Global Information Assurance Certification Paper

INCIDENT ILLUSTRATION: WEB SERVER COMPROMISE

JAMES A. DOLAN

JULY 2001

SANS 2001
BALTIMORE, MD
GCIH PRACTICAL ASSIGNMENT
VERSION 1.5c

Executive Summary

On condition of the participants identifying information in the following incident account has been changed in order to maintain the privacy and confidentiality of the organizations and persons involved.

On Thursday July 22, 1999 representatives from VinSystems, Inc responded to the offices of NSJ, Inc in order to assist the Systems Administrator to contain, analyze and protect against a penetration of the www.nsjjobs.com web site. It was reported that a hacker had apparently discovered NSJ network vulnerability. As a result, the web site that was used to market NSJ services was defaced and the business faced embarrassment and financial loss.

NSJ is a small firm specializing in the recruitment and placement of personnel for the information systems industry. The firm is based near a large metropolitan region and services clients worldwide. The firm relies heavily upon its IT systems for communications, personnel and contact database and Web marketing.

The firm uses a small Ethernet local area network with Internet connectivity to support its business operations. This IP based network is distributed over two floors of a leased commercial 4-story building. Space occupied by NSJ is protected by a commercial intrusion detection system. Access is controlled by a standard commercial lock and key system.

The NSJ network consists of 50 Windows NT workstations, 2 Windows NT 4.0 Servers, 1 Linux Server, 3 HP Laser Jet IV network printers, 3 Netgear 10/100 DS Network hubs, 1 BayStack 450 10/100/1000 Switch and a Cisco 2601 router. The workstations and servers are predominantly versions of the Dell computer, Optiplex and Power Edge models.

The primary NT server (NSJ_APPS) acts as the domain controller and database application server. The second NT server (NSJ_MAIL) is used to run Microsoft Exchange Server, the firms Email system. The Linux server (NSJ_WEB) is the firms Web and FTP server.

A drawing of NJS, Inc. network topology is depicted in Figure 1.

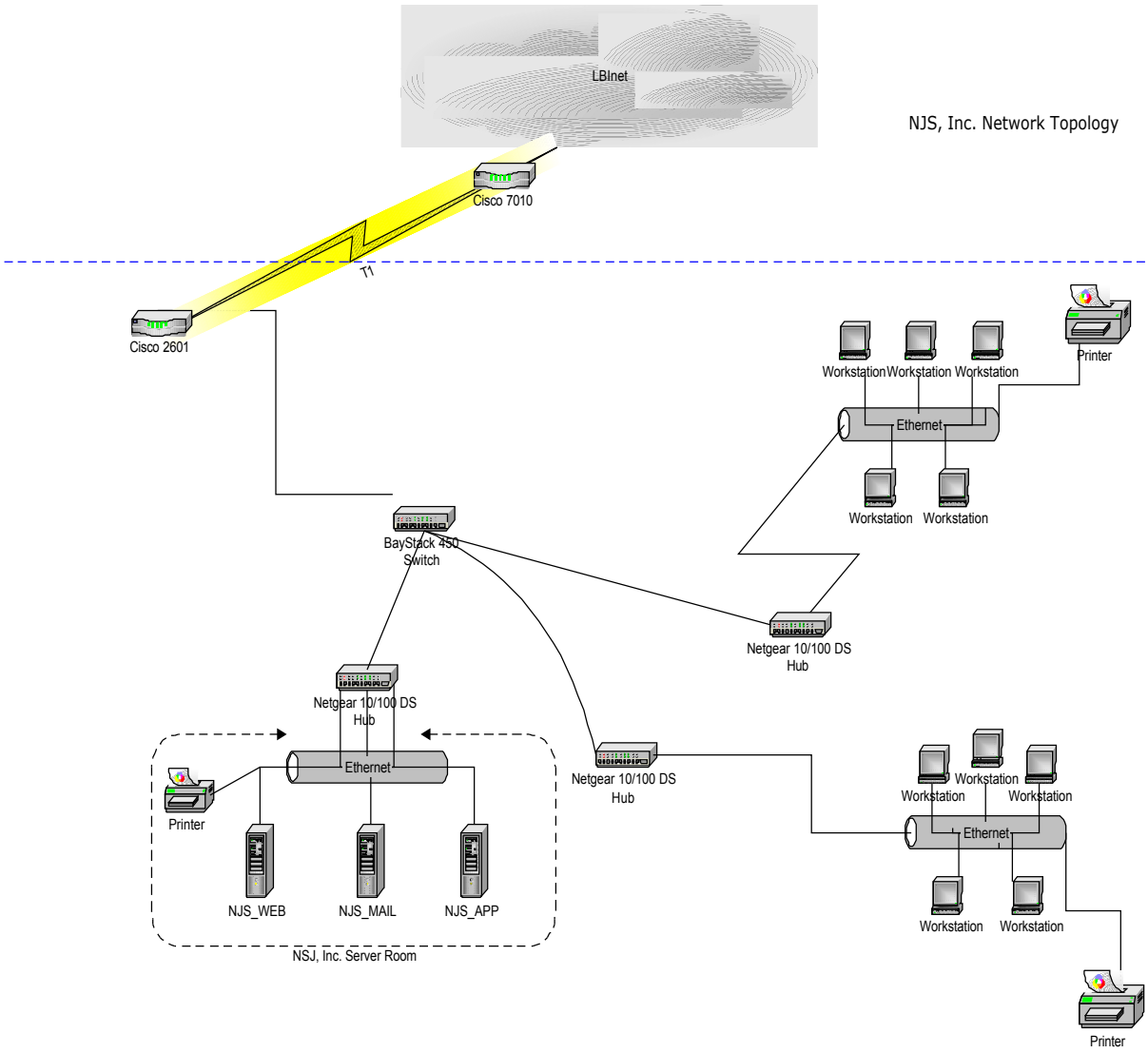

Figure 1 NJS, Inc. Network Topology

LBInet is the ISP that provides NSJ with Internet service via leased line T1 service originating from a Cisco 7010 router.

In order to market the firm's services and to enhance its image, NSJ management contracted with an individual to develop a corporate Web site. In late 1998 the site was developed and hosted at NSJ on a server using Linux 5.2 and Apache HTTPD Web server version 1.0.3.

Preparation

Preparation is perhaps the most critical aspect of incident handling. Unfortunately NSJ personnel gave little or no thought to system security or incident handling. No warning banners existed on systems. Anti virus software implementation was inconsistent and uncontrolled. Corporate IT security policies were poor or non-existent. It was also apparent that communication between the system administrator and firm management was not ideal. There was a general lack of awareness regarding system vulnerabilities. Management was clearly unaware of the importance of IT security even in a small company, as well as the business consequences of a defaced Web site.

It was unfortunate that VinSystems incident response personnel had not had the opportunity to attend any SANS training. (Many VinSystems personnel have since attended SANS training and have returned with glowing accolades.) Fortunately, the VinSystems team was comprised of members with backgrounds and certifications in corporate and government systems administration and IT security. The team leader had recently attended computer intrusion training and is a veteran of intrusion investigations as a Systems Administrator.

The VinSystems team had met routinely for several months prior to this incident, to discuss the network attack methodologies and countermeasures. The team constructed a small IT laboratory for red team exercises and to simulate various attack-defend scenarios.

A jump kit had been assembled by team personnel consisting of notebook PC's (dual boot Linux / NT), digital camera, several 4 port hubs, portable zip drives, portable CDROM drive, blank diskettes and miscellaneous network and telecommmunications supplies and equipment. Utility software including Ethereal, Nmap, SAINT & Nessus was collected, installed on notebooks PC's and available on diskette. "Clean" copies of various operating systems including Linux and Windows NT were also available on CDROM.

The Vinsystems team was supported by of additional staff including management and administrative personnel. VinSystems personnel established and maintained contacts with various government and corporate entities, including local law enforcement.

Identification

On Thursday July 22, 1999 the VinSystems team leader interviewed NSJ management personnel and the NSJ Systems Administrator. NSJ personnel claimed that beginning sometime during the weekend of July 16-18, 1999 an unauthorized entry into the NSJ Web server had occurred during which the main page (index.html) of www.njsjobs.com had been overwritten by an unknown intruder. Unfortunately, a major client who had attempted to view the site had reported this incident to NSJ management.

NSJ management's primary concern was for the Web site to be returned to a secured and normal operation. The VinSystem consultants were further advised that criminal or civil prosecution of the intruder or intruders was not of primary concern.

NSJ management further stated that since initial development and implementation the only person authorized (and capable) of making changes to the Web site was the NSJ Systems Administrator.

The NSJ Systems Administrator advised that on Monday July 19, 1999 he had restored the NSJ_Web files from backup tape and changed all NSJ_Web passwords. However, on Wednesday July 21 the Systems Administrator discovered that the file had again been overwritten.

The NSJ Systems Administrator further advised that a review of NT server log files and interviews with users indicated that the intrusion was isolated to the NSJ_Web server.

Vinsystems personnel independently verified the Web defacement by establishing a dial-up Internet connection and navigating to the NSJ corporate Web site. The current page contained the words "Don't screw with CyberStorm – Howard Stern rulz" and a link to a pornographic site.

Considering all the evidence collected at this point it was concluded that this was if fact an actual incident.

NSJ management provided the VinSystems team with authorization to access all NSJ systems. The team agreed to make every effort to protect the confidentiality of the information contained on these systems, provided that this information had not already been compromised. The NSJ Systems Administrator was instructed to provide all necessary support to the

VinSystems team.

Containment

The team proceeded to the "LAN room" and began the process of containing and analyzing the incident. One team member made initial contact with LBInet, NSJ's Internet service provider. The team leader continued to interview the Systems Administrator. Another team member was tasked with recording events of the incident. A third team member set up a hub between the network and NSJ_Web to which a notebook PC, used for analysis, was attached. Once the team leader completed the interview of the system administrator, he logged onto the NSJ_Web as root from the console and began an analysis of the system.

First it was determined that there were no additional users logged on by using the "w", finger, who & netstat commands.

NSJ_Web was a Dell Power Edge 4500 server with an integrated 8mm tape backup system. A complete 20GB hard drive backup of this system was created to tape using the following command:

```
[root@NSJ_Web / root]# dd if = /dev/hda of= /dev/nrst0
```

Using the dd command provided an accurate block by block image copy of the entire hard drive.

Tapes created during this process were labeled as follows:

Client:
Location:
System:
Partition:
Date:
Time:
Operator:

Tapes were sealed in clear plastic envelopes and placed in an onsite corporate (Class 6) safe for potential forensic and evidentiary use.

After a brief period during which network activity was monitored via the attached analysis notebook PC (on which Ethereal network sniffer software was running), an update meeting with NSJ management was held. Based upon the recommendation of the VinSystems team, and consistent with NSJ

goals, NSJ management decided to disconnect the NSJ_Web server from the network.
As a precaution VinSystems team personnel also reviewed the log files located on the two NSJ NT servers, NSJ_APPS and NSJ_MAIL using the NT event viewer. Log files reviewed were:

Application Log – APPEVENT.EVT
Security Log – SECEVENT.EVT
System log – SYSEVENT.EVT

At this point the VinSystems team was satisfied that the intrusion was isolated to the NSJ_Web server. Since NSJ_Web was disconnected from the network the incident was contained.

Eradication

Eradication began by changing passwords on all accounts on the NSJ_Web Linux server. As a precaution passwords for all accounts on the NSJ domain were also forced.

Log files on the NSJ_Web server were examined and showed signs of tampering. The WTMP file which tracks logins appeared to have gaps or missing entries. The History file that displays most recent commands used; and Message Log that contains system messages both had no entries.

Since hackers are known to use cron to periodically perform malicious processes, all files run by "cron" and "at" were examined. These processes were found to be consistent with normal system operation.

Upon examination of the /etc/password file several accounts with root privileges were identified which were unknown to the NSJ Systems Administrator. These accounts were deleted.

The file used to configure services "/etc/inetd.conf" was inspected for unauthorized changes or additions. Some running services were identified which had potential vulnerabilities. These services are discussed later. Initialization files "init" or "rc" (i.e. /etc/rc.d/rc.local) were inspected for execution of unauthorized programs.

The following Linux commands were used during the forensic analysis of the NSJ_web server.

Common Linux Forensic Commands

sh, csh, bash	Switch shell (to disable command history)
dd	Block level disk image (backup)
w	Reads utmp log data. Shows users currently logged in and what process they are currently running
finger	Reads utmp log data. Shows users currently logged in and from where they are logged in. (Often disabled)
who	Reads utmp log data. Shows users currently logged in and from where they are logged in.
mount	Shows filesystems currently mounted
netstat (-a)	Shows users currently logged in and from where they are logged in, as well as services and network connection information
uname -a	Shows system information (OS etc)
uptime	Shows current system time, how long system has been up, how many users are logged on, and system load.
last	Reads wtmp log data. Shows users logged in and logged out times.
history	Shows command history file
ps (-aux)	Shows processes currently running on the system
lsof	Shows all open files
Ifconfig -a	Shows current configuration of all interfaces

The following file system analysis was included during the forensic examination of the NSJ_web server.

Basic Linux Forensic File Analysis

/etc/passwrd	User and password file. Check for new or modified entries.
/var/log & /var/adm	Various log files. Check for suspicious log file entries
/etc/inetd.conf	Services configuration. Inspect for unauthorized additions or changes
"init" or "rc" files (/etc/rc.d/rc.local)	Initialization or runlevel configuration. Inspect for unauthorized additions or changes
"cron" or "at" files (/etc/crontab)	Periodic process execution. Inspect for unauthorized additions or changes

The VinSystems team suspected that an unauthorized sniffer was installed when the **ifconfig** command was run. The command output showed that interface eth0 was in promiscuous mode.

```
eth0      Link encap:Ethernet  HWaddr 00:00:86:51:2A:A2
          inet addr:10.119.36.234  Bcast:10.119.39.255  Mask:255.255.252.0
          UP BROADCAST RUNNING PROMISC  MTU:1500  Metric:1
          RX packets:10521992 errors:28 dropped:0 overruns:8 frame:28
          TX packets:1468 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:100
          Interrupt:3 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

An example of ifconfig output showing eth0 in promiscuous mode.

Running the **ps –aux** command displayed a suspicious process that was subsequently identified as a sniffer program. This program (sniffit) was capturing network traffic and storing it to a hidden file on the system. The process was stopped and the sniffer program was deleted. The capture file was inspected and revealed data that had been captured over

approximately a 24-hour period. This time period was consistent with the file creation date of the index.html file written by the intruder.

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.4   1292   528 ?        S    Aug12   0:06 init [3]
root         2  0.0  0.0      0     0 ?        SW   Aug12   0:00 [kflushd]
root         3  0.0  0.0      0     0 ?        SW   Aug12   0:00 [kupdate]
root         4  0.0  0.0      0     0 ?        SW   Aug12   0:00 [kpiod]
root         5  0.0  0.0      0     0 ?        SW   Aug12   0:00 [kswapd]
root         6  0.0  0.0      0     0 ?        SW<  Aug12   0:00 [mdrecoveryd]
root       318  0.0  0.4   1352   576 ?        S    Aug12   0:00 syslogd -m 0
root       385  0.0  0.3   1276   420 ?        S    Aug12   0:00 /usr/sbin/apmd -p
nobody     439  0.0  0.4   7588   596 ?        S    Aug12   0:00 identd -e -o
daemon     458  0.0  0.2   1324   356 ?        S    Aug12   0:00 /usr/sbin/atd
root       489  0.0  0.7   2176   928 ?        S    Aug12   0:00 xinetd -reuse -pi
root       504  0.0  0.6   2428   768 ?        S    Aug12   0:00 /usr/sbin/sshd
                                   ……….
apache    1019  0.0  1.4   7160  1836 ?        S    Aug12   0:00 /usr/sbin/httpd -
root      7839  0.1  3.4   7528  4348 ?        S    00:12   0:02 gnome-terminal --
root      7841  0.0  0.4   1328   564 ?        S    00:12   0:00 gnome-pty-helper
root      7842  0.0  1.0   2308  1332 pts/0    S    00:12   0:00 bash
root      7956  0.0  1.0   2308  1344 pts/1    S    00:28   0:00 bash
root      7965  0.0  0.4   1580   560 pts/0    S    00:29   0:00 sniffit -p 23 -A
root      7992  0.0  0.7   2748   916 pts/1    R    00:37   0:00 ps -aux
```

An example of ps -aux output showing sniffer program command line.

Trusted host relationship files .host and host.equiv were reviewed and showed no unauthorized entries.

Nmap, which was installed on the VinSystems analysis notebook PC, is a powerful (and free) port scanner (www.insecure.org). Nmap was used to scan for all ports on NSJ_Web and the 2601 and 7110 Cisco routers for vulnerable or unnecessary services. Open services that were investigated and found to be unnecessary were disabled. The Nmap scan found the following:

RedHat Linux 5.2 NSJ_Web

Port	State	Service
21/tcp	open	ftp
23/tcp	open	telnet
25/tcp	open	smtp
53/tcp	open	dns
80/tcp	open	http
110/tcp	open	pop3
111/tcp.udp	open	portmapper
139/tcp	open	netbios-ssn
143/tcp	open	imap2
515/tcp	open	printer
635/tcp.udp	open	moutnd
6000/tcp	open	X11

Cisco 7010 Router (LBInet)

Port	State	Service
7/tcp	open	echo
9/tcp	open	discard
13/tcp	open	daytime
19/tcp	open	chargen
79/tcp	open	finger
520/tcp	open	route
6001/tcp	open	X11

Cisco 2601 Router

Port	State	Service
7/tcp	open	echo
9/tcp	open	discard
19/tcp	open	chargen
79/tcp	open	finger
520/tcp	open	route
6001/tcp	open	X11

Contact was again made with representatives of LBInet to discuss router configuration. A review of router configurations also showed that many other areas needed to be addressed. The ACL's (Access Control Lists) showed that more traffic was allowed thru the router than was necessary. For example, the ACL that controlled traffic into the NSJ_Web allowed ALL traffic.

The VinSystems team made some initial changes to the router configuration and ACL's in order to tighten security. It was recommended that a complete evaluation of the router configurations be conducted with onsite assistance from the ISP, LBInet.

Nessus is a powerful (and free!) network security scanner that will remotely audit a network and identify potential vulnerabilities. A Nessus vulnerability scan of the NSJ network was executed using the VinSystems analysis Notebook PC. The scan identified a NSJ_Web server vulnerability within the example CGI code that was installed with the Apache Web server. Know as the cgi.phf script vulnerability (CA-1996-06 & CVE-1999-0067), this script contains a library function escape_shell_cmd() which is vulnerable to attack. An attacker can use this vulnerability shell commands on the target system.

The Nessus scan of the NSJ_Web server reported the following:

> The "phf" CGI is installed. This CGI has
> a well known security flaw that lets an attacker execute arbitrary
> commands with the privileges of the http daemon (usually root or nobody).
>
> Solution : remove it from /cgi-bin.
>
> Risk factor : Serious

At the time of this incident this vulnerability was well know and there were a number of scripts available to identify and exploit this vulnerability. The VinSystems team postulated that the intruder gained access using this vulnerability.

The Carnegie Mellon Software Engineering Institute, in it's CERT advisory CA-1996-06, provides the following assessment of this vulnerability:

> 1. Description
>
> A security vulnerability has been reported in example CGI code, as provided with the NCSA httpd 1.5a-export and APACHE httpd 1.0.3 (and possibly previous distributions of both servers). The example code contains a library function escape_shell_cmd() (in cgi-src/util.c). This function, which attempts to prevent exploitation of shell-based library calls, such as system() and popen(), contains a vulnerability.
> Any program which relies on escape_shell_cmd() to prevent exploitation of shell-based library calls may be vulnerable to attack.
>
> In particular, this includes the "phf" program which is also distributed with the example code. Some sites may have installed phf by default, even though it is not required to run httpd successfully.
>
> Any vulnerable program which is installed as a CGI application may allow unauthorised activity on the HTTP server.
>
> Please note that this vulnerability is not in httpd itself, but in CGI programs which rely on the supplied escape_shell_cmd() function. Any HTTP server (not limited to NCSA or Apache) which has installed CGI programs which rely on escape_shell_cmd() may be vulnerable to attack.
>
> Sites which have the source code to their CGI applications available can determine whether their applications may be vulnerable by examining the source for usage of the escape_shell_cmd() function which is defined in cgi-src/util.c.
>
> Sites which do not have the source code for their CGI applications should contact the distributors of the applications for more information.
>
> It is important to note that attacks similar to this may succeed against any CGI

program which has not been written with due consideration for security. Sites using HTTP servers, and in particular CGI applications, are encouraged to develop an understanding of the security issues involved. References in Section 4 provide some initial pointers in this area.

2. Impact

A remote user may retrieve any world readable files, execute arbitrary commands and create files on the server with the privileges of the httpd process which answers HTTP requests. This may be used to compromise the http server and under certain configurations gain privileged access.

3. Workarounds

The use of certain C library calls (including system() and popen()) in security critical code (such as CGI programs) has been a notorious source of security vulnerabilities. Good security coding practice usually dictates that easily exploitable system or library calls should not be used. While secure CGI coding techniques are beyond the scope of this advisory many useful guidelines are available.
Sites planning to install or write their own CGI programs are encouraged to read the references in Section 4 first.

3.1. Remove CGI programs

Any CGI program which uses the escape_shell_cmd() function and is not required should be disabled. This may be accomplished by removing execute permissions from the program or removing the program itself.
In particular, sites which have installed the "phf" program and do not require it should disable it. The "phf" program is not required to run httpd successfully. Sites requiring "phf" functionality should apply one of the workarounds given in sections 3.2 and 3.3.

3.2. Rewrite CGI programs

The intent of the escape_shell_cmd() function is to prevent passing shell meta-characters to susceptible library calls. A more secure approach is to avoid the use of these library calls entirely.
AUSCERT recommends that sites which are currently using CGI programs which use shell-based library calls (such as system() and popen()) consider rewriting these programs to remove direct calls to easily compromised library functions.

Sites should note that this is only one aspect of secure programming practice. More details on this approach and other guidelines for secure CGI programming may be found in the references in Section 4.

3.3. Recompile CGI programs with patched util.c

For sites that still wish to use programs using the escape_shell_cmd() function, a patched version of cgi-src/util.c has been made available by NCSA which addresses this particular vulnerability. The patched version of util.c is available as part of the http1.5.1b3-export distribution. This is available from:
http://hoohoo.ncsa.uiuc.edu/beta-1.5

> Please note that this is a beta-release of the NCSA httpd and is not a stable version of the httpd. The patched version of cgi-src/util.c may be used independently.
>
> CGI programs which are required and use the escape_shell_cmd() should be recompiled with the new version of cgi-src/util.c and then reinstalled.
>
> Apache have reported that they intend to fix this vulnerability in a future release. Until then the patched version of util.c as supplied in the http1.5.1b3-export release should be compatible.
>
> 4. Additional measures
>
> Sites should consider taking this opportunity to examine their httpd configuration. In particular, all CGI programs that are not required should be removed, and all those remaining should be examined for possible security vulnerabilities.
> It is also important to ensure that all child processes of httpd are running as a non-privileged user. This is often a configurable option. See the documentation for your httpd distribution for more details.

The VinSystems team, assisted by the NSJ systems administrator, deleted the "phf" script and other cgi scripts which were identified as unnecessary.

At this point the team had worked for two days (18 hour shifts) to contain and eradicate the problem.

Recovery

On July 24,1999 a complete backup of the NSJ_web was done. In addition several Linux OS upgrades and security related patches were installed by the VinSystems team.

NSJ management was update on the progress and asked to review the corporate web site prior to making a decision to bring the system back online. The NSJ_Web server was brought back online locally and several NSJ managers and employees participated in the review process. Once the process was complete NSJ management approved the reconnection of the system to the Internet.

The VinSystems team closely monitored the system for several days to ensure that the intrusion did not reoccur. In the process they reviewed logs, monitored connections and worked to educate the NSJ Systems Administrator.

On July 27, 1999 the VinSystems team conducted a final briefing for NSJ management. The briefing included the following recommendations:

- Provide IT Security training for the Systems Administrator
- Consider hiring a second Systems Administrator
- Develop and maintain an IT security policy
- Enforce frequent password changes
- Monitor Internet Security sites for intelligence (i.e.www.sans.org)
- Apply security software patches quickly
- Update router configurations to tighten access
- Consider using an Intrusion Detection System

At the conclusion of the meeting the VinSystems team recovered remaining equipment (notebook PC and hub), and departed the NSJ facility.

Follow-up / Lessons Learned

On July 28, 1999 the VinSystems team met to discuss the incident. During the meeting an outline for a report on the incident was agreed to. Once the draft report was generated all team members would review it prior to finalization.

It was agreed that more training was necessary to prepare for future incidents. It was found that frequent inter-team communication was essential to success. It was decided that a team meeting several times each day would have enhanced the process.

Despite the fact that the intruder had not been identified, the VinSystems team was generally pleased with its incident response capability as well as the fortunate outcome of this incident.

References

Scambray, Joel, et al, "Hacking Exposed Second Addition". Osborne / McGraw-Hill, 2001

Casey, Eoghan. "Digital Evidence and Computer Crime". Academy Press, 2000

Malisow, Ben. "Moment's Notice: The Immediate Steps of Incident Handling". Friday, July 7, 2000
URL: http://www.securityfocus.com/

Wright, Timothy. "An Introduction to the Field Guide for Investigating Computer Crime". Monday, April 17, 2000
URL: http://www.securityfocus.com/

Spitzner, Lance. "Know Your Enemy: A Forensic Analysis". Tuesday, May 23, 2000
URL: http://project.honeynet.org/