

15.4.3 Network Security Facts

Adhering to general security procedures can simplify and enhance efforts to increase network security,


This lesson covers the following topics.

- General security procedures.
- Network and system security configuration with `/etc/sysctl`

General Security Procedures

General procedures for increasing network security of a Linux system include:

Security Task	Procedure
Remove unneeded software	<p>Unneeded software takes disk space and could introduce security risks. To remove unneeded software:</p> <ol style="list-style-type: none"> 1. Run one of the following commands: <ul style="list-style-type: none"> ◦ Use dnf list installed to see installed RPM packages on the computer. ◦ Use dpkg -get-selections to see installed Debian packages on the computer. 2. Research the function of any unrecognized package to determine whether it is necessary. 3. Use yum, rpm, or dpkg to uninstall unneeded packages.
Check for unneeded network services	<p>Unneeded network services waste the computer's resources and might provide attackers with an entry point for an attack. To view a list of installed services, use one of the following commands:</p> <ul style="list-style-type: none"> • For init-based systems, run chkconfig at the shell prompt. • For systemd-based systems, run systemctl list-unit-files at the shell prompt. <p>Review the output of these commands and look for unusual or unrecognized services. Then use the man command and the internet to determine whether they can be safely removed or disabled. Use chkconfig, insserv, or init to disable the service on init-based systems. On systemd distributions, you can use the systemctl disable or the systemctl mask command to disable a service. Alternatively, you could use yum, zypper, rpm, or dpkg to remove the package entirely.</p>
Locate open ports	<p>Open ports can provide information about what operating system a computer uses and can provide entry points for an attack. To locate open ports:</p> <ol style="list-style-type: none"> 1. Install the nmap utility (if not already installed). 2. Use one of the following commands to scan for open ports: <ul style="list-style-type: none"> ◦ nmap -sT host_IP_address scans for open TCP ports

	<ul style="list-style-type: none"> ◦ nmap -sU <i>host_IP_address</i> scans for open UDP ports <ol style="list-style-type: none"> 3. From the results of the scan, determine which ports to close and which services use the ports. 4. Disable the services using those ports. 5. Consider changing default port assignments to different ports. <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Consider running nmap on the local system as well as from a different network host. This will reveal what ports are open and which services are actually allowed through the host's firewall.</p> </div>
Check network connections	<p>Open network connections (e.g., open sockets) on a computer also create a security risk. A <i>socket</i> is an endpoint of a bidirectional communication flow across a computer network. Use the following netstat options to identify the open network connections on the Linux system:</p> <ul style="list-style-type: none"> • -a lists both listening and non-listening sockets. • -l lists listening sockets. • -s displays statistics for each protocol. • -i displays a table of all network interfaces.

If you have application or hardware issues, check to make sure that you have not blocked an essential component in your efforts to increase security. You can check availability as follows:

- Software: use **dnf list installed** to see install RPM packages or **dpkg -get-selections** to see installed Debian packages.
- Network services: use **chkconfig** for init-based systems. Use **systemctl list-unit-file** for systemd-based systems.
- Ports: use **nmap** as previously described.
- Connections: use **netstat** as previously described.
- Protocols or firewall ACLS: use **iptables -L**
- for the current firewall configuration or **cat/etc/sysconfig/iptables** to read the firewall file.

Network and System Security Configuration with /etc/sysctl

You can harden your Linux network and system setting using **sysctl**. The **sysctl** command is used to modify kernel parameters at runtime. **/etc/sysctl.conf** is a text file containing **sysctl** values to be read in and set by **sysctl** at boot time.

sysctl and settings in **/etc/sysctl.conf** include the ability to:

- Limit network-transmitted configuration for IPv4 and IPv6.

- Turn on Exec Shield buffer overflow protection.
- Protect against syn flood attacks.
- Turn on source IP address verification.
- Protect against a spoofing attack on the IP address of the server.
- Log several types of suspicious packets, such as spoofed packets, source-routed packets, and redirects.

The parameters available for **sysctl** are listed under `/proc/sys/`. Procfs is required for **sysctl** support in Linux. You can use **sysctl** to both read and write **sysctl** data. The following table identifies **sysctl** parameters.

The following table are parameters you can use with **sysctl**.

Parameter	Description
<i>variable</i>	The name of a key to read from. An example is <code>kernel.ostype</code> .
<i>variable=value</i>	Variable is the key and value is the value to set it to. If the value contains quotes or characters which are parsed by the shell, enclose the value in double quotes.
<code>-n</code>	This option disables printing the key name when printing values.
<code>-q</code>	This option prevents displaying the values sent to stdout.
<code>-w</code>	When all arguments prescribe a key to be set, use this option.
<code>-p[file]</code>	This option loads sysctl settings from the file specified or <code>/etc/sysctl.conf</code> if a file name is not given. Using this option indicates arguments to sysctl are files. The files are read in the order they are specified.

The following table identifies some examples of parameters you can set in `/etc/sysctl.conf`

Action	Parameter
Control IP packet forwarding.	<code>net.ipv4.ip_forward = 0</code>
Not accept source routing.	<code>net.ipv4.conf.default.accept_source_route = 0</code>
Ignore all ICMP ECHO and TIMESTAMP requests sent via broadcast/multicast.	<code>net.ipv4.icmp_echo_ignore_broadcasts = 1</code>
Prevent common syn flood attack.	<code>net.ipv4.tcp_syncookies = 1</code>
Enable source validation by reversed path.	<code>net.ipv4.conf.all.rp_filter = 1</code>