GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at http://www.giac.org/registration/gcih

Exploit in Action:

The NetDevil Trojan in an Educational Environment

by

Ryan L. Means

For GCIH v2.1a (Option 1)

© SANS Institute 2003, Author retains all rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Abstract

A student takes advantage of an instructor’s trust to execute the NetDevil trojan using an AutoRun CD. The student then connects from home and downloads a copy of the upcoming final exam. He proceeds to exploit other machines on campus before being caught by IT staff at a professional school in the University.

This paper will describe the NetDevil trojan, give an overview of the networking environment on which the attack occurred, and provide the details of the attack. After analyzing client/server communications, we will be able to create a Snort IDS rule that will alert security personnel to a potential conversation between the client and server. The paper will also describe what can be done to protect against the trojan or to clean it from a compromised machine.

The final portion of the practical will fit the incident handling procedure of the IT staff at the professional school into the SANS-defined six stages of the incident handling process.

This paper has been sanitized to disguise the identities of the users, the attacker, and all of the machines.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Table of Contents

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The Exploit

Name

Anti-virus or anti-trojan vendors give different names to the NetDevil 1.1 trojan used by the attacker.

A search on Vgrep[1] offers some of the possible names from the vendors:

- **Computer Associates:** Win32.NetDevil.11.A
- **McAfee:** Backdoor-RP.svr
- **Sophos:** Troj/NetDevil11
- **Symantec:** Backdoor.trojan and Backdoor.NetDevil(.B/.Dr)
- **Trend Micro:** BKDR_NETDEVL.11A

Operating System

Version 1.1 of the NetDevil server works only on Windows 95/98 with Dial-Up Networking and Winsock 2. If the victim machine does not have Dial-Up Networking or Winsock 2 installed, the trojan executable will fail to find the appropriate libraries in the missing DLLs and will not execute.

NetDevil 1.1 will not affect Windows NT, Windows ME, Windows 2000, or Windows XP because they are missing the specific DLL versions that the server needs to operate. Likewise, the trojan will not execute on Macintosh or any UNIX variants.

The client software used by the attacker to access the server seems to work on all versions of Windows from 95 to XP; however, version 1.1 of the client software does not transfer files correctly on XP.

Protocols/Services/Applications

By default, NetDevil 1.1 opens port 901/TCP for accepting commands from the client, 902/TCP for transmitting keystrokes, and 903/TCP for transferring files between the server and client.

NetDevil 1.1 can also optionally use the following services to notify the attacker that the compromise has been successful:

- **SMTP (Port 25):** Notification through email message sent to the STMP port of the configured mail server
- **HTTP (Port 80):** Notification through HTTP GET request, passing information about the compromised machine in HTTP parameters
- **ICQ (Port 4000):** Notification through ICQ instant message to an account accessed by the attacker

This trojan, like most others, can affect any and all applications running on the victim's machine.

[1] Vgrep: http://www.virusbtn.com/resources/vgrep/index.xml

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Brief Description

NetDevil 1.1 is a client/server trojan written by “Nilez” that runs on Windows 95 and 98. The server portion is configured through a utility that allows the attacker to specify what ports to use, notification methods, as well as several other options. The server is then executed on the victim’s machine and provides access to the client through the previously specified ports. The client can then transfer files, control processes and windows, capture screen contents and keystrokes, execute arbitrary commands, and perform other “annoying” activities on the victim’s machine.

Variants

While the attack studied in this paper was carried out using NetDevil 1.1, there were previous and subsequent versions with minor differences:

- Version 1.0 used port 6667/TCP as the control port and did not offer mail and CGI notification.
- Version 1.2 changed the default server name from `SHELLAPI32.EXE` to `ADVAPI.EXE`.
- Version 1.3 changed the default server name again to `NETAPI32.EXE` and added “webcam spy” functionality. The client software in 1.3 is now fully compatible with Windows XP.
- Version 1.4 supports Windows NT4 and Windows 2000 and provides added registry editing functionality.
- Version 1.5 changed the default server name again to `KERNEL32.DLI` and added a registry entry to execute .DLI files.

There are also references to two variants called Backdoor.NetDevil.B and Backdoor.NetDevil.Dr on Symantec’s virus encyclopedia. Variant B seems to be the same as version 1.5 and variant Dr is the NetDevil trojan packaged with an antivirus process killer and another trojan.

References

Information about the NetDevil trojan can be found at the following sites:

Symantec Security Response
http://securityresponse.symantec.com/avcenter/venc/data/backdoor.netdevil.html

MegaSecurity.org (screenshots and file sizes)
http://www.megasecurity.org/trojans/n/netdevil/Netdevil_all.html

McAfee Security
http://vil.mcafee.com/dispVirus.asp?virus_k=99295

Computer Associates
http://www3.ca.com/virusinfo/virus.aspx?ID=10948

© SANS Institute 2003, Author retains full rights.

Version 1.1 of NetDevil is no longer available for download. Version 1.5 of NetDevil is available at:
http://astalavista.com/trojans/tools/trojans/net-devil-1.5.zip

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The Attack

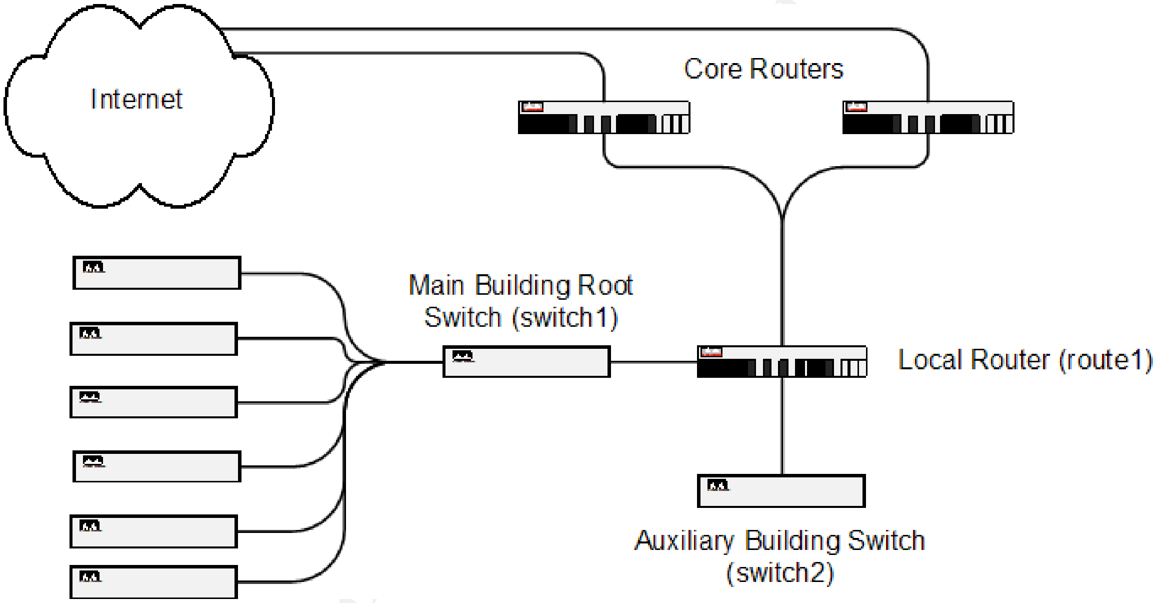
Description and diagram of network

For the purposes of this paper, the scope of the attack will be limited to the systems directly under my supervision. While the attacker did successfully compromise machines outside of my control, I have no knowledge of the attack methodology or incident handling procedures conducted on those systems.

Network Topography

At the time of the attack, the network was configured as illustrated in Figure 1.

Figure 1: Network topography of the professional school


Internet
Core Routers
Main Building Root Switch (switch1)
Local Router (route1)
Auxiliary Building Switch (switch2)

Traffic from the Internet to the professional school comes down through the border routers into one of the two core routers and is then sent to our local router (route1). Depending on which of the four buildings the packet is destined for, it is then routed to either the root switch (switch1) of the main three buildings or to a switch in the auxiliary offices (switch2). Switch 1 houses seven of the eight subnets in the professional school, while switch 2 handles the twenty or so users on the subnet in the other building. Switch 1 then distributes the traffic to six other switches throughout the three main buildings of the school before it reaches its destination.

Other than ingress and egress filtering on the border routers of SNMP traffic, there are no firewalls on the networks that were utilized in this attack. The professional school's network is wholly maintained by central campus networking, who traditionally did not allow the installation of firewalls due to the problems they create for network management tools. The campus networking group has since released a set of recommendations for firewalls on campus, but their proposal has yet to be funded. Several schools and departments, including the professional school discussed in this attack, have recently proceeded to independently research and fund their own firewalls in lieu of this deficiency.

The IT staff at the professional school also maintain their own web, mail, file, and print servers for the use of staff and faculty in the school. These servers are hosted on various subnets in the main building and are independent from the web and mail servers administered by central campus.

Affected System

The machines in the main three buildings of the professional school are centrally managed and maintained. They are almost exclusively Windows 2000 systems on an NT domain and are all regularly patched and kept up-to-date with the latest virus definitions and engines by the workstation support personnel. Unfortunately, the targeted machine was in the auxiliary building at the school. The machine that belonged to the attacker's instructor was running Windows 95 OSR2 on a Pentium-II/400 with very out-of-date virus definitions. The machines in the auxiliary building were not supported by the IT staff in the main buildings because of a historical precedent of separation between the two groups. The staff and faculty in the auxiliary building were independently funded and had a different mission than the mission of the main professional school. This is a common occurrence in a university environment. It is also important to note that the users in the auxiliary office were using Eudora as an email client; this will be relevant in the examination of the attack.

Protocol/transmission vector description

This attack did not involve the exploit of a particular protocol, service, or application, but rather of a user's trust. The trojan was executed by AutoRun from a CD that was placed in the instructor's CD-ROM drive and was not installed over the network. NetDevil could, however, be transmitted through various other means. Email or instant messaging services are the primary vectors, where trojans are usually attached to a "dropper". A dropper is another application that makes the trojan attractive to the user (i.e. a game or utility), and when the dropper is executed, the attached trojan is installed before the dropper application runs. These droppers can be attached to email messages, sent through file-transfer systems on instant messaging services, or directly installed on the machine if the attacker has access. None of these methods of transmission exploit a particular vulnerability in a protocol, service, or application, but it is possible that another exploit could be used to give the attacker means to install the trojan.

Once the NetDevil trojan was installed, the attacker communicated with it through TCP ports 901, 902, and 903. NetDevil most likely uses TCP ports over UDP ports because it needs a reliable data connection to transfer screenshot, keystroke, and file data. Further analysis of the traffic between a NetDevil client and server will be provided in the following sections.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

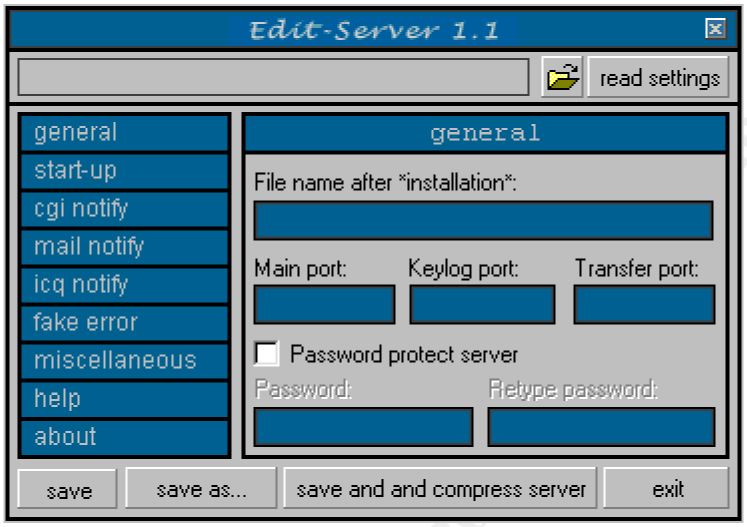
How NetDevil works

NetDevil is a client/server trojan application, and as such comes in two parts. This paper will first examine the server application and then the client before delving into the specifics of the communication between them.

Server

The `Server.exe` file that comes with the NetDevil archive is the heart of the trojan. A server configuration utility is included in the file `Edit-server.exe`. The interface for the server configuration utility is displayed in Figure 2.

Figure 2: The edit-server utility



After the `Server.exe` file is opened in the configuration tool, the attacker has the option of configuring a host of different options.

First, the file name after installation must be set. By default in 1.1 this is set to `SHELLAPI32.EXE`, but it can be changed to any value. Then, the attacker has the option of changing the main, keylog, and transfer ports from their defaults of 901, 902, and 903, respectively. The server can also be password protected to prevent other attackers from using the compromised machine. The start-up menu gives the attacker the option of choosing the method by which the trojan will be executed. A personalized key can be placed in either `HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices` or `HKLM\Software\Microsoft\Windows\CurrentVersion\Run` to execute NetDevil on system startup.

Next, the attacker can choose from several notification options: CGI, mail, and ICQ. The notification will be sent to the attacker upon successful compromise of the machine. The CGI notification option will cause the installed server to send an HTTP GET request to the specified server. The request will contain a set of parameters that corresponds to the victim's IP address, port number on which the trojan is installed, username, and server password. The server can also notify the attacker by sending an email message to an open

SMTP relay, which is configurable with the tool. Additionally, NetDevil could notify an attacker of a compromised machine by sending an ICQ WebPager request. This is basically a GET request to http://people.icq.com/wwp/1,,,00.html, formatted so that the CGI script at that page will transmit a message to the specified user on the ICQ network.

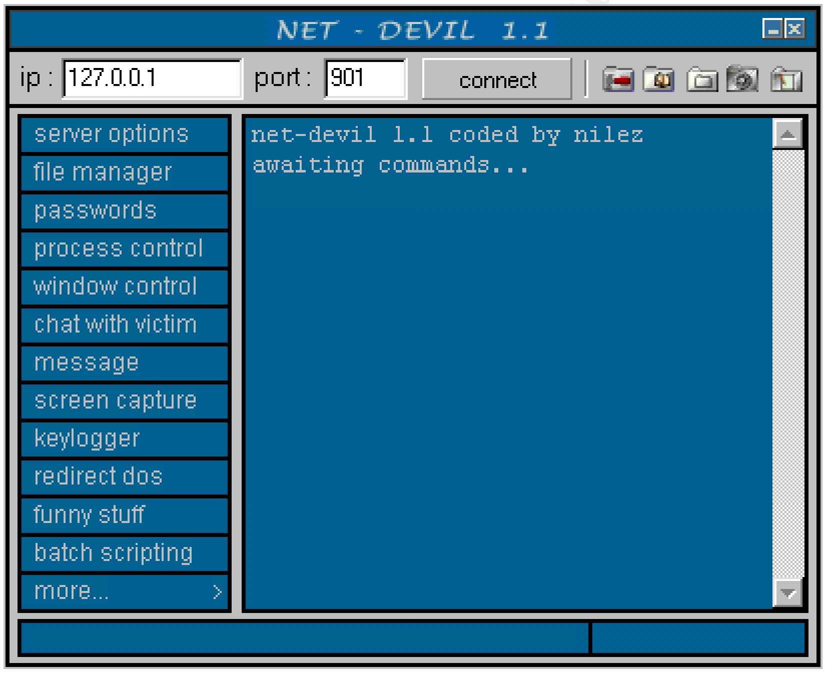
The server configuration utility also gives the attacker the option of setting up the server to give a fake error message on installation, perhaps to cover up any suspicion by making the victim think that the program did not successfully execute.

Once the server has been configured, it can be automatically byte-packed by the included `upx.exe` application by selecting "save and compress server". This supposedly makes the server harder to detect when it's scanned by anti-virus and anti-trojan programs. After the server has been packed, it is ready to distribute to the victim through one of the previously outlined vectors.

Client

The client provides the attacker with an interface to the server that has been installed on the victim's machine. The client for NetDevil 1.1 is in the file `Net-Devil.exe`, as illustrated in Figure 3.

Figure 3: The NetDevil client



```
net-devil 1.1 coded by nilez
awaiting commands...
```

To connect to the server, the attacker must enter the IP address and port number of the server on the compromised machine. After clicking "connect", the client will respond with a status message indicating the success or failure of the connection attempt. After the client is connected, the attacker can use any of the

Author retains full rights.

options on the left menu to control the victim's machine. Each of the menu items is briefly described below:

- **File Manager:** Allows the attacker to transfer files to or from the victim's machine.
- **Passwords:** Reads cached passwords from the registry for RAS and Internet Explorer.
- **Process Control:** Gives the attacker the option of killing running processes or removing their executables from the hard drive.
- **Window Control:** Allows the attacker to close, show, or hide windows as well as change window focus, window titles, or just send a string of text to a window.
- **Chat with Victim:** Opens a chat window on the victim's machine where the attacker can send and receive messages.
- **Message:** Creates a Windows pop-up dialog box on the victim's machine with the attacker's message.
- **Screen Capture:** Allows the attacker to receive small or large JPEG screenshots of the victim's machine. The small screenshots are sent in rapid succession, to simulate motion, while the large screenshots are transmitted on request.
- **Keylogger:** Logs the victim's keystrokes and then gives the attacker the option of saving them to a file.
- **Redirect DOS:** Executes a DOS command and redirects the output to the attacker.
- **Funny stuff:** Plays tricks with the victim's mouse, start menu, keyboard, etc.
- **Batch scripting:** Copies a batch file onto the victim's computer and then executes it on command.
- More…: Gives the attacker the option of modifying system files, sending files to the victim's printer, and doing a host of other "annoying" things.

The NetDevil client gives the attacker a great deal of control over the victim's machine by sending commands to the server on port 901 (or another configured port). In the next section, we will look at the specific communication that occurs between the client and the server, in order to better understand how it can be identified and blocked.

Communication

To analyze the connection between the client and the server, two test machines are set up on a private network. The server runs Windows 95, while the client machine runs Windows XP and tcpdump to analyze the traffic.

The NetDevil client communicates with the server by sending ASCII strings as commands and then awaiting the appropriate response.

© SANS Institute 2003, Author retains full rights.

Connection

The connection is always initiated by a packet exchange resembling the following:

```
IP 192.168.0.2.3614 > 192.168.0.1.901:
S 1053887573:1053887573(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
IP 192.168.0.1.901 > 192.168.0.2.3614:
S 17172037:17172037(0) ack 1053887574 win 8760 <mss 1460,nop,nop,sackOK> (DF)
IP 192.168.0.2.3614 > 192.168.0.1.901: . ack 1 win 64240 (DF)
```

These three packets show a standard three-way handshake between the client at 192.168.0.2 and the server at 192.168.0.1. The client initiates the connection from an ephemeral port randomly chosen by the TCP stack of the client machine for each connection, in this case 3614. The client sends a zero-data SYN packet with a starting sequence number for the connection. For this connection, the sequence number is 1053887573. The server then responds to the SYN packet with a packet that has both the SYN and ACK flags set. This second packet is letting the client know to use the sequence starting at 1712037. The client signifies that it has received the SYN-ACK by returning a packet with just the ACK flag set. It is also important to note that all of the packets sent between the server and client have the don't-fragment flag set. The previous packet sequence is used to establish a connection between the client and the server on port 901 of the victim's machine. After the connection is built, the following exchange always occurs:

```
IP 192.168.0.1.901 > 192.168.0.2.3614: P 1:7(6) ack 1 win 8760 (DF)
       passed
IP 192.168.0.2.3614 > 192.168.0.1.901: P 1:8(7) ack 7 win 64234 (DF)
       version
IP 192.168.0.1.901 > 192.168.0.2.3614: P 7:45(38) ack 8 win 8753 (DF)
       ver1.1<CR><LF>C:\WINDOWS\SYSTEM\<CR><LF>903<CR><LF>902
IP 192.168.0.2.3614 > 192.168.0.1.901: . ack 45 win 64196 (DF)
```

The first packet is sent from the server to the client with the push, ack, and don't fragment (DF) flags set and it contains the data "passed". The push flag forces the receiving TCP stack to pass the data along to the application immediately rather than keeping it until the stack buffer is full. The ack flag is merely an acknowledgement of the reception of the previous packet and in this case is set on every packet except the first. The DF flag tells a router or host not to fragment the packet to fit on a network with a smaller window size, but to drop the packet and return a response to the sender that the packet is too big. The data string indicates to the client that the connection has been established and that the server is valid and ready to accept commands. After this initial acknowledgement, the client sends another pushed don't fragment packet that says "version". This packet requests the version of the server. The server then responds with a pushed DF packet that contains the data: "ver1.1" – the version of the server, "C:\WINDOWS\SYSTEM" – the installation directory of the server, "903" – the port number of the keystroke logger, and "902" – the port on which file transfers will occur. Each of these values is separated by a carriage return

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003,   As part of GIAC practical repository.   Author retains full rights.

and line feed pair (CRLF). These responses in the third packet will vary from server to server depending on how it was initially configured by the attacker. The fourth packet is simply an acknowledgement from the client that the requested information is received. After the seven packets above have been exchanged, the client will indicate to the attacker that it is connected to the server and ready to issue commands to the victim's machine.

Command execution

To show the command execution communication, we will look at the packet exchange when the client enumerates the processes on the victim's machine. However, almost all of the commands that the client issues to the server follow the same pattern. The packet sequence for the process enumeration command is:

```
IP 192.168.0.2.3614 > 192.168.0.1.901: P 8:20(12) ack 45 win 64196 (DF)
       getprocesses
IP 192.168.0.1.901 > 192.168.0.2.3614: P 45:341(296) ack 20 win 8741 (DF)
       getprocessesC:\WINDOWS\SYSTEM\KERNEL32.DLL<CR><LF>
       C:\WINDOWS\SYSTEM\MSGSRV32.EXE<CR><LF>C:\WINDOWS\SYSTEM\MPREXE.EXE…
IP 192.168.0.2.3614 > 192.168.0.1.901: . ack 341 win 63900 (DF)
```

This three packet exchange shows the client requesting the process list from the server with a pushed, DF packet containing the ASCII string "getprocesses". The server replies with "getprocesses" and then a CRLF-separated list of the processes running on the victim's machine. This list is parsed by the client software and displayed to the attacker in a window. After receiving all of the requested information, the client sends a zero-data ACK packet back to the server to finish the conversation.

File Transfer

File transfer is conducted over NetDevil's transfer port, which is set to 902 by default. The client connects directly to this port to request file upload or download, without first notifying the server through the main port. A sample file transfer capture of C:\NETLOG.TXT follows:

```
IP 192.168.0.2.1423 > 192.168.0.1.902:
S 451197968:451197968(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
IP 192.168.0.1.902 > 192.168.0.2.1423:
S 455576:455576(0) ack 451197969 win 8760 <mss 1460,nop,nop,sackOK> (DF)
IP 192.168.0.2.1423 > 192.168.0.1.902: . ack 1 win 64240 (DF)
```

The transfer starts off with the standard three-way handshake as described previously, except this time to the file transfer port (902) and using a different ephemeral port (1423).

```
IP 192.168.0.2.1423 > 192.168.0.1.902: P 1:16(15) ack 1 win 64240 (DF)
       C:\NETLOG.TXT<CR>0
IP 192.168.0.1.902 > 192.168.0.2.1423: P 1:5(4) ack 16 win 8745 (DF)
       1104
IP 192.168.0.2.1423 > 192.168.0.1.902: P 16:17(1) ack 5 win 64236 (DF)
       C
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003,   As part of GIAC practical repository.   Author retains full rights.

Here, the client sends a packet to the server with the data "C:\NETLOG.TXT", which is the filename of the file, then a carriage return and either a 0 or a 1. The last bit determines whether the file is to be sent from the server (0) or received to the server (1). In this case, the client is downloading a file, so it keeps the bit off. The next packet is sent from the server to the client and it contains the size of the file to be transmitted in bytes (1104 bytes). This way, the client can know when it has retrieved all of the files from the server. The final packet contains the character "C", which seems to instruct the server to begin transmitting the file data as illustrated below.

```
IP 192.168.0.1.902 > 192.168.0.2.1423: P 5:261(256) ack 17 win 8744 (DF)
        File data
IP 192.168.0.2.1423 > 192.168.0.1.902: P 17:18(1) ack 261 win 63980 (DF)
IP 192.168.0.1.902 > 192.168.0.2.1423: P 261:517(256) ack 18 win 8743 (DF)
        File data
<trimmed>
```

Here we see the file data transmitted from the server to the client in 256 byte chunks. Upon receipt of each packet, the client responds with a lone ack packet to acknowledge the transfer of the previous data.

```
IP 192.168.0.2.1423 > 192.168.0.1.902: F 21:21(0) ack 1109 win 63132 (DF)
IP 192.168.0.1.902 > 192.168.0.2.1423: . ack 22 win 8740 (DF)
IP 192.168.0.1.902 > 192.168.0.2.1423: F 1109:1109(0) ack 22 win 8740 (DF)
IP 192.168.0.2.1423 > 192.168.0.1.902: . ack 1110 win 63132 (DF)
```

After all of the data has been transferred, the client sends the server a packet with the fin and ack flags set. This both acknowledges the receipt of data from the previous packet and indicates the client's intention to terminate the connection. Upon receiving this request, the server sends an acknowledgement packet and then its own request to terminate the connection back to the client. The client then responds with an ack to finalize the exchange before the connection is torn down by the TCP stack.

This same process is used for the screenshots taken by the NetDevil server. They are requested by the client on port 901 and then transferred to the client from the server on port 902.

Keystroke logging

NetDevil has the ability to log keystrokes in real time and transmit them back to the client for analysis. The keystrokes are transmitted on the keylog port, 903, by default. Keystroke logging is initiated on port 901 by the client with the following packet exchange:

```
IP 192.168.0.2.3614 > 192.168.0.1.901: P 4455:4646(11) ack 4366 win 63725 (DF)
        keylog_open
IP 192.168.0.1.901 > 192.168.0.2.3614: P 1:17(16) ack 11 win 8708 (DF)
        keylog_open_done
IP 192.168.0.2.3614 > 192.168.0.1.901: . ack 17 win 63709 (DF)
```

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The client sends a request to the server with the string “keylog_open” to request that the server begin logging the keystrokes. The server then responds with the string “keylog_open_done” to indicate that the logging has begun and that the client should connect to 902 to retrieve the data. The following packet exchange then occurs on port 902:

```
IP 192.168.0.2.1428 > 192.168.0.1.903:
S 506960761:506960761(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
IP 192.168.0.1.903 > 192.168.0.2.1428:
S 699383:699383(0) ack 506960762 win 8760 <mss 1460,nop,nop,sackOK> (DF)
IP 192.168.0.2.1428 > 192.168.0.1.903: . ack 1 win 64240 (DF)
```

First the client opens a connection with a three-way handshake on port 903 as described previously.

```
IP 192.168.0.1.903 > 192.168.0.2.1428: P 1:27(26) ack 1 win 8760 (DF)
       <CR><LF>*[ The Window Title ]*<CR><LF>a
IP 192.168.0.2.1428 > 192.168.0.1.903: . ack 27 win 64214 (DF)
IP 192.168.0.1.903 > 192.168.0.2.1428: P 27:38(11) ack 1 win 8760 (DF)
       bcdefghij<CR><LF>
IP 192.168.0.2.1428 > 192.168.0.1.903: . ack 38 win 64203 (DF)
IP 192.168.0.1.903 > 192.168.0.2.1428: P 38:39(1) ack 1 win 8760 (DF)
       l
IP 192.168.0.2.1428 > 192.168.0.1.903: . ack 39 win 64202 (DF)
IP 192.168.0.1.903 > 192.168.0.2.1428: P 39:40(1) ack 1 win 8760 (DF)
       m
<trimmed>
```

Then, the server sends the client a packet containing a CRLF and then the title of the window in which the typing is occurring, in star-brackets (*[,]*). After the window title is another CRLF pair and then the actual keystrokes typed into the window. In this case the string “abcdefghij<CR><LF>lm” was typed. The server intercepts the keyboard input and sends it out in packets at regular intervals. Because of the small scale of these intervals, I’m not sure of the frequency at which the keyboard input is packaged and sent to the client.

```
IP 192.168.0.2.1428 > 192.168.0.1.903: F 1:1(0) ack 52 win 64189 (DF)
IP 192.168.0.1.903 > 192.168.0.2.1428: . ack 2 win 8760 (DF)
IP 192.168.0.1.903 > 192.168.0.2.1428: F 52:52(0) ack 2 win 8760 (DF)
IP 192.168.0.2.1428 > 192.168.0.1.903: . ack 53 win 64189 (DF)
```

The connection is then terminated by the client when the user selects to stop the logging. The termination follows the same FIN-ACK/ACK/FIN-ACK/ACK sequence described previously.

Description of the attack

The actual execution of the attack was incredibly straightforward. The information in this description was obtained from a police interview with the attacker as described later in this paper. Over the course of several days, the attacker sent email messages to his instructor with the NetDevil trojan attached. He forged the email to look like it was coming from associates of the instructor that the attacker knew the instructor would trust. Fortunately, over this period of

© SANS Institute 2003, Author retains full rights.

time, the instructor never opened any of the attachments from the attacker, mostly due to the fact that the content of the message didn't match up with the inclusion of an attachment. For example, the email message would advertise an upcoming conference for a particular organization and then include an executable attachment called conference.exe. The instructor knew enough to not click on the attachment, but he did not notify the IT staff at the school.

After several days, the attacker became frustrated that his attempts to access his instructor's machine were not producing results. He scheduled a meeting with the instructor in our auxiliary offices under the guise of wanting to "show him a website that he developed". He also wished to obtain one of the professor's papers for posting on his website, which was geared to freelance writers. When the student arrived, he placed a data CD in the instructor's CD-ROM drive containing the website that he wished to demonstrate. The professor claims that the attacker did in fact show him a website from the CD, but at the same time, the machine was executing an `autorun.ini` file on the disc.

The AutoRun file tells a Windows machine which program to automatically execute when removable media is inserted in a drive. The `autorun.ini` file is placed at the root of the removable disk and has the following syntax:

```
[AutoRun]
open=whatever.exe
```

This simple file causes the operating system to execute `whatever.exe` when the removable disc is mounted.

This particular AutoRun file executed the NetDevil server on the instructor's machine. Because this machine was in the auxiliary offices, it was not centrally managed by our IT department and did not have a standard image. The virus scanner was very out-of-date and very few patches had been applied to the operating system. The impotent virus scanner was unable to detect the trojan as it was being executed. This entire process was carried out under the eye of the instructor with a great deal of bravado. At the end of the meeting, the professor had no idea that the student had installed a trojan on his machine.

The attacker then immediately returned home from his meeting and proceeded to connect to the NetDevil trojan and scour the instructor's hard drive for the upcoming final exam. He used the file transfer mechanism in NetDevil to copy over fifty files from the professor's machine on to his own computer. In fact, he copied so many files that he had to create subdirectories on his own machine to organize the documents he downloaded. He searched for anything that mentioned a final exam or assignment, grabbing everything related to his course and any previous offering of the course. The attacker also copied a file with his own name as the title and several jpegs, probably just snooping around for anything else that he might find interesting.

Not satisfied with his take so far, the student also fired up the keystroke logger and proceeded to snoop the instructor's active applications. In the span of just a few minutes, he was able to obtain the professor's email username and password as well as the logins for several websites relating to work done at the professional school.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

All of this malicious activity took place from the student's machine on a network belonging to a well-known broadband provider in California. As explained previously, there were no network impediments to the attacker as he accessed the trojan server with the client application. Neither the broadband provider nor any network operation group within the campus border was monitoring for this kind of traffic. In addition, the anti-virus software on the instructor's computer was horribly out of date and unable to play any role in preventing the installation of the backdoor.

Signature of the attack

In this case, the trojan was nowhere near stealthy, and a number of indicators could have alerted IT staff to its presence. First, any active virus scanner with up-to-date definitions would have detected NetDevil immediately by signatures inherently present in the binary. Secondly, the trojan installs a registry key in one of two places in the HKEY_LOCAL_MACHINE registry hive identified previously in the paper. It also places a copy of itself somewhere on the machine, but in this case the default of `C:\WINDOWS\SYSTEM\SHELLAPI32.DLL` was left unchanged. Once the trojan is installed, it communicates with the client through a very predictable, easy to isolate TCP conversation. The connection sequence alone could be identified by the following Snort rule:

```
alert 192.168.1.0/24 any -> any any (flags: P,12; \
                                     fragbits: D+; \
                                     content: "passed"; \
                                     msg: "Possible NetDevil connect";
```

The syntax of this rule is relatively easy to follow. It's looking for any outbound traffic from the 192.168.1.0 subnet that matches the following criteria:

- Push flag set
- Don't fragment (DF) flag set
- Content of the data is the string "passed"

This rule would trigger an alert on any packet matching these conditions. Not only that, it would match the first packet that a NetDevil server sends out to a client when the client connects to it, as seen previously in the packet analysis. Given the fact that little regular traffic would transmit the single string "passed" with those particular flags set, this rule would not be overly broad and cause a flurry of false-positives. It is important to recognize that the rule triggers on any port, because while the NetDevil trojan uses ports 901, 902, and 903 by default, they can be changed. Using the detailed analysis given earlier, any number of Snort rules could be crafted to alert on particular NetDevil commands, though this seems unnecessary.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

How to protect against it

This section will be covered in two parts: how to isolate and remove the trojan once it has been installed, and how to protect against its installation in the first place.

Cleaning a compromised machine

In the case of compromise with the NetDevil trojan, the only safe thing to do with respect to restoring a compromised machine to good working order is to rebuild it from scratch. It is relatively easy to find the registry setting that runs the NetDevil process when the machine starts up. It is located in one of the two registry keys mentioned previously. Once the registry setting has been found, it will point the administrator to the location of the trojan itself, after which it can be deleted.

This would clean the machine of the NetDevil trojan itself, but rarely does an administrator know what else was done to the machine using the trojan. The attacker could have installed another trojan that is not easily detected by anti-virus or anti-trojan programs. In addition, the attacker could plant an application that would wait a specified amount of time before destroying all of the data on the disk or opening another backdoor. Without a file integrity system like Tripwire, there is no way to tell exactly what has been done. For that reason, it's essential to make a backup of the drive (for evidence or future study) and then wipe it clean before reinstalling the operating system.

Protecting against compromise

The best way to protect against compromise by a trojan is to run a good anti-virus program with real-time scanning at all times. A good virus scanner would have caught the student involved in the attack on his instructor red-handed. Virus scanners from Symantec, McAfee, and Trend Micro are capable of detecting this particular trojan. In addition to having the virus scanner, its of paramount importance that the virus definitions be kept up-to-date through updates that do not require the intervention of the user.

It is also extremely important to be aware of the dangers of "social engineering" and to educate yourself and your users that the network is not the only way that malicious executables can be installed on their machines. In this case, the instructor should have refused to allow his student to insert the disk into his drive without having its contents independently verified.

Turning off the AutoRun feature of Windows would also help reduce the risk with this kind of activity, but it's better to abstain entirely from introducing foreign media into your computer. To turn off the AutoRun feature, change the following registry setting on Windows NT, 2000, and XP:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Cdrom
Autorun = 0
```

Additionally, an IDS system at the border of the campus network with the rule illustrated previously would alert the IT staff to the presence of a NetDevil

© SANS Institute 2003,   As part of GIAC practical repository.   Author retains full rights.

trojan on their network. While it does not prevent the compromise of a computer, it would make the IT staff aware of the problem. Hopefully, they would be able to isolate the machine and remove the trojan before the attacker could use the server to do anything malicious.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The Incident Handling Process

Preparation

Before the incident occurred, there was very little in place in terms of formal incident handling procedures. It's not that we didn't have incidents, but none of them were particularly damaging. In addition, we had yet to be the target of specific attack on one of our users, rather than on just the hardware resource that a compromised machine provides.

Every now and then, a machine that was not secured by our image (kiosk machines, laptops, etc.) would be compromised and we would usually find out from the user of the machine, who would complain about excessive network traffic or hard drive activity. In every instance, the compromise was really just a script kiddie stashing some pirated software or music. Furthermore, it was a relatively simple task to forgo any kind of forensic analysis in favor of the quick and easy "Just Ghost it." With our small systems administration staff of two, it was not worth the expenditure of resources to track down an attacker or even research the vulnerability they exploited to compromise the machine. Whatever vulnerability it was, it was almost certainly patched in our standard image, which has yet to be compromised (knock on wood).

Because of this attitude, we never established any formal procedures. Our casual procedure would be for a user or one of the Helpdesk staff to notify one of the two systems administrators of a suspected compromise. One of the two administrators was responsible for server administration, while the other took care of workstations. In the case of an incident, either employee could act as a handler. After notification, an administrator would go to the affected box to make the decision as to whether or not the machine was in fact compromised. Because the notification usually came from the user, it was almost always immediately evident that the machine was not acting normally. In a few cases, the administrator may have had to run "netstat" to find the open port causing the trouble. At this point, the machine would either be Ghosted or be left alone, depending on whether the incident was real or a false alarm.

There is also a central campus network security group (NSG) with the mission of detecting intrusions and disseminating information to the administrators responsible for handling. Some of the goals of the NSG, as listed on their website, were to:

- Publicize current threats to computer security
- Work with departments to evaluate and improve their computer security practices
- Coordinate response to computer security incidents
- Assist with forensic analysis
- Ensure that responsible personnel are given all available information about an incident and understand what steps need to be taken to secure affected systems

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Though the group was scanning the border with IDS systems, if they had detected any intrusions before this incident, we were not alerted. Up until the time of this attack, the professional school was very self-reliant and we had never contacted the NSG.

Identification

Day One

It all started on a Friday afternoon in December 2001. Our professional school has a mailing list set up for users to communicate with our Helpdesk staff, and at around 2:30pm we received the following message (portions of the message have been sanitized):

```
maybe i'm paranoid, but this seems more like
a virus than a real message since i normally
would not get such a message and an executable
file is attached..

-------- Original Message --------
Subject: FWD: Change in Location of Today's Faculty Prandium & Talk
Date: Fri, 7 Dec 2001 02:39:59 -0800
From: Pete Johnson <pete.johnson@school.edu>
Reply-To: pete.johnson@school.edu
To: user@school.edu

Hi All,
Attached is the new luncheon schedule presentation for the rest of the
semester.
Please refer to it and try to make yourself available at that time.
Thanks,
Pete

>Dear All,
>There is a last-minute change in the location of today's faculty
>prandium and talk.  It is now being held in the Goldberg Room.  Lunch
>service begins at noon and the talk begins at 12:30 pm.  Thanks very
>much.
>Pete
```

The email was sent from a secretary for one of our faculty members to our support mailing list. The email also included an attachment named `luncheon.exe`. We were very surprised by the email that the secretary forwarded to us, mostly because the From and Reply-To addresses belonged to a user that had never existed at the school. It was also extremely anomalous because it contained new text attached to a message that had been sent out to all of the faculty several days earlier. We concluded that because of this, the message was not just a virus forging headers but it was indeed an intentional attack.

After coming to this conclusion, two of our IT staff decided to find out the contents of the attachment. They set up two clean computers on a private network with Windows 95 and executed the trojan on one of them. Then, they used the second machine to do a port-scan of the test machine using nmap, and

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

found port 901 open. Port 901 is not open by default on a clean installation of Windows 95, so they knew that the attachment had spawned a process that opened that port. One of the analyzers then took the executable and sent it to Symantec Anti-Virus Research Center (SARC) to see if they could determine exactly what it was. Several hours later, SARC returned an analysis showing that the file was a trojan, but they were not able to provide any further details about its name or origin.

At this point we contacted the secretary again and told her that the messages were from a malicious user and that she should let us know immediately if she saw any more. She then told us that one of the faculty she supported had also received similar messages the same day and had deleted them.

Curious about the sender, I searched through the SMTP (mail) logs for the day, looking for the message to the secretary with the same message ID as the mail that she forwarded us. My search produced the following results, which have been sanitized:

```
smtp:[07/Dec/2001:02:39:55 -0800] mail smtpd[14431]: General Notice:
SMTP-Accept:
GNYYXU01.S31:<4JZFzu9eHCj00000002@school.edu>:[x.x.4.94]:x.x.4.94:<pete
.johnson@school.edu>:331403:1:<user@school.edu>
```

This result shows that the message was sent to our mail server from an SMTP server at x.x.4.94. A quick traceroute showed the IP address belonging to a pool of IP addresses handed out to residential users of a local broadband service via DHCP. Still curious, I then searched through our web server logs from the past week or so and found some interesting results. The interesting results are shown below:

```
x.x.4.94 - - [07/Dec/2001:00:25:42 -0800] "GET /faculty/profiles
HTTP/1.1" 200 16038 "http://www.school.edu/faculty/" "Mozilla/4.0
(compatible; MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:25:46 -0800] "GET /faculty/profiles/xxx
HTTP/1.1" 200 6169 "http://www.school.edu/faculty/profiles"
"Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:25:47 -0800] "GET
/faculty/profiles/xxx/photo HTTP/1.1" 200 16735
"http://www.school.edu/faculty/profiles/xxx" "Mozilla/4.0 (compatible;
MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:26:32 -0800] "GET /faculty/profiles/yyy
HTTP/1.1" 200 6638 "http://www.school.edu/faculty/profiles"
"Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:26:33 -0800] "GET
/faculty/profiles/yyy/photo HTTP/1.1" 200 17783
"http://www.school.edu/faculty/profiles/yyy" "Mozilla/4.0 (compatible;
MSIE 5.5; Windows NT 5.0)"
```

While the times don't match up very well, I assumed that once the attacker's machine had gotten a lease with one address it would keep that address for a while. These particular logs provide a good amount of information.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

First, we can see that the attacker was accessing the site just a little after midnight on the day the forged email was sent. A little research, maybe? Also, I could see that the attacker was probably using Microsoft Internet Explorer on Windows 2000 (5.0). This data can be easily forged, but I assumed that was not the case. The attacker's choice of pages was also very interesting. First, he looked at the faculty profile of “xxx”, the Dean of our school. Just a minute later, he opened the faculty profile for “yyy”, the same faculty member that had been sent messages like those sent to the secretary. Interestingly enough, the faculty profile page contains the name of the secretary for the faculty member.

```
x.x.4.94 - - [07/Dec/2001:00:29:23 -0800] "GET /computing/howto/mail/
HTTP/1.1" 200 5311 "http://www.school.edu/computing/faculty-
staff.shtml" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:29:36 -0800] "GET
/computing/howto/mail/remote/ HTTP/1.1" 200 2210
"http://www.school.edu/computing/howto/mail/" "Mozilla/4.0 (compatible;
MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:29:48 -0800] "GET
/computing/howto/mail/netscape/messenger HTTP/1.1" 301 361
"http://www.school.edu/computing/howto/mail/remote/" "Mozilla/4.0
(compatible; MSIE 5.5; Windows NT 5.0)"
x.x.4.94 - - [07/Dec/2001:00:29:48 -0800] "GET
/computing/howto/mail/netscape/messenger/ HTTP/1.1" 200 421
"http://www.school.edu/computing/howto/mail/remote/" "Mozilla/4.0
(compatible; MSIE 5.5; Windows NT 5.0)"
```

These logs show that about three minutes after the first access, the attacker went to some informational pages kept on the IT portion of our site. These pages instruct faculty and staff how to configure their email clients from home, providing the name of the SMTP server that needs to be configured in the client. This is most likely the place where the attacker learned the DNS name of our mail server.

After all of this research, the end of the day was near. We decided to send out an announcement to or users warning them about the attachments that had been sent to the faculty member and his secretary. The email also informed our users that they should contact us immediately if they received any similar emails.

Day Two

The following Tuesday afternoon, the secretary contacted us again and said that she had received another email with a strange attachment. This attachment had a different name and size from the last one, but the format of the email was very similar. Like the first email, this email had a forged address that was similar to but not exactly the same as a real email address at the professional school. This email also had text attached to a message that had been sent out a few days earlier. We were very curious at this point and really wanted to find out how this individual was getting the initial email to add the text to. We assumed the attacker was either a user at the school on the mailing list or someone from the outside with a login and password for one of our email accounts.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Frustrated at the attacker's attempts to compromise our users' machines, I decided to investigate the machine that was sending the messages. I searched the mail log the same way I did on Friday and found the attacker's IP address. I fired up nmap and scanned the machine that had just an hour earlier sent the message to the secretary. The nmap scan revealed a number of open ports, including those for SMTP, HTTP, IMAP, NNTP, and the remote console application VNC. Using my own machine, I then opened my web browser to see if any pages were being served from the attacker's HTTP server. The first page to pop up had the name "myfriends.jp". A quick Google search on "myfriends.jp" revealed the name of the page's creator, Thomas Chang (this name has been changed). Our staff contacted the secretary and asked her if she knew of a Thomas Chang. We discovered that not only was he a student in her faculty member's class, but he had been suspected of cheating on a midterm examination.

At this point we realized the severity of the situation, that a student was attempting to illegally install a trojan on the machines of his instructor and of his instructor's secretary. Given the suspicion of his cheating on past exams, we came to the conclusion that the student was most likely trying to access material to cheat on the upcoming final. We also still had the unanswered question of how the student got the emails to the faculty mailing list in the first place and suspected that he had already compromised another machine.

We then called the University police department, who sent out an officer to take the report.

Day Three

On day three, the police took the evidence that my department had collected so far and took it to a judge to obtain a search warrant for Chang's residence. The police searched his apartment and retrieved his computer and any media or computer-related peripherals nearby. They also asked him to come in a few days later for an interview.

Day Four

The police asked me to conduct the forensic analysis on Chang's computer because of my knowledge of the facts leading up to the confiscation and because of their lack of resources to have the machine professionally analyzed. Under the supervision of a police officer, maintaining the chain of custody, one of the system administrators and I removed two hard drives from Chang's machine and used a Ghost boot-disk and a separate computer to clone the contents of the original drive onto another drive. We used the boot disk to ensure that the contents of the hard drive would not be modified during the cloning process. The Ghost application can be executed with the special switch, –IR, to ensure a forensic, or sector copy.[2] We then placed the original disks back into Chang's machine and created a second Ghost image of the duplicate disks,

[2] http://service1.symantec.com/SUPPORT/ghost.nsf/docid/1999110813413225

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

which I used to perform the forensic analysis. I submitted the following affirmation to the officer after the completion of this task:

> Under my supervision, [sys admin 1] removed the two hard drives from Chang's desktop. He used Symantec Ghost to image the original drives onto two new drives and reinstalled the original hard drives into Chang's machine. I then took another Ghost image of the duplicate hard drives and used those copies to perform my analysis. Neither the original drives nor the duplicate drives were altered for my analysis.

Day Five

On day five, I began analyzing the image from Chang's machine. My analysis was done directly on the Ghost image of the duplicate hard drive. Ghost provides a utility known as Ghost Explorer that lets a user view the contents of an image and extract files from the image for analysis. Having never performed such an in-depth forensic analysis before, I very methodically went through every directory looking for files that the attacker may have used in his attack or gathered from any of the machines that he attacked.

The hard drive was a gold mine. Chang kept his work organized by department and by machine, and I found that he had attacked not only the faculty member and secretary from my school but another faculty member as well as four graduate student instructors from other departments. He kept on his hard drive a list of the email addresses that he had sent the trojan to, ten in all, along with the specific email attachments and messages that were sent. There were also twelve subdirectories named by IP address containing an assortment of files. I assumed that these were files that he had downloaded from compromised machines.

I was particularly interested in information about the second faculty member in the school, seeing as how this could answer the question about how he got the original emails that were sent to the faculty member and secretary. I contacted the second faculty member to come over and look at the files that were stored on Chang's machine to see if he recognized any of them. When he arrived, he identified all of the files as past assignments or midterms as well as the upcoming final exam.

Day Six

The interview with the attacker revealed a great deal of information about the entire sequence of events, which actually began with the attack described in the earlier section of this paper. The events on days one through five actually occurred after the initial attack on the second instructor's machine. Apparently, the attacker's ego was bolstered by his ability to compromise the first machine and it led him to believe that he could compromise all of his instructors' systems. After compromising the first machine, he downloaded the final exams and then used the keystroke logging functionality in NetDevil to capture the professor's email password. He then used this email password to obtain the messages which

© SANS Institute 2003, Author retains full rights.

were then used in the subsequent attempts to compromise the machines of the other faculty member and his secretary.

Containment

In my opinion, the primary containment procedure for this incident was really the confiscation of the attacker's machine. Because this attack was carried out explicitly by an individual rather than by a self-propagating system of some sort, we felt that taking away the student's machine and subjecting him to police interview would put a stop to his activities. In addition. because we didn't have any kind of firewall on our network, there wasn't any way to block the attacker from connecting to those machines again.

Internally, as soon as we learned that the instructor's machine had been compromised, we walked with him down to his office to immediately disconnect his machine from the network, shut it off, and bring it back to our offices.

Eradication

The process of eradication for the single compromised machine in my school was very simple. As soon as the instructor determined that the files on the attacker's machine were copied from his system, we took his machine away and gave him a brand new one with our standard image. Our staff then copied all of his documents and local mail folders onto a Zip disk which was then scanned for viruses with an up-to-date copy of Symantec Antivirus. After we initially submitted the trojan from the email attachment to the Symantec Antivirus Research Center, Symantec included the signature for that trojan in their next definition release a day later. Scanning the disk with this new definition would make sure that there were no remnants of the trojan on any of the files to be transferred. The data from the Zip disk was then copied onto the instructor's new machine, and his old machine was placed in storage in case it was needed in the investigation at a later date.

We chose not to perform a forensic analysis on the machine that was compromised by the student for several reasons. First, the student had already confessed to committing the crime and had explicitly outlined the steps that he took to install the trojan, download the exams, and capture the professor's email login and password. An analysis of his hard drive and discussion with the affected instructor corroborated this story. Secondly, at the point at which we would have conducted the forensic analysis, our IT staff had collectively contributed over 100 hours to this incident and it was affecting out ability to do other important work. We felt that putting the machine in storage would give us the ability to do the analysis at a later date if necessary without taking any more time from our regular duties.

The root cause of the incident was the awareness level of the instructor who was initially compromised. Had the instructor refused to insert the AutoRun disk, the student would have been unable to install the NetDevil trojan on his machine so easily. Granted, the antivirus software on the machine was horribly out of date, and there was no firewall preventing the compromised machine from being accessed from the Internet. I think that while almost all of our users knew

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

enough not to open a random attachment from a stranger, very few of them were aware of the dangers of allowing a stranger to insert foreign media into their computers.

Recovery

With respect to the compromised machine, the recovery process was its replacement with a non-compromised machine. In order to preserve the evidence for the police, we couldn't rebuild the machine in place and were obligated to provide the instructor with a "known good" machine.

One of the most important recovery steps that we took was to inform all of our users of what had taken place. We capitalized heavily on this incident, using it to remind our users to always be vigilant and to report any kind of anomalous email or computer activity to our department. We also took advantage of this opportunity to bring the users in the auxiliary building under the support umbrella of our department. They all received new hardware with the standard image and began to use our network file and print resources. As a part of our managed network, they were also automatically distributed the latest virus definitions on a daily basis. If this system were in place before the attack occurred, I don't think that it would have been successful.

We also made sure that the latest virus definitions were pushed to all of our users and double-checked that it did in fact identify all of the variants of the NetDevil trojan used by the attacker.

Lessons Learned

This incident was a real lesson to not just our school but to the entire university. While this paper only covers the portion of the attack that directly affected the machines my staff controls, the attacker compromised many other machines on campus, some containing very sensitive data.

The most important outcome from this incident was a campus-wide initiative to bolster our incident response and handling procedures, a previously ignored aspect of the charter of the campus NSG. Before our incident, each department was responsible for handling an incident on their own, contacting the police on their own, and performing their own forensic analysis. Now, there has been a significant push to centralize these functions to a certain degree. The NSG does not have the resources to handle every incident to the level of detail that we handled ours, but a working group was established on campus to determine a threshold at which this kind of investigation would occur. The NSG also has become much more closely tied to the police department, who previously had very little experience with handling computer crime cases.

The incident also highlighted and helped draw attention to some of the serious problems with security on campus. While our incident was rather devastating for the faculty involved, it paled in comparison to the other incidents that the student was responsible for. The attacker compromised the machines of several graduate student instructors in other departments. These graduate student machines are not supported by any central group on campus, primarily because they are not owned by the University. So when a graduate student

© SANS Institute 2003, Author retains full rights.

machine is compromised due to an attack from one of their students, who is liable for handling that machine and bringing it back to a known-good state? It's a difficult question to answer, because as a University we feel responsible for any personal damage done as a result of employment with a department, but there are no resources to provide computer support to the thousands of graduate student instructors on campus. In this particular incident, we were able to secure resources to rebuild these machines, but it sparked a great deal of conversation which is still ongoing about how this kind of situation will be handled in the future. All of the other departments affected by the attacker also took significant steps after the incident to heighten their own security.

Within our school, the incident taught us a number of lessons as well. Most importantly, as mentioned previously, we used this incident as an opportunity to push for bringing all of the machines in the school under our support umbrella. For various reasons, financial and political, there were a number of groups that operated within the school but outside of the scope of our support. Using horror stories from this incident, we were able to convince the people in charge of all of these other groups to capitulate and allow us to maintain and monitor their machines.

The incident also taught us the importance of having a firewall to protect our school from traffic coming from the Internet and even other parts of campus. Because of reasons discussed earlier in the paper, there was no firewall on the network to protect the compromised machine from access by a client on the Internet. Only recently have we been able to work with the campus networking group and come up with a firewall solution that will provide us with the protection that we need and still give them manageability of the network.

After taking the SANS course, I plan on coming up with an in-depth set of policies and procedures based on the six stages outlined in the course so that we can have a more consistent approach to incident handling in the future.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

References

Vgrep
http://www.virusbtn.com/resources/vgrep/index.xml

Symantec Security Response
http://securityresponse.symantec.com/avcenter/venc/data/backdoor.netdevil.html

MegaSecurity.org (screenshots and file sizes)
http://www.megasecurity.org/trojans/n/netdevil/Netdevil_all.html

McAfee Security
http://vil.mcafee.com/dispVirus.asp?virus_k=99295

Computer Associates
http://www3.ca.com/virusinfo/virus.aspx?ID=10948

Forensic imaging using Ghost
http://service1.symantec.com/SUPPORT/ghost.nsf/docid/1999110813413225

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.