Unauthorized Access via the IRIX Telnetd Exploit in a College Computer Laboratory

Advanced Incident Handling and Hacker Exploits

Practical

Dennis McGrath
November 21, 2000

Executive Summary

An incident involving unauthorized access in a college computer laboratory is described in the context of the six stages of incident handling. The incident involved several SGI workstations and a file server. Very little preparation preceded the incident as security in the laboratory was not considered a high priority. No procedure or response team existed. The incident was identified when a missing syslog file on the file server triggered a review of syslogs from the workstations attached to the server. The review showed an aborted telnet session followed by a remote login (using telnet) into a user account called dos1. Only one day had elapsed between the initial intrusion and the detection of the incident. Further investigation showed that dos1 and dos2 accounts had been created on a workstation using a recently-identified buffer overflow exploit of the IRIX telnetd daemon. The bogus user accounts were used to create similar accounts on the file server and six other workstations by exploiting the trust relationships and NFS file sharing arrangement employed in the laboratory. The incident was contained by pulling the network connections from each machine. A potentially malicious application was discovered on some workstations (called "fam"), but the application did not execute properly due to an architecture incompatibility. A mysteriously-open port 31337 was identified as an indication of a possible Trojan horse running on the server, but the application associated with the open port was not identified. The effects of the attack were eradicated after user data was dumped to a neutral AFS file server by wiping the disks completely clean and restoring the operating system. User data was then restored from the AFS server. Several upgrades make the recovered system more secure than it was before the attack. The recovered system does not allow remote access with plain text applications (including telnetd), but instead requires secure shell (SSH) and secure ftp. The NFS file sharing system and the associated trust relationships have been replaced by an AFS file sharing system that uses Kerberos for user authentication. Clearly, the rapid identification of the incident minimized the damage, but luck played a bigger role in that identification than did thorough preparation. The system administrators clearly dodged a bullet and have taken steps toward securing the network from future attack, but clear procedures should be published for the handling of future incidents.

Introduction

On October 17, 2000 a computer laboratory at a small but picturesque New England college was compromised by unauthorized access. The incident was discovered the following day, and steps were taken to eradicate the effects of the incident before extensive damage was done. This report summarizes the incident from the perspective of the six-step incident handling process which includes preparation, identification, containment, eradication, recovery, and follow-up. The host names, IP addresses, and names of individuals have been sanitized for this report to prevent unnecessary disclosure of data to any would-be hackers.

The incident involved eight machines, all on the same subnet, all manufactured by SGI, all running the same operating system (IRIX 6.5.8). They included several O2's, Indigo, and Indy workstations and an Origin 200 file server. Table 1 summarizes the host and types, and figure 1 shows the flat network configuration.

Host Name	Model	Type
Sloth	O2	Workstation
Lust	O2	Workstation
Anger	O2	Workstation
Greed	Indy	Workstation
Pride	Indy	Workstation
Envy	Indigo	Workstation
Gluttony	Indigo	Workstation
Lucifer	ORIGIN 200	File Server

Table 1

The file server was a recent addition to the network. Previously, file sharing via NFS was used extensively to give users access to disk storage. NFS continued to be used after the addition of the file server, and trust relationships existed between Lucifer and the seven client workstations. There were other non-SGI workstations on the same subnet, but they were not affected by the incident.

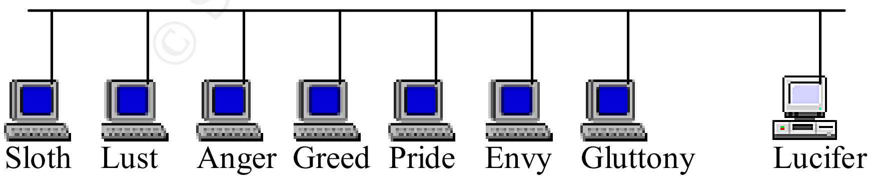

Figure 1

Preparation

No written security policy existed prior to the incident at the laboratory. As a college lab with many transient users, security was arguably lax. System administrators were given freedom to establish and enforce their own security policies. New system administrators are not necessarily given training in network security. No incident response team existed, nor was any incident handling procedure in place. No "jump kit" for rapid system recovery existed. There were not, nor are there yet any clear campus-wide guidelines on the handling of potentially criminal cases of unauthorized access. Lacking any guidelines, system administrators tend to treat incidents with an emphasis on containment and recovery rather than preservation of forensic evidence for investigation or prosecution.

Several months before the incident, the system administrator for the lab (Abraham) moved to a new assignment and handed responsibility over to a new system administrator (Isaac). Abraham conducted an informal security audit using "Saint" at the end of his tenure as system administrator after attending a network security class. The audit did not indicate any obvious vulnerabilities.

Banners for new login sessions consisted of a "welcome" message, an identification of the operating system version, and a URL for a web page containing system administration information. The message did not contain warnings about unauthorized usage. User authentication for login and telnet sessions was "plain text", non-encrypted. No firewall existed, but there were recent discussions about adding one to the network configuration. No intrusion detection systems were employed.

Identification

On October 18, Isaac noticed a missing log file on the file server (Lucifer). He called Abraham to ask advice on re-boot and restoration of the file, and Abraham suspected that the missing file might indicate unauthorized activity. An examination of the user files (/etc/passwd and /etc/shadow) showed the recent addition of two suspicious user accounts: dos1 and dos2. The dos1 account had normal user privileges, while the dos2 account had root privilege. No user accounts with those names had been assigned by the system administrator. A review of the syslog files of the workstations associated with the server revealed that a telnet session occurred on Sloth on October 17 in which a user logged in as dos1.

```
Oct 17 01:20:27 6E:Sloth login[131225]: ?@blah.ct.home.com as dos1
Oct 17 01:22:27 6E:Sloth login[133202]: ?@blah.ct.home.com as dos1
Oct 17 01:24:25 6E:Sloth login[113754]: failed: ?@blah.ct.home.com as dos1
Oct 17 01:24:31 6E:Sloth login[113754]: ?@blah.ct.home.com as dos1
```

The same syslog showed that several minutes prior to the dos1 telnet session an aborted telnet session which appears to contain garbage:

```
Oct 17 01:18:54 5B:Sloth overly long syslog message detected, truncating
Oct 17 01:18:54 0F:Sloth telnetd[133228]: ignored attempt to setenv(_RLD,     ^?D^X^\
^?D^X^^   ^D^P^?^?$^B^Cs#^?^B^T#d~^H#e~^P/d~^P/`~^T#`~^O^C^?^?L/bin/sh
```

Abraham identified this earlier session as a likely buffer overflow exploit of the telnet daemon (telnetd) in the IRIX, the SGI operating system. The telnetd exploit was identified by the Last Stage of Delirium group in August, 2000. The exploit was posted to Bugtraq, and SGI issued an alert in September acknowledging the security hole. According to the SGI security alert, all IRIX versions from 5.2 to 6.5.9 are vulnerable to this attack. IRIX 6.5.10 is the only version that is immune to the vulnerability without a patch. Patches are available for 6.5.x versions of the operating system. The advisory warns that "A local user account on the vulnerable system is not required in order to exploit telnetd daemon. The telnetd daemon can be exploited remotely over an untrusted network. The exploitable buffer overflow vulnerability can lead to a root compromise."

A detailed description of the exploit can be found at http://msgs.securepoint.com/cgi-bin/get/bugtraq0008/152.html. The following excerpt from that site describes the basic idea behind the exploit:

> The vulnerability we've found belongs to the most recently discussed class of the so-called "format bugs". IRIX telnetd service upon receiving the IAC-SB-TELOPT_ENVIRON request to set one of the _RLD family environment variables calls the syslog() function with a partially user supplied format string. The syslog message that is generated upon detecting such an attempt is of the following format: "ignored attempt to setenv(%.32s,%.128s)".
> The strings enclosed by the setenv() brackets are adequately: variable name and variable value. If variable name/value pairs are appropriately constructed, arbitrary telnetd process image memory values can be overwritten and execution flow can be redirected to the user supplied machine code instructions.

In other words, executable byte code is sent into the input buffer of the telnet application disguised as excess user data. The excess data overflows the input buffer and stays in the stack. If the return pointer (of a subroutine or function) is also overwritten, it can be altered to point back to the stealthily-inserted machine code (see figure 2).

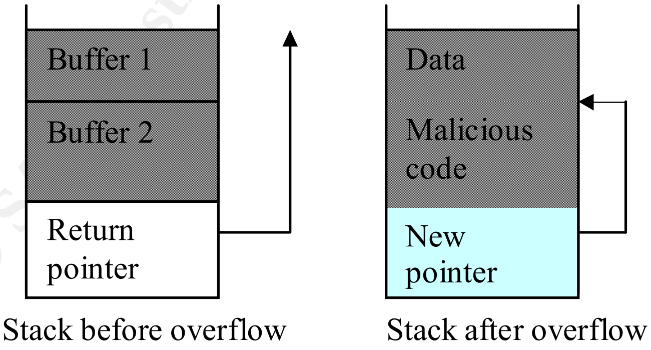

Figure 2

The Last Stage of Delirium post not only identified the exploit, they also provided a sample program that would enable anyone capable of compiling C code to initiate the attack. In other words, an attacker would not need to understand the difficult process of creating exactly the right amount of padding, inserting the malicious code, or creating the phony return pointer. A script-kiddie armed with the code would only need to identify the IP of a machine running a version of IRIX before initiating an attack .

The code inserted in this particular exploit created new user accounts. Examination of the user accounts on the other machines showed that the dos1 and dos2 accounts existed on seven other SGI machines, including the file server Lucifer. The other machines, however, did not show any signs of the buffer overflow exploit in their respective syslogs. Abraham identified the likely entry point for the attacker as the Sloth workstation.

While the other workstations used the same operating system and were therefore susceptible to the same exploit, the attacker's jump to the other hosts was probably enabled by trust relationships between the hosts and open Network File System (NFS) between the file server and the other workstations. The sulog shows that after logging on to Sloth as dos1 (an ordinary user), the attacker opened a session as dos2 (a user with root privilege). The /etc/export file listed the file server Lucifer as server that makes use of the file system on Sloth via NFS. With Lucifer listed in the .rhosts file as a trusted host, a remote shell (rsh) session from Sloth to Lucifer would allow the dos2 user to remotely log in to the server as root without password authentication. Similarly, having gained root access on the server the attacker could perform rsh to the other workstations and establish the dos1 and dos2 accounts. The likely flow of events is shown in figure 3; an initial attack on Sloth using the telnetd exploit, a jump to the file server exploiting the trust relationship, and then a similar jump to six other IRIX workstations from the server.

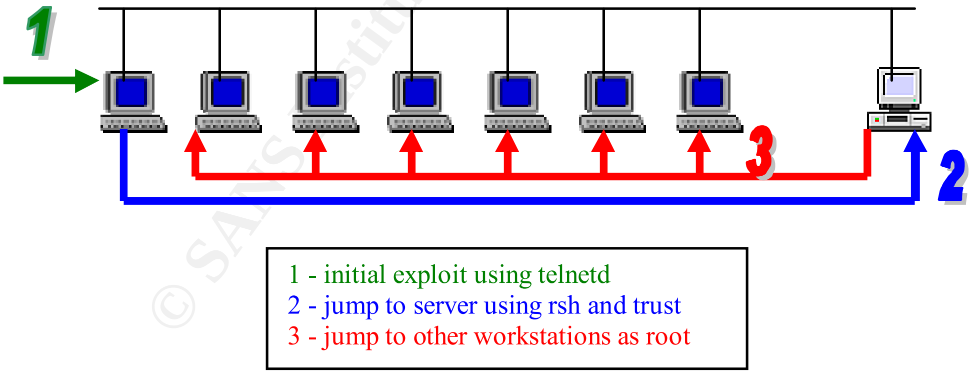
1
2
3
1 - initial exploit using telnetd
2 - jump to server using rsh and trust
3 - jump to other workstations as root

Figure 3

Containment

The ethernet cables were removed from each machine to prevent any further unauthorized access. A search for malicious code showed that a binary file called "fam" was placed in the /usr/sbin directory of every host. The root crontab had been modified to point to this file by the attacker, apparently in an effort to have this binary executed periodically. Unfortunately for the hacker, the binary was compiled under a different architecture than the attacked machines, and therefore could not be executed. The resulting error caused an error report to be mailed to root every minute. The purpose of the fam executable is not clear. There is a fam daemon available for IRIX that remotely tracks changes to a file system, but fam is included in the IRIX operating system distribution, and it seems strange that the attacker would copy a foreign fam to the machines instead of activating the existing fam on the machines. It is possible that the installed fam is a some kind of root kit that was intended to prevent the real system administrator from patching the telnetd vulnerability.

A port scan of the compromised hosts showed that port 31337 was open on Lucifer. This is a popular port for Trojan horse applications, so this was interpreted as a strong indication that an application installed by the attacker was using the port for unknown purposes. The Trojan horse most commonly associated with port 31337 is Back Orifice 2000 (BO2K), but BO2K runs only on Windows platforms so it could not be the culprit in this case. The process using the port could not be identified, but with the network cables removed the open port did not present an immediate danger.

The syslog used to identify the exploit on Sloth shows that the first telnet session using the dos1 username originated from a cable modem user (@home network). Abraham initiated a port scan of this machine and discovered several open ports, including ports for telnet and ftp. Abraham tried an ftp session to that machine which revealed a WinGate banner. WinGate is a product of Deerfield.com which advertises itself as a “Windows based Internet sharing solution”, primarily so that multiple hosts on a small network (including home computers) can use a single entry point to the internet. When configured poorly, a Wingate host can act as a wide-open proxy for anyone with internet access to use as a launching point for an attack. A telnet session to that Wingate host does not require any user authentication, and it will allow anyone to bounce anonymously to another host. So either the cable modem user was the attacker and made no attempt to disguise his/her identity, or the attacker discovered this open Wingate host and used it (and probably continues to use it) as a launch point for attacks. The latter is more likely. Abraham notified the cable modem provider about the situation but received no response.

Several IBM workstations running AIX were located on the same subnet were examined for signs of contamination, but no evidence was found to indicate that the hacker had compromised the IBM machines.

The college has a campus-wide mailing list for system administrators to exchange ideas, experience, and warnings. Prior to the incident, no one on the mailing list identified the telnetd buffer overflow exploit as a potential hazard in spite of the large number of SGI hosts running IRIX on campus. A security posting after this incident made other system administrators aware of the exploit, and at least two other IRIX machines were identified with the mysterious dos1 and dos2 user accounts.

Eradication

Having removed the machines from the network and determined the extent of the exploit, the effort focused on removing all traces of the attack without unnecessarily destroying user data. One by one, machines affected by the incident were reassigned with private subnet numbers, and data in user directories was saved on a server. Data in system directories was not saved, with the exception of the syslog files. The user accounts created by the attacker (dos1 and dos2) were obviously not restored.

After wiping the disks clean, the operating systems were restored from CD distribution. Telnetd, ftp, and other clear-text authentication methods were disabled (via inetd.conf). SSH and a secure ftp will be used in the future in place of the clear-text telnet and ftp applications. The restored systems were patched with the latest OS distribution from SGI.

Recovery

A complete system backup from October 1 was available, but restoration from that backup might have resulted in a loss of three week's worth of user data. Instead, the system administrator opted to restore user data from the user directories saved on the Andrew File System (AFS) server as described above. There was no evidence that user data was compromised or altered during the attack, and no users have reported lost data. Users were asked to recompile binaries from source code wherever possible instead of using restored binaries. Users were also asked to use change to new passwords, as the old password files may have been cracked by the attacker who had access to the /etc/shadow and /etc/passwd files for at least 24 hours.

The old NFS arrangement was scrapped, and the network is now under the control of a larger AFS cluster, and user authentication is now under the control of Kerberos. Kerberos (named for the mythological three-headed dog that guarded the entrance to Hades) is a network authentication system for use on physically insecure networks, designed to provide strong authentication for client/server applications by using secret-key cryptography. It allows entities communicating over networks to prove their identity to each other while preventing eavesdropping or replay attacks. It also provides for data stream integrity (detection of modification) and secrecy (preventing unauthorized reading) using cryptography. This substantially improves the security of the network because passwords do not pass across the network in plaintext, and encrypted passwords no longer need to be visible in the /etc/passwd or /etc/shadow directories.

Unlike NFS, which makes use of /etc/filesystems (on a client) to mount between a local directory name and a remote filesystem, AFS does its mapping (filename to location) at the server. This has the advantage of making the served filespace location independent, and in the case of this incident it might have prevented the attacker from jumping from Sloth to Lucifer and the other client workstations. Without the direct listing of Lucifer as an NFS host in the /etc/filesystem, the exploitation of trust would have been more difficult. AFS uses mutual authentication, so that both the service provider and service requester prove their identities. AFS also uses access control lists (ACLs) to enable users to restrict access to their own directories.

This new arrangement inhibits some of the autonomy of Isaac, because the AFS system which now controls Lucifer and Lucifer clients is controlled by another system administrator. Isaac can no longer create and maintain accounts and passwords on the local level. The loss of autonomy is considered a minor inconvenience when compared to the improved security of the new system.

Follow Up and Lessons Learned

Damage from this incident was minimal, but that was largely due to the rapid diagnosis and containment of the affected hosts. If the attacker had not been discovered after one day, the incident could have easily involved more extensive contamination of the network. Given the choice of user names (dos1 and dos2) it is highly likely that the attackers meant to use the network as a launch point for a distributed denial of service (DDoS) attack. The rapid detection, however, was dependent on luck more than good preparation. Fortunately, Abraham's new assignment involved extensive exposure to security loopholes and hacker exploits, and the missing log file coupled with the recent Bugtraq post triggered his suspicion. Had Issac not consulted with Abraham about the missing syslog file the unauthorized access might have continued indefinitely.

The initial Achilles heel in this system was the unpatched IRIX telnet daemon susceptible to a buffer overflow exploit, but a series of security holes made the spread of unauthorized access relatively simple for the attacker. The replacement of NFS with AFS, the use of encrypted applications for remote access, and the stronger authentication imposed by Kerberos will contribute to a much more secure network. A firewall will likely be installed in the near future. But the awareness of security holes and associated patches cannot be undervalued. Encryption, firewalls, and authentication are not sufficient in the current climate of computer security.

At no time was this incident considered a candidate for criminal investigation. As mentioned earlier, no guidelines exist on campus for determining a threshold for criminal behavior. In this case, no data was lost and no proprietary information was duplicated or stolen because no such data existed on the network. No attempts were made to preserve syslogs or disk archives for future investigation or prosecution. Nevertheless there is clearly the potential that this incident might have led to criminal activity in the form of some future, more malicious attack. Whoever created the dos1 and dos2 accounts is still at large, perhaps preparing for denial of service attacks, and a deeper investigation of this incident might help to identify the attacker.

Possible remedies for the lack of preparedness in the lab include the establishment of a campus-wide security policy, creation of an incident response team complete with jump kits (and cool T-shirts, preferably black), and clear guidelines on preservation of data for criminal investigation. System administrators need to have basic security training. In the academic environment, severe restrictions on privileges are neither feasible or warranted as they might be in a corporate or government environment, but steps can be taken to limit vulnerabilities without disrupting research and learning.

References

More information on the telnetd exploit is available at:
http://www.ciac.org/ciac/bulletins/k-066.shtml
http://msgs.securepoint.com/cgi-bin/get/bugtraq0008/152.html.
http://lsd-pl.net/files/get?IRIX/irx_telnetd

The SGI security advisory for the telnetd exploit
ftp://sgigate.sgi.com/security/20000801-01-P

SGI has a security toolbox site:
http://www.sgi.com/support/security/toolbox.html

Some of the many sites with Kerberos information:
http://www.mit.edu/afs/athena/astaff/project/kerberos/www/papers.html
http://www.contrib.andrew.cmu.edu/~shadow/kerberos.html
http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html

AFS Information is available at:
http://www.angelfire.com/hi/plutonic/afs-faq.html
http://www.msu.edu/dig/update/afs.html

WinGate information:
http://wingate.deerfield.com