An Analysis of a Vulnerability Scanning Incident for a Small Research Network

by

Mark D. Tollison

Submitted

March 05, 2001

Research Paper Index

Executive Summary

In today's world, everyone is increasingly dependent on the ability to have instant access to information. The explosion of the internet, along with wireless and broadband technologies, allow companies and individuals, unprecedented "Real Time" access to vast amounts of information. In our daily lives we are inundated with email, voice mail, facsimile, pager and other types of information. In our personal lives, we use our computers to chat with friends, listen to digitized music, make travel reservations and buy products. As Internet access costs have plummeted, corporations are using the Internet as the media of choice for corporate data and, increasingly, voice communications. Any outage in any of these systems is not only a nuisance but a major event to productivity. The Internet has and will continue to revolutionize the way business is conducted.

Unfortunately, there is a dark side to the use of the Internet. The many advantages, such as cost, openness and flexibility of this vast computer network are heavily impacted by security risks. It is a daily occurrence to read about another malicious hacker who has defaced a web site, gained unauthorized access to a large corporations information resources, or shut down an Ecommerce site via a distributed denial of service attack. The recent attack on Microsoft via a Unicode Bug or Web Server Folder Traversal Vulnerability, seemed to cause minor monetary damage to the company but this incident definitely raised credibility issues with its information security program [1].

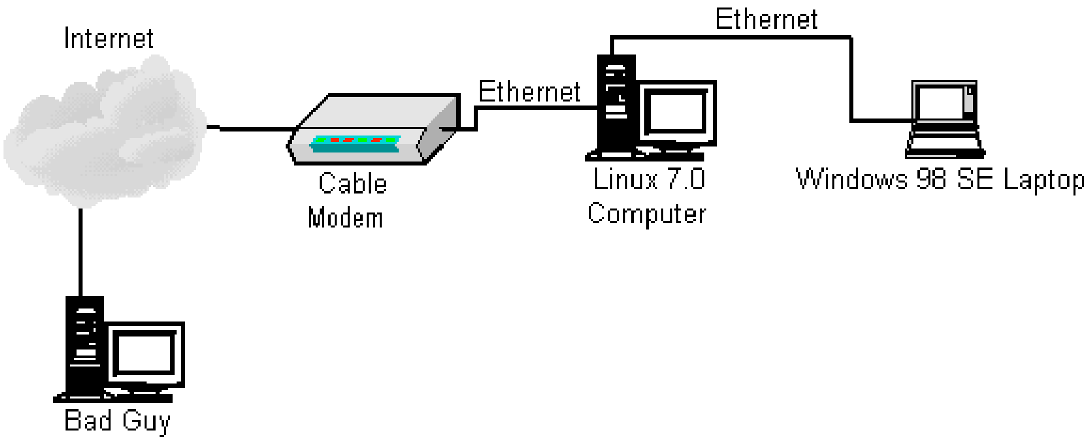
Virus or malicious code is distributed via email and forces system administrators to scan for, and if found, remove the code from these systems. A recent report stated that about 50,000 viruses had been created in the last 14 years [2]. Some of these attackers are initiated by novices who are just trying out one of the many new "Hacker Tools" available on the Internet. However, I believe a greater threat is from the dedicated individual who has a financial, theological, or personal reasons to disrupt a companies vital information flow. Espionage is a valid threat to the intellectual property of a company. There is no more cost effective way to gain a competitive advantage than to have access to your rivals information. These types of threats are just a few of the reasons that information security and incident handling is becoming increasingly important. Just as we lock our homes and offices to keep out intruders, we must protect our vital computing resources from unauthorized access. Unfortunately, the tools and methods by which malicious users obtain unauthorized access to these vital computer systems are rapidly evolving. Even with the most effective Intrusion Detection Systems (IDS) and administration programs, it is very likely that a security incident will occur.

The focus of this research paper is to educate the reader about the six steps of incident handling. Specifically, I will detail one such incident as it applies to a small computer network used in a research environment.   Also, I will provide some reference material on one open source tool, Snort [3], and how it was used to record information about the incident. In addition, I will present some background information on one widely used vulnerability scanning tool, Nmap [4].

Index

Network Diagram

The diagram detailing the small research computer network used in this discussion of incident handling is shown in Figure 1. This network consists of two computers, a gateway node running Red Hat Linux 7.0 and a second node running Microsoft Windows 98 Second Edition. The gateway node has dual network interface cards and is running the Snort network intrusion detection system. Information on Snort is included as an appendix to the document. This network was established to allow for the evaluation of various intrusions detection systems (IDS) and vulnerability tools. For this project, a known vulnerability program was used to simulate an incident. Since this system was used as a "test bed", no proprietary or critical files were stored on this system. The latest software patches were applied to each of the operating systems.



Index

Incident Background

On March 05, 2001 during a routine evaluation of SNORT alert files, the following information was found.

```
[**] ICMP Unknown Type [**]
03/05-20:22:21.995918 0:60:97:8A:F5:31 -> 0:D0:58:E3:E5:54 type:0x800
len:0x3C
24.88.220.174 -> 24.88.217.10 ICMP TTL:47 TOS:0x0 ID:37433 IpLen:20
DgmLen:28
```

```
Type:8  Code:0  ID:6331   Seq:0  ECHO

[**] ICMP Redirect (for Network or Subnet) [**]
03/05-20:22:22.033537 0:D0:58:E3:E5:54 -> 0:60:97:8A:F5:31 type:0x800
len:0x46
10.225.128.1 -> 24.88.220.174 ICMP TTL:255 TOS:0x0 ID:54346 IpLen:20
DgmLen:56
Type:5  Code:0  REDIRECT

[**] ICMP Unknown Type [**]
03/05-20:22:22.039691 0:D0:58:E3:E5:54 -> 0:50:BA:A4:EA:97 type:0x800
len:0x3C
24.88.220.174 -> 24.88.217.10 ICMP TTL:46 TOS:0x0 ID:37433 IpLen:20
DgmLen:28
Type:8  Code:0  ID:6331   Seq:0  ECHO

[**] ICMP Unknown Type [**]
03/05-20:22:22.131363 0:50:BA:A4:EA:97 -> 0:D0:58:E3:E5:54 type:0x800
len:0x2A
24.88.217.10 -> 24.88.220.174 ICMP TTL:255 TOS:0x0 ID:3476 IpLen:20
DgmLen:28
Type:0  Code:0  ID:6331  Seq:0  ECHO REPLY

[**] ICMP Unknown Type [**]
03/05-20:22:22.140335 0:D0:58:E3:E5:54 -> 0:60:97:8A:F5:31 type:0x800
len:0x3C
24.88.217.10 -> 24.88.220.174 ICMP TTL:254 TOS:0x0 ID:3476 IpLen:20
DgmLen:28
Type:0  Code:0  ID:6331  Seq:0  ECHO REPLY

[**] spp_portscan: PORTSCAN DETECTED from 24.88.220.174 (THRESHOLD 3
connections exceeded in 0 seconds) [**]
03/05-20:22:22.644348
[**] spp_portscan: portscan status from 24.88.220.174: 246 connections
across 1 hosts: TCP(246), UDP(0) [**]
03/05-20:22:28.672156
[**] spp_portscan: portscan status from 24.88.220.174: 243 connections
across 1 hosts: TCP(243), UDP(0) [**]
03/05-20:22:34.602154
[**] spp_portscan: portscan status from 24.88.220.174: 288 connections
across 1 hosts: TCP(288), UDP(0) [**]
03/05-20:22:40.193477
[**] spp_portscan: portscan status from 24.88.220.174: 205 connections
across 1 hosts: TCP(205), UDP(0) [**]
03/05-20:22:46.180249
[**] spp_portscan: portscan status from 24.88.220.174: 387 connections
across 1 hosts: TCP(387), UDP(0) [**]
03/05-20:22:52.015962
[**] ICMP Destination Unreachable (Undefined Code!) [**]
03/05-20:22:57.149981 0:50:BA:A4:EA:97 -> 0:D0:58:E3:E5:54 type:0x800
```

```
len:0x172
24.88.217.10 -> 24.88.220.174 ICMP TTL:255 TOS:0xC0 ID:5031 IpLen:20
DgmLen:356
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
24.88.220.174:59282 -> 24.88.217.10:1 UDP TTL:50 TOS:0x0 ID:30097
IpLen:20 DgmLen:328
Len: 308
** END OF DUMP

[**] ICMP Destination Unreachable (Undefined Code!) [**]
03/05-20:22:57.189377 0:D0:58:E3:E5:54 -> 0:60:97:8A:F5:31 type:0x800
len:0x172
24.88.217.10 -> 24.88.220.174 ICMP TTL:254 TOS:0xC0 ID:5031 IpLen:20
DgmLen:356
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
24.88.220.174:59282 -> 24.88.217.10:1 UDP TTL:50 TOS:0x0 ID:30097
IpLen:20 DgmLen:328
Len: 308
** END OF DUMP
```

This alert data did not show any specific threat but indicated that the research network was being scanned. Port scans, though not as serious as other types of incidents, are still valid threats and should be reported. A scan attempt could be the precursor to a full scale attack on this or other computer networks and must be handled appropriately. Since the extent on the activity was unknown, this event was treated as a possible security breach and compromise of the network. A detailed analysis of this incident using the six steps of incident handling is given below. The six steps of incident handling are:

- Preparation
- Identification
- Containment
- Eradication
- Recovery
- Follow up / Lessons Learned

Index

1.0 Preparation

As with many activities, preparation and planning are vital to the success of the incident handling process. I believe that one of the most important lessons taught by the SANS GIAC Level Two Incident Handling and Hackers Exploits Course is that spending enough time early in the process is vital to expediting the investigation and recovery from an incident. For this research network, a few basic polices and steps have been taken to prepare for an incident. These steps fall into the following basic categories.

- Background and Initial Planning
- System Backups
- Tools and Reporting

Index

1.1 Background and Initial Planning

Since this is a small research network, not directly connected to any mainstream corporate resources, a small incident handling team has been established. The team is a cross functional organization consisting of management, technical, security and help desk personnel. An experienced and trained team will allow for a thorough investigation and resolution of the incident.

Also, this system is in operation periodically but monitored frequently for any unusual activity. Currently this research network is only used by a small team of individuals.

Following the course material and the guidance in the "Incident Handling Step by Step" guide, [5] the following safeguards were established.

- A warning banner was added to both the computers and is displayed during system login and when certain services are accessed. The Banner reads as follows, "USE OF THIS SYSTEM IS FOR AUTHORIZED USE ONLY. THESE RESOURCES ARE MONITORED FOR ACCEPTABLE USE. USE OF THESE COMPUTERS AND SYSTEMS ARE CONSENT FOR MONITORING."

- The latest updates and patches have been installed for the operating systems. The Redhat Linux 7.0 Patches were obtained from the Redhat support site. The Windows 98 operating system updates were obtained from Microsoft.

- Version 1.7 of the Snort network intrusion detection system was installed on the Linux 7.0 and Windows 98 computers. This open source intrusion detection system is explained in more detail in Appendix A.

- A very simple form was developed to capture information about the incident. Eventually, I plan for the data from these incident report forms to automatically populate a database or similar data structure. This information could then be queried by some graphical or html tool for display. The form is listed in Appendix B. Also, other forms are listed in the "Incident Handling Step by Step" guide, [5]

- A communications call list and call tree was developed. Also, a PGP email network is currently being established to allow for the confidential transfer of incident reports and messages.

Index

1.2 Backups

As detailed in this and other courses, computer system backups are very important to the incident handling process. Proper system backups can save a considerable amount time in the event of an incident. This applies whether this incident is caused by a hardware failure, user error or a security compromise of a computing system or network. Typically, backups take place in two stages. The first stage is making the backup and verifying its integrity. The second stage is recovery from a system incident using the backup.

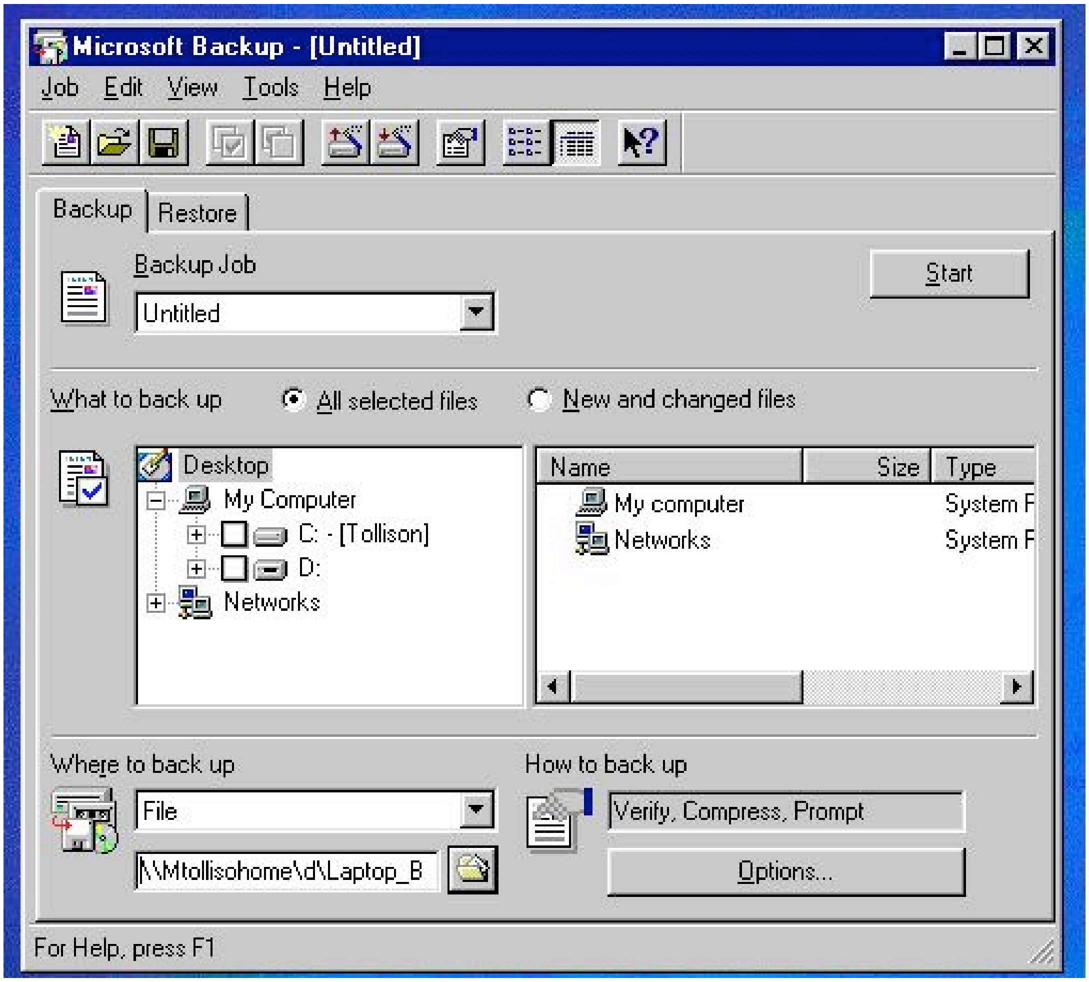
For this small research network, the ability to perform proper backups are limited. This is not due to the operating systems but because of hardware deficiencies. Neither of these computers have a large tape backup system. However the Linux 7.0 computer does have access to a CD-RW drive. Also, the Windows 98 computer is connected to an Iomega Zip removable media drive. Methods for using these hardware devices for backup are detailed in the following paragraphs.

Windows 98 System Backup

Included with Windows 98 is the standard backup tool, Microsoft Backup. Microsoft Backup allows a system administrator to save critical data and system files. In case of operating system compromise, the plan is to either replace or reformat the hard disk and reload the operating system via the original purchased media. The known uncompromised file and registry system backup would then be used to reload critical data and system files. Unfortunately, Microsoft backup does not have the capability to perform any disk images or block backups.

Microsoft backup can be accessed via the following command path:

Start > Programs > Accessories > System Tools > Backup. Once this program is started the following opening menu appears:



Backup Job: Untitled

What to back up: ● All selected files ○ New and changed files

- Desktop
  - My Computer
    - □ C: - [Tollison]
    - □ D:
  - Networks

Name	Size	Type
My computer		System F
Networks		System F

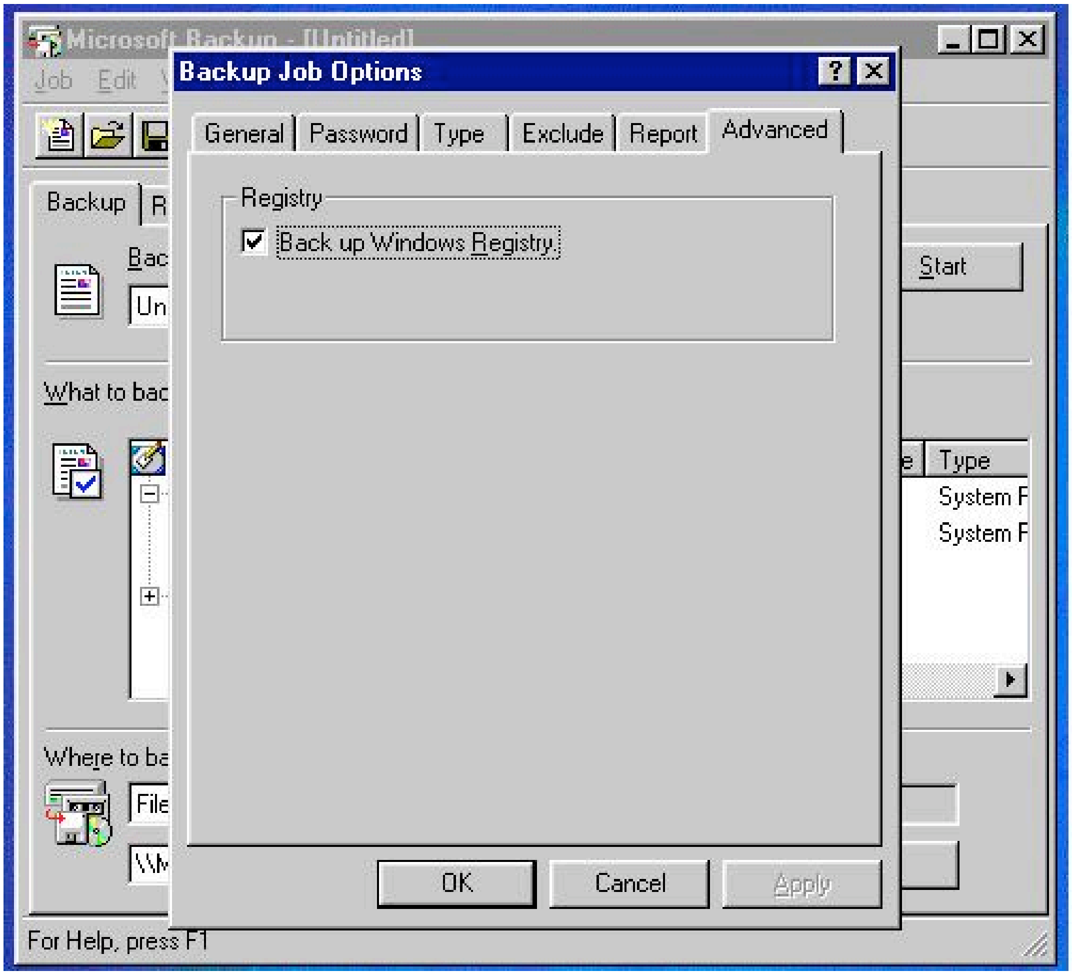
Where to back up: File

\\Mtollisohome\d\Laptop_B

How to back up: Verify, Compress, Prompt

For this research system, a few of the critical system files which should be backed up are the Windows Registry entries, Snort intrusion detection log and other select research related data files. In order to back up Windows Registry, an option must be selected from the advanced menu. This option is chosen via the following Microsoft Backup Tab path:

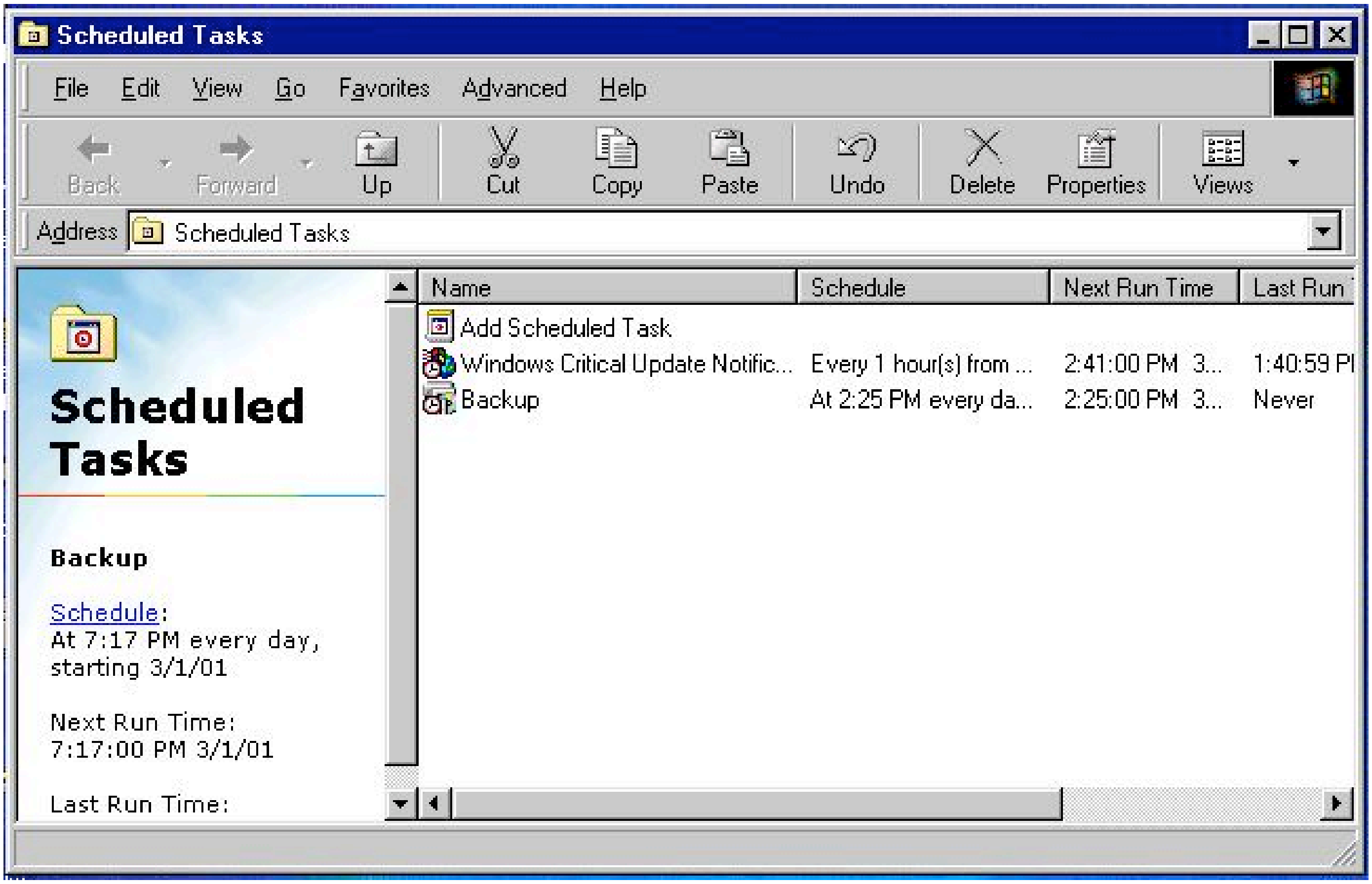
Job > Options > Advanced > menu. A screen shot showing the "Back up Windows Registry" option is shown below.



☑ Back up Windows Registry

After the backup is complete, the media is stored in a locked cabinet. Ideally, it would be preferably to make two copies of the backup and store this in another facility. The current backup scheme is for a weekly system backup with nightly storage of registry and intrusion detection logs. This backup is can be automated using the Windows scheduled tasks feature. Microsoft scheduled Tasks can be accessed via the following command path:

Start > Programs > Accessories > System Tools > Scheduled Tasks. The opening menu of this feature is shown below.


Scheduled Tasks
File Edit View Go Favorites Advanced Help
Back Forward Up Cut Copy Paste Undo Delete Properties Views
Address Scheduled Tasks
Scheduled Tasks

Name	Schedule	Next Run Time	Last Run
Add Scheduled Task			
Windows Critical Update Notific...	Every 1 hour(s) from ...	2:41:00 PM 3...	1:40:59 PI
Backup	At 2:25 PM every da...	2:25:00 PM 3...	Never

Backup

Schedule:
At 7:17 PM every day, starting 3/1/01

Next Run Time:
7:17:00 PM 3/1/01

Last Run Time:

Linux 7.0 System Backup

For this system, the current backup strategy is for the storage of Linux system and Snort intrusion detection logs files. This process has been automated using a simple script that stores these files on removable media. Other methods of storing this data, including using the local CD-RW drive would make this process more secure and less prone to intruder tampering. The addition of a large tape backup unit and separate log server would enhance this capability. Access to a tape drive would allow for use of the standard TAR, Dump and DD Linux archival functions.

Index

1.3 Tools and Reporting

It is very important have a readily available set of tools to use during incident handling. These tools should be on a media that is protected from alteration. I have created a "Read Only" CD-ROM with the following set of analysis tools for both the Linux and Windows operating systems. Some of the items include:

ping
ls
Find

ff
rm
netstat
traceroute
vi
find
lsof
who
last
lastb
NT Resource Kit

Also, we have access to other items as required during an incident investigation. Some of the items include:

Tape Recorder
Video Camera
Cellular Phone
Linux 7.0 Distribution CDs
Windows 98 Distribution CDs
License Keys
Backup Media

Reporting is a vital part of incident handling process. In this case, reporting would be to the CIRT or local law enforcement.

Index

2.0 Identification

Identification is the process by which the incident handling team investigates the symptoms, error messages, intrusion detection logs, etc., to try to pinpoint what happened or is happening to a computer or network. Did a system component fail, did the user erase critical files or was the system compromised by a malicious person. Did an incident really occur. These are a few of the many responsibilities during the identification phase of incident handling.

Also, chain of custody issues are very important here. Extreme care must be taken not to destroy valuable data and evidence. Good notes and proper handling of evidence are key issues. All evidence should be properly identified and witnessed. All evidence should be properly labelled, signed and dated. These items should be stored in a container with limited access. This will ensure that the evidence can be used in court. To fulfil the chain of custody requirements the following files were copied to a floppy disk, labelled, witnessed and stored in a locked container.

```
/home/Mark/snort/snort-1.7/logs/alert
```

```
/home/Mark/snort/snort-1.7/logs/log
/home/Mark/snort/snort-1.7/logs/xx.xxx.xx.xx
```

Also, copies of all notes and forms used during the incident handling process were stored in a sealed plastic bag with the floppy disk.

In this case, the Snort alert log showed that unusual ICMP packets and scan packets were being generated. This did not act like a denial of service event since only a few packets were sent. However, as detailed in this and other SANS courses, certain TCP/IP packets can be used to verify open ports on a system and that certain patterns are used to fingerprint systems. The scanning and fingerprinting of a network or system can be fairly serious event. Especially, since reconnaissance is possibly a precursor to an actual intrusion. To verify that the Snort alerts were associated with a scan, the following tools and commands we used to check external connections and recent user access to the Linux system.

NETSTAT -a

```
> netstat -a
```

Using the "netstat -a command" the system was scanned for the presence of any unexpected network ports or those with unknown origins. A portion of the output is shown below.

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address
State
tcp        0      0 *:X                     *:*
LISTEN
tcp        0      0 *:587                   *:*
LISTEN
tcp        0      0 *:smtp                  *:*
LISTEN
tcp        0      0 *:printer               *:*
LISTEN
tcp        0      0 *:ssh                   *:*
LISTEN
tcp        0      0 *:login                 *:*
LISTEN
tcp        0      0 *:shell                 *:*
LISTEN
tcp        0      0 *:telnet                *:*
LISTEN
tcp        0      0 *:ftp                   *:*
LISTEN
tcp        0      0 *:finger                *:*
LISTEN
tcp        0      0 *:auth                  *:*
```

```
LISTEN
tcp        0      0 *:1024                  *:*
LISTEN
tcp        0      0 *:sunrpc                *:*
LISTEN
udp        0      0 *:1025                  *:*
udp        0      0 *:1005                  *:*
udp        0      0 *:1024                  *:*
udp        0      0 *:sunrpc                *:*
raw        0      0 *:icmp                  *:*                     7
raw        0      0 *:tcp                   *:*                     7
```

None of the netstat output showed any unexpected network connections or open ports.

WHO

```
> who
```

The command "who" is used to access the var/run/utmp file which shows a current "snapshot" of users. This review did not show any unexpected user connections.

```
Mark     :0       Feb 26 06:11
Mark     pts/0    Feb 27 09:23
```

LAST

```
> last
```

The command "last" is used to access the var/log/wtmp file which lists the login-logut history. This review did not show any unexpected login connections.

```
Mark     pts/0        :0                Tue Feb 27 09:23   still logged
in
Mark     pts/0        client.xxxxx.com Mon Feb 26 15:02 - 16:51
(01:49)
Mark     pts/0        client.xxxxx.com Mon Feb 26 14:54 - 15:01
(00:07)
Mark     pts/0        :0                Mon Feb 26 06:12 - 06:15
(00:02)
Mark     :0                             Mon Feb 26 06:11   still logged
in
reboot   system boot  2.2.16-22         Mon Feb 26 06:10
(1+07:57)
Mark     :0                             Tue Feb 20 16:38 - down
(00:19)
```

```
reboot   system boot  2.2.16-22        Tue Feb 20 16:37
(00:20)
Mark     :0                            Tue Feb 20 16:35 - down
(00:00)
reboot   system boot  2.2.16-22        Tue Feb 20 16:34
(00:02)
ftp      ftpd1289     cc23985-a.assen1 Wed Feb  7 15:34 - 15:35
(00:00)
Mark     pts/0        :0               Wed Feb  7 12:51 - 07:33
(18:41)
Mark     :0                            Wed Feb  7 12:51 - down
(18:42)
reboot   system boot  2.2.16-22        Wed Feb  7 12:49
(18:44)
Mark     :0                            Mon Feb  5 19:11 - down
(00:05)
Mark     pts/0        :0               Mon Feb  5 07:06 - 07:07
(00:00)
Mark     :0                            Mon Feb  5 07:05 - 07:07
(00:01)
reboot   system boot  2.2.16-22        Sun Feb  4 22:15
(21:00)
Mark     :0                            Sun Feb  4 17:36 - down
(00:05)
reboot   system boot  2.2.16-22        Fri Feb  2 06:56
(2+10:45)
Mark     pts/0        :0               Thu Feb  1 10:35 - 11:06
(00:30)
Mark     pts/0        :0               Thu Feb  1 09:34 - 10:35
(01:00)
Mark     :0                            Thu Feb  1 09:34 - down
(01:32)
reboot   system boot  2.2.16-22        Thu Feb  1 09:33
(01:33)
Mark     pts/0        :0               Thu Feb  1 09:06 - 09:31
(00:25)
Mark     pts/0        :0               Thu Feb  1 08:42 - 09:06
(00:24)
Mark     :0                            Thu Feb  1 08:39 - down
(00:52)
reboot   system boot  2.2.16-22        Thu Feb  1 08:38
(00:53)
Mark     pts/1        :0               Thu Feb  1 07:33 - 07:33
(00:00)

wtmp begins Thu Feb  1 07:33:22 2001
```

LASTB

```
> lastb
```

The command "lastb" is used to access the var/log/btmp file which lists the bad login history. A sample of a btmp file is shown below.

```
mark     pts/0        client.XXXX.com Mon Feb 26 15:02 - 15:02
(00:00)
root***  pts/0        client.XXXX.com Mon Feb 26 15:02 - 15:02
(00:00)
root     pts/0        client.XXXX.com Mon Feb 26 15:02 - 15:02
(00:00)
root     pts/0        client.XXXX.com Mon Feb 26 15:01 - 15:01
(00:00)
root     pts/0        client.XXXX.com Mon Feb 26 15:01 - 15:01
(00:00)
root     pts/0        client.XXXX.com Mon Feb 26 15:01 - 15:01
(00:00)
Mark     pts/1        client.XXXX.com Tue Jan 30 14:42 - 14:42
(00:00)
mark     pts/1        client.XXXX.com Tue Jan 30 14:42 - 14:42
(00:00)

btmp begins Tue Jan 30 14:42:44 2001
```

A review of the btmp output did not indicate that any unexplained access attempts had occurred.

A review of these files did not show that any unexpected services were connected to the Linux computer. No unknown individuals or services had logged in to the system. So far the system did not show any indication of an intrusion or compromise. One utility, Lsof, could have been used in the before preparation stage to perform a "Poor Man's Tripwire" of the system. This was not done earlier so this type of data was not available for analysis.

Index

3.0 Containment

Containment is the process by which the incident handling team decides what steps need to be taken to limit exposure to the problem. If a laptop computer is infected with a virus, it probably should be isolated from the rest of the corporate or research network. If a network has been compromised and is flooding other systems via a Denial of Service (DOS) attack, then the team might decide to disconnect the outgoing connection to the internet. Many possibilities and options are possible. Use the "Incident Handling Step by Step" guide, [5] as a reference.

For this event, the Snort alert logs and the identification activity did not show any compromise of the Linux system. If evidence of a comprise was suspected, this is the stage when a decision would be made whether to disconnect the computer or continue to gather evidence. Using a laptop or other known good binaries, a backup of the system would be made. One objective during this stage is to collect as much evidence as reasonable and contain the problem.

Another step that should be taken during this stage of incident handling is that the Linux system passwords should be changed. Even though this incident did not indicate any possible compromise of the research network, the passwords were changed to ensure integrity of the system.

Before moving forward, a decision was made to further evaluate the Internet Control Message Protocol (ICMP) messages as reported by the Snort intrusion detection system. As reported by "Using TCP/IP" [7] ICMP messages are of the following types:

0 - Echo Reply
3 - Destination Unreachable
4 - Source Quench
5 - Redirect
8 - Echo
11 - Time Exceeded
12 - Parameter Problem
13 - Timestamp
14 - Timestamp Reply
15 - Information Request
16 - Information Reply

These various messages are used to provide information from between gateways or hosts when an error in datagram processing has occurred. The ICMP Echo Reply and Echo messages are a useful part of the "Ping" command which is used during network troubleshooting to determine if a host is responding. Unfortunately, the ICMP commands can be used maliciously to map networks or for other malicious uses, an example is the "Ping of Death." In our case, the ICMP messages were being manipulated to assist with the scanning of the research network.

Index

4.0 Eradication

Eradication is the process by which the incident handling team decides how and when to get rid of the problem. If a email virus has infected the main server, then the deletion of the offending files is might be required. If an operating system has been corrupted then installation of a new drive and a reload of the system is needed. Many variations are possible depending on the type of incident. The goal in this stage of the incident handling process is to eliminate the problem while ensuring that the same incident does not reoccur. Good forensics at this point is very valuable. Also, vulnerability testing can be a great asset in determining if the problem has been corrected. Again, use the "Incident Handling Step by Step" guide, [5] as a reference and a tool.

For this incident, although the Snort alert logs and the identification activity did not show any compromise of the Linux computer and associated network, it was decided to perform a vulnerability analysis of the system. The vulnerability analysis was performed using the Nmap program. A full description of Nmap is included in Appendix C.

Using the Nmap vulnerability scanner, version 2.54BETA7, the following information was determined about the Linux 7.0 system.

```
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Host XXX.triad.rr.com (XX.XX.XXX.XX) appears to be up ... good.
Initiating SYN Stealth Scan against YYY.triad.rr.com (YY.YY.YYY.YY)
Adding TCP port 22 (state open).
Adding TCP port 111 (state open).
Adding TCP port 79 (state open).
Adding TCP port 6000 (state open).
Adding TCP port 1024 (state open).
Adding TCP port 113 (state open).
Adding TCP port 25 (state open).
Adding TCP port 514 (state open).
Adding TCP port 23 (state open).
Adding TCP port 21 (state open).
Adding TCP port 515 (state open).
Adding TCP port 513 (state open).
Adding TCP port 587 (state open).
The SYN Stealth Scan took 36 seconds to scan 1534 ports.
For OSScan assuming that port 21 is open and port 1 is closed and
neither are firewalled
Interesting ports on YYY.triad.rr.com (YY.YY.YYY.YY):
(The 1521 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
25/tcp     open        smtp
79/tcp     open        finger
111/tcp    open        sunrpc
113/tcp    open        auth
513/tcp    open        login
514/tcp    open        shell
```

```
515/tcp    open        printer
587/tcp    open        submission
1024/tcp   open        kdm
6000/tcp   open        X11

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2419956 (Good luck!)

Sequence numbers: FC367BFB FBEC96A7 FC6982BC FBE2BD99 FC18BE4D
FBECA0AA
Remote operating system guess: Linux 2.1.122 - 2.2.16

Nmap run completed -- 1 IP address (1 host up) scanned in 41 seconds
```

The vulnerability testing showed that the Linux computer has many services active in its default configuration. Many of these services are not essential, smtp, finger, etc., and should be disabled. It is very probable that the attacked obtained this same information and could launch an attack against any unpatched and vulnerable services.

Improving defences is another important element during this stage of information handling. One important feature that has not been enabled on the Linux computer is the TCP Wrappers utility. This utility provides the means for additional logging and filtering of configured services and ports. TCP Wrappers will be a great addition to the monitoring provided by Snort.

Index

5.0 Recovery

Recovery is the process by which the incident handling team decides how to return the affected system to an operational state. If a new hard drive was installed into a system, the operating system, drivers and other files must be reinstalled or reloaded from a known good backup. This could be a time consuming task depending on the severity of the incident and the type of recovery needed. An important consideration during this phase of incident handling is to ensure that the restoral does not use compromised code. The legitimacy of the incident handlers abilities and skills would questioned if the same vulnerability or malicious code was reintroduced into the system. Again, use the "Incident Handling Step by Step" guide, [5] as a reference.

In this example, since no eradication activity was needed the system recovery was not a factor. I emphasize that this will not be true for most recovery efforts.

However, the system will be closely monitored for any new activity that would warrant a change to this policy. The Snort intrusion detection system was restarted to monitor the network for any other scan or intrusion attempts. The latest Snort Ruleset, dated 03/01/2001 was loaded and configured for use.

Index

6.0 Follow Up / Lessons Learned

In this stage of incident handling, the emphasis should be to verify proper operation of the affected system and plan for the next incident handling event. An evaluation should be done to determine the strong and weak points of the effort. Improvements should be discussed, evaluated and implemented. Complacency is sometimes hard to fight and overcome. However, the rapid pace of technology and the complexity of the hacker tools available, make this a continual process of upgrade and revaluation.

Working on this simulated incident made me aware of a number of factors. First, I now better understand the complexity involved in determining what happens during an incident. I must become better educated and more experienced with incident handling. I plan to to implement a trip wire system using the Lsof tool to be used as a verification tool during the identification phase.

Second, I must implement a better schedule and system for backups. This exercise really showed me how vulnerable even a small research system is to imperfect data backup schemes.

Third, a great deal of effort would be required to expand the scope of this incident handling to a larger network. I better understand how much time and effort is required to effectively establish and maintain this competency.

For this operating system, I was fairly limited in the type of backups available. I don't have access to any newer backup tools or system utilities such as Norton Utilities Suite of backup and recovery tools.

Index

Conclusion

Even for a small research network, an incident handling situation can be a very detailed, complicated, time consuming evolution. However, using good judgement and following the six steps of incident handling can make make an unpleasant and stressful situation successful. From my perspective, the the goal of incident handling is to efficiently analyse and recover from an incident. Many other events, fires, floods, malicious damage, etc., cause problems similar to a security incident. These aren't incidents caused by unauthorized access to a computing system but can be handled in many of the same ways. Being prepared and learning how to better harden the various systems against attack are vital lesson of incident handling.

Index

References

[1] Delio,Michelle. "Hackers Crack Into MS System." 27 October 2000.
URL: http://www.wired.com/news/culture/0,1284,39778,00.html (26 February 2001)

[2] "Virus Woes Lead to New Tactics." Investors Business Daily. 30 November 2000.

[3] Roesch, Martin "Snortorg Latest News" URL: http://www.snort.org/snortnews/news.asp
(26 February 2001)

[4] Fyodor. "General Information." URL: http://www.insecure.org/nmap/index.html#intro
(26 February 2001)

[5] "Computer Security Incident Handling Step By Step." The SANS Institute. May 1998.

[6] Roesch, Martin "What is Snort?" URL: http://www.snort.org/what_is_snort.htm
(26 February 2001)

[7] Ray, John. "Using TCP/IP." January 1999. QUE Corporation.

Index

Links of Interest

Redhat Linux 7.0 Updates - http://www.redhat.com/support/errata/rh7-errata-security.html

Microsoft Windows Updates - http://www.windowsupdate.microsoft.com

Snort 1.7 Source Code - http://www.snort.org/Files/snort-1.7.tar.gz

Snort-1.7-win32-source.zip - http://download.datanerds.net/source/snort-1.7-win32-source.zip

Snort-1.7-win32-static.zip -

http://download.datanerds.net/binaries/snort-1.7-win32-static.zip

Nmap Downloads - http://www.insecure.org/nmap/index.html#download

Index

APPENDIX A.

Snort, "Lightweight Network Intrusion Detection System"

The following information from the Snort web site [6] describes the system.

"Snort is a libpcap-based [PCAP94] packet sniffer and logger that can be used as a lightweight network intrusion detection system (NIDS). It features rules based logging to perform content pattern matching and detect a variety of attacks and probes, such as buffer overflows [ALE96], stealth port scans, CGI attacks, SMB probes, and much more. Snort has real-time alerting capability, with alerts being sent to syslog, Server Message Block (SMB) "WinPopup" messages, or a separate "alert" file. Snort is configured using command line switches and optional Berkeley Packet Filter [BPF93] commands. The detection engine is programmed using a simple language that describes per packet tests and actions. Ease of use simplifies and expedites the development of new exploit detection rules. For example, when the IIS Showcode[IISBT99] web exploits were revealed on the Bugtraq mailing list [BTQ99],Snort rules to detect the probes were available within a few hours." [5]

Also, Snort has an extensive set of features and command line options. These are documented fully in the accompanying "Readme" file which is included in the Snort distribution. A sample of the basic command line options are summarized in the following table.

snort -[options] <filters>

COMMAND LINE OPTION	DESCRIPTION
-A <alert>	<alert> mode can be either Full,Fast or None. Type of alerting to the alert file Full mode, normal mode. Fast mode - Only Timestamp,Message, IPs, and P. None - No alerting.
-a	Display ARP packets

-b	Log packets in tcpdump format. FASTEST OPERATION
-c <cf>	Use configuration or rules file <cf>.
-C	Dump the ASCII characters in packet payloads only. NO HEXDUMP
-d	Dump the application layer data
-e	Display/ and log the layer 2 packet header data
-F <bpf>	Read BPF filters from file <bpf>. Can be used for complex filters
-i <if>	Use network interface <if>.
-l <ld>	Log packets to directory <ld>.
-N	Turn off logging. Alerts still function normally.
-p	Turn off promiscuous mode sniffing.
-r <tf>	Read the tcpdump-generated file <tf>.
-v	Verbose output to console. Limited use. Will caused slowdown and possible packet loss.
-V	Show the version number and exit.
-?	Show the usage summary and exit.

Two very valuable features of Snort are the support for filters and rules. Filters allow the program to be tailored for specific needs such as monitoring a single host computer or subnet. The use of rules allow Snort to be used as a fully functional network intrusion detection system. Through third party add-ons, the system can be enhanced to make logs reviews and administration easier. For this small research network, Snort is a cost effective method of monitoring system access.

Index

APPENDIX B.

Incident Handling Form

The following is a simple form for use during incident handling.

Incident Handling Form
(Sensitive Information Once Completed)

Initial Incident Contact	

Information	.
Date:	Time:
System:	Security Lead Investigator:
Initial Symptoms:	Analysis Description and Justification:
Incident Closeout	.
Date:	Time:
Final Incident Analysis:	Tools Used:
CIRT/Authorities Notified? Yes or No	.
Report Filed? Yes or No	.
Incident Closed By: Name:	.
Signature:	Date:

Index

APPENDIX C.

Nmap

The following information from the Nmap web site [4] describes the functions of the program.

nmap is a utility for port scanning large networks, although it works fine for single hosts. The guiding philosophy for the creation of nmap was TMTOWTDI (There's More Than One Way To Do It). This is the Perl slogan, but it is equally applicable to scanners. Sometimes you need speed, other times you may need stealth. In some cases, bypassing firewalls may be required. Not to mention the fact that you may want to scan different protocols (UDP, TCP, ICMP, etc.). You just can't do all this with one scanning mode. And you don't want to have 10 different scanners around, all with different interfaces and capabilities. Thus I incorporated virtually every scanning technique I know into nmap. Specifically, nmap supports:

Vanilla TCP connect() scanning,
TCP SYN (half open) scanning,
TCP FIN, Xmas, or NULL (stealth) scanning,
TCP ftp proxy (bounce attack) scanning
SYN/FIN scanning using IP fragments (bypasses some packet filters),
TCP ACK and Window scanning,
UDP raw ICMP port unreachable scanning,
ICMP scanning (ping-sweep)
TCP Ping scanning
Direct (non portmapper) RPC scanning
Remote OS Identification by TCP/IP Fingerprinting, and
Reverse-ident scanning.
nmap also supports a number of performance and reliability features such as dynamic delay time calculations, packet timeout and retransmission, parallel port scanning, detection of down hosts via parallel pings. Nmap also offers flexible target and port specification, decoy scanning, determination of TCP sequence predictability characteristics, and output to machine parseable or human readable log files.