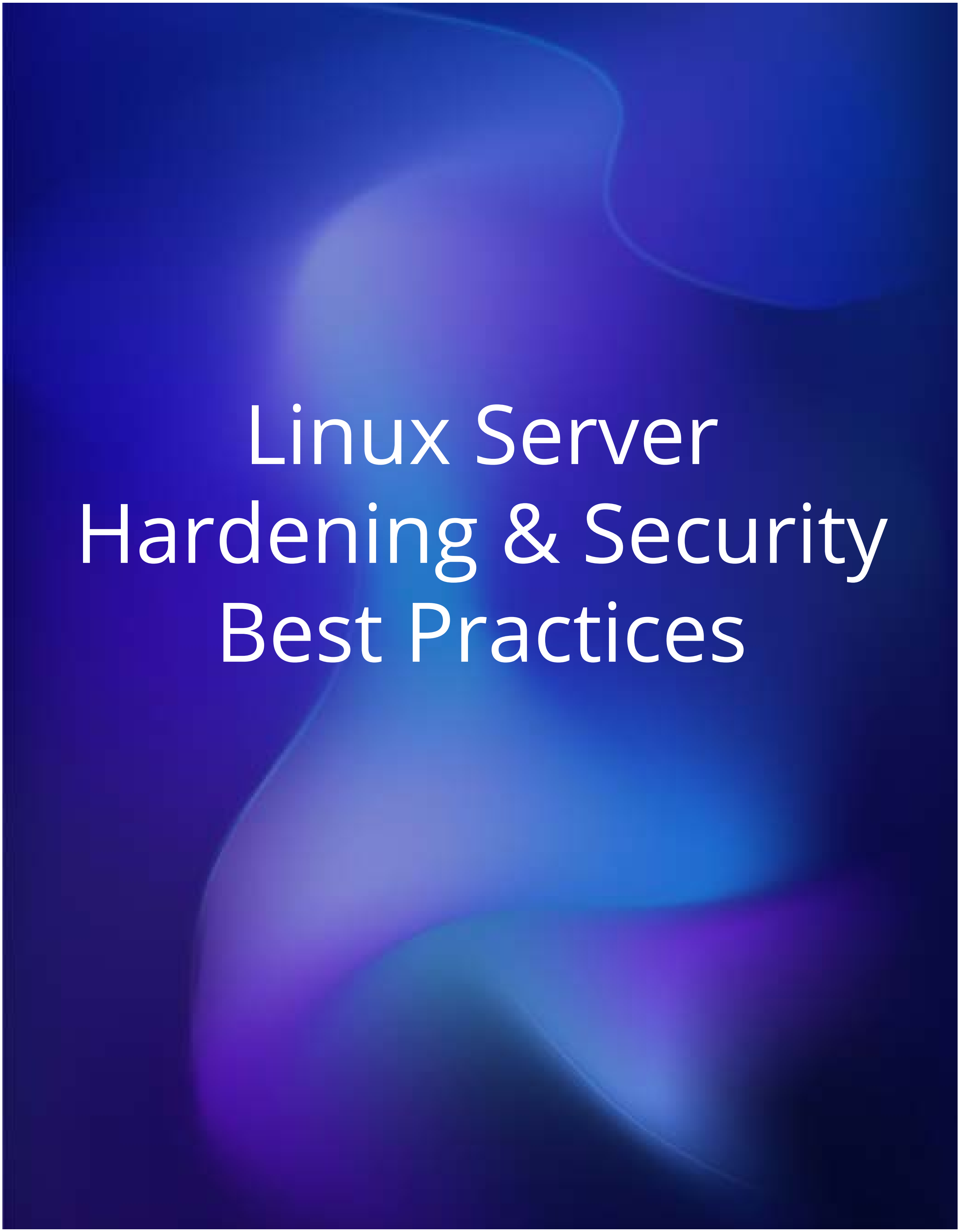
netwrix



Linux Server Hardening & Security Best Practices

Contents

Introduction

Implementing secure configurations across your computing environment, including your Unix and Linux systems, is a key security best practice because it reduces your attack surface area and limits the damage that cyberattacks can do. Indeed, system hardening is a core control in many compliance directives.

This guide explains how what configuration hardening is and how to establish hardened build standards for your Linux and Unix systems.

The Basics of System Hardening

The most secure Linux server or other computer is the one that is powered off and disconnected from the network. But if we want to actually use the machine to provide IT services, we need to maximize its security defenses when it is booted up and attached to the network or even the internet.

This is where hardening comes into play. Hardening a system means optimizing its configuration for secure operations and data protection. The range of exploitable vulnerabilities is vast, so there are three main rules to follow:

- **Remove functionality that isn't needed for the role of the computer.** For example, operating systems and other applications are typically delivered in a "quick start" state, helpfully pre-loaded with utilities and features that users might want. However, the more functions that are provided, the greater the attack surface presented by the host. Therefore, be sure to disable functions you don't need and avoid adding unnecessary functions wherever possible.

- **Patch and update promptly.** Flaws in software design create vulnerabilities that attackers can exploit. Software vendors provide patches or updated versions of the software to remediate these issues, so make sure all software, including operating systems, is fully updated at all times.

- **Maintain secure configuration settings.** It's crucial that secure settings be enabled and maintained.

Linux System Hardening

Unsurprisingly, Linux security hardening is a specialized procedure in its own right, given the wide-range of subtly different Linux distributions. While there is no shortage of guides, checklists and tips for best practices in Linux server hardening — including the Center for Internet Security (CIS) benchmarks and the DISA STIGs — this guide will provide the key config options to adopt.

Important: Make sure you back up your systems before applying hardened settings and test after a restart. It’s easy to get locked out!

Remove Unneeded Functionality

Start by stripping out any features, utilities and services that are not required for running the server. Uninstall unnecessary software and disable system services that are not needed. Since the hardware in your infrastructure also presents exploitable vulnerabilities, make sure any accessible interfaces, such as USB ports, are disabled or even physically removed from the machine.

The workflow here is to log in and get a report of installed packages and services, and review it to identify any that can be removed or disabled. For the essential items that remain, check for available patches to remediate known vulnerabilities against exploits.

How to List Software on Linux/Unix Systems

System	Commands to Use
Linux (e.g., RedHat or CentOS)	`rpm -qa --qf ‘%{NAME} %{VERSION} %{VENDOR}\n’`
Debian Linux (e.g., Ubuntu)	`dpkg -l`
Fedora Linux	To list software from a terminal or putty session:

System	Commands to Use

	`service --status-all`
	`chkconfig --list`
	`systemctl -a`
	
	To stop a service:
	`Service <Service-Name> stop`
	`Chkconfig <Service-Name>`
	`Systemctl stop <Service-Name>`
	
	To disable a service:
	`Systemctl disable <Service-Name>`
	`Chkconfig <Service-Name> off`
	
	Also inspect the `/etc/init.d/` path for any service control scripts and run ls `/etc/init.d/` to expose all startup scripts; then rename or remove any that are to be disabled.

Minimize Open Ports and Other Network Vulnerabilities

Network-based attacks are among the most common threats. To reduce your risk, identify open network-accessible ports and remove any corresponding processes that are not needed. To list open ports on Linux, use the following command:

```
ss -tulpn | egrep “LISTEN” | awk ‘{print “IP-Port “ $4 “ PID/Name “ $7}’
```

In addition:

- Ensure that the TCP Wrappers service is active.
- Define both an etc/hosts.allow whitelist and an etc/hosts.deny blacklist.
- Use the iptables or Firewalld services with a Deny All policy for both IP V4 and V6 traffic, even if you are using on-network third-party WAFs and firewall protection.
- Block ICMP traffic to thwart ping scans used by hacker tools for device discovery.

Review User Accounts and Authentication

Review local user accounts and remove any that are not needed. For local user accounts that remain, a strong password policy should be configured that covers password complexity, length, expiration, re-use and change frequency. Also be sure to use strong hashing algorithms for stored passwords.

User accounts and authentication should be governed using a centralized control system such as Active Directory or, even better, a modern privileged access management (PAM) solution that allows a Zero Standing Privilege strategy, which negates many of the problems with traditional user accounts and permanently assigned privilege. Of course, never use *root* and always make sure that sudo elevation is used only on an as-needed basis.

How to Configure a Password Policy for Local Accounts

Locate and edit the following configuration file: `/etc/security/pwquality.conf`

To enforce a 14-character password:

```
minlen = 14
```

To enforce complexity for all passwords:

```
minclass = 4
```

OR

```
dcredit = -1
ucredit = -1
ocredit = -1
lcredit = -1
```

To enforce a strong password lockout policy:

Update the `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` files to set the following options:

```
auth required pam_faillock.so preauth silent deny=5 unlock_time=900
auth required pam_faillock.so authfail deny=5 unlock_time=900
```

To enforce a password history policy of 5 or more passwords:

Add the `remember=5` option to the `pam_pwhistory.so` and `pam_unix.so lines in /etc/pam.d/password-auth:`

```
password requisite pam_pwhistory.so try_first_pass local_users_only
enforce-for-root retry=3 remember=5

password sufficient pam_unix.so sha512 shadow try_first_pass use_authtok
remember=5
```

To enforce strong password hashing:

Update the `/etc/pam.d/system-auth and /etc/pam.d/password-auth` files as follows

```
/etc/pam.d/password-auth:
password sufficient pam_unix.so sha512 shadow try_first_pass use_authtok

/etc/pam.d/system-auth:
password sufficient pam_unix.so sha512 shadow try_first_pass use_authtok
remember=5
```

Review Service Accounts

System and service accounts should be reviewed and any that are no longer required should be removed.

Service accounts are “hardened” in that they only support the operation of a locally run process and do not provide a user shell, so these accounts cannot ever be used to access the server by a user logon. However, a core hardening mantra is to minimize functionality, so unused service accounts should still be removed.

SSH Hardening for Linux and Unix

As the primary route for remotely administering your Linux systems, SSH requires particular attention. There are a number of default settings in the /etc/ssh/sshd_config file that need to be enabled in order to harden the SSH server operation.

SSH is used for all Linux and Unix access, so the following guidelines apply for Unix hardening, too.

Enable v2 of the SSH protocol

For example, the default configuration enables the outdated and less secure version 1 of the SSH protocol. CIS hardening guidance recommends enabling version 2 to enhance security. To do so, simply uncomment the `Protocol 2` setting in the configuration file `(/etc/ssh/sshd_config)` by removing the #, as follows:

Default entry in `/etc/ssh/sshd_config`:

```
# The default requires explicit activation of protocol 1
#Protocol 2
```

A more secure configuration:

```
# The default requires explicit activation of protocol 1
Protocol 2
```

Additional hardening

In addition, apply the following recommended CIS hardened settings to the config file:

```
LogLevel INFO
IgnoreRhosts yes
…
```

```
PermitEmptyPasswords no
…
LoginGraceTime 60
PermitRootLogin no
MaxAuthTries 4
HostbasedAuthentication no
…
X11Forwarding no
…
PermitUserEnvironment no
…
ClientAliveInterval 300
ClientAliveCountMax 0
…
Banner /etc/issue.net
…
AllowUsers <Specify user names, separated by spaces, e.g. user1 user2>
```

Additional Notes

`/etc/issue.net` will need to be created/edited to include an appropriate Banner, e.g.

```
Warning! Unauthorized access to this system is forbidden and will
be prosecuted by law. By accessing this system, you agree that your
actions may be monitored if unauthorized usage is suspected
```

Together with `AllowUsers`, `AllowGroups` is also supported as a config switch/keyword.

Apache Web Server Hardening

Linux is the platform of choice for hosting internet-based web applications, and Apache Tomcat and Apache HTTP Server (often referred to as Apache HTTPD) are two of the most popular options for delivering web content.

The approach to Apache configuration hardening is the same: minimize functionality and implement secure configuration settings where available.

Verify that only essential Apache modules are enabled:

```
httpd -M
```

For example, `mod_dav` and `mod_dav_fs` should always be disabled, while the `log_config` module should always be loaded and enabled.

Confirm the `package.access` definition only includes the following allowed packages:

```
package.access = sun.,org.apache.catalina.,org.apache.coyote.,
org.apache.tomcat.,org.apache.jasper.
```

Configure the server.xml file:

The `server.xml` file on the `CATALINA_HOME/conf` path is the core configuration store for the web server. The recommended settings to include in `server.xml` are as follows:

- Disable the shutdown port:

```
<Server port=”-1” shutdown=”SHUTDOWN”>
```

- Remove unused connectors, including the default `HTTPConnector` and the `AJPConnector. To` remove the `HTTPConnector`, delete or comment out this tag:

```
<Connector className=”org.apache.catalina.connector.http.HttpConnector”
...
connectionTimeout=”60000”/>
```

Remove the default presence-advertising settings:

For Tomcat:

Set the xpoweredBy attributes to false:

```
xpoweredBy=”false” />
```

For HTTP Server:

- Remove the default `index.html` and comment out the below from the `/etc/httpd/conf.d/welcome.conf` using a # or ## for each line:

```
##<LocationMatch “^/+$”>
## Options -Indexes
## ErrorDocument 403 /error/noindex.html
##</LocationMatch>
```

- Comment out the Server-Status section:

```
##<Location /server-status>
## SetHandler server-status
## Order deny,allow
## Deny from all
## Allow from .example.com
##</Location>
```

- Comment out the `server-info` section:

```
##<Location /server-info>
## SetHandler server-info
## Order deny,allow
## Deny from all
## Allow from .example.com
##</Location>
```

Disable diagnostic trace facilities:

Set the `allowTrace` attribute for each `Connector` to `false`

Disable auto deployment of applications, and disable deployment on startup:

```
autoDeploy=”false”
deployOnStartup=”false”
```

Additional hardening:

- Ensure that file and folder permissions are restrictive and that Apache services run using non-shell service accounts (i.e., accounts that cannot be used to log on to the system). Disable Directory Listing and browsing.

```
$CATALINA_HOME
$CATALINA_BASE
```

Kernel Hardening for Linux

SELinux modifies the Linux kernel to enforce mandatory access controls, restricting how Linux processes can access files and programs. This additional layer of restriction provides a fundamental protection mechanism against root kit malware. AppArmor provides an equivalent level of MAC for Debian distributions.

SELinux Settings

Install SELinux on CentOS/RHEL:

```
dnf install libselinux
```

Enable SELinux:

By default, SELinux will be disabled. To enable SELinux at boot, edit /etc/default/grub and remove these default settings:

```
selinux=0
enforcing=0
```

Set SELinux config to enforcing

Edit the `/etc/selinux/config` file to include the following:

```
SELINUX=enforcing
SELINUXTYPE=default
```

Check that all services are being run with a security context controlled by SELinux:

The following command will list any services being launched from the init process, which may require a non-default security context assigned to them:

```
ps -eZ | egrep “initrc” | egrep -vw “tr|ps|egrep|bash|awk” | tr ‘:’ ‘ ‘ |
awk ‘{ print $NF }’
```

Restart the computer to make sure all settings have been loaded.

AppArmor Settings

Install AppArmor on Ubuntu/Debian:

```
apt-get install apparmor apparmor-profiles apparmor-utils
```

Enable AppArmor at boot:

Edit `/etc/default/grub` and add the following settings:

```
apt-get install apparmor apparmor-profiles apparmor-utils
```

Ensure all AppArmor profiles are in enforce mode:

```
aa-enforce /etc/apparmor.d/*
```

Embedded Linux Hardening

Embedded Linux provides a stripped-down operating system for embedded devices or embedded systems, such as smart phones, smart TVs, set-top boxes and broadband internet routers.

A simplified operating system with a reduced footprint is attractive when the full range of function usually provided by server-grade Linux devices is not required and hardware resources such as storage, memory and CPU have been minimized to save manufacturing costs. For example, the open-source Android platform developed by Google is optimized for smart phones and TVs, and the OpenWrt router firmware is used for a wide range of broadband routers.

The functionality included in the operating system varies widely depending upon the intended application. For example, Android includes screen lock, face recognition, PIN entry support and location services; these are not needed in operating systems for home routers, but VPN, guest Wi-Fi and bandwidth throttling features are included. As a result, hardening guidance is specifically linked to the particular device and platform in use.

Android Hardening Highlights

Ensure device firmware is up to date:

Go to Settings | `System | System Updates`; select Check for Update and install any updates that are available.

Install Google Play and security updates:

Go to `Settings | Security`:

- Select `Google Security Check-Up` and apply any updates available.
- Select `Google Play system update` and apply any updates available.

Enable general device security:

1. Go to `Settings | Security | Device Security` and do the following:

- Set `'Screen Lock' to 'Enabled'.`
- Set `'Pattern Visible' to 'Disabled'.`
- Set `'Automatically Lock' to 'Immediately'.`
- Set `'Power Button Instantly Locks' to 'On'.`
- Set Device Admin Apps `'Find My Device' to 'Enabled'.`
- Set `'Allow remote lock and erase' to 'Enabled'.`

2. Go to `Settings | System | Advanced` and do the following:

- Set `'Developer Options'` to `'Off'.`

3. Go to `Settings | Apps & Notifications | Advanced | Special App Access` and do the following:

- Set `'Install Unknown Apps' to 'Not Allowed'.`

4. Launch `Play Store |` Menu and do the following:

- Select `'My Apps & Games'` and click `'Update All'.`

OpenWrt Hardening Highlights

- Change the admin password from the default.
- Ensure the firmware is always updated to the latest version.

Kali Linux Hardening

Kali Linux has been optimized to be the pen testers' platform of choice, so it has a wide range of security auditing and pen testing utilities.

Since Kali Linux is a Debian-based Linux distribution, you can use the Linux hardening tips above to address the security weaknesses in Kali Linux systems.

Linux Mint Security Hardening

Similarly for Linux Mint, as an Ubuntu-derived Desktop Linux platform, the same hardening procedures used for Debian-Linux should be adopted. Work through the earlier Linux Hardening Checklist steps and apply these to your Linux Mint systems.

Hardening Arch Linux

As an independent Linux distribution not directly derived from Debian or Fedora, Arch Linux is an individual case when it comes to hardening measures, although the concepts of how and what to harden are similar. Arch code is intentionally built as a streamlined 'no filler' Linux distribution and as such will require less hardening work when it comes to removing/disabling unnecessary services

Final Word

Even a hardened system can still be compromised, especially by the following:

- Zero-day threats — exploits we had no knowledge of and therefore did not protect against
- Ransomware and other malware
- Insider threats, including both hackers using hijacked credentials and users abusing their access

Therefore, in addition to hardening your systems, you need additional defenses. In particular, it's vital to monitor system and file integrity. Any unexpected change could lead to a breach or other security incident. Netwrix® Change Tracker provides real-time alerts on all unplanned changes, and it supports all Linux and Unix platforms.

Harden Linux/Unix Server configurations with Netwrix® Change Tracker

- Establish secure Linux/Unix server configurations faster and stop configuration drift.
- Enhance breach forensics with constant monitoring and verification of all changes to Linux/Unix server configurations.
- Pass compliance audits with ease using 250+ reports covering CIS, NIST, PCI DSS, CMMC, STIG and NERC CIP.

Request Free Trial

About Netwrix

Netwrix makes data security easy thereby simplifying how professionals can control sensitive, regulated and business-critical data, regardless of where it resides. More than 11,500 organizations worldwide rely on Netwrix solutions to secure sensitive data, realize the full business value of enterprise content, pass compliance audits with less effort and expense, and increase the productivity of IT teams and knowledge workers.

Founded in 2006, Netwrix has earned more than 150 industry awards and been named to both the Inc. 5000 and Deloitte Technology Fast 500 lists of the fastest growing companies in the U.S.

For more information, visit www.netwrix.com

Next Steps

Netwrix products — Check out the full portfolio of Netwrix products: netwrix.com/products

Live demo — Take a product tour with a Netwrix expert: netwrix.com/livedemo

Request quote — Receive pricing information: netwrix.com/buy

CORPORATE HEADQUARTER:	PHONES:
300 Spectrum Center Drive Suite 200 Irvine, CA 92618	1-949-407-5125 Toll-free (USA): 888-638-9749
565 Metro Place S, Suite 400 Dublin, OH 43017	1-201-490-8840
5 New Street Square London EC4A 3TW	+44 (0) 203 588 3023

OTHER LOCATIONS:	
Spain:	+34 911 982608
Netherlands:	+31 858 887 804
Sweden:	+46 8 525 03487
Switzerland:	+41 43 508 3472
France:	+33 9 75 18 11 19
Germany:	+49 711 899 89 187
Hong Kong:	+852 5808 1306
Italy:	+39 02 947 53539

SOCIAL:

netwrix.com/social