Global Information Assurance Certification Paper

SANS IHHE PRACTICAL EXAMINATION

By Robert T. Lee

Executive Summary

On 25 March 2000 the local Domain Name Server running RedHat Linux 6.0 was compromised using the BIND NXT vulnerability. The following is a synopsis of the events from 25 March 2000 to 3 April 2000 that the Incident Response Team followed in accordance with checklists and law enforcement.

Multiple connections from multiple sites to VICTIM box using primarily using FTP and TELNET. Added user *phiber* to VICTIM system. Files downloaded to server and run (*ss*, *nn-linux* (with configuration files), *login*, and *vanish*) SUBJECT switched his trojanized login program to another trojanized version of it. The first version granted access if the virtual terminal was set to *911* and the second grants access to the user *rewt* and the password *w00p!.* SUBJECT also used a log wiper in addition to completely removing the logfile directory from the system. SUBJECT ran a program (*ss*), which attempts to take over channels on IRC servers. SUBJECT also ran another program that keeps channel operations on remote IRC servers (*nn-linux*).

SUBJECT's home computer was possibly identified due to analysis provided. Law enforcement is serving a search warrant. Results of which are unknown.

Law Enforcement has evidence and is opening a case on the SUBJECT.

System was restored, patched, secured, and additional security implementations were added to the VICTIM system and the network it resided on.

Preparation Phase

The local technical and incident response team consists of a small on-call unit ready to deploy to the site that was compromised. Each member of the deployment team is well trained. Combining the experience of computer forensic processing, monitoring, intrusion detection, and incident analysis. In addition training on procedures to take and step-by-step guides were accomplished. Web based and on paper step by step guides on what to do is provided to each team member as well as training on how to use them.

The team structure is typically centered on the deployed team with a central operator serving as the buffer between management and other operational entities leaving the majority of the busy work away from the deployed team.

Communication between our central and the remote location is accomplished using secure, encrypted, communication over email and the network. In addition to using the plain old telephone system.

The policy on monitoring is set up using a very strict process. These process guidelines are set by the organization lawyers following US Federal Code and Title 18 laws. The system administrators are allowed to monitor due from

FROM TITLE 18 CRIMES AND CRIMINAL PROCEDURES in CHAPTER 119. WIRE AND ELECTRONIC COMMUNICATIONS INTERCEPTION AND INTERCEPTION OF ORAL COMMUNICATIONS.

2) (a) (i) It shall not be unlawful under this chapter [18 USCS §§ 2510 et seq.] for an operator of a switchboard, or an officer, employee, or agent of a provider of wire or electronic communication service, whose facilities are used in the transmission of a wire or electronic communication, to intercept, disclose, or use that communication in the normal course of his employment while engaged in any activity which is a necessary incident to the rendition of his service or to the protection of the rights or property of the provider of that service, except that a provider of wire communication service to the public shall not utilize service observing or random monitoring except for mechanical or service quality control checks.

Banners are installed on all systems. This banner is legally tested through passing it through the lawyer's office and

The process of monitoring typically follows the following standards. This standard states the monitoring can occur when a warning banner is present and can be seen upon login or access to the machine. This means, of course, that it is also limited to specific ports that allow access to the machine as well. If the port cannot be bannered the monitoring cannot occur without further action from legal. Typically, bannering is accomplished on each system via TCP Wrappers.

Each team member is equipped with an email pager for instant communication. The deployed team also carries a cell phone that enables

Incident Response Toolkits

The Jump Bag that the team carries consists of a laptop computer with dual-operating systems. A system to process evidence which includes a 8mm tape drive and a CD-Rom burner, an LS-120 Floppy Drive, and three removable hard drive trays to accommodate new media. The hardware was tested and required to easily make copies of the evidence on site. In addition, the system can be used to monitor the compromised system to watch for further compromise.

In addition to the computers, each kit includes a medusa cable (transmit wire cut Ethernet cable), 10 and 100 port hubs, patch cables, extra-unused drives, extra tapes, CD-Rs, and tools. A CD-Rom containing all of the forensic binaries for the operating system and forensic processing tools is also mandatory. These tools provide easy means to grab system backups, status, and log files without damaging any evidence on the system or leave any tracks on the system itself.

On the CDROM it contains fresh binaries for any operating system in question. This is used to ensure that fresh copies of each tool is used and without fear that trojan is in existence. On each cdrom contains the following statically linked tools. *Arp, cp, dd, des, df, diff, dmesg, du, file, find, finger, ifconfig, last, lsmod, lsof, md5, modinfo, mv, nc, netstat, ps, pstree, rpcinfo, showmount, strings, top, uname, uptime, w, who gdb, strace, ltrace, fdisk, dig,* and a bunch of compression tools namely *gzip, compress, uncompress, and gunzip*.

Also on the CDROM is The Corner's Toolkit, an evidence collection tool written by Dan Farmer and Venema Wieste. It was recompiled specifically to use the static binaries loaded on the cdrom. The tools used were mactime, graverobber, and lazarus.

Identification and Containment

System Administrators at local site Alpha were called on 2 April by security personnel at a remote site Bravo regarding possible intrusion of system at local facility. Apparently, this system had been used to access the remote site Bravo.

The remote site Bravo was equipped with an intrusion detection system that detected a remote login via TELNET over port 23. The login and password matched a known system administrator from site Alpha. Site Bravos Intrusion Detection System alarmed because the user from site Alpha was installing a rootkit, an Internet Relay Chat client, and several sniffers on the network at site Bravo. This was not normal activity from site Alpha, thus site Alpha was also probably compromised.

At this point, the IRT discovered that the system was a primary domain name server for site Alpha running RedHat Linux 6.0 Kernel 2.2.12. The computer was a Pentium III 600 MHZ Dell Computer with an Exabyte Tape Drive and a 7 Gigabyte SCSI hard drive. This particular system was exploited using the BIND NXT Exploit (CVE-1999-0833). The IRT had already seen five other systems in the past month exploited using the bug.

Site Alpha was contacted via the security personnel and at the same time the local Incident Response Team was also contacted. The IRT contacted the system administrators of the system and had them follow an incident checklist. The checklist had them disconnect the computer from the network, run a full backup, and to not have anyone touch or log on to the computer until further

instructions were given. The date and time of each event was recorded and logged by the system administrator. When the backup was done, they listed the date, the time, the way it was backed up (using DD using a block size of 4096), and the executor of the backup. The tape was then taken, marked on the tape itself, and placed within a safe that only the system administrator had access to. Two copies of the tape were also made. One tape would be sealed in storage and another to be analyzed in a forensics environment.

The backups were made by the system administrator using the command ***dd if=/dev/sda of=/dev/st0*** which made a backup of the entire physical disk including swap space and partition information to the tape drive located in the system itself.

At this point, the management of site Alpha was contacted and all members were briefed up to speed on the status of the incident.

Site Bravo's security personnel rebuilt the system before any evidence could be taken from it. No backups or network information was gathered. Site Bravo's personnel were given a brief overview of why it helps to gather evidence to help catch the SUBJECTs from further compromise.

The IRT assigned one person to be the main contact and to travel to the compromised location at site Alpha. When he arrived at the location he interviewed the system administrator about the system the system administrator opened up the safe handed over copies of the evidence to the IRT team member. The IRT member signed for and handled the evidence following chain of custody procedures.

The IRT team member also noted that there was a firewall and an Intrusion Detection System on the network logging all connections over Telnet and FTP to and from the Demilitarized Zone on the network. Logs of these were written to CD-ROM, marked as evidence and stored as well. Two copies of each of the logs were made.

At this point the IRT Team member still did not know what had happened to the system. Using a duplicate computer the IRT Team member restored a copy of the file system to another computer to pull the system logs from it. In addition, the Team member used the compromised system to look at the contents of the /proc directory, all the open network connections, and what was currently running on the system. The Team member did this by going through the "vi" editor and running shell commands through the editor so that output could easily be redirected to a floppy disk without writing or being logged on the system itself. The team member did this to keep a low as a profile as possible so it wouldn't damage any evidence or alter any of the settings as to alert the SUBJECT that he had been found. The reason you have to grab these on the system itself is that these only reside in the memory of the machine. If the machine was

powered off any running binaries could not be pulled for analysis thus destroying evidence.

After all the evidence had been collected, the file system backups, the log files, the IDS and the Firewall logs, and the evidence from the memory of the machine itself, the following synopsis of events was determined. This is written so it could be understood in laymen's term by anyone who is not familiar with computer crimes, specifically any law enforcement, management, or legal/judicial personnel.

All evidence and the synopsis below were turned over to law enforcement officials using proper chain of custody. Evidence tags were used to enforce the chain of custody which showed that we made one Exabyte 8mm tape backup of the VICTIM system signed and dated by the system administrator and his assistant as a witness and one CD-ROM of system logs from the firewall and the local intrusion detection system signed and dated by the system administrator.

The following was arranged by first looking at logs from the firewall and the local intrusion detection system, which logs all incoming telnets and ftp access to the demilitarized zone.

On the system itself, we used the commands ps and netstat to determine where the SUBJECTs tools were running and from the logs gathered from the firewall we already knew which directory they resided in. Analyzing the running binaries using strace, lsof, and gdb and the network connections we were able to make determinations what and how they were used. Typically analysis is handled in the manner described in appendix 1 (please reference Appendix 1). Complete details of that analysis are still being worked by law enforcement so case details could not be released as to their content.

The following has been scrubbed of actual relation to names, places, locations, or networks. Any similarity to another event is purely coincidental.

SUBJECTS FINGERPRINT:

Uses handle: h@x0r, h@x

Email address: hax0r@rage.net

Possible Base Computer: 24.223.11.109:raid_me.org also resolves to cv2910-212-b.someplace1.ju.homers.com

Known username(password combinations): r00t(w00p!), t(31337ness), t(jasmine), h@x0r(riddler)

Methods: Internet Relay Chat, Telnet, FTP

Other known systems used by SUBJECT: 211.123.210.18:network.blah.statesys, 166.72.82.181:ppp166-72-82-181.ca.us.prserv.net, 209.80.195.177: linuxstudentwebs.springfield.cc.co.us, 153.121.12.230: OSIRIS.CIS.UPICO.EDU, 216.30.97.91: got.r00t.com, 38.27.198.22: ip22.charlotte.nc.pub-ip.psionic.net, 216.145.221.12:packet-grabber.net

KEYWORDS USED: r00t, h@x0r,

Hacker Skill Level: Low

PERTINENT CONNECTIONS:

SYNOPSIS:

On Saturday Mar 25 02:14 2000 EST, SUBJECT opened up a FTP (TCP port 21) connection to (**24.223.11.109**:*raid_me.org* also resolves to *cv2910-212-b.someplace1.ju.homers.com*) (RESPONSE TEAM note: FTP is the Internet standard for file transfer. The file transfer provided by FTP copies a complete file from one system to another system.) SUBJECT used the username *h@x0r* and the password *riddler.*

SUBJECT downloaded the following files to the VICTIM machine:

login: (a trojanized login program). (RESPONSE TEAM note: a trojan program is a program that acts and feels like another program with "undocumented" features that allow the SUBJECT to generally keep access on a system. These are typically associated with rootkits.)

SUBJECT replaced the file */usr/bin/login* program with the login program he downloaded to the system.

SUBJECT ended the connection on **Mar 25 at 02:15 EST**

On Thursday Mar 30 22:51 2000 EST, the SUBJECT (**211.123.210.18**:*network.blah.statesys*) connected to the VICTIM machine (**193.232.87.15**:*dnsone.sc.doh.gov*) using TELNET (TCP port 23) which grants *super user* access to the VICTIM machine without having to input a password if the Virtual Terminal is set to *911*. (RESPONSE TEAM note: Telnet (TCP port 23) is a remote access protocol that allows one to have a “virtual terminal” connected to any host on the Internet possessing a Telnet server. Once a Telnet session is established and a valid user name and password are entered, keystrokes entered at the user's keyboard are executed on the remote host, and output from the remote host is displayed on the user's screen.) (RESPONSE TEAM note: A root-level account is an account with super-user privileges that have unrestricted read, write, and execute access for the entire computer.) The consent to

monitoring banner was seen and sent to the SUBJECT's computer upon connection to the VICTIM.

SUBJECT switched to a hidden directory that was created */root/".. "* directory and executes the command *ss*. (RESPONSE TEAM note: There is a character space after the double dots). After executed the program writes to the screen *werd, forking into pid 5078* . (RESPONSE TEAM note: Some analysis on the file *ss* shows that it is definitely related to Internet Relay Chat (IRC) in function. It is not entirely clear what the tool does, but based on some crude analysis it looks as though it is a IRC client to attempt to take over remote IRC servers) SUBJECT then issued the command *killall -9 ss* which killed the process running at process number *5078*.

SUBJECT looked at the */etc/passwd* file which showed that the system incorporated shadowed passwords. (RESPONSE TEAM note: The */etc/passwd* file is where all the system passwords are stored. Shadowed password files take the encrypted hash out of the world readable /etc/password file and places it in the restricted file */etc/shadow*)

The SUBJECT issued the command *su http* (switch user to http) to change his effective user identification to the user *http*. As user *http*, he made a new directory in the /*home/http/* directory named */home/http/".. "* he then switched to that directory.

As user *http*, the SUBJECT opened up a FTP (TCP port 23) connection on **Thursday Mar 30 22:53 EST 2000** to (**166.72.82.181**:*ppp166-72-82-181.ca.us.prserv.net*) (RESPONSE TEAM note: FTP is the Internet standard for file transfer. The file transfer provided by FTP copies a complete file from one system to another system.) SUBJECT used the username *t* and the password *jasmine*. Within the FTP session he switched to the exploits directory. He transferred the file *ss* to the VICTIM system. (RESPONSE TEAM note: The file *ss* is an IRC client program to take over remote IRC server channels.) The SUBJECT then executed the file on the system that wrote to the screen *werd, forking into pid 5091*. (RESPONSE TEAM note: IRC (Internet Relay Chat) provides a way of communicating in real time with people from all over the world. It consists of various separate networks (or "nets") of IRC servers, machines that allow users to connect to IRC. The largest nets are EFNET (the original IRC net, often having more than 32,000 people at once), UNDERNET, DALnet, and Newnet. Generally, the user runs a program (called a "client") to connect to a server on one of theIRC nets. The server relays information to and from other servers on the same net.)

On Thursday Mar *30 2356 EST 2000*, the SUBJECT (**209.80.195.177:** *linuxstudentwebs.springfield.cc.co.us)* connected to the VICTIM machine (**193.232.87.15**: *dnsone.sc.doh.gov*) using TELNET (TCP port 23) SUBJECT attempted to access the machine using the user *t* and the password *31337ness*.

SUBJECT was not successful and stopped connecting at **Thursday Mar 30 23:56:56 2000 EST.**

On Friday Mar 31 0036 EST 2000, the SUBJECT (**153.121.12.230**: *OSIRIS.CIS.UPICO.EDU)* connected to the VICTIM machine (**193.232.87.15**: *dnsone.sc.doh.gov*) using TELNET (TCP port 23) which grants super user access to the VICTIM machine without having to input a password if the Virtual Terminal is set to 911. (RESPONSE TEAM note: Telnet (TCP port 23) is a remote access protocol that allows one to have a "virtual terminal" connected to any host on the Internet possessing a Telnet server. Once a Telnet session is established and a valid user name and password are entered, keystrokes entered at the user's keyboard are executed on the remote host, and output from the remote host is displayed on the user's screen.) (RESPONSE TEAM note: A root-level account is an account with super-user privileges that have unrestricted read, write, and execute access for the entire computer.) The consent to monitoring banner was seen and sent to the SUBJECT's computer upon connection.

The SUBJECT reset his virtual terminal to *vt100.* (RESPONSE TEAM note: *vt100* is the default terminal used on most terminal emulation programs). SUBJECT then opened up an IRC client to ***irc.concentric.net*** using the nickname *classic* and joining the channel *#metal*. (RESPONSE TEAM note: Internet Relay Chat is a program that allows different user to communicate in "channels" across the Internet) The SUBJECT joined the channel and mainly observed the session already going on. He only typed once while on the server.

(RESPONSE TEAM note: IRC (Internet Relay Chat) provides a way of communicating in real time with people from all over the world. It consists of various separate networks (or "nets") of IRC servers, machines that allow users to connect to IRC. The largest nets are EFNET (the original IRC net, often having more than 32,000 people at once), UNDERNET, DALnet, and Newnet. Generally, the user runs a program (called a "client") to connect to a server on one of theIRC nets. The server relays information to and from other servers on the same net. Once connected to an IRC server on an IRC network, you will usually join one or more "channels" and converse with others there. On EFnet there often are more than 12,000 channels each devoted to a different topic. Conversations may be public (where everyone in a channel can see what you type) or private (messages between only two people, who may or may not be on the same channel).)

<mr.E> i dont do that no more

<classic> -phreaker's

<mr.E> i got some shitt newnick sthi

<mr.E> tho

<mr.E> heh

* mr.E hacks it up

<h@x> i scare script kiddies with that

<h@x> frecks

<mr.E> dude

<h@x> TERM=vt911

<h@x> telnet dnsone.sc.doh.gov

<mr.E> i stole like

<h@x> your root

<h@x> =]

<mr.E> so many edus

<mr.E> from phreaker

<h@x> ya

<mr.E> without his passwd

<mr.E> but

<mr.E> when i seen it

<h@x> so did i

<h@x> i knew his pw

<mr.E> it was the same encryption

<mr.E> on all boxes

<mr.E> haha

<mr.E> so

<h@x> cause he gave me the pw to one ns to use

<h@x> i just started jacking whatever the bnc'd on

<h@x> :T

<mr.E> ya

<h@x> h@x

<mr.E> what

<h@x> you want that shell?

<mr.E> no

<mr.E> its lame

<h@x> ima put newnick on it

<h@x> :T

<mr.E> dont

<mr.E> its ugly

<mr.E> as fuck

<h@x> hehe

<mr.E> it is dude

<h@x> ya but it wont get packeted my most kiddies

<mr.E> ya right

SUBJECT then quit the IRC server.

SUBJECT opened up a FTP (TCP port 21) connection on **Friday Mar 31 0041 EST 2000 to** (**24.223.11.109**:*raid_me.org* also resolves to *cv2910-212-b.someplace1.ju.homers.com*) (RESPONSE TEAM note: FTP is the Internet standard for file transfer. The file transfer provided by FTP copies a complete file from one system to another system.) SUBJECT used the username *h@x0r* and the password *riddler.*

SUBJECT downloaded the following files to the VICTIM machine:

login: (a trojanized login program). (RESPONSE TEAM note: a trojan program is a program that acts and feels like another program with "undocumented" features that allow the SUBJECT to generally keep access on a system. These are typically associated with rootkits.)

nn-linux (newnick.c v8.4b by Volatile http://www.megatokyo.com/~vol/newnick.html)

nick.cfg (IRC BOT configuration program for newnick.c)

efnet.serv (LIST of IRC servers to be used with newnick.c) {irc.idle.net,irc.prison.net,irc.mindspring.com,irc.core.com,irc.mcs.net,

irc.concentric.net,irc.emory.edu,irc.nethead.com,irc-e.frontiernet.net,irc-w.frontiernet.net, irc-roc.frontiernet.net, irc.colorado.edu, irc.lightning.net,irc2.homers.com}

vanish (a system log wiper)

(RESPONSE TEAM Note: I did an extensive research on who raid_me.org belongs to. The site was registered through www.register.com. The following is what www.register.com had on the domain name.)

Organization: Xcel Security Networks N.W. Xcel St Xcelville, CT 31337

US Registrar..: Register.com (http://www.register.com)

Domain Name: raid_me.org <http://www.raid_me.org>

Created on.....: Sat, Nov 20, 1999

Expires on.....: Mon, Nov 19, 2001

Record last updated on..: Sat, Nov 20, 1999 A

Administrative Contact: Security, Xcel hax0r@rage.net

*<**mailto:hax0r@rage.net>** 1-800-3133700*

Technical Contact, Zone Contact: Internic, Registrar

internic-free@register.com <mailto:internic-free@register.com> 212-594-9880

(RESPONSE TEAM Note: The registration is fake. The 31337 "ELEET" corresponds to a well-known hacker port used for back-orifice and is used for a

zip code and the contact number. The email address is probably correct since he has to be billed in some manner. .)

SUBJECT copied the downloaded *login* program into the */bin* directory effectively trojanizing the *login* program (RESPONSE TEAM note: the new trojan login program now grants super-user access to anyone who gives the username *r00t* with the password *w00p!).* SUBJECT copied *nick.cfg*, *efnet.serv*, and *nn-linux* to the */tmp* directory. The SUBJECT changes to the */tmp* directory and changes the ownership of all the files to the user *h@x0r* and is not successful. SUBJECT adds the user *h@x0r* to the system tries the change ownership command again and is successful. The SUBJECT switches user to the username *h@x0r.*

The SUBJECT opens up a *pico* editor to edit the file *nick.cfg*. (RESPONSE TEAM note: *PICO* is a Linux editor similar to *vi*.) The file nick.cfg is seen below in the technical section extracted with the tags stripped off from the editor. SUBJECT only changed two lines. He changed line 2 *p-r00t* to *p-gov* and line 8 from *r00t* to *gov*.

Lookups on the IP addresses in configuartion

SUBJECT then ran the file *nn-linux*. (RESPONSE TEAM note: The is an IRC program "bot" that keeps channel operator at all times in addition to being a backdoor into the system it was installed on. A BOT is a program that performs tasks so the suer does not have to be at the console the entire time.)

SUBJECT used the file *vanish* to erase his tracks on the system and was successful.

./vanish root OSIRIS.CIS.UPICO.EDU 153.121.12.230

./vanish h@x0r raid_me.org 24.223.11.109

./vanish h@x0r cv2939-303-a .julberry2.de.homers.com 24.223.11.109

SUBJECT closed the session at **Friday Mar 31 0044 EST 2000.**

On Friday Mar 31 0048 EST 2000 , the SUBJECT (**153.121.12.230**: *OSIRIS.CIS.UPICO.EDU*) connected to the VICTIM machine (**193.232.87.15**: *dnsone.sc.doh.gov*) using TELNET (TCP port 23) which grants super user access to the VICTIM machine using a backdoor in the login program. The SUBJECT used the user *r00t* and the password *w00p!.* (RESPONSE TEAM note: Telnet (TCP port 23) is a remote access protocol that allows one to have a “virtual terminal” connected to any host on the Internet possessing a Telnet server. Once a Telnet session is established and a valid user name and password are entered, keystrokes entered at the user's keyboard are executed on the remote host, and output from the remote host is displayed on the user's

screen.) (RESPONSE TEAM note: A root-level account is an account with super-user privileges that have unrestricted read, write, and execute access for the entire computer.) The consent to monitoring banner was seen and sent to the SUBJECT's computer upon connection to the VICTIM.

The SUBJECT attempted to delete the user *t* from the system but was unsuccessful.

The SUBJECT reset his virtual terminal to *vt100.* (RESPONSE TEAM note: vt100 is the default terminal used on most terminal emulation programs). SUBJECT uses pico to edit the /etc/passwd file, changes nothing, and ends his connection while still editing the file at **Fri Mar 31 0048 EST 2000.**

On Friday Mar 31 0223 EST 2000 , the SUBJECT (**38.27.198.22**: ip22.charlotte.nc.pub-ip.psionic.net) connected to the VICTIM machine (193.232.87.15: dnsone.sc.doh.gov) using TELNET (TCP port 23) which grants super user access to the VICTIM machine using a backdoor in the login program. The SUBJECT used the user *r00t* and the password *w00p!.* (RESPONSE TEAM note: Telnet (TCP port 23) is a remote access protocol that allows one to have a "virtual terminal" connected to any host on the Internet possessing a Telnet server. Once a Telnet session is established and a valid user name and password are entered, keystrokes entered at the user's keyboard are executed on the remote host, and output from the remote host is displayed on the user's screen.) (RESPONSE TEAM note: A root-level account is an account with super-user privileges that have unrestricted read, write, and execute access for the entire computer.) The consent to monitoring banner was seen and sent to the SUBJECT's computer upon connection to the VICTIM.

SUBJECT switched to the */usr/sbin* directory and runs the command *uptime* which outputs how long the system has been running since last reboot. SUBJECT then removes the entire */var/log* directory (*rm -rf /var/log*) deleting all evidence of system logs recording his presence. SUBJECT ends the connection on ***Friday Mar 31 0224 EST 2000***.

DISCUSSION OF CONNECTIONS:

Initially thought that there might be more than SUBJECT. After following connections through it is apparent that all the connections to the VICTIM system are linked. The SUBJECT changed his trojan login program which changed the method of entry which initially threw me off. The SUBJECT repeatedly used the user name *t* for FTP transfers in addition to several attempted TELNET sessions. It is my opinion that all of these connections are indeed the same SUBJECT. It is unknown why he deletes the */var/log* directory after running the log-file cleaner unless he just doesn't trust the capabilities of the program. It seemed he came back into the system just to remove that file.

H@x0r in the channel did explain how to access the VICTIM machine using the virtual terminal and giving the hostname. This would explain why later on he changed the trojan to keep others from being able to use that as a means to connect to the VICTIM.

End Synopsis:

The information provided to law enforcement enabled a search warrant to be executed on the address of the subscriber of the cable modem. Law enforcement verified that the SUBJECT illegally broke into the computer during the interview. No further information is known.

Eradication

The cause of the compromise was an unpatched DNS server running BIND that was shipped standard with Redhat version 6.0.

Since the SUBJECT was found based on evidence gathered, no need to consider honey pots or leaving the system open was necessary.

The system at site Alpha was restored using the last known good version used prior to the incident took place to ensure that the backdoors were successfully eradicated.

With the restored system, the latest patches for BIND and the latest revisions were checked. The system security settings were checked and all rogue services were shut down.

A vulnerability analysis was done on the VICTEM system and the other systems surrounding it on the demilitarized zone. Systems were patched and extraneous services were shut down as necessary.

A Red Team was used to try and get in using similar exploits and was unsuccessful on this machine and any other on the demilitarized zone. No other machines were found to be vulnerable at all.

Recovery

After the systems were backed up and security checks were in place, additional methods of security were considered. A policy review of the settings on the external firewall were conducted in addition, a new intrusion detection system was purchased and installed on the network which would watch for more than just connection on FTP or TELNET.

In addition to the intrusion detection system, TRIPWIRE was installed and configured on the VICTIM DNS.

With the additional security measures in place, the system properly patched and unnecessary services turned off, and the firewall reconfigured no further action was necessary. The DNS has been functioning properly and no further attempts at compromise have been detected.

The DNS was restored on the network on 5 April 2000.

Follow-Up

The Intrusion Detection System was sub-par since it did not detect the break-in from the offset. This was fixed by replacing the intrusion detection system.

The incident was contained and no further compromises on the network occurred.

Site bravo's evidence would have been nice to correlate the evidence gathered at the local site. It might have helped produce a more stringent case against the SUBJECT.

Operations were not upset due to the fact that the DNS was used primarily for incoming connections only. Administration deemed it did not affect operations to lose the ability for outside users to access the internal systems.

The IRT handled the situation efficiently and quickly. Law Enforcement was able to use synopsis to locate and identify a subject.

Parts of this report were also printed up in a follow up report submitted to management and the system administrators.

No further action is necessary. Incident Closed. **18 April 2000**

APPENDIX 1 PROCEDURES and SUGGESTIONS TO HELP ANALYZE BINARY FILES FROM A VICTIM SYSTEM (process used to assess and contain and operating system commands.

(These are the procedures used to analyze binaries found on VICTIM system written by myself. I cannot show the actual output of many of these commands due to the fact that the results lead to sensitive case related details. The procedures are the same though.)

If you have a test network I would move the binaries to that network.

The FIRST SECTIONS COVERS the commands FILE, GREP, AND STRINGS for GENERAL FILE ANALYSIS.

The first one is the command "file".

When one first compiles a computer program you will find that either the binary is dynamically linked, statically linked, with debug output, or stripped.

+ Dynamically Linked
    -- Requires the availability of libraries loaded by the operating system.
+ Statically Linked
    -- All functions are included in the binary. But with a price that the results in much larger executable file. This is typically seen in commercial apps like Netscape, Staroffice, etc.
+ Debug Output
    -- Includes Debugging Symbols (e.g. variable names, functions, internal symbols, source line numbers, source file information)
+ Stripped
    -- Discards symbols from object files and makes executable much smaller

It is interesting to note that lesser skilled hackers will compile programs straight from the "makefile" without adjusting the properties inside that file. For instance, the default for tfn (TRIBE FLOOD NETWORK) and some other programs compiles the program with debugging symbols. If a hacker is smart and he is good he will statically link his executable and then strip it. This would make the binary VERY hard to take apart. But luckily for us security types and forensic experts, most hackers do not go to this detail. But THE GOOD ONES WILL!!!

The second one is the command "strings -a" which will display printable characters sequences that are at least 4 characters long and the -a command shows all strings in the binary file.

+ You can tell a BUNCH from just this alone. Compare it against strings of other binaries so you get used to knowing what to look for.

The third one is "grep". Hopefully I dont need to go into too much detail here.

THE SECOND SECIONT COVERS the NON-RUNNING PROGRAM FILE ANALYSIS

The Unix debuggers "gdb, ltrace, strace, xxgdb, memprof"
"gdb, xxgdb, memprof"
+ Allows the actions taken in a program to be monitored
+ displays function and variable names

"strace"
+ System Call Tracer
+ Wiretap on the interactions between a program and the operating system
+ displays information on file access, network access, memory access, and tons of other calls
+ Drawback is that the program is actually run (USE A STANDALONE MACHINE!)

"ltrace"
+ log every library routine call

"nm"
+ display compiler and runtime linker symbol table

"ldd"
+ identify dynamic libraries used

THE THIRD SECTION DEALS WITH RUNNING PROGRAMS

Finally once you have gotten this far execute the program on a stand-alone machine OR if you have a program in MEMORY that no longer resides on the box itself you need to do the following. DO NOT CONNECT TO THE PORT THE PROGRAM IS RUNNING ON (BAD THINGS COULD HAPPEN!).

If you are on a live box, SUSPEND the process until you have figured out what it is. KILL -STOP <pid>

The /proc directory -- pseudo-filesystem which is used as an interface to kernel data structures
+ contains links to executable code (EVEN IF DELETED---> CAN RETIREVE BINARY OF A DELETED PROCESS)
+ File Descriptors (EVEN IF UNLINKED--> CAN RETIRVE CONTENTS OF AN UNLINKED FILE)
+ there is a Numerical Subdirectory for each Running Process

+ To Attain a Process

1. Identify the Process ID of the Target Process Using
    ++ ps -auxeww for command, environment, and more
    ++ reviewing contents of /proc
    ++ using network monitor logs
    ++ using the program lsof
2. Change directories to the /proc/<pid>
3. Copy the exe Link to your chosen destination
4. Once you have the binary. Do the initial file analysis, the debugging, etc...

For Solaris:
    object/a.out (program file)
    ms (process memory)
    map (memory map)

For Linux
    exe (program file)
    mem (process memory)
    maps (memory map)

For FreeBSD
    file (program file)
    mem (process memory)
    map (memory map)

The program "top"

    ++ useful process information

THE program "lsof"

++ This program will list all the open files and sockets on a system. You will be able to tell if something is listening for connections.
    ++ You can see the inode of the files and configuration files that were used. This is useful if you want to look at the inodes that had deleted files in their place. You should be able to pull out the data.
    ++ You can see the libraries that are used with the program. Really easy to see sniffers here. (PCAP?)

        ++ run lsof -p <pid>

The program "gcore" (note gcore is not available on LINUX but other alternatives exist)

++ This program will produce a core file dump from the running process. VERY useful if you would run strings on it.

Binary analysis is long, time consuming, and sometimes a non-result process.