Global Information Assurance Certification Paper

Cracking LDAP User Passwords and Associated Exploits

Kenneth R. Dean
April 30, 2004

GCIH Practical Assignment Version 3

Statement of Purpose

I've worked as a senior IT security engineer for many years and have witnessed poor security practices pertaining to LDAP directories implemented in many companies.
There are literally hundreds of articles, manuals, and documentation regarding LDAP security on the Internet. I found them all the same, boring, dry and not unique.
I will take a more intriguing approach with this paper and weave in some of the mundane aspects of LDAP, with security threats that can occur. Some of the exploits I employ are not unique to LDAP and can compromise other aspects of a company's IT infrastructure.
My paper involves the approach of the insider.
The insider can fall into the following categories:

- Disgruntled Employee – An employee looking to do harm to an employer. Generally this individual is easy to spot, the one with the bad attitude. Every company has a certain number of disgruntled employees, but ones who wish to do harm will not advertise their infuriation. This insider will poses knowledge of the company's IT infrastructure and may enlist others. This is the most dangerous type of insider.
- Contracted Employee – This individual may seek to make a profit off of the selling of company information or by obtaining unauthorized company services. The contracted employee may not pose the greatest threat to a company, but they are the most ingenious.
- Others – Building tenants and vendors that may have access to company's IT services. These insiders will be the most stealthiest.

The insider I'm portraying in my paper is the profiteering consultant who will take advantage of a company's lax internal infrastructure. These include the LDAP directory servers being used for development, testing, and quality assurance purposes. The insider will first attempt to locate and gain access to a pc (other than his own) to assist in the exploit. Note: If this were a disgruntled employee, they would utilize this pc to launch their attacks against the company.
Then gaining unauthorized access to this pc, the insider will attempt to gain access to the company's interior LDAP directories. Finally, the insider will perform a dictionary attack on the user's passwords stored on within the LDAP directory.

To allow the reader a greater understanding of some of the exploits centering around LDAP directories, I recommend reading the extras section of this paper first, to get an overview of how a LDAP directory works, additional exploits, and to get a better understand of the how and why an insider may want to target the company's LDAP directories.

The Exploit

Some of the security concerns around a LDAP directory are unauthorized access, data tampering, or denial of service. The exploit I'm targeting is the LDAP directory port 389 'the clear text port' and ldap-enabled applications using the administrator's id. Since many corporations' production directories are hardened, it's the development, testing and quality assurance directories that may be less secure and in many cases are an exact replica of the production directory. What better directory to develop or test against than one that has actual data. These directories are located within the internal network, which makes it harder to attack from the outside. We are not concerned with the kid sitting in his home, but the disgruntled employee or profiteering consultant. An insider could make money selling user information (especially if a directory holds millions of users), providing access to the company's paid online services, circumventing security, and reconnaissance information.

Many corporations believe that their internal networks are secure so security policies regarding the internal network are either lax or nonexistent and many savvy IT professionals know this. Ok, it's not the end of the world if the development or testing directories aren't wired shut. If you're just placing non-production, test data on a directory, there really isn't a security issue. However, placing actual data, including user passwords, private keys, and certificates, on these directories does pose a security risk. Especially, if the 389 port is opened and the administrator login id and password is being used for ldap-enabled applications.

The main exploit the insider in my paper is interested in is gaining unauthorized access to a LDAP directory via the administration account. However, any unauthorized access to a directory is a security risk, but unauthorized access via the administration account will allow the insider to export the directory's infrastructure to an ldif. Taking the ldif off site and analyzing it will reveal a great deal of information pertaining to; users, user group information, corporate structure, network devices, vendors, certificates, ldap-enabled application that store configuration information within the directory and hashed user's passwords.

The final result is to attempt to crack all of the user's passwords stored in the LDAP directory. To do this the insider will have to employ other exploits to obtain this goal.

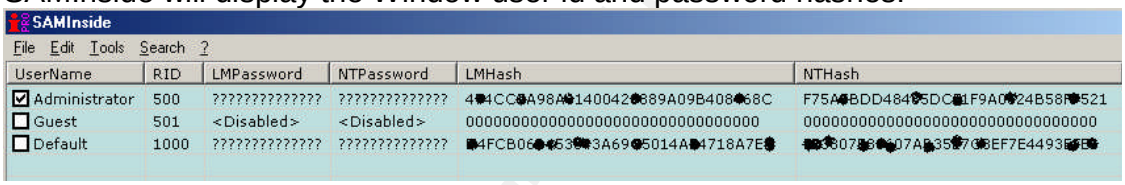
Exploit 1 – Gain unauthorized access to a clueless pc

Name: Bart's Network Boot Disk
Operating System: Windows 9x/ME/NT4/2000/XP or Linux Samba machine
Protocols/Services/Applications: Bart's Network Boot Disk will boot a pc in Windows Dos 98 and supports TCP/IP, Nwlink and Netbeui protocols.
Variant: n/a
Description: The insider will need to get a hold of the SAM and SYSTEM file to initiate the cracking process. Since the SYSTEM file is too large to fit on a floppy, we'll need to copy it to a remote machine. This tool will allow us to boot up a pc from a floppy & map a network drive to another pc.
This tool supports many network cards or you can select from a long list of additional network cards plug-ins to install.
Addition information on creating and running this tool can be obtained at Bart's website: www.nu2.nu/bootdisk/network/
Signature of the attack: none

Name: NTFS Boot Disk
Operating System: Windows 9x/ME/NT4/2000/XP
Description: Enables viewing and copying files stored on a Windows NTFS drives.
Operating System: Windows 9x/ME/NT4/2000/XP
Protocols/Services/Applications: FreeDos
Variant: n/a
Description: This tool will assist us in obtaining the SAM and SYSTEM file stored on a Windows NTFS disk. The only portion of this disk we are interested in is the ReadNTFS.exe. Insert the disk into a pc, a dos menu will appear, select 0 to launch the NTFS reader. Navigate to the 'Logical C:' drive, hit enter to start reading the drive. Then navigate to the desired directory by selecting it and hitting the enter key. To copy a file, navigate to the file, ctrl-c, which will bring up a save operation menu which you can use to select a mapped drive.
Addition information on creating and running this tool can be obtained at NTFS.com website: www.ntfs.com/boot-disk.htm
Signature of the attack: none

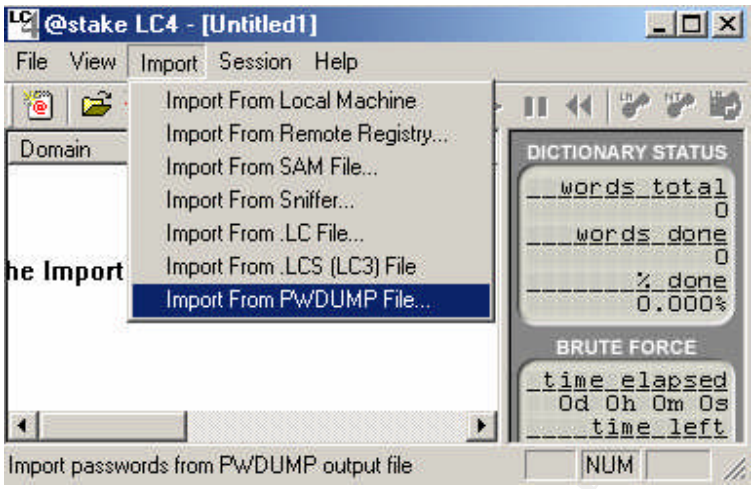
Name: SAMInside
Providing SAMInside with a Windows SAM and SYSTEM file will break the syskey encryption and produce the NT and LM hash of the Window's user accounts passwords. This hash will be exported to a PWDump file that will be cracked by @stake LC 4.
Operating System: Windows XP/2000/NT/ME/98/95
Protocol/Services/Applications: SAMInside needs the Windows SAM file and SYSTEM file, if syskey encryption is enabled. The

SAM and SYSTEM file can be found in either the c:\winnt\system32\config or c:\windows\system32\config directory. The Windows SAM (Security Accounts Manager) file holds user information and passwords. The passwords stored in the SAM are encrypted with the MD5 algorithm. Then the SAM is syskey'd or encrypted with the Window's system key using RC4 encryption.
Variants: n/a
Description: SAMInside is a simple application to use. Execute SAMInside, import the SAM file, SAMInside will prompt you to select the SYSTEM file if the SAM file had been syskey'd, SAMInside will display the Window user id and password hashes.



Export these to a PWDump file.
Signature of the attack: none

Name: @stake LC 4
@stake LC4 (formerly known as L0phtCrack) is a password auditing and recovery tool for Windows user passwords. To utilize this tool you'll need the user password's NT & LM hash generated by SAMInside.
Operating System: Windows XP/2000/NT(SP5 & SP6)/ME 98SE
Protocol/Services/Applications:
Variants: L0phtCrack
Description: @stake LC 4 will allow the insider to crack a Windows user password by providing it will hash user passwords contained in a PWDump file. To start using @stake LC 4 you'll need to go @stake website (www.atstake.com), download @stake LC 4 and install it. If you want the full version including the brute force crack you'll need to purchase a license. After installing @stake LC 4 launch execute LC4 and LC4 it will prompt you with a wizard, just cancel it. Since we all ready have the PWDump file created by SAMInside we'll need to import it.



Signature of attack: None if used remotely with an obtained PWDump file.

Name: Keyboard-Monitoring device
A keyboard-monitoring device is a perfect tool for an insider to use to gain information from a user's pc. This stealthy device can be installed in seconds, and can record millions of keystrokes.
Operating System: All
Protocols/Services/Applications: PS/2 keyboards.
Variants: Most common is Keyghost.
Description: This small, stealthy device is attached between the keyboard plug and the pc. It records every keystroke that the unsuspecting user types in. Removing Keyghost from the pc will not lose the keystrokes.
This is an optional way of gaining the access to an unsuspecting pc other than attempting to crack the password.
Signature of the attack: Spotting the device.

Exploit 2 – Locating and unauthorized access to a LDAP directory

Name: Netcat
Netcat is a back door tool.
Operating System: Unix, Windows 9x/NT/2000
Protocols/Services/Applications: Supports TCP, UDP, and NETBIOS ports.
Variants: n/a
Description:
Netcat can be launched in either client mode or server mode. In server mode it opens up a port and executes a command when a Netcat client communicates with that port. We will use Netcat to gain access to the files that Windump generates.
Signature of the attack:

On the host scan for any unusual ports and unusual applications running in the Windows task manager's processes.

Name: Windump
A network sniffing and analyzing tool.
Operating System: Unix and Windows 9x/ME/NT4/2000/XP
Protocols/Services/Applications: TCP, UDP, relies on WinCap installation and a network card that supports promiscuous mode.
Variants:
Description:
We will use Windump to analyze network traffic targeting traffic to a LDAP port 389, which is the 'clear text' port.
Signature of the attack:
Sniffers are nearly impossible to detect outside of the host machine. Check the Windows task manager's processes on the host to detect unusual applications.

Exploit 3 – Cracking the LDAP directory's user passwords

Name: Setanta
A simple java program that will parse user ids and passwords from a LDAP's ldif and perform a dictionary attack, cracking the password.
Operating System: Any OS that support a JRE.
Protocols/Services/Applications: Java JRE 1.4 or later. Also requires SCOWL (Spell Checking Oriented Word Lists) unzipped in the same directory that Setanta is run from.
Variants: n/a
Description: Setanta works with SHA and SSHA hashed passwords that may exist in a LDAP ldif. It will read through the SCOWL wordlist, hashing the words using either SHA or SSHA algorithm or then compare the hash with the hash contained in the ldif. The initial version just performs a dictionary attack, later versions will incorporate multithreading to enhance performance and brute force attacks.

The Platforms/Environment

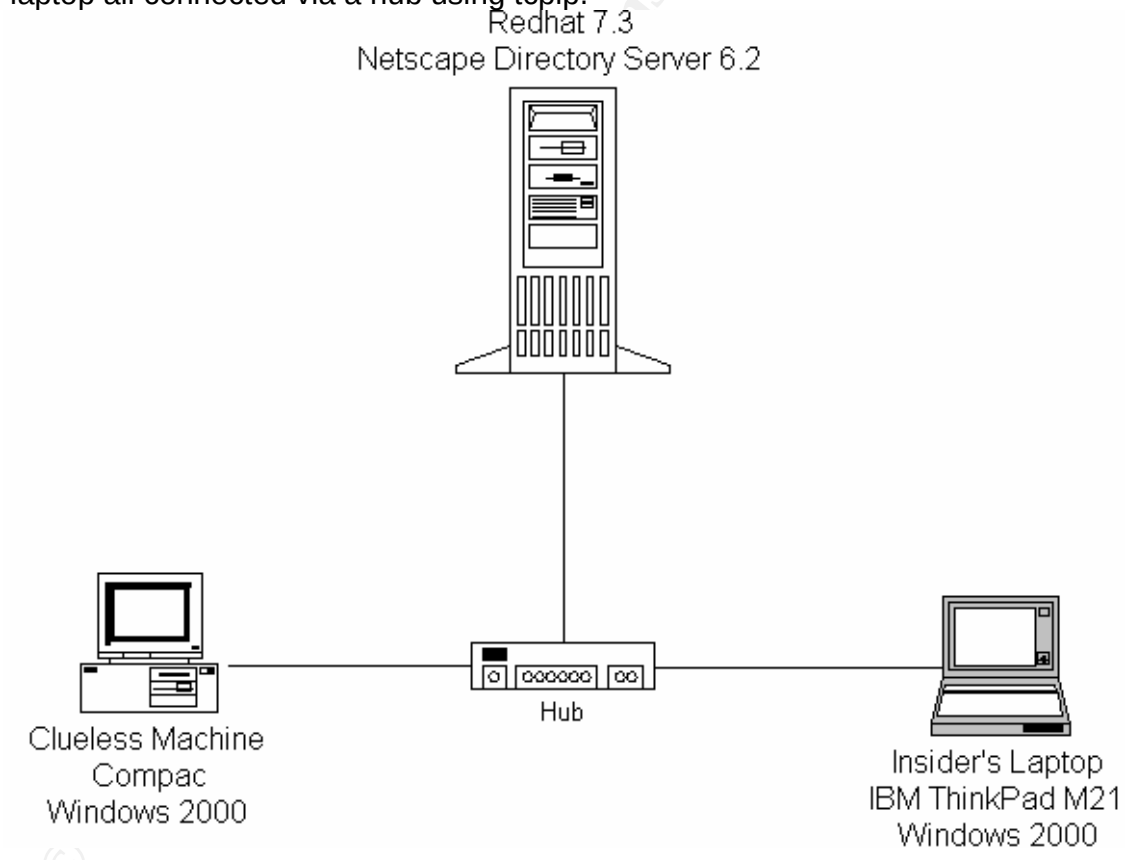
I've setup my lab to mimic many corporations' LDAP development, testing, and quality assurance environment that I've worked with on many assignments. My lab includes a directory server, a laptop, and what I call a clueless pc. The clueless pc will be used to perform the insider's handy work, while distancing themselves.
Victim's Platform
Server:
Netscape Directory Server 6.2

RedHat 7.3 no patches
Generic Server, 1.2mhz, 768m

Clueless PC:
Compac Amanda, 500mhz, 256m
Windows 2000

Insider's Laptop:
IBM ThinkPad, 750mhz, 512m
Windows 2000, SP 3
Redhat 7.3 (separate drive)

Network
This internal test lab contains one directory server, one clueless pc, and one insider laptop all connected via a hub using tcpip.



Stages of the Attack

Reconnaissance

The insider will need to find a clueless pc to launch their scrupulous software from, while keeping a distance from the exploit. A quick look around their area

will reveal many candidates. The choices are an other user's pc or that lone pc sitting in the abandon cube.
Some of the information the insider will need to know about the clueless pc is the operating system, network card make/model, and of course user login id and password. Getting the operating system is as easy as walking up to the machine and wiggling the mouse. The insider will also find out the login id used to access the machine. Finding the network card make/model will be necessary to verify if it supports promiscuous mode.

It would be helpful if the insider knew as much information about the LDAP directory servers as possible. Snooping around the networking department might reveal information about the internal network topology. It's not uncommon for a companies networking topology to be displayed in someone's cube. This will generally outline the corporation's servers, server operating systems, server ip addresses, and applications residing on those servers.
Locating internal ldap-enable application documentation probably will contain information used to connect to the LDAP server, like server host name and/or ip address, login id to connect to the directory and possible, the password.

Scanning

Exploiting the System

A simple and evasive to get the login id and password to the clueless pc would to attach a keyboard-monitoring device to the clueless pc like Keyghost. Wait a couple of days, retrieve the device and view its logs. Not only will the insider get the clueless pc's user login id and password, but other account/password information to other applications and if lucky, some departmental gossip.
If the insider chooses not to go the keyboard-monitor route, they can attempt to crack the SAM file on the clueless pc. To do this the insider will need to get the SAM and SYSTEM file from the clueless pc. Since the SYSTEM file is larger that what a floppy can hold, they'll need a way to copy it to their pc. First, the insider will need two blank floppies. The first floppy will be used to create network boot disk that supports a large range of network cards and supports 9x/ME/NT4/2000/XP or Linux Samba. Well I found one at http://www.nu2.nu/bootdisk/network/. I downloaded the auto-create (only works on NT4/2000/XP), exploded the zip file and ran the auto installer. There is also manual process to create a disk outside of NT4/2000/XP operation systems. However, the auto-create was simple. Now the insider has a network boot disk, they'll need to grab a second blank floppy and find a utility to browse and copy files on NTFS. Well, I found one at http://www.ntfs.com/boot-disk.htm.
Follow the simple instruction to create the disk and after creating the NTFS disk copy the ReadNTFS.exe to the network boot disk. Now the insider has a single disk that will allow them to map a network drive to a remote pc and copy the SAM and SYSTEM file to it.

Walk over to the clueless pc, place a network boot disk in the floppy, turn on the computer and the boot disk guides you thru mapping a drive. Note: when you get to the 'Identification Setting' use your login name and password of your pc.
After successfully mapping a network drive to a remote machine, navigate to the (a:) drive and execute ReadNTFS.exe. Once the NTFS utility has started, navigate to either the C:\WINNT\SYSTEM32\config or C:\WINDOWS\SYSTEM32\config directory and copy the SAM and the SYSTEM file to the mapped drive the insider created to a remote pc.
Now it's time to get the LMHash, by launching SAMInside, which I got from http://www.sharewareorder.com/SAMInside-download-19325.htm. Opened up the SAM and SYSTEM files and presto the insider got the hashes. The insider will want to perform a PWDump on the hashes obtained by SAMInsider. This and the cracking process are outlined at http://www.schizm.netfirms.com/docs/syskeyhackingfinal.htm. Launch your LC4 application, open a new session, import the PWDump and begin the audit. To perform a brute force crack you'll have to have purchase the license.
Now the insider has the username and password to the clueless pc.
Using a keyboard-monitor device may not get you the administrator password, since the user may not be logging into the clueless pc as administrator. However, using the above approach will yield the administration password. By the way, the process of gaining access to someone else's machine isn't limited to exploiting a directory server. You could read their archived email, browse their Internet history, download and upload files.
Now the insider needs a way remotely to gain access, since getting caught on someone else's pc would be hard to explain.
If the clueless pc has a drive shared we could just map to it. But, lets be a little less stealthy about our access and place netcat on the clueless pc. While we are at it we'll schedule netcat to run every time the user starts the machine.
Load our handy dandy netcat program (nc.exe) onto a floppy, boot the clueless pc, login as administrator, and copy the netcat program to an inconspicuous directory. Also, rename the netcat executable to something other than nc.exe, like drwats.exe. This way it won't draw any attention.
We'll need to schedule the netcat program to execute in the background at a specific time. To do this we use the Windows Scheduler program. Create a batch file that will execute our netcat program, for an example I'd call it something like drwats.bat and add the following entry:
C:\<inconspicuous directory>\drwatsnc.exe –L –p 2154 –e cmd.exe

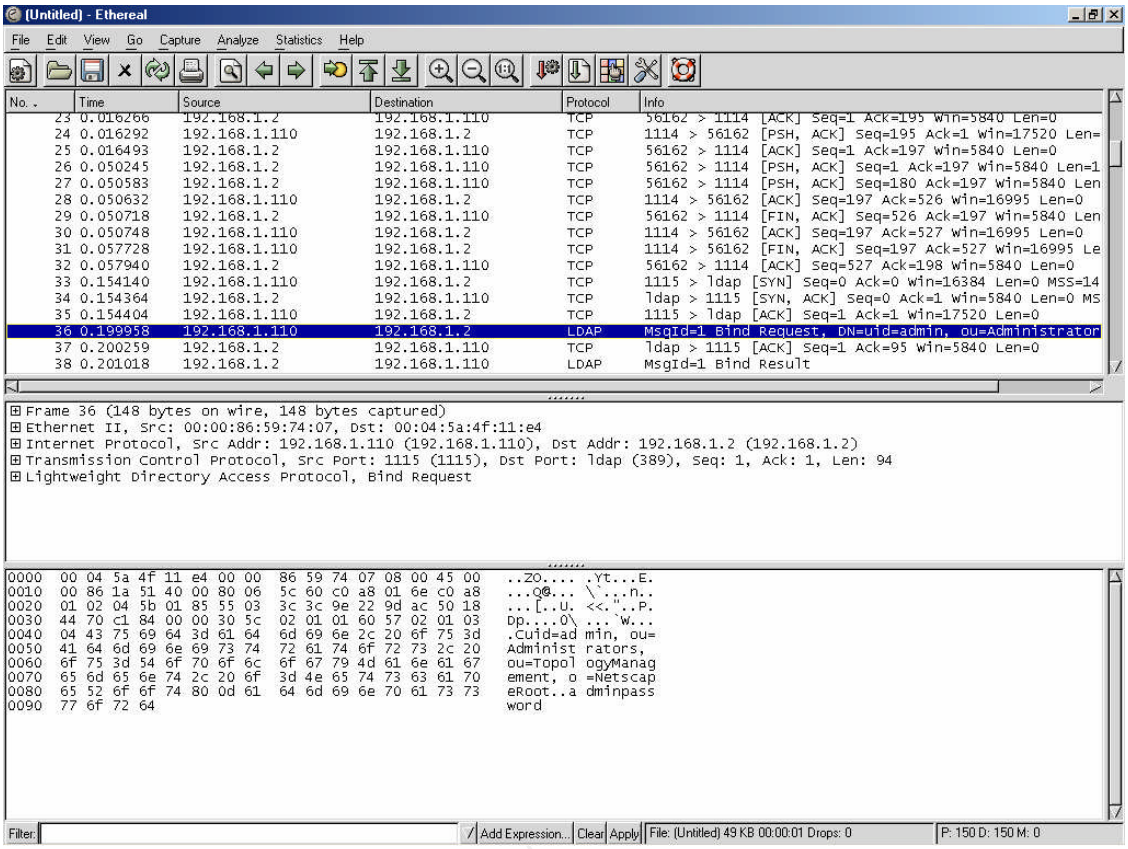
Open a command line and schedule the netcat program:
C:/> at \\127.0.0.1 9:00A C:\<inconspicuous directory>\drwats.bat

At nine o' clock in the morning the Windows Scheduler will kick off the netcat program and you can access the clueless pc's command line from your pc by launching netcat in client mode.
Don't forget to get the ip address of the clueless pc when you're on it and load a sniffer program on it. We'll use windump since it's free and have a copy of

WinPCap on a floppy and ready to install, just in case it isn't installed already on the clueless pc.
Copy windump into an inconspicuous directory, rename it (drwatswd.exe) and attempt to execute it. This will inform you if you need to install WinPCap on the clueless pc. If all is fine, get off of the clueless pc and wait for the netcat program to start at the specified time.
When the specified time has arrived, from your pc launch the netcat program in client mode:
nc <clueless machine's ipaddress> 2154

You should have a command prompt to the clueless pc, navigate to the inconspicuous director and start you windump program:
C:\<inconspicuous directory>\drwatswd.exe –e –vvv –w drwats.ini

The best way to read the drwats.ini file is through Ethereal. Using netcat access the clueless pc and get the file. You could either ftp the file to a Unix machine that you have access or retrieve the file manually at the end of the day. Remember, this file can get very large quickly; you may want to check the clueless pc's disk size to ensure that you won't topple it. Also, it may not be a good idea to run windump too long.
When you get the drwats.ini file open it up in Ethereal and look for the entries that have 'ldap' as the protocol and the word 'Bind' in the info column.
This is when you can start collecting network traffic looking for traffic entries that contains the port number 389 or protocol ldap. O' by the way, the network card on the clueless pc must support promiscuous mode and not be on a switched network. Check the nic card's documentation and the clueless pc's nic card settings to enable promiscuous mode.
As you can see windump gathers a lot of information in a short amount of time. Most of the information is useless, but as you can see there are a lot of jems in the output of windump. Just a quick scan of the output of windump reveals the admin password and admin dn. This information is not only useful to login to the directory server but it starts to reveal the LDAP directory's dit structure.
This is an example of traffic produced by Netscape's Administration Console with port 389 opened.



As you can clearly see, having port 389 enabled is a huge security risk. Not only would the directory administration password be compromised, but also any other user account that 'binded' to the directory.

Armed with the administrator's password and the administrator's dn (distinguished name) we can now login to the directory by either the directory vendor's administration program (Netscape Administration Console for the Netscape directory, ConsoleOne for Novell eDirectory) or a third party ldap browser like LDAP Browser. As admin we can do anything in and to the directory. However, this directory isn't really of interest in to the insider, the user passwords, directory schema, application configuration, and dit structure is more valuable.

So far I've outlined a couple of directory exploits: cracking user passwords to a Windows machine and obtaining a directory's user id and password. Lets outline a third exploit.

This has to do with any directory that exports hashed user password, to an ldif file.

An ldif file is a way to migrate directory objects to other ldap complaint directories.

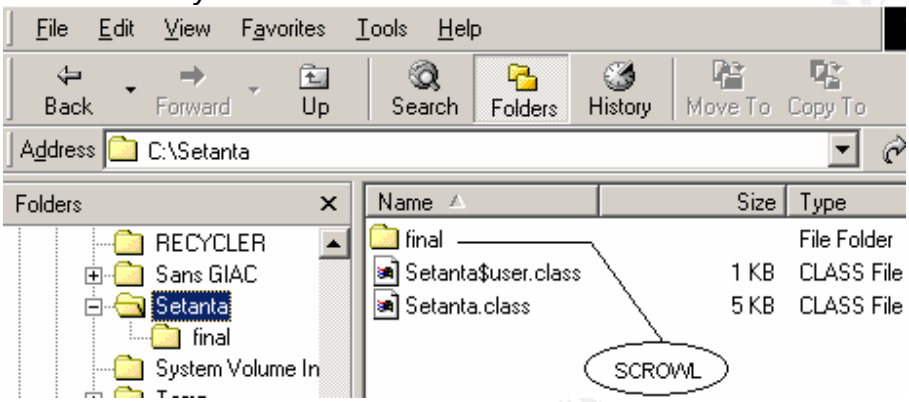
An example of a user entry in an ldif file:

dn: uid=jbreedon,ou=Contractors,ou=People, dc=sullnet,dc=com
uid: jbreedon
givenName: john
objectClass: top

objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: breedon
cn: john breedon
userPassword: {SSHA}BuTUTz2cqdO/lDKm2kWLKQ783G08war3PXVOsQ==
creatorsname: uid=admin,ou=administrators,ou=topologymanagement,o=netscaperoot
modifiersname: uid=admin,ou=administrators,ou=topologymanagement,o=netscaperoot
createtimestamp: 20040402211043Z
modifytimestamp: 20040402211043Z
nsuniqueid: 12335f81-1dd211b2-80968d18-205b0000
parentid: 23
entryid: 26
entrydn: uid=jbreedon,ou=contractors,ou=people,dc=sullnet,dc=com
hassubordinates: FALSE
numsubordinates: 0
subschemasubentry: cn=schema

As we can see there is a lot of information regarding this user. If the user had personal information like a social security number or salary information we would see that too.
But, what is of interest is the attribute userPassword, which is hashed using SSHA (Salted Secure Hash Algorithm). The salted part of the password is hashed with a random value to ensure the hashed values are different. By default, Netscape uses sha (older versions) or ssha to hash the user password. For this exploit I am using Netscape's Directory Server 6.2. This exploit can occur in other directory servers and I am not condoning Netscape's Directory Server. Netscape Directory Server is a very good directory server and properly configured is hard to exploit. The biggest culprit is the lazy or incompetent ldap administrator.
The insider now has the admin's id and password. The insider will need to install Netscape's Administration Console on either his or her own machine (risky) or the clueless pc. The insider can then export the entire directory's entries into an ldif file, including the hashed user passwords.
You can export the directory's objects via a third party ldap tool, but won't get the hashed user password. I can't say that this is 100% true because I haven't tested all of them. But, if you export the ldif file from Netscape Administration Console program you will get the hashed password. Since Novell's eDirectory store its user passwords in a secret store, you'll never get any user password information from an ldif file.
The insider has exported the entire directory entries into an ldif file, and got a great deal of information that can be use. But wouldn't it be nice to extract the user password from the hash. I've searched the Internet and didn't find a tool that would extract each user's hashed password, determine the salt (if ssha is used), multithreaded, and perform a dictionary attach on the hash passwords. Of course this would be done in the comfort of the insider's home. Since the insider has done Java security programming for many years using packages like RSA's crypto-J and IAIK-JCE, he has decided to write his own. There is a great article

on sha/ssha written by John Kristian that can used this as the starting point for an ldif cracker.
The insider created a simple ssha cracker called Setanta that can be used to decrypt the hashed user's passwords contained in an ldif. Setanta is a Java program and uses standard Java packages. It requires a 1.4 Java jre and an English wordlist. The English wordlist can be obtained in zip format from Kevin Atkinson home page called scowl.
To run Setanta create a directory to hold the Setanta.class file and open the scowl-#.zip file in winzip and extract the files within the 'final' directory inside the same directory as the Setanta.class.



Open up a command line; navigate to the Setanta directory and type in the following:
java Setanta userdirectory.ldif

This program will parse the ldif, grabbing all the users ids and user's hashed passwords and performs a dictionary attach on the hashed passwords. Cracked passwords will be displayed, with the user's dn in the command line console. The insider may enhance the program to include multithreading, alphabet characters – special characters replacement, commonly used passwords and brute force. The main purpose in creating this cracker is to demonstrate how easy it was to create.
A person of average intellect, who is Internet savvy, knows a programming language and has a general knowledge of encryption, can write such a program.

Keeping Access

Netcat is used to keep access to the clueless pc or the insider could just map a drive from their pc to the clueless pc. Since the insider can access the LDAP directory via the administrator's account, the insider could also create users account in the directory and give them administration rights.

Covering Tracks

Renaming the Windump and netcat executables is a way of covering the insider's tracks. Once the insider has the information they need, removing the Windump and netcat executables and associated files will avoid detection.

The Incident Handling Process

Preparation

The company has in place standard policies regarding backups, email use, patch management, virus protection, privacy, and password protection. Currently, the password management tool is being tested and hasn't been deployed to production. The online web applications are ssl enabled, protected by LDAP user name and password authentication and logging (error and access) is enable. Since the company rushed these online applications into production, the incident handling process took a back seat, including the associated policies and procedures. The incident handling team would consist of the web server administrator, the backup web server administrator, and myself. It was good to know that the web administrator is a Sans certified professional and has a personal 'jump bag' containing items such as an external cd burner, jaz drive, bitstream backup software, flashlight, tools, blank cds, plastic bags, magic marker, notebook, personal recorder, and walkie talkies. The web administrator being security minded, ensured that the company invested in IDS software and had it properly configured and operating within the production environment.
The online applications were protected by Netegrity Siteminder single sign-on solution, which provides authentication and authorization services and extensive logging.

Identification

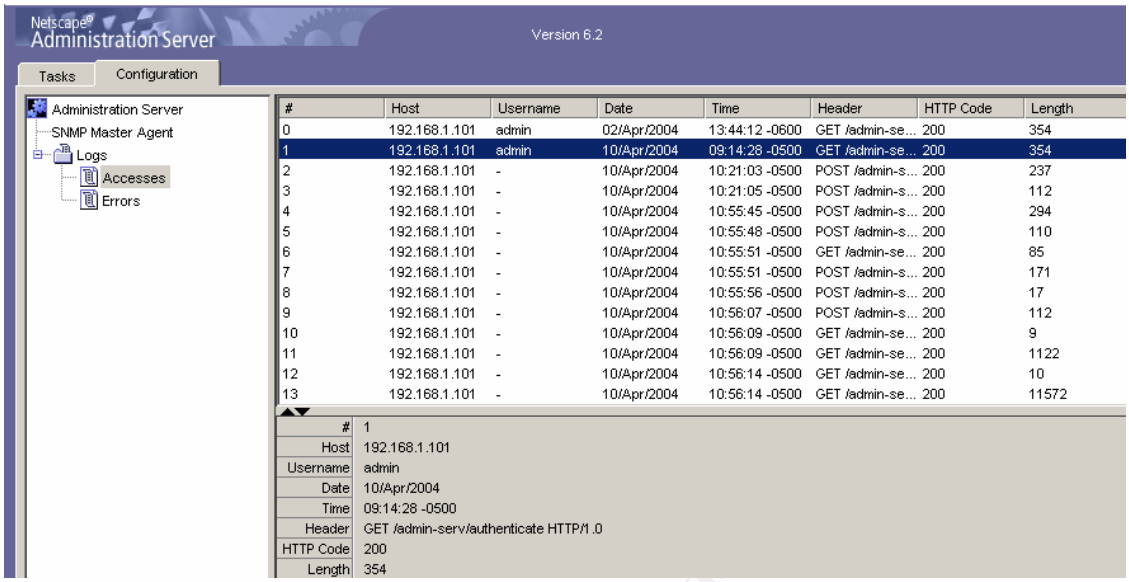
The company that I am contracting with has asked me to help troubleshoot one of their online applications. Users of one of the application have complained that documents they haven't accessed are being marked as 'read'. The development team responsible for the application has run it thought a number of tests, which indicated that the application is working properly.
The web server administrator examined the IDS logs and didn't see anything unusual. The web server administrator then examined the web server's error and access log checking for any failed login attempts or unusual login times. The web server administrator didn't locate any failed login attempt, but did view valid authentication at unusual times. There were many suspicious entries with different users login ids, so it looked like a great number of the user passwords may have been compromised. A check of the Siteminder's logs validated the web server logs. The question was how did so many users' password get compromised, since it didn't appear that a brute force attack has been launched against the online applications.
I instruct the web server administrator to attach the cd burner to the production server and burn two copies of the all the logs (web server, Siteminder, LDAP). One will be labeled and secured for possible evidence, the other for investigation purposes.
An example of an unusual entry in a web server access log.

216.219.253.170 – jsmith [04/Apr/2004:01:55:36 -0700] "GET /sullnet.gif HTTP/1.0" 200 2326 "http://www.sullnet.com/purchase.do" "Mozilla/4.08 [en] (Win98; I ;Nav)"

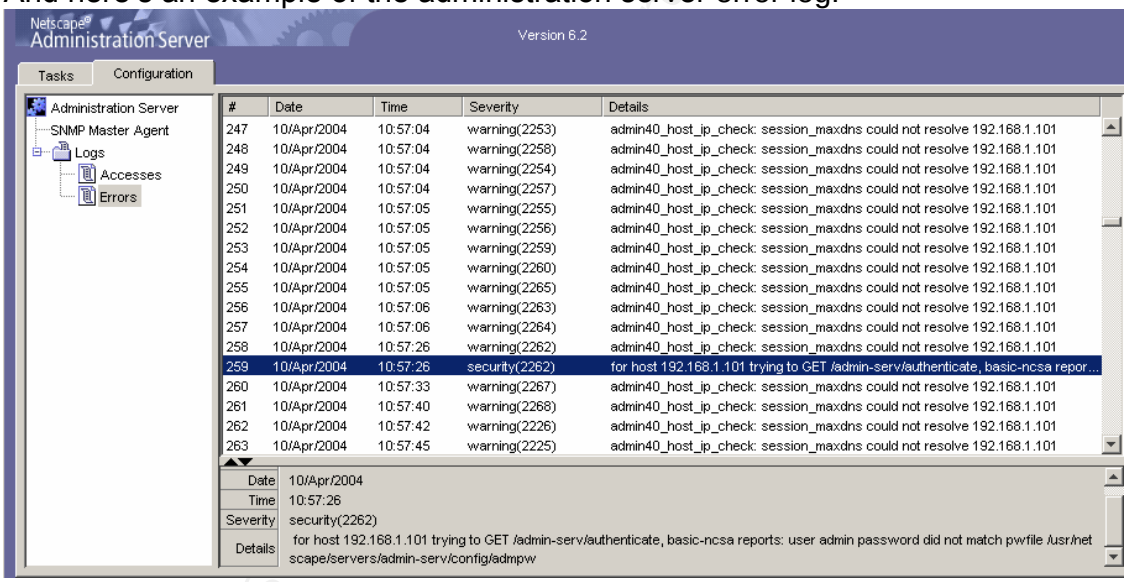
This log entry states that jsmith accessed the company's purchasing application at 1:55:36 in the morning on the 4th of April of this year.
Since I am the senior level person that the scene I assumed responsibility of the incident and informed management of the situation. Management felt that it was premature to involve law enforcement at this time. The backup administrator is tasked with keeping a written account of all activities regarding to the incident.
The applications were SSL enabled and the communications between Siteminder and the LDAP server are secure. Concluding that the production environment hadn't been sniffed.
The company uses the same LDAP directory software and version in all of their environments, with the production LDAP servers being hardened.
We need to examine the log files of Netscape's administration and directory server. Netscape comes with two servers, the actual directory and an administration server used to administrate the directory. Each has a separate logging process and associated log files.
The logs can be view via file or via the administration console and are appropriately named:

- access – used to log access to the directory
- errors – used to log errors relating to the directory
- audit – used to log changes to the directory

It is a good idea to check the administration server access log to see whom, when and where (host or ip address) has been logging into the directory's administration port. There should only be a few machines with in the company that have the Netscape Administration Console installed on them. Obviously, these are the machines that the LDAP administrator uses to administrate the directory.
The directory logs can give the administrator clues as to unauthorized operations being performed within the directory.
An example of the administration access log:

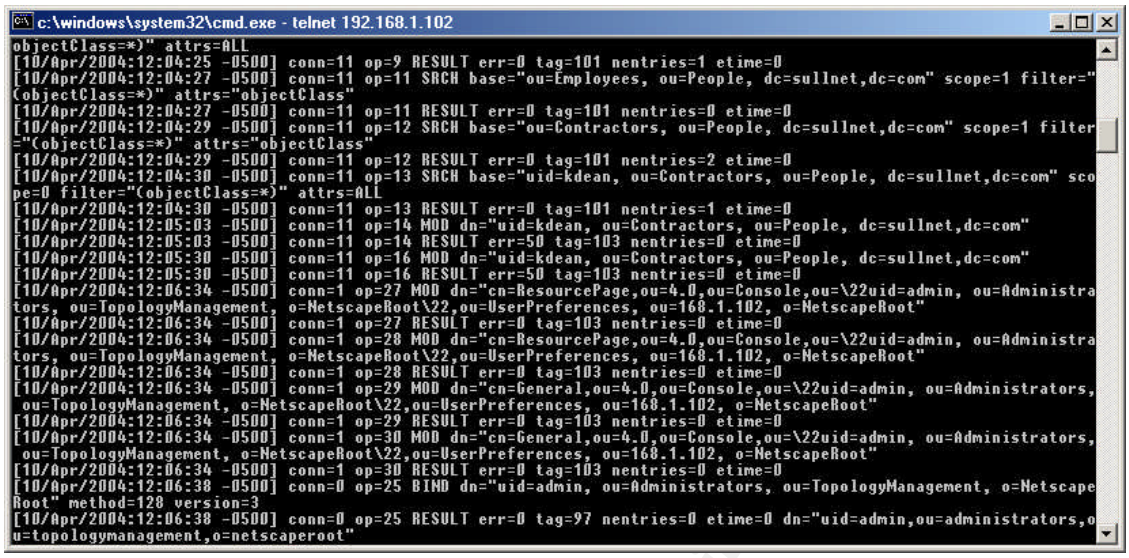


And here's an example of the administration server error log:



Security violations are identified in the severity column as security.

An example of a directory's (not administration) access log entries:



There are a lot entries contained in the access log file, but the ones we are interested in are the BIND (login binds to the directory) and MOD (modifications) entries.
We didn't find any unusual administration activities regarding the production LDAP directories.
Since it is common knowledge that the production and internal LDAP directories are replicated; we turned our attention to the internal LDAP directories log files. Examining the administration server access log we found an unusual ip address accessing the administration server. Generally, only the administration client accesses the administration server via a randomly generated port number.

Containment

All applications that rely on the LDAP directory for authentication and authorization services are potentially compromised. We'll need to keep the directory running for investigative, eradication and recovery purposes, but we will need to protect the online applications. The most effective way to contain the incident is to disable Siteminder, which allows us to keep the web server operational, but prevents authentication to the online applications.

During the identification phase we didn't find any unusual activity-taking place on the server running the LDAP directory or the server's operating system. As we mention above, the production LDAP directory is replicated with the internal LDAP directory. We need to shutdown the replication between the directories due to the possibility that the unauthorized users may have poisoned the directory's entries.

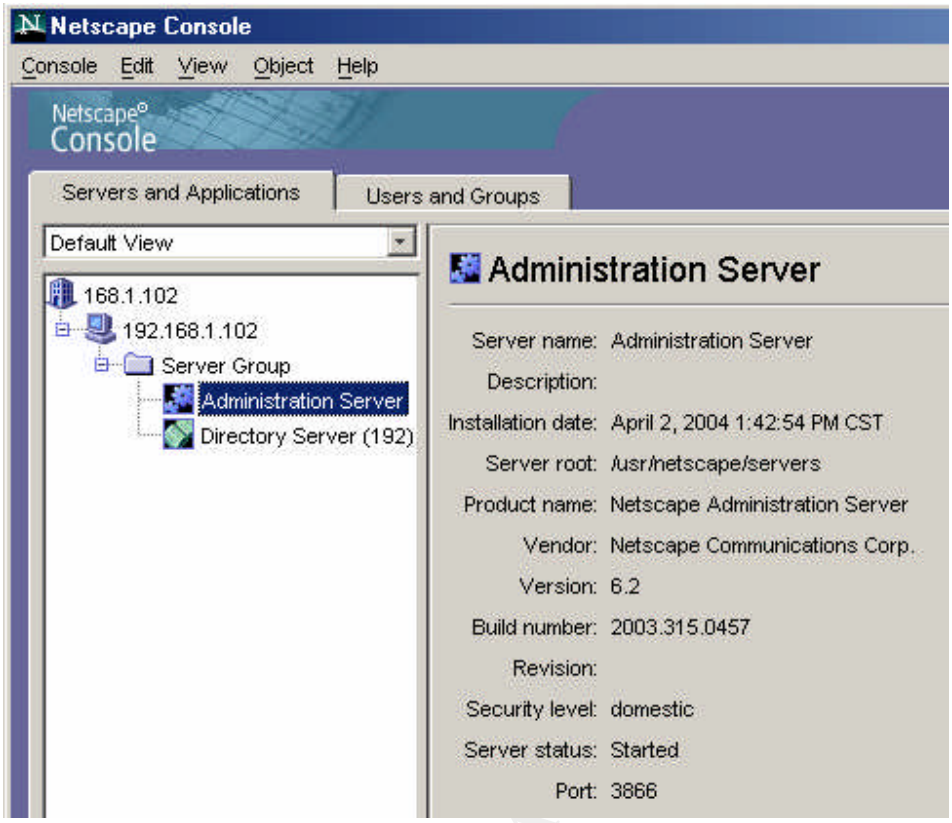
Eradication

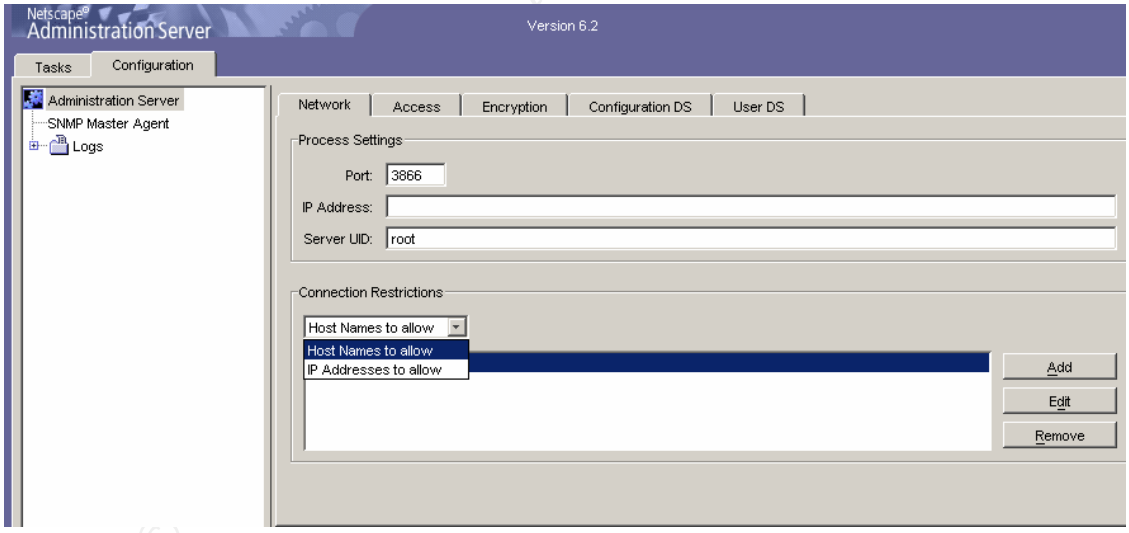
The issues we need to address in the eradication phase is resetting the LDAP directory user passwords, tracking down the pc that accessed the LDAP

administration server and only allow authorized users to administrate the LDAP server.
We have the ip address of the machine that made a suspicious connection to the internal LDAP administration server. A quick check of the dhcp server's active ip table could reveal the hostname, ip address and mac address of the pc. Or we could use nslookup to find the machine's host name.
For example:
C:\nslookup
>set q=all
>1.0.0.127.in-addr.arpa
1.0.0.7.in.addr.arpa name = localhost

The steps regarding the handling the clueless pc are outlined in the recovery section.

The LDAP administration server needs to be configured to only allow certain host names or ip addresses to connect to its administration port. Launching the Netscape Console, logging in, navigating to the administration server, and opening it, will allow us to do this.





While we are at it we might want to ssl enable our internal LDAP servers. The steps are as follows:

1. Generate a Certificate Request
2. Send the Certificate Request to the Certificate Authority
3. Install the Certificate
4. Trust the Certificate Authority
5. Confirm That Your New Certificates Are Installed

Refer to your LDAP Servers Administration's Guide for detailed information.

At this point all the internal LDAP directories data need to be purged.
I recommend that replication between directory servers be utilized only if there is a business requirement. Fail-over, round robin, remote locations, or different LDAP directory vendor are a good reasons to replicate. Internal testing and development directories should never be replicated to any other directory.

Recovery

If, actual data is desired for testing purposes, export an ldif of the production LDAP directory, filter out user passwords, sensitive information, and unnecessary entries and import them into the internal directories. If user authentication needs to be tested then manually set the password. Ensure that all generated ldif files are deleted.

This may be a good time to move the password management tool out of testing and into production. These tools provide a way to force a password change on all users and provide a stronger password policy.

At this point don't know if the owner of the clueless pc is the victim or the perpetrator, so proper handling of the clueless pc is important. The initial assumption is that the owner of the clueless pc conducted the exploit. The machine will need to be confiscated and examined for evidence. This is were the familiarization of the Federal Rules of Evidence, state and local laws, and company's policies and procedures come in handy. Strict logs of activities centered on the investigation of the clueless pc must be kept.
Management needs to be aware of the situation so they can determine if law enforcement needs to get involved. Since the assumption is that the incident doesn't involve public safety and the monetary lose is minimal, the investigation will be handled internally.
First, the clueless pc's owner needs to removed from the pc and interviewed. This can be a touchy process since we don't know if the owner is actually the perpetrator and if the interviewer is too harsh the person may refuse to answer questions. If possible a record of the interview should be make.
Photos of the clueless pc should be taken and labeled.
The clueless pc needs to be powered down hastily by disconnection the power plug.
A bitstream backup of the clueless pc's hard drive needs to be made at this point by attaching a external drive like a jaz drive and booting the clueless pc from a bootable floppy. At least two backups need to be made, one for evidence and one for investigation purposes. The original hard drive and one of the backups hard drives need to be logged, tagged and bagged for evidence. A chain of custody needs to be in place and the hard drives stored in a secured location.
The hard drive used for the investigation (not the original) should be reinstalled into a similar machine and the machine started. We want to examine the event logs (Application, Security, System) and windows scheduler for any unusual entries. The hard drive should be scanned for hacker type software. Since my

exploit renamed the netcat and windump files no identifiable hacker software will be found. But, an entry in the windows scheduler may exists or may not exist. Since the insider is a clever one, they got what they needed and cleaned out the windows scheduler.
At this point it is unclear if the owner is the victim or the perpetrator. I feel if the owner is the perpetrator that there isn't enough evidence to satisfy a court of law. Without extensive analysis of the hard drive, it still may contain hacker software, Trojans, or malicious software that remains a threat to the company. So the machine will need to rebuild.

Lessons Learned

This is the most important of the incident handling steps. This is where the complete assessment of the incident can be evaluated and steps to prevent similar attacks are formulated. Good documentation regarding the incident should be mandatory, and not only for evidence purposes, but it helps justifies additional security measures.
Some of the activities within this step should include:

- A review of the overall handling of the incident.
- The total cost to handle the incident; manpower, lost services, compromised services, lose of company secrets.
- Additional action and safeguards to take.

I've outlined a few exploits centered on directory servers that may exist in a company's development, testing, and quality assurance environments. If the practice is to use actual data within these environments, then these environments should be as secure as the production environment.
Not only did I attempt to raise concern regarding ldap directory servers, I've hopefully demonstrated that managers should keep a close eye on their people. An insider, especially a skilled IT professional can wreak havoc to a company's infrastructure. Unauthorized access to a user's pc via the user or administrator password can go beyond attacking or compromising any ldap directory server. Reading documents, loading virus, launching a denial of service attacks, reformatting the hard drive, sniffing traffic, reading and writing emails is just a small example of what an insider can do once access to a user's pc has been obtained. Security awareness and physical security are simple measures that can prevent unauthorized access to a user's pc.

I've always felt that physical security is grossly overlooked in many companies. Physical security is the simplest and most cost effective form of security there is. To prevent a person for booting up another person's pc is as easy as password enabling the machines bios and disabling boot up on the floppy and cdrom drive. Why would the end user need to monkey with the bios in the first place? The company's pc service group should maintain a database of every machine in the company, including administrator passwords, bios passwords, serial numbers, operational system, hardware, etc… This is a good

practice and makes the pc service team feel important. Including physical security measures in the security policy is also recommended.

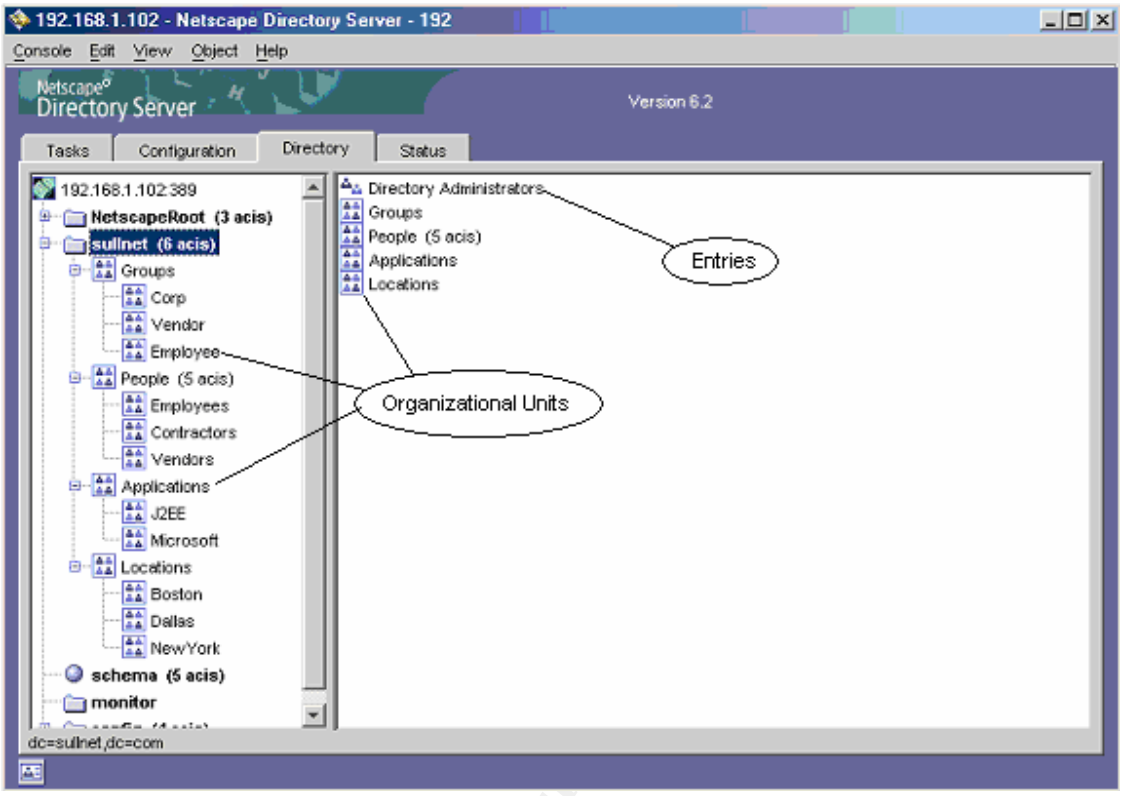
Unless you are a person who checks the back of your computer ever morning, keyboard-monitoring devices are hard to detect. If you are a person who checks the back of your computer every morning, then people will think you are schizophrenic and mistake you for the Windows administrator.
However, it's pretty difficult to install a keyboard-monitoring device on a laptop. A phase prohibiting keyboard-monitoring device in the security policy gives the company ammunition against anyone who has been caught using one.
Installing network cards thru out the company that don't support promiscuous mode will prevent a user from sniffing the network traffic or using a switched network.

It's important to understand the laws centering on computer crimes. The successful prosecution of the insider depends upon how the evidence was gathered, handled, and documented. Policies outlining the gathering and use of information pertaining to an investigation strengthen a company's position in a court of law and provide the incident handlers with guidelines. The enforcement of these policies must remain in the forefront of management.
All employees must have fair notice regarding appropriate activities regarding the company's resources. Warning banners, and clearly posted policies and procedures are examples of fair notice.

Extras

LDAP was born out of X.500 or what was then called DAP (Directory Access Protocol). LDAP is a protocol, not a database or even a directory. It's defines a way a client can access data, not only from a directory server, but also from other data sources like the file system. However, this paper will focus on unauthorized access of LDAP-enabled directories and I will use Netscape Directory Server 6.2 to illustrate this.

Some of the applications that communicate with a directory server are browsers, security related products (single sign-on, identity management, certificate authority), OS, other directories, email, web, and application servers. A directory differs from a database because it's hierarchal (like a file system) and not relational. A directory is sort of a like a corporation's white pages, and may mirror a corporation's organizational structure. The internal structure of a directory is called a DIT (directory information tree) and is made up of component objects and leaf objects (entries). It resembles the file system on many operating systems with the directory objects 'organizational' and 'organizational units' resembling the directory and the entries resembling the files. The main purpose of a DIT is to categorize entries, similar to the way file directories categorize files. The DITs can be either shallow or deep and generally depends on how large (or complex) your corporate infrastructure is. Also, the DIT can span multiple directories on different servers. This design provides enhanced performance, flexibility, and is often referred to as a meta-directory.
An example of a 'shallow' DIT:



If insider with modify rights were to corrupt the DIT on a production LDAP directory, say by renaming some of the organizational units, it will cause all applications that relies on the LDAP directory services to malfunction. Think of what would happen if you renamed the windows 'windows' or 'winnt' directory. If the LDAP directory replicates to other LDAP directories, the effect will ripple.

Every object (except the 'top') in a directory has a parent and is made up of attributes. In the Java world 'top' is equal to Object and attributes are the fields. Attributes can contain either single or multiple values and the attribute can only store data that is a certain type: string and binary being the most common.

The directory schema is a template and dictates what objects and attributes can be stored in the LDAP directory. LDAP directories come with a standard schema that enables the directory to store commonly used objects and attributes like users and user's name. The directory schema can be extended, however, tight control should be used when allowing a schema to be extended to ensure that common objects and attributes aren't removed or changed. If an attribute needs to be added to a directory object it is preferable to use an auxiliary class which is associated with an object and not to create attributes that contain sensitive information.

An example of a directory's schema definition:

```
dn: cn=schema
objectClass: top
objectClass: ldapSubentry
objectClass: subschema
cn: schema
modifiersname: uid=admin,ou=administrators,ou=topologymanagement,o=netscaperoot
modifytimestamp: 20040402194253Z
objectClasses: ( 2.5.6.0 NAME 'top' DESC 'Standard LDAP objectclass' ABSTRACT
      MUST objectClass X-ORIGIN 'RFC 2256' )
objectClasses: ( 2.5.6.1 NAME 'alias' DESC 'Standard LDAP objectclass' SUP top ABSTRACT
      MUST aliasedObjectName X-ORIGIN 'RFC 2256' )
attributeTypes: ( 2.16.840.1.113730.3.1.364 NAME 'nsMCSmtpUseSSL' DESC 'Netscape defined attribute type'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'Netscape Mission Control Desktop - Mail' )
```

Since, a directory shouldn't be used as a database, sensitive information (social security numbers, salary information, etc…) should remain in a database, even though there may be a corresponding attribute. Any user with read access to the users entries will be able to view the information within the attributes.
A directory does store sensitive information like private keys, user passwords, and other authentication credentials. But this information is stored securely in the directory's internal design. For example Novell utilizes a SecretStore to secure its security objects and Netscape Directory Server hashes the user password and stores it in the userPassword attribute.

A directory is generally at the heart of a corporation's authentication and authorization process. Authentication is accomplished by proving who you are. This is accomplished by simple authentication (user name & password), certificate (over ssl), or sasl (simple authentication & security layer). Some directory enables anonymous access for general searching.
For basic authentication, user passwords are hashed using either one of the following algorithms:

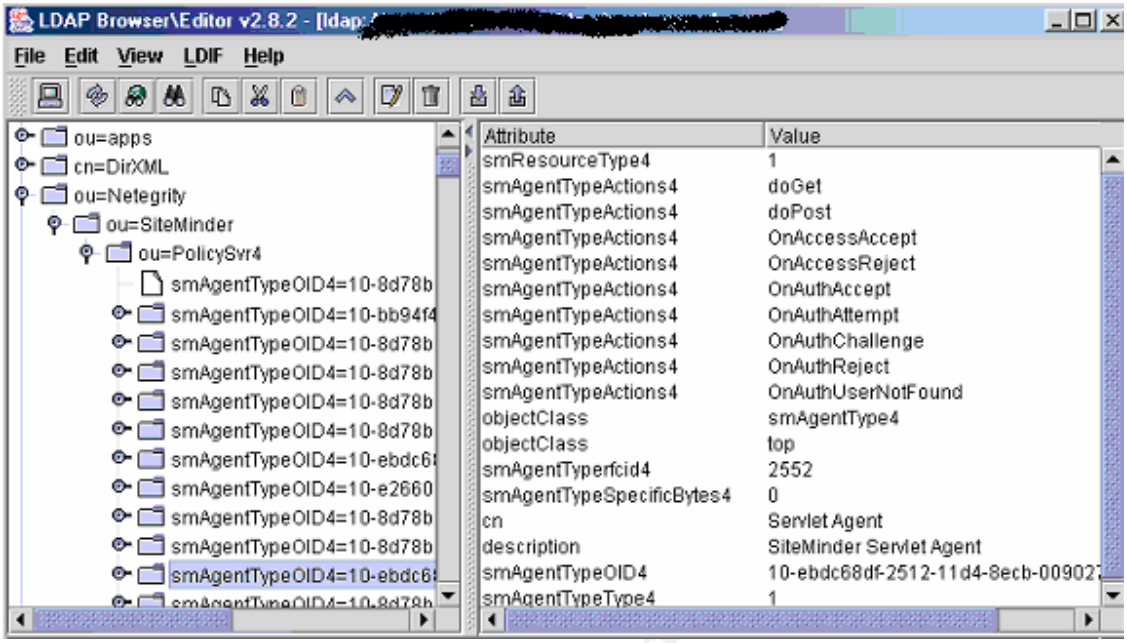
- CRYPT
- SHA
- SHAA
- MD5
- SMD5

Authorization is the granting of permissions to an authorized user and can either be role-based (group membership) or rule-based (attribute value).
Role base approach evaluates whether a user is a member of a group and membership of that group grants access to an application. What type of operations that user may perform within the application is commonly handled by an ACL (access control list) and what type of operations a user can perform within the directory is controlled by an ACI (access control instruction). An ACI is an attribute that can be assigned to any object in the directory, not just a user object.
A user with modify rights could add or remove users from a group's 'member' attribute. This would either allow or deny user access to any application that leverages this group's membership. Modifying a user's 'groupMembership' will achieve the same result.
A rule-based approach evaluates data contained in a user's attributes and grants access and/or permissions based on rules. For example, a company sales application may only allow access if the login user attribute 'departmentNumber'

contains “123” and allow delete operations if the user attribute ‘employeeType’ contains “manager”. The access rule would look something like this: (&(objectclass=inetOrgPerson)(departmentNumber=123)).
A user with modify rights could change an entry’s attribute values, which would break the rule and prevent an application from processing that entry. Or that entry may get processed when it wasn’t supposed to.

What kind of information does a typical directory hold? Basically, a directory can hold anything, but more commonly directory will hold user information, corporation groups (logical grouping of entries), application configuration information, network devices, and security objects. Many third-party PKI vendors utilize a directory to store and organize security objects like certificates, certificates revocation lists (CRL), and private keys.
Special consideration needs to be considered when working with private keys. Misuse of a user’s private key by someone other than that user, can lead to liability issues and tampering with private keys can lead to criminal charges. A legal and though security policy regarding private keys must exists and be enforced.
A directory may be used as a publishing directory, which allows it to interact with a Certificate Manager and/or Registration Manager. This enables storing of CA certificate information like certificate revocation lists (CRL) and end-entity (user) certificates.
User certificate information is stored in the user’s entry’s attribute userCertificate, the CA certificate is stored in the CA entry’s attribute caCertificate, and the certificate revocation list (CRL) is stored in the CA entry’s attribute certificateRevokationList, all stored as binary. The publishing directory may replicate out certificate information to other directories.
Lets say that a company’s infrastructure supported certificate-based authentication and an insider had modify rights. Deleting or corrupting the end-entity would prevent authentication, deleting or corrupting the CRL could deny legitimate certificates or allow illegitimate certificates authentication. Deleting or corrupting the CA certificate would screw up everything, especially if the CA certificate was cross-certified with another CA certificate.

As I mention above, application’s sometime store configuration information within the directory. To do this the LDAP server’s schema is extended to support the objects and the objects are created when the application is first installed.
I will use Netegrity Siteminder as an example of how an application that stores configuration objects within a LDAP directory. But this could pertain to any application that stores configuration information within a LDAP directory.



If a user with modify or delete rights corrupted the right configuration entries, this would create a denial of service attack preventing users from logging in.
Since Netegrity Siteminder is a web single sign-on application, all web applications protected by Siteminder would be unavailable.
If the user had an in-depth understanding of how Siteminder works, they could circumvent Siteminder's authentication and authorization abilities by modifying these objects attributes. An extended amount of time may pass before the Siteminder administrator detects the changes.

Some of the more common directories are Novell's eDirectory, SunOne (formerly known as iplanet), Microsoft Active Directory, Lotus Notes, Netscape Directory Server and IBM SecureWay.
Each directory being ldap complaint, but some incorporate additional functionality like Novell's eDirectory comes with a CA (certificate authority), which provides standard PKI functionality and DirXML, which is a data sync engine.
Directories can be setup to replicate to one another, which enhances fail-over, performance, and load balancing. The replication scheme can be master - slave or master - master.
If a user poisoned the directory's data, even if the directory is a slave directory, can be devastating. In a cascading replication scheme, the slave is the master to another slave. Directory replication must be well thought out and unnecessary information should be filtered out.

Sometimes a company will deploy two directories, one external and one internally. The external directory will either sit in front of the dmz or in it and is used for external customers. The internal directory will generally sit in behind the dmz and is used for internal users and resources. These directories, if possible should be logically separated to prevent one from poisoning the other.

LDAP directory servers can store Java objects by extending the directories schema, allowing Java programs to obtained Java objects remotely.

This allows for easier administration and provides Java applications with a smaller footprint so they can run on a pc or pda with limited storage space. JMS factory objects are commonly stored on LDAP directories, which allow a pc or pda to interact with JMS application remotely. Storing Java objects on a LDAP server protects the integrity of the code. It is easy to de-compile a Java class to view the source, however, de-compiling a Java object stored on a LDAP directory is much difficult. So, what's the downside of storing Java objects on a LDAP directory server, well the Java developer could hide scrupulous code within the Java object, making it more of a Trojan horse object. On the operating system a Java applications runs under the rights of the user that executed it and on the LDAP directory the application obtains the rights of the user id that 'binded' to the LDAP directory. More often that not, these user's accounts have powerful or excessive rights. An insider acting as a LDAP developer could hide code within the Java object and upload that Java objects into the LDAP directory. The Java application gets launched and the hidden code executes.
A solid code review and a strict migration policy can help prevent this.

I feel I made it pretty clear why a hacker or insider would be interested in accessing or corrupting a directory. A directory holds valuable corporation information and if the directory were integrated with another corporation directory, then poisoning the directory would do great harm to your company's reputation.

Referrals

SchiZM http://www.schizm.netfirms.com/docs/syskeyhackingfinal.htm
John Kristian http://developer.netscape.com/docs/technote/ldap/pass_sha.html
Kevin Atkinson http://wordlist.sourceforge.net/
Netscape http://developer.netscape.com/docs/manuals/directory/41/de/
Bart Lagerweij http://www.nu2.nu/bootdisk/network/
Jarek Gawor, LDAP Browser/Editor: http://www.iit.edu/~gawojar/ldap
Peter Stephenson The Role of Forensic Computer Analysis in a Fraud Investigation
http://www.ntfs.com/boot-disk.htm