Global Information Assurance Certification Paper

Breaking Windows 2000 Passwords via LDAP Password Crackers

by

Charles Hamby

Global Information Assurance Certification

GIAC Certified Incident Handler (GCIH)

Practical Assignment Option 1

Version 2.1a

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

TABLES

FIGURES

ACRONYMS AND ABBREVIATIONS

ACL	Access Control List
CD	Compact Disk
CERT	Computer Emergency Response Team
CIRT	Computer Incident Response Team
CVE	Common Vulnerabilities and Exposures
DAP	Directory Access Protocol
DC	Domain Controller
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
Dos	Denial of Service
IDE	Integrated Drive Electronics
IDS	Intrusion Detection System
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
P2P	Peer-to-Peer
PC	Personal Computer
RFC	Request for Comment
SCSI	Small Computer Standard Interface
SNMP	Simple Network Management Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
VPN	Virtual Private Network
WAP	Wireless Access Point
WEP	Wired Equivalent Privacy

1.0 INTRODUCTION

Starting with Windows 2000, Microsoft introduced the Lightweight Directory Access Protocol (LDAP) into its operating systems. LDAP is a Transmission Control Protocol (TCP)-based software protocol whose core purpose is to allow users to quickly search for resources in a tree-based directory. While this speed allows for a great deal of convenience and ease of use, LDAP's speed comes at a cost of some security.

This paper explores how the LDAP can be used to launch a brute force or dictionary password attack against Windows 2000 domain-user passwords. The discussion includes a description of the exploit, a detailed examination of the attack, a discussion of the vulnerabilities inherent in the default installation options in Windows 2000 as pertains to this attack, and the proper methods for configuring Windows 2000 domain controllers (DCs) in order to detect and defeat this attack.

This paper uses a fictitious company, SecNet, and describes an incident response plan from beginning to end. The incident response plan for SecNet covers six phases: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned.

2.0 THE EXPLOIT

This section provides an overview of the exploit itself. Information such as the name of the exploit, the operating systems affected, variants seen "in the wild", attack signatures, and defensive strategies will all be discussed.

2.1 Exploit Name

The name of the exploit described in this paper is Windows 2000 password cracking via the LDAP. There are currently no Common Vulnerabilities and Exposures (CVE) numbers or CERT Coordination Center advisories currently associated with this exploit. For more information, or to check for updated information on the status on CVEs or CERT advisories for this exploit, see the following locations:

- http://www.cert.org/ - The CERT Coordination Center
- http://www.cve.mitre.org/ - Common Vulnerabilities and Exposures List

2.2 Operating Systems Affected

The following operating systems are affected by the LDAP password-cracking exploit:

- Windows 2000 Server SP4
- Windows 2000 Server SP3
- Windows 2000 Server SP2
- Windows 2000 Server SP1
- Windows 2000 Server
- Windows 2000 Advanced Server SP4
- Windows 2000 Advanced Server SP3
- Windows 2000 Advanced Server SP2
- Windows 2000 Advanced Server SP1
- Windows 2000 Advanced Server

In order for these operating systems to be affected, they must be functioning as DCs. Servers that are not functioning as DCs will not have Active Directory installed and therefore will not be vulnerable to this attack.

2.3 Protocols Used by This Exploit

The following protocols can be used by this exploit:

- LDAP (389/tcp)
- Microsoft Global Catalog (3268/tcp)

2.4 Brief Description of the Exploit

Several programs can be used to cause the following exploit: Windows 2000 password cracking via the LDAP. While each tool performs this function in a slightly different manner (see Section 2.5 Variants), the core functionality remains the same. All of the tools bind with the host LDAP directory, enumerate one or more accounts, attempt to crack the password(s) via either dictionary or brute force means and then dump any successfully cracked passwords.

2.5 Variants

There are two major variants of this exploit. The difference between the two variants is primarily based on the way they bind to the LDAP service in Windows 2000, as described below:

1. One variant binds using the distinguished name of the user (i.e., CN=administrator,CN=users,DC=secnet,DC=com). An example of a tool that uses this type of strategy is w2kdad.

2. The other uses the OpenLDAP ldap_simple_bind_s binding function to attempt to bind to the Windows 2000 LDAP service. Tools such as K0ld and bf_LDAP use this functionality.

If all else fails, it is still possible to attempt to manually brute force an account via LDAP using Microsoft's ldp program. ldp is ostensibly used for making updates and changes to Active Directory and making LDAP queries directly to Active Directory. However, it can also be used to manually brute force a domain-level account. LDP is available on the Windows 2000 server compact disk (CD) in the \SUPPORT\TOOLS\SUPPORT.CAB file.

2.6 References

The web sites listed below provide resources to individuals who are interested in this issue:

- bf_LDAP exploit code and explanation, from Insecure.org. Web site: http://lists.insecure.org/lists/vuln-dev/2001/Jun/0215.html

- FX's discussion of how k0ld exploits LDAP's weakness to crack passwords. Web site: http://www.phenoelit.de/kold/docu.html

- Weisman's (brief) description of w2kdad's functionality. Web site: http://www.geocities.com/real_wiseman/w2kdad_readme.txt

- w2kdad Perl script, from SecuriTeam.com. Web site: http://www.securiteam.com/tools/5HP0E209FG.html

- IETF Draft Policy concerning LDAP password policies. Web site: http://www.globecom.net/ietf/draft/draft-behera-ldap-password-policy-03.html

3.0 THE ATTACK

An LDAP-based password cracker has been used against the SecNet company network. This section describes the SecNet network and provides a diagram of the attack.

3.1 Description and Diagram of SecNet Network

The core of the SecNet network is a Windows 2000 domain (http://www.secnet.com) with a Linux Red Hat 9.0 server that is used for tasks that cannot be efficiently carried out by the Windows servers. In addition to its wired clients, SecNet also supports several wireless clients: primarily laptops and a few desktops.

The SecNet perimeter is maintained by a Netscreen-25 firewall. This firewall separates the company network from the Demilitarized Zone (DMZ) and the Internet. The Access Control Lists (ACLs) for this firewall are shown in Figure 1. Setup both in the SecNet DMZ and inside the SecNet perimeter is a Snort Network Intrusion Detection System (IDS) (version 2.0.2). Figure 2 shows the current SecNet network.

ID	Source	Destination	Service	NAT	Action	Option	Configure
7		Outside Any	ANY	✓	✓		Edit Remove
1		Outside Any	Trojan Port Blocker	✓	✗		Edit Remove
4		Outside Any	Kill P2P	✓	✗		Edit Remove
9		Outside Any	Slammer Block	✓	✗		Edit Remove
5		Outside Any	H.323	✓	✗		Edit Remove
3		Outside Any	ANY	✓	✓		Edit Remove

Figure 1: Outgoing Netscreen -25 Firewall Rulebase

The rulebase of the firewall is set-up to meet the needs of the users while at the same time enforcing a reasonable corporate policy. SecNet employees are allowed unrestricted access to Internet with the exception of peer-to-peer (P2P) file sharing, which is blocked at the firewall. All other traffic is allowed out except for certain ports that are considered to be well-known ports for Trojan horse programs, and those are also blocked. The H.323 protocol is also blocked because of historical problems of abuse within the company. Rule seven is setup up specifically to allow outbound access for one computer to a remote third party billing system. This is necessary because of the dynamic way in which the billing software uses high ports to connect to the parent application.

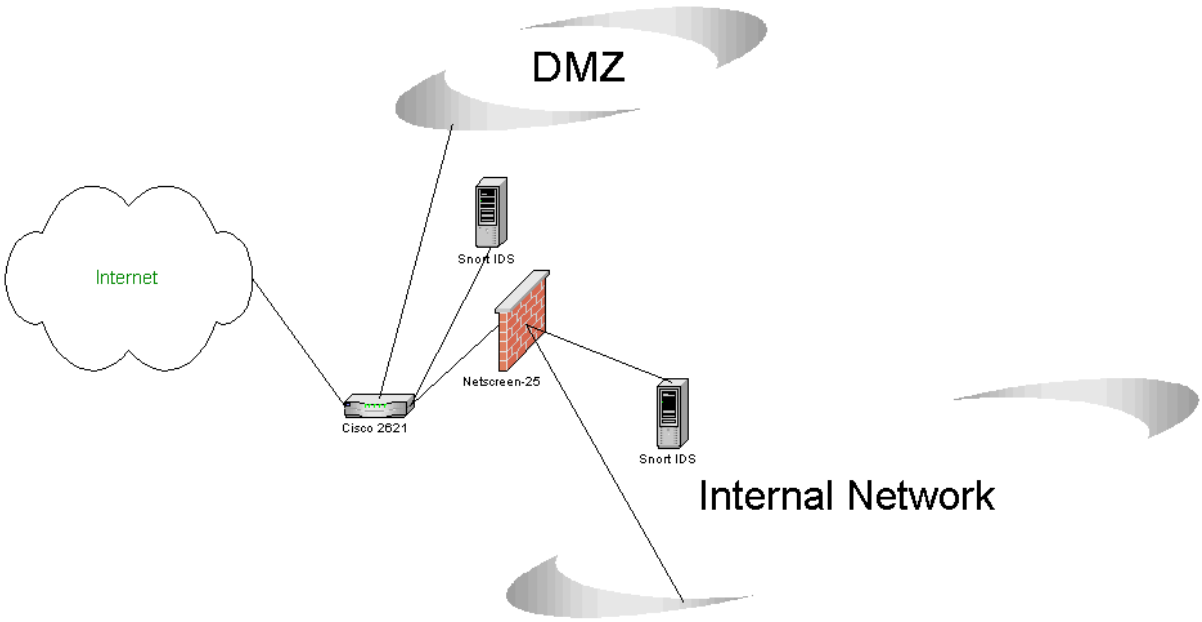

Figure 2: SecNet Network Diagram

3.2 Protocol Description

LDAP, the Lightweight Directory Access Protocol, was designed to provide TCP-based access to X.500 directories while at the same time reducing the overhead associated with the Directory Access Protocol (DAP).

The DAP protocol places much too high of a resource burden on a personal computer (PC) to be used as a directory service protocol. LDAP (also called X.500 lite) was developed as a complementary alternative to DAP that would run over TCP-based networks (by default LDAP listens on TCP 389). (See <http://www.intranetjournal.com/foundation/ldap.shtml> for more information).

The core functionality of LDAP lies in the interaction between the client and the LDAP server. When a client makes a request to the LDAP server, the request is sent to the LDAP server; the server processes the entire transaction and then sends the results of the transaction back to the client. This is different from other protocols where the server and the client may communicate several times during the course of one transaction. According to Wahl, Howes, and Kille, the purpose of limiting communications in this manner is “to minimize the complexity of clients so as to facilitate widespread deployment of applications capable of using the directory.” (The complete text of RFC 2251 can be found at <http://www.ietf.org/rfc/rfc2251.txt>)

3.3 How the Exploit Works

The LDAP-based password crackers start out by binding to the target LDAP server. Depending on the particular variant in use, this is accomplished either by using a simple bind or by binding to a specific distinguished name using Microsoft's naming format.

Once the program has successfully bound itself to the LDAP server, it attempts to enumerate a list of users to run its password-cracking list against. Once again, different programs have different methods of obtaining this list. All of them will have the option of letting the attacker provide a list of specified users or to specify a particular user that the attacker wants to target. Some, however, will have more novel ways of obtaining user lists. According to Wiseman's description of his program, w2kdad uses snmputil.exe, a Windows 2000 Resource Kit Tool that is available for download on the Internet to attempt to enumerate the user list on the target machine via Simple Network Management Protocol SNMP. (See <http://www.geocities.com/real_wiseman/w2kdad_readme.txt> for more information on the functionality of w2kdad).

Once the program has obtained its user list, it then proceeds to start cracking passwords. Here again, different programs operate differently. While some programs are limited to a dictionary-only attack, others give the attacker the option of performing either a dictionary or a brute force attack against the chosen accounts.

Finally, any passwords that are successfully cracked can be written to a file or shown on the screen.

3.4 Description and Diagram of the Attack

Regardless of the variant used, all LDAP-based password crackers rely on the premise of binding to the LDAP server and attempting numerous username and password combinations until the cracking program either runs out of combinations or a valid username/password combination is found. While the actual methods may vary from program to program, the steps for LDAP binding remain consistent. It is interesting to note, however, that only the first two steps of the LDAP binding process are required to perform an LDAP brute force attack, as the second step of the binding process is where authentication is determined.

Once an attacker has obtained the fully qualified domain name and the Internet Protocol (IP) address of a Windows 2000 DC that he or she wishes to attack, the next step is to give the information to the cracking program. The attacker also has the option of specifying customized username and password lists. Once all of this information and any optional switches have been provided, the cracking program will then initiate a connection with the LDAP server and attempt to bind with it. For example, w2kdad uses the following lines of Perl script to initiate the binding process with the targeted LDAP server (SecuriTeam; see <http://www.securiteam.com/tools/5HP0E209FG.html> for the w2kdad source code):

```
print "\n--| Connecting to $host...";
$ldap = Net::LDAP->new($host) or die "$@";
print "done! |--\n";
```

Once w2kdad has established the connection, the various cracking programs will then attempt to enumerate users or brute force username/password combinations, or both. Programs such as bf_LDAP have the ability to try to brute force passwords. This ability can be particularly useful against a single account or when attempting to lockout accounts. The relevant section of bf_LDAP's source code is shown below (Insecure.org; see <http://lists.insecure.org/lists/vuln-dev/2001/Jun/0215.html> for the bf_LDAP source code):

```
ldap_memfree(ret);
if (debug == DEBUG_YES)
printf ("Success\nStart brute force attack...\n");
if (!user_list) {
snprintf (username, sizeof(username), "CN=%s," user);
strncat (username, ldap_user_path, sizeof(username)-strlen(username));
strncat (username, domain, sizeof(username)-strlen(username));
```

The LDAP server will respond to the enumeration by either accepting the given username/password combination or rejecting it. If the LDAP server rejects the username/password combination, the password cracker simply moves onto the next one. If the LDAP server accepts the combination, the password cracker logs that combination and moves on to the next possible combination.

Once the program has launched the exploit and has enumerated valid users and their respective passwords, it will then either dump them to the screen or output them to a text file for later use. This will allow the attacker to later log on to the network as a legitimate user with a valid password, where the attacker can do whatever he or she wants, based on any access control restrictions on the compromised account(s). A simplified diagram of the attack process is shown on Figure 3.

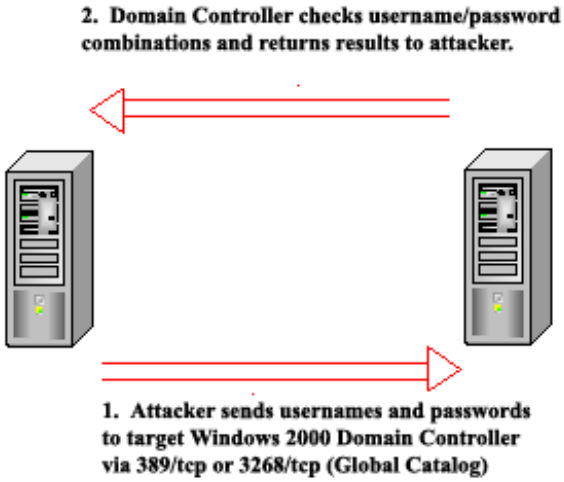

Figure 3: Simplified Diagram of the LDAP Password Cracking Process

As noted in Section 2.5 Variants, in the event that the attacker does not have an automated tool to try many usernames and passwords, it would still be possible to exploit this vulnerability manually using Microsoft's LDP tool. This method would be substantially slower and more time consuming than the use of w2kdad, but if the attacker only needed to try one or two usernames or passwords or did not have access to the aforementioned tools, this would be a viable attack strategy.

3.5 Signature of the Attack

LDAP-based password attacks, like many other brute force and dictionary attacks, leave a distinctive signature in the Windows 2000 security log if auditing has been enabled. If auditing has not been enabled, however, then it will be very difficult to successfully detect this attack.

Once auditing has been enabled, an attack of this nature can be seen after the fact by reviewing the security log. It will quickly be apparent that an attack has taken place by the sheer number of failure audits that are in the security log. A large number of Event ID 681 (Account Logon Audit Failure) occurrences, immediately followed by Event ID 529 (Account Logon / Logoff Audit Failure) occurrences, is an excellent indication of a brute force or dictionary password attack. Additionally, if the Windows 2000 lockout policy has been enabled, which it is not by default (see Section 3.6 The Problem with Default Windows 2000 Installations for further discussion on this deficiency), then Event ID 644 (Account Lockout) will also be present after the specified numbers of account logon failures have occurred to trip the lockout threshold. If the attacker is unaware that the domain is using an account lockout threshold, then the security log may show "loops" where the account is locked out repeatedly as the account lockout duration is reset only to have the account quickly locked out again by the password-cracking

program. Figure 4 is an example of a sequence in a Windows 2000 security log that shows the attack sequence followed by the resulting lockout.

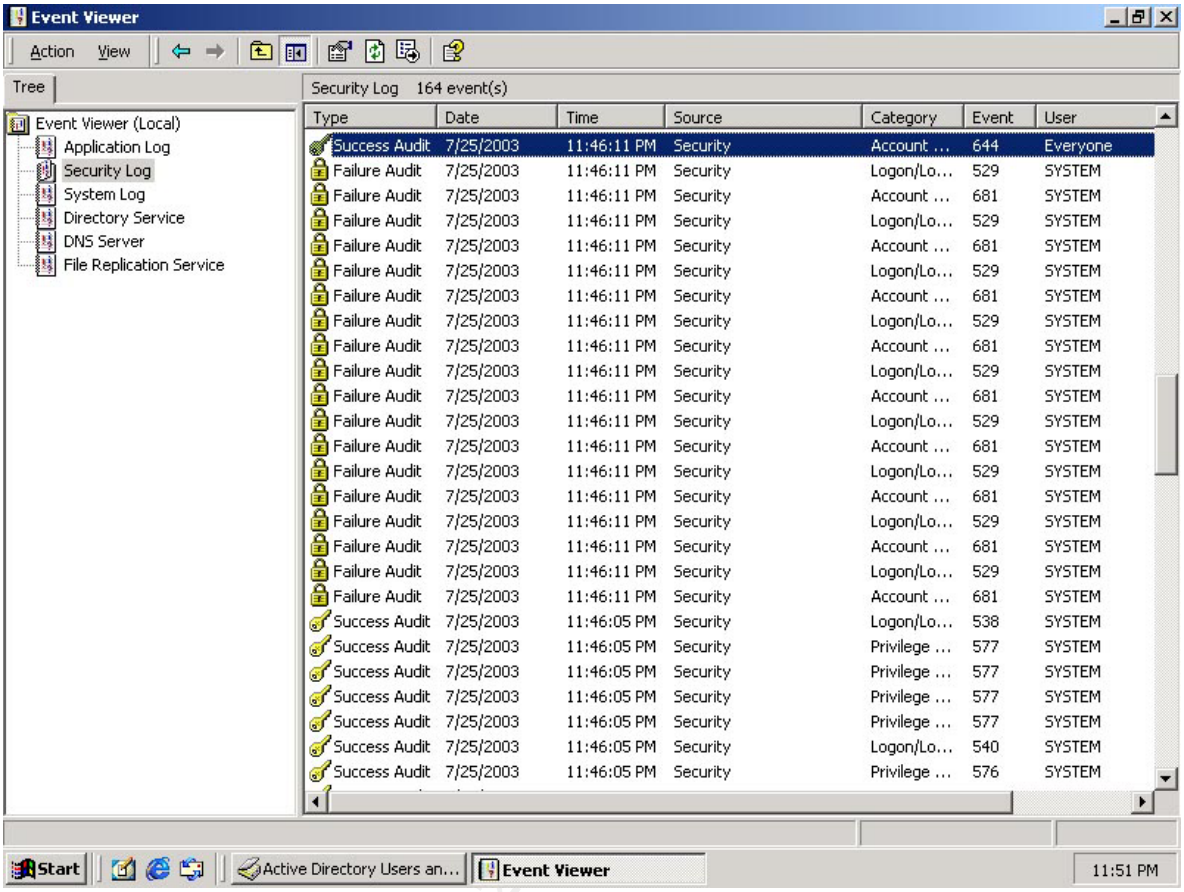

Figure 4: Windows 2000 Security Log Showing Attack Sequence and Subsequent Lockout

As shown on Figure 4, starting at 11:46:11 p.m., the series of Event ID 681's with follow-on 529's (both failure audits) are clearly visible in this DC security log. Upon reaching the preset lockout threshold, the account is locked out and Event ID 644 is recorded (called a "success audit" since it was done successfully). Event ID 681 is indicative of an unsuccessful attempt to log on to a given account (in this case, the Administrator Account). This is immediately followed by Event ID 529, which is an attempt to log on to a valid account with an improper password. The combination of these two events in large numbers for the same account is highly indicative of a brute force or dictionary attack.

After a preset number of tries, Event ID 644 (account lockout) is recorded. This indicates that the account has been locked out for a specified amount of time (in this case, 30 minutes). Figures 5 through 7 depict Event IDs 681, 529, and 644.

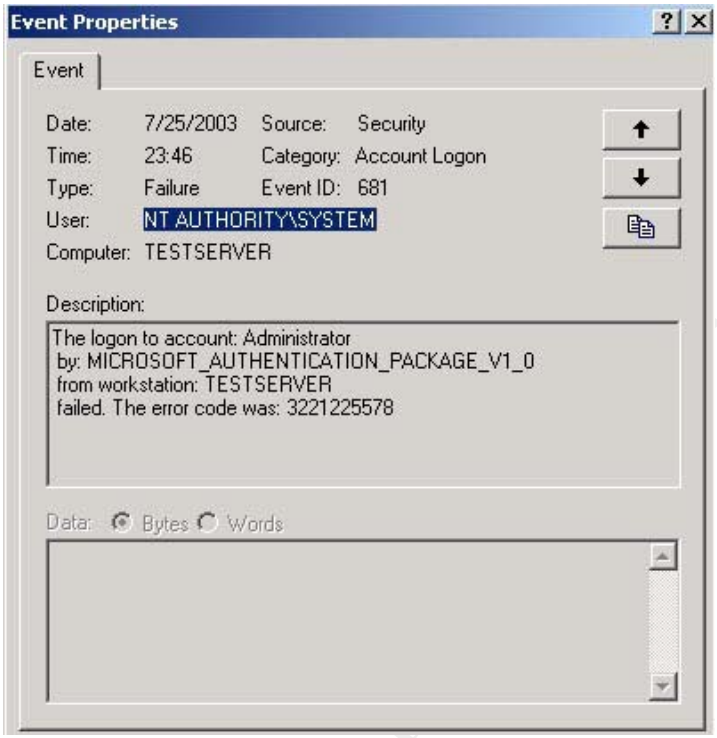

Figure 5: Event ID 681, Account Logon

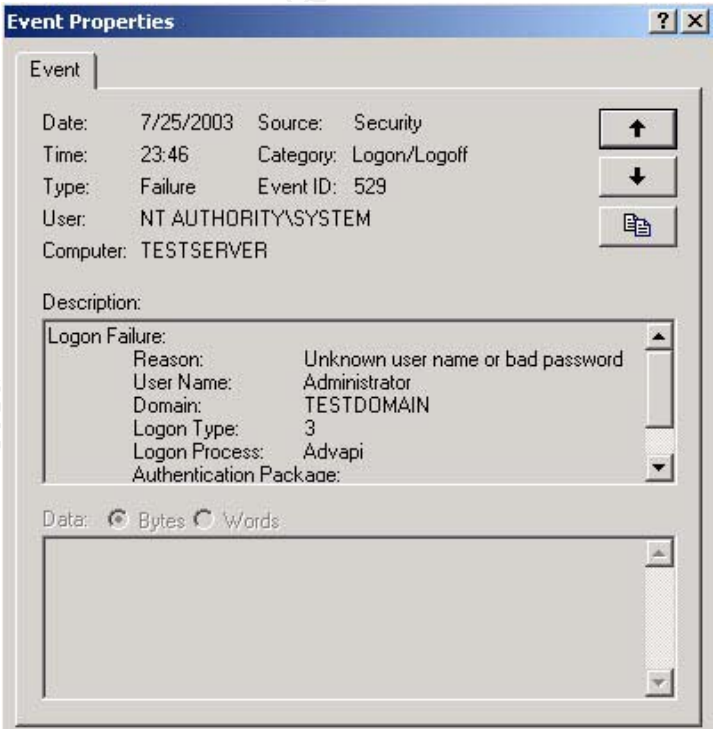

Figure 6: Event ID 529, Logon/Logoff Failure

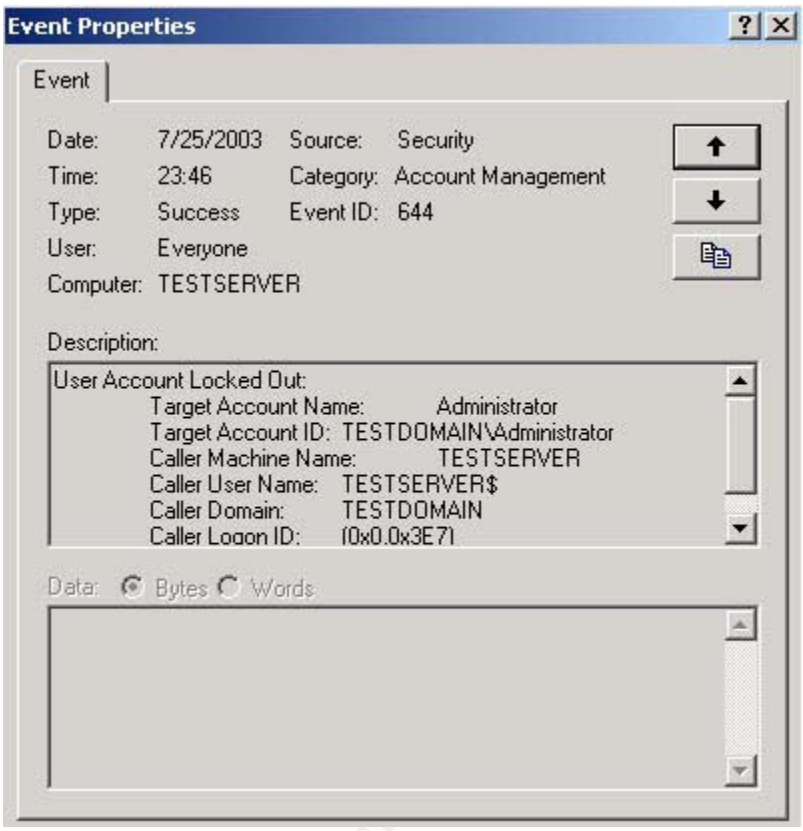

Figure 7: Event ID 644, Account Lockout

By default, Windows 2000 does not offer any other mechanism for specifically identifying password-cracking attacks that are launched via LDAP. One interesting characteristic that is common several of these password crackers is seen at the packet level. When viewing these attacks via Ethereal, some commonalities can be noted. For example, viewing the results of a w2kdad probe shows the following:

```
Frame 114 (137 bytes on wire, 137 bytes captured)
Source: 192.168.0.50 (192.168.0.50)
Destination: 192.168.0.3 (192.168.0.3)
Source port: 2857 (2857)
Destination port: ldap (389)
Flags: 0x0018 (PSH, ACK)
0000  00 a0 c9 5d 0c 50 00 a0 c9 86 8d 2c 08 00 45 00   ...].P.....,..E.
0010  00 7b 91 23 40 00 40 06 27 d4 c0 a8 00 32 c0 a8   .{.#@.@.'....2..
0020  00 03 0b 29 01 85 a5 c7 fc 2b e7 4c 97 bf 50 18   ...).....+.L..P.
0030  7d 78 d3 e8 00 00 30 51 02 01 01 63 4c 04 2c 43   }x....0Q...cL.,C
0040  4e 3d 41 64 6d 69 6e 69 73 74 72 61 74 6f 72 2c   N=Administrator,

0050  43 4e 3d 55 73 65 72 73 2c 44 43 3d 65 6c 64 61
CN=Users,DC=sec
0060  70 63 6f 44 43 3d 6c 6f 63 61 6c 0a 01 02 0a 01
net,DC=local.....
0070  02 02 01 00 02 01 00 01 01 00 87 0b 6f 62 6a 65   ............obje
0080  63 74 63 6c 61 73 73 30 00
ctclass0.
```

This frame was captured as the attacking system was attempting to enumerate the Administrator Account on the target server. Note the presence of the string "objectclass0." During a similar probe using the be_LDAP tool, the following frame was captured:

```
Frame 6 (156 bytes on wire, 156 bytes captured)
Source: 192.168.128.36 (192.168.128.36)
Destination: 192.168.128.5 (192.168.128.5)
Source port: 32785 (32785)
Destination port: ldap (389)
0000  00 01 03 df a6 f7 00 50 56 60 08 d6 08 00 45 00   .......PV`....E.
0010  00 8e 91 a8 40 00 40 06 27 47 c0 a8 80 24 c0 a8   ....@.@.'G...$..
0020  80 05 80 11 01 85 05 ac 65 ff 55 d7 4f 06 80 18   ........e.U.O...
0030  16 d0 a8 12 00 00 01 01 08 0a 00 01 f4 58 00 00   .............X..
0040  00 00 30 58 02 01 01 63 53 04 19 43 4e 3d 55 73   ..0X...cS..CN=Us
0050  65 72 73 2c 44 43 3d 74 65 73 74 2c 44 43 3d 6c   ers,DC=test,DC=l
0060  6f 63 61 6c 0a 01 02 0a 01 00 02 01 00 02 01 00   ocal............
0070  01 01 00 a0 25 a3 13 04 0b 6f 62 6a 65 63 74 43   ....%....objectC
0080  6c 61 73 73 04 04 75 73 65 72 87 0e 73 61 6d 41   lass..user..samA
0090  63 63 6f 75 6e 74 4e 61 6d 65 30 00
ccountName0.
```

This time the string that appears in the capture is slightly different; it is now "objectClass" instead of "objectclass0." While the two strings are not identical, they are similar enough to provide a starting point for a detection signature. In order to be complete, however, it seems pertinent to compare these two packet captures to a packet capture of Microsoft's LDP tool for comparison:

```
Frame 26 (93 bytes on wire, 93 bytes captured)
Source: 192.168.0.50 (192.168.0.50)
Destination: 192.168.0.3 (192.168.0.3)
Source port: 3149 (3149)
Destination port: 389 (389)
0000  00 a0 c9 5d 0c 50 00 a0 c9 86 8d 2c 08 00 45 00   ...].P.....,..E.
0010  00 4f 9d ae 00 00 40 11 5b 6a c0 a8 00 32 c0 a8   .O....@.[j...2..
0020  00 03 0c 4d 01 85 00 3b 24 9f 30 84 00 00 00 2d   ...M...;$.0....-
0030  02 01 01 63 84 00 00 00 24 04 00 0a 01 00 0a 01   ...c....$.......
0040  00 02 01 00 02 01 00 01 01 00 87 0b 6f 62 6a 65   ............obje
0050  63 74 43 6c 61 73 73 30 84 00 00 00 00
ctClass0...
```

Once again the string "objectClass" Is present. It is important to note that LDP uses the User Datagram Protocol (UDP) , thus differentiating it from its password-cracking counterparts, which use TCP.

Based on this, it may be possible to develop a series of intrusion detection system (IDS) signatures to detect LDAP-based password-cracking attempts. A Snort IDS signature would look like this:

```
Alert tcp $EXTERNAL_NET any -> $HOME_NET 389 (msg: "LDAP-based password
cracker in use"; content:"| 6f 62 6a 65 63 74 63 6c 61 73 73 |";
classtype: misc-attack; rev:1;)
```

This signature would allow a perimeter-based Snort IDS to look for packets destined for 389/tcp from the external network that contain the string "objectclass." This type of signature would detect password crackers such as w2kdad.

In order to detect password crackers that contain the string “objectClass,” the signature would have to be modified as follows:

```
Alert tcp $EXTERNAL_NET any -> $HOME_NET 389 (msg: “LDAP-based password
cracker in use”; content:”| 6f 62 6a 65 63 74 43 6c 61 73 73 |”;
classtype: misc-attack; rev:1;)
```

By changing the content string to reflect the signature modification, password crackers such as bf_ldap would be detected. Alternatively, it would also be possible to create an all-purpose signature that would detect all varieties of password crackers by shortening the content string to 6f 62 6a 65 63 74 (“object”); however, this may have the net effect of increasing false positives.

It should be noted that the author wrote these signatures and are currently being beta tested for inclusion into the Snort rule set by Brian Caswell.

The use of an IDS signature to augment the ability of Windows 2000 to detect LDAP-based password crackers would be invaluable, especially in cases where the Windows 2000 DC is new or is still in its default configuration. In these cases such a signature may very well be the only means of detection available.

3.6 The Problem with Default Windows 2000 Installations

When attempting to identify and defeat this attack, one of the first things that a network administrator must understand and remember is that the default configuration of Windows 2000 does nothing to help one detect or identify this attack.

In its default configuration, the Windows 2000 server family is not configured to detect or prevent this type of attack. As shown in Figure 8, the default auditing policy of the Windows 2000 server family is “no auditing”

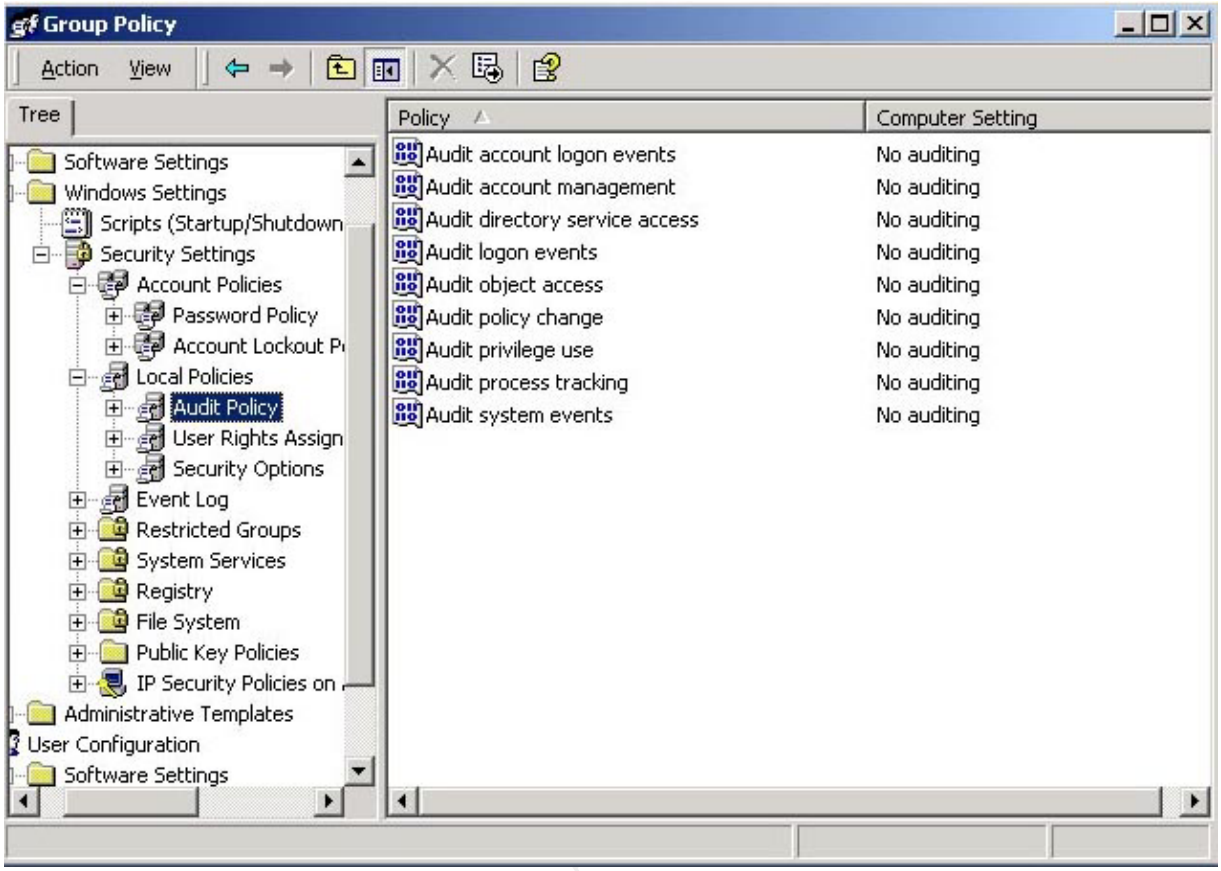

Figure 8: Windows 2000 Server Family Default Auditing Settings

Additionally, Windows 2000 also ships with the lockout threshold turned off by default (see Figure 9). Essentially this means that password-cracking programs such as these can keep trying until they either run out of words or time without fear of tripping any sort of lockout condition.

The net effect of having no auditing policy and no lockout threshold means that an unmodified installation of the Windows 2000 server is potentially in serious danger from LDAP-based password attacks. When properly configured, however, the signature of a brute force or dictionary attack is obvious in the security log. When the Windows 2000 server is left in the default configuration, the evidence of such an attack is nonexistent.

An example of a security log showing an attempted w2kdad attack that was not recorded by the Windows 2000 server is shown in Figure 10. In this example, an instance of w2kdad was run against a Windows 2000 server that was left in its default configuration. W2kdad was given a username file with approximately 78 usernames and a password file with 78 passwords in it. W2kdad verified the existence of one username (administrator), tried all 78 passwords against it, and successfully obtained the password. However, we cannot know any of this from looking at Figure 10 because the

server has been left in default configuration; therefore, there is no security log of the attack.

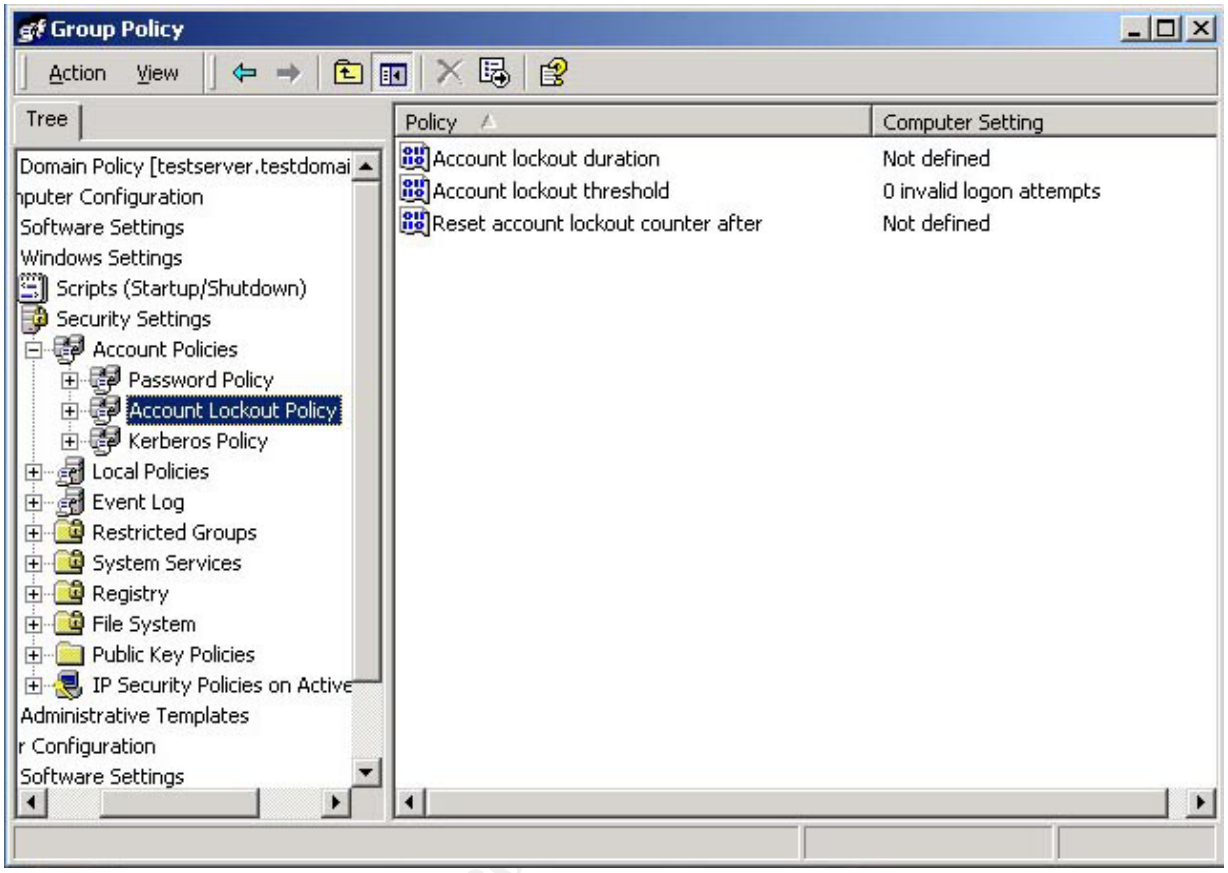

Figure 9: Windows 2000 Default Lockout Configuration

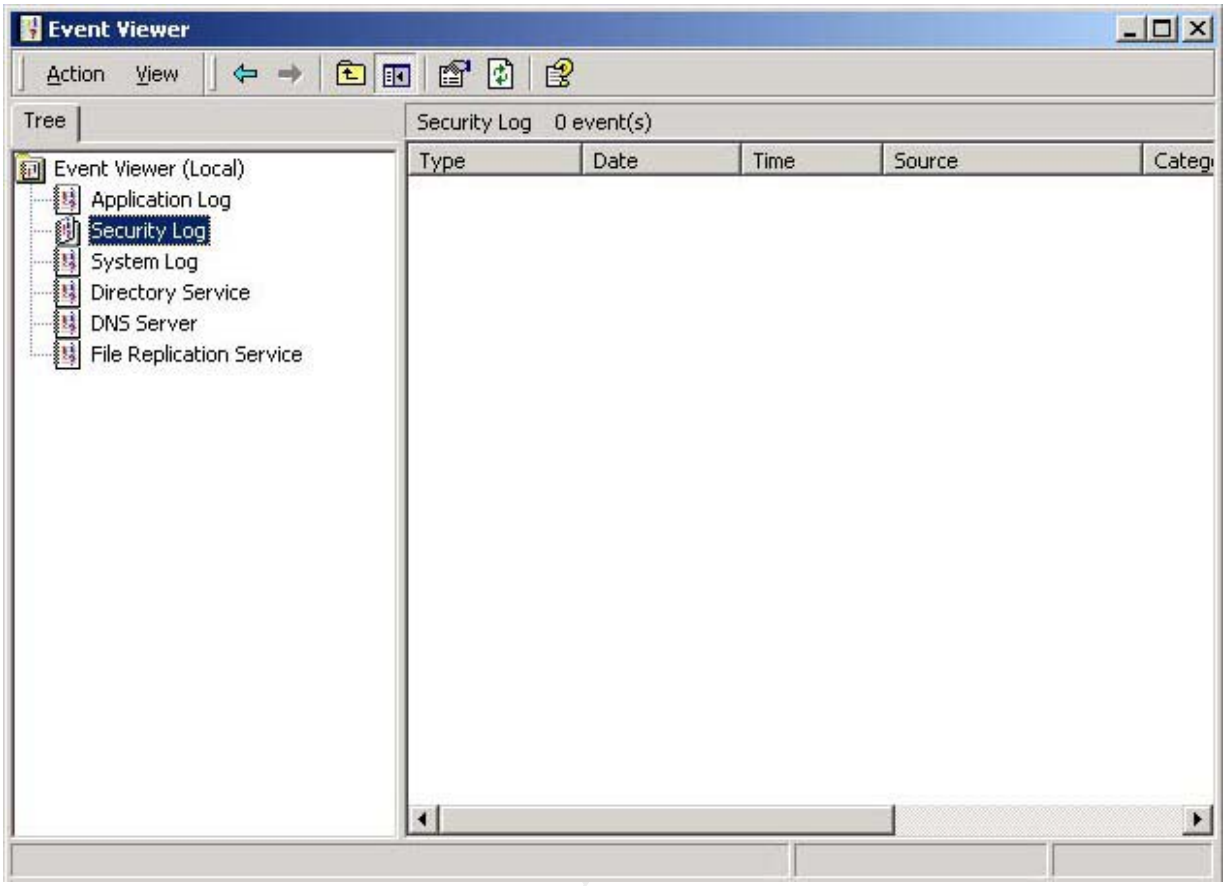

Figure 10: Security Log of Windows 2000 in Default Configuration after a w2kdad Attack against the Administrator Account

3.7 How to Protect Against These Attacks

With dictionary and brute force attacks, defense is the key to success. The first line of defense should be perimeter devices such as firewalls. The network administrator needs to ensure that any port, which can be used to launch an LDAP-based password-cracking attack, is closed. This includes both TCP and UDP ports as well as the Global Catalog ports for Windows 2000 (3268/tcp and 3269/tcp). Additionally, the administrator must ensure that any wireless devices on the network have been properly secured to prevent attackers from launching attacks from behind the firewall.

Once the perimeter devices have been secured, the next layer of defense is the servers themselves. Since Windows 2000 relies on the LDAP as one of the core protocols for Active Directory, it is simply not feasible to block or otherwise disable LDAP, as this would have the net effect of disabling Active Directory. Instead, a sound auditing and account lockout policy will prevent the compromise of accounts from this attack (although this policy can have other residual impacts).

By enabling a lockout policy in group policy, an administrator will ensure that an attacker will be locked out of an account after the attacker attempts to guess the password a

certain number of times. This will prevent a large number of brute force or dictionary guesses against user accounts. It may also have the effect of causing a denial of service (DoS) attack against those user accounts if the attacker keeps trying and does not realize that there is a lockout policy in effect (in fact, the w2kdad program specifically has a feature built in just to DoS accounts).

Enabling Account Lockout Policies: In order to enable the account lockout policies, the administrator should go into Active Directory Users and Computers, right click on the domain, and select properties. Next, select the group polices tab. At this point, the administrator can either select the default domain policy or create a new one.

However, it is important to remember that because of the way that group policy is inherited in Windows 2000, it may be best to ensure that the group policy is set at the DC's organization unit level for security-related matters or to ensure that such policies are set with the "no override" option checked.

Once the administrator has decided how to set up the lockout policy (i.e., either as part of the default policy or as a separate policy), then he or she should select Computer Configuration > Security Settings > Account Policies > Account Lockout Policy. Figure 9 shows the three options that are presented when the Account Lockout Policy menu is selected. The main selection in this menu, Account Lockout Threshold, determines the number of invalid logon attempts before an account is locked out. Setting this too low may cause legitimate users to be locked out if they mistype their passwords once too often, especially if they recently changed their password. Conversely, if this setting is set too high, an account can be brute forced.

Enabling Auditing: Enabling auditing for both logon events and account logon events for success and failure on all DCs ensures that any LDAP-based password-cracking attempts that occur against a server will be recorded in the security log. Therefore, the administrator will have a record of any and all accounts that were successfully or unsuccessfully accessed by the attacker.

In order to enable auditing, the administrator must go into Active Directory Users and Computers, right click on the domain, and select properties. Next, the administrator will select the group polices tab. Then, the administrator should select the policy that he or she wants to create or modify.

Once the group policy has been selected, then the administrator should select Computer Configuration > Windows Settings > Local Policies > Audit Policy. In order to ensure that LDAP-based password-cracking attacks are logged, enable auditing by configuring per the examples in Table 1.

Table 1: Auditing Requirements

Event:	Audit For:
Audit Account Logon Events	Success and Failure
Audit Logon Events	Success and Failure

Auditing for the Audit Account and Audit Logon Events, as shown in Table 1, should only be part of an overall auditing strategy, however, and should not be the only events audited for.

4.0 THE INCIDENT HANDLING PROCESS

An unknown intruder attacked SecNet on the evening of October 1, 2003. This section describes the incident handling process followed by the Computer Incident Response Team (CIRT) and the changes made as a result of the incident.

4.1 Preparation

The Information Technology (IT) Department at SecNet is fairly small and, as such, does not have a dedicated security staff. Nevertheless, the SecNet IT staff, having recognized the growing threat that an insecure network plays to the livelihood of the company, recently persuaded management to send three of their staff to security training. As a result, the IT staff has set up the following guidelines with management approval.

Policy: SecNet has decided that the primary objective when dealing with potential threats and intrusions must be to ensure that any compromised systems are restored to full working order as quickly as possible. After performing a cost/benefit analysis of doing a full forensic analysis of compromised systems and the possibility of prosecuting offenders both criminally and civilly, the company determined that it was not cost effective to do so and that greater cost savings could be achieved through prevention and quick recovery than through any attempt to recover damages in civil court.

Any CIRT Member who has had security training may remove a revenue-generating system from the network if he or she reasonably believes a system has been compromised or if the system contains malware in any form. Any CIRT Member who has not had security training who thinks a revenue-generating system has been compromised or contains malware should first get the opinion of an CIRT Member who has had security training prior to disconnecting a revenue-generating system.

System Maintenance: At this time, SecNet is currently in the market for an effective solution that will allow the company to keep its desktop computers and servers current for all of their various types of software, particularly their operating system patches. The IT staff are currently beta testing various manufacturers' patch management and vulnerability scanners but have yet to find a product that is stable enough and works with all of the software that they use. This is one of their primary concerns, as they are aware that they have several systems, particularly desktops, that do not have up-to-date patches, but they can never seem to "get around to it."

Computer Incident Response Team: The SecNet CIRT is composed of three members of the SecNet IT staff with security training, a network administrator, a member of human resources, and a management representative. The CIRT is headed up by the CIRT Leader (the third SecNet IT staff member, who has received incident handling training).

Response Kit: After the formation of the CIRT, the IT staff asked management for a small stipend (approximately $5,000) with which to purchase a "jump kit" (a kit or bag

containing essential tools and materials that CIRT members require for responding to computer-related incidents). After the IT staff explained the purpose of the jump kit, management agreed and appropriated the necessary funds. The jump kit contains items such as a hub, hard drives (IDE and SCSI), a laptop computer, forensics tools, a USB hard drive for backing up data, a flash drive, spiral bound notebooks for taking notes in and patch cables and other equipment that the team feels may come in handy in an incident.

Dedicated E-Mail: A dedicated e-mail account, incidents@secnet.com, was set up for people inside or outside of the company to report incidents to the CIRT. Aliases of cirt@secnet.com, abuse@secnet.com, and security@secnet.com were also established for this address. These e-mails are automatically forwarded to the CIRT Leader.

Dedicated Communications: The CIRT has decided to purchase walkie-talkies that have approximately a 2-mile range in case an incident occurs. Additionally, each person on the CIRT has a personal cellular phone, and each member of the CIRT has been given a card with the home and cellular numbers of all other members of the team in case of an incident. In the event of an incident wherein personal cellular phones are required for communication for an extended length of time, management has agreed to reimburse CIRT members for the cellular phone minutes used.

Incident Initiation: Every member of the IT staff has the ability to initiate an incident either by e-mailing incidents@secnet.com or calling one of the CIRT members. Once a team member has initially been contacted regarding an incident, that team member will contact of all other team members and start the incident handling process.

4.2 Identification

One of SecNet's junior network administrators recently noticed the addition of two accounts to the Domain Admin's group on one of the servers. Although it was possible that these were simply test accounts, the administrator contacted the CIRT and initiated an Incident.

When the CIRT Leader first arrived, he verified that it was an incident and not simply a matter of staff development. In order to do this, he spoke with the IT staff and asked if anyone had created the accounts for any reason (testing or otherwise). Nobody could remember creating those accounts. Next, CIRT Leader reviewed the information available on the DC.

A review of the list of Domain Administrator Accounts in Active Directory Users and Computers showed that two additional accounts—Wally and Dilbert—had been placed there by an unknown party as shown on Figure 11.

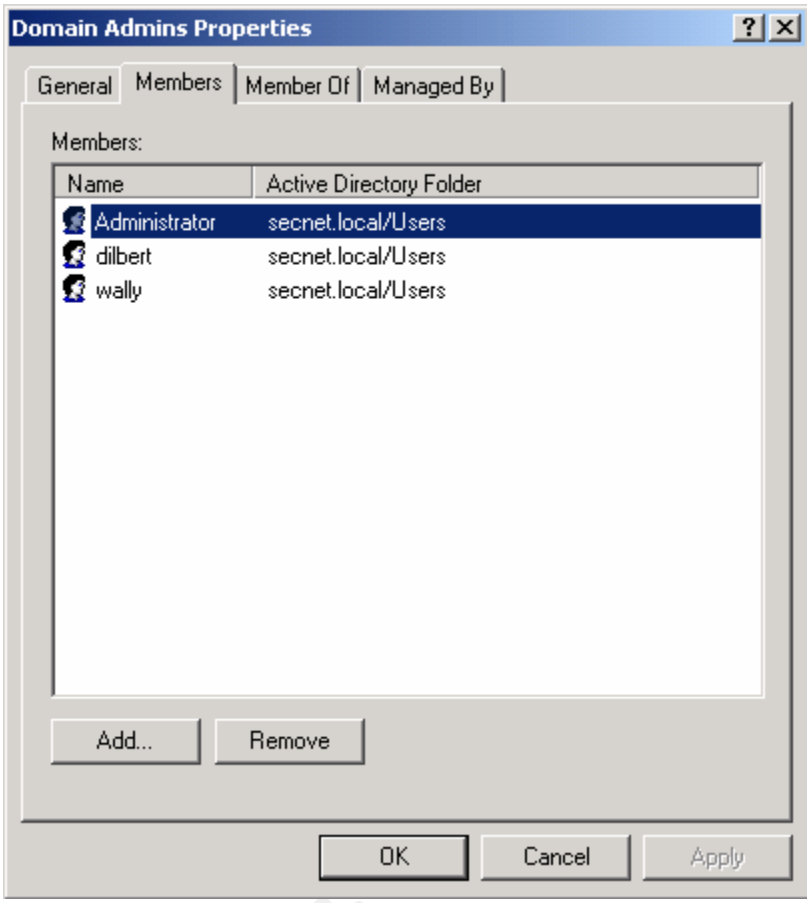

Figure 10: Domain Admins Properties on DC

The CIRT then reviewed the Windows 2000 security logs. The security logs showed repeated failed login attempts against the Administrator Account (Event IDs 529 and 681), followed by a successful login (Event ID 644); all activity had occurred the previous afternoon. After reviewing the lockout policy for the domain, the CIRT determined that the this DC had violated company security policy by not implementing the lockout policy..

Since the security logs of the Windows 2000 DC yielded very little information about the specific source or type of compromise, the CIRT needed another source of information. The CIRT Leader asked to review the Snort IDS logs. Investigation of the logs revealed the following alerts:

```
[**] [1:0:1] LDAP-based password cracker in use. [**]
[Classification: Misc Attack] [Priority: 2]
10/06-14:37:13.356157 192.168.128.35:1060 -> 192.168.128.5:389
TCP TTL:128 TOS:0x0 ID:324 IpLen:20 DgmLen:121 DF
***AP*** Seq: 0x1EBE9A8B  Ack: 0x63CECA38  Win: 0xFAF0  TcpLen: 20


 [**] [1:0:1] LDAP-based password cracker in use. [**]
```

```
[Classification: Misc Attack] [Priority: 2]
10/06-14:37:13.359607 192.168.128.35:1060 -> 192.168.128.5:389
TCP TTL:128 TOS:0x0 ID:325 IpLen:20 DgmLen:113 DF
***AP*** Seq: 0x1EBE9ADC  Ack: 0x63CECA4E  Win: 0xFADA  TcpLen: 20


 [**] [1:0:1] LDAP-based password cracker in use. [**]
[Classification: Misc Attack] [Priority: 2]
10/06-14:37:13.361925 192.168.128.35:1060 -> 192.168.128.5:389
TCP TTL:128 TOS:0x0 ID:326 IpLen:20 DgmLen:114 DF
***AP*** Seq: 0x1EBE9B25  Ack: 0x63CECAEE  Win: 0xFA3A  TcpLen: 20


 [**] [1:0:1] LDAP-based password cracker in use. [**]
[Classification: Misc Attack] [Priority: 2]
10/06-14:37:13.364101 192.168.128.35:1060 -> 192.168.128.5:389
TCP TTL:128 TOS:0x0 ID:327 IpLen:20 DgmLen:116 DF
***AP*** Seq: 0x1EBE9B6F  Ack: 0x63CECB8E  Win: 0xF99A  TcpLen: 20


 [**] [1:0:1] LDAP-based password cracker in use. [**]
[Classification: Misc Attack] [Priority: 2]
10/06-14:37:13.366095 192.168.128.35:1060 -> 192.168.128.5:389
TCP TTL:128 TOS:0x0 ID:328 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x1EBE9BBB  Ack: 0x63CECC2E  Win: 0xF8FA  TcpLen: 20


 [**] [1:0:1] LDAP-based password cracker in use. [**]
[Classification: Misc Attack] [Priority: 2]
10/06-14:37:13.368109 192.168.128.35:1060 -> 192.168.128.5:389
TCP TTL:128 TOS:0x0 ID:329 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x1EBE9C03  Ack: 0x63CECCCE  Win: 0xF85A  TcpLen: 20
```

The alerts continued for approximately 25 pages, but for the purposes of this discussion, they have been truncated in order to avoid redundancy.

The CIRT Leader verified that the destination IP address shown in the alerts matched that of the DC which contained the unknown accounts. Additionally, the CIRT Leader verified that no other IP addresses appeared in the alerts.

The CIRT Leader asked the network administrator in charge of the DC if the password for the Administrator Account on the DC was a strong password (8 characters minimum, upper- and lowercase, numeric and special characters). The administrator admitted that it was not. After being questioned further, the administrator admitted that the password for the Administrator Account was “aardvark.” It appeared that the Administrator Account was easily cracked by a dictionary attack.

The first thing the CIRT checked was the firewall. A review of the firewall rulebase showed that no changes to it had been made over the past three months that would account for this intrusion. There had been changes made to the rulebase to filter outgoing traffic, but no changes had been made that would allow additional incoming traffic into the SecNet network. Based on this information, the CIRT believed that they could effectively eliminate a compromise through the firewall.

A major concern of the CIRT at this point was the fact that the source address appeared to be coming from the internal network. This was disturbing since it meant that the attacker could potentially be a SecNet employee and not an outside attacker, especially since the attack occurred during normal business hours.

The CIRT Leader talked to the network administrator, who then checked the records for the primary Dynamic Host Configuration Protocol (DHCP) server. It was determined that the source address in question was part of a DHCP scope that had recently been set aside for wireless clients.

The CIRT Leader next checked the SecNet wireless infrastructure. Since wireless networking has historically been a double-edged sword for businesses, the CIRT Leader knew that it was quite possible that security issues with the Wireless Access Point (WAP) could be the source of the security breach.

Next, the CIRT Leader accessed the management features of one of the network's WAPs and reviewed the security features of the WAP. These features include Media Access Control (MAC) address filtering and Wired Equivalent Privacy (WEP) encryption. The CIRT Leader discovered that none of these options had been enabled.

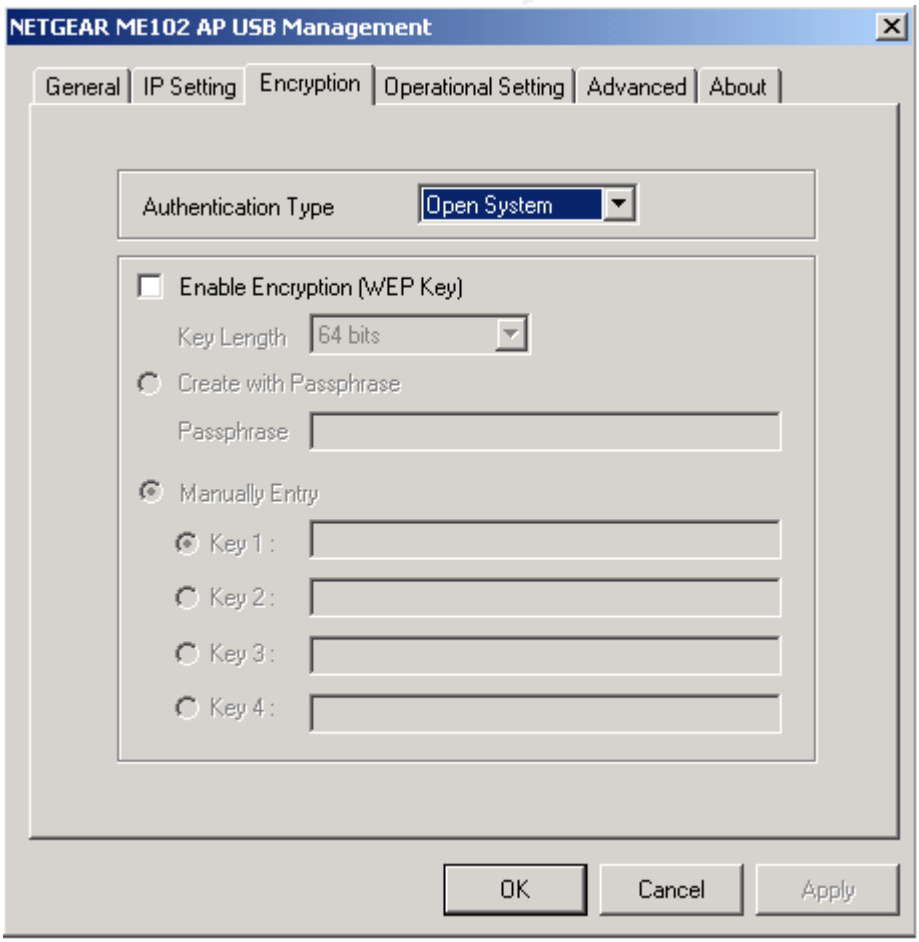

Figure 11: ME102 Access Point with WEP Disabled

Based on this discovery, it seemed highly probable that the CIRT had found the attack vector.

The CIRT then turned their attention to the two new accounts. Further review of the Snort logs from the time period in question yielded the following alerts:

```
[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:53.635042 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35044 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xEA5070D1  Ack: 0x40AB25D4  Win: 0x7BEF  TcpLen: 20

[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:53.653061 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35049 IpLen:20 DgmLen:262 DF
***AP*** Seq: 0xEA5070D1  Ack: 0x40AB25D4  Win: 0x7D78  TcpLen: 20

[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:53.653466 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35050 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0xEA5071AF  Ack: 0x40AB25D4  Win: 0x7D78  TcpLen: 20

[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:54.156018 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35060 IpLen:20 DgmLen:1288 DF
***AP*** Seq: 0xEA507763  Ack: 0x40AB25D4  Win: 0x7D78  TcpLen: 20

[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:54.156180 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35061 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0xEA507C43  Ack: 0x40AB25D4  Win: 0x7D78  TcpLen: 20

[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:56.436193 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35098 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xEA50C1BA  Ack: 0x40AB272E  Win: 0x7D78  TcpLen: 20

[**] [1:0:0] Remote Desktop Protocol. [**]
[Priority: 0]
10/06-15:12:56.436317 192.168.128.35:35406 -> 192.168.128.5:3389 TCP
TTL:63 TOS:0x0 ID:35099 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xEA50C76E  Ack: 0x40AB272E  Win: 0x7D78  TcpLen: 20
```

These alerts continued for approximately 10 pages but have been truncated to avoid redundancy.

After the discovery of this information, the CIRT Leader talked to the Senior Network Administrator who confirmed that the DC did have Terminal Services set up in Remote Administration mode.

At this point, the CIRT Leader fairly certain what occurred: An attacker used SecNet's unencrypted wireless network as the entry point into the network. Once inside, the attacker probably launched a dictionary attack against the Administrator Account (although it was possibly a brute force attack). Once the account was compromised, the attacker logged into the server via Terminal Services and created two additional accounts (Dilbert and Wally) with Domain Admin level rights. The attacker then logged out.

The CIRT Leader reported all of this information to management. The CIRT Leader also informed management that as of this point, only one DC appeared to have been compromised and that the efforts of the CIRT would be to contain the threat and eradicate the problem.

4.3 Containment

Since the attacker managed to compromise a DC, the CIRT Leader had the DC disconnected from the network while the final cause was being determined. All members of the CIRT and management were notified that a compromise had taken place. Notification was also sent to the IT staff that there may be network performance issues as a result of a DC being shut down. Access to the DC was limited to CIRT members only, and all access was logged in a paged-numbered notebook.

The CIRT performed an brief analysis of the hard drive of the DC in order to determine if any additional software such as keystroke loggers, sniffers, back doors, root kits, or other malware had been installed on the DC that might aid the attacker in compromising other computers on the network. In order accommodate this process they booted the server from a forensic CD and ensured that only statically compiled binaries of known good utilities were used throughout the analysis. No evidence of malware was found on the system.

A review of all perimeter and internal logs from the time period in question showed no other suspicious activity of this nature. Additionally, no other bogus accounts were found on other DCs.

Based on this information and after reviewing the logs, the CIRT Leader was satisfied that only one DC was compromised in the incident.

4.4 Eradication

SecNet policy is to rebuild the affected system(s) and return them to normal operations as quickly as possible. There were several lapses in policy that allowed the attacker to compromise this DC:

1. While there is no known CVE or patch to fix this issue, this attack could have been prevented. By enabling a lockout threshold on the DC, any accounts that the attacker tried to brute force would have been locked out after a specified number of failed attempts. To prevent future attacks, the CIRT Leader suggested ensuring that all DCs have account thresholds enabled in the future.

2. The wireless network was a recent addition to the company network and, as such, had not yet been encrypted. While the company firewall blocks 389/tcp and 3268/tcp traffic, the unencrypted wireless network inadvertently allows an attacker a backdoor into the corporate local area network (LAN). To prevent future attacks, the CIRT Leader suggested enabling some form of wireless protection. At an absolute minimum, WEP encryption and MAC filtering should be enabled. Ideally, some form of virtual private network (VPN) tunneling should be employed.

3. Since the attacker could have loaded a wide variety of malware on the DC while logged on as an administrator, the CIRT Leader asked for the DC to be rebuilt using the last full backup before the incident. The CIRT Leader also requested that the IT staff ensured that the DC was current on all its patches.

4. The network administrator was using a default username of “administrator” on the DC. This is not considered a good idea as many attackers as well as worms will try this account name because it is so well known. The CIRT Leader asked for the default username to be changed on the Administrator Account.

5. The administrator was also using a weak password—“aardvark.” This runs counter to the best practice of using upper- and lowercase, numbers, and special characters in all passwords and ensuring they are at least eight characters long and not based on dictionary words. The CIRT Leader insisted that all staff members follow these guidelines in the future.

4.5 Recovery

The DC was rebuilt from a known good full backup and updated with a full set of patches. The DC was brought back up on an isolated network and checked for known backdoors. NETSTAT –an, FPORT, and Process Explorer were run to check for rogue programs that were either running in the background or that were attempting to make a connection outside the SecNet LAN. None was found. Active Directory Users and Computers was also checked to ensure that no extraneous users were present in any of the various administrative groups. None was found.

After notifying management of the steps taken to secure the DC and prevent a similar attack in the future, the IT Staff brought the DC back online. The server was monitored by the CIRT for four days before the incident was closed to ensure that no backdoors or other malware existed on either the server or the restoration media. After four days of seeing no unusual activity, the case was closed.

4.6 Lessons Learned

Immediately after closing the case, the CIRT Leader wrote up an incident response report with the help of the CIRT and the network administrators who were involved with the incident. The report included notes taken by the various parties during the incident and included the following information:

- The date of the incident
- Who reported the incident
- The type of incident
- Physical location of the affected system
- Location on the network of the affected system
- Description of the affected system
  - Operating System
  - Hardware Information
  - Serial Number
  - System Name
  - IP Address/MAC Address
- Cause of the incident
- Actions taken
- Who took the actions
- Who made any backups
- Eradication steps
- Who brought the system back online
- Date and time the system was brought back online
- Monitoring actions after the system was return to an operational state
- When the case was closed

The cause of the attack was described in the report as an LDAP-based password-cracking attack against the Administrator Account via an unencrypted wireless connection. The attacker was able to guess the administrator's password because of a weak password. The attacker then proceeded to connect to the DC via a Terminal Services session and create two Domain Administrator Accounts named Wally and Dilbert.

The report also indicated that the DC had been unable to stop this attack because the account lockout threshold had not been enabled in Group Policy. Additionally, had WEP or MAC address filtering, or both, been activated on the wireless access points, this attack would probably not have been possible.

The report was distributed to all CIRT members, and a follow-up meeting was scheduled. The CIRT Leader stressed to all members that the purpose of the meeting was not to assign blame for the incident but instead was to improve the incident handling process and to improve the security of the company.

During the meeting, the following points were emphasized:

1. Staff members need to feel free to report potential incidents. In this instance, early reporting by the network administrator probably prevented a larger incident later on. This attitude needs to be encouraged.

2. IT staff should be aware of the security ramifications of any new technology that they add to the network infrastructure before they deploy it. Ideally, someone from the CIRT should be consulted.

3. Strong passwords need to be enforced, especially for network administrators.

4. The use of the default “administrator” username should not be allowed.

5. It would be helpful to be able to find notebooks that have page numbers already in them as opposed to the CIRT having to write down the number on the corner of each page. Preprinted page numbers would also look better if an attacker was ever located and taken to court because these documents may later be used in court, and it is important to show that the notes were not tampered with in any way.

6. CIRT members should refrain from sharing details about incidents that are currently under way. Whenever possible, they should use the principle of least privilege for an ongoing incident.

7. Auditing and a lockout threshold need to be enabled on all of the DCs. They can be set to a reasonable level, but they have to be turned on.

The CIRT summarized the meeting in an Executive Summary for management, which the CIRT Leader presented the next day. The Executive Summary made the following recommendations:

1. Ensure that proper auditing and account lockout policies are enabled on all Windows 2000 DCs and are set at a reasonable level. Setting thresholds too low (e.g., 2) will lock out legitimate users while setting thresholds too high (e.g., 50) will put the company at risk for another password-cracking attack.

2. Ensure that all staff, especially IT staff, adhere to the company policy of using secure passwords. The company policy defines a secure password as one that is at least eight characters long and that has upper- and lowercase characters, numbers, and special characters.

3. The default account name “administrator” on Windows 2000 should not be used because it is frequently targeted by attackers and various types of malware. The default account name should be changed on all machines on the SecNet network.

4. The CIRT will need a standardized set of forms and notepads for taking notes during any incident. It is especially important that any notebooks that are used for note taking during an incident have preprinted page numbers because these documents may later be used in court, and it is important to show that the notes were not tampered with in any way.

4.7 Conclusion

As the SecNet incident has shown, failure to abide by the basic rules of security can have far-reaching consequences for a company. An administrator’s failure to maintain a secure password ultimately opened the door that led to the compromise of a DC. Additionally, the company’s wireless network allowed the attacker the initial entryway necessary to start the attack.

When handling any sort of an incident such as this one, it is imperative that attention be paid to all six steps of the incident handling process. It is not enough to simply prepare for an incident, identify it, contain it, and eradicate it. Incident handlers who ignore the recovery and lessons learned phases risks dooming themselves to seeing the same incidents repeatedly. The incident response process must be one of constant evolution and of continual learning in order to benefit the overall security of the organization and the security community as a whole.

5.0 WORKS CITED

Bennett, Gordon. *LDAP: A Next Generation Directory Protocol*. September 2000. <http://www.intranetjournal.com/foundation/ldap.shtml>.

Carnegie Mellon Software Engineering Institute. *CERT Coordination Center*. 17 October 2003. <http://www.cert.org>

FX. *K0LD*. n.d. <http://www.phenoelit.de/kold/docu.html>.

Insecure.org. *Penetration Testing: Re: LDAP Directory*. 1 August 2001. <http://lists.insecure.org/lists/pen-test/2001/Aug/0007.html>.

Mitre Corp. *Common Vulnerabilities and Exposures*. 3 October 2003. <http://www.cve.mitre.org>

Sardanons, Eliel. *Vulnerability Development: Code to Brute Force Win2K Users Passwords via LDAP*. 27 June 2001. <http://lists.insecure.org/lists/vuln-dev/2001/Jun/0215.html>.

SecuriTeam. *LDAP Authentication Brute Forcing*. 15 December 2001. <http://www.securiteam.com/tools/6F00D0U3GK.html>.

SecuriTeam. *Windows 2000 Dictionary Attacker against Active Directory*. 17 March 2003. <http://www.securiteam.com/tools/5HP0E209FG.html>.

The Internet Society. *Password Policy for LDAP Directories.* November 2000. <http://www.globecom.net/ietf/draft/draft-behera-ldap-password-policy-03.html>.

Wahl, M., T. Howes, and S. Kille. *Lightweight Directory Access Protocol (v3).* December 1997. <http://www.ietf.org/rfc/rfc2251.txt>.

Wiseman. *w2kdad_readme*. 19 March 2003. <http://www.geocities.com/real_wiseman/w2kdad_readme.txt>.