Novell Services

www.novell.com

Security Hardening for Novell Products

Connecting for Health

May 2009

Novell Services

www.novell.com

Disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Trademarks

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

* All third-party trademarks are property of their respective owner.

Novell, Inc.

404 Wyman

Suite 500

Waltham

Massachusetts 02451

USA

Novell UK Ltd

One Arlington Square

Downshire Way

Bracknell

Berkshire

RG12 1WA

Prepared By

Jim O'Moore

Best practice recommendations for securing Novell eDirectory, Identity Management and SuSE Linux Enterprise Server

Published: May, 2009

Table of Contents

Novell Services

www.novell.com

Baseline Solution Overview

The Novell Identity Management Solution provides the NHS with a platform to integrate systems and applications both at a national and local level, and provide core capabilities for account life-cycle management with key local and national systems/applications,

The solution is available as a packaged “Baseline System” offering with the overall intention of providing a reference set of IDM components that can be re-used throughout the NHS. It is intended that additional customisations and tailoring will be made to the solution, in order to meet different needs and requirements across the NHS.

The solution is designed to incorporate best and leading practice. This document is produced with the intention of providing detailed security information for the solution components.

Security Considerations

Customers often ask “How do I harden my Novell products?”. In other words, what settings can I change in order to make my default installation less susceptible to attack or mitigate security threats.

The reasons for such a request vary. Understandably customers are concerned about security breaches, and often their experience with other operating systems, particularly in an internet-facing role, may not have been encouraging.

However, it is not possible to react sensibly to a very general request from management to "make these servers as secure as possible" without first making a specific assessment of what the relevant security threats are in a particular environment and what you want to achieve.

In practice, a fully patched internet-facing SLES system running a firewall can be considered extremely safe from external and internal attack. Apart from the threat from external malicious attackers, it is also important to consider the possible potential threats from internal users and administrators, both through error and deliberate action.

Fortunately, SLES provides various tools which can assist both in hardening the system against external attack, and with detecting and auditing undesirable changes to the system.

Document Scope

This document provides “best practice” recommendations on security “hardening” for SuSE Linux Enterprise Server, eDirectory, Identity Management and User Application. It covers steps for increasing the security of each product beyond the default settings applied during installation. These recommendations should the combined with local system knowledge to provide the most secure installation possible, without adversely impacting on system usability.

Intended Audience

This document is written specifically for Novell deployment partners as well as NHS Organisations that intend to deploy the Novell Identity Management Baseline solution. It is anticipated that the content will be of particular interest to the IT Security Officer/Group and Technical Design/Implementation Teams.

1. SuSE Linux Enterprise Server

1.1 Physical Access

The single biggest risk to security on any server is physical access. With physical access, and enough time any system can be compromised. Physical access gives an attacker the ability to boot from other media, effectively bypassing all security measures. There is also the risk of vandalism.

Certain boot options can be typed into the GRUB menu to bring a system up into a maintenance mode without the root password being required. To mitigate this risk you can password protect the GRUB menu itself, but the reality is that with physical access your systems are completely vulnerable

1.2 Updates

Security is an ongoing concern, and although a system can be `hardened` it is important to keep abreast of security patches and updates, and apply them in a timely fashion.

Novell maintains a security update page, located at http://support.novell.com/security-alerts/ from where you can report issues you discover, subscribe to a security list server and view current security related patches. The site also contains links to third-party security sites that you are encouraged to review for information relevant to your systems.

1.3 Permissions

Permissions on system files need to be set correctly so that a user or process cannot make unauthorised changes to these files. This is a function of the operating system installation. In general, provided you do not change permissions on system files, all should be well. However, SUSE systems have various levels of pre-set permissions, which are defined in the files `/etc/permissions`, `/etc/permissions.easy`, `/etc/permissions.local`, `/etc/permissions.secure` and `/etc/permissions.paranoid`. There are also some permissions settings defined for the operation of particular programs in files under `/etc/permissions.d/`

One might ask why it would ever be advisable to set less than the strictest possible set of permissions. The answer is that on multi-user systems, the strictest possible set of permissions will have an effect that prevents normal non-root users from doing certain things that they need to do: accessing particular devices (for example using a program to write to a CD), on a desktop system shutting down the system, and so on.

There is an element of balance that has to be struck between locking the system down tightly so that people cannot do the things that they need to do, and leaving too many possibilities for bad things to happen.

To set one of the pre-set system-wide permissions settings, one can use YaST's "Local Security" module (yast2 security), or they can be set manually in the file `/etc/sysconfig/security`. You can create your own list of file permissions (for example `/etc/permissions.mysite`) and then refer to this in `/etc/sysconfig/security` just using the key word mysite. After making any changes made in `/etc/sysconfig/security`, the program SuSEconfig should be run so that the changes are applied to the system. If using the YaST "Local Security" module, this will be done automatically on exit.

There are a small number of files that need to have SUID root permissions set: such programs are a possible attack vector: you can check which files on the system have SUID permissions with a command such as:

```
find / -type f -perm -u=s -ls
```

Files that are both executable and writeable by others can be found using

```
find / -type f -perm -o=w,u=x -ls
```

1.4 seccheck

The file permissions checks mentioned above and a variety of others are included in the functionality of the *seccheck* package, which runs various tests on the system on a regular basis as cron jobs.

The following daily checks are done:

/etc/shadow check	length/number/contents of fields, accounts with no password
/etc/group check	length/number/contents of fields
user root checks	secure umask and PATH
/etc/ftpusers	checks if important system users are put there
/etc/aliases	checks for mail aliases which execute programs
.rhosts check	checks if users' .rhosts file contain + signs
homedirectory	checks if homedirectories are writable or owned by someone else
dot-files check	checks many dot-files in the home directories if they are writable or owned by someone else
mailbox check	checks if user mailboxes are owned by user and unreadable
NFS export check	exports should not be exported globally
NFS import check	NFS mounts should have the "nosuid" option set
promisc check	checks if network cards are in promiscuous mode
list modules	just lists loaded modules

list sockets	just lists open ports

The following weekly checks are done

rpm md5 check	checks for changed files via rpm's md5 checksum feature
suid/sgid check	lists all suid and sgid files
exec group write	lists all executables which are group/world writeable
writable check	lists all files which are world writable (incl. above)
device check	lists all devices

The following monthly things are done:

The monthly file is not a diff like the daily/weekly ones but the full reports in one file.

1.5 Chroot services

Certain services (FTP server, NTP, postfix, dhcpd, etc) have the option to be run in a *chroot* environment, and this is the default configuration in most cases. Running such services '*chrooted*' ensures that any external access to the server can only affect that part of the file system where that particular *chroot* system resides.

1.6 Security issues relating to particular services

If you are running services which allow direct access to the server, you should take particular care that they are configured correctly. In particular any server that allows FTP uploads should be tested and configured with great care.

1.7 Password policies

The built-in tools on SLES already advise strong passwords. In addition password change can be forced after a specific period of time using the `chage` command or options in the `useradd` command.

Various password cracking tools are available: these can be used against the encrypted passwords stored in /etc/shadow to check their strength.

The Linux-PAM Guides contain very good information about the use of the PAM configuration in general, and examples of how to enforce policies regarding passwords in particular (length, number of non-alphabetic characters required, similarity to the previous password, etc). See: http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/

1.8 sudo

Accidental or malicious damage by systems administrators is a potential risk that can easily be overlooked. The more people that have root access to the system, the greater the risk that damage can be done, whether deliberately or accidentally. The sudo mechanism allows you to specify particular users and particular commands that those users are allowed to run as root. For more information see the man page for sudo. A short guide to sudo for openSUSE is available at http://en.opensuse.org/Administer_with_Sudo

If you wish to have a full audit trail of who has performed which actions on a server, whether as root or not, you can disable root logins completely, and insist that all administration is carried out using sudo sessions. In combination with audit (see below), this provides the capability to maintain a very full audit trail of who has done what

1.9 Unnecessary programs and services

In general it is good practice to choose the smallest possible installation set consistent with your intended use of the system. It is certainly not a good idea to have services running that you are not using, because there is always a possibility, however remote, that an attacker might be able to target an open port on the system which is not actually required. If you are not going to run a service, it is better not to have it installed. So the best plan is to start with a minimal installation pattern, and progressively add the items that you know you will need. Do not, however, break package and pattern dependencies while attempting to trim down your installation: this could affect the installation in unexpected ways and render your system unsupported.

1.10 Apparmor

Apparmor is a powerful tool which can prevent an application with a security vulnerability (or a maliciously replaced binary) from writing to files that it should not be accessing. Profiles can be created for particular executables on the basis of their normal operation, and then those profiles can be enforced by Apparmor. If a profiled application is replaced by a malicious version, or happens to contain a security hole that allows it to write to some system file, Apparmor will prevent this from happening. Apparmor therefore provides pre-emptive protection from potential (but still unknown) security issues, and is an effective way of increasing the security of a server.

There is good documentation for Apparmor: chapter 48 of the SLES Administration Guide describes the setup of Apparmour.

There is also a specific Apparmor manual which is available as the package apparmor-admin_en for SLES 10, and a guide (written for SLES 9, but mostly valid for SLES 10) is available online:

Installing a Secure Server with SUSE Linux Enterprise Server 9 and Novell AppArmor

http://www.novell.com/collateral/4622008/4622008.pdf

There is also a FAQ: http://developer.novell.com/wiki/index.php/Apparmor_FAQ

The AppArmor project page http://developer.novell.com/wiki/index.php/Apparmor

See also: http://www.novell.com/documentation/apparmor/pdfdoc/apparmor_qs_sp2/apparmor_qs_sp2.pdf
http://www.novell.com/documentation/apparmor/apparmor201_sp2_admin/data/apparmor201_sp2_admin.htm

1.11 AIDE

AIDE is the Advanced Intrusion Detection Environment. It monitors changes in files with the particular purpose of detecting changes that have been caused by malware or security breaches. Typically AIDE creates a database immediately after the system is installed. When AIDE is run after this, it reports on changes that have taken place relative to the previous state. A configuration file controls which files and directories you wish to monitor. Documentation is included with the package.

Note: AIDE replaces tripwire(tm).

1.12 logwatch

The logwatch package is available in the SLES 10 SDK. As such it is not a supported package, but it is useful for parsing and analysing log files and producing reports. Among other things this can be useful for reporting failed login attempts, but it can be configured to report on any activity that gets logged in the various system logs. Documentation is included with the package.

1.13 Network security tools

For analysing network vulnerabilities, the tools `nmap` (port scanner) `nessus` (a network vulnerability scanner), and `snort` (a network intrusion prevention and detection system) are also available in the SDK.

The sites for the upstream projects are:

http://nmap.org/

http://www.nessus.org/documentation/

http://www.snort.org/

For analysing network traffic more generally, the tools *tcpdump*, *wireshark* (previously known as ethereal), *iptraf* and *iftop* can be very useful.

1.14 Firewalls

A server which is available from the internet should be protected either by its own firewall or by a third party firewall that forwards packets to it. On a SLES system you can create a firewall simply by using the SuSEfirewall2 package which

creates iptables (netfilter) rules based on your own specifications. You can also, if you wish create your own script to set up the iptables rules without using SuSEfirewall2. Any firewall rules you set should be thoroughly tested: the network security tools mentioned above can be useful for this. Documentation on SuSEfirewall2 is included with the package, and is also included in Chapter 43 of the SLES 10 manual. Extensive documentation on the use of iptables is available at http://www.netfilter.org/documentation/index.html

1.15 SSH configuration

By default SUSE systems install a configuration file for the SSH server (/etc/ssh/sshd_config) that permits root logins over SSH. For an internet facing server it is advisable to edit the file so that it contains the entry: PermitRootLogin no

Automated brute force attacks against the SSH server are quite common: it is possible to apply firewall rules that detect and prevent more than a defined number of connection attempts from a particular host in a particular time period.

For example:

```
iptables -N external
```

```
iptables -A external -i eth0 -p tcp --dport 22 -m state --state NEW -m recent --set
--name SSH
```

```
iptables -A external -i eth0 -p tcp --dport 22 -m state --state NEW -m recent --update
--seconds 600 --hitcount 4 --rttl --name SSH -j LOG --log-prefix "SSH BRUTE FORCE
PROTECTION "
```

```
iptables -A external -i eth0 -p tcp --dport 22 -m state --state NEW -m recent --update
--seconds 600 --hitcount 4 --rttl --name SSH -j DROP
```

These rules will prevent more than 4 connection attempts per 10 minute period from a particular IP number.

If possible it is advisable to switch off username / password authentication for SSH with the line

```
ChallengeResponseAuthentication no
```

Then only public/private key-pair authentication will be possible, and brute force attacks cannot take place.

1.16 TCP Wrappers

Access to many services, including SSH can be controlled by TCP Wrappers to limit access particular IP addresses or networks: see `man tcpd` for information on how to configure the files `/etc/hosts.allow` and `/etc/hosts.deny` to enforce this.

1.17 The audit package

The Linux kernel has auditing capabilities. The user-space tools in the audit package allow an administrator to specify rules in the file `/etc/audit/audit.rules` which will force the logging of matching events. So any time that a specified program is run, or a particular file is read, the audit daemon will log that activity to the file `/var/log/audit/audit.log`. The tool ausearch will search that log file for specific information, and aureport can create various report from the log file. The use of these tools together with careful rules for sudo can create a highly detailed audit trail of all activities on a server. Novell documentation for the audit package is available at:
http://www.novell.com/documentation/sled10/pdfdoc/auditqs_sp2/auditqs_sp2.pdf
http://www.novell.com/documentation/sled10/pdfdoc/audit_sp2/audit_sp2.pdf

1.18 compartm

The compartm package (available in the SDK) contains tools to force untrusted executables to run in a sandboxed environment so that they cannot damage the system.

1.19 chkrootkit

The chkrootkit package checks for signs of a rootkit (malicious software that allows root access after entering the system via a vulnerability of some kind, and probably changing certain system files). It is available from http://www.chkrootkit.org/

Packages are available for openSUSE: it can be built from source for SLES if necessary.

It is important to be aware, however that rootkits that also install or modify the kernel as well as installing malicious versions of other files can in principle be undetectable from the running system, and forensic analysis needs to be done under the rescue system to detect (for instance) changed file checksums.

1.20 Summary of Recommendations

- Restrict physical access
- Ensure you system is updated with the latest patches
- Set file permissions to protect important directories, and check this with the seccheck tool
- Where applicable run service in a chroot jail
- Ensure that services that provide direct access to the server, such as FTP, are properly configured
- Maintain a strong password regime for local server accounts
- Where possible employ the sudo tool to avoid giving root access to too many people
- Remove all unnecessary programs and services
- Use AppArmor to protect against unknown threats and AIDE to monitor file changes

- Regularly test your systems with network security tools and packet capturing programs
- Maintain a properly configured firewall
- Properly configure SSH, removing root access via SSH where applicable
- Configure auditing capabilities on the server to help with investigations into any attacks against your systems

2. Novell eDirectory

2.1 Physical Access

As with any system, physical access to eDirectory servers immediately lowers their security level. With physical access and enough time, tools can be employed to allow unauthorised access to an eDirectory system.

2.2 DIB Backups

Various tools, including (n)dsrepair can create a backup copy of a running eDirectory database. These DIBs can be restored onto other systems and interrogated at leisure, potentially providing valuable information to a would-be cracker. Steps should be taken to ensure that these backups are kept away from unauthorised access, through correctly set file permissions and strict administration policies.

2.3 LDAP

LDAP should be configured to require TLS (Transport Layer Security) for simple binds. Clear text passwords should be disabled as they are readily captured by means of a packet sniffing tool. By default LDAP in eDirectory is configured to use secure port 636, but clear text connection requests can still be sent to port 389, and these are relatively easy to capture before eDirectory has a chance to switch them to port 636.

By default LDAP anonymous binds are granted rights from the [public] user, a special system account that allows browsing throughout the tree. In order to better control the access of anonymous binds a dedicated LDAP proxy user should be created, and rights assigned to it as required. Different LDAP proxy users can be assigned to individual LDAP servers within the tree, to further manage what can be viewed via anonymous binds. Care should be taken that application which are dependent on anonymous binds for their functionality are not adversely effected.

Simple Authentication and Security Layer (SASL) defines various authentication mechanisms that can be used to further secure LDAP communications. These include DIGEST-MD5, EXTERNAL and NMAS_LOGIN

2.4 iMonitor/DHOST

iMonitor/DHOST provides remote tools for managing your eDirectory environment, providing web-based access to tools such as DStrace and DSrepair. By default access is granted over ports 8028 (insecure) and 8030 (secure). On external facing servers these ports should be filtered out, and it may even be advisable to disable the functionality completely by remarking the HTTPSTK module out of `/usr/lib/nds-modules/ndsmodule.conf`.

2.5 Admin user

By default during the installation a user object called Admin is created at the Organisation level of the eDirectory tree. It is recommended that this object be moved to another location, and protected by a very complex password. System administrators should have personal accounts with Admin-equivalent access rather than using the Admin account itself. This also allows for a proper audit trail as actions are not recorded against a "catch all" Admin account.

2.6 Inherited Rights Filters (IRF)

The ability to browse the eDirectory tree, and record account names is a useful tool for a someone attempting to compromise security. Consideration should be given to employing inherited rights filters at various levels within the tree, in order to prevent unauthorised users from interrogating the system and building up a list of valid account names that could be subsequently used for intrusion attempts.

Auditing tools should also be used to ensure that nobody is using IRFs in order to hide “back door” accounts from the system administrators.

2.7 Summary of Recommendations

- Physically protect your eDirectory servers, and prevent unauthorised access to DIB backups
- Configure LDAP to use TLS for simple binds and avoid anonymous binding by employing LDAP proxy users
- Where necessary disable iMonitor/DHOST or filter the port 8028 and 8030, especially on external facing servers
- Relocate the Admin account to a non-default location, and protect it with a highly complex password. Administration staff should use Admin-equivalent logings, also protected by complex paswords
- Where feasible user Inherited Rights Filters to prevent browsing of the tree (social engineering) which can quickly provide an attacker with a long list of targets for a password cracking effort
- Also be aware that IRFs can be used to obscure objects from the administrators view, so auditing tools should be used to verify IRF placement

3. Identity Management

3.1 Physical Access

Once again, physical access to the servers running your Identity Management environment is a very high priority. Without physical security, all other measures are insufficient to properly secure your systems.

3.2 Using SSL

Enable SSL for all transports, where it is available. Enable SSL for communication between the Identity Vault, which hosts the IDM engine and the connected system which generally runs the Remote Loader application. If you don't enable SSL, you are sending sensitive information, such as passwords in clear text.

3.3 Securing Directory Access

The security of the file system for Identity Manager is critical to ensuring the security of the system as a whole. Verify that the directories containing eDirectory, the Identity Vault and the Remote Loader are accessible only to the appropriate administrators. There is an issue with the file system when the Remote Loader is installed on a Windows* 2000 server. For more information, see TID 3243550, Securing a Remote Loader Install on a Microsoft Windows 2000 Server. Identity Manager requires Administrative rights to create Identity Manager objects and configure drivers. Monitor and control who has rights to create or modify Identity Management objects, including drivers,driver sets, filters, style sheets and policies.

3.4 Granting Task-Based Access to Drivers and Driver Sets

In addition to the eDirectory standard object-based access controls, Identity Manager lets you assign trustee rights to perform only certain tasks on an Identity Manager driver, rather than just granting full Supervisor rights to the driver object. For example, you can assign trustee rights so that one user can only configure the driver object (create and modify object properties), while another user can only start and stop the driver. The goal of providing this attribute-based access to driver tasks is to let you create well-defined administrative roles, perhaps using the eDirectory Administrative Role object, that let users perform certain management tasks without exposing all management functionality. Creating these roles can go beyond providing access to the DirXML-Access attributes described above and can include access rights to other attributes, as well as access to other Identity Manager objects.

Information about using iManager to grant eDirectory access rights is available in the *iManager Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_27/data/hk42s9ot.html).

3.5 Managing Passwords

When you choose to exchange information between connected systems, you should take precautions to make sure that the exchange is secure. This is especially true for passwords.

Challenge Questions are publicly readable, to allow unauthenticated users who have forgotten a password to authenticate another way. Requiring Challenge Questions increases the security of Forgotten Password Self-Service, because a user must prove his or her identity by giving the correct responses before resetting their password.

The intruder lockout setting is enforced for Challenge Questions, so the number of incorrect attempts an intruder could make is limited.

3.6 Creating Strong Password Policies

Password policy objects are publicly readable to allow applications to check whether passwords are compliant. This means that an unauthenticated user could query an Identity Vault and find out what password policies are in place. If the password policies require users to create strong passwords, this should not pose a risk, as noted in “Create Strong Password Policies” in the *Password Management 3.2 Administration Guide* (http://www.novell.com/documentation/password_management32/index.html). Identity Manager Password Synchronization lets you simplify user passwords and reduce help desk costs. Bidirectional password synchronization lets you share passwords among eDirectory and connected systems in multiple ways, as described in the scenarios in the *Identity Manager 3.6 Password Management Guide*.

Using Universal Password and password policies allows you to enforce strong password syntax requirements for users. Use the Advanced Password Rules in password policies to define your organization’s best practices for passwords. The Advanced Password Rules features let you manage password syntax by using either Novell syntax or the Microsoft Complexity Policy. For more information, see “Managing Passwords by Using Password Policies” in the *Novell Password Management 3.2 Administration Guide* (http://www.novell.com/documentation/password_management32/pwm_administration/data/ampxjj0.html).

For example, using Novell password syntax options, you can require user passwords to comply with rules such as the following:

- Requiring unique passwords.
- Requiring a minimum number of characters in the password.
- Requiring a minimum number of numerals in the password.
- Excluding passwords of your choice.

You can exclude words that you consider to be security risks, such as the company name or location, or the words “test” or “admin.” Although the exclusion list is not meant to import an entire dictionary, the list of words you exclude can be

quite long. Just keep in mind that a long list of exclusions makes login slower for your users. A better protection from dictionary attacks is to require numerals or special characters.

Keep in mind that you can create multiple password policies if you have different password requirements in different parts of the tree. You can assign a password policy to the whole tree, a partition root container, container, or even an individual user. (To simplify administration, we recommend that you assign password policies as high up in the tree as possible.)

In addition, you can use intruder lockout. As always, this eDirectory feature lets you specify how many failed login attempts are allowed before an account is locked. This is a setting on the parent container instead of in the password policy. See “Managing User Accounts” in the *Novell eDirectory Administration Guide* (http://www.novell.com/documentation/edir88/index.html?page=/documentation/edir88/edir88/data/afxkmdi.html).

3.7 Securing Connected Systems

Keep in mind that the connected systems that you are synchronizing data to might store or transport that data in a compromising manner. Secure the systems with which you exchange passwords. For example LDAP, NIS, and Windows each have security concerns that you must consider before enabling password synchronization with those systems. Many software vendors provide specific security guidelines that you should follow for their products.

3.8 Designer for Identity Manager

When using Designer for Identity Manager, consider the following issues:

- Monitor and control who has rights to create or modify an Identity Manager driver. Administrative rights are needed to create Identity Manager objects and configure drivers.
- Before giving a consultant an Identity Vault administrator password, limit the rights assigned to that administrator to areas of the tree that the consultant must access.
- Delete the project files (.proj) or save them to a company directory. Designer .proj files are to remain at the company’s project site. A consultant does not take the files after completing a project.
- After project files, log files, and trace files are no longer needed, delete them.
- Before discarding or surplusing a laptop, verify that project files have been cleaned.
- Ensure that the connection from Designer to the Identity Vault server is physically secure. Otherwise, someone could monitor the wire and pull sensitive information.
- When you use Document Generator to create documents, be careful with those documents. These documents can contain passwords and sensitive data in clear text.

- If Designer needs to read or write to an eDirectory attribute, do not mark that attribute as encrypted. Designer is unable to read or write to encrypted attributes.
- Do not store passwords that are sensitive. Currently, Designer projects are not encrypted. Passwords are only encoded. Therefore, do not share Designer projects that have saved passwords.

3.9 Industry Best Practices for Security

Follow industry best practices for security measures, such as blocking unused ports on the server. You can use Novell Audit to log events that you consider important for security. For example, you could log password changes for a particular Identity Manager driver (or driver set) by setting the log level to “Log Specific Events” through iManager or the Designer and selecting those events you wish to capture.

3.10 Summary of Recommendations

- Use SSL for Active Directory and eDirectory baseline IDM Drivers.
- If using Windows 2000, follow TID 3243550 to restrict security.
- Ensure that the IDM user defined on the IDM Engine and target system(s) is restricted with the level of rights they require, based on the operations they will support. This will depend on the level of synchronisation required, however the Novell Deployment Partner will be able to advise further on this.
- Ensure that the IDM User Application is configured with SSL.
- At a minimum, adopt the CFH password policy settings.
- Control access rights in the Identity Vault. Also ensure designer files and generated documents are stored in a secure location with file access control restrictions.

4. User Application Configuration

During the installation of the Role Based Provisioning User Application, the configuration panel is displayed allowing you to specify various settings for the User Application module. This configuration page can be re-run at any stage by launching the IdmUserApp.jar file.

4.1 eDirectory Connection Settings

Ensure the LDAP host and ports are configured correctly, especially the secure port. The LDAP administrator should be a dedicated Admin-equivalent user specifically for use by the User Application. Do no use the default Admin user, as previously suggested in the eDirectory section of this document. The password used should be complicated, meeting CFH minimum standards. The Use Public Anonymous Account option should be disabled to prevent anonymous binds, and the Guest LDAP user and password set instead. This will allow users how are not logged in to access public portals, while still securing the connection over SSL. Finally ensure that the Secure Admin Connection and Secure User Connection are both enabled. This will require all traffic between the User Application and users or administrators to be sent over a secured SSL connection.

It should be noted that encrypting all connections poses a traffic and processing overhead, but this needs to be balanced against the security risks involved in not doing so.

4.2 eDirectory DN's

The User Application can be configured to use four different administration accounts for user administration, provisioning, compliance and roles administration. It is recommended that four separate user accounts be created for each of these roles, along with a complex password. In this way the audit trail will clearly show which account performed any given action.

4.3 eDirectory Certificates

Specify the path to the cacerts key store and protect it with a complex password. The default password is documented publicly, and as such presents a possible security risk.

4.4 Private Key Store

Both the private key and the private key store should be protected with a complex password. It is also important to ensure that nobody but authorised administrators have access to the path specified in the Private Keystore Path field, as access to this key could allow an attacker to create an authorise connection to the Use Application and possibly exploit it.

4.5 Trusted Key Store

Again a complex password should be used to protect the Trusted Key Store.

4.6 Summary of Recommendations

- Disable Public Anonymous Account options, and use an LDAP guest account instead for public portal pages
- Enable secure connections for admin and users to force all traffic over SSL
- Use separate accounts for user administration, provisioning, compliance and roles administration, and protect each with a complex password
- Protect keys and key stores with complex passwords

5. JBoss

Jboss is a web application platform from Red Hat, and as such specific security configurations should be sought from Red Hat directly. There is a Wiki page on securing a JBoss server available from this link;

http://www.jboss.org/community/wiki/SecureJBoss

Where the wiki contradicts Novell recommendations then they should be disregarded in favour of the Novell IDM and User Application specific configurations already given.