**CyberPatriot**

**Linux**

Extensive scientific research shows that this is the last image the human brain creates at the point of death. Could this be god himself?

**Table of Contents**

[Readme](#_ycl6jze4d2mw)

[Forensics](#_nbfuh1w7u4x7)

[Updates](#_4auk43tew9k8)

[Users and Groups](#_ff17r0d7cdv8)

[User Policy](#_2x630sgxuc1j)

[Network](#_qvano8tjm4mg)

[Malicious Software and Files](#_8x129t9rwnmj)

[SSH](#_3tyfnkoy5hlz)

[FTP](#_8mbf298xlex7)

[Apache](#_y56h558hnne4)

[Samba](#_cw94kkezr8i5)

[MySQL](#_twid6zvl0vtr)

[PHP](#_yn4gvx2pj6xn)

[Last Ditch Efforts](#_mdbv1dbels8m)

[Notes](#_aixitmpj1lg)

# **Readme**

**Read Readme**

The readme is located on the desktop. It contains vital information about the image and what will be scored.

**Take Notes**

Note all users, tasks, services, etc mentioned in the document.

# **Forensics**

**Complete Forensics Questions**

Forensics questions are located on the desktop. These questions are similar to those found in a CTF.

**Complete Forensics Questions First**

Forensics questions are to be completed before you start anything else. Otherwise, the answer could be tampered with.

# **Updates**

**Check Apt Sources (s)**

Check apt sources to ensure that they only contain the proper links. Otherwise, updates may not work.

**GUI:**

System Settings > Software and Updates > Other Software

**CLI:**

editor /etc/apt/sources.list

**Download Offline Updates**

This will do the same thing as online updates, only faster and will not put pressure on your internet connection.

**CLI:**

apt install apt-offline

apt-offline set /tmp/my.sig

apt-offline get -d /tmp/updates /tmp/my.sig

**Install Offline Updates**

If offline updates are downloaded and ready to be installed, you should use this method to install them.

**CLI:**

apt-offline install /tmp/updates

**Install Updates (s)**

Unlike windows, updates on linux can run in the background and do not require a reboot. However, you should reboot after a full upgrade so that all packages are properly upgraded. Make sure you close all programs while running updates.

**GUI:**

Update Manager

**CLI:**

apt update

apt upgrade

apt dist-upgrade

apt-get autoremove

**Install From Important Security Updates (s)**

Important security updates ensure that the operating software is fully patched and as invulnerable as possible.

**GUI:**

System Settings > Software and Updates > Updates

Check Important Security Updates

**CLI:**

editor /etc/apt/sources.list

deb http://security.ubuntu.com/ubuntu/ xenial-security main universe

deb http://us.archive.ubuntu.com/ubuntu/ xenial-updates main universe

**Unattended Upgrades (s)**

Unattended upgrades run updates automatically.

**GUI:**

System Settings > Software and Updates > Updates

Automatically check for updates daily

When there are security updates download and install automatically

**CLI:**

apt install unattended-upgrades

dpkg-reconfigure -plow unattended-upgrades

editor /etc/apt/apt.conf.d/10periodic

Set all values to 1

# **Users and Groups**

**Look Through Users (s)**

Compare users on the system to users on the readme. Go down the list on the readme and compare to the system and then go down the list on the system and compare to the readme to ensure that you do not miss any users.

**GUI:**

System Settings > User Accounts

**CLI:**

cat /etc/passwd

userdel *user*

useradd *user*

**Check For Fake Root (s)**

If another user other than root has a uid of 0, it is given the same permissions as root.

**CLI:**

cut -d: -f 1,3 /etc/passwd | grep 0

**Look Through Groups**

Compare users and admins on the system to users and admins on the readme. In the CLI, the administrators group is ‘sudo’.

**GUI:**

System Settings > User Accounts

**CLI:**

cat /etc/group

usermod -a -G *group user*

**Change Passwords (s)**

Once all system users are verified, their passwords (INCLUDING ROOT) should be changed to something secure.

**GUI:**

System Settings > User Accounts

**CLI:**

passwd *user*

**Lock Out Root (s)**

Usually, it says in the readme that users should only access root through the sudo command.

**CLI:**

passwd -l root

**Check for Locked Out Users (s)**

While root should be locked, regular users should remain unlocked.

**CLI:**

cat /etc/shadow | grep ‘!’ **OR** passwd -s *user*

passwd -u *user* **OR** usermod -p *password user*

**Disable Guest User (s)**

The guest user allows access to the system without any password.

**CLI:**

editor /etc/lightdm/lightdm.conf

allow-guest=false

**Prevent Display of Usernames at Login Screen**

The usernames on the system being publically displayed leaves the system vulnerable.

**CLI:**

editor /usr/share/lightdm/lightdm.conf/50-ubuntu-conf

greeter-show-manual-login=true

greeter-hide-users=true

**Ensure Only Main Account Has Autologin (s)**

Only the main account - specified in the readme - should be logged in automatically.

**GUI:**

System Settings > User Accounts

**CLI:**

/usr/share.lightdm/lightdm.conf/50-ubuntu-conf

**Check Sudoers (s)**

There are various sudoers files that regulate who has access to sudo and how they access it. The files are /etc/sudoers and /etc/sudoers.d/\*. Check for ‘!authenticate’ or anything else out of the ordinary

**CLI:**

editor /etc/sudoers

editor /etc/sudoers.d

**Check PATH Variable**

The PATH variable determines where the system looks for commands. The PATH variable usually should only contain files that end with bin.

**CLI:**

echo $PATH

**Media Files/Hacking Tools (s)**

Search for any unnecessary files in the /home/ directory.

**GUI:**

Files > /home/

**CLI:**

apt install tree

tree -a /home/

**AND**

find / -iname \*.*ext* | grep -v /usr/

# **User Policy**

Practice extreme caution when tampering with any PAM files. Always keep a root shell open in case anything goes south.

**Enable Password Aging (s)**

The file /etc/login.defs contains various security options. One of them is password aging, which includes maximum password age, minimum password age, and password warn age.

**CLI:**

editor /etc/login.defs

PASS\_MAX\_DAYS 90

PASS\_MIN\_DAYS 10

PASS\_WARN\_DAYS 7

**Enable Encryption (s)**

Another security option in /etc/login.defs is the systems encryption method.

**CLI:**

editor /etc/login.defs

ENCRYPT\_METHOD SHA512

Comment ROUNDS lines

**Install Cracklib (s)**

Cracklib checks to make sure that passwords meet requirements before they are used.

**CLI:**

apt install libpam-cracklib

**Configure common-password (s)**

/etc/pam.d/common-password configures password complexity and password history. Add the lines below to the end of the file.

**CLI:**

editor /etc/pam.d/common-password

password requisite pam\_cracklib.so retry=3 minlen=8 difok=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1

password requisite pam\_pwhistory.so use\_authtok remember=24 enforce\_for\_root

**Configure common-auth (s)**

/etc/pam.d/common-auth enforces the lockout policy. This means that you must exercise extreme caution and keep a root shell open before and after tampering with it. Add the lines below to the end of the file.

**CLI:**

editor /etc/pam.d/common-auth

auth required pam\_tally2.so deny=5 onerr=fail unlock\_time=1800

**Configure su (s)**

/etc/pam.d/su controls what users and groups can use the su command. Add the lines below to the end of the file or simply uncomment it.

**CLI:**

auth required pam\_wheel.so

**Ensure authentication for single user mode**

Requiring authentication in single user mode prevents an unauthorized user from

rebooting the system into single user to gain root privileges without credentials.

**CLI:**

**If** # grep ^root:[\*\!]: /etc/shadow

**Has no results:**

Passwd root

# **Network**

**Firewall (s)**

UFW comes installed by default on Ubuntu. You may need to install it if you are on Debian. A firewall is vital for system security.

**GUI:**

Install GUFW

**CLI:**

ufw enable

ufw allow *port/service*

ufw default deny incoming

ufw default allow outgoing

ufw logging on

service ufw restart

**Check Open Ports (s)**

An unauthorized listening port is insecure. All listening ports should be critical or integrated with the system by default. Check if the port closed after running through the commands below.

**CLI:**

netstat -tulpen

ps -aux | grep *pid*

**OR**

lsof -i :*port*

whereis *program*

dpkg -S *location*

**IF PACKAGE:**

apt purge *package*

**IF PROGRAM:**

rm *location* && killall -9 *program*

**Check Hosts File (s)**

The hosts file controls any redirects on the Internet. The file should only be seven lines and should not contain anything suspicious.

**CLI:**

editor /etc/hosts

**Check DNS File (s)**

The file /etc/resolv.conf contains DNS information.

**CLI:**

editor /etc/resolv.conf

**Sysctl (s)**

All these edits are to be appended to /etc/sysctl.conf. This file handles networking configuration.

**Disable IPv6**

**CLI:**

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.lo.disable\_ipv6 = 1

sysctl -p reload config

**IPv4 Security**

**CLI:**

net.ipv4.ip\_forward = 0

net.ipv4.conf.default.accept\_source\_route = 0

net.ipv4.tcp\_syncookies = 1

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.all.log\_martians = 1

net.ipv4.conf.default.secure\_redirects = 1

net.ipv4.icmp\_echo\_ifnore\_broadcasts = 1

net.ipv4.conf.all.rp\_filter = 1

net.ipv4.conf.default.rp\_filter = 1

# **Malicious Software and Files**

**Install and Run Lynis (s)**

Unlike windows, installing an antivirus will not give points, but it may point towards a vulnerability.

**GUI:**

Ubuntu Software Center

**CLI:**

apt install lynis

Lynis audit system

**Install and Run RKHunter (s)**

RKhunter searches for rootkits in the system. Rootkits are difficult to locate without a tool. Check /var/log/rkhunter.log for more details on a WARNING.

**CLI:**

apt install rkhunter

rkhunter --propupd

rkhunter --checkall

**Install and Run ChkRootkit (s)**

ChkRootkit, like RKHunter, searches for rootkits. Known false positives include ‘chkproc: possible LKM trojan installed’, ‘Possible Operation Windigo installed’, and anything that says ‘OooPs’.

**CLI:**

apt install chkrootkit

chkrootkit -q

**Search for Malicious Packages and Files (s)**

Use this method to search for malicious or unnecessary packages. Some things to search for are listed below.

**GUI:**

Synaptic  *(Search every package)*

**CLI:**

dpkg -l | grep *package*

**OR**

find / -iname *pattern*

find / -type d -perm 0777

**SEARCH TERMS:**

* John
* Server
* Sql
* Apache
* Web
* Netcatcmat
* Nc
* Php
* Cain
* Lightweight
* Password
* openbsd-inted
* Crack
* Game
* Ssh
* Vsftpd
* Port
* Metasploit
* Xinetd
* Irc (some clients are integrated)
* Dns (bind9 NOT dnsmasq)
* OPHs
* Nmap
* Tcpdump
* apache
* Cat
* tor
* Wireshark
* Netbus
* Keylog
* Web
* VNC
* Cryptcat
* Hydra
* Msf
* Gopher
* Finger
* Hosts2-ns
* Xfer
* MongoDb
* Mit-ml-dev
* OpenVAS
* Nessus
* TeamViewer
* Ctf
* Rje
* TuxRacer
* Bootpc
* Wordpress ( **DO FORENSIC QUESTION INVOLVING IT FIRST!)**
* Metasploit
* \*.mp4
* \*.mp3
* \*.ogg
* \*.mpg
* \*.mpeg
* \*.xls
* \*.pdf
* \*.txt
* \*.7z
* \*.txt
* \*.zip
* \*.exe
* \*.avi
* \*.mov
* \*.wmv

**Startup Files (s)**

Bash runs code upon starting up in several places. It is important to check these files and root’s home folder, /root/, as well.

**/etc/profile and /etc/profile.d/\***

Check for malicious code

Do not worry about vte file

**/etc/bash.bashrc**

Check for malicious code

**~/.profile**

Check if identical to default

cmp --silent *homefolder*/.profile /etc/skel/.profile && echo “Identical” || echo “Not Identical”

**~/.bashrc**

Can be long, search for specific commands like nc, wall, echo, sudo…

Check if identical to default

cmp --silent *homefolder*/.bashrc /etc/skel/.bashrc && echo “Identical” || echo “Not Identical”

**~/.bash\_logout**

Check if identical to default

cmp --silent *homefolder*/.bash\_logout /etc/skel/.bash\_logout && echo “Identical” || echo “Not Identical”

**/etc/rc.local**

Should be empty

**/etc/rc[0-6].d/**

Holds service start files for different runlevels

Go through each and clean

ls | grep -r \*.sh

**Cron (s)**

Cron runs specific commands at specific times/dates specified in the cron file of each user. You will not have to search through every user (you can in extra time), but search through root and main user for sure. Cronjobs can also hide in /etc/crontab, /etc/cron.d/\*, /etc/crond.hourly, daily, weekly, monthly, /var/spool/cron, /var/spool/cron/crontabs

**CLI:**

crontab -l -u *user*

# **SSH**

SSH may be listed as a critical service on the readme. This means that it should be installed and configured it to be as secure as possible.

**Install OpenSSH (s)**

OpenSSH should be installed.

**GUI:**

Ubuntu Software Center

Install OpenSSH

**CLI:**

apt install openssh-server

**Use Pam (s)**

The pam modules can be imported from the system and used on the SSH server.

**CLI:**

editor /etc/ssh/sshd\_config

UsePAM yes

**Disable Forwarding (s)**

SSH forwarding should be disabled unless specified in the readme.

**CLI:**

editor /etc/ssh/sshd\_config

AllowTcpForwarding no

X11Forwarding no

**Set Login Grace Time**

Login grace time is the amount of time you can spend before signing into an SSH account.

**CLI:**

editor /etc/ssh/sshd\_config

LoginGraceTime 30

**Drop Inactive Connections**

Inactive connections can leave the server vulnerable.

**CLI:**

editor /etc/ssh/sshd\_config

ClientAliveInterval 300

ClientAliveCountMax 0

**Use Protocol 2 (s)**

Protocol 2 is the most secure protocol for SSH. There can only be one “Protocol” line.

**CLI:**

editor /etc/ssh/sshd\_config

Protocol 2

**Disable Host-Based Authentication**

Host-based authentication is not always secure.

**CLI:**

editor /etc/ssh/sshd\_config

HostBasedAuthentication no

IgnoreRHosts yes

**Disable Empty Passwords (s)**

Empty passwords are extremely insecure and should not be used.

**CLI:**

editor /etc/ssh/sshd\_config

PermitEmptyPasswords no

**Enable Strict Mode (s)**

Strict mode should be enabled.

**CLI:**

editor /etc/ssh/sshd\_config

StrictModes yes

**Use Privilege Separation**

Privilege separation should be enabled.

**CLI:**

editor /etc/ssh/sshd\_config

UsePrivilegeSeparation yes

**Disable Root Login (s)**

Root login is insecure and should be disabled.

**CLI:**

editor /etc/ssh/sshd\_config

PermitRootLogin no

**Disable Log Printing**

The log contains vulnerable information and should not be printed.

**CLI:**

editor /etc/ssh/sshd\_config

PrintLastLog no

**Set Login Banner**

The login banner is a critical part of the SSH server and should be configured.

**CLI:**

editor /etc/ssh/sshd\_config

Remove or comment lines that start with “banner”

echo “this is a secure server” > /etc/issue.net

**Create List of Allowed Users**

A list of allowed user should be configured if the readme calls for it.

**CLI:**

editor /etc/ssh/sshd\_config

AllowUsers *user*, *user*…

**Restart SSH**

Points may not be received until the service is restarted.

**CLI:**

service ssh restart

# **FTP**

File Transfer Protocol (FTP) may be listed on the readme as a critical service. If it is listed, it must be installed and configured.

**Install Vsftpd (s)**

Very Safe FTP Daemon (Vsftpd) should be installed.

**GUI:**

Ubuntu Software Center

Install Vsftpd

**CLI:**

apt install vsftpd

**Disable Anonymous Access (s)**

Anonymous access should be disabled.

**CLI:**

editor /etc/vsftpd.conf

anonymous\_enable no

**Jail Users to Home (s)**

Changing the root directory to the users home folder ensures maximum security.

**CLI:**

editor /etc/vsftpd.conf

chroot\_local\_user yes

**Jail Specific Users to Home**

This does the same as above, except only for specific users. Users should be listed one per line in /etc/vsftpd.chroot\_list

**CLI:**

editor /etc/vsftpd.conf

chroot\_list\_enable=YES

**Disable Root Login**

Same as SSH, root login should be disabled. Remove root from the list of ftp users.

**CLI:**

editor/etc/ftpusers

**Disable Writes**

Direct writes should be disabled.

**CLI:**

editor /etc/vsftpd.conf

write\_enable=NO

**Set Max Clients**

Max clients should be set to a reasonable number to ensure maximum security.

**CLI:**

editor /etc/vsftpd.conf

max\_clients=30

**Restart FTP**

If points are not awarded, you may have to restart the service.

**CLI:**

service vsftpd restart

# **Apache**

Apache may be a critical service listed in the readme. If so, it should be installed and secured.

**Install Apache (s)**

Apache should be installed when it is required.

**GUI:**

Ubuntu Software Center

Install Apache

**CLI:**

apt install apache2

**Hide Host OS (s)**

The host OS should not be shown in the webpage.

**CLI:**

editor /etc/apache2/conf-enabled/security.conf

ServerSignature Off

ServerTokens Prod

**Disable Symlinks and Directory Browsing**

These settings leaves the system vulnerable and should be disabled

**CLI:**

editor /etc/apache2/apache2.conf

Options -FollowSymLinks -Includes \_ExecCGI

AllowOverride None

Require all granted

**Limit Large Requests (s)**

Requests should be limited to ensure maximum security.

**CLI:**

editor /etc/apache2/apache2.conf

LimitRequestBody 204800

**Restrict Access to Root Directory**

Access to root directory should be restricted, also, the ability to override this option should be disabled.

**CLI:**

editor /etc/apache2/apache2.conf

‘<Directory>

Require all denied

AllowOverride None

Options None

</Directory>’

**Check PHP Backdoors (s)**

PHP backdoors can be used to access a terminal from within the host system. They should be removed as they are malicious.

**CLI:**

tree -a /var/www | grep \*.php

**Restart Apache**

Sometimes, points will not be awarded until the service restarts.

**CLI:**

service apache restart

# **Samba**

If you need, Creating a samba share for individual users or groups

(Edit words in caps and read,write,valid, with your own preferences)

**Configure valid/invalid users and groups**

**CLI:**

gedit smb.conf

(Edit below:)

[YOURUSER]

​ path = /data/YOURUSER

​ read only = no

​ writeable = yes

​ browseable = yes

​ valid users = YOURUSER for groups use @YOURGROUP

​ create mask = 0640

​ directory mask = 0750

force user = YOURUSER

valid users = YOURUSER YOURUSER YOURUSER (add how many valid users you need)

Invalid users = YOURUSERS YOURUSERS (add how many invalid users you need)

Invalid groups = YOURGROUO YOURGROUP (add how many invalid groups you need)

**Configuring Logging**

**CLI:**

editor smb.conf

log file = /var/log/samba/%m.log

**Setting Maximum Log File Size**

**CLI:**

editor smb.conf

max log size = 10000

**Setting Universal Log Level**

**CLI:**

editor smb.conf

log level = 3

**Setting Individual Log Levels for Debug Classes**

**CLI:**

editor smb.conf

log level = 1 auth:5 winbind:5

**Restart samba**

$smbcontrol all reload-config

# **MySQL**

**Set a strong password**

Ensures that MySQL has a strong password that can aid against default skins

**CLI:**

editor my.cnf

user = root

password = (insert secure password here)

**Use Unix file permissions**

Permits access to only the root user

**CLI:**

$ sudo chown root:root */root/*.my.cnf

$ sudo chmod 0600 */root/*.my.cnf

**Remove anonymous users**

Removes MySQL accounts with no username and no password, secures it by making sure a hacker cannot manipulate your database without a password

**CLI:**

$ /usr/bin/mysql -u root -p

(Enter your previously set password for root)

> SELECT Host, User FROM mysql.user;

(Remove any account with a blank User using):

> drop user “”@”*insert user to delete here*”;

> flush privileges;

**Enable TLS**

Secure network traffic over TLS

**CLI:**

editor my.cnf

(add below lines user [mysqld]

ssl-ca=/path/to/ca.crt

ssl-cert=/path/to/server.crt

ssl-key=/path/to/server.key

(Go into shell)

$ /usr/bin/mysql -u root -p

> grant all privileges on mydb.\* to *someuser*@10.0.1.0/255.255.255.0 identified by ‘*astrongpassword*’ REQUIRE SSL;

> flush privileges;

# **PHP**

**Restrict PHP Information Leakage**

**Restricts PHP information leakage**

**CLI:**

editor /etc/php.d/secutity.ini

expose\_php=Off

**Log All PHP Errors**

**Does not expose PHP errors to all site visitors**

**CLI:**

editor /etc/php.d/security.ini

display\_errors=Off

log\_errors=On

error\_log=var/log/httpd/php\_scripts\_error.log

**Disallow Uploading Files**

**Disable file uploads for security reasons**

**CLI:**

editor /etc/php.d/security.ini

file\_uploads=Off

**Turn Off Remote Code Execution**

**Helps prevent against injection attacks**

**CLI:**

editor /etc/php.d/security.ini

allow\_url\_fopen=Off

allow\_url\_include=Off

**Control POST size**

**Attackers may attempt to send oversized POST requests**

**CLI:**

editor /etc/php.d/security.ini

post\_max\_size=1K

**Resource Control**

**Setting maximum execution time of PHP scripts that could eat up system resources**

**CLI:**

editor /etc/php.d/security.ini

max\_execution\_time = 30

max\_input\_time = 30

memory\_limit = 40M

**Disabling Dangerous PHP Functions**

**PHP has many functions that can be used to crack the server**

**CLI:**

editor /etc/php.d/security.ini

disable\_functions =*insert prohibited functions here*

**PHP Fastcgi / CGI – cgi.force\_redirect Directive**

**Prevents anyone from calling PHP directly with a URL like http://www.cyberciti.biz/cgi-bin/php/hackerdir/backdoor.php.**

**CLI:**

editor /etc/php.d/security.ini

cgi.force\_redirect=On

**Restrict File and Directory Access**

**All files and directory should be owned by non-root user**

**CLI:**

chown -R apache:apache /var/www/html/

chmod -R 0444 /var/www/html

Find /var/www/html/ -type d -print0 | xargs -0 -I {} chmod 0445 {}

**Write Protect Apache , PHP, and, MySQL Configuration**

**Files**

**WARNING: WRITE PROTECTS CONFIG FILES!!**

**CLI:**

chattr +i /etc/php.ini

chattr +i /etc/php.d/\*

chattr +i /etc/my.ini

chattr +i /etc/httpd/conf/httpd.conf

chattr +i /etc/

chattr +i /var/www/html/file1.php

chattr +i /var/www/html/

**Search for PHP Backdoors**

**Searches C99, R57 shells**

**CLI:**

grep -iR 'c99' /var/www/html/

grep -iR 'r57' /var/www/html/

find /var/www/html/ -name \\*.php -type f -print0 | xargs -0 grep c99

grep -RPn "(passthru|shell\_exec|system|base64\_decode|fopen|fclose|eval)" /var/www/html/

# **DNS**

**Install Stubby:**

***Utility that will help encrypt DNS***

**CLI:**

# **Last Ditch Efforts**

This section is for when you cannot find the final vulnerabilities.

**Reread the Readme**

The readme can contain vital information, so you should double-check to make sure all requirements are met.

**File Permissions**

Check important files to make sure their permissions are properly configured. Some files to check are /etc/passwd, /etc/shadow, /etc/sudoers, and /etc/group

**CLI:**

ls -al *file*

**Check Important Directories**

Check important directories to make sure they do not contain anything out of the ordinary. Some directories to check are the root directory, /root/, /etc/, /bin/.

**GUI:**

Nautilus

**CLI:**

ls -al *directory*

**Review Benchmarks**

CIS and similar benchmarks can be very useful when searching for more obscure vulnerabilities.

# **Notes**

This guide is still a work in progress and is optimized for Ubuntu, but the CLI commands should work on Debian.

Remote Desktop (Desktop Sharing) should be enabled or disabled depending on readme.

Personal File Sharing

Startup Applications and bum

Use [packages.ubuntu.com](https://packages.ubuntu.com/) to research any packages before removing them

If there are ports opened by a command, grep startup/cron files for the command

Security and Privacy > Files and Applications > Record file and application usage

Install auditd and auditctl -e 1

Use last to check what users have logged in

Webmin for configuring services

Enforce SELinux if it is a critical service

by editing /etc/selinux/config and set the parameter to

SELINUX=enforcing

SECURE FTP GUIDE - <http://xmodulo.com/secure-ftp-service-vsftpd-linux.html>

SECURE SAMBA GUIDE - <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-samba-share-for-a-small-organization-on-ubuntu-16-04>

SECURE PHP GUIDE - <https://www.cyberciti.biz/tips/php-security-best-practices-tutorial.html>

SECURE MYSQL GUIDE - <https://www.pontikis.net/blog/how-and-when-to-enable-mysql-logs>

<https://dev.mysql.com/doc/refman/8.0/en/password-security.html> (more stuff on webpage)

<https://medium.com/linode-cube/5-essential-steps-to-hardening-your-mysql-database-591e477bbbd7>

SECURE LAMP GUIDE - <https://www.linode.com/docs/web-servers/lamp/lamp-on-ubuntu-14-04/>

<https://tecadmin.net/security-tips-for-lamp-stack-on-linux/>

<https://wiki.debian.org/LaMp>

SECURE GRUB STOOPID

$ grub-md5-sum