. Verify LDAP Queries

Confirm that LDAP user lookups to Active Directory are working correctly:

```
# id ad-user41
uid=40000041(ad-user41) gid=10000002(rhel-users) groups=10000003(Domain
Users),10000002(rhel-users)

# getent passwd ad-user41
ad-user41:*:40000041:10000002:AD AD. User 41:/home/REFARCH-AD/ad-
user41:/bin/bash
```

6. Enable NSS Caching

NSS caching stores previously authenticated Active Directory user sessions on the local Red Hat Enterprise Linux system. Start the **nscd** service and configure it to start on boot:

```
# service nscd start
Starting nscd:                                             [  OK  ]
# chkconfig nscd on
```

6.4.5 Verification of Services

Verify the services provided by Configuration 4 by performing the tasks outlined in the following sections:

1. Login Access

```
# ssh ad-user41@rhel-srv41
ad-user41@rhel-srv41's password:
Creating home directory for ad-user41.

$ hostname
rhel-srv41.cloud.lab.eng.bos.redhat.com

$ id
uid=40000041(ad-user41) gid=10000002(rhel-users) groups=10000002(rhel-
users),10000003(Domain Users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user41
$ ls -ld
drwxr-xr-x. 4 ad-user41 rhel-users 4096 Apr  5 18:39 .

$ echo $SHELL
/bin/bash
```

Verify access from another Red Hat Enterprise Linux 6 system, using a different Active Directory user account:

```
# hostname
rhel-srv42.cloud.lab.eng.bos.redhat.com

# ssh ad-user42@rhel-srv41
ad-user42@rhel-srv41's password:
Creating home directory for ad-user42.

$ hostname
rhel-srv41.cloud.lab.eng.bos.redhat.com

$ id
uid=40000042(ad-user42) gid=10000002(rhel-users) groups=10000002(rhel-
users),10000003(Domain Users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ pwd
/home/REFARCH-AD/ad-user42
$ ls -ld
drwxr-xr-x. 4 ad-user42 rhel-users 4096 Apr  5 19:11 .

$ echo $SHELL
/bin/bash
```

This completes the process of integrating a Red Hat Enterprise Linux 6 system into an Active Directory domain using Kerberos/LDAP. If there are multiple Red Hat Enterprise Linux 6 systems to be integrated, repeat the integration tasks for each system and verify the services provided.

7 Conclusion

This reference architecture details the components, considerations and configurations available for selecting, deploying, and integrating Red Hat Enterprise Linux 6 systems into Windows Active Directory domains. Basic concepts are introduced, deployment and integration tasks outlined, best practices and guidelines provided. The configuration selection process is simplified through the use of a decision tree that guides the reader towards a focused set of recommended configurations.

These configurations can be deployed as presented here, or customized to meet the specific requirements of system administrators needing to integrate Red Hat Enterprise Linux 6 systems into their existing Microsoft Windows Active Directory domain environments.

Appendix A: References

Red Hat Enterprise Linux 6

1. Red Hat Enterprise Linux 6 Installation Guide
   Installing Red Hat Enterprise Linux 6 for all architectures
   Edition 1.0
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Installation_Guide/Red_Hat_Enterprise_Linux-6-Installation_Guide-en-US.pdf
2. Red Hat Enterprise Linux 6 Deployment Guide
   Deployment, Configuration and Administration of Red Hat Enterprise Linux 6
   Edition 1.0
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Deployment_Guide/Red_Hat_Enterprise_Linux-6-Deployment_Guide-en-US.pdf
3. Red Hat Enterprise Linux 6 Virtualization Getting Started Guide
   Virtualization Documentation
   Edition 0.2
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Virtualization_Getting_Started_Guide/Red_Hat_Enterprise_Linux-6-Virtualization_Getting_Started_Guide-en-US.pdf
4. Red Hat Enterprise Linux 6 Virtualization Administration Guide
   Virtualization Documentation
   Edition 0.1
   http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Virtualization_Administration_Guide/Red_Hat_Enterprise_Linux-6-Virtualization_Administration_Guide-en-US.pdf

Microsoft Windows Server 2008 R2

5. Install and Deploy Windows Server
   August 6, 2009
   http://technet.microsoft.com/en-us/library/dd283085.aspx

Samba/Winbind

6. The Official Samba 3.5 HOWTO and Reference Guide
   http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/
7. “What steps do I need to follow to join a Red Hat Enterprise Linux Samba server to an Active Directory domain in security = ADS mode?”
   Red Hat Knowledge Article - 3049
   http://access.redhat.com/knowledge/articles/DOC-3049
8. “How do I set up winbind on our Samba server to create users and groups from our domain controller?”
   Red Hat Knowledge Article - 4821
   http://access.redhat.com/knowledge/articles/DOC-4821
9. “How do I configure Kerberos for Active Directory (AD) integration on Linux?”
   Red Hat Knowledge Solution - 4734
   http://access.redhat.com/knowledge/solutions/DOC-4734

10. “What changes do I need to make to nsswitch.conf for winbind to work?”
Red Hat Knowledge Article - 4761
http://access.redhat.com/knowledge/articles/DOC-4761

Active Directory

11. Active Directory Domain Services
April 18, 2008
http://technet.microsoft.com/en-us/library/cc770946.aspx
12. Active Directory Lightweight Directory Services
August 18, 2008
http://technet.microsoft.com/en-us/library/cc731868.aspx

Configuration 1 – Samba/Winbind w/idmap_rid

13. “How can I configure winbind to synchronize use and group id's across multiple Red Hat Enterprise Linux hosts on Active Directory accounts?”
Red Hat Knowledge Article - 2341
https://access.redhat.com/knowledge/articles/2341

Configuration 2 - Samba/Winbind w/idmap_ad

14. “The 'wbinfo -i' command does not resolve user info”
Red Hat Knowledge Solution - 67701
https://access.redhat.com/knowledge/solutions/67701

Configuration 3 – SSSD/Kerberos/LDAP

15. “Configuring SSSD to authenticate with a Windows 2008 Domain Server”
https://fedorahosted.org/sssd/wiki/Configuring sssd to authenticate with a Windows 2008 Domain Server

Configuration 4 – Kerberos/LDAP

16. “How do I authenticate RHEL6 to Windows 2008 R2 system using LDAP and kerberos?”
Red Hat Knowledge Solution - 43646
https://access.redhat.com/knowledge/solutions/43646
17. “Deploying Single Sign-On for Red Hat Enterprise Linux 5 in an existing Active Directory environment”
Red Hat Tech Brief Article
https://access.redhat.com/knowledge/techbriefs/deploying-single-sign-red-hat-enterprise-linux-5-existing-active-directory-environment

Appendix B: Glossary

A

ACL (Access Control List)

A method for controlling access to files, directories on a file system.

Access Rights

In the context of access control, specify the level of access granted or denied. Access rights are related to the type of operation that can be performed on the directory. The following rights can be granted or denied: read, write, add, delete, search, compare, selfwrite, proxy and all.

AD (Active Directory)

A suite of directory services developed by Microsoft and based on Novell eDirectory. Active Directory utilizes a number of standardized protocols (*DNS, LDAP, Kerberos*). Active Directory provides a scalable, centralized database infrastructure for securely managing objects (*users, systems, groups, printers, applications*). Directory objects are stored in a hierarchy consisting of *nodes, trees, forests* and *domains*.

Active Directory services include Active Directory Certificate Services (AD CS), Active Directory Domain Services (AD DS), Active Directory Federation Services (AD FS), Active Directory Lightweight Directory Services (AD LDS), and Active Directory Rights Management Services (AD RMS)

AD DS (Active Directory Domain Services)

An update to Active Directory (AD) introduced in Windows Server 2008 R2. Active Directory Domain Services is included in Windows Server 2008 R2 and is activated as a Server Role.

A Record (Address Record)

A DNS record used to point domain and host names to a static IP address.

authconfig

A command line tool under Red Hat Enterprise Linux for simplifying the configuration of user identity and system authentication services. A graphical equivalent (*authconfig-gtk*) is also available.

authentication

The process of proving the identity of a user or client in order to grant access to a server resource.

B

Builtin

One of the default containers defined when an Active Directory domain is created. The Builtin container defines user groups that are within the local scope of the domain - Account Operators, Administrators, Backup Operators, Guests, Network Configuration Operators, etc.

C

CA (Certificate Authority)

Company or organization that sells and issues authentication certificates that are known, trusted.

CIFS (Common Internet File System)

A standard file and print sharing system for Microsoft clients in Windows server environments. CIFS uses the SMB (Server Message Block) protocol for client to server communications.

D

daemon

A UNIX/Linux program that runs without human intervention to perform a given task. For example, ***smbd*** is the Samba server daemon that provides file sharing and print services to clients.

dcdiag (Domain Controller Diagnostics)

A command line diagnostics tool that provides a framework for running a series of tests to verify various components of a server in an Active Directory domain. The dcdiag tool is included with Windows Server 2008 R2 and is available for earlier versions of Windows Server.

Directory Service

A database application that provides a structure for organizing and managing common network objects into collections of name-value mappings. Commonly used directory services include Active Directory (AD), Domain Name System (DNS), Lightweight Directory Access protocol (LDAP), Network Information Service (NIS), OpenLDAP and Red Hat Directory Server (RDS).

Distinguished Name (DN)

The String representation of an entry's name and location in an LDAP directory.

DNS (Domain Name System)

A hierarchical, distributed naming system for managing the mappings of human-friendly domain, host and service names to IP addresses. DNS also defines the protocol for communication exchanges in DNS as part of the Internet Protocol (IP) suite.

Domain
A human-friendly name for an IP address representing a collection of computer and network IP addresses.

Domain Realm
The name of the Active Directory domain.

E

Enumeration

Enumeration is the process of listing the users, groups in an Active Directory domain.

F

Forest

A forest is a collection of trees that share a common global catalog, directory schema, logical structure, and directory configuration. The forest represents the security boundary within which users, computers, groups, and other objects are accessible.

Forward Lookup Zone

A FQDN mapped to an IP Address under Active Directory Domain Services.

FQDN (Fully Qualified Domain Name)

A domain name specifying an exact location in a DNS hierarchical tree. For example, a host named *rhel-srv1* that resides in the *bos.redhat.com* domain, has the fully qualified domain name (FQDN) of *rhel-srv1.bos.redhat.com.*

G

GID (Group ID)

A numeric value assigned to represent a group of UNIX/Linux users. Groups identify user access to system resources and membership is managed through entries in the file */etc/group*.

GSS-API

Generic Security Services. The generic access protocol that is the native way for UNIX-based systems to access and authenticate Kerberos services; also supports session encryption.

H

Hostname

The name for a host in the form *host.domain.com*, which is translated into an IP address. For example, *rhel-srv1.bos.redhat.com* is the machine *rhel-srv1* in the subdomain *bos* and *redhat.com* domain.

I

IdM (Identity Management)

IdM is Red Hat's implementation of IPA. IdM provides a way to create identity stores, centralized authentication, domain control for Kerberos and DNS services, and authorization policies on Linux systems Identity Management provides a unifying skin for standards-defined, common network services, including PAM, LDAP, Kerberos, DNS, NTP, and certificate services, and it allows Red Hat Enterprise Linux systems to serve as the domain controllers. Identity Management defines a domain, with servers and clients who share centrally-managed services, like Kerberos and DNS.

ID Mappings

The mapping of UID, GID on the Red Hat Enterprise Linux 6 system to SID in a Windows Active Directory domain.

IMU (Identity Management for UNIX)

An additional role service that enables Red Hat Enterprise Linux 6 system to integrate with Active Directory. IMU is standard on Windows Server 2008 R2 and replaced the optional Services For UNIX (SFU) product on earlier Windows Server versions.

IPA (Identity, Policy and Management)

An open source project for providing centralized, secure user identity management and authorization policies.

IP Address (Internet Protocol Address)

A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 198.93.93.10).

K

KDC (Key Distribution Center)

The Kerberos database server that manages the secure database of secret keys used for Kerberos trusted authentication. Kerberos clients request a “ticket” from the KDC that has a configurable expiration date and must be renewed on a regular basis.

Kerberos

A network authentication protocol developed at the Massachusetts Institute of Technology (MIT). Kerberos uses strong cryptography to provide highly secure Single Sign-On (SSO) capabilities between client and server applications.

LDAP (Lightweight Directory Access Protocol)

L A hierarchical directory service and an application protocol for performing lookups and updates to a remote directory service. LDAP data is transmitted securely over networks via SSL, TLS or SASL. Common LDAP implementations include Active Directory, Apache Directory Server, OpenLDAP, Oracle Internet Directory and Red Hat Directory Server.

LDIF (LDAP Data Interchange Format)

Format used to represent LDAP server entries in text form.

N **NetBIOS (Network Basic Input/Output System)**

An older protocol for providing OSI model session layer services. NetBIOS provides three services – *Name service* (for name registration and resolution), *Session service* (for connection-oriented communications) and *Datagram service* (for connection-less communications).

NIS (Network Information Service)

A client-server directory service protocol developed by Sun Microsystems. NIS provides a method to distribute and share configuration data such as users, hosts and files between networked systems. Due to scalability and security concerns, the function of NIS has been largely replaced by LDAP.

nmb

A daemon included in the Samba suite for providing NetBIOS name services.

ns-slapd

Red Hat's LDAP Directory Server daemon or service that is responsible for all actions of the Directory Server

NSS (Name Service Switch)

A UNIX/Linux facility that manages a variety of common configuration databases and name resolution sources - */etc/passwd*, */etc/group*, */etc/hosts*, *DNS*, *NIS*, *LDAP*, etc.

NTP (Network Time Protocol)

An Internet protocol used to synchronize computer system clocks to a time reference. On Red Hat Enterprise Linux the ntpd daemon handles the actual synchronization.

O **Objects**

An Active Directory object represents a single entity—whether a user, a computer, a printer, or a group—and its attributes. Objects fall into two general categories: resources (e.g., printers) and security principals (user or computer accounts and groups). Security principals are assigned unique security identifiers (SIDs).

P **PAM (Pluggable Authentication Modules)**

A set of libraries that handle the authentication tasks of applications.

Permissions

In the context of access control, permissions define whether access to the directory information is granted or denied and the level of access that is granted or denied.

Principal

A user or computer in a Kerberos realm. Principals are stored in the Kerberos authentication database.

Protocol

A set of rules that describes how devices on a network exchange information.

R

Realm

A collection of Kerberos principals. In the Kerberos configuration file, the realm includes the name of the KDC and the administration server.

Replication

Act of copying data between servers.

Reverse Lookup Zone

An IP Address mapped to a FQDN under Active Directory Domain Services.

RHDS (Red Hat Directory Server)

An LDAP-compliant server that centralizes user identity and application information. RHDS provides an operating system-independent, network-based registry for storing application settings, user profiles, group data, policies and access control information.

RID (Relative IDentifier)

The portion of the security identifier (SID) that identifies a user or group in relation to the authority that issued the SID.

RFC (Request For Comments)

Procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

Root User

The most privileged user available on Unix machines. The root user has complete access privileges to all files on the machine.

S

Samba

A suite of programs that provide seamless file and print services for Windows clients in Linux environments. Like CIFS, Samba uses the SMB protocol for client to server communications.

SASL (Simple Authentication and Security Layer)

An authentication framework for clients as they attempt to bind to a directory.

Schema

Definitions describing what types of information can be stored as entries in a database or directory service. When information that does not match the schema is stored in the directory, clients attempting to access the directory may be unable to display the proper results.

SELinux (Security-Enhanced Linux)

A flexible, mandatory access control architecture that provides support for the enforcement of access control policies.

Service

A background process on a machine that is responsible for a particular system task. Service processes do not need human intervention to continue functioning.

SID (Security IDentifier)

A data structure for identifying user, group and system accounts in Windows operating system environments.

slapd

LDAP Directory Server daemon or service that is responsible for most functions of a directory except replication.

smb

A server daemon included in the Samba suite for providing file sharing and print services to clients.

SSL (Secure Sockets Layer)

A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

SSSD (System Security Services Daemon)

SSSD contains a set of daemons to manage access to remote directories and authentication mechanisms. It provides PAM (authentication) and NSS (name resolution) modules, a pluggable backend to connect to multiple different account sources and a D-BUS based interface. It is also the basis to provide client auditing and policy services for IdM.

system-config-authentication

A graphical tool (symbolic link to authconfig-gtk) under Red Hat Enterprise Linux for simplifying the configuration of user identity and system authentication services. A command line equivalent (*authconfig*) is also available.

T

TCP/IP (Transmission Control Protocol/Internet Protocol)

A standard set of communications protocols organized into four layers – *Application*, *Transport*, *Internet* and *Link*. TCP/IP is the most commonly deployed protocols for computer and network communications.

TLS (Transport Layer Security)

The standard for secure socket layers (SSL); a public key based protocol.

Tree

A tree is a collection of one or more Active Directory domains and domain trees in a contiguous namespace, linked in a transitive trust hierarchy.

Trust

A method to allow users in one Active Directory domain to access the resources in another domain. Under Active Directory a variety of trusts types can be configured:

One-way: One domain allows access to users on another domain, but the other domain does not allow access to users on the first domain.

Two-way: Two domains allow access to users on both domains.

Transitive: A trust that can extend beyond two domains to other trusted domains in the forest.

Intransitive: A one way trust that does not extend beyond two domains.

Explicit: A trust that an admin creates. It is not transitive and is one way only.

Cross-link: An explicit trust between domains in different trees or in the same tree when a descendant/ancestor (child/parent) relationship does not exist between the two domains.

U

UID (User ID)

A numeric value assigned to represent a UNIX/Linux user. A UID identifes user access to system resources and is managed through an entry in the file */etc/passwd*.

URL (Uniform Resource Locator)

Uniform Resource Locator. The addressing system used by the server and the client to request documents. It is often called a location. The URL format is *protocol*://*machine*:*port*/*document*. The port number is necessary only on selected servers, and it is often assigned by the server, freeing the user of having to place it in the URL.

W

winbindd

A daemon included in the Samba suite for providing unified user logons. Winbind handles the *Authentication* of user credentials, *Identity Management* and *Mappings* of UID, GID on the Red Hat Enterprise Linux 6 system to SID in a Windows Active Directory domain.

Winsync

Winsync is a utility for synchronizing user data and passwords between Red Hat Enterprise Linux 6 hosts running RHDS or IdM and Windows servers. Winsync supports the 32-bit and 64-bit versions of Windows Server 2003 and 2008.

Appendix C: Winbind Backend Reference

idmap_tdb

Description

The idmap_tdb plugin is the default backend used by winbindd for storing ID mapping tables (UID/GID <-> SID).

In contrast to read only backends like idmap_rid, it is an allocating backend: This means that it needs to allocate new user and group IDs in order to create new mappings. The allocator can be provided by the idmap_tdb backend itself or by any other allocating backend like idmap_ldap or idmap_tdb2. This is configured with the parameter `idmap alloc backend`.

Note that in order for this (or any other allocating) backend to function at all, the default backend needs to be writeable. The ranges used for uid and gid allocation are the default ranges configured by "idmap uid" and "idmap gid".

Options

`range = low - high`

Defines the available matching UID and GID range for which the backend is authoritative.

Example

This example shows how tdb is used as a the default idmap backend. This configured range is used for UID and GID allocation:

```
[global]
        # "backend = tdb" is redundant here since it is the default
        idmap config * : backend = tdb
        idmap config * : range   = 1000000-2000000
```

idmap_rid

Description

The idmap_rid backend provides a way to use an algorithmic mapping scheme to map UIDs/GIDs and SIDs. No database is required as the mapping is deterministic.

Options

range = low - high

Defines the available matching uid and gid range for which the backend is authoritative. Note that the range acts as a filter. If any of the algorithmically determined UID or GID fall outside the range are ignored and the corresponding map is discarded. It is intended as a way to avoid accidental UID/GID overlaps between local and remotely defined IDs.

Example

This example shows how to configure the default id mapping with tdb and two domains with idmap_rid - a principal domain (***MAIN***) and a trusted domain (***TRUSTED***). Also shown is the use of the base_rid parameter for the trusted domain (***TRUSTED***):

```
[global]
        security = domain
        workgroup = MAIN

        idmap config * : backend = tdb
        idmap config * : range   = 1000000-1999999

        idmap config MAIN : backend = rid
        idmap config MAIN : range   = 10000 - 49999

        idmap config TRUSTED : backend  = rid
        idmap config TRUSTED : range    = 50000 - 99999
        idmap config TRUSTED : base_rid = 1000
```

idmap_ad

Description

The idmap_ad plugin provides a way for Winbind to read id mappings from an AD server that uses RFC2307/SFU schema extensions. This is a *read-only* module that only implements the "idmap" API. Mappings must be provided in advance by the administrator by adding the posixAccount/posixGroup classes and relative attribute/value pairs to the user and group objects in the AD.

Note that the idmap_ad module has changed considerably since Samba versions 3.0 and 3.2. Currently, the ad backend does not work as the the default idmap backend, but one has to configure it separately for each domain for which one wants to use it, using disjoint ranges. One usually needs to configure a writeable default idmap range, using for example the tdb or ldap backend, in order to be able to map the BUILTIN sids and possibly other trusted domains. The writeable default config is also needed in order to be able to create group mappings. This default idmap configuration should have a range that is disjoint from any explicitly configured domain with idmap backend ad. See the example below.

Options

range = low - high

Defines the available matching UID and GID range for which the backend is authoritative. The range specified acts as a filter so that any UID or GID stored in AD that fall outside the range is ignored and the corresponding map is discarded. It is intended as a way to avoid accidental UID/GID overlaps between local and remotely defined IDs.

schema_mode = <rfc2307 | sfu >

Defines the schema that idmap_ad should use when querying Active Directory for user and group information. This can be either the RFC2307 schema support included in Windows 2003 R2 or the Service for Unix (SFU) schema.

Example

The following example shows how to retrieve id mappings from a principal and trusted AD domains. If trusted domains are present id conflicts must be resolved beforehand, there is no guarantee on the order conflicting mappings would be resolved at this point. This example also shows how to leave a small non conflicting range for local id allocation that may be used in internal backend(s) like BUILTIN:

```
[global]

        idmap config * : backend = tdb
        idmap config * : range   = 1000000-1999999

        idmap config CORP : backend = ad
        idmap config CORP : range   = 1000-999999
```

idmap_ldap

Description

The idmap_ldap plugin provides a means for Winbind to store and retrieve ID mapping tables (UID/GID <-> SID) in an LDAP directory service.

In contrast to read only backends like idmap_rid, it is an allocating backend: This means that it needs to allocate new user and group IDs in order to create new mappings. The allocator can be provided by the idmap_ldap backend itself or by any other allocating backend like idmap_tdb or idmap_tdb2. This is configured with the parameter `idmap alloc backend`.

Options

ldap_base_dn = DN

Defines the directory base suffix to use for UID/GID/SID mapping entries. If not defined, idmap_ldap defaults to using the "ldap idmap suffix" option from smb.conf.

ldap_user_dn = DN

Defines the user DN to be used for authentication. The secret for authenticating this user should be stored with net idmap secret. If absent, the ldap credentials from the ldap passdb configuration are used, and if these are also absent, an anonymous bind is performed as a last fallback.

ldap_url = ldap://server/

Specifies the LDAP server to use for SID/UID/GID map entries. If not defined, idmap_ldap assumes that ldap://localhost/ should be used.

range = low - high

Defines the available matching uid and gid range for which the backend is authoritative.

Examples

The following example shows how an ldap directory is used as the default idmap backend. It also configures the idmap range and base directory suffix. The secret for the ldap_user_dn has to be set with "net idmap secret ´*´ password":

```
[global]
        idmap config * : backend      = ldap
        idmap config * : range        = 1000000-1999999
        idmap config * : ldap_url     = ldap://localhost/
        idmap config * : ldap_base_dn = ou=idmap,dc=example,dc=com
        idmap config * : ldap_user_dn = cn=idmap_admin,dc=example,dc=com
```

This example shows how ldap can be used as a read-only backend while tdb is the default backend used to store the mappings. It adds an explicit configuration for some domain DOM1, that uses the ldap idmap backend. Note that a range disjoint from the default range is used:

```
[global]
        # "backend = tdb" is redundant here since it is the default
        idmap config * : backend = tdb
        idmap config * : range   = 1000000-1999999

        idmap config DOM1 : backend      = ldap
        idmap config DOM1 : range        = 2000000-2999999
        idmap config DOM1 : read only    = yes
        idmap config DOM1 : ldap_url     = ldap://server/
        idmap config DOM1 : ldap_base_dn = ou=idmap,dc=dom1,        \
                                           dc=example,dc=com
        idmap config DOM1 : ldap_user_dn = cn=idmap_admin,dc=dom1,  \
                                 dc=example,dc=com
```

In order to use authentication against ldap servers, it may be necessary to provide a DN and a password. To avoid exposing the password as plain text in the configuration file, it is stored into a security store. The "net idmap " command is used to store a secret for the DN specified in a specific idmap domain.

idmap_hash

Description

The idmap_hash plugin provides similar support as the idmap_rid module. However, UID's and GID's are generated from the full domain SID using a hashing algorithm that maps the lower 19 bits from the user or group RID to bits 0 - 19 in the Unix ID and hashes 96 bits from the Active Directory domain SID to bits 20 - 30 in the Unix id. The result is a 31 bit UID or GID that is consistent across machines and provides support for trusted domains.

This plugin also implements the nss_info API and can be used to support a local name mapping files if enabled via the "winbind normalize names" and "winbind nss info" parameters in smb.conf.

Options

name_map

Specifies the absolute path to the name mapping file used by the nss_info API. Entries in the file are of the form "unix name = qualified domain name". Mapping of both user and group names is supported.

Example

The following example utilizes the idmap_hash plugin for the idmap and nss_info information:

```
[global]

        idmap config * : backend = hash
        idmap config * : range   = 1000-4000000000

        winbind nss info         = hash
        winbind normalize names  = yes
        idmap_hash:name_map      = /etc/samba/name_map.cfg
```

idmap_tdb2

Description

The idmap_tdb2 plugin is a substitute for the default idmap_tdb backend used by winbindd for storing SID/UID/GID mapping tables in clustered environments with Samba and CTDB.

In contrast to read only backends like idmap_rid, it is an allocating backend: This means that it needs to allocate new user and group IDs in order to create new mappings. The allocator can be provided by the idmap_tdb2 backend itself or by any other allocating backend like idmap_tdb or idmap_ldap. This is configured with the parameter `idmap alloc backend`.

Options

range = low - high

Defines the available matching uid and gid range for which the backend is authoritative.

script

This option can be used to configure an external program for performing id mappings instead of using the tdb counter. The mappings are then stored into the tdb2 idmap database.

Example

This example shows how tdb2 is used as a the default idmap backend.

```
[global]
        idmap config * : backend = tdb2
        idmap config * : range   = 1000000-2000000
```

idmap_nss

Description

The idmap_nss plugin provides a means to map Unix users and groups to Windows accounts and obsoletes the "winbind trusted domains only" smb.conf option. This backend provides a simple means of ensuring that the SID for a Unix user (*e.g. - jsmith*) is reported as the one assigned to the Domain user (*e.g. - DOMAIN\jsmith*) - necessary for reporting ACLs on files and printers stored on a Samba member server.

Options

None.

Example

This example shows how to use idmap_nss to check the local accounts for its own domain while using allocation to create new mappings for trusted domains:

```
[global]
        idmap backend = tdb
        idmap uid     = 1000000-1999999
        idmap gid     = 1000000-1999999

        idmap config SAMBA : backend  = nss
        idmap config SAMBA : range    = 1000-999999
[global]
        idmap backend = tdb
        idmap uid     = 1000000-1999999
        idmap gid     = 1000000-1999999

        idmap config SAMBA : backend  = nss
        idmap config SAMBA : range    = 1000-999999
```

Appendix D: Active Directory Domain Services – Configuration Summary

This summary is provide as a guide to the installation and configuration of Active Directory Domain Services on Windows Server 2008 R2.

Prerequisites

The following are required before Active Directory can be configured on a Windows Server 2008 R2 server:

- Administrator account access
- Properly configured NIC (Static IP)
- NTFS partition with 250mb free space for Active Directory
- Functional DNS server (can be installed on the AD server itself or point to an existing DNS server)
- Domain name to use

Installation Summary

Refer to the following Microsoft TechNet article for the most current and comprehensive details:

http://technet.microsoft.com/en-us/library/cc770946.aspx[11]

An Active Directory installation involves the following series of steps on a Windows Server 2008 R2 server:

1. Install Active Directory Domain Services Role
2. Configure Active Directory Domain Services
3. Configure Windows Time Service
4. Create DNS Forward Lookup Zone
5. Restart DNS Service
6. Verify Active Directory Domain Services
7. Create User Accounts
8. Verify Client Access to Active Directory Domain
9. Add Red Hat Enterprise Linux 6 Server DNS A Record (*optional*)

Details on each of these steps are provided in the next section.

Installation Details

1. Install Active Directory Domain Services Role

    - Open *Server Manager* from the Quick Launch toolbar
    - Select **Roles** -> **Add Roles**
    - The *Add Roles Wizard* opens. Select **Next** to continue.
    - Under *Roles* select **Active Directory Domain Services**

      Note: If .NET Framework 3.5.1 is not installed, a prompt appears asking whether or not to install it. Select "Add Required Features" to continue.

    - Select **Next**
    - Select **Next**
      (after reading *Introduction to Active Directory Domain Services*)
    - Select **Install** (*Confirm Installation Selections*)
    - Select **Close** after confirming the Active Directory Domain Services (and if applicable .Net Framework 3.5.1) Installation Results.

2. Configure Active Directory Domain Services

    - Under *Roles Summary*, select the **Active Directory Domain Services** link
    - At the top of the *Summary* section select the **Run the Active Directory Domain Services Installation Wizard** (dcpromo.exe) link
    - Select **Next**
      (*Welcome to the Active Directory Domain Services Installation Wizard*)
    - Select **Next** (*Operating System Compatibility*)
    - In the *Choose a Deployment Configuration* window select **Create a new domain in a new forest**, then select **Next**
    - Enter the *Fully Qualified Domain Name (FQDN)* of the new forest domain

      Note: Do not use single label domain names – e.g. mycorp, eng, finance, etc. but use a fully qualified domain name (FQDN) – e.g. mycorp.com, eng.net, finance.mycorp.com, etc.

      Example: ***refarch-ad.cloud.lab.eng.bos.redhat.com***

    - Select **Next** to continue after the wizard has verified the domain name is not already in use on the local network
    - Select the appropriate *Forest functional level - Windows Server 2008 R2*.

      Note: If this is a forest in an existing domain then select the appropriate minimum server level appropriate to your environment.

    - Select **Next** to continue
    - In the *Additional Domain Controller Options* window, make sure **DNS server** is selected then select **Next**

- If a static IP address was not previously configured, then the *Static IP Assignment* window warns "*This computer had dynamically assigned IP address(es)*" if one or more network interfaces is set to a dynamic IP.

  Depending on your configuration select either of the following options below:

  "*Yes, the computer will use an IP address automatically assigned by a DHCP server (not recommended)*"

  ...or...

  "*No, I will assign static IP addresses to all physical network adapters*"

  Note: For production servers it is highly recommended that static IP addresses be used.

- The *Active Directory Domain Services Installation Wizard* warns that no DNS has been configured yet. Select **Yes** to continue.
- Select the locations for the Active Directory domain controller database, log files and SYSVOL folders. The default locations are:

  Database folder: ***C:\Windows\NTDS***
  Log files folder: ***C:\Windows\NTDS***
  SYSVOL folder: ***C:\Windows\SYSVOL***

  Note: For large installations each of these should be placed on separate volumes to maximize performance and recoverability

- Select **Next** to continue
- Enter the password for the *Directory Restore Mode Administrator Password*

  Note: Unlike regular domain user passwords this password remains constant and must remain secure and confidential. This password should be complex and at least 7 characters long. It is highly recommended not to use the administrator's password and that it be securely stored.

- Select **Next**
- After reviewing the *Summary* window select **Next**
- After the wizard creates the Active Directory Domain select **Finish**
- Select **Restart Now** to activate the changes

  Note: From this point forward, the AD domain name (*e.g. - refarch-ad*) must be specified for all user logins

3. Configure Windows Time Service

   - From a Command Window (*Start -> Run*: `cmd.exe`) run:

```
C:\WIN-SRV1> w32tm /config                              \
             /manualpeerlist:"ns1.bos.redhat.com"  \
             /syncfromflags:manual /update
```

   Note: Use the time server most appropriate to your environment

- To verify, enter:

```
C:\WIN-SRV1> w32tm /query /status
Leap Indicator: 0(no warning)
Stratum: 3 (secondary reference - syncd by (S)NTP)
Precision: -6 (15.625ms per tick)
Root Delay: 0.0999298s
Root Dispersion: 7.8073692s
ReferenceId: 0x0A10FF02 (source IP:  10.16.255.2)
Last Successful Sync Time: 3/22/2012 1:20:31 PM
Source: ns1.bos.redhat.com
Poll Interval: 6 (64s)
```

4. Create DNS Forward Lookup Zone

- Open *Server Manager* from the *Quick Launch* toolbar
- Select on **Roles** -> **DNS Server**
- Expand **DNS Server**
- Expand **DNS**
- Expand computer name (***win-srv1***)
- Right click **Forward Lookup Zones** and select **New Zone** from the drop-down
- The *New Zone Wizard* opens – select **Next**
- Select **Secondary zone**
- Select **Next**
- Enter Zone name: ***cloud.lab.eng.bos.redhat.com***
- Select **Next**
- Enter the IP Address of the Master Server: ***10.16.143.247***
- Select **Next**

5. Restart DNS Service

- Open *Server Manager*
- Select **Configuration** -> **Services**

  In the list of Services select **DNS Server**

- Select **Restart the service**
- Verify DNS is forwarding lookups. Open a Command Window and run:

```
C:\WIN-SRV1> nslookup www.redhat.com
Server:  ra-ns1.cloud.lab.eng.bos.redhat.com
Address:  10.16.143.247

Non-authoritative answer:
Name:    e1890.b.akamaiedge.net
Address:  23.64.247.214
Aliases:  www.redhat.com
          wildcard.redhat.com.edgekey.net
          wildcard.redhat.com.edgekey.net.globalredir.akadns.net
C:\WIN-SRV1>ipconfig /all

Windows IP Configuration
```

```
   Host Name . . . . . . . . . . . . : WIN-SRV1
   Primary Dns Suffix  . . . . . . . : refarch-
ad.cloud.lab.eng.bos.redhat.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : refarch-
ad.cloud.lab.eng.bos.redhat.com

Ethernet adapter Local Area Connection 5 - Public:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Broadcom BCM57711E
NetXtreme II 10 GigE (
NDIS VBD Client) #5
   Physical Address. . . . . . . . . : 00-17-A4-77-24-44
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.16.142.3(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Default Gateway . . . . . . . . . : 10.16.143.254
   DNS Servers . . . . . . . . . . . : 10.16.143.247
                                       127.0.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

6. Verify Active Directory Domain Services

   - Run the *Microsoft AD DS Best Practices Analyzer*:
   - Select **Roles** -> **Active Directory Domain Services**
   - Scroll down to the *Best Practices Analyzer*
   - Select **Scan This Role**
     Review the results and correct any errors or warnings.

     Note: The most common error is not having an NTP server set to synchronize time services. If this has not yet been done, follow the steps outlined in **Step 3. Configure Windows Time Service** before continuing.

     If this has already been done, then synchronize/update by running the following:

```
C:\WIN-SRV1> w32tm /config /computer:{hostname}.{domain}
/syncfromflags:domhier /update
```

     Example:

```
C:\WIN-SRV1> w32tm /config /computer:Win-srv1.refarch-
ad.cloud.lab.eng.bos.redhat.com /syncfromflags:domhier /update
```

   - Open a Command Window (*Start* -> *Run*: `cmd.exe`) and run `dcdiag`:

```
C:\WIN-SRV1> dcdiag
```

   - If any errors are found, run the `dcdiag` in verbose mode for additional details:

```
C:\WIN-SRV1> dcdiag /v
```

7. Create User Accounts

   - Open *Server Manager* from the *Quick Launch* toolbar
   - Select **Roles** -> **Active Directory Domain Services**
   - Select **Active Directory Users and Computers**
   - Open ***refarch-ad.cloud.lab.eng.bos.redhat.com*** (Domain)
   - Right click on **Users**, select **New User** and enter:

     First name: **AD** Initials: AD
     Last Name: **User 11**
     Full name: **AD AD.User 11**
     User logon name:

     ad-user11@refarch-ad.cloud.lab.eng.bos.redhat.com

   - Select **Next**
   - Enter Password: **********
     Confirm password: **********
   - Optionally uncheck the option ***User must change password at next logon***
   - Select **Password never expires**
   - Select **Next**
   - Select **Finish**

     For more detail on Windows password policy requirements, see the following Microsoft TechNet article:

     http://technet.microsoft.com/en-us/library/cc736605.aspx

8. Verify Client Access to Active Directory Domain

   Join Win7 Client to the domain

   - *Computer -> Properties*
   - Under **Computer name, domain and workgroup settings** right click on **Change settings**
   - Under the *Computer Name* tab, select **Change**
   - Under **Member of** click **Domain** and enter the name of the domain

     Example: ***REFARCH-AD***

   - In the *Windows Security* window, enter the username and password created on the Windows 2008 Server in the previous step:

     User name: **ad-user11**
     Password: **************

   - Select **OK**
   - Select **OK** to activate the changes
   - Select **Restart Now** to apply the changes

     Note: After the restart, be sure to login as the new domain user:

     e.g. - \\REFARCH-AD\ad-user11

*Create **`Win-Data`** file share (on Windows 2008 R2 Server)*

- *Computer -> Open* (or *Start -> Computer*)
- Right click on drive to share
- Select **Open**
- Select **New Folder**, enter a name (***e.g. - Win-Data***)
- Right click on the new folder:
  - Select **Share with**
  - Select **Specific People**
  - Enter the name of the domain user to grant file sharing access to *(**eg. ad-user11**)*
  - Select **Add**
  - Adjust the Permission Level accordingly (***e.g. - Read, Read/Write***)
  - Select **Share**
  - Select **Done**

*Populate **`Win-Data`** file share with a test file (on Windows 2008 R2 Server)*

- *Computer -> Open* (or *Start -> Computer*)
- Right click on drive containing the new share
- Select **Open**
- Right click on share folder (***e.g. - Win-Data***)
- Select **Open**
- Right click in empty folder
- Select **New**, **Text Document**
- Enter name (***e.g. - Win-Srv1***)
- Press **Enter**
- Right click on the new file
- Select **Open**
- Enter text into the test file with Notepad

```
+-----------------------------------------------------+
+  This file is located on the Windows Server 2008 R2   +
+  server named 'win-srv1.cloud.lab.eng.bos.redhat.com' +
+  located in the Active Directory domain 'REFARCH-AD'  +
+-----------------------------------------------------+
```

- Select **File**, **Save**
- Select **File**, **Exit**

*Verify **`Win-Data`** File Share Mapping (from Windows 7 Client)*

- *Computer -> Map Network Drive*
- Select Drive: ***W:***
- Enter Folder: ***\\win-srv1.refarch-ad.cloud.lab.bos.redhat.com\Win-Data***

  ...or...

  \\win-srv1\Win-Data (NetBIOS name)
- User name: **ad-user11**
- Password: **************
- Select **Finish**

  The new test file should be seen in the drive window. Verify write access on the Win-Data share from the Win7 client by creating a new file.

Supplemental Tasks

1. Add Red Hat Enterprise Linux 6 Server DNS A Record (*All Recommended Configurations*)

   In most environments it is necessary to add a DNS A (Address) records:

   - Open *Server Manager* from the *Quick Launch* toolbar
   - Select **Roles** -> **DNS Server**
   - Expand **DNS Server**
   - Expand **DNS**
   - Expand computer name (***win-srv1***)
   - Expand **Forward Lookup Zones**
   - Right click on the Active Directory domain (***refarch-ad.cloud.lab.eng.bos.redhat.com***)
   - Select **New Host** (A or AAAA)...
   - Enter Name: ***rhel-srv11***
   - Enter IP address: ***10.16.142.11***
   - Select **Add Host**

   Repeat this for each Red Hat Enterprise Linux 6 server.

2. Install Identity Management for UNIX Role (*Recommended Configuration 2 only*)

   - Open *Server Manager* from the *Quick Launch toolbar*
   - Select **Roles** -> **Add Role Services**
   - Select **Identity Management for UNIX**. The following is automatically selected:
     - Server for Network Information Services
     - Password Synchronization
     - Administration Tools
   - Select **Next**
   - Select **Install**
   - Restart the server to activate

3. Configure Group for Red Hat Enterprise Linux Users (*All Recommended Configurations*)

    - Open *Active Directory Users and Computers* (*Start -> Administrative Tools*)
    - Right click **Users** and select **New**, then select **Group**
    - Under the *New Object – Group* screen enter the following fields:
        - Group Name: ***rhel-users***
    - Select **OK**
    - Add each of the AD users by right-clicking each user and selecting **Properties**
    - Under the *Member Of* tab, select **Add**
    - Enter ***rhel-users*** for the object name
    - Select **Set Primary Group**
    - Select **OK**

4. Configure User Accounts for RFC2307 Support (*Recommended Configurations 2,3*)

    - Open *Active Directory Users and Computers* (*Start -> Administrative Tools*)
    - Right click a user (***e.g. AD AD.User 11***) and select **Properties**
    - Under the *UNIX Attributes* tab set the following fields:
        - NIS Domain: ***refarch-ad***
        - UID: ***10000011***
        - Login Shell: ***/bin/bash***
        - Home Directory: ***/home/REFARCH-AD/ad-user11***
        - Primary group name/GID: ***rhel-users***
    - Select **OK**

    Repeat this for each user account as needed.

5. Create User Account for LDAP Binds (*Recommended Configuration 4*)

    - Open *Active Directory Users and Computers* (*Start -> Administrative Tools*)
    - Right click *New -> User*
    - Under the *New Object - User* window set the following fields:
        - First Name: ***AD*** Initials: ***ADLB***
        - Last Name: ***LDAP-Bind***
        - User logon name (pre-Windows 2000): ***ad-ldap-bind***
    - Select **Next**
    - On the next screen enter and confirm a password (***eg – LDAPBind!!***)
    - Select **Next**

    Modify the Unix Attributes for the account

    - Select **OK**
    - Right click a user (***e.g. AD LDAP-Bind***) and select **Properties**

- Under the *UNIX Attributes* tab set the following fields:
  - NIS Domain: ***refarch-ad***
  - UID: ***40000000***
  - Login Shell: ***/bin/false***
  - Home Directory: ***/home/REFARCH-AD/ad-ldap-bind***
  - Primary group name/GID: ***rhel-users***
- Select **OK**

Appendix E: Active Directory User Account Mappings

Configuration	User	Group	Active Directory			Red Hat Enterprise Linux	
			NIS Domain	UID	GID	UID	GID
1	ad-user11	rhel-users	refarch-ad	10000011	10001115	10001103*	10001115*
1	ad-user12	rhel-users	refarch-ad	10000012	10001115	10001104*	10001115*
2	ad-user21	rhel-users	refarch-ad	20000021	10001115	20000021	10000002
2	ad-user22	rhel-users	refarch-ad	20000022	10001115	20000022	10000002
3	ad-user31	rhel-users	refarch-ad	30000031	10001115	30000031	10000002
3	ad-user32	rhel-users	refarch-ad	30000032	10001115	30000032	10000002
4	ad-user41	rhel-users	refarch-ad	40000041	10001115	40000042	10000002
4	ad-user42	rhel-users	refarch-ad	40000042	10001115	40000042	10000002

*Values are calculated via Winbind idmap_rid formula:

The Unix ID for a RID: ID = RID - BASE_RID + LOW_RANGE_ID.
The RID for a Unix ID: RID = ID + BASE_RID – LOW_RANGE_ID.

Example - ad-user11

SID = S-1-5-21-2363606248-3119620943-2416876723-1103 (# **wbinfo –name-to-sid ad-user11**)
RID = 1103
BASE-RID = 0
LOW_RANGE_ID = 10000000 (idmap config REFARCH-AD:range=10000000-19999999 in */etc/samba/smb.conf*)

10001103 = 1103 - 0 + 10000000 (RID – BASE_RID + LOW_RANGE_ID)

Appendix F: Command Reference – net, wbinfo

Command	Options { short | long }	Argument	Description
net	join	-W domain -S server -U user	Join the Active Directory domain (-W) through server (-S) as user (-U)
net	ads testjoin		Verify domain has been joined
net	ads info		Display Active Directory server details
net	ads lookup		Display Active Directory details
net	getdomainsid		Display local machine SID and domain SID
net	getlocalsid		Display the SID of the local domain
net	getlocalsid	domain	Display the SID of the specified domain
wbinfo	{ -? | --help }		Display help
wbinfo	{ -a | --authenticate }	username%password	
wbinfo	{ -p | --ping }		Ping the winbindd daemon
wbinfo	{ -D | --domain-info }	domain	Display domain information
wbinfo	{ --getdcname }	domain	Display domain controller information
wbinfo	(-m | --trusted-domains }		Display list of trusted domains
wbinfo	{ -n | --name-to-sid }	Name	Queries winbindd for SID of Name
wbinfo	{ -u | --domain-users }	UID	Lists (enumerates) all users in the domain
wbinfo	{ -g | --domain-groups }	GID	Lists (enumerates) all groups in the domain

Command	Options { short | long }	Argument	Description
wbinfo	{ -r | --user-groups }	username	List the GIDs for groups that username belongs to
wbinfo	{ -R | --lookup-rids }	RID1, RID2, ...*RIDn*	Converts RID(s) to name
wbinfo	{ --separator }		Display the winbind separator character
wbinfo	{ --online-status }		Display the domain status (*online*, *offline*)
wbinfo	{ -s | --sid-to-name }	SID	Convert a SID to a user or group name
wbinfo	{ -S | --sid-to-uid }	SID	Convert a SID to a UID
wbinfo	{ -t | --check-secret }		Verify machine account credentials
wbinfo	{ --user-domgroups }	SID	Print SIDS for user domain groups
wbinfo	{ --user-sids }	SID	Print SIDs for all groups the user is a member of

Appendix G: Reference Architecture Configurations

This section provides an overview of the hardware components used in the development of this reference architecture. The Red Hat Enterprise Linux 6 systems were deployed as eight KVM virtual machines hosted by one HP ProLiant BL460c G6 Blade server running Red Hat Enterprise Linux 6 and KVM. A second HP ProLiant BL460c G6 Blade server was dedicated for use as the Windows 2008 R2 Server. Both blade servers were contained within an HP BladeSystem c7000 enclosure and interconnected by a 10 Gb/s ethernet network. The IP addressing for this network was flat and all systems configured on the 10.16.142.x network.

For each Red Hat Enterprise Linux 6 system, a single 30G logical volume was created on the KVM host machine (*kvm-srv1*):

```
# lvcreate –size 30G –name rhel-srv11-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv12-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv21-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv22-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv31-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv32-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv41-lv rhel-win-vg
# lvcreate –size 30G –name rhel-srv42-lv rhel-win-vg
```

KVM virtual machines were then deployed for each Red Hat Enterprise Linux 6 system. Using the **virt-install** utility, virtual machines were provisioned with 2 processors, 4 GB memory and a bridged network interface as follows:

```
# virt-install --name rhel-srv11 --ram 4096 --vcpus=2 --cpuset=auto  \
               --os-type=linux   --os-variant=rhel6   --accelerate   \
               --network=bridge:vm-bridge             --vnc          \
               --disk path=/dev/rhel-win-kvm-vg/rhel-srv11=lv        \
               --cdrom /home/kits/rhel-server-6.2-x86-64-dvd.iso
```

For further details on the provisioning of KVM virtual machines, please consult the Red Hat Enterprise Linux 6 Virtualization Getting Started Guide[3].

Figure G-1: Reference Architecture Environment depicts an overview of the systems configuration:

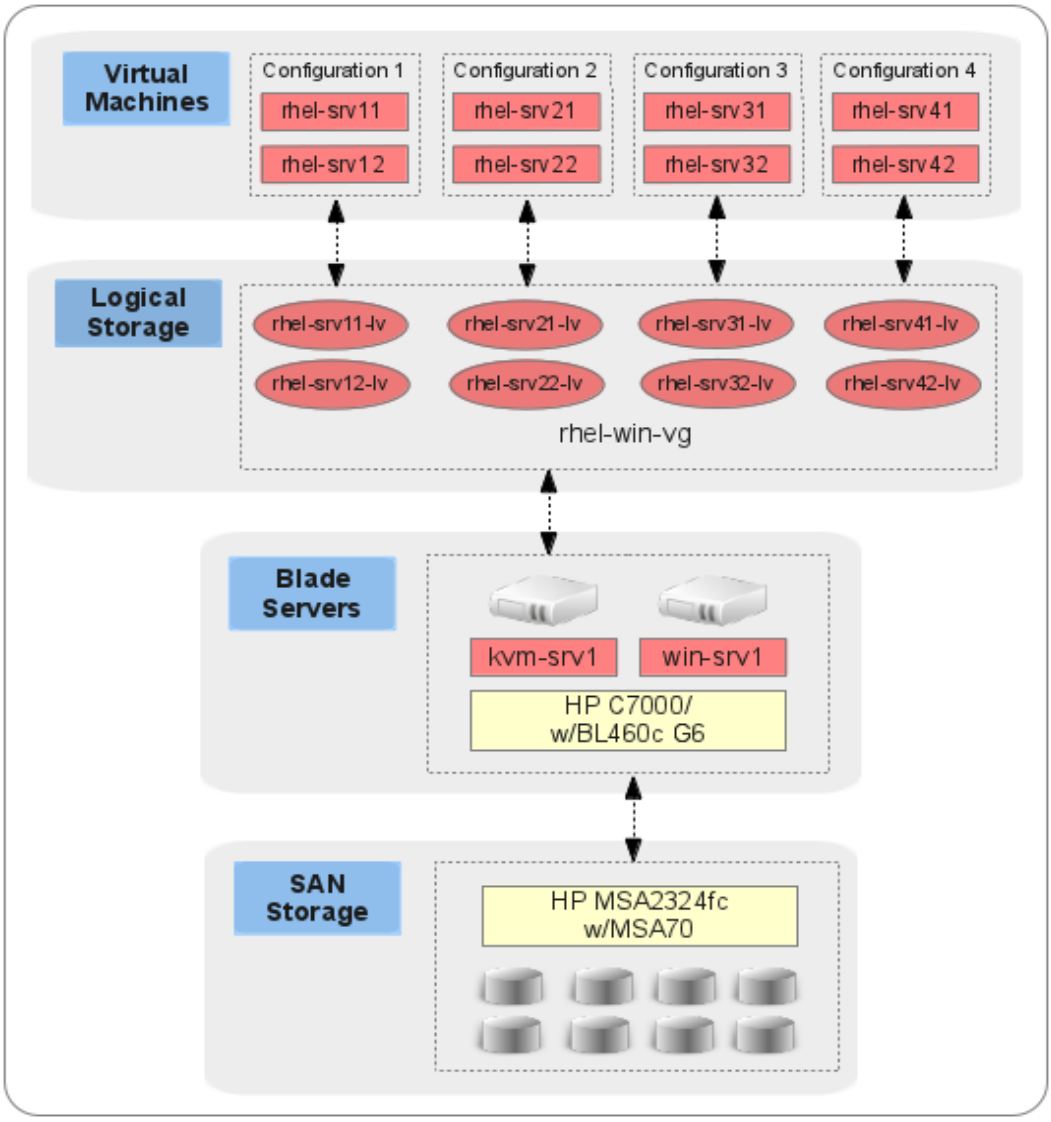

Figure G-1: Reference Architecture Environment

Red Hat Enterprise Linux KVM Host

Component	Detail
Hostname	*kvm-srv1*
Operating System	Red Hat Enterprise Linux 6.2 (64-bit) (2.6.32-220.7.1.el6.x86_64 kernel)
System Type	HP ProLiant BL460c G6
Processor	Quad Socket, Quad Core (16 cores) Intel® Xeon® CPU X5550 @2.67GHz
Memory	48 GB
Storage	2 x 146 GB SATA internal disk drive (RAID 1) 2 x Qlogic QMH2562 8Gb FC HBA
Network	8 x Broadcom NetXtreme II BCM57711E XGb

Table G-1: Red Hat Enterprise Linux KVM Host Configuration

Virtual Machines

Component	Detail
Hostnames	*rhel-srv11, rhel-srv12 (Recommended Configuration 1)* *rhel-srv21, rhel-srv22 (Recommended Configuration 2)* *rhel-srv31, rhel-srv32 (Recommended Configuration 3)* *rhel-srv41, rhel-srv42 (Recommended Configuration 4)*
Operating System	Red Hat Enterprise Linux 6.2 (64-bit) (2.6.32-220.7.1.el6.x86_64 kernel)
System Type	Virtual Machine (KVM)
Processor	2 Core
Memory	4 GB
Storage	30 GB LV (logical volume) – 1 for each virtual machine
Network	2

Table G-2: Virtual Machine Configurations

Windows 2008 R2 Server

Component	Detail
Hostname	*win-srv1*
Operating System	Windows 2008 Server R2 – Enterprise Edition (64-bit) Version 6.1 (Build 7601: Service Pack 1)
System Type	HP ProLiant BL460c G6
Processor	Quad Socket, Quad Core (16 cores) Intel® Xeon® CPU X5550 @2.67GHz
Memory	48 GB
Storage	2 x 146 GB SATA internal disk drive (RAID 1) 2 x Qlogic QMH2562 8Gb FC HBA
Network	8 x Broadcom NetXtreme II BCM57711E XGb

Table G-3: Windows 2008 R2 Server Configuration

Fibre Channel Storage Array

Component	Detail
Hostname	*ra-msa20*
System Type	HP StorageWorks MSA2324fc (1 x HP MSA70 expansion shelf)
Controllers	CPU Type: Turion MT32 1800MHz Cache: 1GB 2 x Host Ports
Firmware	Storage Controller Code Version: M112R14 Memory Controller FPGA Code Version: F300R22 Storage Controller Loader Code Version: 19.009 Management Controller Code Version: W441R39 Management Controller Loader Code Version: 12.015 Expander Controller Code Version: 1112 CPLD Code Version: 8 Hardware Version: 56
Physical Drives	48 x 146GB SAS drives (24 enclosure, 24 expansion shelf)
Logical Drives	4 x 1.3 TB Virtual Disks (12 disk, RAID 6)

Table G-4: Storage Array Configuration

Appendix H: Deployment and Integration Checklist – Configuration 1 (Samba/Winbind - idmap_rid)

Task	Task Description	Location	Details
	Deployment Tasks		
1	Deploy Windows Server 2008 R2	Windows Server 2008 R2 Server	Section 5.1 Appendix A[5]
2	Configure Active Directory	Windows Server 2008 R2 Server	Section 5.2 Appendix D
3	Deploy Red Hat Enterprise Linux 6	Red Hat Enterprise Linux 6 System(s)	Section 5.3 Appendix A[1]
4	Configure SELinux Security Parameters	Red Hat Enterprise Linux 6 System(s)	Section 5.4
5	Install/Configure Samba	Red Hat Enterprise Linux 6 System(s)	Section 5.5
6	Synchronize Time Services	Red Hat Enterprise Linux 6 System(s)	Section 5.6
7	Configure DNS	Red Hat Enterprise Linux 6 System(s)	Section 5.7
8	Install/Configure Kerberos Client	Red Hat Enterprise Linux 6 System(s)	Section 5.8
9	Install oddjob-mkhomedir	Red Hat Enterprise Linux 6 System(s)	Section 5.9
	Integration Tasks		
10	Configure Authentication	Red Hat Enterprise Linux 6 System(s)	Section 6.1.4 - Step 1
11	Verify/Test Active Directory	Red Hat Enterprise Linux 6 System(s)	Section 6.1.4 - Step 2
12	Modify Samba Configuration	Red Hat Enterprise Linux 6 System(s)	Section 6.1.4 - Step 3
13	Verification of Services	Red Hat Enterprise Linux 6 System(s)	Section 6.1.5

Appendix I: Deployment and Integration Checklist – Configuration 2 (Samba/Winbind - idmap_ad)

Task	Task Description	Location	Details
	Deployment Tasks		
1	Deploy Windows Server 2008 R2	Windows Server 2008 R2 Server	Section 5.1 Appendix A[5]
2	Configure Active Directory	Windows Server 2008 R2 Server	Section 5.2 Appendix D
3	Deploy Red Hat Enterprise Linux 6	Red Hat Enterprise Linux 6 System(s)	Section 5.3 Appendix A[1]
4	Configure SELinux Security Parameters	Red Hat Enterprise Linux 6 System(s)	Section 5.4
5	Install/Configure Samba	Red Hat Enterprise Linux 6 System(s)	Section 5.5
6	Synchronize Time Services	Red Hat Enterprise Linux 6 System(s)	Section 5.6
7	Configure DNS	Red Hat Enterprise Linux 6 System(s)	Section 5.7
8	Install/Configure Kerberos Client	Red Hat Enterprise Linux 6 System(s)	Section 5.8
9	Install oddjob-mkhomedir	Red Hat Enterprise Linux 6 System(s)	Section 5.9
	Integration Tasks		
10	Configure Authentication	Red Hat Enterprise Linux 6 System(s)	Section 6.2.4 - Step 1
11	Verify/Test Active Directory	Red Hat Enterprise Linux 6 System(s)	Section 6.2.4 - Step 2
12	Modify Samba Configuration	Red Hat Enterprise Linux 6 System(s)	Section 6.2.4 - Step 3
13	Verification of Services	Red Hat Enterprise Linux 6 System(s)	Section 6.2.5

Appendix J: Deployment and Integration Checklist – Configuration 3 (SSSD/Kerberos/LDAP)

Task	Task Description	Location	Details
Deployment Tasks			
1	Deploy Windows Server 2008 R2	Windows Server 2008 R2 Server	Section 5.1 Appendix A[5]
2	Configure Active Directory	Windows Server 2008 R2 Server	Section 5.2 Appendix D
3	Deploy Red Hat Enterprise Linux 6	Red Hat Enterprise Linux 6 System(s)	Section 5.3 Appendix A[1]
4	Configure SELinux Security Parameters	Red Hat Enterprise Linux 6 System(s)	Section 5.4
5	Synchronize Time Services	Red Hat Enterprise Linux 6 System(s)	Section 5.6
6	Configure DNS	Red Hat Enterprise Linux 6 System(s)	Section 5.7
7	Install/Configure Kerberos Client	Red Hat Enterprise Linux 6 System(s)	Section 5.8
8	Install oddjob-mkhomedir	Red Hat Enterprise Linux 6 System(s)	Section 5.9
Integration Tasks			
9	Configure Authentication	Red Hat Enterprise Linux 6 System(s)	Section 6.3.4 - Step 1
10	Enable LDAP Searches	Red Hat Enterprise Linux 6 System(s)	Section 6.3.4 - Step 2
11	Modify SSSD Configuration	Red Hat Enterprise Linux 6 System(s)	Section 6.3.4 - Step 3
12	Verification of Services	Red Hat Enterprise Linux 6 System(s)	Section 6.3.5

Appendix K: Deployment and Integration Checklist – Configuration 4 (Kerberos/LDAP)

Task	Task Description	Location	Details
	Deployment Tasks		
1	Deploy Windows Server 2008 R2	Windows Server 2008 R2 Server	Section 5.1 Appendix A[5]
2	Configure Active Directory	Windows Server 2008 R2 Server	Section 5.2 Appendix D
3	Deploy Red Hat Enterprise Linux 6	Red Hat Enterprise Linux 6 System(s)	Section 5.3 Appendix A[1]
4	Configure SELinux Security Parameters	Red Hat Enterprise Linux 6 System(s)	Section 5.4
5	Synchronize Time Services	Red Hat Enterprise Linux 6 System(s)	Section 5.6
6	Configure DNS	Red Hat Enterprise Linux 6 System(s)	Section 5.7
7	Install/Configure Kerberos Client	Red Hat Enterprise Linux 6 System(s)	Section 5.8
8	Install oddjob-mkhomedir	Red Hat Enterprise Linux 6 System(s)	Section 5.9
	Integration Tasks		
9	Configure Authentication	Red Hat Enterprise Linux 6 System(s)	Section 6.4.4 - Step 1
10	Modify LDAP Configuration	Red Hat Enterprise Linux 6 System(s)	Section 6.4.4 - Step 2
11	Add PAM Libraries	Red Hat Enterprise Linux 6 System(s)	Section 6.4.4 - Step 3
12	Modify NSS	Red Hat Enterprise Linux 6 System(s)	Section 6.4.4 - Step 4
13	Verify LDAP Queries	Red Hat Enterprise Linux 6 System(s)	Section 6.4.4 - Step 5
14	Enable NSS Caching	Red Hat Enterprise Linux 6 System(s)	Section 6.4.4 - Step 6
15	Verification of Services	Red Hat Enterprise Linux 6 System(s)	Section 6.4.5