Vince Kornacki
GCUX Practical Version 1.9
"Hardening Solaris 8 For Check Point FireWall-1"
November 1, 2002

Abstract
This paper describes how to harden the Sun Solaris 8 operating system for enhanced security and performance. This paper is specifically targeted at hardening Solaris for a Check Point FireWall-1 installation. The security and performance of the Solaris operating system are specifically tailored to optimize the FireWall-1 application.

Description Of The System

This section describes the server that will be hardened, including the hardware, software, and final system state.

Hardware

The hardware will be a Sun Enterprise 250 server with the following resources:

- 300-MHz UltraSPARC-II processor
- 2 GB of memory
- 18 GB hard drive
- QuadFast ethernet card (supports four separate network interfaces)

This Enterprise 250 server will provide the physical resources necessary to build a powerful firewall, and will also allow for future expansion. For more information regarding the Sun Enterprise 250 server, see the "Sun Enterprise 250 Server" website [1].

Software

The software will be the Sun Solaris 8 operating system. After proper hardening, Solaris 8 will provide strong security and performance as the underlying firewall operating system. For more information regarding the Sun Solaris 8, see the "Solaris 8 Operating Environment" website [2].

In addition, the following software packages will also be installed:

- TCP Wrappers 7.6

  TCP Wrappers will be used to provide additional access control for services such as SSH (Secure Shell).

- OpenSSH 3.4

  OpenSSH, which is an implementation of the SSH protocol, will be used to provide secure remote management and file transfer capabilities for firewall administrators.

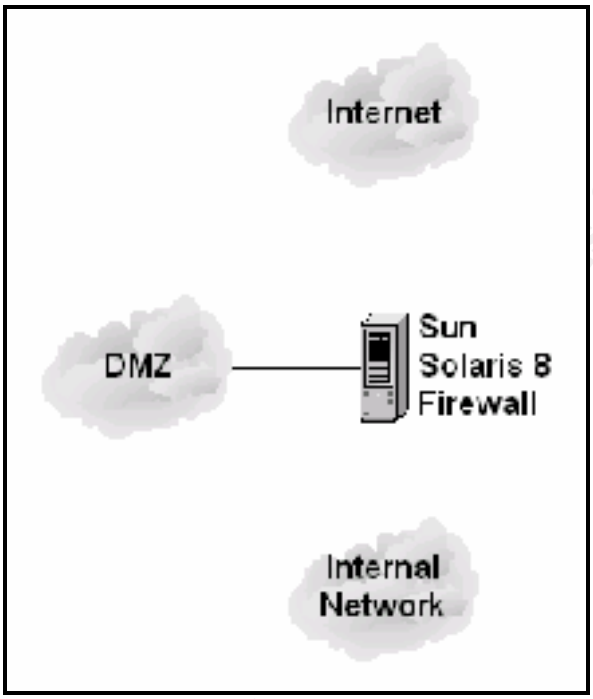
- Tripwire 2.3

  Tripwire will be used to ensure filesystem integrity, alerting firewall administrators in the event of a security breach.

These additional packages will further strengthen the security of the firewall.

Final System State

The final system state will be a hardened firewall. The firewall will run Check Point FireWall-1 4.1, and will support three network interfaces:



A description of the network interfaces follows:

- Internet Network Interface

  The Internet Network Interface will connect the firewall to the Internet. This connection is necessary to allow inbound connections from the Internet to the DMZ, and outbound connections from the Internal Network to the Internet.

- DMZ Network Interface

  The DMZ Network Interface will connect the firewall to the DMZ (Demilitarized Zone). This connection is necessary to allow inbound connections from the Internet and Internal Network to the DMZ, where publicly accessible servers such as the corporate webserver, mailserver, and DNS servers reside.

- Internal Network Interface

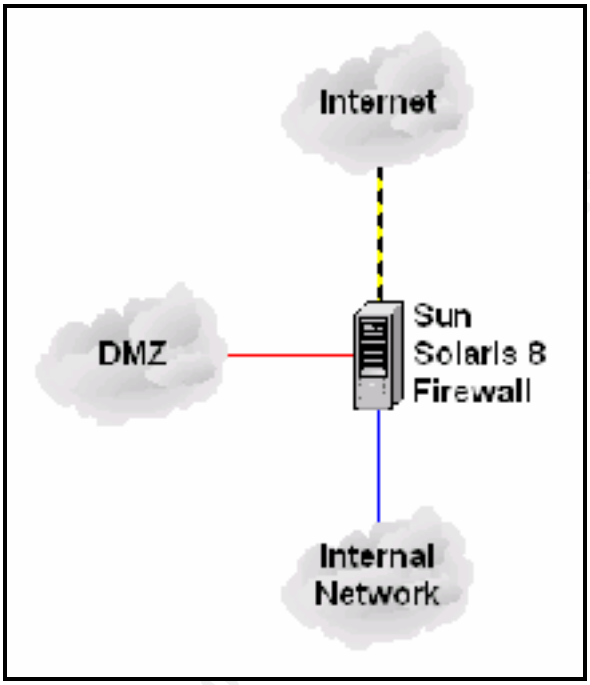
  The Internal Network Interface will connect the firewall to the Internal Network. This connection is necessary to allow outbound connections from the Internal Network to the Internet and DMZ.

The FireWall-1 application will enforce a security policy on the firewall, selectively accepting, dropping, or rejecting connections as defined by the corporate security policy. In addition, FireWall-1 will implement advanced firewall functionality such

as user authentication and NAT (Network Address Translation), and will also terminate VPNs (Virtual Private Networks).

Risk Analysis Of The System

This section performs a risk analysis of the system. As previously mentioned, the firewall will support three network interfaces:



In this diagram, the different colored lines represent the relative risk of each network connection:

- The black and yellow dashed line signifies that the Internet is an unprotected network. The source of most (but not all) attacks directed at the firewall will be the Internet.
- The red line signifies that the DMZ is a partially protected network. The DMZ is protected by the firewall, but the firewall does allow anonymous inbound connections to servers on the DMZ. For this reason, the DMZ is considered partially protected. Servers on the DMZ include:
    - External DNS Server

      The External DNS Server provides name resolution for public servers such as the External Mail Server and External Webserver.
    - External Mail Server

      The External Mail Server sends and receives electronic mail.
    - External Webserver

      The External Webserver is the home of the corporate website.

Note that anonymous inbound connections are allowed to all of these servers.

- The blue line signifies that the Internal Network is a fully protected network. The firewall does not allow any inbound connections to the Internal Network. For this reason, the Internal Network is considered fully protected. In addition, the following servers on the Internal Network allow users to access approved services on the Internet:
    - Internal DNS Server

      The Internal DNS Server provides name resolution for internal servers, and communicates with DNS servers on the Internet to provide name resolution for external servers.
    - Internal Mail Server

      The Internal Mail Server connects to the External Mail Server on the DMZ, allowing users to send and receive email.
    - Proxy Server

      The Proxy Server connects to web servers on the Internet, allowing users to browse the web.

  Note that all of these servers initiate outbound connections to the Internet or DMZ. The firewall will not allow any inbound connections to the Internal Network.

Since the firewall will act as the bridge between unprotected, partially protected, and fully protected networks, the firewall itself must be configured in an extremely secure manner. The firewall will be constantly subjected to a myriad of attacks from the Internet, and possibly from the Internal Network and DMZ as well. While users will not reside on the DMZ, it is possible that attackers could compromise the security of a server on the DMZ, which could then be used as a stepping stone for further attacks. Consequently, the firewall could be subjected to attacks not only from the Internet and Internal Network, but from the DMZ as well.

The key security concern is if an attacker is somehow able to compromise the security of the Solaris 8 operating system, which would then allow the attacker to subvert the security of the FireWall-1 application. In addition, an attacker could compromise another service installed on the server, such as SSH or even FireWall-1 itself. These scenarios could leave the DMZ and (worse yet) the Internal Network unprotected from attackers on the Internet. This could lead to loss or unauthorized disclosure of corporate data, or even civil liability if servers on the DMZ or workstations on the Internal Network are used to attack other networks. For these reasons, the firewall must be configured in an extremely secure manner.

The only services that need to be running on the firewall are NTP (Network Time Protocol), SSH, and the FireWall-1 services. Although users on the Internal

Network and Internet will "use" the firewall to access network resources, only a few administrators will actually require access to the firewall itself. Access will be provided by the SSH client and FireWall-1 Policy Editor. Access to the firewall itself will be restricted with both FireWall-1 rules and the TCP Wrappers program. Only approved workstations will be allowed access.

Step-By-Step Guide

This section describes each step of the Solaris 8 hardening process. The actual Solaris commands and parameters are shown in blue, and a description of each command is provided. Note that all steps in the hardening process should be performed as the root user account, which has full control of the system. The step-by-step guide follows:

1. First, install the Solaris 8 operating system. Although an in-depth discussion of Solaris 8 installation is beyond the scope of this document, general recommendations for a secure Solaris installation are offered here:

   - The installation will begin by asking you a series of questions. The following table can be used to help answer these questions:

Question	Answer
Language?	English
Locale?	English C-7bit ASCII
Networked?	Yes
DHCP?	No
Hostname?	firewall (the hostname of the firewall)
IP address?	192.168.1.1 (the address of the external interface)
Part of subnet?	Yes
Netmask?	255.255.255.0 (the subnet mask)
IPv6?	No
Kerberos?	No
Name service?	None
Time zone?	US/Central (the appropriate time zone)
Date and time?	Confirm that the date and time are correct
Installation?	Initial
64 bit?	No

     Note that the hostname configured during installation should not be registered with DNS, so a descriptive name such as "firewall" is acceptable. In addition, note that FireWall-1 4.1 does not support a 64-bit installation. If you choose a 64-bit bit installation FireWall-1 will not function properly, and you will be forced to reinstall the operating system.

   - When prompted to select the software bundle to install, choose "Core System Support". This bundle includes the core system software plus basic administration utilities, but does not include other dangerous packages that could be maliciously used by

attackers. In addition, the following packages should also be selected for installation:

Packages	Installs	Description
SUNWlibC SUNWlibCx SUNWter SUNWadmc SUNWadmfw	Compilers, terminal support, and system and network administration libraries	These packages are necessary for proper FireWall-1 operation
SUNWdoc SUNWman	Man pages	Manual pages for system commands and configuration files
SUNWntpr SUNWntpu	NTP	NTP daemons and configuration files

These packages should be selected by clicking on the "Customize" button.

- After choosing the hard drive for installation, when prompted to preserve existing data, select "Continue".
- When prompted for filesystem partitioning information, select "Manual Layout" and configure the filesystem as follows:

Partition	Size	Description
swap	2,048 MB	Swap space
/	1,024 MB	Core operating system files
/usr	3,072 MB	User files
/opt	3,072 MB	Software packages including FireWall-1
/var	9,216 MB	Log files including FireWall-1 logs

This filesystem occupies a total of 18,432 MB, or 18 GB. Note that the /var directory has plenty of space to store log files, including the FireWall-1 fw.log file. Depending on the amount of network traffic, this log file can grow rapidly. Also, since the /var directory is located a dedicated partition, if the log files grow to fill the entire partition, the firewall will still function properly, albeit without further logging. Conversely, if the root partition (the / partition) was filled, the firewall would panic and reboot, resulting in a DoS (Denial of Service) attack.

- When prompted to mount remote filesystems, select "Continue". Mounting remote filesystems is not recommended, as several vulnerabilities in the NFS (Network File System) protocols could compromise the security of the firewall.
- Click the "Begin installation" button to start the installation.

2. Once the installation is complete, login in as the "root" user. The root password has not yet been configured, so simply hit the "Enter" key when prompted for a password.

3. Set a secure password for the root user account. The password should not be based on any word that appears in the dictionary, and should contain a combination of uppercase and lowercase letters, numbers, and special characters. Do not write the password down. Set the root password with the following command:

   solaris $ passwd

   A strong root password is essential to the security of the firewall.

4. Configure OpenBoot security settings. To access the OpenBoot prompt, simultaneously depress the "Stop" and "A" keys. Set an OpenBoot password with the following command:

   ok> password

   When prompted, enter the desired password. Configure OpenBoot security with the following command:

   ok> setenv security-mode command

   This requires the OpenBoot password to be entered to execute any OpenBoot commands other than "continue". Finally, configure OpenBoot to automatically boot into the Solaris operating system with the following command:

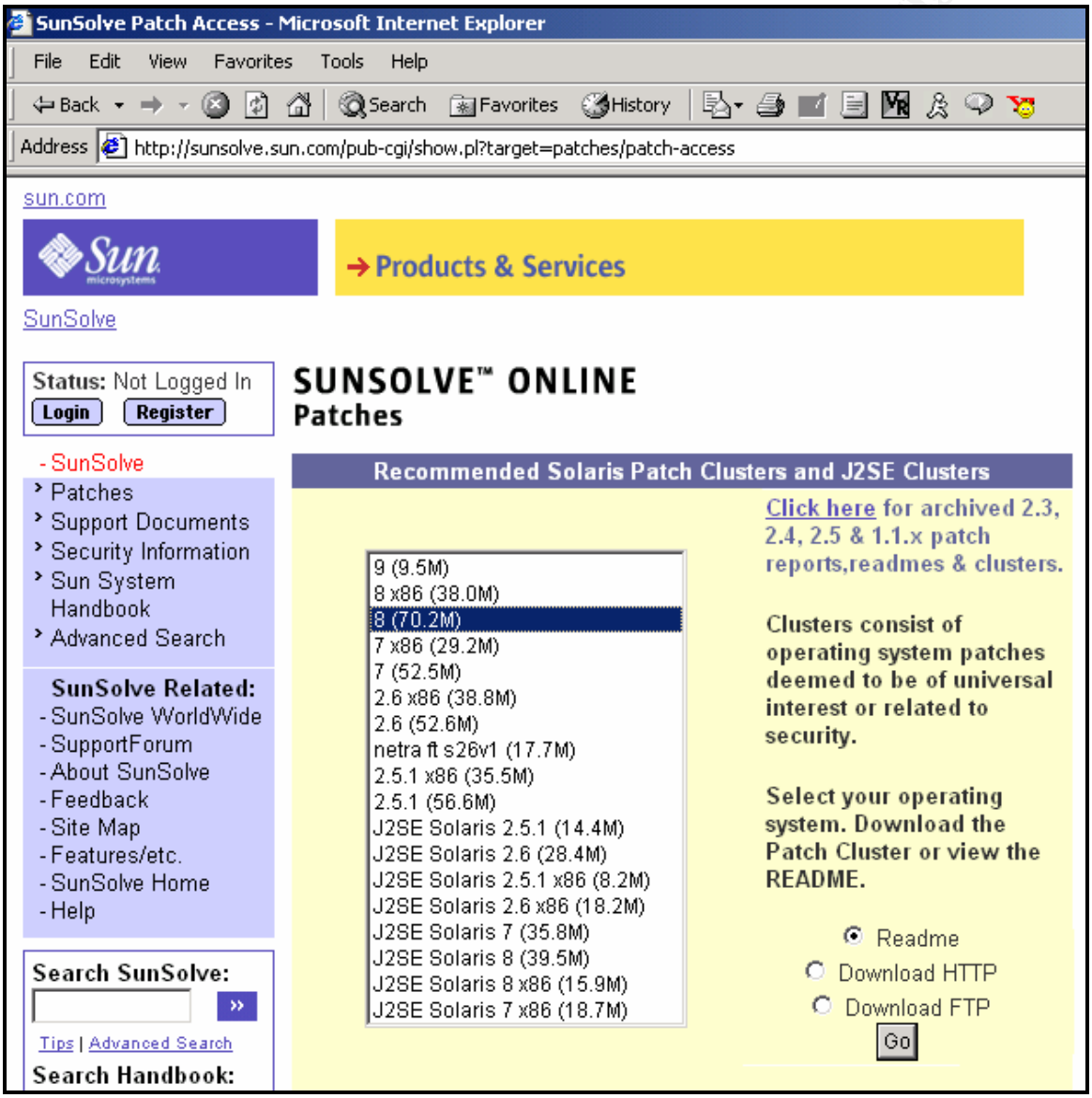
   ok> setenv auto-boot? True

   To complete OpenBoot configuration, reset the system with the following command:

   ok> reset

   The system reset will activate all of the OpenBoot changes.

5. Next, Solaris patches will be installed. Installing patches is critical to the security of the firewall, as patches fix existing vulnerabilities. From another server or workstation, download the latest Solaris 8 patch cluster from the "SunSolve Online Patches" website [3]. Browse to the "Recommended Solaris Patch Clusters and J2SE Clusters" section and select "8". Select either "Download HTTP" or "Download FTP" and click "Go" to download the patch cluster:



It is important that the firewall **is not** connected to the Internet before it is properly hardened. Consequently, the patch cluster should be downloaded to another server (henceforth referred to as the "patch server") and then securely transferred to the firewall. The following steps will demonstrate how to securely transfer and install the patch cluster.

6. Disconnect the patch server from the Internet, and use a crossover cable to connect the patch server directly to the Solaris server. If the patch

server has multiple network interfaces, ensure that they are all disconnected from the Internet.

7. Configure IP addresses on both servers. Configure the firewall with the following command:

   ```
   solaris $ ifconfig qfe0 10.1.1.1 netmask 255.255.255.0 broadcast
             10.1.1.255
   ```

   Configure the patch server with the following command:

   ```
   patch $ ifconfig eth0 10.1.1.2 netmask 255.255.255.0 broadcast
           10.1.1.255
   ```

   Note that the IP addresses must be on the same subnet, in this case the 10.1.1.0 subnet. No routing is necessary, so default gateways do not need to be configured.

8. On the patch server, connect to the FTP service on the firewall with the following commands:

   ```
   patch $ cd <path_to_patch_cluster>
   patch $ ftp 10.1.1.1
   ```

   When prompted, login as root. Next, upload the patch cluster to the Solaris server with the following commands:

   ```
   ftp> cd /var/tmp
   ftp> bin
   ftp> put 8_Recommeded.zip
   ```

   Alternatively, the patch cluster could be downloaded to the patch server and burned to CDROM. After transferring the CDROM to the firewall, the patch cluster could be copied to the appropriate directory with the following commands:

   ```
   solaris $ mount –F hsfs –o ro /dev/dsk/c0t6d0s0 /mnt
   solaris $ cp /mnt/8_Recommeded.zip /var/tmp
   ```

   This method requires a CDROM burner on the patch server, but eliminates the need for the crossover cable connection. Both methods are effective.

9. Halt the firewall with the following command:

   ```
   solaris $ halt
   ```

   Press <ENTER> to access the OpenBoot prompt. At the OpenBoot prompt, boot into single-user mode with the following command:

   ```
   'ok' boot –s
   ```

   It is important to boot the server into single-user mode before installing the patch cluster. If the server is not booted into single-user mode before installing the patch cluster, the kernel could become corrupted, which would require a complete reinstallation of the operating system.

10. Unzip and install the patch cluster with the following commands:

```
solaris $ cd /var/tmp
solaris $ unzip 8_Recommended.zip
solaris $ cd 8_Recommended
solaris $ ./install_cluster –nosave
```

The "–nosave" option specifies that the system should not be backed up before installing the patches, which saves a considerable amount of disk space. If you want to backup the system, omit the "–nosave" option. Ignore error codes 2 and 8, as they indicate that individual patches have already been installed, or do not apply to the installed software. The firewall will need to be rebooted after the patches have been successfully installed.

11. After rebooting, remove the patch files from the firewall with the following commands:

```
solaris $ cd /var/tmp
solaris $ rm –Rf 8_Recommended.zip 8_Recommended
```

Be sure to exercise extreme care with the "rm –Rf" command, as this command recursively deletes files and subdirectories. For example, if you accidentally appended a "/" to the command, the resulting "rm –Rf 8_Recommended.zip 8_Recommended /" command would begin deleting the entire system, file by file and subdirectory by subdirectory.

12. Next, the network configuration files will be created. To configure network interfaces, first edit the /etc/hosts file and add a line with the hostname of the network interface. For example, to associate the IP address 192.168.1.1 with the hostname fw-internet, add the following line to the /etc/hosts file:

```
fw-internet      192.168.1.1
```

The /etc/hosts file can be edited with the following command:

```
solaris $ vi /etc/hosts
```

The vi editor is not very intuitive, and can be confusing for novice users. For more information regarding the vi editor, refer to the vi man page with the following command:

```
solaris $ man vi
```

Create a corresponding /etc/hostname.qfe*n* file, where *n* is the number of the interface you are configuring. For example, if you are configuring the qfe0 interface, create the /etc/hostname.qfe0 file with the following command:

```
solaris $ echo "fw-internet" > /etc/hostname.qfe0
```

Note that the /etc/hostname.qfe0 file contains the hostname "fw-internet", which will cause Solaris to reference the /etc/hosts file and configure the

qfe0 interface with the IP address of "fw-internet", which is 192.168.1.1. To add additional interfaces, just add similar lines to the /etc/hosts file and create corresponding /etc/hostname.qfe*n* files. For the firewall, entries for the fw-dmz and fw-internal network interfaces should be created.

13. To configure the default route, create the /etc/defaultrouter file. For example, if the IP address of the default router is 192.168.1.254, create this file with the following command:

    solaris $ echo "192.168.1.254" > /etc/defaultrouter

    Since this server will function as a firewall, it is extremely important that the default route is correctly configured. After a subsequent reboot, the routing table can be checked with the following command:

    solaris $ netstat –nr

    Although the presence of the /etc/defaultrouter file prevents Solaris from starting the routing daemon in.routed, creating the file /etc/notrouter will provide additional routing security. Create this file with the following command:

    solaris $ touch /etc/notrouter

    This command will prevent the firewall from forwarding packets should the FireWall-1 application abnormally terminate.

14. To configure DNS name resolution, add a line to the /etc/resolv.conf file for each nameserver. For example, if the IP addresses of the nameservers are 172.16.1.1 and 172.16.1.2, add lines to the /etc/resolv.conf file with the following commands:

    solaris $ echo "nameserver 172.16.1.1" >> /etc/resolv.conf
    solaris $ echo "nameserver 172.16.1.2" >> /etc/resolv.conf

    For the sake of redundancy, it is recommended to add at least two nameservers to the /etc/resolv.conf file.

15. To configure the firewall to use DNS, edit the /etc/nsswitch.conf file with the following command:

    solaris $ vi /etc/nsswitch.conf

    Change the line "hosts: files" to "hosts: files dns". This will cause the firewall to first attempt name resolution using the /etc/hosts file, and then using DNS. This is the final network configuration file that needs to be created. Do not, however, connect the server to the Internet, as the hardening process is not yet complete. Later, after the hardening process is complete, proper network operation can be tested with the following command:

    solaris $ ping www.yahoo.com

    If "echo-reply" responses are received from this command, then the network interface, default route, and DNS are all properly configured.

16. Next, unnecessary services will be disabled. Unnecessary services may be vulnerable to attack and should be disabled whenever possible. Firewalls should run as few services as possible. For additional security, the following startup files can be deleted instead of simply being renamed. Furthermore, the actual binary programs that these startup files reference can be deleted as well. It is important to exercise caution, however, when deleting system initialization files and binaries. Remove Internet services from the /etc/rc2.d/S72inetsvc startup file using the following command:

    solaris $ vi /etc/rc2.d/S72inetsvc

    Change the line "/usr/sbin/inetd" to "#/usr/sbin/inetd". This disables the inetd daemon, which handles inbound connections for services such as Telnet and FTP. Since this server will function as a firewall, however, the inetd daemon is not needed and can be safely disabled. For additional security, comment out the individual services in the inetd configuration file /etc/inetd.conf with the following command:

    solaris $ vi /etc/inetd.conf

    Prepend a "#" character to any line that does not already begin with a "#" character. This ensures that each service in the /etc/inetd.conf file is disabled. Consequently, even if the inetd daemon is started (inadvertently or otherwise), no services will be available.

17. If the Telnet service is running, the Telnet banner provides attackers with exact server version information, which can be extremely helpful when searching for applicable vulnerabilities. Remove the Telnet banner with the following command:

    solaris $ echo "BANNER=\"\"" > /etc/default/telnetd

    In addition, create a Telnet warning banner with the following command:

    solaris $ echo "\nThis system is to be used for authorized \nbusiness purposes only. All activity \nis subject to monitoring and logging.\n" > /etc/issue

    The exact wording of the warning banner can be modified to suit the needs of your organization. In addition, your legal department should review the wording of the warning banner. For additional Telnet security, prevent root from logging in remotely with the following command:

    solaris $ vi /etc/default/login

    Change the line "CONSOLE" to "#CONSOLE". This will ensure that the root user account cannot login remotely, which could lead to a security breach. In order to obtain root access, users will have to first login as another user, and then su to the root account.

    Note, however, that Telnet is an insecure protocol that transmits usernames and passwords in the clear. Consequently, the Telnet protocol

should not be used unless absolutely necessary. The SSH utility provides a secure replacement for Telnet, and should be used instead.

18. Similar to the Telnet banner, the FTP banner provides attackers with exact server version information, which can be extremely helpful when searching for applicable vulnerabilities. Remove the FTP banner with the following command:

    ```
    solaris $ echo "BANNER=\"\"" > /etc/default/ftpd
    ```

    In addition, FTP access control can be configured using the following commands:

    ```
    solaris $ cat /etc/passwd | cut -f1 -d: > /etc/ftpusers
    solaris $ chmod 600 /etc/ftpusers
    solaris $ chgrp sys /etc/ftpusers
    ```

    This prevents system accounts from connecting to the FTP service. If the FTP service is enabled (which is not recommended), the user account which will be used to login to the FTP service should be removed from the /etc/ftpusers file. User accounts whose name appear in the /etc/ftpusers file are not allowed to connect to the FTP service.

    Note, however, that FTP is an insecure protocol that transmits usernames and passwords in the clear. Consequently, the FTP protocol should not be used unless absolutely necessary. The SSH utility provides a secure replacement for FTP, and should be used instead.

19. The "R" services (rexec, rlogin, and rsh) provide remote command execution capabilities. Unfortunately, these commands provide weak authentication based on the source IP address of the connection, and are vulnerable to attack. Consequently, the "R" services configuration files should be disabled with the following commands:

    ```
    solaris $ touch /.rhosts
    solaris $ chmod 000 /.rhosts
    solaris $ touch /etc/hosts.equiv
    solaris $ chmod 000 /etc/hosts.equiv
    solaris $ touch /.netrc
    solaris $ chmod 000 /.netrc
    ```

    Consequently, even if the "R" services are started (inadvertently or otherwise), attackers will not be able to authenticate based on their source IP address, which can be easily spoofed.

20. The Routing Information Protocol (RIP) provides dynamic routing functionality. Since this server will function as a firewall, however, the RIP service can be safely disabled with the following command:

    ```
    solaris $ vi /etc/rc2.d/S69inet
    ```

    Change the line "in.routed –s" to "#in.routed –s" and the line "in.routed –q" to "#in.routed –q". This disables the in.routed routing daemon and RIP.

21. NCSD (Name Server Cache Daemon) caches nameserver requests. Since this server will function as a firewall, however, NSCD can be safely disabled with the following commands:

```
solaris $ mv /etc/rc2.d/S76nscd /etc/r2.d/s76nscd
solaris $ chmod 000 /etc/rc2.d/s76nscd
```

These commands rename the NSCD daemon startup file from /etc/rc2.d/S76nscd to /etc/rc2.d/s76nscd. Only startup files that begin with an "S" are processed during system initialization, so renaming the startup file in this manner prevents the NSCD daemon from starting.

22. NFS (Network File System) is used by to allow remote clients to access local filesystems. Since this server will be function as a firewall, however, NFS can be safely disabled with the following commands:

```
solaris $ mv /etc/rc3.d/S15nfs.server /etc/rc3.d/s15nfs.server
solaris $ chmod 000 /etc/rc3.d/s15nfs.server
solaris $ mv /etc/rc2.d/S73cachefs.daemon
          /etc/rc2.d/s73cachefs.daemon
solaris $ chmod 000 /etc/rc2.d/s73cachefs.daemon
solaris $ mv /etc/rc2.d/S73nfs.client /etc/rc2.d/s73nfs.client
solaris $ chmod 000 /etc/rc2.d/s73nfs.client
solaris $ mv /etc/rc2.d/S74autofs /etc/rc2.d/s74autofs
solaris $ chmod 000 /etc/rc2.d/s74autofs
solaris $ mv /etc/rc2.d/S93cacheos.finish /etc/rc2.d/s93cacheos.finish
solaris $ chmod 000 /etc/rc2.d/s93cacheos.finish
```

These commands rename the NFS startup files so they are not processed during system initialization. In addition, the NFS configuration files can be disabled with the following commands:

```
solaris $ mv /etc/auto_home /etc/_auto_home
solaris $ chmod 000 /etc/_auto_home
solaris $ mv /etc/auto_master /etc/_auto_master
solaris $ chmod 000 /etc/_auto_master
```

These commands prepend a "_" character into the name of each NFS configuration file. Consequently, even if NFS is somehow started (inadvertently or otherwise), attackers will not be able to access this vulnerable service. This is extremely important since NFS has historically been plagued by security holes.

23. The vi buffer preserve service is used to make automatic backups of edited documents. Since this server will function as a firewall, however, the vi buffer preserve service can be safely disabled with the following commands:

```
solaris $ mv /etc/rc2.d/S80PRESERVE /etc/rc2.d/s80PRESERVE
solaris $ chmod 000 /etc/rc2.d/s80PRESERVE
```

These commands rename the vi buffer preserve service startup file so that it is not processed during system initialization.

24. The sendmail service is used to provide email service to clients. Since this server will function as a firewall, however, the sendmail service can be safely disabled with the following commands:

    ```
    solaris $ mv /etc/rc2.d/S88sendmail /etc/rc2.d/s88sendmail
    solaris $ chmod 000 /etc/rc2.d/s88sendmail
    ```

    These commands rename the sendmail startup file so that it is not processed during system initialization. This is extremely important since sendmail has historically been plagued by security holes. It is also important, however, to clear the mail queue in order to process any outgoing email. Configure a cron job to clear the mail queue with the following command:

    ```
    solaris $ crontab –e root
    ```

    Add the line "0 * * * * /usr/lib/sendmail –q" to the crontab file. This will cause the firewall to clear the sendmail queue every hour.

25. Disable the autoinstall startup file, which reinstalls the Solaris operating system if the file /AUTOINSTALL exists:

    ```
    solaris $ mv /etc/rc2.d/S72autoinstall /etc/rc2.d/s72autoinstall
    ```

    This command prevents the system from reinstalling Solaris if the file /AUTOINSTALL exists, preventing DoS attacks.

26. Note that the Syslog service is left enabled in the startup file /etc/rc2.d/S74syslog. Prevent Syslog from accepting network connections with the following command:

    ```
    solaris $ vi /etc/init.d/syslog
    ```

    Change the line "/usr/sbin/syslogd >/dev/msglog 2>&1 &" to "/usr/sbin/syslogd –t >/dev/msglog 2>&1 &". The "–t" option prevents Syslog from accepting network connections.

27. Note that NTP is left enabled in the startup file /etc/rc2.d/S74xntpd. NTP is used to synchronize the date and time between servers on a network, and is extremely useful for security event correlation. Other important servers, such as syslog servers and IDS, should also be configured to use NTP. NTP is configured in the /etc/inet/ntp.conf file. If the IP addresses of the local NTP servers are 10.1.1.25, 10.1.1.26, and 10.1.1.27, configure the /etc/inet/ntp.conf file as follows:

    ```
    server 10.1.1.25
    server 10.1.1.26
    server 10.1.1.27
    server 127.127.1.1
    fudge 127.127.1.1 stratum 5
    driftfile /etc/ntp.drift
    ```

```
restrict default ignore
restrict 127.0.0.1 nomodify
```

The pseudo-clock 127.127.1.1 is included for redundancy, in case the other NTP servers are unreachable. In the event of a security breach, NTP will make it much easier to accurately trace the steps of an attacker through your network.

28. Next, network drivers will be configured for optimal performance. This not only boosts the performance of the FireWall-1 application, but also strengthens the security of the firewall. For example, the tcp_xmit_hiwat and tcp_recv_hiwat drivers help secure the firewall against "SYN Flood" attacks. The following commands create the startup file /etc/rc2.d/S99netdrivers, which will properly configure the network drivers each time the server is rebooted. Descriptions precede each command (or each set of commands). These commands increase the size of the default TCP send and receive buffers, allowing the firewall to more efficiently process network traffic:

```
solaris $ echo "ndd -set /dev/tcp tcp_xmit_hiwat 65535" >
        /etc/rc2.d/S99netdrivers
solaris $ echo "ndd -set /dev/tcp tcp_recv_hiwat 65535" >>
        /etc/rc2.d/S99netdrivers
```

These commands increase the size of the requested connections queues, allowing the firewall to handle a greater amount of network connections:

```
solaris $ echo "ndd -set /dev/tcp tcp_conn_req_max_q 1024" >>
        /etc/rc2.d/S99netdrivers
solaris $ echo "ndd -set /dev/tcp tcp_conn_req_max_q0 4096" >>
        /etc/rc2.d/S99netdrivers
```

This command prevents network congestion due to the TCP "slow-start" bug:

```
solaris $ echo "ndd -set /dev/tcp tcp_slow_start_initial 2" >>
        /etc/rc2.d/S99netdrivers
```

This command prevents the firewall from forwarding directed broadcasts, which can be used by attackers to cause network congestion:

```
solaris $ echo "ndd -set /dev/ip ip_forward_directed_broadcasts 0" >>
        /etc/rc2.d/S99netdrivers
```

This command prevents the firewall from responding to ICMP "echo-request" packets sent to broadcast addresses, which can be used by attackers to quickly map the network topology:

```
solaris $ echo "ndd -set /dev/ip ip_respond_to_echo_broadcast 0" >>
        /etc/rc2.d/S99netdrivers
```

These commands prevent the firewall from responding to ICMP "timestamp request" and "broadcast timestamp request" packets, which

can be used by attackers to probe the firewall for system information and cause network congestion:

```
solaris $ echo "ndd -set /dev/ip ip_respond_to_timestamp 0" >>
        /etc/rc2.d/S99netdrivers
solaris $ echo "ndd -set /dev/ip ip_respond_to_timestamp_broadcast
        0" >> /etc/rc2.d/S99netdrivers
```

This command prevents the firewall from processing source-routed packets, which can be used by attackers to map the network topology and launch trust-based attacks against vulnerable servers:

```
solaris $ echo "ndd -set /dev/ip ip_forward_src_routed 0" >>
        /etc/rc2.d/S99netdrivers
```

This command prevents the firewall from processing IP redirects, which can be used by attackers to manipulate the routing table on the firewall:

```
solaris $ echo "ndd -set /dev/ip ip_ignore_redirect 1" >>
        /etc/rc2.d/S99netdrivers
```

Complete the network driver optimization with the following commands:

```
solaris $ chmod 744 /etc/rc2.d/S99netdrivers
solaris $ chgrp sys /etc/rc2.d/S99netdrivers
```

These commands configure secure permissions and ownership for the network driver configuration startup file.

29. Next, unnecessary set-UID and set-GID permissions will be removed. Programs with the set-UID or set-GID set run with elevated privileges, and may be vulnerable to attack. Find all set-UID programs with the following command:

```
solaris $ find / –type f –perm –u+s –ls
```

If any of these programs are not needed, remove the set-UID bit with the following commands:

```
solaris $ chmod u–s <filename>
solaris $ chmod u+x <filename>
```

Find all set-GID programs with the following command:

```
solaris $ find / –type  f –perm –g+s –ls
```

If any of these programs are not needed, remove the set-GID bit with the following commands:

```
solaris $ chmod g–s <filename>
solaris $ chmod g+x <filename>
```

Disabling unnecessary set-UID and set-GID programs will greatly enhance the security of the firewall. For a list of which set-UID and set-GID programs can be safely disabled, see the Solaris Security Digest [4].

30. Configure secure TCP ISN (Initial Sequence Number) randomness with the following command:

    solaris $ vi /etc/default/inetinit

    Then change the line "TCP_STRONG_ISS=1" to "TCP_STRONG_ISS=2". This ensures that Solaris will select strong ISS (Initial Send Sequence) numbers. This helps protect the server against IP spoofing attacks.

31. Configure stack parameters to help prevent buffer overflows with the following commands:

    solaris $ echo "set noexec_user_stack=1" >> /etc/system
    solaris $ echo "set noexec_user_stack_log=1" >> /etc/system

    Buffer overflows may be present in poorly coded software, and could allow attackers to execute arbitrary commands on the server. These commands help secure the firewall against buffer overflows by preventing the user stack from being executable.

32. Prevent core dumps with the following command:

    solaris $ echo "set sys:coredumpsize = 0" >> /etc/system

    Core dumps can be used by attackers to glean sensitive server information.

33. Set non-login shells for system accounts with the following command:

    solaris $ vi /etc/passwd

    Then add "/bin/false" to any line that ends with the ":" character. This prevents attackers from compromising system accounts and logging in to the server.

34. Configure a secure path for the root account with the following command:

    solaris $ echo "PATH=/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin" >> /.profile

    This command configures a secure path for the root account. The path determines which command is executed when an absolute path is not given. By restricting the directories in the path, the root account is protected from trojaned commands.

35. Configure a secure umask for the root account with the following command:

    solaris $ echo "umask 077" >> /.profile

    This command configures a secure umask. The umask is used to determine the permissions applied to new files and directories. The umask value "077" prevents new files and directories from being readable, writeable, or executable by anyone other than root.

36. Configure a secure system daemon umask with the following commands:

```
solaris $ echo “umask 022” > /etc/init.d/umask.sh
solaris $ chmod 744 /etc/init.d/umask.sh
solaris $ ln –s /etc/init.d/umask.sh /etc/rc2.d/S00umask
solaris $ ln –s /etc/init.d/umask.sh /etc/rc3.d/S00umask
```

The umask value “022” prevents new files and directories from being writeable by anyone except the user that created the file or directory.

37. Remove unnecessary crontab files with the following commands:

```
solaris $ crontab -r adm
solaris $ crontab -r lp
```

Crontab files are used to automatically run scheduled jobs, and can be used by attackers to run malicious code. These commands remove unnecessary crontab files. For additional security, limit cron and at access to the root user account with the following commands:

```
solaris $ echo “root” > /etc/cron.d/cron.allow
solaris $ echo “root” > /etc/cron.d/at.allow
```

This will prevent other users from scheduling jobs with cron or at.

38. Next, additional security logs will be configured. Some of these security logs are only utilized if they have been created. A description of each log precedes each set of commands. The /var/log/authlog log captures authorization events from the system logging daemon. Configure this log with the following commands:

```
solaris $ echo "\n# Authorization
          information\nauth.info\t/var/log/authlog" >> /etc/syslog.conf
solaris $ touch /var/log/authlog
solaris $ chmod 640 /var/log/authlog
solaris $ chgrp sys /var/log/authlog
```

The first command configures the system logging daemon to send authorization events to the /var/log/authlog file, while the other commands create this file and set secure file permissions and ownership. The /var/adm/loginlog log captures consecutive failed login attempts. Configure this log with the following commands:

```
solaris $ touch /var/adm/loginlog
solaris $ chmod 640 /var/adm/loginlog
solaris $ chgrp sys /var/adm/loginlog
```

These commands create the /var/adm/loginlog file and set secure file permissions and ownership. The /var/adm/sulog log captures failed su attempts. Configure this log with the following commands:

```
solaris $ touch /var/adm/sulog
solaris $ chmod 640 /var/adm/sulog
solaris $ chgrp sys /var/adm/sulog
```

These commands create the /var/adm/sulog file and set secure file permissions and ownership. These additional logs will provide a wealth of useful security information on the firewall.

39. Reboot the firewall with the following command:

    solaris $ init 6

    Rebooting the firewall will activate all of the changes made to the Solaris operating system.

40. Next, several useful security tools will be installed. Although an in-depth discussion of each tool is beyond the scope of this document, the tools are summarized here. In addition, basic installation and configuration commands are provided for each tool. Compiled binaries for some tools can be downloaded from the Sun Freeware website, which is an excellent source for a wide variety of Solaris tools [5].

    - TCP Wrappers 7.6

      The TCP Wrappers program provides granular access control for Internet services, and can be used to provide enhanced SSH security. The TCP Wrappers program will be compiled on a separate server (henceforth referred to as the "development server"), on which the Solaris "Entire Distribution" software bundle was installed. This is necessary because compiling TCP Wrappers requires the cc and make commands, which were not installed on the firewall. After downloading the TCP Wrappers source code, compile the tool with the following commands:

      development $ gzip –d tcp_wrappers_7.6.tar.gz
      development $ tar xvf tcp_wrappers_7.6.tar
      development $ cd tcp_wrappers_7.6
      development $ vi Makefile

      Change the line "#REAL_DAEMON_DIR=/usr/sbin" to "REAL_DAEMON_DIR=/usr/sbin". Finish compiling the tool with the following command:

      solaris $ make sunos5

      Once the compiling is complete, FTP the "tcpd" binary to the /usr/sbin directory on the firewall (since the FTP service is disabled on the firewall, use the "ftp" command on the firewall to connect to the FTP service on the development server). Next, create appropriate /etc/hosts.allow and /etc/hosts.deny access control files. For example, to allow SSH access from the 10.1.1.0 network, configure the /etc/hosts.allow file with the following command:

      solaris $ vi /etc/hosts.allow

Add the line "sshd : 10.1.1. ALLOW" to this file. Configure the /etc/hosts.deny file with the following command:

```
solaris $ vi /etc/hosts.deny
```

Add the line "ALL : ALL" to this file. This configuration will allow SSH connections from the 10.1.1.0 network, and will deny all other connections. For more information regarding TCP Wrappers, including source code and comprehensive documentation, see the "TCP Wrappers" package [6].

- OpenSSH 3.4

OpenSSH provides strong authentication and encryption for interactive login sessions and file transfers, and is a secure replacement for the Telnet and FTP programs. First, download the libgcc-3.2, openssl-0.9.6g, and openssh-3.4p1 packages from the Sun Freeware website [5]. Although the libgcc package will be temporarily installed, it will be removed after SSH has been successfully installed. This is necessary to simplify the OpenSSH installation process. Install the libgcc package with the following commands:

```
solaris $ gzip –d libgcc-3.2-sol8-sparc-local-gz
solaris $ pkgadd –d ./ libgcc-3.2-sol8-sparc-local
```

Next, install the OpenSSL package with the following commands:

```
solaris $ gzip –d openssl-0.9.6g-sol8-sparc-local-gz
solaris $ pkgadd –d ./ openssl-0.9.6g-sol8-sparc-local
```

Finally, install the OpenSSH package with the following commands:

```
solaris $ gzip –d openssh-3.4p1-sol8-sparc-local
solaris $ pkgadd –d ./ openssh-3.4p1-sol8-sparc-local
```

Now that OpenSSH has been successfully installed, remove the libgcc package with the following command:

```
solaris $ pkgrm libgcc-3.2-sol8-sparc-local
```

In addition, create an OpenSSH startup file with the following command:

```
solaris $ vi /etc/rc2.d/opensshd
```

Add the following lines to the file:

```
case "$1" in
'start')
        if [ -x /usr/local/sbin/sshd -a -f
        /usr/local/etc/sshd_config ]; then
                /usr/local/sbin/sshd-f /usr/local/etc/sshd_config
```

```
        fi
        ;;
'stop')
        kill 'cat /etc/sshd.pid'
        ;;
esac
```

To configure OpenSSH, first create server keys with the following commands:

```
solaris $ /usr/local/bin/ssh-keygen –b 1024 –f
          /usr/local/etc/ssh_host_key –N “”
solaris $ /usr/local/bin/ssh-keygen –d –f
          /usr/local/etc/ssh_host_dsa_key –N “”
```

Next, configure an OpenSSH banner with the following command:

```
solaris $ vi /usr/local/etc/sshd_config
```

Add the line “Banner /etc/isssue” to this file. The /etc/issue file was configured with a warning message earlier during the hardening process. Finally, start OpenSSH with the following command:

```
solaris $ /etc/init.d/sshd start
```

OpenSSH is now ready to use. For more information regarding the use of OpenSSH, see the Check Your Configuration section. For more information regarding OpenSSH, including source code and comprehensive documentation, see the “OpenSSH” website [7].

- Tripwire 2.3

  The Tripwire program monitors system files for any type of modification, detecting possible security breaches. The Tripwire program will be compiled on the development server, on which the Solaris “Entire Distribution” software bundle was installed. This is necessary because compiling Tripwire requires the cc and make commands, which were not installed on the firewall. After downloading the Tripwire source code, compile the tool with the following commands:

```
development $ gzip –d tripwire-2.3-47.tar.gz
development $ tar xvf tripwire-2.3-47.tar
development $ cd tripwire-2.3-47
development $ cp config/tw.conf.sunos5 /etc/tw.config
development $ make
```

  Next, copy the binaries src/tripwire and src/siggen to the /usr/local/sbin directory on the firewall (since the FTP service is disabled on the firewall, use the “ftp” command on the firewall to

connect to the FTP service on the development server). In addition, copy the files/tw.conf.solaris file to /etc/tw.config on the firewall. Edit this configuration file with the following command:

```
solaris $ vi /etc/tw.config
```

Add the following lines the file (assuming $FWDIR is the default /opt/CPfw1-41):

```
/opt/CPfw1-41/conf/objects.C          R
/opt/CPfw1-41/conf/rulebases.fws      R
/opt/CPfw1-41/conf/fwauth.NDB         R
/opt/CPfw1-41/conf/gui-clients        R
/opt/CPfw1-41/conf/snmp.C             R
/opt/CPfw1-41/lib/setup.C             R
/opt/CPfw1-41/lib/control.map         R
```

These additional lines include critical FireWall-1 configuration files in the Tripwire database. If these files are modified and you did not make any FireWall-1 configuration changes, a security breach may have occurred. Tripwire is now ready to use. For example, create a database on a floppy disk with the following commands:

```
solaris $ mount -n /dev/disk /floppy
solaris $ tripwire -initialize
solaris $ cp /etc/tw.config /floppy
solaris $ cp /usr/local/sbin/tripwire /floppy
```

At this point, it would be prudent to write-protect the floppy disk. Finally, run Tripwire with the following commands:

```
solaris $ cd /floppy
solaris $ ./tripwire -c ./tw.config -d ./databases/tw.db_fw-
          internet
```

The Tripwire report will reveal any files that have changed. For more information regarding Tripwire, including source code and comprehensive documentation, see the "The Tripwire Open Source Project" website [8].

- Fix–Modes

  The Fix–Modes program configures restrictive permissions on critical system files. After downloading Fix–Modes, run the tool with the following commands:

```
solaris $ gzip –d fix–modes.tar.gz
solaris $ tar xvf fix–modes.tar
solaris $ cd fix–modes
solaris $ make
solaris $ sh ./fix-modes
```

For more information regarding Fix–Modes, including source code and comprehensive documentation, see the "Fix–Modes" package [9].

It is recommended that each of these security tools is installed and configured, as each greatly enhances the security of the firewall.

41. Next, the Check Point FireWall-1 packages will be installed. FireWall-1 4.1 Service Pack 5 or later must be installed, as previous versions of FireWall-1 were not supported on Solaris 8. Another option is to install the latest version of FireWall-1, which is codenamed FireWall-1 NG (Next Generation). This is not recommended, however, as FireWall-1 4.1 is currently a more stable product. Although FireWall-1 NG is laden with new features, the product has been plagued by bugs and unpredictable behavior. Since a firewall is not a good place to experiment with new software, FireWall-1 4.1 is recommended. Although an in-depth discussion of FireWall-1 installation is beyond the scope of this document, several recommendations for a secure FireWall-1 installation are offered here:

    - Install the "VPN-1 & FireWall-1 Enforcement Module" on the firewall. For optimal performance, the "VPN-1 & FireWall-1 Enterprise Management" should be installed on another server.
    - When prompted, choose not to install the High Availability (HA) module.
    - When prompted, allow FireWall-1 to disable IP forwarding during system initialization.
    - When prompted, allow FireWall-1 to install a default filter during system initialization.
    - When prompted, choose not to install a license. Instead, after the FireWall-1 installation is complete, copy and paste the "fw putlic" command directly from your licensing email to the command prompt. This will save tedious work and frustration.
    - When prompted, choose not to configure group permissions.
    - When prompted, choose not to enable SNMP.
    - When prompted, choose to configure SMTP. Enter the information for your SMTP server.
    - When prompted, be sure to create a secure password for the fwadmin user account, as this user can install security policies on the firewall. Note that fwadmin is not a Solaris user account, but rather a FireWall-1 user account defined in the $FWDIR/conf/fwauth.NDB configuration file.
    - Note that the log files created in the $FWDIR/log directory are actually symbolic links to the /var/$FWDIR/log directory. Since the

/var directory is located on a dedicated partition, the firewall will be protected against DoS attacks. If the FireWall-1 logs somehow grow to fill the entire /var partition, the server will not reboot, thus protecting the firewall from DoS attacks.

- Boost FireWall-1 performance with the following commands:

  solaris $ echo "set tcp:tcp_conn_hash_size=16384" >> /etc/system

  This command increases the size of the TCP connections table, allowing the FireWall-1 application to handle more connections.

  solaris $ echo "set rlim_fd_max=16384" >> /etc/system

  This command increases the amount of available file descriptors, allowing the FireWall-1 application to handle more connections.

  solaris $ echo "set fw:fwhmem=0x1000000" >> /etc/system

  This command increases the amount of memory allocated to the FireWall-1 application.

- Once FireWall-1 is installed, you should create a security policy for your firewall. A simple policy is show below:

Source	Destination	Service	Action	Track
Any	Firewall	Any	drop	Long
Any	Any	Any	drop	Long

  The first rule is called the "stealth rule", and drops all traffic destined for the firewall itself. The second rule is called the "cleanup rule", and drops all other traffic. Between the "stealth rule" and "cleanup rule" you should add other rules that accept network connections as defined by your security policy. For more information regarding creating a security policy, see my GIAC Certified Firewall Analyst (GCFW) practical [10].

For more information regarding FireWall-1 installation, see the "Check Point Support" website [11].

42. Last but not least, create a backup of the hardened system. The ufsdump command can be used to create raw filesystem backups. For example, create a full backup of the first partition to the first tape drive with the following command:

    solaris $ ufsdump 0uv /dev/rdsk/c0t0d0s1

    The "0" option specifies a full backup (rather than an incremental backup). The differences between full and incremental backups are detailed in the Ongoing Maintenance section. The backup can be restored with the following command:

```
solaris $ ufsrestore r
```

For more information regarding the ufsdump and ufsrestore commands, refer to the appropriate manpages.

After following the steps in the hardening process, the default Solaris 8 installation will be much more secure. The following section will describe how to maintain this high level of security. For additional information regarding Solaris hardening, see the SANS "Solaris Security Step-by-Step" booklet [12].

Ongoing Maintenance

This section describes a strategy for maintaining the security of the hardened Solaris installation. Just because your server is secure today does not mean that it will be secure tomorrow. Consequently, ongoing maintenance is required. Recommendations for an ongoing maintenance strategy follow:

- Stay up-to-date with Solaris patches

  Staying up-to-date with Solaris patches is critical to the security of the firewall. Visit the "SunSolve Online Patches" website on a regular basis, and install patches in a timely manner [3]. For information regarding how to install Solaris patches, see the Step-By-Step Guide section. As previously mentioned, the importance of staying up-to-date with Solaris patches cannot be overstated.

- Stay up-to-date with FireWall-1 Service Packs

  At the application level, staying up-to-date with FireWall-1 Service Packs is also critical to the security of the firewall. Visit the "Check Point Support" website on a regular basis, and install Service Packs in a timely manner [11]. For example, after downloading FireWall-1 4.1 Service Pack 6, install the Service Pack with the following commands:

  ```
  solaris $ gzip –d patchadd CPFWSP410006-01.tar.gz
  solaris $ tar xvf CPFWSP410006-01.tar
  solaris $ cd CPFWSP410006-01
  solaris $ patchadd CPFWSP410006-01
  ```

  As with Solaris patches, the importance of staying up-to-date with FireWall-1 Service Packs cannot be overstated.

- Consistently create server backups

  Server backups are essential in the event of a security breach (or in the event of hardware failure). Consequently, it is extremely important to consistently create server backups. It is important to utilize both full backups (backing up the entire server) and incremental backups (backing up only what has changed since the last full backup). Full backups should be created on a weekly basis, while incremental backups should be created on a daily basis. In addition, it is also important to test server backups to ensure the effectiveness of the backup media and restoration process. As previously mentioned, the

ufsdump and ufsrestore commands can be used to create and restore server backups. For information regarding the ufsdump and ufsrestore commands, see the Step-By-Step Guide section. To automate server backups, create cron jobs with the following command:

```
solaris $ crontab -e
```

Add the following lines to the cron file:

```
0 0 * * 0 ufsdump 0uv /dev/rdsk/c0t0d0s1
0 0 * * 1 ufsdump 1uv /dev/rdsk/c0t0d0s1
0 0 * * 2 ufsdump 2uv /dev/rdsk/c0t0d0s1
0 0 * * 3 ufsdump 3uv /dev/rdsk/c0t0d0s1
0 0 * * 4 ufsdump 4uv /dev/rdsk/c0t0d0s1
0 0 * * 5 ufsdump 5uv /dev/rdsk/c0t0d0s1
0 0 * * 6 ufsdump 6uv /dev/rdsk/c0t0d0s1
```

These cron jobs will perform a full filesystem backup every Sunday night at midnight, and will perform an incremental backup every other night at midnight. The system will keep track of which files were backed up using the /etc/dumpdates file. By utilizing this file, the system will know which files have changed since the last backup, and therefore will know which files to backup. In addition, it is important to periodically test your backups. Attempt to restore backups to another server. If backups do not work, they are worthless. Make sure your backups work.

- Consistently monitor critical system files with Tripwire

  Tripwire can help ensure that critical system files are not altered. When Tripwire is installed, it creates cryptographic checksums of critical system files, and stores these checksums in a database. When run, Tripwire regenerates the cryptographic checksums to ensure that the files have not been modified in any way. For more information regarding Tripwire operation, see the Step-By-Step Guide section. To run Tripwire on a nightly basis, add a cron job with the following command:

  ```
  solaris $ crontab –e
  ```

  Add the line "0 0 * * * /floppy/tripwire -c /floppy/tw.config -d /floppy/databases/tw.db_fw-internet" to the cron file. This will run Tripwire every night at midnight. For more information regarding Tripwire, see the "Tripwire Open Source Project" website [8].

- Consistently monitor the Solaris log files

  Suspicious activity in the Solaris log files could be a sign of hacking attempts or a security breach. Consequently, it is important to consistently monitor the log files, including those created during the hardening process. This includes the Solaris log files /var/log/authlog,

/var/adm/loginlog, and /var/adm/sulog, and the FireWall-1 log file $FWDIR/log/fw.log.

- Perform port scans of the firewall on a regular basis

  The freeware tool "Nmap" can be used to perform port scans of the firewall. Nmap is highly flexible, and can scan both TCP and UDP ports. After downloading Nmap, install the tool with the following commands:

  ```
  solaris $ gzip –d nmap-2.54.tgz
  solaris $ tar xvf nmap-2.54.tar
  solaris $ cd nmap-2.54
  solaris $ ./configure
  solaris $ make
  solaris $ make install
  ```

  Nmap is run from the command line, and has several powerful options. For more information regarding running Nmap scans against your firewall, see the Check Your Configuration section. If suspicious ports are open, a security breach may have occurred. For more information regarding Nmap, including source code and comprehensive documentation, see the "Nmap Security Scanner" website [13].

- Perform vulnerability scans of the firewall on a regular basis.

  The freeware tool "Nessus" can be used to perform vulnerability scans of the firewall. Nessus is highly flexible, and can not only scan for open TCP and UDP ports, but also for actual vulnerabilities as well. After downloading Nessus, install the tool with the following command:

  ```
  solaris $ sh nessus-installer.sh
  ```

  Nessus is run from an intuitive GUI, has several powerful options, and is easily updated with new vulnerability checks. To run a vulnerability scan, first update your Nessus vulnerability checks with the following command:

  ```
  solaris $ nessus-update-plugins
  ```

  Next, start the Nessus server (if it was not already started) with the following command:

  ```
  solaris $ nessusd &
  ```

  Finally, start the Nessus client with the following command:

  ```
  solaris $ nessus &
  ```

  Once Nessus is started, select the target and various scanning options such as which vulnerability checks to include ("Enable all but dangerous plugins" is a good choice) and what ports to scan ("Nmap + privileged ports" is a good choice). Once you have configured the scan, simply click the "Start the scan" button and wait for the results.

The Nessus report format is quite intuitive, and can also be saved in HTML format. For more information regarding Nessus, including source code and comprehensive documentation, see the "Nessus Security Scanner" website [14].

- Consistently watch for new NTP, SSH, and FireWall-1 vulnerabilities

  Since new NTP, SSH, or FireWall-1 vulnerabilities could surface at any time, it is important to consistently monitor the "BugTraq" and "FireWall-1" mailing lists for new vulnerabilities. The BugTraq mailing list provides detailed information regarding the latest bugs in a variety of software packages, including NTP and SSH. The FireWall-1 mailing list provides a wealth of information regarding FireWall-1, including information regarding new vulnerabilities. For more information regarding the BugTraq mailing list, see the "BugTraq" website [15]. For more information regarding the FireWall-1 mailing list, see the "FireWall-1 Mailing List" website [16].

- Ensure that physical security is not overlooked.

  If an attacker has physical access to your firewall, they will be able to compromise your firewall. It is that simple. Consequently, it is important to maintain strong physical security for your firewall. Ensure that your firewall is stored in a locked room, and that only appropriate personnel has access to that room.

Following this ongoing maintenance strategy will help ensure the ongoing security of your firewall.

Check Your Configuration

This section verifies that the firewall has indeed been properly hardened. Five distinct checks are performed. Each check includes a detailed description and appropriate Solaris commands. The checks follow:

1. Check that the most up-to-date patches are installed

   The "Patch Check" tool, which is available from the SunSolve website, can be used to verify that all applicable patches have been installed [17]. The Patch Check tool determines exactly what patches have been installed on your firewall, and then compares the results with the current list of available patches. Patch Check should be installed on the development server, on which the Solaris "Entire Distribution" software bundle was installed. This is necessary because Patch Check requires the Perl interpreter, which was not installed on the firewall. After downloading Patch Check, install the tool with the following commands:

   ```
   development $ gzipd –d pchk_1.1.tar.Z
   development $ tar xvf pchk_1.1.tar
   ```

   In addition, download the latest patch cross-reference file from the SunSolve website [18]. Copy this file to the directory where you

installed Patch Check, and name it "patchdiag.xref". Next, run the following diagnostic commands on the firewall:

```
solaris $ /usr/bin/showrev –p > /tmp/showrev_output
solaris $ /usr/bin/pkginfo –l > /tmp/pkginfo_output
solaris $ uname –pr
```

FTP the files /tmp/showrev_output and /tmp/pkginfo_output to the development server, to the directory where you installed Patch Check (since the FTP service is disabled on the firewall, use the "ftp" command on the firewall to connect to the FTP service on the development server). In addition, note the processor type and operating system version returned by the uname command on the firewall. Next, from the directory on the development server where you installed Patch Check, run the tool with the following command:

```
development $ perl patchk.pl –b –p pkginfo_output showrev_ouput
5.8 sparc
```

In this command, "5.8" is the operating system version returned by the uname command, and "sparc" is the processor type returned by the uname command. After running Patch Check, an HTML report will be created in the /tmp directory. This report will specify whether or not more up-to-date patches should be installed. If more up-to-date patches should be installed, download and install them as detailed in the Ongoing Maintenance section. As previously mentioned, the importance of staying up-to-date with Solaris patches cannot be overstated. Consequently, the Patch Check tool should be run on a regular basis.

2. Check that the server reboots properly, and that FireWall-1 is started

   If the FireWall-1 application is not running, your network will be "down". Therefore, if the firewall is rebooted, it is important that the FireWall-1 application is restarted in a timely manner. For this to happen, two distinct events must occur:

   - The server must reboot into the Solaris operating system
   - The Solaris operating system must start the FireWall-1 application

   To verify that the FireWall-1 application is properly started, reboot the server with the following command:

```
solaris $ init 6
```

   The server should automatically reboot into the Solaris operating system, not into OpenBoot. If the server boots into OpenBoot, the OpenBoot "setenv auto-boot? True" command was not properly executed during the hardening process. To verify that the FireWall-1 application has started, check the status of FireWall-1 with the following command:

```
solaris $ fw stat
```

This command should provide information regarding network interfaces and installed security policies. If it does not, FireWall-1 was not properly installed during the hardening process. If the firewall reboots into the Solaris operating system and the FireWall-1 application is started, your firewall has been properly configured.

3. Check that dangerous services (such as Telnet and FTP) are not running

First, Nmap will be used to scan a default Solaris 8 installation before the hardening process. The scan is run from the development server, which is located on the internal network. The scan is run against the internal network interface of the firewall, whose IP address is 10.1.1.1. The Nmap "–sT" option specifies a TCP scan, and "–p 1–65535" specifies to scan all ports from 1 through 65,535. Nmap is run against the default Solaris installation before the hardening process:

```
development $ nmap –sT –p 1–65535 10.1.1.1
Starting nmap V. 2.54BETA32 ( www.insecure.org/nmap/ )
Interesting ports on 10.1.1.1 (10.1.1.1):
(The 65513 ports scanned but not shown below are in state:
closed)

Port        State    Service
7/tcp       open     echo
9/tcp       open     discard
13/tcp      open     daytime
19/tcp      open     chargen
21/tcp      open     ftp
23/tcp      open     telnet
25/tcp      open     smtp
37/tcp      open     time
79/tcp      open     finger
111/tcp     open     sunrpc
512/tcp     open     exec
513/tcp     open     login
514/tcp     open     shell
515/tcp     open     printer
540/tcp     open     uucp
587/tcp     open     submission
4045/tcp    open     unknown
5987/tcp    open     unknown
6000/tcp    open     x11
6112/tcp    open     unknown
7100/tcp    open     xfs
32773/tcp   open     unknown
```

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

As you can see, Nmap detects a colossal 22 open TCP ports, including both Telnet and FTP. The hardening process disables these unnecessary services. Nmap is run again, this time after the hardening process:

```
development $ nmap –sT –p 1–65535 10.1.1.1
Starting nmap V. 2.54BETA32 ( www.insecure.org/nmap/ )
Interesting ports on 10.1.1.1 (10.1.1.1):
(The 65529 ports scanned but not shown below are in state:
closed)

Port        State       Service
22/tcp      open        ssh
259/tcp     open        esro-gen
264/tcp     open        bgmp
265/tcp     open        unknown
900/tcp     open        unknown

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

As you can see, the Telnet and FTP services are no longer running. Only NTP (which listens on UDP port 123), SSH, and the FireWall-1 services are running. If other services are running, they were not properly disabled during the hardening process. The importance of monitoring listening services cannot be overstated. Consequently, Nmap scans should be run against the firewall on a regular basis. In addition, these Nmap scans should be run against all network interfaces of the firewall, not just the internal network interface.

4. Check that the system is logging properly

System logs provide detailed information regarding critical operating system processes, and are instrumental in tracking an attacker in the event of a security breach. First, verify that the log /var/log/authlog is properly logging authorization events. Attempt to login as the root user at the system console, but enter an incorrect password. Note the time of the attempted login. Next, successfully login to the firewall and verify that the failed login attempt was logged with the following command:

```
solaris $ tail /var/log/authlog
```

The /var/log/authlog file should have logged the failed login attempt. Use the timestamps to identify the corresponding entry. Next, verify that the log /var/adm/loginlog is properly logging consecutive failed login attempts. The system adds an entry to this log whenever a user

incorrectly logs in five consecutive times. This log is extremely useful for identifying brute-force password cracking attempts. Attempt to login as the root user five consecutive times, but enter an incorrect password each time. Note the time of the attempted logins. Next, successfully login to the firewall and verify that the failed consecutive login attempts were logged with the following command:

```
solaris $ tail /var/adm/loginlog
```

The /var/adm/loginlog file should have logged the consecutive failed login attempts. Use the timestamps to identify the corresponding entry. Finally, verify that the log /var/adm/sulog is properly logging failed su attempts. The system adds an entry to this log file whenever a user fails to correctly su to another user. This log is useful for identifying password cracking attempts. As a user other than root, attempt to su to root, but enter an incorrect password. Next, successfully login to the firewall and verify that the failed su attempt was logged with the following command:

```
solaris $ tail /var/adm/sulog
```

The /var/adm/sulog file should have logged the failed su attempt. Use the timestamps to identify the corresponding entry. Finally, verify that all log files are configured with the proper ownership and permissions with the following command:

```
solaris $ ls –l /var/log/authlog /var/adm/loginlog /var/adm/sulog
```

All of the log files should be owned by the root user and sys group, and should have a permission of 640, which is "rw-r-----". If not, logging was not properly configured during the hardening process.

5. Check that SSH is functioning properly

For effective remote administration, SSH should be functioning properly. To verify SSH functionality, first install the OpenSSH client on the remote host. For more information regarding the OpenSSH client, including source code and comprehensive documentation, see the OpenSSH website [7]. Next, from the firewall, generate SSH keys for the desired user account (in this case the user "alice"). Generate SSH keys for Alice with the following commands:

```
solaris $ su alice
solaris $ cd
solaris $ ssh-keygen
```

Next, Alice should copy the ~/.ssh/identity.pub file on the firewall to the ~/.ssh/authorized_keys file on the remote host. After that, ensure that the IP address of the remote host appears in the sshd entry of the /etc/hosts.allow file on the firewall. If not, edit this file with the following command:

```
solaris $ vi /etc/hosts.allow
```

Add an appropriate entry for the sshd service. For example, to allow the entire 10.1.1.0 network to access SSH, add the following line:

```
sshd  :  10.1.1.  ALLOW
```

Next, a FireWall-1 rule must be added to allow inbound SSH connections to the firewall. Once this rule is created, Alice will be able to connect to the firewall from the development server, whose IP address is 10.1.1.21:

```
development $ su alice
development $ ssh 10.1.1.1
```

In this case 10.1.1.1 is the IP address of the internal network interface of the firewall, and the remote host has an IP address of 10.1.1.21, which is located on the internal network. If the connection is accepted and Alice is able to login, then SSH is functioning properly. In addition, the warning banner configured in the /etc/issue file should be displayed. If the connection is not accepted or the warning banner is not displayed, SSH was not properly installed or configured during the hardening process. Note that if Alice attempts to connect from an IP address that is not on the 10.1.1.0 network, the connection should be refused. For example, Alice could attempt to connect from an Internet host whose IP address is 192.168.1.21. After copying the ~/.ssh/authorized_keys file to the remote host, the connection is attempted:

```
remote $ ssh 192.168.1.1
```

In this case 192.168.1.1 is the IP address of the external network interface of the firewall. Since the source of the connection is an IP address that is not located on the 10.1.1.0 network, the connection should be refused.

References

1. Various authors. "Sun Enterprise 250 Server." 1 November 2002. http://www.sun.com/servers/entry/250/index.html.
2. Various authors. "Solaris 8 Operating Environment." 1 November 2002. http://wwws.sun.com/software/solaris/8/index.html.
3. Various authors. "SunSolve Online Patches." 1 November 2002. http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access.
4. Various authors. "Solaris Security Digest Tips." 1 November 2002. http://www.boran.com/solarisdigest/solaristips20010706.html - Whacking%20SUIDs.
5. Various authors. "Freeware For Solaris." 1 November 2002. htttp://www.sunfreeware.com/.

6. Venema, Wietse. "TCP Wrappers." 1 November 2002. ftp://ftp.cerias.purdue.edu/pub/tools/unix/netutils/tcp_wrappers/.
7. Various authors. "OpenSSH." 1 November 2002. http://www.openssh.com/.
8. Various authors. "Tripwire Open Source Project." 1 November 2002. http://tripwire.org/.
9. Various authors. "Fix–Modes." 1 November 2002. http://www.sun.com/blueprints/tools/FixModes.tar.Z.
10. Kornacki, Vince. "GCFW Practical v.1.7." 1 November 2002. http://www.giac.org/GCFW.php.
11. Various authors. "Check Point Support." 1 November 2002. http://support.checkpoint.com/login.html.
12. Pomerantz, Hal. "Solaris Security Step-by-Step." 1 February 2001.
13. Fyodor. "Nmap Security Scanner." 1 November 2002. http://www.insecure.org/nmap/.
14. Various authors. "Nessus Security Scanner." 1 November 2002. http://www.nessus.org/.
15. Various authors. "BugTraq." 1 November 2002. http://online.securityfocus.com/archive/1.
16. Various authors. "FireWall-1 Mailing List." 1 November 2002. http://www.checkpoint.com/services/mailing.html.
17. Various authors. "Sun Patch Check." 1 November 2002. http://sunsolve.sun.com/pub-cgi/show.pl?target=patchk.
18. Various authors. "Patch Cross-Reference File." 1 November 2002. http://sunsolve.sun.com/pub-cgi/patchDownload.pl?target=patchdiag.xref&method=H.