Securing Sun Linux Systems: Part II, Network Security

By Glenn Brunette, Michael Hullhorst, and Gé Weijers

Sun BluePrints™ OnLine—July 2003

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 817-3421-10
Revision 1.0, 6/26/03

Please
Recycle

Securing Sun Linux Systems: Part II, Network Security

This article is the second part of a two-part article that provides recommendations for securing the Sun™ Linux 5.0 operating system. This part provides recommendations for network security. Part I provides recommendations for securing local access and file systems. The information in this article applies only to the Sun Linux 5.0 distribution, although some techniques or recommendations might apply to other Linux distributions.

Sun Linux 5.0 is based on the Red Hat 7.2 Linux Distribution and is a flexible, general purpose operating system. To secure a Sun Linux system against unauthorized access and modification requires changes to its default configuration. Although these changes are, in most cases, relatively minor, we strongly recommend that you make these changes to improve the security posture of a system. The changes and recommendations in this article address the majority of methods that intruders use to gain unauthorized or privileged access to Sun Linux systems. You can implement many of the changes during or immediately after system installation.

As with most security strategies, a balance must be achieved between system manageability and security. Some recommendations in this article might not apply to your environment. Removing some services as recommended in this article might negatively impact your ability to manage a system. You must know your system and security requirements before starting. Implementing the changes recommended in this article requires planning, testing, and documentation.

This article contains the following topics:

Securing Network Services

Secure network services and configure network parameters for better security. The network services a system provides are entry points that can be attacked and exploited to gain access. It is important to understand the default configuration of Sun Linux, including its services and the methods used to disable or protect them. Often, organizations need to use protocols or services that are not secure. Where possible, we provide recommendations for improving the security of those services.

Network services enable distributed computers and their users to communicate, access remote systems and information, transfer files, send electronic mail, print on network printers, and manage remote systems. Multi-user operating systems such as Sun Linux typically provide many network services. These services are either necessary for the operation and management of an application or are essential to the service an application provides.

By default, Sun Linux only runs the `portmap`, `rpc.statd`, and `sshd` services. This default improves the overall security posture of a system by limiting the number of unnecessary services that are started on a system. In general, all network services that are not required to meet a business requirement should be disabled.

Protect services that are offered by a system by using as many layers as feasible. The most secure systems are not necessarily those with the most layers. The layers added must support the practice of defense-in-depth by implementing complementary and mutually reinforcing security controls. The layers must adhere to the principle of proportionality, whereby the costs of the controls are weighed against the value of the assets being protected. In this article, we describe layers at the network level, such as filtering and logging options, available in both `xinetd` and TCP Wrappers.

Network services can be attacked in many ways. These services might contain programming flaws, use weak or no authentication, transfer sensitive data in unencrypted format, or allow connections from any network host. These weaknesses allow attackers to access entry points and potentially exploit weaknesses in the installation or configuration.

Some simple methods are available to reduce the risk of attacks against systems. In addition to disabling unneeded services and applying all security patches, use security features (for example, encryption, strong authentication, etc.) with network services whenever possible. You can deploy firewalls like `iptables` on servers and desktops alike where IP forwarding is not required, but network service protection is. Massive deployments and management of firewalls on many systems can be burdensome, so plan appropriately.

Additionally, there are well-regarded open source and commercial tools that add protection. These tools address security concerns by providing the following protection: access control, logging, strong authentication, and privacy through encryption.

The open source toolkit OpenSSH is a suite of tools to replace UNIX® network commands such as `telnet`, `ftp`, `rlogin`, `rsh`, and `rcp`. OpenSSH provides strong authentication and privacy through the use of public-key authentication and support of PAM. Also, OpenSSH provides the capability to securely tunnel X Window network communications over an encrypted channel. When built with the TCP Wrappers library, as is the case in Sun Linux, it provides additional access control. It is a very valuable tool because of the unsafe commands it replaces. After deploying OpenSSH, disable the replaced network services in favor of OpenSSH services.

The OpenSSH tool is very powerful and highly configurable. As a result, it is essential that the tool be configured to be as secure as possible. For recommendations on OpenSSH configuration and tuning, refer to the Sun BluePrint Online articles titled “Configuring the Secure Shell Software” and “Integrating the Secure Shell Software.”

To secure network services, perform the following tasks:

Enable and Disable Network Services

Network services are generally started either at boot time by `/sbin/init` or on demand by the `xinetd` daemon.

Sun Linux provides the general purpose command `chkconfig` to enable and disable services that are started by `init` and `xinetd`. Using `chkconfig` is easier and less error prone than editing configuration files.

To enable or disable a service started by `init`, use the following command.

```
# chkconfig --level <runlevels> <service> [on | off]
```

To enable the Apache Web Server to run on levels 3, 4, and 5, use the following command.

```
# chkconfig --level 345 httpd on
```

This command configures the `httpd` service to start at boot time for run levels 3, 4, and 5 only. This command only enables or prevents a service from starting; it does not actually start the service itself. To start the service manually, use the following command.

```
# /etc/rc.d/init.d/httpd start
```

Services started from `xinetd` are managed in a similar way. The exception is that the run level should not be specified, as it is irrelevant to services managed by `xinetd`. By default, the `xinetd` service is automatically started at run levels 4 and 5.

To enable Telnet, use the following command.

```
# chkconfig telnet on
```

This command enables Telnet and reloads the `xinetd` daemon, which makes Telnet available immediately.

Limit Access to Services Managed by `xinetd`

Sun Linux uses the `xinetd` daemon to control availability to minor network services. All network services are disabled by default. Services that are required to support a business requirement should be enabled as appropriate.

Use the `/sbin/chkconfig` command to control `xinetd` services. The format and operations are the same for `init` services. When listing services using `chkconfig`, the `init` services are listed first followed by the services controlled by `xinetd`.

An ideally secured server often does not run `xinetd`, because the daemons started in the `/etc/xinetd.conf` are frequently not needed.

The `xinetd` daemon provides a number of facilities that are useful for limiting and monitoring access to the services it manages. Access can be limited by network address, time, and other parameters.

▼ To Change a Managed Service

1. **Edit the `/etc/xinetd.d/<`*service*`>` file, where <*service*> is the name of the network service to be customized.**

2. **Use the `chkconfig` command to switch the service on.**

   This command sends a USR2 signal to the `xinetd` process to reread its configuration files. Note that the signal differs from the signal used to reconfigure `inetd`.

   Address-based access control is provided through two configuration parameters: `only_from` and `no_access`. Both parameters specify an access list, which can contain IP addresses and ranges, host names, domain names, and other parameters. Refer to the `xinetd.conf`(5) manual page for details.

   Of the daemons started from the `/etc/xinetd.d` directory, the Telnet, FTP, and standard `r*` commands are described in the following sections. In Sun Linux, these services are disabled by default.

   For restricted access servers, all connections to services managed by `xinetd` should be logged. Logging is enabled by default in Sun Linux.

Sun recommends that you review and consider the use of the `xinetd` options listed in TABLE 1. You can use them to enforce additional policies such as limiting access times. Also, they allow for binding sockets to individual network interfaces.

TABLE 1 `xinetd` Options

Option	Description
`only_from`	Specifies a list of host names, IP addresses, and/or IP address ranges that are allowed access to this service. Packets originating from one of the permitted addresses or address ranges are permitted, while all others are denied access to this service. Note that, especially in the case of UDP services, a packet with a spoofed source address is permitted to access the service. The `xinetd` daemon filters UDP services as well as TCP services, unlike TCP Wrappers, which only filter the first packet to a UDP server. However, `xinetd` runs a separate process that acts as filter, which is a relatively inefficient process and should be limited to low-traffic services.
`no_access`	Specifies a list of host names, IP addresses, and/or IP address ranges that are not allowed access to a service. If a source address matches an entry on both `only_from` and `no_access` lists, the most-specific entry takes precedence.
`banner_fail`	References a file containing contents sent over a TCP data connection that does not meet the access criteria. The connection is closed immediately after the data is sent.
`bind`	Listens only for connections on a specified interface address. We strongly recommend that you use this capability to configure services to listen only on those interfaces where the service is required. If a service is not required to run on a specific network, do not make it available. For example, consider binding management services to listen for connections only on management networks.

The following example shows a possible configuration for the Telnet service.

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
        only_from       = 192.168.1.0/24
        bind            = 192.168.1.50
        banner_fail = /etc/telnet_banner
}
```

In this example, the Telnet service is configured to bind to the system's IP address 192.168.1.50 and to accept connection requests only from systems on the same network, 192.168.1.0/24. In addition, the banner message defined in `/etc/telnet_banner` is displayed to any user attempting to connect to this service from an IP address not on the 192.168.1.0/24 network.

Enable TCP Wrappers

The TCP Wrappers tool is an open source tool, developed by Wietse Venema, that provides a flexible configuration mechanism for controlling incoming connections based on pattern matching for hostnames, DNS domains, and network addresses. With Sun Linux, this functionality is integrated into the OS and provides protection for most network services, even those started by `xinetd`. The tool provides a good logging capability and detects DNS hostname discrepancies that can indicate an attack.

TCP Wrappers are enabled by default on Sun Linux. TCP Wrappers are configured into a number of services, which include `ssh`, `portmap`, and every service managed by `xinetd`. A simple TCP Wrapper configuration could have the following configuration in its `/etc/hosts.allow` and `/etc/hosts.deny` files.

```
# cat /etc/hosts.allow
ALL:  LOCAL 10.8.10.0/255.255.255.0 10.8.11.0/255.255.255.0
portmap: 10.8.31.100
sshd: 10.0.0.0/255.0.0.0 192.168.0.0/255.255.0.0
# cat /etc/hosts.deny
ALL: ALL
```

This sample configuration restricts access to all services using TCP Wrappers to the local subnet and two class C IP ranges (10.8.10.0 and 10.8.11.0). In addition, `ssh` and `portmap` allow other IP addresses to connect. For more information about TCP Wrapper capabilities, refer to the `tcpd`(8) manual page.

Sun Linux uses `xinetd` as a replacement for the venerable `inetd` managed network services daemon. The `xinetd` provides facilities that are similar to TCP Wrappers, and is generally much more flexible than `inetd`. Unlike TCP Wrappers, `xinetd` can provide protection for UDP services by intercepting all packets and matching them with access control lists before passing them on. Also, `xinetd` can perform rate limiting and provides many logging options. On Sun Linux, `xinetd` has support built in for TCP Wrappers. It is not necessary to use the `tcpd` binary.

Enable Packet Filtering

Linux kernels based on version 2.4 and newer, such as Sun Linux 5.0, come equipped with a built-in packet filter called *netfilter*, that is also referred to by the name of the command that controls it: `iptables`. Netfilter supports all the features commonly found in modern IP packet filters, including stateful inspection filtering.

A stateful inspection filter such as netfilter uses knowledge of packets that were previously seen as a factor in its filtering decisions. It is no longer necessary to pass all TCP packets that have the ACK flag set; DNS responses can be matched to requests. It is only necessary to add rules for the initial packet of a session, then the stateful inspection mechanism takes care of the remaining traffic. Rule sets become more concise, and there is less interference between rules for different protocols, which is a common problem with static packet filters.

Netfilter also supports application level gateways (ALGs). These are rules that handle problem protocols such as FTP that negotiate additional connections, which have to be passed by the filter. The ALGs inspect the contents of packets to determine which additional connections to expect. Only a few protocols are supported at this time, however.

Netfilter is an excellent tool for constructing host-based firewalls to protect against attacks from other hosts, not just those outside a perimeter firewall. They provide an additional layer of defense to protect against unneeded services being enabled by mistake, or when other security tools fail and are circumvented. Always practice defense in depth where possible.

More information on the netfilter tool including source code, how-to documents, and tutorials are available at: `http://www.netfilter.org/`.

An exhaustive list of netfilter and iptables related material is available at: `http://www.linuxguruz.com/iptables/`.

Secure Telnet Connections

Telnet is a user-interactive service for accessing remote systems on a network. Unfortunately, this service provides little in the way of security. By default, the only authentication information required is user name and password. Neither of these items are encrypted while in transit, and are, therefore, vulnerable to a variety of attacks including: man in the middle attack, session hijacking, and network sniffing. Tools implementing the Secure Shell protocol can serve as an effective replacement and are strongly recommended.

By default, the Telnet service is disabled in Sun Linux. If this service must be used, then consider using strong authentication mechanisms such as Kerberos, One-time passwords, tokens, or other methods. In addition, consider restricting access to the Telnet service using `xinetd` filtering, TCP Wrappers, or host-based firewalls. That way, the risks associated with running a nonsecure service are reduced by limiting who can access it.

Host-based firewalls and TCP Wrappers limit the hosts that may connect to a system. By restricting access to services based on IP addresses, a system can limit its exposure to network attacks. Note that firewalls do not prevent sniffing of Telnet connections.

Caution – One-time passwords and filtering access do not protect the contents of a session from being disclosed through network sniffing. In addition, be aware that these alternatives typically do not protect a session against being hijacked by a malicious user. The malicious user, in effect, can take over a session from an authorized user.

Secure Remote Access Connections

Access control and accountability are critical to the security of a system. Access control should involve strong authentication for system access, while accountability information should provide tracking data relative to system changes.

The standard `r*` commands (`rsh`, `rlogin`, and `rcp`) break both of these recommendations, because most implementations of `r*` commands involve zones of trust. Within a zone of trust, all systems are trusted and no additional authentication is required. Hence, an intruder need only gain access to one server to gain access to all servers.

The default authentication mechanism of the `r*` commands uses the hostname or IP address of a system and a user ID for authentication. No additional authentication is required. Considering the ease with which an IP address and user ID can be stolen or misused, this default is clearly not a secure mechanism. We recommend that you do not use the `r*` commands in this manner, and that you do not allow servers to offer a service in this manner.

Where possible, the use of the `r*` commands should be replaced with a more secure alternative such as Secure Shell. There might be times when this is not possible. In those cases, it is recommended that the host-based authentication mechanism used by the `rlogin` command be disabled.

▼ To Disable Host-Based Authentication

- **Comment out the** `pam_rhosts_auth.so` **entry in** `/etc/pam.d/rlogin` **file as follows:**

```
#%PAM-1.0
# For root login to succeed here with pam_securetty, "rlogin" must
be
# listed in /etc/securetty.
auth       required /lib/security/pam_nologin.so
auth       required /lib/security/pam_securetty.so
auth       required /lib/security/pam_env.so
#auth       sufficient   /lib/security/pam_rhosts_auth.so
auth      required /lib/security/pam_stack.so service=system-auth
account   required /lib/security/pam_stack.so service=system-auth
password  required /lib/security/pam_stack.so service=system-auth
session   required /lib/security/pam_stack.so service=system-auth
```

Shown in bold text, this change forces `rlogin` to prompt for a password. The `rsh` protocol does not allow for password authentication, and should therefore always be disabled. Note that the pound sign (#) is added to the line.

Secure FTP

The `ftp` daemon has many of the same security issues as the `telnet` daemon. By default, all authentication information transmitted over a network is in clear-text, in much the same fashion as the Telnet protocol. This exposes the FTP protocol to many of the same attack scenarios as Telnet, including man in the middle, session hijacking, and network sniffing. For these reasons, consider alternatives to FTP when FTP transport functionality is required.

OpenSSH provides a secure alternative to FTP, providing most of the functionality of command-line based FTP access.

The Washington University FTP (`wu-ftp`) service is provided in Sun Linux. This implementation of the FTP service supports several security enhancements not commonly found in its predecessors. In particular, the `wu-ftp` service supports:

- Enhanced access control using the `/etc/ftpaccess` file. Use this file to restrict access to specific users and restrict access based on group membership, source address or network, and user classes.
- Extended logging capabilities that permit you to log connection events, individual FTP commands, individual file transfers, upload or download, and security rule violations as defined by the FTP configuration.
- Explicit definition of the control and data ports used by the FTP server. This capability is very useful for controlling FTP access in firewall environments.
- Containment using the `chroot`(1M) facility. This facility allows a service to be configured to run in a contained area, such as a user's home directory, thereby limiting the file system objects a user can access.

For more information on these capabilities and other configuration options provided by this service, refer to `ftpaccess`(5).

Sun Linux includes a fully configured set of directories for anonymous access. A default `/etc/ftpusers` file is included as well. In this file, specify all accounts *not* allowed to use the incoming FTP service. At a minimum, include all system accounts (for example, `games`, news, `uucp`, and so forth) in addition to the root account.

Frequently, intruders use FTP with these accounts to gain unauthorized access. Commonly, root access to a server over Telnet or Secure Shell is disabled while root FTP access is not. This configuration provides a backdoor for intruders who might modify a system's configuration by uploading modified configuration files, thereby accessing a system remotely.

The second security feature of the `in.ftpd` daemon is the ability of the daemon to log IP addresses of all connections and commands issued to the `ftp` daemon through the `syslog` service. Logging is enabled with the `-l` option. Commands issued to the `ftp` daemon are logged when the `-d` option is used. By logging FTP connection requests and commands to a log server for parsing, you can track and resolve unauthorized access attempts.

Caution – Using the `-d` option has the potential for logging passwords that are erroneously entered at the login prompt. It is important that the log files be sufficiently protected to prevent the disclosure of this sensitive information.

The `wu-ftp` daemon can change the banner message shown before login. Use this daemon to hide from potential attackers the version of the FTP server used.

▼ To Change Banner Messages for Incoming FTP Connections

1. **Review the `/etc/ftpaccess` file to determine if the following entry is present:**

```
banner=/etc/ftpd/banner.msg
```

2. **If the entry is not present, add it.**

3. **Replace the contents of the `/etc/banner.msg` file with a line similar to the following, as appropriate for your environment:**

```
Authorized Use Only
```

Secure the Remote Procedure Call (RPC) Services

The Remote Procedure Call (RPC) mechanism provides a way for network services to communicate and make procedure calls on remote systems. When a new RPC service is started, it registers with `portmap`, the central RPC service agent. The `portmap` maintains a table of RPC services (listed by program number) and the network addresses on which they listen for clients to connect. A client first communicates with the `portmap` service to determine the network address it must use to contact an RPC service. You can list current RPC services by using the `rpcinfo` command, which communicates with the `portmap` service.

Caution – Typically, the RPC services available in most Linux distributions are not secure. This situation is not true of other operating systems, such as the Solaris OE, that implement protocols such as AUTH_DH and RPCSEC_GSS and have the ability to restrict RPC usage to a local system using RPC that is implemented over TLI. For more information on AUTH_DH, RPCSEC_GSS, and TLI, refer to the Solaris documentation available at `http://docs.sun.com/`. Given that the Sun Linux operating system does not support these capabilities, unless RPC-based services are required to support a business function, such as accessing or sharing files via NFS, then disable them.

▼ To Disable the RPC `portmap` Service

- **Use the following commands together:**

```
# /sbin/chkconfig -level 0123456 portmap off
# /etc/rc.d/init.d/portmap stop
```

The first command prevents the RPC `portmap` service from starting at boot time and the second command stops any currently running instances of the `portmap` service.

The second command illustrates how you can disable a service without the need for a reboot. You could omit the second command, but then a system reboot would be required to completely disable the service.

Note – This approach to disabling services applies to any services managed by the `chkconfig` command. Use this command to either allow or prevent a service from starting at boot time. It does not start or stop services on a running system.

Caution – Be aware that stopping the `portmap` service does not directly prevent RPC services from running on a system. In particular, if an RPC service is configured to listen on a fixed port, such as through `xinetd`, then it can still be accessed. Be sure to review the contents of the `/etc/xinetd.d` directory for services with type parameters that include RPC. For these services, disable or restrict access to them.

Disable or Secure `automount` Services

The automount service automates the process of mounting file systems and devices without requiring that users have administrative privileges on a system. This process is accomplished using the kernel level autofs service with the automountd system daemon. File systems mounted by this service are automatically unmounted after a predefined period of inactivity.

In Sun Linux, the automount service is capable of mounting not only NFS shares but also removable media such as CD-ROMs and diskettes. Typically, the automount service is used heavily in environments that use naming services such as NIS or LDAP, because remote file system information used by this service can be easily stored in these directories. It can be used on a standalone system as well, where there might be a requirement for automatically mounting local removable media.

As with any service, if the automount service is not required to support business functions, it should be disabled or removed.

▼ To Disable `autofs`

- **Use the following command:**

```
# /sbin/chkconfig -levels 0123456 autofs off
```

▼ To Remove `autofs`

Use the following command:

```
# rpm -e autofs
```

If the automount service is required, then take steps to ensure that its configuration is as secure as possible. First, ensure that the automount service is configured to obtain its information from the correct naming service (for example, files, NIS, LDAP, etc.). Next, ensure that the file systems listed in the automount configuration are mounted appropriately.

For example, the `/etc/auto.misc` file can be configured to mount various forms of removable media. If this file is configured improperly, a user could mount a diskette, formatted as a Linux (ext2) file system, that contains set-user-ID binaries. These binaries could then be run, giving the user added privileges and possibly allowing the entire system to be breached. To prevent this situation, ensure that only valid devices are listed and that mount options are added, where appropriate, to the automount service configuration files such as `/etc/auto.master` and `/etc/auto.misc`.

In the next example, the keywords `nosuid` and `nodev` are added to the `/etc/auto.misc` file for the `cd` and `floppy` resources. This action instructs the system to ignore set-user-ID or set-group-ID bits set on any files mounted on the devices, and to not interpret any character or block special devices stored on them as well. This change is made because the Sun Linux NFS implementation does not currently support NFSSEC. Consequently, the NFS services are not able to use strong authentication or encryption such as is provided by Kerberos.

```
cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
floppy -fstype=auto,nosuid,nodev :/dev/fd0
```

Note – These options are part of the standard configuration provided by Sun Linux. For other mount options, refer to the `mount`(8) manual page.

Caution – Some mount options are specific to certainly file system types. Always be sure to use the correct options for the file systems that you are using.

Secure the NFS Services

A Sun Linux system can be an NFS server, an NFS client, both, or neither. From a security perspective, the best option is to neither provide NFS services nor accept them from any other systems as long as they are not required.

If both NFS server and client services are not required, they can be uninstalled by removing the `nfs-utils` package as follows:

```
# rpm -e nfs-utils
```

If either the NFS client or server services are needed, then this package should not be disabled because both the client and server software is contained in the same package.

By default, the NFS client service is enabled, whereas the server service is disabled.

▼ To Disable the NFS Client Service

- **Use the following commands:**

```
# chkconfig --level 0123456 nfslock off
# /etc/rc.d/init.d/nfslock stop
```

The first command prevents the NFS client service from starting at boot time. The second command stops the NFS client service if it is already running on a system.

Frequently, business requirements require an NFS server. If this is the case, ensure that the NFS server configuration is as secure as possible.

▼ To Improve Security of the NFS Server Configuration

1. **Explicitly list hosts allowed access to NFS server directories.**
2. **Do not open access to all systems.**
3. **Export only the lowest directory necessary.**

4. **Never share more data than is needed to meet the business requirement.**

5. **Export read-only wherever possible. In addition, consider the use of other flags such as** `secure` **and** `root_squash`**.**

   For more information on these options, refer to the `exports`(5) and `mount`(8) manual pages. The NFS server and various mechanisms available to secure it encompass more material than can be covered here.

Secure the `sendmail` Services

The `sendmail` daemon forwards and receives mail from other systems. You should use centralized mail servers to receive mail instead of using local servers. However, allow local systems to generate outgoing mail and forward it to other servers.

Ideally, a more secure Mail Transport Agent (MTA) should be used instead of the MTA bundled with Sun Linux. The `sendmail` daemon bundled with Sun Linux has been subject to denial of service, buffer overflow, and misconfiguration attacks. Alternative MTAs with smaller and more robust code are available. These other MTAs are more security conscious and if configured properly, compromise the security of the server less than `sendmail`.

If `sendmail` must be used as the Mail Transfer Agent (MTA), then refer to recommendations made at SendMail Consortium, in the Sendmail O'Reilly publications, and through the SunSolve OnLine^{SM} service. There are a wide variety of sendmail versions in use, and there are differences in the associated `sendmail.cf` configuration files. Because of this, a sample `sendmail.cf` file is not included with this article.

If `sendmail` is not required or only outgoing `sendmail` is required, we recommend that you remove, disable, or enable only outgoing `sendmail`. The following sections provide instructions and recommendations.

▼ To Remove or Disable sendmail

If no sendmail functionality is required to support a business need, remove or disable it. Base your decision on the need to provide outgoing mail from a system. If this capability is needed, then do not remove the sendmail software.

- To remove the sendmail software, use the following command:

```
# rpm -e sendmail sendmail-cf sendmail-doc
```

- To disable the sendmail service, use the following commands:

```
# chkconfig --level 0123456 sendmail off
# /etc/rc.d/init.d/sendmail stop
```

The first command prevents the sendmail service from starting at boot time. The second command stops the sendmail service if it is already running on a system.

▼ To Allow Only Outgoing sendmail

The sendmail daemon is not needed for email delivery to other systems. All messages that can be are immediately delivered. Messages that cannot be immediately delivered are queued for future delivery. The sendmail daemon, if running, retries to deliver these messages again.

- **To enable only outgoing sendmail, use a cron job to start sendmail every hour to process undelivered messages.**

  The following cron entry starts sendmail every hour to flush the mail queue:

```
0 * * * * /usr/lib/sendmail -q
```

Configure Name Service Caching

The name service cache daemon `nscd` provides caching for name service requests. It exists to provide a performance boost to pending requests and to reduce name service network traffic. By default, the `nscd` package is installed, but the `nscd` daemon is disabled. This section applies only if you need to enable the `nscd` daemon.

The `nscd` daemon maintains cache entries for databases such as `passwd`, `group`, and `hosts`. For security reasons, it does not cache the shadow password file. All name service requests made through system library calls are routed to `nscd`.

Because caching name service data makes spoofing attacks easier, we recommend that you modify the configuration of `nscd` to cache as little data as possible. This task is accomplished by setting the positive `ttl` to zero in the `/etc/nscd.conf` file for the name service requests deemed vulnerable to spoofing attacks. In particular, modify the configuration so that `passwd` and `group` have positive and negative `ttl` values of zero.

Note – There might be a performance impact on systems that use name services intensively.

Use the `nscd -g` option to view the current `nscd` configuration on a server. This option is a helpful resource when tuning `nscd`.

Secure Print Services

When a Sun Linux system is installed, the line printing package is installed, but needs to be configured. By default, the `lpd` daemon is not started.

▼ To Secure Print Services

1. **To ensure that line printer services are not started, disable the package at startup by using the following command.**

```
# /sbin/chkconfig --level 0123456 lpd off
```

2. **If printing services are not required for the system, remove the package by using the following command:**

```
# rpm -e LPRng
```

Display Access Warnings

Displaying access warnings allows companies to pursue full legal recourse if a system is accessed or compromised by unauthorized users. These warnings contain messages about inappropriate and unauthorized use of a system. They generally warn users that their sessions and accounts might be monitored for illegal or inappropriate use. Consult your legal counsel for requirements and verbiage.

The contents of the `/etc/issue` file are displayed for local and serial access. The contents of `/etc/issue.network` are displayed for incoming Telnet connections. Also, you can use the message of the day `/etc/motd` file.

The following is an example warning.

```
This system is for the use of authorized users only.
Individuals using this computer system without authority, or in
excess of their authority, are subject to having all of their
activities on this system monitored and recorded by system
personnel.

In the course of monitoring individuals improperly using this
system or in the course of system maintenance, the activities
of authorized users may also be monitored.

Anyone using this system expressly consents to such monitoring
and is advised that if such monitoring reveals possible
evidence of criminal activity, system personnel may provide the
evidence of such monitoring to law enforcement officials.
```

If `sendmail` service must be configured to run on a system, then consider modifying the banner message that is displayed to clients connecting to a service on TCP port 25. By default, the banner includes host name and version information that might be used by attackers and automated vulnerability scanners. Changing the banner text does not impact the ability of the service to function.

▼ To Change the Banner Message

1. **Add the following line to the `/etc/mail/sendmail.mc` file and rebuild the configuration file:**

```
define(`confSMTP_LOGIN_MSG', ``Mail Server Ready'')dnl
```

2. **Place the line after the `include` directive in this file, and before any** FEATURE **lines.**

   If you modify the contents in `/etc/sendmail.cf` file, which is generated from the `/etc/mail/sendmail.mc` file, all changes are lost at the next generation cycle.

3. **To generate the `/etc/sendmail.cf` from the `/etc/mail/sendmail.mc` file, do the following:**

```
# m4 /etc/mail/sendmail.mc > /tmp/sendmail.cf
```

Turning Kernel Network Parameters

Configure kernel variables to improve network security. Some changes might cause a system to not strictly comply with relevant RFCs and might require testing before being placed on production systems.

Configure kernel variables by performing the following tasks:

For more information about IP forwarding, source routing, broadcast ICMP ECHO packets, and source route verification, refer to the detailed descriptions in the Sun BluePrint OnLine article titled "Solaris Operating Environment Network Settings for Security: Updated for Solaris 9 Operating Environment."

Configure IP Forwarding

During startup, the `/etc/sysctl.conf` file is read by the `sysctl` command. This file contains settings for kernel parameters.

You can disable or enable IP Forwarding by assigning `net.ipv4.ip_forward` to the kernel parameter. Assigning a 0 disables and assigning a 1 enables forwarding. Be aware of the following:

- Assigning a 0 resets all IPv4-related variables to conform with RFC 1122 (requirements for Internet hosts–communication layers).
- Assigning a 1 resets all variables to conform with RFC 1812 (requirements for IP version 4 routers).

Note – It is important to set this kernel parameter first, before changing any other related parameters.

▼ To Disable or Enable IP Forwarding

- To disable IP forwarding, set the parameter in the `/etc/sysctl.conf` file as follows:

```
net.ipv4.ip_forward = 0
```

- To enable IP Forwarding, set the parameter in the `/etc/sysctl.conf` file as follows.

```
net.ipv4.ip_forward = 1
```

Disable Source Routing

Source routing has been used in attacks, and legitimate uses of source routing are few. It is a good idea to discard all packets that use source routing, unless you have a specific need for them.

▼ To Disable Source Routing

- **Add the following lines to the** `/etc/sysctl.conf` **file:**

```
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.accept_source_route = 0
```

The first line disables source routing on all current interfaces. The second line provides a default for any new interfaces that might be configured later.

Ignore Broadcast ICMP ECHO Packets

Many operating systems respond to ICMP ECHO (ping) packets that are sent to the network broadcast address. This behavior has been used to mount denial-of-service attacks by causing all hosts on a network segment to send ICMP REPLY packets to a host under attack. Our advice is to disable this behavior.

▼ To Disable Echo Broadcasts

- **Add the following line to the `/etc/sysctl.conf` file:**

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

Log Invalid Addresses

When a kernel receives packets with obviously invalid addresses, they are discarded.

▼ To Log Invalid Addresses

- **Add the following two lines to the** `/etc/sysctl.conf` **file.**

```
net.ipv4.conf.default.log_martians = 1
net.ipv4.conf.all.log_martians = 1
```

Configure ICMP Redirect Messages

ICMP redirect messages are used by network gateways to inform a host sending data to forward packets to a different gateway. If a Sun Linux system is not configured to act as a gateway, that is the `net.ipv4.ip_forward` parameter is set to 0, then the system should never need to send ICMP redirect messages.

To configure the system to never send ICMP redirect messages, add the following line to the `/etc/sysctl.conf` file.

```
net.ipv4.send_redirects = 0
```

Similarly, if you only have one gateway on the network to which the host is attached, then it is safe to ignore any incoming ICMP redirect messages. These messages could not be generated in such a case, because there is only one path out of the network.

An attacker can forge redirect messages to install bogus routes. This action might initiate a denial of service attack if a newly specified router is not a router at all. Similarly, this technique could be used to force network packets to be routed through an attacker's machine, where the packets could be inspected, captured, or modified. Although there are rules governing valid ICMP redirect messages, all of them can be easily spoofed.

If possible, configure the system to ignore ICMP redirect messages by adding the following line to the `/etc/sysctl.conf` file.

```
net.ipv4.accept_redirects = 0"
```

Use Source Route Verification

The Sun Linux source route verification mechanism verifies that a packet comes in on an expected network interface. The routing table is consulted for each incoming packet. The interface the packet comes in on must match the interface that would be used to reach the source of the packet. If these interfaces do not match, the packet is discarded. This feature is enabled by default.

Source route verification adds overhead to packet processing and might not work in environments where asymmetric routing occurs. Source route verification is controlled by the following parameters.

- `net.ipv4.conf.default.rp_filter`
- `net.ipv4.conf.all.rp_filter`

Our recommendation is to leave it enabled unless it causes performance or routing problems.

▼ To Disable Source Route Verification

- **Add the following lines to the** `/etc/sysctl.conf` **file:**

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
```

Disable Protocol Stacks

The kernel supports on-demand loading of kernel modules. Many components of the kernel such as file systems, protocol stacks, and device drivers are loaded the first time they are needed. The process of loading a required kernel module is quite simple: the kernel creates a new user process and runs the program `/sbin/modprobe`, which loads the required module.

This simple and effective mechanism has one drawback: protocol stacks other than TCP/IP might be loaded inadvertently. When a network socket is created for protocol family *N*, the kernel executes the following command.

```
# /sbin/modprobe -s net-pf-N
```

The protocol stack is loaded. We recommend that you disable all unnecessary protocol stacks.

▼ To Disable All Unnecessary Protocol Stacks

- **Modify the following lines to the** `/etc/modules.conf` **file:**

```
alias net-pf-4 off # IPX
alias net-pf-5 off # Appletalk
alias net-pf-10 off # IPv6
alias net-pf-12 off # Decnet
```

References and Related Resources

Publications

- Hatch, Brian, and Osborne, James Lee. *Hacking Linux Exposed*, Second Edition McGraw-Hill, ISBN: 0072225645, November 2002.
- Nemeth, Evi, Snyder, Garth, Seebass, Scott, and Hein, Trent R. *UNIX System Administration Handbook*, 3rd Edition, Prentice Hall PTR, ISBN: 0130206016, August 2000.
- Noordergraaf, Alex. "Solaris Operating Environment Network Settings for Security: Updated for Solaris 9 Operating Environment," Sun BluePrints OnLine, June 2003, `http://www.sun.com/solutions/blueprints/0603/816-5240.html`.
- Noordergraaf, Alex and Watson, Keith. "Solaris Operating Environment Security: Updated for the Solaris 9 Operating Environment," Sun BluePrints OnLine, December 2002, `http://www.sun.com/solutions/blueprints/1202/816-5242.pdf`.
- *Red Hat Linux 9: Red Hat Linux Security Guide*, Red Hat Inc., 2002.
- Reid, Jason. "Building OpenSSH - Tools and Tradeoffs," Sun BluePrints OnLine, January 2003, `http://www.sun.com/blueprints/0103/817-1307.pdf`.
- Reid, Jason. "Configuring the Secure Shell Software," Sun BluePrints OnLine, April 2003, `http://www.sun.com/blueprints/0403/817-2485.pdf`.
- Reid, Jason. "Integrating the Secure Shell Software," Sun BluePrints OnLine, May 2003, `http://www.sun.com/blueprints/0503/817-2821.pdf`.
- Stevens, Richard W. *TCP/IP Illustrated, Volume 1*, 1st Edition, Addison-Wesley Publishing Company, ISBN: 0201633469, January 1994.

Web Sites

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other material on or available from such sites or resources. Sun will not be responsible or liable for any damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through any such sites or resources.

- Center for Internet Security - Linux Benchmark `http://www.cisecurity.org/`
- OpenSSH tool: `http://www.openssh.com/`
- Sendmail Consortium: `sendmail` configuration information, `http://www.sendmail.org/`
- SSH Communications Security, Secure Shell (SSH) tool: `http://www.ssh.com/`
- Sun BluePrints: `http://www.sun.com/blueprints`
- TCP Wrappers tool, Wietse Venema: `ftp://ftp.porcupine.org/pub/security/index.html`

About the Authors

Glenn Brunette

Glenn Brunette is a Sun Principal Engineer with over a decade of experience in information security. Glenn works in the Sun Professional Services division as the Americas Lead Security Architect. In this role, he is responsible for the development and execution of the region's security services strategy. He works with teams throughout the Americas and the world to improve the quality and security of services delivered to Sun's customers.

Previously, Glenn worked in the North East and Financial Services Areas developing and delivering a wide array of tailored security solutions supporting the lifecycle of assessment, architecture, implementation, and management. His customers have included major financial services firms, service providers, and life sciences and government organizations. In addition to contract services, Glenn works closely with teams across Sun on the development and delivery of security strategy, methodologies, best practices, training, and tools. Glenn is a co-founder of the very

popular freeware Solaris Security Toolkit software. Glenn is a Certified Information Systems Security Professional (CISSP) and has been trained in the National Security Agency's INFOSEC Assessment Methodology (IAM).

Michael Hullhorst

Michael Hullhorst is a Staff Engineer working within Sun Microsystems as a Lead Security Architect and evangelist for the Sun Linux Security Group. The group is responsible for driving security into Sun's Linux products.

An engineer with three decades of system development experience, Michael has worked in the capacity of Director of Engineering for Progressive Systems, developing Linux based security products including the Phoenix Adaptive Firewall Appliance. Further, Michael has been an independent consultant with a wide range of experience including the development of complex applications and networks, as well as working with embedded systems. Additionally, Michael's background includes being the Director of Information Systems for a large telecommunication provisioning group.

Michael has been trained in the National Security Agency's INFOSEC Assessment Methodology (IAM).

Gé Weijers

Gé Weijers is a staff engineer working for the Sun Linux Security Group in Columbus, Ohio. He specializes in Security Engineering, the construction of systems that perform their intended function in a dependable way.

Recently Gé worked on the design and delivery of software that reduces the vulnerability of Linux-based Sun products to network-based attacks, and he worked on the development of new initiatives within Sun to improve the security of Sun products.

Prior to Sun, Gé worked for Progressive Systems, Inc. of Columbus, Ohio. The main product was the Phoenix Adaptive Firewall Appliance. He designed the cryptographic tools used to allow secure remote configuration of firewalls using a user interface based on Java™ technology.

Some of Gé's professional interests are security engineering, cryptography, protocol design, and provable security.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`