Global Information Assurance Certification Paper

Securing OpenLDAP For User Authentication

Alfred Amos
January 18, 2004

GIAC Certified UNIX Security Administrator (GCUX)
Practical Assignment
Option 1, version 1.9

Table of Contents

1 Abstract

In a network environment, each host traditionally manages its own user authentication. Administration of the network may become difficult and labor-intensive. Security in such an environment is often lacking due to the distribution of authentication credentials and the use of multiple policies. One answer to this issue is to implement centralized authentication from a single server. The Lightweight Directory Access Protocol (LDAP) is open source software that can provide such a solution.

This implementation of the LDAP server provides user authentication on a private local area network. Applied security measures address concerns for access control, server authentication, host authentication, integrity and confidentiality, logging, auditing and software vulnerabilities. Discussions on other security issues include client authentication, availability, data recovery and physical security. The implementation described in this paper suggests OpenLDAP, with proper configuration, provides secure centralized authentication.

2 Notation

Commands executed at the command line will begin with a # symbol, if root privilege is required and a $ symbol for non-root privilege. If a command spans multiple lines, the standard UNIX notation of a back slash, \, will be used to indicate the continuation.

3 Introduction

This implementation of the LDAP server provides user authentication on a private local area network (LAN). The LAN is isolated from the Internet with an external firewall configured to filter out connection attempts from outside the network (see figure 1). The LDAP server installed will be the latest version of OpenLDAP. One objective is to keep the rest of the Red Hat system intact with minimal modifications necessary to implement a working and secure server.

The test configuration, due to the lack of resources, consists of one server and only one client host. This client runs an existing installation of the Linux operation system from the Red Hat 7.3 distribution with some upgrade patches applied from the Red Hat Web site. Modifications to this system will be minimal and restricted to adding the necessary configurations and required upgrades to operate securely with the server.

A novice administrator should be able to implement the LDAP server with the necessary security mechanisms by following the instructions below. The objective is to provide centralized authentication at least as secure as that exhibited with standard UNIX authentication for a single host. Discussion and implementation will focus on improving security where appropriate. In the following description on building, installing and configuring the LDAP service, the intent is to provide the important steps and interesting issues. Configuration is broken into several stages. The first step is to implement the basic configuration with minimal access controls to get it working. Subsequent stages implement Transport Layer Security (TLS), a minimum level of security for access, server authentication, host authentication and client filtering mechanisms. Testing of the

implementation follows each stage where appropriate. Further discussion will cover client authentication using certificates and other measures, which add to the security of the server. Configuration files included in the Appendix for reference should provide good examples and should be usable with a few modifications. The Appendix also shows screen dumps from certain commands used in this implementation.

4 Security Policy

Implementation of the LDAP server should be in accordance with an existing security policy. In the absence of such a policy, the "Server Security Policy" from the SANS site[1] can be adapted. Each of the sections on "Ownership and Responsibilities", "General Configuration Guidelines", "Monitoring" and "Compliance" have items applicable to the secure implementation of the server.

5 Description of System

5.1 Network Configuration

To help identify the components throughout this implementation refer to figure 1. The domain for the 10 Mbps Ethernet test LAN is mytestlan.com. The LDAP server has the fully qualified domain name (FQDN) of ormoc.mytestlan.com (192.168.20.10). The name of the client system is baybay.mytestlan.com (192.168.20.11).

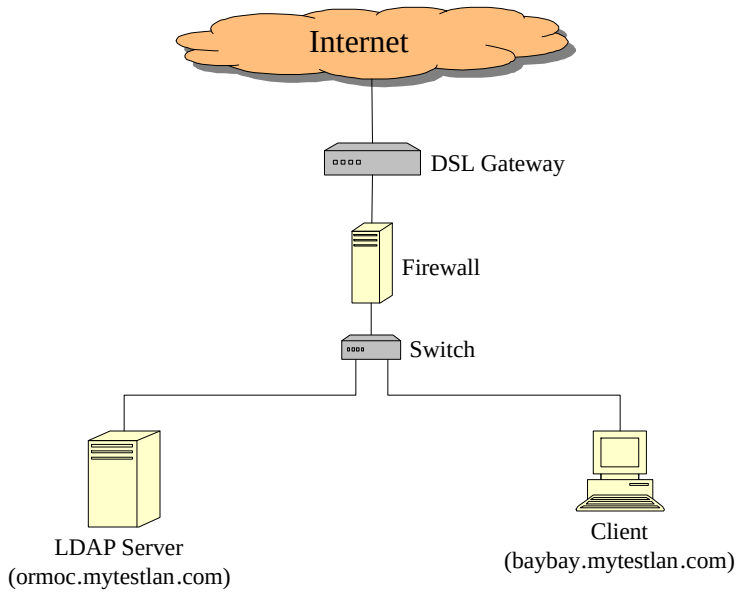

Figure 1 – LDAP Server Test Network

[1] "Server Security policy", http://www.sans.org/resources/policies/Server_Security_Policy.pdf.

The server and a single client make up the local network connected through the switch. The Firewall protects the LAN from Internet traffic. The LANs only connection to the Internet is at the Firewall through the DSL Gateway. Serving centralized authentication to a single client is not cost effective, but conceptually should be scalable to a number of clients for larger networks.

5.2 Hardware and Operating System

The server is installed on a Pentium II computer with a 266MHz CPU and 256 MB ram. The Linux operating system from the Red Hat 9 distribution is installed as an i686 configuration running the 2.4 version of the kernel. For this test implementation, disk space requirements include 2.5GB actually used with at least five GB recommended. More disk space may be required depending on the size of the database and the log files used for server operation. This OS is the latest stable Linux distribution from Red Hat and should be support for a longer time than the previous releases. The distribution also includes patches for security updates, bug fixes and enhancements encountered in the previous releases, requiring fewer upgrades. Security of the Red Hat 9 distribution may raise concerns, since it has not tested as well as some of the previous releases and may exhibit new bugs and vulnerabilities. The client host contains a Pentium II processor with a 350 MHz CPU. The operating system is an existing Red Hat 7.3 installation with relevant upgrade patches from the Red Hat site.

5.3 Final State of the System

The purpose of the server is to provide secure centralized user authentication in a private local area network environment. Authentication through the LDAP server will only be required for authorized users on the LAN and not for the root account or services. User accounts will exist on each client host, where authorized. Network File System (NFS) accounts will not implemented at this time, requiring the existence of user accounts on each authorizing host. If the LDAP server becomes unavailable for any reason, preventing users to log into their accounts, the root user can gain access through the standard UNIX authentication mechanisms. The LDAP server installed will be the latest from the OpenLDAP package, version 2.1.22. This package requires version 4.1 of the Berkeley database, which is an upgrade from version 4.0 of the database provided with the Red Hat distribution.

5.4 Implemented Security Measures

When accessing their Linux account, the standard way for users to authenticate is with passwords applied through the Pluggable Authentication Module (PAM). PAM provides authentication in clear text without any form of encryption. Centralized authentication allows anyone to view the unencrypted credentials sent over the network, raising security concerns. Application of Transport Layer Security (TLS) and Secure Socket Layer (SSL) mechanisms provide data integrity and confidentiality. TLS and SSL employ public key encryption with certificates. Another method of client authentication for non-repudiation involves the use of client certificates.

Through the Name Service Switch (NSS) a client host knows where to find the authentication server. In a secure system, assurances are necessary to verify that the

correct server is accessed. Otherwise, a spoof attack rerouting network traffic to a rogue server might compromise user's private authentication credentials. Certificates provide assurance to help prevent this type of attack.

On the server, there are several mechanisms to control access. Access Control List (ACL) rules specify what information can be accessed, who can access it, how it can be accessed and with what encryption level it can be accessed. Host authentication identifies the authorizing host for a particular user. An attempt to access a restricted host, results in abrupt termination logged by the server. TCP Wrappers and IPTABLES provide filtering mechanisms, respectively, for the LDAP service and server host with logging capabilities.

Other server security measures include the SYSLOG utility for logging user access information. Auditing this information can point to events indicating use and abuse of the systems on the network. Vulnerabilities from software defects are always a concern. To mitigate their effect the LDAP server runs as an unprivileged user, using a non-interactive invalid shell. A vulnerability that might normally yield a shell with root privilege should not produce a shell at all and certainly not with root privilege.

Many unnecessary services and applications may be running on a system. X-Windows, at run level 5, is a typical Linux service that opens up ports to communicate with other X-Window systems. The LDAP server does not need this service and there is no need to open these ports for possible attack. To remove this concern and close these ports the server will boot into run level 3.

6 Risk Analysis

6.1 Issues

One major concern with the LDAP server is the password that must remain secret to maintain access control over the database. As will be seen later, losing the password will compromise the whole system. The problem with this password is that it needs be resident on the disk in clear text. The LDAP manager may be a user other than root for managing the data in the database and does not require root access. The manager is privileged to change a user's password, add accounts and delete accounts. Full access to the information in the database is required and the manager has it through use of the LDAP tools and the LDAP password. The root user on the server also needs to change user's passwords, but does not require write access to other database information. If the LDAP password stored on the server host, preventing access to the database by root may be pointless.

An expectation from a user's perspective is that their authentication credentials remain confidential. Users must change their own password, but no other user's password. Users require read access to their account information. They probably do not need access to other portions of the database. Likewise, users may not require access to all hosts on the network and may be restricted from using certain systems. Server and client configuration help enforce measures that accomplish these goals. The user must

also have assurance that they are accessing the correct LDAP server. To maintain productivity, users must be able to access their accounts and applications on network hosts. This requires the LDAP service availability. Many factors may affect availability including network attacks, maintenance and performance issues.

6.2 Required Services

Only a few services are required for this server. The LDAP service uses two ports. The ldap TCP port 389 is the default port for receiving clear text and TLS encrypted transactions. The ldapssl TCP port 636 is the default to receive SSL encrypted transactions. To provide a secure connection from a remote client, the SSH service will be run on the default TCP port 22. This is not required if all the work will be done from the console, but it may be useful when accessing the server from a client host. IP Tables will provide a personal firewall to filter access to the server host. The built-in TCP Wrappers filter access to the service and do not require the use of the tcpd daemon. For a large network, access to a DNS server will also be required.

6.3 Key Security Concerns and Threats

6.3.1 Access Control

The threat of unauthorized access to database information, remote hosts on the network and configuration files on the server can have severe consequences, resulting in compromised systems, corrupted data and data loss. Private database information includes user passwords. Restricted portions of the database may contain proprietary and confidential information. Controlled access to certain hosts, such as servers, is mandatory. Many files on the server need access controls to ensure a secure environment.

6.3.2 Server, Client and Host Authentication

An attack using a rogue server may result in misdirected authentication requests. Compromise of the whole system is possible. Access to the correct authentication server is mandatory for maintaining a secure system. All users accessing the LDAP server and other hosts must authenticate and be authorized users.

6.3.3 Data Integrity and Confidentiality

The threat of a man-in-the-middle attack capturing authentication credentials raises a serious security concern. Encryption is a common way of reducing this threat, providing data integrity and confidentiality. Consequences for failing to encrypt network traffic may include unauthorized access to clients and servers, resulting in a compromised network of systems.

6.3.4 Logging and Auditing

The absence of logging and auditing mechanisms and procedures presents a serious threat. A malicious attack may be unnoticed, until it is too late. Logging and auditing the server connection attempts might catch abuse from an attacker, either inside or outside the network. Logged operations provide clues of what the attacker was doing. Results

recorded in the log provide information on whether the attack was successful. This can show how the system is being used or abused, aiding a forensics analysis or helping to discover configuration deficiencies.

6.3.5 Software Vulnerabilities

Software vulnerabilities present a non-specific threat for which there may be no direct defense. Implement measures to help guard against an exploit, without knowing the nature of the exploit. Proper server setup helps in this regard.

6.3.6 Other Concerns

Other important concerns that are worth mentioning include availability, data recovery and physical security. Service interruption results if authentication data in the LDAP database is corrupted or destroyed. Denial of service attacks also may prevent legitimate users from accessing the client systems. In the event of data corruption or loss, a planned data recovery procedure is critical to avoid financial loss. Implement physical protections at the console, especially to overcome some of the default insecurities. If an attacker gains direct access to the server console, the whole system may be lost.

7 LDAP Implementation

When building and installing the server it is a good idea to isolate the server and client hosts on their own test network and disconnect the test network from the rest of the networks. One good way to do this is by disconnecting the Ethernet cable at the router. Keep in mind that a DNS server may also be required for hostname resolution.

7.1 Operating System

The Linux installation for this implementation was from CDs containing the Red Hat 9 distribution. When installing from CDs there is a prompt, at the beginning, asking whether or not to test the media on the CD. It is a good idea to test each CD at least the first time. The installation is straightforward and provides online help messages throughout the process.

Perform the installation of a new operating system in accordance with the security policy. Do not install unnecessary services and applications. Their removal will be required later. Select a custom installation setting a static IP address, hostname, gateway and DNS servers. A static IP address is required so all the clients hosts can locate the server. Restrict access to the server host by setting the firewall at the high security level. Allow incoming TCP traffic, SYN packets, for the service SSH port 22. Also add the LDAP port 389 and LDAPS port 636 for incoming traffic. The firewall can also be set up later with the lokkit utility. Other incoming service ports are not required. With DNS servers, rules allowing input from the UDP port 53 will be set. Continuing with the operating system installation, enable MD5 and shadow passwords. Select the defaults plus KDE Desktop, Editors, Server Configuration Tools, Development Tools, Administration Tools, and System Tools.

If problems appear with traffic not getting through to the server, clear the firewall rules until after the configuration with the command

```
# iptables –F
```

Be sure the host is on a trusted network before doing this. For more information, see the section on “IP Tables” below. To verify the firewalls are set up properly display the rules with

```
# iptables –nL
```

These commands for iptables require root privilege. After the system installation, create some user accounts on the server. For this implementation, create user accounts with the usernames ajones and jsmith. Use a supplied GUI tool or the command line tools, useradd and groupadd, for creating new accounts. Choose a UID and GID for each user that is unique on the server and all clients.

7.2 Getting, Building and Installing the Software

Download the required software packages from various sites on the Internet. There is no full proof way of knowing if a package has been tampered with, but there are ways to get a reasonable assurance that a package is authentic. These methods use cryptographic hashes for integrity and digital signatures for both authenticity and integrity. Available files on Internet sites may have both, one or none of these. In any case, make sure it is from a trusted site. The signature key might also exist in a public key repository[2] for further verification and trust. To verify just the integrity use either of the two commands

```
$ md5sum –c MD5SUM or
$ md5sum <filename>
```

In the first command, the file MD5SUM contains the cryptographic hash and corresponding filename. The hash created for the specified file is compared it to the hash in MD5SUM. The second produces a hash of the file, which can be compared to the hash provided on the Web site. On an installed Red Hat distribution the public signature key should also be available in the file /usr/share/rhn/RPM-GPG-KEY on disk. With root privilege, add the RPM-GPG-KEY public key to the rpm database with the command

```
# rpm --import /usr/share/rhn/RPM-GPG-KEY
```

Display all imported public keys in the rpm database type

```
$ rpm -qa gpg-pubkey*
```

[2] “MIT PGP Public Key Server”, http://pgp.mit.edu/.

After importing the key, check the digital signature of the file <filename>.rpm with

```
$ rpm --checksig –v <filename>.rpm
```

Some of the packages used in this implementation did not have a cryptographic hash or signed file. In this case, establish a reasonable assurance of trust by getting the software from the owner's Web site with referrals from well known trusted sites. Since downloading the software and verifying their integrity and authenticity can be a lot of work, it is good practice to burn and label a CD containing the packages and any relevant notes. A system rebuild due to unforeseen circumstances, such as a compromised server, may require repeating this whole procedure.

Before building the software, upgrade relevant packages for vulnerability fixes, bugs fixes and enhancements. At least check for the latest updates for openssl, nss_ldap, openssh and glibc. The package nss_ldap includes LDAP libraries for NSS and PAM. The glibc package is always good to upgrade when available. Many applications use the glibc package. If available, download the rpm files from the Red Hat general advisories site[3] and verify their authenticity and integrity before installation. Be sure to review the installation instructions on the download site. The upgrade packages found in the repository with previous versions currently installed are

- openssl-0.9.7a-20.i686.rpm,
- openssl-devel-0.9.7a-20.i686.rpm,
- glibc-2.3.2-27.9.7-i686.rpm and
- glibc-common-2.3.2-27.9.7-i386.rpm.

In compliance with the security policy, download and build the software packages from a non-privileged user account. The only part that should require root privilege is the installation. After the server implementation and configuration, it is good practice to create and label a backup of the installation on appropriate media, pursuant with the security policy. Making the backup before placing it in operation, it will provide a known clean checkpoint.

7.2.1 Red Hat Supplied Tools

The Red Hat distribution supplies tools required to build the server and supporting software. OpenSSL, version 0.9.7a provides the libraries for SSL and TLS encryption. This package also creates the required certificates. PERL is necessary to run the migration scripts. The gcc compiler builds the applications.

After building and configuring the server, set up an LDAP boot script to start the service automatically at system startup. The Red Hat distribution already provides a script, in the package openldap-servers-2.0.27-8.i386.rpm. To quickly check if it is already installed, look for the /etc/rc.d/init.d/ldap file. If not there, the following procedure will get

[3] "Red Hat Linux 9 General Advisories", https://rhn.redhat.com/errata/rh9-errata.html.

the ldap script without installing the rest of the files. Extract the file using the rpm2cpio and cpio tools.

```
$ rpm2cpio openldap-servers-2.0.27-8.i386.rpm \
        | cpio -i --make-directories ./etc/rc.d/init.d/ldap
```

Once it is extracted, move the ldap script to the directory /etc/rc.d/init.d/. This extraction should be done as a non-privileged user, but copying and editing the file requires root privilege. After copying the file check that the owner is root and permissions are 755. Now, modify the ldap script to point to the location where the new version for the slapd, slurpd and slapd.conf files will be installed.

- Change /usr/sbin/slapd to /lusr/local/ldap/libexec/slapd.
- Change /usr/sbin/slapd to /lusr/local/ldap/libexec/slurpd.
- Change /etc/openldap/slapd.conf to /usr/local/ldap/etc/openldap/slapd.conf.

This rpm package also contains schema files, migration tools and OpenLDAP server files. Schema files provided by the OpenLDAP package are sufficient for this implementation, but additional files from the rpm package may provide additional attributes. There is no need to use the migration tools and server files in this package, as newer versions will be downloaded later.

7.2.2 Berkeley Database

The Berkeley database version 4.1.25 with strong cryptography is the latest version, as of this implementation. The OpenLDAP package recommends this database and requires version 4.1. The installation of Red Hat 9 only provides version 4.0 of the Berkeley database. The package db-4.1.25.tar.gz, along with a patch, patch.4.1.25.1, is available from Sleepycat Software[4]. No cryptographic hash or signature file was found for these files.

As a user, without root privilege, expand the database package with

```
$ tar -xzvf db-4.1.25.tar.gz
```

Change directory to the root directory of the distribution, db-4.1.25 and apply the patch

```
$ patch –p0 < patch.4.1.25.1
```

In each of these steps, capture the screen output. Look for errors and assurance that the outcome is successful. The following directory references are relative to the distribution root directory db-4.1.25.

To configure the database, follow the instructions in the file docs/index.html. Configure and build the software in the directory build_unix. There are several configuration

[4] Sleepycat Software, http://www.sleepycat.com/download/index.shtml.

options available, but for this implementation use the defaults, except for defining the prefix. This option sets the installation directory. The command to configure the build is

```
$ ../dist/configure --prefix /usr/local/bdb.4.1.25
```

The file config.log will contain information on the configuration.

After a successful configuration, build the software with the make utility as described in the docs/index.html page. The default is to use static libraries. If the build is successful, install the database with root privilege, with the command

```
$ su –c 'make install'
```

Installation will be in the directory /usr/local/bdb.4.1.25 and documentation can be found in /usr/local/bdb.4.1.25/docs/. All files under the directory /usr/local/bdb.4.1.25 must be owned by root and provide no world write access. The user owns some of the files copied with this command, so be sure to check the owner and permissions.

7.2.3 OpenLDAP

The source package for OpenLDAP version 2.1.22 is available from the OpenLDAP download site[5]. Download the files openldap-2.1.22.tgz and openldap-2.1.22.md5. The file with the extension md5 contains the cryptographic hash of the tar package. Use this hash to check the integrity of the package file. Building OpenLDAP from the source is required to customize the configuration and add security features. The Administrator's Guide[6] is very helpfully toward understanding the various parts of the server and getting the correct configuration. The build will also generate the server and client tools. The server tools are mainly for backing up and restoring the database, indexing the entries for better performance and generating passwords. Client tools search the database and change the user password, as well as add, remove and modify entries.

Configure, build and test this server as a non-privileged user. In a working directory, expand the package with

```
$ tar -xzvf openldap-2.1.22.tgz
```

The source and instructions are available in the directory openldap-2.1.22/. The README file identifies required software, how to get support and points to other important files included with the package. Verify the installation of the required and recommended software with the correct versions. Often a higher version will also work. The ANOUNCEMENT file includes information on the enhancements and components provided, in addition to license information. The INSTALL file describes how to build the server. Pay close attention to the environment settings and options to configure the build. In each of the following steps, check the output files for errors and assurance that the outcome is successful.

[5] OpenLDAP Download Site, http://www.openldap.org/software/download/.
[6] "OpenLDAP 2.1 Administrator's Guide", http://www.openldap.org/doc/admin21/.

From the distribution root directory, openldap-2.1.22, configure the build

```
$ env  CPPFLAGS="-I/usr/local/bdb.4.1.25/include -I/usr/kerberos/include" \
       LDFLAGS="-L/usr/local/bdb.4.1.25/lib" \
  ./configure --enable-wrappers --with-tls --enable-syslog \
       --enable-crypt --prefix=/usr/local/ldap
```

The command executed is “configure”. The environment variable settings start with the “env” key word. The variable CPPFLAGS, points to the include files for the Berkeley database and the location for the krb5.h file. The LDFLAGS variable points to the library files for the database. Kerberos is not being used in the server, but it is necessary to include the /usr/kerberos/include/krb5.h file to find the Red Hat supplied OpenSSL components. Following the configure command are several options for customizing the server. TCP Wrappers are integrated into the server. This may cause slow performance due to the additional filtering. SSL and TLS provide integrity and confidentiality through encryption. SYSLOG will support logging messages from the server. This is useful for both debugging the configuration and operational use. Crypt enables the use of the OpenSSL cryptographic library, crypto, and the password encryption function crypt(). The application will be installed in the directory /usr/local/ldap. This directory was chosen to separate it from other applications that might be placed in /usr/local/. A listing of all the options is displayed with the command

```
$ ./configure –help
```

The server is more secure when compiled with static libraries, as opposed to linking with dynamic libraries. Use of static libraries, the default, avoids the possibility for altering or disrupting the server operation by swapping out the shared libraries. The down side to this is a bigger executable and the necessity to rebuild the LDAP application when upgrading, instead of just rebuilding the shared libraries. The build generates both library types by default. Other applications may use the LDAP shared libraries.

After a successful configuration, build the server.

```
$ make depend; make
```

Once the build is satisfactory, there are a number of built in tests for accessing the Berkeley database in various ways. All of the tests numbered 0 through 16 should return the result of “OK” for the bdb backend.

```
$ make test
```

Finally, install the server with root privilege.

```
$ su –c ‘make install’
```

The installation will be in /usr/local/ldap as configured and all of the server files will be relative to this directory. The build generates the LDAP server (slapd) and replication server (slurpd) and places them in the directory named libexec. Default configuration files are located in the /usr/local/ldap/etc/openldap directory. Access the manual pages, installed in the directory man/, for slapd using the command

```
$ man -M /usr/local/ldap/man slapd
```

Server tools can be found in the /usr/local/ldap/sbin directory and client tools are in /usr/local/ldap/bin/. Remove other versions of these tools that exist on the system to avoid confusion.

OpenLDAP includes several database backends that perform the database transactions. The two more commonly used backends are LDBM and BDB. LDBM, the default in Red Hat 9, does not take advantage of all the features in the Berkeley database. BDB is the default in OpenLDAP version 4.1.25 and takes advantage of most of the Berkeley database features.

7.2.4 Migration Tools

As a server used in a network with existing users, the authentication information may already exists in the passwd, group and shadow files. An easy way to convert the data from these standard files to the LDAP database is to use the migration tools[7]. These tools are a collection of PERL scripts that convert data from many more system files than just these three, but for user authentication, these are the only necessary files. These scripts convert the data in the passwd, shadow and group files to an intermediate format called the LDAP Data Interchange Format (LDIF). Insert these LDIF files into the database using the LDAP client tools. Download the most recent version of the migration tools, in the file MigrationTools.tgz, from the PADL Software site[8]. An older version of these tools also resides in the openldap-servers-2.0.27-8.i386.rpm package. No cryptographic hash or signature file was found for the Migration Tools. Since they are PERL scripts, they can be examined as needed.

Building the migration tools is not required, since they are PERL scripts. Just expand the package into a user directory. The directory MigrationTools-45 will be created and contain all of the PERL scripts. The name of this directory may be different if using another version of the migration tools. Change to the MigrationTools-45 directory and read the README file that contains useful information on using the scripts. It is necessary to modify a few variables in the migrate_common.ph file, as follows.

- $DEFAULT_MAIL_DOMAIN = “mytestlan.com”
- $DEFAULT_BASE = “dc=mytestlan,dc=com”
- $DEFAULT_MAIL_HOST = “mail.mytestlan.com”

[7] Migration Tools site, http://www.padl.com/OSS/MigrationTools.html.
[8] MigrationTools download site, http://www.padl.com/download/.

7.2.5 Sniffer (ngrep)

Testing for clear text and encrypted traffic on the network and on the local interface for the server requires a sniffer, such as ngrep. The source package ngrep-1.41.tar.bz2 is available from the Packetfactory Web site[9]. Download the manual page for this tool from the same site. Expand the package with the tar utility and the, -j, option to filter the archive through bzip2.

View the README file for an explanation on how to use it and the INSTALL file on building the tool. Configure and make the executable in the directory ngrep-1.41 with the command

```
$ ./configure; make
```

Place the generated ngrep executable in a convenient location for access, such as /usr/local/sbin. Verify that the owner and permissions restrict access according to the security policy. Analysis of the source code is possible if there is concern about having a Trojan copy.

7.3 Basic Configuration

7.3.1 Preliminary Details

If a DNS server is not available, the IP addresses and the FQDN for the server and clients referenced in the database must be listed in the /etc/hosts file. This allows resolution of the IP address for the FQDN used in the host attribute, configuration files and the certificates. This will work for testing on a small network, as was done for this implementation, but for large network environments a DNS server is necessary.

As for any service on the Linux operating system, create an account for the LDAP server. This requires unique names and identifiers for the LDAP user and group. Set an invalid password to prevent login access. Also, grant an invalid non-interactive shell type. Entries for the passwd, group and shadow files are as follows.

```
passwd file entry

ldap:x:81:81:LDAP User:/var/lib/ldap:/bin/false

group file entry

ldap:x:81:

shadow file entry

ldap:!!:12385:0:99999:7:::
```

[9] "ngrep – network grep", http://www.packetfactory.net/projects/ngrep/.

In the passwd file, the shell /bin/false is an invalid non-interactive shell setting that should never yield an interactive session to a user. The file /etc/shells contains all the valid shells available to the operating system. Since this shell is not in that file, the system will not recognize it. This has no effect on the LDAP operation since the server never requires a shell, but it may prevent access through an exploit against vulnerabilities from a software defect. However, this may not be the case if the defect is in the /bin/false shell. For the shadow file the password field contains the characters "!!" which is an invalid representation for a password.

7.3.2 Database Structure

Before continuing with the server configuration, a rudimentary understanding of the LDAP database structure is in order. Figure 2 shows a simple architecture created for the implementation and demonstration of the LDAP server. For the actual system, a more complex structure to fit the network will be required.

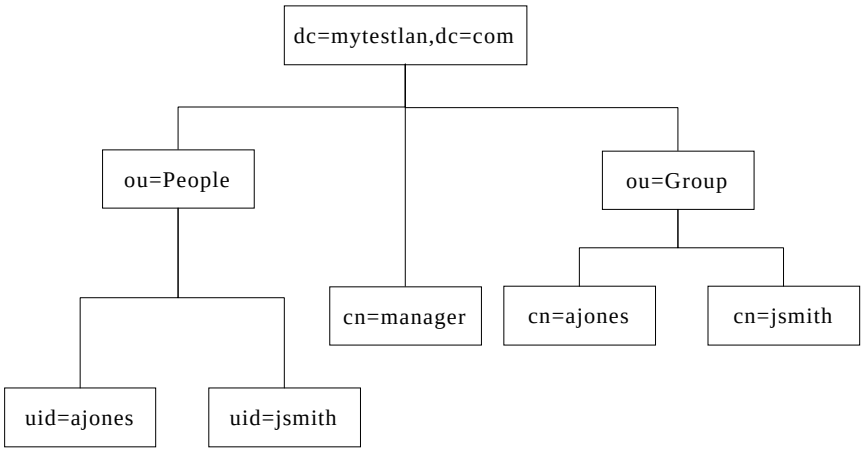

Figure 2: LDAP Directory Structure

In this architecture, a Distinguished Name (DN) represents each node. The structure is similar to the Linux directory structure, where the DN is comparable to the full directory path in Linux. For example, the user ajones is represented with the DN 'uid=ajones,ou=People,dc=mytestlan,dc=com'. The group ajones has the DN of 'cn=ajones,ou=Group,dc=mytestlan,dc=com'. All user information resides under the 'ou=People' node and all group information is under the 'ou=Group' node. Each DN entry in the database should be unique to avoid confusion. One difference from the Linux directory structure is the order for declaring the nodes. The DN starts at the bottom of the tree identifying each node in the chain up to the root. In this case, the root node is 'dc=mytestlan,dc=com'. The Linux file structure starts at the top, which is root (/), and works its way down the tree. Another obvious difference is the use of LDAP attributes, such as uid and cn, with an equal sign and the nodes separated by commas.

In Linux, a forward slash separates the directories. For a more detailed description, see the “OpenLDAP 2.1 Administrator’s Guide”.

7.3.3 Schemas

To better understand the LDAP architecture, it is necessary to discuss schema files. Schema files contain structural objects called “objectclass”. These objects contain attribute types. The key word “attributetype” defines each attribute type in a schema file. An LDAP entry includes attribute types with assigned values and the object class to which it belongs. For an example, look at the following objectclass definition for posixGroup and the attributetype definition for gidNumber from the nis.schema file.

```
excerpt from the nis.schema file

objectclass ( 1.3.6.1.1.1.2.2 NAME 'posixGroup' SUP top STRUCTURAL
        DESC 'Abstraction of a group of accounts'
        MUST ( cn $ gidNumber )
        MAY ( userPassword $ memberUid $ description ) )

attributetype ( 1.3.6.1.1.1.1.1 NAME 'gidNumber'
      DESC 'An integer uniquely identifying a group in an administrative domain'
      EQUALITY integerMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

To create a group id for a user, assign a value to the attribute type gidNumber (e.g. gidNumber: 600). The “MUST” keyword identifies attributes that must be included if the posixGroup objectclass is used. Therefore, assigning a value to the cn attribute type, such as the group name (e.g. cn: ajones), is also required. The “MAY” key word identifies attribute types that are not required, but may be set in an LDAP entry. For a more detailed description, see the “OpenLDAP 2.1 Administrator’s Guide”. Look at the LDAP Data Interchange Format files in Appendix C, for more examples.

7.3.4 Server Configuration

The basic server configuration is customized in the slapd.conf file in the directory /usr/local/ldap/etc/openldap/. When the OpenLDAP package was installed it created a basic slapd.conf file. Modify this file as needed to reflect the changes described below. The general layout for this file consists of sections on global configuration directives and general database directives. There can be multiple general database sections, but this implementation will only contain one. The following discussion describes key parameter settings for this implementation. More options are available in the OpenLDAP Administrator’s Guide. Refer to the final configuration for slapd.conf in Appendix A. The basic configuration is the same except for the TLS options at the end of the file and the ACL security strength factors (ssf). The remaining portions of the ACL rules still apply.

In the global configuration section, the required schema files are included first. An optional setting for the log level of –1 will log everything through the SYSLOG utility.

This can be useful for debugging the configuration, but it will also flood the log file with messages not useful for a production environment.

```
# Global configuration directives

include         /usr/local/ldap/etc/openldap/schema/core.schema
include         /usr/local/ldap/etc/openldap/schema/cosine.schema
include         /usr/local/ldap/etc/openldap/schema/nis.schema
include         /usr/local/ldap/etc/openldap/schema/misc.schema
include         /usr/local/ldap/etc/openldap/schema/inetorgperson.schema

loglevel -1
```

A loglevel value of 256 may be more appropriate for an operational LDAP server, logging messages for connections, operations and results. This still puts out many messages to the log file. The global configuration section can also contain Access Control List (ACL) rules, which are processed only after the rules in the database sections. These rules are useful for general access controls that are not database specific. For this implementation, all ACL rules are database specific.

Parameters for the configuration in the general database section are listed below.

```
# General database directives

database bdb
suffix dc=mytestlan,dc=com

rootdn          "cn=manager,dc=mytestlan,dc=com"
rootpw {SSHA}CPfU1xQ0opiHWTP9TDtrZZu0YvSqDhle

directory       /usr/local/ldap/var/openldap-data
mode 0600

index  objectClass  eq
index cn,uid  pres,eq,sub
index uidNumber,gidNumber,memberUid  eq

# Database specific ACL rules
```

The database type, bdb, refers to the Berkeley database. The suffix parameter is the DN that corresponds to the domain name of the server. The domain mytestlan.com translates to the suffix dc=mytestlan,dc=com. The rootdn parameter is the DN for the manager of the LDAP server. The DN for the manager is required when adding database entries and other operations with the LDAP tools. A password, specified with the parameter rootpw, is required when accessing the database as the manager. Specify this password in clear text, as “rootpw mysecret”, or in an encrypted form, which

will be more secure. If encrypted, use the server tool slappasswd to create the password hash. The various types of password hash generated include Crypt, MD5, SMD5, SHA and SSHA the default. Descriptions for these exist in the LDAP manual page for slappasswd. To generate an SSHA encrypted password string use the command

```
# /usr/local/ldap/sbin/slappasswd -h {ssha}
```

The hash type created uses the SHA-1 algorithm with a seed. Copy and paste the encrypted password displayed on the screen into the value for rootpw. Using the DN for the manager with the LDAP password, bypasses all access controls in the ACL rules. This can become a critical breach of security if the password does not remain secure. One way to help keep the password secure is to comment out the rootdn and rootpw parameter assignments not allowing access to the manager. The manager can gain access again by reactivating these configuration lines and restarting the server.

The directory parameter identifies where the database resides. The mode parameter specifies the permissions for newly created database index files in the database directory. The default and recommended value is 0600, for owner read and write access, but it is good to explicitly specify it so there is no question of what the value is. The index parameter, used with the server tool slapindex, indexes the database for faster searches. Use slapindex with caution. To avoid corrupting the database, the server must not be running. This is the basic configuration. For more details on options for the bdb database section, see the manual pages for slapd-bdb.

Before running the server, check the ownership and permissions for the configuration file and directory. A listing should show something like the following.

```
# ls -l

-rw-------   ldap ldap  ….    /usr/local/ldap/etc/openldap/slapd.conf
drwx------  ldap ldap  ….    /usr/local/ldap/var/openldap-data
```

The slapd.conf file should only be accessible by the owner, ldap, with read and write permissions. The openldap-data directory should also have the ldap owner and group settings, with permissions set to read, write and execute by owner. These settings are critical, since the server will run as the ldap user.

The startup script, /etc/rc.d/init.d/ldap, has already been installed and modified, but make sure the service runs as the ldap user by specifying the, –u, option for slapd in the ldap script (e.g. slapd –u ldap). To configure the system to launch the LDAP server on system startup, add ldap as a new service for management.

```
# chkconfig --add ldap
```

Now set the startup script to launch for run levels 2, 3 and 5.

```
# chkconfig --levels 235 ldap on
```

Start the service from the command line.

```
# service ldap start
```

Other useful options with the service command include status, restart, and stop. Another option is to start the server at the command line with the command

```
# /usr/local/ldap/libexec/slapd –u ldap
```

Root privilege is required for these commands. If the server does not run, revisit the configuration, permissions and ownership of the files and directories.

The database does not yet have data, but use the LDAP client tool, ldapsearch, to verify connectivity.

```
$ ldapsearch –x –H ldap://localhost/ -b ‘dc=mytestlan,dc=com’ ‘(objectclass=*)’

# extended LDIF
#
# LDAPv3
# base <dc=mytestlan,dc=com> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 32 No such object

# numResponses: 1
```

The resulting screen display should look similar to that above.

7.3.5 Migrating Accounts

The data placed into the LDAP database for authentication is the same information that is in the passwd, group and shadow files. To migrate existing and new user account information to the server, the LDAP Data Interchange Format (LDIF) is used. Change to the MigrationTools-45 directory, where the migration tools are installed. The execution of these tools must be done in this directory. With root privilege, execute

```
# ./migrate_passwd.pl /etc/passwd passwd.ldif
```

All information in the passwd and shadow files will be represented in the file named passwd.ldif. If root privilege is not used, the information stored in the shadow file, such as the encrypted password, will not be transferred. This LDIF file will also contain information for the root account and all the services that are listed in the passwd file. Now is a good time to edit passwd.ldif and extract all users and services that are not to be placed into the LDAP database.

The group information from the group file can be migrated with

```
$ ./migrate_group.pl /etc/group group.ldif
```

This command can be executed from a non-privileged user account, since all the data is accessible. Information on groups for all services and accounts listed in the group file will be migrated to the group.ldif file. Again, edit this file to extract all but the desired user group information.

The DNs in these two LDIF files represent the bottom nodes in Figure 2. Before adding these entries to the LDAP database, all preceding nodes shown in the figure must exist in the database. This requires creation of another LDIF file defining all of the parent node entries. This file named mytestlan.ldif is listed in Appendix C along with the other LDIF files used.

The migration tools create files with the basic necessary entries. Additional items may be added or removed, but must be consistent with the schema files. For example, the host attribute will be added later to provide host authentication. The LDIF file consists of sections separated by blank lines. Each section contains object classes and attribute/value pairs defining a particular user or group.

Before inserting these entries into the database, the server must be running. Insert the entries from the LDIF files into the LDAP database with the ldapadd client tool. The three commands for adding these entries are listed below.

```
$ ldapadd -x -D 'cn=manager,dc=mytestlan,dc=com' -W -f mytestlan.ldif
$ ldapadd -x -D 'cn=manager,dc=mytestlan,dc=com' -W -f passwd.ldif
$ ldapadd -x -D 'cn=manager,dc=mytestlan,dc=com' -W -f group.ldif
```

Simple authentication must be specified with the option, -x. The Distinguished Name to bind, as manager, is 'cn=manager,dc=mytestlan,dc=com'. For simple authentication, a password is required. The option, -W, forces a password prompt. This password is the value of the rootpw parameter specified in the slapd.conf file. The LDIF file containing the entries is specified with the option, -f. When entries are added or modified, the schema files are checked by default for valid entries. That is, the object classes and attributes must exist in the included schema files. If not found, then an error is thrown and processing stops at that location in the LDIF file. Additions up to that spot are added, but partial entries for a user are not. If this becomes a problem, setting the parameter schemacheck to off in the slapd.conf file in the global configuration section

can disable the checks. These problem entries should be resolved to conform to the definitions in the schema files and the schemacheck parameter turned on or commented out.

Add new users manually by creating the appropriate LDIF file. Follow the same pattern of object classes and attributes in the passwd.ldif and group.ldif files. Then change the values for the attributes for the new user, accordingly. Remember to change the user password. Another way is to use a tool like the GQ LDAP client[10]. This graphical utility, based on GTK, supports TLS and makes administration easier.

After adding users, test the server again by displaying all the information placed into the database on the screen.

```
$ ldapsearch –x –H ldap://localhost/ -b ‘dc=mytestlan,dc=com’ ‘(objectclass=*)’
```

7.3.6 SYSLOG

Before configuring and testing the client, a mechanism for logging debug information is needed. The SYSLOG utility is useful for this purpose, as well as logging access information during normal operation. All commands in this section require root privilege.

To configure SYSLOG edit the file /etc/syslog.conf. OpenLDAP writes to local4 by default. Set to debug level of output by adding the following line to the end of the file.

```
local4.debug            /var/log/ldap
```

The file /var/log/ldap must be created manually and permissions set for read and write only by root.

```
# touch /var/log/ldap
# chmod 600 /var/log/ldap
```

Restart the syslog service with the service command

```
# service syslog restart
```

All files that SYSLOG writes to can be display with the lsof command. Look for the ldap file for verification.

```
# lsof –c syslogd /var/log | grep ldap

syslogd   539    root    7w   REG      3,29   0  32106 /var/log/ldap
```

[10] “GQ LDAP Client”, http://biot.com/gq/.

The displayed output should show a line similar to the one above, where the command is syslogd and the file name is /var/log/ldap. The level of output logged to the ldap file is set with the loglevel parameter in the file slapd.conf as described in the basic server configuration.

Configure the logrotate utility to rotate the ldap log, as is done with other logs. In the first line of the /etc/logrotate.d/syslog file add /var/log/ldap with the other log file names. The logrotate program should already be set up to run daily as a cron job, so there is no service to restart. Test the log rotation before proceeding to the client configuration.

```
# logrotate -f /etc/logrotate.conf
```

This will rotate all logs. Verify the rotation configuration by checking for the existence of the file /var/log/ldap.1.

7.3.7 Client Configuration

The Pluggable Authentication Module (PAM) provides authentication and the Name Service Switch (NSS) directs authentication requests to the LDAP server. By default, authentication credentials (i.e. password) are sent in the clear. For PAM to provide secure transactions with TLS, the pam_ldap.so library must be built with the OpenSSL libraries. In the recent Red Hat distributions, the library is already TLS enabled.

The following description applies to each host used as a client, including the server. The easiest way to set up the configuration for PAM is with the authconfig tool. This tool will produce two screens. On the first screen, select the options “Use LDAP” and “Use TLS”. Then enter the FQDN for the server host (e.g. ormoc.mytestlan.com) and the Base DN (e.g. dc=mytestlan,dc=com). Select “NEXT” to proceed to the second screen. On this screen, select options “Use Shadow Passwords”, “Use MD5 Passwords”, “Use LDAP Authentication”. Select “OK” to finish.

The authconfig tool creates or modifies several files. The file /etc/pam.d/system-auth is rewritten, with included lines referencing the pam_ldap.so library. This library enables the application of TLS and SSL encryption to transactions generated from compliant utilities, such as login, passwd, ssh and su. In the Red Hat distribution, PAM configuration files for these and other utilities reference the system-auth configuration. In Redhat 9, each line in this file contains a reference to the environment variable $ISA, as part of the path definition. This could enable an intruder to reroute the library references by replacing the $ISA environment variable. The associated risk may be acceptable, provided the environment variable setting is monitored regularly. Another option could be to remove it from the system-auth file, as well as from other files in the /etc/pam.d directory. Appendix B lists the configuration file, system-auth.

Another file modified is the NSS configuration file, /etc/nsswitch.conf. References to ldap are added for the passwd, group and shadow files, as well as other services. Edit this file to remove ldap references from lines where it is not needed. This will ensure that under normal conditions the LDAP server is only used for the authentication

database searches. The configuration file may also include references to the NIS+ service. Since NIS+ is not running, remove the nisplus references from these lines in the file. The required lines are as follows.

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

The authconfig tool also creates or modifies another NSS configuration file /etc/ldap.conf. It was interesting to find that the only documentation found for this file is within the comments of the file itself. Modify this file with the changes listed below. The final configuration for /etc/ldap.conf can be found in Appendix B.

```
host ormoc.mytestlan.com
base dc=mytestlan,dc=com

# Server only
rootbinddn cn=manager,dc=mytestlan,dc=com
# Server only – Password is stored in /etc/ldap.secret (mode 600)
```

The parameter, host, is the FQDN of the server. This tells NSS where to find the LDAP server. The parameter, base, is the DN for the search base and the same as the suffix in the slapd.conf file. The parameter rootbinddn identifies the distinguished name to bind the LDAP server. This is the same name used for rootdn in slapd.conf. The password used for the value of rootpw in the slapd.conf file must be placed in a file on disk with the default name /etc/ldap.secret. This password must be stored in clear text with a return entered after the password. To keep it secure, set the file owner to root with read and write privilege by root. Anyone with this password has total control over the LDAP database by using the client and server tools. If this becomes a concern, remove the secret file. However, doing so, removes the privilege for the root user to change the password for other users, using the standard UNIX command, passwd. An alternative to this command would be to use the ldappasswd tool, as the manager. The configuration for rootbinddn and /etc/ldap.secret should only be set on the server and not the remote client. The remote client should not need either of these values.

Other configuration settings include the NSS filters identifying where to find the account, password and group information in the LDAP database.

```
scope one
nss_base_passwd ou=People,dc=mytestlan,dc=com?one
nss_base_shadow ou=People,dc=mytestlan,dc=com?one
nss_base_group ou=Group,dc=mytestlan,dc=com?one
```

PAM requires mapping of the objectclass and attribute for username, group id and password type through the following configuration.

```
pam_filter objectclass=posixAccount
pam_login_attribute uid
pam_member_attribute gidNumber
pam_password md5

# ssl start_tls
```

Finally, comment the line “ssl start_tls” to test the configuration without TLS. This setting is only required with the implementation of TLS.

OpenLDAP requires a client configuration file also named ldap.conf. Don’t confuse this with the NSS configuration file just discussed. On the remote client this file is located at /etc/openldap/ldap.conf, but on the server host it is located at /usr/local/ldap/etc/openldap/ldap.conf. Refer to the listing in Appendix B. The following parameters need to be set.

```
BASE dc=mytestlan,dc=com
URI ldap://ormoc.mytestlan.com ldaps://ormoc.mytestlan.com
```

BASE is the same as the suffix in the slapd.conf file. URI identifies the FQDN of the server host name in Uniform Resource Identifier notation. The key words “ldap” and “ldaps” respectively represent the ports 389 and 636. ldaps is not required now, but will be used for SSL connections later. The URI parameter replaces the HOST parameter used in previous versions of OpenLDAP. If HOST was set in the file, either comment or remove it. Peruse the ldap.conf manual page for other useful parameters.

A few more issues need to be addressed. The host for the LDAP server must be resolvable without the LDAP server, either by a DNS server or by adding the FQDN to the /etc/hosts file. Now, edit the passwd, shadow and group files to remove any users and groups placed in the LDAP database. It would be a good idea to make temporary backup copies of these files before removing the entries, but be sure to keep the temporary shadow file safe from non-privileged users. At this stage, the client accounts are active. However, on a remote client the users home directory may not exist. This will drop the user into the root directory, /, when logging onto the client host. To manually create a home directory for user ajones, follow the steps below using root privilege.

```
# mkdir /home/ajones
# chown ajones:ajones /home/ajones
# chmod 700 /home/ajones
# cp -R /etc/skel/.??* /home/ajones
# chown –R ajones:ajones /home/ajones/.??*
```

Now, log on as the user ajones. For testing on a remote client, the OpenLDAP client tools are required. If not already available, extract them from the package openldap-clients-2.0.27-8.i386.rpm.

7.3.8 Testing the basic configuration

When testing for clear text and cipher text transactions, a sniffer tool, such as ngrep, is required. As root, run ngrep on the LDAP server at port 389 for network traffic from the remote client.

```
# /usr/local/sbin/ngrep -d eth0 port 389
```

View the traffic generated from the server on localhost with the following command.

```
# /usr/local/sbin/ngrep -d lo port 389
```

As a non-privileged user, the following search command allows the user to bind as the manager, requiring authentication with the LDAP password.

```
$ ldapsearch -x -b 'dc=mytestlan,dc=com' \
-D ‘cn=manager,dc=mytestlan,dc=com’ -W \
-H ldap://ormoc.mytestlan.com '(objectclass=*)'
```

The clear text output from the network, in Appendix E, shows the LDAP password, mysecret, in clear text, along with the user password as a cryptographic hash.

The tool getent retrieves entries from the administrative database. For instance, user and group information can be display with the following commands. These commands first read the passwd, group and shadow files. Then they read the corresponding LDAP entries.

```
$ getent passwd
$ getent group
$ getent shadow
```

Retrieving the passwd and shadow entries both serve up the cryptographic hash for the intruder sniffing the network. In addition, displaying the shadow entries may provide the hash of each user in the database, as shown below. This should raise serious security concerns.

```
$ getent shadow

jsmith:$1$LxahCgmN$UfgQKSG8.RnxJftcm/htk/:12385::90:7:::0
ajones:$1$mGXBV2cP$GWJ9D3GVyRWlhgtTi4otG.:12385::90:7:::0
```

Next, test the ability to change the user password with the client tool ldappasswd.

```
$ ldappasswd -x -D ‘cn=manager,dc=mytestlan,dc=com’ –W -S \
–H ldap://ormoc.mytestlan.com ‘uid=ajones,ou=People,dc=mytestlan,dc=com’
```

In this command, -S, forces a prompt for a new password. The option, -A, which prompts for the old password, does not seem to work properly. This tool works similar to the passwd command executed by root, except ldappasswd requires the additional password for the manager. Without specifying the manager's distinguished name, this command fails. The network traffic also shows the passwords passed in the clear. See Appendix E for the output display.

As user ajones on a remote client, change the password with the standard passwd tool. This command should fail with a screen dump similar to that below.

```
$ passwd

Changing password for user ajones.
Enter login(LDAP) password:
New password:
Retype new password:
LDAP password information update failed: Unknown error

passwd: Permission denied
```

On the server this command succeeded, due to the existence of the /etc/ldap.secret file. A couple of other issues arose when testing this utility. First, to work properly the host parameter in /etc/ldap.conf must be set to the FQDN for the LDAP server. Second, with the current configuration in the slapd.conf file, the user may not have privilege to write to the database, resulting in a failure to change the password. To fix this problem apply ACL rules.

7.3.9 Access Control List (ACL)

ACL rules add security by restricting who can access specific parts of the database. The rules will be set in the database section of the slapd.conf file.

The following rules allow users to change their passwords.

```
access to attr=userPassword
        by self write
        by anonymous compare
        by * none
```

The "self write" clause allows a user to initiate a password change with the passwd command. Anonymous access to compare is required to complete the password change operation with the passwd command. The "by * none" clause which is the default forces the search through the ACL rules to stop whether or not a rule was satisfied. These rules also prevent the display of the encrypted password hash with the "getent shadow" command.

Other rules permit anonymous access control.

```
access to dn.children="ou=People,dc=mytestlan,dc=com"
        by anonymous read
        by * none
access to dn.children="ou=Group,dc=mytestlan,dc=com"
        by anonymous read
        by * none
```

These rules are required so the user accounts can find their user and group names. The only allowed access is for user account information under the “ou=People” and “ou=Group” nodes.

After changing the slapd.conf file, restart the LDAP service with “service ldap restart”. Testing with the ldapsearch command without the manager’s DN and password will only display the user and group information.

```
$ ldapsearch –x –H ldap://ormoc.mytestlan.com \
-b ‘dc=mytestlan,dc=com’ ‘(objectclass=*)’
```

With ACL rules, the order of placement is important. Checking through the rules stops after the first one is satisfied.

Now the passwd command will allow the users to change their passwords. The following portion of the transaction shows the users old password, oldpassword, and the cryptographic hash for the new password.

Network sniffer display – passwd (unencrypted)

```
##
T 192.168.20.11:33176 -> 192.168.20.10:389 [AP]
  0?...`:....(uid=ajones,ou=People,dc=mytestlan,dc=com..oldpassword
##
T 192.168.20.10:389 -> 192.168.20.11:33176 [AP]
  0....a.....
##
T 192.168.20.11:33176 -> 192.168.20.10:389 [AP]

0s...fn.(uid=ajones,ou=People,dc=mytestlan,dc=com0B0@...0;..userPassword1+
.
  ){crypt}$1$LiiIZYlB$0m5Ue8RZZ8S8XoWJ2TLe3/
##
```

Now, retest the “getent shadow” command. The display of the user’s encrypted hash is no longer visible.

```
$ getent shadow

jsmith:x:12385::90:7:::0
ajones:x:12385::90:7:::0
```

One other feature includes the ability for the server root to change a user's password. Root users on remote clients no longer have this privilege. If an intruder gets a root shell on a client host, changing a user's password is not an option. This limits the privilege to the server administrator or LDAP manager and is in conformance with the security policy, by supporting the principle of least privilege.

7.4 Options for Securing Access

7.4.1 Access Issues

The configuration so far uses simple authentication with no encryption making the transactions very insecure. Client authentication is implemented through PAM with the use of a password in clear text. This opens the possibility for a man-in-the-middle attack and sniffing passwords. The host name for the server is used instead of the IP address, possibly resulting in authentication credentials ending up at a rogue server instead of the intended server. Server authentication is certainly needed. The ACL rules provide some access control for the database, but encrypted access needs to be enforced.

7.4.2 Authentication Options

Several options are available through the Simple Authentication and Security Layer (SASL) included in the OpenLDAP package for providing authentication. Simple authentication is the easiest to implement, but information is passed as clear text providing no security. An external mechanism must be used to provide confidentiality and data integrity. Traditional ways for securing authentication services include Kerberos, GSSAPI and DIGEST-MD5. These methods can be difficult to set up and require the use of additional servers (e.g. Kerberos), which must also be secured. A simpler option to implement is SSL and TLS. This is an external mechanism for encrypting clear text messages when simple authentication is used. With TLS X.509-based public key technology, strong authentication for both the server and client can be achieved.

7.4.3 SSL/TLS

Secure Socket Layer (SSL) and Transport Layer Security (TLS) are accepted mechanisms for providing encrypted transactions over a network. SSL was originally defined by Netscape and is widely used for Web-based transactions. TLS version 1 is very similar to SSL version 3. Its use with LDAP is defined in RFC 2830[11]. Both SSL and TLS use certificates for server and client authentication, providing confidentiality and data integrity through encryption. The OpenLDAP server uses the two default ports,

[11] Hodges, "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830. http://www.isi.edu/in-notes/rfc2830.txt.

389 and 636, for receiving network messages. Port 389 is used for unencrypted, as well as TLS encrypted communications. Port 636 is used for SSL encrypted messages.

7.4.4 Certificates

TLS uses X.509 certificates as a basis for authentication with public key cryptography. Certificate files provide the public key, while key files contain the private key. The OpenSSL package provided by the Red Hat 9 distribution will be used to generate the required certificates. OpenLDAP requires the server to have valid certificates for secure operation, but client certificates are optional. The creation of server certificates involves several steps, using the openssl tool. First, create a Certificate Authority (CA) certificate and key pair. Then generate a server key and “certificate signing request” (CSR). The self-signed CA certificate will be used to sign the server certificate. When creating the CA certificate and CSR, information will be requested from the user that will become part of the certificate. Only two fields are critical. The “Common Name” must be the same as the LDAP server name. This is the FQDN used when clients contact the server (e.g. ormoc.mytestlan.com). The other field is the “Organizational Unit Name”. The value for this field in the CA certificate, must be different from that used in the “certificate signing request”.

Create a CA key file.

```
$ openssl genrsa -des3 -out serverca.key 2048
```

The parameter genrsa tells openssl to generate an RSA private key. The option, -des3, forces triple DES encryption of the CA key on disk, requiring a pass phrase entered by the user. Use of other encryption ciphers, such as DES and IDEA, is also possible with this version of openssl. Any use of this key to generate the CA and server certificates will require the pass phrase. Without a cipher, the key is stored on disk without encryption. Encrypting the key adds security, but also adds another pass phrase for management. The name of the file containing the private key is serverca.key. The size of the key is 2048 bits.

Generate the CA certificate with

```
$ openssl req -new -x509 -days 365 -key serverca.key -out serverca.cert
```

The parameter req with the option, -x509, creates a self-signed CA certificate. The option, -new, forces user prompts for information. The length of time the certificate will be valid is set at 365 days. This value should be set in accordance with the security policy. The following reason is offered for using this time frame. The CA certificate is for server authentication. As with passwords, a periodic change to this certificate adds to the server security. A one-year interval does not place an unreasonable workload on administrators to distribute the certificate to the client systems. The CA key and corresponding pass phrase is required to generate the CA certificate. There is no need to encrypt the certificate on disk since it should be world readable.

Generate the server key with a private key length of 1024 bits. Options are available to encrypt the key on disk, but according to the OpenLDAP user's guide the LDAP server does not support this feature. Other means are necessary to properly secure this key file.

```
$ openssl genrsa -out server.key 1024
```

Create the certificate signing request from the server key.

```
$ openssl req -new -key server.key -out server.csr
```

As with the CA certificate, the "Common Name" must be the FQDN for the server. The "Organizational Unit Name" must be different from that used for the CA certificate. The user may be prompted for additional attributes, such as a challenge password and optional company name. According to the OpenSSL manual page, these requested attributes are currently ignored, so they can be left blank.

Now, create the server certificate.

```
$ openssl x509 -req -in server.csr -out server.cert -CA serverca.cert \
-CAkey serverca.key -CAcreateserial -days 365
```

The parameter x509 has a multi-purpose functionality. In this case, it generates a certificate for the server. The option, -req, requires a certificate request, supplied with the option, -in. The CA certificate and CA key are required for signing the server certificate. The option, -CAcreateserial, creates a serial number file named server.srl. When the CA certificate signs the server certificate, the serial number in server.srl is used and then incremented for the next certificate. Consistent with the CA certificate, the time frame for validity is 365 days. The pass phrase for the CA key is again required.

Now that they all exist, find a place to store the certificates and keys. Since these are generated strictly for the ldap server, create the directories /usr/local/ldap/cert and /usr/local/ldap/cert/key with the following ownership and permission changes. Root privilege is required.

```
# chown ldap:ldap /usr/local/ldap/cert
# chown ldap:ldap /usr/local/ldap/cert/key
# chmod 755 /usr/local/ldap/cert
# chmod 700 /usr/local/ldap/cert/key
```

Key files should be owned by the ldap user, owner readable and stored in the /usr/local/ldap/cert/key directory. Placing keys in a separate directory from the certificates help with securing and managing the keys.

```
# chmod 400 serverca.key
```

```
# chmod 400 server.key
# chown ldap:ldap serverca.key
# chown ldap:ldap server.key
```

Certificates should be owned by the ldap user, and be world readable, For this implementation they will be stored in the /usr/local/ldap/cert directory. Store the CSR and SRL files with the same permissions as the certificates. Root can maintain ownership of these files.

```
# chmod 644 serverca.cert
# chmod 644 server.cert
# chmod 644 server.csr
# chmod 644 server.srl
# chown ldap:ldap serverca.cert
# chown ldap:ldap server.cert
# chown root:root server.csr
# chown root:root server.srl
```

Verify the server certificates with the command

```
$ openssl verify -CAfile serverca.cert server.cert
```

A list of error codes is available from the manual page on “verify”. Also, examine the certificate contents with the commands

```
$ openssl x509 -in serverca.cert -text –noout
$ openssl x509 -in server.cert -text –noout
```

These certificates are intended for the LDAP server and not for general use with a certificate authority. Generation and use of the certificates require additional administration tasks for certificate and key management on the server and client hosts. If certificates have a time interval for validity, new certificates need to be issued and implemented before expiration to avoid user down time. Certificates and keys used for authentication need to be backed up. The keys can be kept in a locked safe and off site in a secure facility. Any pass phrases used in creating the certificates, especially for the CA key, must also be kept safe along with the keys. Since the certificates are public, they can be backed up with other database information. These issues also need to be addressed in the security policies and procedures.

7.5 Secure Configuration with TLS

7.5.1 Server Configuration

The server configuration for TLS is in the slapd.conf file. The same TLS parameters are used for both SSL and TLS configuration.

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2
TLSCertificateFile /usr/local/ldap/cert/server.cert
TLSCercificateKeyFile /usr/local/ldap/cert/key/server.key
TLSCACertificateFile /usr/local/ldap/cert/serverca.cert
TLSVerifyClient allow
```

The first parameter in the listing above identifies which ciphers are valid. HIGH allows all ciphers with key lengths greater than 128 bits. MEDIUM allows all ciphers with key lengths equal to 128 bits. "+SSLv2" allows all ciphers specified in SSL protocol version 2, regardless of key strength. This might be used if clients use version 2 of SSL. All available ciphers are displayed with the command

```
$ openssl ciphers -v ALL
```

The next three TLS parameters specify where to find the certificate, key and CA files for the server. The last parameter specifies when to check for client certificates. There are four options for TLSVerifyClient. The default is "never", which does not check for a client certificate and always allows access. "Allow" forces a check for a client certificate, but allows access even if it is invalid or non-existent. "Try", allows access if the client certificate is valid and if no certificate is offered, but denies access if the offered certificate is not valid. "Demand" will grant access only if a valid client certificate is offered. This option is preferred for strict client authentication, using client certificates. Since this implementation does not use client certificates, the "allow" option will suffice. Restart the server to see if it still runs. At this point transactions are still sent in the clear. The client must be configured to take advantage of TLS.

7.5.2 Client Configuration

Client configuration is contained in the two files named ldap.conf. Configuration in the file /etc/ldap.conf requires a line with "ssl start_tls". Reactivate this line, commented out earlier. If the default port is used, TLS encrypted messages are sent to the server port 389. Unencrypted traffic also uses this port. Client configuration required by the LDAP package is set in the file /etc/openldap/ldap.conf on the remote client and /usr/local/etc/openldap/ldap.conf on the server, but no additional settings are required.

7.5.3 Testing TLS

It is now time to test the TLS configuration. Run the sniffer (ngrep) again on the server at port 389 (clear text and TLS) and port 636 (SSL). The same tests with ldapsearch, ldappasswd, getent and passwd can be run as before on both the server and a remote client. The output from these tests can be searched for passwords in either clear text or encrypted hash form. Neither should appear if TLS or SSL is applied.

First, try the ldappasswd command without TLS on the remote client.

```
$ ldappasswd -x -D 'cn=manager,dc=mytestlan,dc=com' -W -S \
-H ldap://ormoc.mytestlan.com 'uid=ajones,ou=People,dc=mytestlan,dc=com'
```

Looking at the network traffic, the password, mysecret, is visible along with the users password, newpassword.

Transactions in clear text occur for all the LDAP client tools. To use these tools with encrypted transactions, apply the options, -Z or –ZZ. The option, -Z, attempts encryption with TLS on port 389, but sends in the clear if encryption fails. Use of, –ZZ, forces encryption with TLS on port 389 and aborts the operation if encryption fails. Change the password using TLS.

```
$ ldappasswd -x -D 'cn=manager,dc=mytestlan,dc=com' -W -S \
-H ldap://ormoc.mytestlan.com 'uid=ajones,ou=People,dc=mytestlan,dc=com' -Z
```

If neither, -Z nor -ZZ, is used, SSL encrypted transactions on port 636 can be sent by replacing the URI with ldaps://ormoc.mytestlan.com. The ldaps port represents port 636. There is nothing inherently secure about port 636. The LDAP tools just use SSL with this port. Appendix E contains listings for both the clear text and cipher text transactions from a remote client.

A couple security concerns need to be addressed. When the LDAP manager uses the client tools on a remote host, make sure the correct options are used. Shell scripts can help enforce this. The other concern is on the LDAP server, where these options no longer work with the client tools. Clear text transactions are sent in the clear over the local interface lo. Pursuant to the security policy, this may be acceptable provided there is tight control over users and processes running on the server that monitor local traffic.

The tools getent and passwd should work as before on both the server and client hosts. Examining the TLS encrypted transaction, from the passwd tool, no password or cryptographic hash is visible anymore, as can be seen in Appendix E.

7.5.4 Client and Server Authentication

Server authentication enables the client host to verify that the server accessed is legitimate. This is easily accomplished by copying the CA certificate, serverca.cert, to the remote client into a suitable directory, such as /etc/openldap/cert. The client should never have the CA key. Set the owner and group for the certificate to root with world read permissions. Next, configure the NSS settings in the file /etc/ldap.conf to the following.

```
tls_checkpeer   yes
tls_cacertfile  /etc/openldap/cert/serverca.cert
```

This can also be configured on the server with the correct location for the serverca.cert file, if the policy allows client logins.

Account logins and accessing the LDAP server should behave as before, but now the client should be checking for a valid CA certificate. To test whether this is working, perform the following steps. Create another CA certificate exactly as before (e.g.

bogusca.cert) as well as the key (e.g. bogusca.key). Copy the file bogusca.cert into the same directory on the remote client host and with the same owner and permissions as the serverca.cert file. In the file /etc/ldap.conf set the tls_cacertfile parameter to /etc/openldap/cert/bogusca.cert. Now try executing the command “ls -l”. All of the user and group names should show up as their corresponding id numbers. Log in access and password changes from the client should be refused. Now set tls_cacertfile to the correct certificate, serverca.cert and let’s continue.

Client authentication requires client certificates and provides further authentication to the server that the request came from a valid client. The following discussion describes some parameters provided in OpenLDAP for client authentication, but implementation will not be shown. Client certificates use the distinguished name to authenticate with the LDAP server. This requires activating the SASL EXTERNAL mechanism by explicitly setting the TLSVerifyClient parameter to the desired option in the slapd.conf file. The options tls_cert and tls_key in the “.ldaprc” file must be set to valid user certificate and key files. The “.ldaprc” file must reside in the user’s home directory on each host the user has an account. The key file must be kept secret and owned by the user, with read access only to the user.

Client authentication, using certificates, can increase security by providing a second factor for authentication, with the password being the first factor. There is often a risk of someone acquiring a user’s password through social engineering, shoulder surfing or brute force guessing, but with client authentication the potential intruder will still not be allowed to log on from a host without the proper certificates. This does not prevent an insider from using the users console and account, but adds another hurdle that must be jumped. Client certificates provide the element of non-repudiation, but on a large network, they may increase the workload for the administrator to generate, distribute and manage the certificates for each user. The additional administration duties required by client certificates may not be warranted by the associated risk, but if this is shown, through auditing mechanisms, to be a problem then they can be implemented. This is all subject to compliance with the security policy.

7.5.5 ACL and Security Strength Factor

ACL rules already implemented do not specify any type of encryption requirements. A security strength factor (ssf) can be set to require a client to use a minimum level of encryption before access is granted. According to the OpenLDAP 2.1 Administrator’s Guide, the ssf value roughly correlates to the effective encryption key lengths. For example, an ssf of 56 would correspond to a DES level of encryption. The ACL rules previously configured in slapd.conf can be adjusted to include security strength factors. For password control, the first rule allows users to change their own password only if the request uses encryption with strength of 128 or better. The second permits authentication with a strength factor of 56.

```
access to attr=userPassword
        by ssf=128 self write
        by ssf=56 anonymous compare
        by * none

access to dn.children="ou=People,dc=mytestlan,dc=com"
        by ssf=56 anonymous read
        by * none
access to dn.children="ou=Group,dc=mytestlan,dc=com"
        by ssf=56 anonymous read
        by * none
```

The rules allowing access to read user and group information require at least a DES level encryption.

After restarting the LDAP server, test the security strength factors from a remote client. Recall the ldapsearch tool to search the database for user and group information

```
$ ldapsearch –x –H ldap://ormoc.mytestlan.com \
-b ‘dc=mytestlan,dc=com’ ‘(objectclass=*)’
```

Since this command passes clear text messages with no encryption, access to information in the database is no longer allowed. By using SSL or TLS, the information is accessible. The LDAP client tools with the manager DN and password are not affected by security strength factors. Unfortunately, the LDAP manager can still send clear text transactions, if not careful. The getent and passwd commands should also work as expected, since the appropriate encryption level is used.

To further test this feature, edit the /etc/ldap.conf file and comment out the lines with “ssl start_tls”, “tls_checkpeer”, “tls_cacertfile”. Access should now be refused, except for the privileged LDAP manager using the client tools. Be sure to re-enable the three TLS parameters that were just disabled for testing.

7.5.6 Host Authentication

Host authentication is used to enable login rights on specified hosts for selected users and to disallow rights on hosts not specified. This is a good measure to help protect against compromise of the whole network if a few systems are compromised.

Configuring the server requires a change to the user definitions in the LDAP database. The host attribute defined in the objectclass “account” in the file cosine.schema will be needed. This attribute specifies a host that the user is allowed to access. Multiple host attributes can be set with one host per attribute. This attribute is added to users in the database using the following ldapmodify tool.

```
$ ldapmodify -x -H ldap://ormoc.mytestlan.com \
-D ‘cn=manager,dc=mytestlan,dc=com’ -W -f addhost.ldif
```

The file addhost.ldif, listed in Appendix C, contains the lines necessary to add the host attribute.

The host restricting access also requires the following configuration addition in the /etc/ldap.conf file.

```
pam_check_host_attr        yes
```

This forces the LDAP server to check for login rights for the requesting host. The host attribute in the user definition must contain the FQDN of the system.

With the changes in the addhost.ldif file, user ajones is granted access to the LDAP server ormoc.mytestlan.com and the client host baybay.mytestlan.com. User jsmith is only granted access to the host baybay.mytestlan.com. When jsmith tries to log onto the server the connection should be refused. User ajones can still access either system.

To demonstrate, first set the loglevel parameter in slapd.conf to 256 and restart the server. This loglevel setting is more suitable for an operational server recording connections, operations and results. Now, let the user jsmith log onto the server from the client through ssh. The recorded segment in Appendix D shows the connection attempt in the ldap log. Information gleaned from this log includes the time the connection was initiated at 22:43:52, the source IP address of 192.168.20.11 to the destination port 389, the username of jsmith and the closed connection at 22:43:53. This was one of five unsuccessful connections made to the server for this single login attempt amounting to about 90 lines.

7.5.7 TCP Wrappers

TCP Wrappers are built into the LDAP server to filter clients by IP address. The tcpd daemon is not required. The hosts.allow and hosts.deny files are used to permit and restrict access to the service. Entries generally consist of the service name and IP addresses.

In the hosts.allow file access to the LDAP service, slapd, is allowed by the first line. The second line allows access through ssh. Access is allowed from clients on the 192.168.20 network, except for the client at 192.168.20.1.

hosts.allow file

```
slapd:192.168.20.0/255.255.255.0 EXCEPT  192.168.20.1
sshd:192.168.20.0/255.255.255.0 EXCEPT  192.168.20.1
```

For the hosts.deny file add the line that denies all that have not been allowed in hosts.allow.

hosts.deny file

```
ALL:ALL
```

With this setup, care must be taken to specify all the required services in the hosts.allow file or risk locking out some critical services, if not the whole system. If the server is administered remotely, this could result in an unexpected visit to the site. Failed access attempts should be logged in the /var/log/ldap file through the SYSLOG utility.

7.5.8 IP Tables

Access to the server host from the network is controlled by IP Tables and is configured in the /etc/sysconfig/iptables file. This service can block unauthorized traffic that finds its way through the network firewall. To make the LDAP server usable, SYN packets must be allowed into the server on ports 389 for TLS and 636 for SSL. These rules can be set using the Red Hat supplied utility, lokkit. If this utility was not run during the Red Hat installation, then run it now. Set the firewall to a high security level and follow the instructions under the section above labeled “Operating System”. By default, the firewall allows all IP addresses through to the LDAP service ports. The rules can be adjusted to only allow access from IP addresses on the network. Edit the iptables file and find the two rules allowing SYN packets to enter on ports 389 and 636. Use the source and destination options to set the rules to look similar to the following, but change the source network address accordingly.

```
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp -s 192.168.20.0/24 -d 0/0 --dport 389
--syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp -s 192.168.20.0/24 -d 0/0 --dport 636
--syn -j ACCEPT
```

These rules allow TCP SYN packets on port 389 (ldap) and 636 (ldaps), if the source is on the local network 192.168.20. These rules do not prevent an unauthorized client from physically plugging into the local network to get past the server firewall. The firewall can use these new settings by typing the command “service iptables restart” as root. As was stated several times, these changes should be done in accordance with the security policy and procedures.

To verify the correct settings with root privilege, the active firewall rules can be inspected with the following command. A listing of these rules is shown in Appendix D.

```
# iptables –nL
```

Incoming local network traffic will be allowed through to the services on TCP ports ldap, ldaps and ssh. Rules for receiving DNS responses allow incoming traffic on UDP port 53. All other TCP and UDP ports for incoming traffic are blocked.

7.5.9 File System Integrity

In the event of an intrusion or by accident, critical server files might be changed affecting the actions of the server. To watch for unauthorized changes a file system integrity checker, such as Tripwire, should be implemented. For this server, the following files have been identified for monitoring. Checks for the database and replication database files are optional depending on how often they change. Those strictly used for authentication may be good candidates for monitoring.

- The Berkeley database engine files under /usr/local/bdb.4.1.25
- Server database files in /usr/local/ldap/var/openldap-data
- Server replication database files in /usr/local/ldap/var/openldap-slurp
- LDAP server configuration file /usr/local/ldap/etc/openldap/slapd.conf
- LDAP client configuration file /usr/local/ldap/etc/openldap/ldap.conf
- Schema files in /usr/local/ldap/etc/openldap/schema
- LDAP client tools in /usr/local/ldap/bin
- LDAP server tools in /usr/local/ldap/sbin
- NSS LDAP configuration file /etc/ldap.conf
- Startup script /etc/rc.d/init.d/ldap
- Startup options file /etc/sysconfig/ldap
- LDAP secret file /etc/ldap.secret
- Libraries in /usr/local/ldap/lib
- Servers slapd and slurpd in /usr/local/ldap/libexec
- File under the /usr/local/ldap/share subdirectory
- Certificates, keys and other related files
- /etc/hosts.allow and /etc/hosts.deny
- passwd, group and shadow files

These are the obvious LDAP service related files that need to be monitored. Files for PAM, NSS, IP Tables, TCP Wrappers and other system related functions, should also be monitored to help ensure the security of the LDAP server. Monitoring the passwd, group and shadow files listed above may catch the installation of unauthorized accounts and authorized users setting up local accounts to bypass ldap authentication. A file integrity checker should run daily to verify the integrity of the critical files.

7.5.10 Server Availability and Recovery

Other important security concerns include server availability and the ability to recover from a disaster. Several measures can be implemented to help ensure this, pursuant with the security policy. The replication server, slurpd, provided with the OpenLDAP package, can be implemented to provide authentication services when the main server is unavailable. The main server may become unavailable due to an attack, maintenance or other events. Backups with slapcat require the server be shut down or in read-only mode to ensure a consistent copy of the database. Data restoration with slapadd requires the server be shut down to ensure an uncorrupted database. This procedure results in users unable to access their accounts. However, authentication service can still be made available with replication servers.

7.5.11 Additional Measures

For critical files and directories, check owner and permission settings. The ldap user should own the following files and directories, with the specified permission settings.

- Directory /usr/local/ldap/var/openldap-data (0700)
- Directory /usr/local/ldap/var/openldap-slurp (0700)
- Database files in /usr/local/ldap/var/openldap-data (0600)
- Replication files in /usr/local/ldap/var/openldap-slurp (0600)
- Certificate file directory (0755)
- Certificate files serverca.cert and server.cert (0644)
- Certificate key directory (0700)
- Certificate key files serverca.key and server.key (0400)
- Configuration file slapd.conf (0600)

The ldap.secret file must be owned by root, with read and write permissions (0600) by root. The files named ldap.conf for the NSS and OpenLDAP configurations, plus the schema files must be world readable and owned by root. All other LDAP related files and directories should be owned by root with no world write access.

Another measure is to run the server in a "jailed" environment. OpenLDAP has a built-in capability to chroot itself by using the option, -r, when launching the slapd daemon. This will force the server to run in a captive environment, restricting access to specific directories and files. Setting up a chrooted environment can be both a complicated and tedious process. Since the implementation described here runs the slapd daemon as the non-privileged ldap user and uses a non-interactive invalid shell, the chroot feature may not be needed. The decision to chroot can be made after monitoring the use and abuse of the operational system.

For this server, there is no reason to run extra services like X-Windows. If using a GUI tool to update the LDAP database, run it from a client. Starting the system and running the server in run level 3 will help remove unnecessary services and open ports. Two changes to the /etc/inittab file are listed below. The first forces the system to boot into run level 3. The second forces a password prompt when booting into single user mode. Red Hat does not require this password by default.

```
# Boot into run level 3
id:3:initdefault:
# id:5:initdefault:

# Force password on boot to single user mode
su:S:wait:/sbin/sulogin
```

To help secure physical access to the server console, locate the server in a controlled environment with the appropriate access controls. Another option is to require a password at the Grub or Lilo prompt. For example with Grub, use the grub-md5-crypt

program to create an MD5 password at the command line. Cut and paste the returned password into the grub.conf file, with a line prefixed by "password --md5" as follows.

```
default=8
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
password --md5 $1$mzBN0$HcVAQ7v2duGeW1LWGynwjz/
```

Also, set a password at the BIOS prompt and disable the feature to boot from floppy and cdrom drives.

8 Ongoing Maintenance

8.1 General Issues

The first issue concerning ongoing maintenance is that all maintenance, including the server implementation, be in accordance with the security policy. The LDAP manager must also maintain user accounts with the client and server tools. A GUI tool such as GQ[12] could be very helpful, especially for large networks. Certificates must remain valid and require regeneration before they expire. Create and distribute new certificates and keys before expiration. This implementation requires generation and distribution at least every 365 days. This may coincide with a particular scheduled backup time or some other annual maintenance event. Similarly, change the LDAP password annually at the same time as the certificates.

Software vulnerabilities are a major concern. Stay aware of recent vulnerabilities as they relate to the server and clients. Upgrade the software packages when necessary. Several Web sites provide information on vulnerabilities. The site "Common Vulnerabilities and Exposures"[13] hosts the official CVE list plus the candidate vulnerability descriptions under consideration. The site at securityfocus.com[14] provides an easy to use search facility for vulnerabilities. A third site is the "CERT Coordination Center"[15].

A few additional issues are worth mentioning. If replication servers are used, their hosts should be treated the same as the primary server. Verify physical protections for the host computer and server software on a predefined schedule. Save the packages used to build and implement the server on external media, such as cdrom. This ensures a clean source for a full rebuild and installation, if required.

8.2 Upgrades

When package upgrades are available or configuration modifications suggested, consider their effect before implementation. Several factors affect the decision to

[12] "GQ LDAP Client", http://biot.com/gq/.
[13] "Common Vulnerabilities and Exposures". http://www.cve.mitre.org/.
[14] "Vulnerabilities", SecurityFocus, http://www.securityfocus.com/bid.
[15] CERT Coordination Center, "CERT/CC Advisories", http://www.cert.org/advisories/.

upgrade, including the nature of the upgrade, server environment, resources available and the security policy. Upgrading a package without understanding the changes may result in a more vulnerable system. Installing and testing the upgrades and configuration changes, on a representative test system prior to implementation is a good idea.

Several packages may need to be upgraded on the server. The Berkeley database and OpenLDAP software require rebuilding and testing the applications as described in this guide. Consult the package documentation for changes in the requirements and build process. If the decision is made to switch to the Red Hat rpm packages for these applications, reconfiguration will be required to point to the new locations for slapd and other files. Other packages can be upgraded with the Red Hat rpm files. OpenSSL upgrades are critical, since they include the libraries for encryption and certificate generation. The glibc packages are used by many applications and should be kept up to date. The nss_ldap package provides the NSS and PAM libraries for the LDAP server. These upgrades will likely require other packages to be upgraded, as well. When downloading upgrade software always check for package integrity and authenticity when possible. These and additional operating system packages should be upgraded when deemed necessary, based on information about vulnerabilities assessments and system requirements.

8.3 Backup and Restore

On a predefined schedule, make backups of the server software, configuration, database files, certificates and certificate keys. The backup interval depends on how often the data changes. This will vary depending on the organization size, employee turnover rate and other factors. Set up a rotation schedule for saving backups in case it is necessary to go back several cycles. The LDAP server tools slapcat and slapadd will, respectively, backup and restore the database. These tools require the server be shut down or in read-only mode, for slapcat, to avoid data corruption. Service availability can still be maintained during backup through replication servers. Standard operating system tools such as tar, dd and gzip are also helpful for system backup, restore and data compression. In addition to the regularly scheduled backups, incremental backups can be saved when changes are made to the database and files. Securely label and store the backup media in a safe place on and off the company site. These schedules should be determined in accordance with the security policy.

8.4 System Integrity

System integrity checks with Tripwire can be executed daily as a cron job, when the system is not heavily used. For a list of candidate server files, see the above section on “File System Integrity”. Recheck and verify the system security after any upgrade and configuration changes. On the remote clients, the system integrity checker should monitor changes to system files, especially those for the LDAP client configuration, certificates and keys. Other client files monitored should include the ldap libraries, SSL libraries and system files used by PAM and NSS, since these are core for secure user authentication. Monitor the traditional UNIX authentication files, including passwd, group and shadow, to catch the installation of unauthorized accounts and authorized users setting up local accounts to bypass ldap authentication. Additional monitoring of the

environment variables should be done to ensure system integrity. See the section on "Client Configuration" under "Basic Configuration" for more details on a potential problem with the system-auth file.

8.5 System Audits

Logs from integrity checks, LDAP transactions and other events can be monitored manually, but using tools like LogSentry and Swatch will be helpful. Since The LDAP server log records a lot of information, a good monitoring tool, with training to use it, will be very beneficial. An automated monitoring tool can run daily, with a more detailed manual check performed weekly. If an event occurs or a recent vulnerability is reported, more frequent monitoring may be done.

Periodically and when changes are made to the server, the network should be scanned for open ports and vulnerabilities. Nmap is a good network-scanning tool and a current version of Nessus can scan for potential network vulnerabilities. Perform heavy scanning of the server at a time when there is little impact on the users. This may coincide with a scheduled maintenance interval to minimize server unavailability issues.

9 Network Security Check

9.1 Server Security Verification

The server was tested and verified at various stages in the implementation. The following discussion will cover the key security concerns with procedures for final verification.

9.1.1 Access Control

ACL rules and the LDAP password provide the main mechanisms for access control. ACL rules also enforce a requirement for a minimum level of security with security strength factors.

The getent tool tests access to the authentication databases. From a non-privilege account, this command applied to the passwd and group entries should display information from the corresponding files followed by the LDAP database entries. Their operation should give the normal response with no surprises. The critical test, with the shadow entries, will test the ACL rules.

```
$ getent passwd
$ getent group
$ getent shadow
```

This command should return values only from the LDAP server without the user's cryptographic hash.

```
$ getent shadow

jsmith:x:12385::90:7:::0
ajones:x:12385::90:7:::0
```

The next two tools demonstrate the access available to the LDAP manager. The ldapsearch tool, with manager privilege, displays all entries, under the distinguished name dc=mytestlan,dc=com, from the LDAP database.

```
$ ldapsearch -x -b 'dc=mytestlan,dc=com' \
-D ‘cn=manager,dc=mytestlan,dc=com’ -W '(objectclass=*)'
```

Note that the display contains the entry for the userPassword field. The ldappasswd tool, with manager privilege, will change the password for any user in the database. The following command changes the password for the user ajones.

```
$ ldappasswd -x -D ‘cn=manager,dc=mytestlan,dc=com’ –W -S \
 ‘uid=ajones,ou=People,dc=mytestlan,dc=com’
```

For both client tools, remote client generated transactions are sent to port 389 on the server by default. This is the same as using the ldap URI. Clear text transactions over the network will result, without the options for TLS encryption.

Perform three tests with normal use of the standard UNIX command, passwd.

- First, to test the ACL rules, verify that the non-privileged user can only change their password from their account.
- Second, a user with root privilege on the LDAP server can change any users password, provided the user authenticates through LDAP.
- Third, a user with root privilege on a remote host should not be able to change any user’s password, except their own.

As a non-privileged user from the remote client, try searching the database without encryption. The ACL security strength factors should allow no access.

```
$ ldapsearch -x -b 'dc=mytestlan,dc=com' '(objectclass=*)'

version: 2

#
# filter: (objectclass=*)
# requesting: ALL
#

# search result
search: 2
result: 0 Success

# numResponses: 1
```

Now try the above with TLS encryption.

```
$ ldapsearch -x -b 'dc=mytestlan,dc=com' '(objectclass=*)' -Z
```

The results in Appendix D, unlike the display by the LDAP manager, only contain user and group information. Also, note that the userPassword field is not shown.

As server root, verify the correct configuration for the firewall and TCP Wrappers. Check the firewall rules, filtering IP addresses to allow traffic on expected ports and disallow it on other ports. The following command displays the active rules, which are also shown in Appendix D.

```
# iptables –nL
```

Check the hosts.allow and hosts.deny files for the appropriate configuration. Examples are listed below.

hosts.allow file

```
slapd:192.168.20.0/255.255.255.0 EXCEPT  192.168.20.1
sshd:192.168.20.0/255.255.255.0 EXCEPT  192.168.20.1
```

hosts.deny file

```
ALL:ALL
```

Actual testing of the firewall and TCP Wrapper rules will be done in the “Port Scan” section. For access verification to specific hosts, see host authentication in the next section.

9.1.2 Server, Client and Host Authentication

Server authentication requires the server CA certificate to be placed on the client host, with the appropriate setting in the /etc/ldap.conf file. To verify server authentication, either remove the CA certificate or replace it with another certificate. See the results of the su command below. Remember to replace the correct CA certificate when done.

```
$ su – ajones

su: user ajones does not exist
```

Client authentication as implemented consists of user passwords. Logging onto an account with invalid and valid passwords can test this feature.

Host authentication uses the “host“ attribute in the user’s LDAP definition to match an authorizing host with the user. Try to log onto the server, a restricted host, as a user without privilege to do so.

```
$ ssh ormoc.mytestlan.com –l jsmith

jsmith@ormoc.mytestlan.com's password:
Read from remote host ormoc.mytestlan.com: Connection reset by peer
Connection to ormoc.mytestlan.com closed.
```

After entering the password for user jsmith, the connection abruptly closed. Now, log in as ajones to verify that that the login still works from an authorized user.

9.1.3 Data Integrity and Confidentiality

For these tests, a sniffer, such as ngrep, is required. Sniff the network traffic on the server, pursuant to the security policy, to verify encrypted transactions. Apply the same commands used earlier in this section and inspect the transactions for clear text and encrypted traffic.

Use the client tools ldapsearch and ldappasswd with the options, -Z or –ZZ, for TLS encrypted traffic on port 389. For testing the SSL transactions on port 636, apply the URI ldaps://ormoc.mytestlan.com to the client tools, without the TLS options. Appendix E shows excerpts from sniffed transactions generated with these client tools. Standard logins plus the utilities getent, passwd, ssh and su should produce cipher text. The Pluggable Authentication Module provides encryption for these utilities, using the NSS supplied ldap libraries.

9.1.4 Logging and Auditing

Monitor the server log file at /var/log/ldap for attempted connections, operations performed and response from the LDAP service. The following command, executed as server root, displays messages to the screen. See Appendix D for a sample of the log file.

```
# tail -f /var/log/ldap
```

Another file, /var/log/messages, also logs connection information for the LDAP server. Monitor these logs manually or with tools like LogSentry and Swatch.

9.1.5 Software Vulnerabilities

Execute these tests on the server to verify configuration settings. With server root privilege, display the server user name and location with the lsof command.

```
# lsof -c slapd

COMMAND PID USER FD TYPE DEVICE   SIZE     NODE NAME
slapd   688  ldap cwd  DIR    3,24      1024        2 /
slapd   688  ldap rtd  DIR    3,24      1024        2 /
slapd   688  ldap txt  REG    3,27 1217340   608012 /usr/local/ldap/libexec/slapd
.
.
.
```

With root privilege, run the following command to display the password hash for the ldap user. The result should show "!!" for the password.

```
# getent shadow
```

Next, try logging onto the ldap account, as a non-privileged user.

```
$ su – ldap
```

Any password used should produce the following results.

```
su:  incorrect password
```

9.1.6 Other Concerns

Securing the physical console includes many items beyond the scope of this paper, but perform some of the system access methods to verify access controls. Boot into single user mode to verify the root password is required. For grub or lilo implementations, verify that a password is required to modify the boot options. If implemented, check for password-required access to the system BIOS. Finally, try to boot from a floppy and cdrom disk.

9.2 Port Scan

As part of the above security checks and to test the firewall and TCP Wrapper configuration, a scan for open ports from a client system on the network is required. Nmap is a network based port scanner that can determine if a host is alive with the ping utility, find open ports and attempt to determine the operating system of the host. Nmap

will be used to identify ports that might be open to attack, but it cannot determine whether they are actually vulnerable. The server at the IP address 192.168.20.10 will be scanned for open TCP, UDP and RPC ports. For this test, the latest rpm version 3.48 was used, from the Nmap Security Scanner download directory[16]. Both the Nmap rpm binary package and the Nmap "frontend" rpm package were installed. The "frontend" package includes a graphical user interface (GUI) application named nmapfe. The scans were run from the GUI, with root privilege, so the SYN scan could be utilized. Scanning the TCP and RPC ports on the server produced the following results.

```
nmap -sS -sR -p- -P0 192.168.20.10

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-01-17 19:03 PST
Interesting ports on ormoc.mytestlan.com (192.168.20.10):
(The 65532 ports scanned but not shown below are in state: filtered)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh
389/tcp open  ldap
636/tcp open  ldapssl

Nmap run completed -- 1 IP address (1 host up) scanned in 2661.395 seconds
```

The nmap command, on the first line, uses a SYN ("half-open") scan, to identify open TCP ports. All 65535 ports were scanned with only three open ports found, for ssh, ldap and ldapssl, as expected. The other 65532 ports were found filtered. No RPC ports were found to be open. The next scan is for UDP ports. Instead of scanning all ports, only the most important ports, as determined by the nmap services file, were scanned. UDP scans take a considerable amount of time and using the most important ports is an acceptable compromise.

```
nmap -sU -sR -F -P0 192.168.20.10

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-01-17 19:51 PST
All 1008 scanned ports on ormoc.mytestlan.com (192.168.20.10) are: closed

Nmap run completed -- 1 IP address (1 host up) scanned in 1003.439 seconds
```

All UDP and RPC ports scanned were found closed as expected.

These results are consistent with the firewall and TCP Wrapper configuration. Taking port scanning a step further, the source IP address can be spoofed with an address outside the network.

[16] NMAP Security Scanner download directory, http://download.insecure.org/nmap/dist/?M=D.

```
nmap -sS -O -p 389,636,22 -P0 -e eth0 -S 192.168.10.11 192.168.20.10

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-01-17 22:41 PST
Warning:  OS detection will be MUCH less reliable because we did not find at
least 1 open and 1 closed TCP port
Interesting ports on ormoc.mytestlan.com (192.168.20.10):
PORT    STATE    SERVICE
22/tcp  filtered ssh
389/tcp filtered ldap
636/tcp filtered ldapssl
Too many fingerprints match this host to give specific OS details

Nmap run completed -- 1 IP address (1 host up) scanned in 256.409 seconds
```

The allowable addresses set in the firewall and TCP Wrapper rules are in the 192.168.20 network. By spoofing the source address with 192.168.10.11, the previously open ports now appear filtered. In addition, the scanner was unable to identify the OS.

10 Conclusion

Implementing the LDAP server requires many steps, resulting in a customized secure authentication service. Building the server from scratch ensures availability of the required built-in security options. Security, with proper server and client configuration, requires the use of ACL rules, security strength factors, SSL/TLS and certificates. Data integrity and confidentiality, using SSL and TLS, integrates well with PAM. Secure implementation requires a good understanding of each of these elements. As emphasized many times in this paper, the LDAP password is a weak link, requiring proper monitoring and controls to keep it secure.

Client authentication has potential to provide strong non-repudiation capability by issuing each user unique client certificates. This increased security also comes with increased administration tasks for certificate and key management and distribution. Server authentication provides good assurance that the server is legitimate. Host authentication allows easy administrative control over the network host access from a single database and server. This also adds to administration tasks to maintain an up to date database. These authentication mechanisms require proper client host configuration. If an attacker gains sufficient access to the client, compromise of the configuration, as well as certificates and keys, may disrupt normal operation.

A reasonable sized network requires a good GUI administration tool, as well as tools for auditing the logs. Logging information at the default operational level (loglevel 256) is quite verbose. Recording operations, in addition to connection attempts and results, might help in a forensics analysis.

Server performance affects availability and scalability. On this 10 Mbps Ethernet test network with a small database, authentication seemed a little slow at times. However,

no statistics were actually collected or optimization implemented. For a production server, a performance study may be required.

A very important part of the server security that received little attention in this paper is the replication servers provided in the OpenLDAP package. Replication servers and data recovery measures are certainly required for service availability and disaster recovery.

The "OpenLDAP Administrator's Guide" from the authors of the package seemed to be the best source of information on the LDAP server. Careful reading of this guide and understanding what it discusses is an important step in getting the server to work. Other requirements are patience and persistence. The biggest hurdle encountered was the lack of knowledge on PAM, certificates and understanding what the LDAP server was doing. The limited testing and implementation described in this paper suggests OpenLDAP, with proper configuration, provides secure centralized authentication.

11 Appendix A – Server Configuration Files

11.1 LDAP Configuration

File: /usr/local/ldap/openldap/slapd.conf

```
# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.23.2.8 2003/05/24 23:19:14
# kurt Exp $

#######################################################
# global configuration directives
#######################################################
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /usr/local/ldap/etc/openldap/schema/core.schema
include         /usr/local/ldap/etc/openldap/schema/cosine.schema
include         /usr/local/ldap/etc/openldap/schema/nis.schema
include         /usr/local/ldap/etc/openldap/schema/misc.schema
include         /usr/local/ldap/etc/openldap/schema/inetorgperson.schema

# Define global ACL rules to disable default read access.
#schemacheck     off

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

pidfile         /usr/local/ldap/var/slapd.pid
argsfile        /usr/local/ldap/var/slapd.args
```

```
# log all information for debugging
#loglevel -1
# log connections, operations and results for operational mode
loglevel 256

# Sample security restrictions
#       Require integrity protection (prevent hijacking)
#       Require 112-bit (3DES or better) encryption for updates
#       Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64

# Sample access control policy:
#       Root DSE: allow anyone to read it
#       Subschema (sub)entry DSE: allow anyone to read it
#       Other DSEs:
#               Allow self write access
#               Allow authenticated users read access
#               Allow anonymous users to authenticate
#       Directives needed to implement policy:
# access to dn.base="" by * read
# access to dn.base="cn=Subschema" by * read

# if no access controls are present, the default policy is:
#       Allow read by all
#
# rootdn can always write!

#######################################################
# bdb database definitions
#######################################################

# Berkeley database
database        bdb
# DN suffix of the queries passed to the database
suffix          "dc=mytestlan,dc=com"

# Distinguished name that is not subject to access control
# or administrative limit restrictions for operations on this database.
# An empty rootdn (default) allows no root access,
# so comment out in production mode
rootdn          "cn=manager,dc=mytestlan,dc=com"

# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
```

```
rootpw {SSHA}CPfU1xQ0opiHWTP9TDtrZZu0YvSqDhle

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.

# Directory where database lives
directory       /usr/local/ldap/var/openldap-data
# Mode 600 recommended.
# Mode for newly created database index files
mode 0600

# Indices to maintain
index  objectClass  eq
index cn,uid  pres,eq,sub
index uidNumber,gidNumber,memberUid  eq

# Database access control definitions
# For the basic configuration the following ACL rules are valid,
# but the ssf factors can be removed.

# Note: with ssf values, if TLS or other appropriate cryptography is not
#       used by the client, access will not be granted.
#
# (1)Self write is needed to allow a user to initiate a password change
# with the passwd command. Anonymous access to compare is needed to
# complete the password change operation with the passwd command.
# Also required to prevent the password hash from being displayed with
# “getent shadow” command.
# (2) "ssf=56" forces the client to use a minimum security strength of 56
# (3) "ssf=128" forces the client to use a minimum security strength of 128
access to attr=userPassword
        by ssf=128 self write
        by ssf=56 anonymous compare
        by * none

# Allows read and search access for anonymous
# to all entries for user and group information.
# Required so user accounts can identify their user and group names.
# Minimum security strength of 56,
access to dn.children="ou=People,dc=mytestlan,dc=com"
        by ssf=56 anonymous read
        by * none
access to dn.children="ou=Group,dc=mytestlan,dc=com"
        by ssf=56 anonymous read
        by * none
```

```
#
# The basic configuration ends here
#
#############################################################
# TLS options (Not included in the basic configuration  without TLS)
#############################################################
# HIGH - all ciphers with key lengths > 128 bits
# MEDIUM - all ciphers with key lengths = 128 bits
# +SSLv2 - all ciphers specified in SSL protocol version 2, regardless of key strength
# use openssl ciphers -v ALL to see all available ciphers
TLSCipherSuite HIGH:MEDIUM:+SSLv2
# Server certificate with public key
TLSCertificateFile /usr/local/ldap/cert/server.cert
# Server private key
TLSCertificateKeyFile /usr/local/ldap/cert/key/server.key
TLSCACertificateFile /usr/local/ldap/cert/serverca.cert

# Client authentication
# No client certificate is checked
#TLSVerifyClient never

# Check for client certificate
# Allow access if valid certificate, fail if invalid or no certificate
#TLSVerifyClient demand
# Allow access if valid or no certificate, fail if certificate is invalid
#TLSVerifyClient try
# Allow access if certificate is valid, invalid or non existent
TLSVerifyClient allow
```

12 Appendix B – Client Configuration Files

12.1 PAM Configuration

File: /etc/pam.d/system-auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     [default=bad success=ok user_unknown=ignore service_err=ignore
system_err=ignore] /lib/security/$ISA/pam_ldap.so
```

```
password    required      /lib/security/$ISA/pam_cracklib.so retry=3 type=
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok md5 shadow
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     optional      /lib/security/$ISA/pam_ldap.so
```

12.2 LDAP Configuration

Remote Client File: /etc/openldap/ldap.conf
Server File: /usr/local/ldap/etc/openldap/ldap.conf

```
# $OpenLDAP: pkg/ldap/libraries/libldap/ldap.conf,v 1.9 2000/09/04 19:57:01 kurt Exp $
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.
BASE dc=mytestlan,dc=com
URI ldap://ormoc.mytestlan.com ldaps://ormoc.mytestlan.com
```

12.3 NSS Configuration

File: /etc/ldap.conf

```
# @(#)$Id: ldap.conf,v 1.24 2001/09/20 14:12:26 lukeh Exp $
#
# This is the configuration file for the LDAP nameservice
# switch library and the LDAP PAM module.
#
# PADL Software
# http://www.padl.com
#
# Your LDAP server. Must be resolvable without using LDAP.
host ormoc.mytestlan.com

# The distinguished name for the search base
base dc=mytestlan,dc=com

# Another way to specify your LDAP server is to provide an
# uri with the server name. This allows to use
# Unix Domain Sockets to connect to a local LDAP Server
#uri ldapi://%2fvar%2frun%2fldapi_sock/
# Note: %2f encodes the '/' used as directory separator

# The LDAP version to use (defaults to 3
```

```
# if supported by client library)
#ldap_version 3

# CAUTION: The configuration for rootbinddn and /etc/ldap.secret only
# needs to be set on the server and must not be used on the remote client.
# The distinguished name to bind to the server with
# if the effective user ID is root. Password is
# stored in /etc/ldap.secret (mode 600)
rootbinddn cn=manager,dc=mytestlan,dc=com

# The port.
# Optional: default is 389.
# Search timelimit
#timelimit 30
# Bind timelimit
#bind_timelimit 30
# Idle timelimit; client will close connections
# (nss_ldap only) if the server has not been contacted
# for the number of seconds specified below.
#idle_timelimit 3600

# RFC2307bis naming contexts
# Syntax:
# nss_base_XXX              base?scope?filter
# where scope is {base,one,sub}
# and filter is a filter to be &'d with the
# default filter.
scope one
nss_base_passwd ou=People,dc=mytestlan,dc=com?one
nss_base_shadow ou=People,dc=mytestlan,dc=com?one
nss_base_group ou=Group,dc=mytestlan,dc=com?one

# attribute/objectclass mapping
pam_filter objectclass=posixAccount
pam_login_attribute uid
pam_member_attribute gidNumber
pam_password md5
##### End of the LDAP configuration without TLS

##### The following parameters are needed to implement TLS
# Check the 'host' attribute for access control
# Default is no; if set to yes, and user has no
# value for the host attribute, and pam_ldap is
# configured for account management (authorization)
# then the user will not be allowed to login.
pam_check_host_attr yes
```

```
# OpenLDAP SSL mechanism
# start_tls mechanism uses the normal LDAP port, LDAPS typically 636
#ssl on
ssl start_tls

# OpenLDAP SSL options
# Require and verify server certificate (yes/no)
# Default is "no"
# CA certificates for server certificate verification
tls_checkpeer yes
# Remote client
tls_cacertfile /etc/openldap/cert/serverca.cert
# Server
#tls_cacertfile /usr/local/ldap/cert/serverca.cert
```

13 Appendix C – LDAP Data Interchange Format files

13.1 Directory Nodes Definition

File: mytestlan.ldif – created manually

```
dn: dc=mytestlan,dc=com
objectClass: dcObject
objectClass: organization
objectClass: top
o: Test Systems, Inc.
dc: mytestlan
description: A World Class Company

dn: cn=manager,dc=mytestlan,dc=com
objectClass: organizationalRole
objectClass: top
cn: manager
description: Directory Manager

dn: ou=Group,dc=mytestlan,dc=com
objectClass: organizationalUnit
objectClass: top
ou: Group
description: User Groups

dn: ou=People,dc=mytestlan,dc=com
objectClass: organizationalUnit
objectClass: top
ou: People
description: Users
```

13.2 User Definition File

File: passwd.ldif - generated from migrate_passwd.pl

```
dn: uid=ajones,ou=People,dc=mytestlan,dc=com
uid: ajones
cn: A. Jones
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}$1$mGXBV2cP$GWJ9D3GVyRWlhgtTi4otG.
shadowLastChange: 12385
shadowMax: 90
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 600
gidNumber: 600
homeDirectory: /home/ajones
gecos: A. Jones

dn: uid=jsmith,ou=People,dc=mytestlan,dc=com
uid: jsmith
cn: J. Smith
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}$1$LxahCgmN$UfgQKSG8.RnxJftcm/htk/
shadowLastChange: 12385
shadowMax: 90
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 601
gidNumber: 601
homeDirectory: /home/jsmith
gecos: J. Smith
```

13.3 Group Definition File

File: group.ldif - generated from migrate_group.pl

```
dn: cn=ajones,ou=Group,dc=mytestlan,dc=com
objectClass: posixGroup
objectClass: top
cn: ajones
userPassword: {crypt}x
gidNumber: 600
```

```
dn: cn=jsmith,ou=Group,dc=mytestlan,dc=com
objectClass: posixGroup
objectClass: top
cn: jsmith
userPassword: {crypt}x
gidNumber: 601
```

13.4 Host Modification Definition File

File: addhost.ldif – created manually

```
dn: uid=ajones,ou=People,dc=mytestlan,dc=com
changetype: modify
add: host
host: ormoc.mytestlan.com
host: baybay.mytestlan.com

dn: uid=jsmith,ou=People,dc=mytestlan,dc=com
changetype: modify
add: host
host: baybay.mytestlan.com
```

14 Appendix D – Sample Output

14.1 Successful Client Search Results - ldapsearch

```
$ ldapsearch -x -b 'dc=mytestlan,dc=com' '(objectclass=*)'  -Z

version: 2

#
# filter: (objectclass=*)
# requesting: ALL
#

# ajones, People, mytestlan, com
dn: uid=ajones,ou=People,dc=mytestlan,dc=com
uid: ajones
cn: A. Jones
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 90
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 600
```

```
gidNumber: 600
homeDirectory: /home/ajones
gecos: A. Jones
host: ormoc.mytestlan.com
host: baybay.mytestlan.com
shadowLastChange: 12417

# jsmith, People, mytestlan, com
dn: uid=jsmith,ou=People,dc=mytestlan,dc=com
uid: jsmith
cn: J. Smith
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowLastChange: 12385
shadowMax: 90
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 601
gidNumber: 601
homeDirectory: /home/jsmith
gecos: J. Smith
host: baybay.mytestlan.com

# ajones, Group, mytestlan, com
dn: cn=ajones,ou=Group,dc=mytestlan,dc=com
objectClass: posixGroup
objectClass: top
cn: ajones
gidNumber: 600

# jsmith, Group, mytestlan, com
dn: cn=jsmith,ou=Group,dc=mytestlan,dc=com
objectClass: posixGroup
objectClass: top
cn: jsmith
gidNumber: 601

# search result
search: 3
result: 0 Success

# numResponses: 5
# numEntries: 4
```

14.2 IP Table Rules Display

iptables –nL

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Lokkit-0-50-INPUT  all  --  0.0.0.0/0          0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Lokkit-0-50-INPUT  all  --  0.0.0.0/0          0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Lokkit-0-50-INPUT (2 references)
target        prot opt source               destination
ACCEPT     tcp  --  192.168.20.0/24   0.0.0.0/0       tcp dpt:389 flags:0x16/0x02
ACCEPT     tcp  --  192.168.20.0/24   0.0.0.0/0       tcp dpt:636 flags:0x16/0x02
ACCEPT     tcp  --  192.168.20.0/24   0.0.0.0/0       tcp dpt:22 flags:0x16/0x02
ACCEPT     udp  --  192.168.192.12    0.0.0.0/0       udp spt:53
ACCEPT     udp  --  192.168.192.4     0.0.0.0/0       udp spt:53
REJECT     tcp  --  0.0.0.0/0         0.0.0.0/0       tcp flags:0x16/0x02 reject-with icmp-
port-unreachable
REJECT     udp  --  0.0.0.0/0         0.0.0.0/0         udp reject-with icmp-port-
unreachable
```

14.3 LDAP Log File - failed connection

```
LDAP log file: /var/log/ldap
Dec 29 22:43:52 ormoc slapd[685]: conn=319 fd=34 ACCEPT from
        IP=192.168.20.11:32925 (IP=0.0.0.0:389)
Dec 29 22:43:52 ormoc slapd[685]: conn=319 op=1 BIND dn="" method=128
Dec 29 22:43:52 ormoc slapd[685]: conn=319 op=1 RESULT tag=97 err=0 text=
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=2 SRCH
        base="ou=People,dc=mytestlan,dc=com" scope=1
        filter="(&(objectClass=posixAccount)(uidNumber=601))"
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=2 SRCH attr=uid userPassword
        uidNumber gidNumber cn homeDirectory loginShell gecos description
        objectClass
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=2 SEARCH RESULT tag=101 err=0
        nentries=1 text=
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=3 SRCH
        base="dc=mytestlan,dc=com" scope=1 filter="(uid=jsmith)"
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=3 SEARCH RESULT tag=101 err=0
        nentries=0 text=
```

```
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=4 SRCH
        base="ou=Group,dc=mytestlan,dc=com" scope=1
        filter="(&(objectClass=posixGroup)(memberUid=jsmith))"
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=4 SRCH attr=cn userPassword
        memberUid uniqueMember gidNumber
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=4 SEARCH RESULT tag=101 err=0
        nentries=0 text=
Dec 29 22:43:53 ormoc slapd[685]: conn=319 op=5 UNBIND
Dec 29 22:43:53 ormoc slapd[685]: conn=319 fd=34 closed
```

15 Appendix E – Sniffer Display

15.1 passwd (cipher text)

$ passwd

```
##
T 192.168.20.11:33392 -> 192.168.20.10:389 [AP]
  ....X......^U=..7.I.A.ui.)Z...7.{.G...K8..$.d..K.u=..y....'.A...I.*@zC3.Up}
  ..0..`*...Az.. <.
#
T 192.168.20.10:389 -> 192.168.20.11:33392 [AP]
  ..... ...".>..'.h.Q.......V......(3Z..(z.k5......(...YA@...k~3 ..0a...3'C.
##
T 192.168.20.11:33392 -> 192.168.20.10:389 [AP]
  ........6F. .N.^.....r]9.`..S..2.....3.......7...B+..u./....}p....B.<......
  ..|..(...;.....H=.O...D.&.+.........a.P...HY..B}.3...&......:.......F,Q..d
#
```

15.2 ldapsearch (clear text)

$ ldapsearch -x -b 'dc=mytestlan,dc=com' -D ‘cn=manager,dc=mytestlan,dc=com’ \
-W -H ldap://ormoc.mytestlan.com '(objectclass=*)'

```
########
T 192.168.20.11:33141 -> 192.168.20.10:389 [AP]
02...`-.....cn=manager,dc=mytestlan,dc=com..mysecret
##
T 192.168.20.10:389 -> 192.168.20.11:33141 [AP]
.
.
(uid=ajones,ou=People,dc=mytestlan,dc=com0...0...uid1...ajones0...cn1...A.
Jones0:..objectClass1+..account..posixAccount..top..shadowAccount0...shadowMax1...
900...shadowWarning1...70...loginShell1.../bin/bash0...uidNumber1...6000...gidNumber1
...6000...homeDirectory1.../home/ajon  es0...gecos1...A.
Jones03..host1+..ormoc.mytestlan.com..baybay.mytestlan.com0...shadowLastChange1.
..1241504..userPassword1$."{SMD5}mnI1+atiPEUevslr7ru3duMdUvw=0....e.....
##
```

15.3 ldapsearch (cipher text)

$ ldapsearch -x -b 'dc=mytestlan,dc=com' -D ‘cn=manager,dc=mytestlan,dc=com’ \
-W –H ldap://ormoc.mytestlan.com '(objectclass=*)' -Z

```
##
T 192.168.20.11:33142 -> 192.168.20.10:389 [AP]
  .....................cfr.z..G..M...{...g...;..,.E0b..S...z. ..!.../]..~..
  .d9...........t..........M$&.._....05.BP...P...@...30..mG.Z.........).B.lQ.
  ...........()?...ADZ....nbd......K........)..|.W....
#
T 192.168.20.10:389 -> 192.168.20.11:33142 [AP]
  ..........(.....6".Q....R.DT,.T.\.i%.N..._z.....U.r
#
```

15.4 ldappasswd (clear text)

$ ldappasswd -x -D ‘cn=manager,dc=mytestlan,dc=com’ –W –S \
–H ldap://ormoc.mytestlan.com ‘uid=ajones,ou=People,dc=mytestlan,dc=com’

```
interface: eth0 (192.168.20.0/255.255.255.0)
filter: ip and ( port 389 )
####
T 192.168.20.11:32859 -> 192.168.20.10:389 [AP]
  02...`-.....cn=manager,dc=mytestlan,dc=com..mysecret
##
T 192.168.20.10:389 -> 192.168.20.11:32859 [AP]
  0....a........
##
T 192.168.20.11:32859 -> 192.168.20.10:389 [AP]
  0Y...wT..1.3.6.1.4.1.4203.1.11.1.907.(uid=ajones,ou=People,dc=mytestlan,dc=
  com..newpassword
#
```

15.5 ldappasswd (cipher text)

$ ldappasswd -x -D ‘cn=manager,dc=mytestlan,dc=com’ –W –S \
–H ldap://ormoc.mytestlan.com ‘uid=ajones,ou=People,dc=mytestlan,dc=com’ -Z

```
####
T 192.168.20.11:32862 -> 192.168.20.10:389 [AP]
  0....w...1.3.6.1.4.1.1466.20037
##
T 192.168.20.10:389 -> 192.168.20.11:32862 [AP]
  0....x........
##
T 192.168.20.11:32862 -> 192.168.20.10:389 [AP]
  .z....Q... ..............f..................e..d..c..b..a..`...........@....
  ..................F.w.~.F.x5.........FBV..RO....:.
```

#

16 References

Berkeley Database Download. Sleepycat Software. http://www.sleepycat.com/download/index.shtml.

CERT Coordination Center. "CERT/CC Advisories." Software Engineering Institute, December, 2003. http://www.cert.org/advisories/.

"Common Vulnerabilities and Exposures". The MITRE Corporation, 2003. http://www.cve.mitre.org/.

Danen, Vincent. "Using OpenLDAP For Authentication." MandrakeSoft SA, June 18, 2002. http://www.mandrakesecure.net/en/docs/ldap-auth.php.

Elson, David "Del". "Authentication on Linux Using Open LDAP, Part Two." SAAS, July 9, 2001. http://www.saas.nsw.edu.au/solutions/ldap-auth2.html.

Findlay, Andrew. "Security with LDAP." Skills 1st Ltd, February 2002. http://www.skills-1st.co.uk/papers/security-with-ldap-jan-2002/security-with-ldap.html.

"GQ LDAP Client." BIOT.COM. http://biot.com/gq/.

Granquist, Lamont. "Nmap guide." Apr 5, 1999. http://www.insecure.org/nmap/lamont-nmap-guide.txt.

Hodges, J., R. Morgan and M. Wahl. "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security." Network Working Group, RFC 2830. The Internet Society 2002. http://www.isi.edu/in-notes/rfc2830.txt.

Howes, Timothy A., et al. Understanding and Deploying LDAP Directory Services. Macmillan Computer Publishing, 1999.

"LDAP Authentication for Linux." Metaconsultancy, 2002. http://www.metaconsultancy.com/whitepapers/ldap-linux.htm.

"MigrationTools". PADL Software Pty Ltd, 2003. http://www.padl.com/OSS/MigrationTools.html.

"MigrationTools Download Site". PADL Software Pty Ltd. http://www.padl.com/download/.

"MIT PGP Public Key Server." Athena Server Operations. http://pgp.mit.edu/.

"ngrep – network grep." The Packetfactory, August 9, 2003. http://www.packetfactory.net/projects/ngrep/.

Nmap Security Scanner download directory. http://download.insecure.org/nmap/dist/?M=D.

"Red Hat Linux 9 General Advisories." Red Hat, Inc. 2002. https://rhn.redhat.com/errata/rh9-errata.html.

"Red Hat Linux 9, Red Hat Linux Reference Guide." Ch 13, 14. Red Hat, Inc. 2003. http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/.

SANS Security Policy Project. "Server Security Policy." The SANS Institute, 2003. http://www.sans.org/resources/policies/Server_Security_Policy.pdf.

The OpenLDAP Project . "OpenLDAP 2.1 Administrator's Guide." OpenLDAP Foundation, January 10, 2003. http://www.openldap.org/doc/admin21/.

The OpenLDAP Project . "OpenLDAP Download Site." OpenLDAP Foundation, 2004. http://www.openldap.org/software/download/.

"Vulnerabilities." SecurityFocus, 2004. http://www.securityfocus.com/bid.