1. Hide the Apache version
    a. Open Apache's main configuration file

```
sudo nano /etc/httpd/conf/httpd.conf
```

    b. Add the following lines at the end of file:

```
ServerSignature Off
ServerTokens Prod
```

    c. Save the file and restart the Apache service to reflect these changes:

```
sudo service httpd restart
```

2. Turn Off Directory Listing
    a. Turn off the directive listing setting by using the Options directive in the Apache configuration file for a specific web directory

```
sudo nano /etc/httpd/conf/httpd.conf
```

    b. Find the section that begins with Directory /var/www/html and add -Indexes in the Options directive:

```
<Directory /var/www/html/>
    Options -Indexes
    AllowOverride None
    Require all granted
</Directory>
```

    c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

3. Disable Unnecessary Modules
    a. List all enabled modules on the server using the following command:

```
sudo grep LoadModule /etc/httpd/conf.modules.d/00-base.conf
```

From the enabled modules in 00-base.conf file, some modules like mod_info, mod_userdir, mod_autoindex are enabled but not needed.

b. Disable this modules by editing the 00-base.conf file:

```
sudo nano /etc/httpd/conf.modules.d/00-base.conf
```

c. Insert a # at the beginning of the following lines to disable the modules:

```
#LoadModule info_module modules/mod_info.so
#LoadModule info_module modules/mod_info.so
#LoadModule userdir_module modules/mod_userdir.so
```

d. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

4. Disable Apache's FollowSymLinks
   a. To turn off Apache's FollowSymLinks, make the changes in httpd.conf file:

```
sudo nano /etc/httpd/conf/httpd.conf
```

   b. Find the section that begins with Directory /var/www/html. Add -FollowSymLinks in option directive:

```
<Directory /var/www/html/>
    Options -Indexes -FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

   c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

5. Turn Off Server-Side Includes (SSI) And CGI Execution

Server-side includes (SSI) are directives present on Web applications that are placed in HTML pages. An SSI attack allows a web application to be exploited by remotely executing arbitrary codes. The attacker can access sensitive information like password files, and execute shell commands. It is recommended that you disable server side includes and CGI execution if they are not needed

a. Edit the main Apache config file:

```
sudo nano /etc/httpd/conf/httpd.conf
```

b. Find the section that begins with Directory /var/www/html, Add -ExecCGI and -Includes in option directive:

```
<Directory /var/www/html/>
    Options -Indexes -FollowSymLinks -ExecCGI -Includes
    AllowOverride None
    Require all granted
</Directory>
```

c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

6. Limit Request Size

By default Apache has no limit on the size of the HTTP request. This can allow hackers to send a large number of data. You can set value from 0 (unlimited) to 2147483647 (2GB) in the main Apache config file. To limit the request size in Apache, please follow below mentioned steps:

a. To limit the request size for the /var/www/html/www.example.com directory to 200K:

```
sudo nano /etc/httpd/conf/httpd.conf
```

b. Add the following line:

```
<Directory /var/www/html/www.example.com>
    LimitRequestBody 204800
</Directory>
```

c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

7. Disallow Browsing Outside the Document Root
    a. Open httpd.conf file

```
sudo nano /etc/httpd/conf/httpd.conf
```

    b. Add/edit the following line:

```
<Directory />
    Options None
    Order deny,allow
    Deny from all
</Directory>
```

    c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

8. Keep Apache up to date

```
sudo yum update httpd
```

9. Secure Apache From Clickjacking Attacks

Clickjacking, also known as "User Interface redress attack," is a malicious technique to collect an infected user's clicks. Clickjacking tricks the victim (visitor) into clicking on an infected site. To avoid this, you need to use X-FRAME-OPTIONS to prevent your website from being used by clickjackers.

    a. Open httpd.conf file

```
sudo nano /etc/httpd/conf/httpd.conf
```

    b. Add the following line:

```
Header append X-FRAME-OPTIONS "SAMEORIGIN"
```

    c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

10. Disable ETag

ETags (entity tags) are a well-known point of vulnerability in Apache web server. ETag is an HTTP response header that allows remote users to obtain sensitive information like inode number, child process ids, and multipart MIME boundary. ETag is enabled in Apache by default.

a. Open httpd.conf file

```
sudo nano /etc/httpd/conf/httpd.conf
```

b. Add the following line:

```
FileETag None
```

c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

11. HTTP Request Methods

Apache supports the OPTIONS, GET, HEAD, POST, CONNECT, PUT, DELETE, and TRACE method in HTTP 1.1 protocol. Some of these may not be required, and may pose a potential security risk. It is a good idea to only enable HEAD, POST, and GET for web applications.

a. Open httpd.conf file

```
sudo nano /etc/httpd/conf/httpd.conf
```

b. Find the section that begins with Directory /var/www/html. Add the following lines under this section:

```
<LimitExcept GET POST HEAD>
    deny from all
</LimitExcept>
```

c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

12. Secure Apache from XSS Attacks

Cross-site scripting (XSS) is one of the most common application-layer vulnerabilities in Apache server. XSS enables attackers to inject client-side script into web pages viewed by other users. Enabling XSS protection is recommended.

a. Open httpd.conf file

```
sudo nano /etc/httpd/conf/httpd.conf
```

b. Add the following line:

```
<IfModule mod_headers.c>
    Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

13. Protect Cookies with HTTPOnly Flag

You can protect your Apache server from most of the common Cross Site Scripting attacks using the HttpOnly and Secure flags for cookies.

a. Open httpd.conf file

```
sudo nano /etc/httpd/conf/httpd.conf
```

b. Add the following line:

```
<IfModule mod_headers.c>
    Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure
</IfModule>
```

c. Save the file and restart Apache service to reflect these changes:

```
sudo service httpd restart
```

14. Ensure XDCMP is not enabled

Command

grep -Eis

Remediation

Refrance Image

Audit:

Run the following command and verify the output:

```
# grep -Eis '^\s*Enable\s*=\s*true' /etc/gdm/custom.conf

Nothing should be returned
```

Remediation:

Edit the file `/etc/gdm/custom.conf` and remove the line

```
Enable=true
```

15. *Ensure updates, patches, and additional security software are installed (Manual)*

Audit:

Run the following command to verify there are no updates or patches to install.

```
# yum check-update
```

Remediation:

Use your package manager to update all packages on the system according to site policy.
The following command will install all available packages

```
# yum update
```

16. Ensure xinetd is not installed.

Audit:

Run the following command to verify `xinetd` is not installed:

```
# rpm -q xinetd

package xinetd is not installed
```

Remediation:

Run the following command to remove `xinetd`:

```
# yum remove xinetd
```

17. Ensure time synchronization in use.

Audit:

Run the following commands to verify that a time synchronization packages is installed:

```
# rpm -q chrony ntp

chrony-<version>

# rpm -q ntp

ntp-<version>
```

Remediation:

Run **One** of the following commands to install chrony **or** NTP:
To install chrony, run the following command:

```
# yum install chrony
```

18. Chrony is configured.

Audit:

IF chrony is installed on the system:
Run the following command and verify remote server is configured properly:

```
# grep -E "^(server|pool)" /etc/chrony.conf

server <remote-server>
```

Multiple servers may be configured.
Run the following command and verify `OPTIONS` includes '`-u chrony`':

```
# grep ^OPTIONS /etc/sysconfig/chronyd

OPTIONS="-u chrony"
```

Additional options may be present.

Remediation:

Add or edit server or pool lines to `/etc/chrony.conf` as appropriate:

```
server <remote-server>
```

Add or edit the `OPTIONS` in `/etc/sysconfig/chronyd` to include '`-u chrony`':

```
OPTIONS="-u chrony"
```

19. Cpus is not installed.

Audit:

Run the following command to verify `cups` is not installed:

```
# rpm -q cups

package cups is not installed
```

Remediation:

Run the following command to remove `cups`:

```
# yum remove cups
```

20. DHCP server is not installed.

Audit:

Run the following command to verify `dhcp` is not installed:

```
# rpm -q dhcp

package dhcp is not installed
```

Remediation:

Run the following command to remove `dhcp`:

```
# yum remove dhcp
```

21. DNS server is not installed.

Audit:

Run one of the following commands to verify `bind` is not installed:

```
# rpm -q bind

package bind is not installed
```

Remediation:

Run the following command to remove `bind`:

```
# yum remove bind
```

22. FTP server is not installed.

Audit:

Run the following command to verify `vsftpd` is not installed:

```
# rpm -q vsftpd

package vsftpd is not installed
```

Remediation:

Run the following command to remove `vsftpd`:

```
# yum remove vsftpd
```

23. IMAP and pop3 server are not installed.

Audit:

Run the following command to verify `dovecot` is not installed:

```
# rpm -q dovecot

package dovecot is not installed
```

Remediation:

Run the following command to remove `dovecot`:

```
# yum remove dovecot
```

24. HTTP proxy server is not installed.

Audit:

Run the following command to verify `squid` is not installed:

```
# rpm -q squid

package squid is not installed
```

Remediation:

Run the following command to remove the `squid` package:

```
# yum remove squid
```

25. Telnet server is not installed

Audit:

Run the following command to verify the `telnet-server` package is not installed:

```
rpm -q telnet-server

package telnet-server is not installed
```

Remediation:

Run the following command to remove the telnet-server package:

```
# yum remove telnet-server
```

26. rpcbind server is not installed

Audit:

Run the following command to verify `rpcbind` is not installed:

```
# rpm -q rpcbind

package rpcbind is not installed
```

OR

If the `rpcbind` package is required as a dependency, run the following commands to verify that the `rpcbind` and `rpcbind.socket` services are masked:

```
# systemctl is-enabled rpcbind

masked

# systemctl is-enabled rpcbind.socket

masked
```

Remediation:

Run the following command to remove `nfs-utils`:

```
# yum remove rpcbind
```

27. rsh client is not installed

Audit:

Run the following command to verify that the `rsh` package is not installed:

```
# rpm -q rsh

package rsh is not installed
```

Remediation:

Run the following command to remove the `rsh` package:

```
# yum remove rsh
```

28. LDAP is not installed.

Audit:

Run the following command to verify that the `openldap-clients` package is not installed:

```
# rpm -q openldap-clients

package openldap-clients is not installed
```

Remediation:

Run the following command to remove the `openldap-clients` package:

```
# yum remove openldap-clients
```

29. IP forwarding is disabled

Audit:

Run the following commands and verify output matches:

```
# sysctl net.ipv4.ip forward

net.ipv4.ip_forward = 0

# grep -E -s "^\s*net\.ipv4\.ip forward\s*=\s*1" /etc/sysctl.conf
/etc/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf

No value should be returned
```

IFIPv6 is enabled:

Run the following commands and verify output matches:

```
# sysctl net.ipv6.conf.all.forwarding

net.ipv6.conf.all.forwarding = 0

# grep -E -s "^\s*net\.ipv6\.conf\.all\.forwarding\s*=\s*1" /etc/sysctl.conf
/etc/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf

No value should be returned
```

30. Audit is installed.

Audit:

Run the following command and verify auditd is installed:

```
# rpm -q audit audit-libs

audit-<version>
audit-libs-<version>
```

Remediation:

Run the following command to Install auditd

```
# yum install audit audit-libs
```

31. Audit service is enabled.

Audit:

Run the following command to verify `auditd` is enabled:

```
# systemctl is-enabled auditd

enabled
```

Run the following command to verify that `auditd` is running:

```
# systemctl status auditd | grep 'Active: active (running) '

   Active: active (running) since <time and date>
```

Remediation:

Run the following command to enable and start `auditd`:

```
# systemctl --now enable auditd
```

32. Audit log storage size is configured

Audit:

Run the following command and ensure output is in compliance with site policy:

```
# grep max_log_file /etc/audit/auditd.conf

max_log_file = <MB>
```

Remediation:

Set the following parameter in `/etc/audit/auditd.conf` in accordance with site policy:

```
max_log_file = <MB>
```

33. Login and logout events are collected.

Audit:

Run the following commands:

```
# grep logins /etc/audit/rules.d/*.rules
# auditctl -l | grep logins
```

Verify output of both includes:

```
-w /var/log/lastlog -p wa -k logins
-w /var/run/faillock/ -p wa -k logins
```

Remediation:

Edit or create a file in the `/etc/audit/rules.d/` directory ending in `.rules`
Example: `vi /etc/audit/rules.d/50-logins.rules`
Add the following lines:

```
-w /var/log/lastlog -p wa -k logins
-w /var/run/faillock/ -p wa -k logins
```

34. Session information is collected.

```
# auditctl -l | grep -E '(session|logins)'
```

Verify output includes:

```
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins
```

Remediation:

Edit or create a file in the `/etc/audit/rules.d/` directory ending in `.rules`
Example: `vi /etc/audit/rules.d/50-session.rules`
Add the following lines:

```
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins
```

35. rsyslog is installed.

Audit:

Run the following command to Verify `rsyslog` is installed:

```
# rpm -q rsyslog

rsyslog-<version>
```

Remediation:

Run the following command to install rsyslog:

```
# yum install rsyslog
```

36. Check rsyslog is enabled.

Audit:

Run one of the following commands to verify `rsyslog` is enabled:

```
# systemctl is-enabled rsyslog

enabled
```

Run the following command to verify that `rsyslog` is running:

```
# systemctl status rsyslog | grep 'active (running) '

 Active: active (running) since <Day date time>
```

Remediation:

Run the following command to enable and start `rsyslog`:

```
# systemctl --now enable rsyslog
```

37. crond daemon is enabled.

Audit:

If `cron` is installed:
Run the following commands to verify `cron` is enabled and running:

```
# systemctl is-enabled crond

enabled

# systemctl status crond | grep 'Active: active (running) '

Active: active (running) since <Day Date Time>
```

Remediation:

Run the following command to enable and start `cron`:

```
# systemctl --now enable crond
```

OR
Run the following command to remove `cron`:

```
# yum remove cronie
```

38. Permission on /etc/ssh/sshd_config are configured.

Audit:

Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other`:

```
# stat /etc/ssh/sshd config

Access: (0600/-rw-------)  Uid: (    0/    root)   Gid: (    0/    root)
```

Remediation:

Run the following commands to set ownership and permissions on `/etc/ssh/sshd_config`:

```
# chown root:root /etc/ssh/sshd_config

# chmod og-rwx /etc/ssh/sshd_config
```

39. SSH log level is appropriate.

Audit:

Run the following command and verify that output matches `loglevel VERBOSE` or `loglevel INFO`:

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname)
/etc/hosts | awk '{print $1}')" | grep loglevel

loglevel VERBOSE or loglevel INFO
```

Run the following command and verify the output matches:

```
# grep -i 'loglevel' /etc/ssh/sshd_config | grep -Evi '(VERBOSE|INFO)'

Nothing should be returned
```

40. SSH root login is disabled

Audit:

Run the following command and verify that output matches:

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname)
/etc/hosts | awk '{print $1}')" | grep permitrootlogin

permitrootlogin no
```

Run the following command and verify the output:

```
# grep -Ei '^\s*PermitRootLogin\s+yes' /etc/ssh/sshd_config

Nothing should be returned
```

Remediation:

Edit the `/etc/ssh/sshd_config` file to set the parameter as follows:

```
PermitRootLogin no
```

41. SSH PAM is enabled.

Audit:

Run the following command and verify that output matches:

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname)
/etc/hosts | awk '{print $1}')" | grep -i usepam

usepam yes
```

Run the following command and verify the output:

```
# grep -Ei '^\s*UsePAM\s+no' /etc/ssh/sshd config

Nothing should be returned
```

Remediation:

Edit the `/etc/ssh/sshd_config` file to set the parameter as follows:

```
UsePAM yes
```

42. Permission on /etc/passwd

Audit:

Run the following command and verify Uid and Gid are both 0/root and Access is 644 or more restrictive:

```
# stat /etc/passwd-

Access: (0644/-rw-------)  Uid: (    0/    root)   Gid: (    0/    root)
```

Remediation:

Run the following commands to set owner, group, and permissions on `/etc/passwd-`:

```
# chown root:root /etc/passwd-

# chmod u-x,go-wx /etc/passwd-
```

43. Permission on /etc/shadow

Audit:

Run the following command and verify Uid and Gid are 0/root , and Access is 0000 :

```
# stat /etc/shadow

Access: (0000/----------)  Uid: (    0/    root)   Gid: (    0/    root)
```

Remediation:

Run the following commands to set owner, group, and permissions on `/etc/shadow`:

```
# chown root:root /etc/shadow

# chmod 0000 /etc/shadow
```