Linux Hardening Checklist

System Installation & Patching

1 If machine is a new install, protect it from hostile network traffic until the operating system is installed and hardened. ☐

---

2 Use the latest version of the Operating System if possible ☐

Refer to the vendor support documentation to confirm the lifecycle of the version. Consider both the major and minor (or service pack) release where a vendor releases both.

3 Create a separate volume with the nodev, nosuid, and noexec options set for /tmp. ☐

Since /tmp is intended to be world writable, creating a separate partition for it can prevent resource exhaustion. Setting nodev prevents users from creating or using block or special character devices. Setting noexec prevents users from running binary executables from /tmp. Setting nosuid prevents users from creating set userid files in /tmp.

4 Create separate volumes for /var, /var/log, and /home. ☐

Any directories where non-admin users have write access should be separate from the root volume to limit the impact of those volumes being filled.

5 Set sticky bit on all world-writable directories. ☐

The sticky bit stops users with write access to the directory deleting files owned by other users.

6 Ensure the system is configured to be able to receive software updates ☐

For Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES) this requires a subscription to be allocated to the system. For most other major distributions this is a simple configuration change.

OS Hardening

1 Restrict core dumps. ☐

Core dumps are intended to help determine why a program aborted. They may contain sensitive or confidential data from memory. It is recommended that core dumps be disabled or restricted.

2 Remove legacy services ☐

Services that provide/rely on unencrypted authentication should be disabled unless there are grounds for an exception. These include telnet-server; rsh, rlogin, rcp; ypserv, ypbind; tftp, tftp-server; talk and talk-server.

3 Disable any services and applications started by xinetd or inetd that are not being utilized. Remove xinetd, if possible ☐

The inetd or xinetd service allows for programs to be ran when a connection is made to a designated network port. All unneeded inetd applications should be disabled - if there are no applications required then disable (x)inetd.

4 Disable or remove server services that are not going to be utilized ☐

(e.g., FTP, DNS, LDAP, SMB, DHCP, NFS, SNMP, etc.)

5 Ensure syslog (rsyslog, syslog, syslog-ng) service is running. ☐

The syslog service manages the logs in /var/log/. Most modern syslog implementations also support remote log forwarding.

6 Enable an Network Time Protocol (NTP) service to ensure clock accuracy ☐

Accurate time keeping facilitates analysis of system logs when needed

7 Restrict the use of the `cron` and `at` services. ☐

These can be used to run commands on the system and should only be allowed to accounts which need this access

User Access & Passwords

1 Create an account for each user who should access the system ☐

Avoiding shared accounts/passwords makes it easier to keep an audit trail and remove access when no longer needed.

2 Enforce the use of strong passwords ☐

Password security rules can be set in /etc/pam.d/password-auth

3 Use sudo to delegate admin access ☐

The sudo command allows for fine-grained control of rights to run commands as root (or other user ids). The configuration file `/etc/sudoers` should be edited with the `visudo` command.

Network Security & Remote Access

1 Limit connections to services running on the host to authorized users of the service via firewalls and other access control technologies. ☐

The iptables firewall is a kernel component common to all linux systems, but the tools use to manage firewall rules differ significantly between vendors so check with the version specific configuration guide.

2 Disable: ☐

- IP forwarding.
- send packet redirects.
- source routed packet acceptance.
- ICMP redirect acceptance.

Enable:

- Ignore Broadcast Requests.
- Bad Error Message Protection.
- TCP/SYN cookies.

These kernel tuning parameters should be set in /etc/sysctl.conf

3 In the SSH server configuration ensure that: ☐

- Protocol version is set to 2
- LogLevel is set to INFO
- PermitEmptyPasswords is set to No

These settings are the default on most platforms, setting them to other values impacts the security of the SSH server.

4 Disable root login over SSH. ☐

Root SSH with password should never be allowed - users should authenticate with their own account and use su or sudo if needed. Valid values for `PermitRootSSH` are *no*, *without-password* and *forced-commands-only* depending on whether key based access is required.

5 Deploy an Intrusion Prevention System (IPS) such as fail2ban ☐

---

fail2ban uses the iptables firewall to block remote systems generating many authentication failures as a way to combat brute force password attempts.

Apache Webserver (HTTPD)

1 Always run apache with a dedicated non-admin account ☐

The system user account the apache server runs in should have minimal permission on the system to limit the potential for this to be exploited. This is the default in all major Linux distributions.

2 Disable any modules not required ☐

Apache is modular in design - each module provides different functionality and almost all are optional for basic use cases. In particular look to disable webdav, status, info, userdir and autoindex unless these are known to be required.

3 Disable HTTP Trace: `TraceEnable Off` ☐

The inetd or xinetd service allows for programs to be ran when a connection is made to a designated network port. All unneeded inetd applications should be disabled - if there are no applications required then disable (x)inetd.

4 Configure SSL in line with best practice ☐

Mozilla provide resources for this https://wiki.mozilla.org/Security/Server_Side_TLS

5 Configure Apache not to advertise the software/OS versions ☐

Set “ServerTokens Prod” and “ServerSignature Off” to limit the system configuration information easily available.

6 Deny access to files by default - only allow access to designated directories. ☐

Only directories containing apache content should be readable by remote clients.