Sai's Mega Useful guide to Linux Hardening (Made for Cyber Centurion VI)

3rd Party tools to consider

- LinEnum.sh - https://github.com/rebootuser/LinEnum
- LinuxAudit.sh - https://raw.githubusercontent.com/sokdr/LinuxAudit/master/LinuxAudit.sh
- Pspy - https://github.com/DominicBreuker/pspy

Webmin

This has a link to every possible thing you may need

Setup

Run

```
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.900_all.deb -O
~/Desktop/webmin_1.900_all.deb

dpkg --install ~/Desktop/webmin_1.900_all.deb
```

Dependencies

For missing dependencies run

```
sudo apt install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-
runtime libio-pty-perl apt-show-versions python
```

Running

The application can be accessed at

```
http://localhost:10000/
```

General

Firefox

Disable pop-ups

Open **FireFox** →Click on the **three lines** on the top right of the browser →Select the menu button on the top right to enter **Preferences** →Select the **Content** tab on the left-hand side of the window →Check the box next to **Block pop-up windows.**

Config files

Add the following lines to the files

`/usr/lib/firefox/defaults/pref/local-settings.js`

```
pref("general.config.filename", "mozilla.cfg");
```

`/usr/lib/firefox/mozilla.cfg`

```
lockPref("browser.safebrowsing.downloads.enabled", true);
lockPref("dom.disable_open_during_load", true);
lockPref("xpinstall.whitelist.required", true);  # Scored
lockPref("xpinstall.signatures.required", true); # Scored
lockPref("app.update.enabled", true);
lockPref("app.update.auto", true);
lockPref("privacy.donottrackheader.enabled", true);
lockPref("browser.safebrowsing.downloads.remote.block_potentially_unwanted",
true);
lockPref("browser.safebrowsing.downloads.remote.block_uncommon", true);
lockPref("browser.safebrowsing.malware.enabled", true); # scored
lockPref("browser.safebrowsing.phishing.enabled", true);
```

Deleting Potentially Dangerous Files

This is basically check for certain files in certain directories

```
# Media
find / -name '*.mp3' -type f -delete
find / -name '*.mov' -type f -delete
find / -name '*.mp4' -type f -delete
find / -name '*.avi' -type f -delete
find / -name '*.mpg' -type f -delete
find / -name '*.mpeg' -type f -delete
find / -name '*.flac' -type f -delete
find / -name '*.m4a' -type f -delete
find / -name '*.flv' -type f -delete
find / -name '*.ogg' -type f -delete
find /home -name '*.gif' -type f -delete
find /home -name '*.png' -type f -delete
find /home -name '*.jpg' -type f -delete
find /home -name '*.jpeg' -type f -delete

# Files
find /home -name '*.txt' -type f -delete
find / -name '*.xlsx' -type f -delete
find / -name '*.csv' -type f -delete

# Finding files with passwords in
find / -name "*password.txt" -type f -delete
find / -name "*passwords.txt" -type f -delete

# Try in /home/, /etc/, /usr/
grep -iRl "Password in Readme" <dir>

# Careful with these and try /home too
find /bin/ -name '*.sh' -type f -delete
find /bin/ -name '*.py' -type f -delete
find /bin/ -name '*.exe' -type f -delete
find /bin/ -name '*.bat' -type f -delete
find /bin/ -name '*.c' -type f -delete
find /bin/ -name '*.pl' -type f -delete
find /bin/ -name '*.php' -type f -delete
```

Logs

A lot of times in these logs, you will find about cronjobs and other commands which run many times. Reading through these logs is vital. These can be found at `/var/log/`

First you should install auditing by doing

```
sudo apt install -y auditd
sudo apt install -y rsyslog

systemctl is-enabled auditd
systemctl enable auditd
```

Two interesting logs to look at are

```
/var/log/auth.log
/var/log/syslog
/var/log/apt/history.log
```

You can `grep` for any `netcat` backdoors straightaway by doing

```
grep -i nc /etc/syslog
```

LAMP Stack

```
Linux
Apache2
MySQL
Php
```

Alternatives

- WAMP: Windows, Apache, MySQL, Php
- WISA: Windows, IIS, SQL, ASP.net

- MAMP: MacOS, Apache, MySQL, PHP

Uninstalling

```
# Main packages
sudo apt purge mysql-server apache2 php5

# All packages (others may depend on these)
sudo apt remove apache2 apache2-mpm-prefork apache2-utils apache2.2-common
libapache2-mod-php5 libapr1 libaprutil1 libdbd-mysql-perl libdbi-perl
libmysqlclient15off libnet-daemon-perl libplrpc-perl libpq5 mysql-client-5.0
mysql-common mysql-server mysql-server-5.0 php5-common php5-mysql
```

Installing

```
sudo apt install mysql-server apache2 php5

# Look under Services->Securing MySQL to learn how to securely set it up
```

Users

Enumeration of regular users

Prints all users

```
getent passwd
```

Since we are only interested in the users that use a shell we can also do:

```
getent passwd | egrep "/bin/.*sh" | cut -d: -f1
```

Prints all `sudoers`

```
getent group | grep sudo
```

We can print it more nicely

```
getent group | grep sudo | cut -d":" -f4
```

Check for UID 0

```
/etc/passwd: paul:x:0:1012
```

Login shell

Authorized users with uid 1-1000 and have login shell (`/bin/sh`, `/bin/bash` etc...)

If user is supposed to exist, change shell to `/bin/false`

```
whoopsie:x:109:117::/nonexistent:/bin/sh

# to

whoopsie:x:109:117::/nonexistent:/bin/false
```

Shadow file

File format:

```
1. Username
2. Encrypted Password
3. Date of last password change
4. Minimum password age
5. Maximum password age
6. Password warning period
7. Password inactivity period
8. Account expiration date
9. Reserved field

example
mageshm:$6$<...>:16088:0:99999:7: : :
|------|--------|-----|-|-----|-|-|-|-|
   1       2       3   4   5   6 7 8 9
```

If password field contains `!` or `*` (not a valid result of crypt(3)) this means **user cannot login with a unix password**

This field may be **empty**, in which case **no passwords are required to authenticate**

Password Expiration

NO users with UID 1000-60000 has `:99999:` as `5th` thing in their line

Root login no password

Root login has `root::` instead of `root:!:`

Users weak password hash

Users have anything other than `$6$` at the start of their password hash

Changing passwords

You can just pipe all the users (other than `root` because we don't want to change that password) into `chpasswd`

The command will look like `echo username:newpassword | chpasswd`

Typically I use a secure password which I want to write down - `Qu1CkF0x!23` be careful that 0 <- is a number.

```
echo "<user>:CyberPatriot1!" | chpasswd
```

We can loop over every user and change their password as such (login as `root` first)

```
for i in `getent passwd | egrep /bin/bash | grep -v root | awk '{split($0, a, ":"); print a[1]}'`; do `echo $i':CyberPatriot1!' | chpasswd`; done;
```

Deleting Users

To do this it is just a simple command (make sure you have root permissions though):

```
userdel -r <username>
```

Warning!!! DONT USE SLAY, INSTEAD DO 2ND METHOD

You can use the `slay` command to stop all user processes and logout the user

```
slay <user>
```

Or you can manually delete

```
Delete entry from /etc/passwd
Delete entry from /etc/shadow
Delete /home/$USER directory
Delete any /var/mail etc...
```

Adding Users

Similar to deleting users, make sure you have root permissions

```
useradd <username>
```

Sudoers

Checks for success

- No mention of NOPASSWD
- No files in /etc/sudoers.d/
- Check /etc/sudoers.d/README
- Only authorised users having access to sudo

Adding

```
gpasswd -a <username> sudo
```

Removing

```
gpasswd -d <username> sudo
```

To also check for Sudoers permissions do

```
# Login as root as well and try this
sudo -l
```

To edit these options you can do

```
visudo
```

The file should look similar to

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.## See the man page for details on how to write a
sudoers file.
#Defaults env_resetDefaults mail_badpassDefaults
secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/
bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
```

```
root ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:#includedir
/etc/sudoers.d
```

Bad lines:

```
%topsecret ALL=(ALL:ALL) NOPASSWD: /usr/bin/sudo, /bin/bash
```

The root account

Actions taken to harden the root user account

Locking

```
usermod -L root
```

Unlocking

```
usermod -U root
```

Disabling

```
passwd -l root
```

Protecting

```
usermod -g 0 root
```

Lightdm

The configuration file should look this this

The two files it can be

```
/etc/lightdm/lightdm.conf
/etc/lightdm/lightdm.conf.d/50-myconfig.conf
```

```
[SeatDefaults]
allow-guest=false # Definitely scored
greeter-hide-users=true
greeter-show-manual-login=true
greeter-allow-guest=false
autologin-user=none # Be careful with this (dont change this is comp)
autologin-guest=false
AutomaticLoginEnable=false # Be careful with this (dont change this is comp)
xserver-allow-tcp=false
```

Make sure to restart **lightdm**

```
sudo service lightdm restart
```

User Policies

PAM

Password

This enforces authentication policy such as password length, characters etc...

First we need to make sure that `libpam-cracklib` is installed

```
sudo apt install --force-yes -y libpam-cracklib
```

Next we edit `/etc/pam.d/common-password`

```
password requisite pam_cracklib.so retry=3 minlen=8 difok=3 dcredit=-1 ucredit=-1 lcredit=-1 ocredit=-1

# Various options retry=3 minlen=8 difok=3 reject_username minclass=3 maxrepeat=2 dcredit=1 ucredit=1 lcredit=1 ocredit=1

password [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass sha512 minlen=8 remember=5p

# Not used before use as last resort
password requisite pam_pwhistory.so use_authtok remember=5 enforce_for_root

# Remove any mention of nullok or nullok_secure
sed -i 's/nullok//g'
```

Then we can install `fail2ban` which is

> an intrusion prevention software framework that protects computer servers from brute-force attacks

```
sudo apt install --force-yes -y fail2ban
```

Login

Edit `/etc/login.defs`

```
PASS_MAX_DAYS             90
PASS_MIN_DAYS             10
PASS_WARN_AGE              7
FAILLOG_ENAB             yes
LOG_UNKFAIL_ENAB         yes
LOG_OK_LOGINS            yes
SYSLOG_SU_ENAB           yes
SYSLOG_SG_ENAB           yes
LOGIN_RETRIES              5
ENCRYPT_METHOD        SHA512
SU_NAME                   su
MD5_CRYPT_ENAB           yes
LOGIN_TIMEOUT             60

UMASK                   077
```

Edit `/root/.bashrc`

```
umask 077 # Uncomment this line and edit
```

Edit `/etc/init.d/rc`

```
umask 027 # Edit the line
```

You can also use `chage`

```
chage -m 10 <user>
chage -M 90 <user>
chage -W 7  <user>
```

Edit `/etc/pam.d/login`

```
auth       optional   pam_faildelay.so  delay=10000000 # 10 seconds
```

Account Policy

This changes how many incorrect logins you're allowed and how long you have to wait. This is done by editing `/etc/pam.d/common-auth` and you have to just add

to the end of the file

```
echo "auth required pam_tally2.so deny=5 onerr=fail unlock_time=1800 audit
even_deny_root_account silent" >> /etc/pam.d/common-auth

# Remove any mention of nullok or nullok_secure
sed -i 's/nullok//g'
```

Su

You don't want any sudo to be able to use su and login as root

To disable this, you can either:

Uncomment the line with `pam_wheel.so` in

```
/etc/pam.d/su
```

Or

Uncomment `SU_WHEEL_ONLY` in

```
/etc/logins.def
```

APT

Sources.list

Each OS version requires different sources.list. I will be talking solely about Ubuntu Linux from `14.04` to `18.04`. Otherwise it only is a simple google search to find the default sources.

To detect Ubuntu version, use the command

```
lsb_release -r
```

Next make a backup of your sources.list

```
mv /etc/apt/sources.list /etc/apt/sources.list.bak
```

Next depending on your version copy over these files:

```
UBUNTU 14.04

wget
https://gist.githubusercontent.com/justbuchanan/d045498c95fa1cd2dc70/raw/789264b
08c87c3ae8d836c91887f57d7d6496851/sources.list -O /etc/apt/sources.list
```

```
UBUNTU 16.04

wget
https://gist.githubusercontent.com/rohitrawat/60a04e6ebe4a9ec1203eac3a11d4afc1/r
aw/fcdfde2ab57e455ba9b37077abf85a81c504a4a9/sources.list -O
/etc/apt/sources.list
```

```
UBUNTU 17.04

wget
https://gist.githubusercontent.com/ChampionCynthia/e4deb4410105c3ecdffb85630a4c5
b96/raw/e3bb77c5045e2a11d5a381476c5559362ccfbbeb/sources.list -O
/etc/apt/sources.list
```

```
UBUNTU 18.04

wget
https://gist.githubusercontent.com/h0bbel/4b28ede18d65c3527b11b12fa36aa8d1/raw/3
14419c944ce401039c7def964a3e06324db1128/sources.list -O /etc/apt/sources.list
```

A useful website is https://mirrors.ustc.edu.cn/repogen/

Example File

```
deb http://us.archive.ubuntu.com/ubuntu/ "NAME" main restricted
deb http://us.archive.ubuntu.com/ubuntu/ "NAME"-updates main restricted
deb http://us.archive.ubuntu.com/ubuntu/ "NAME" universe
deb http://us.archive.ubuntu.com/ubuntu/ "NAME"-updates universe
deb http://us.archive.ubuntu.com/ubuntu/ "NAME" multiverse
deb http://us.archive.ubuntu.com/ubuntu/ "NAME"-updates multiverse
deb http://us.archive.ubuntu.com/ubuntu/ "NAME"-backports main restricted
universe multiverse
deb http://security.ubuntu.com/ubuntu "NAME"-security main restricted
deb http://security.ubuntu.com/ubuntu "NAME"-security universe
deb http://security.ubuntu.com/ubuntu "NAME"-security multiverse
```

Change `"NAME"` :

```
Ubuntu 14.04 - trusty
Ubuntu 16.04 - bionic
Ubuntu 17.04 - zesty
Ubuntu 18.04 - xenial
```

Updating and upgrading

To update and upgrade, they only require two simple commands - although they may take a long time to complete.

```
sudo apt update && sudo apt upgrade

# Or (recommended)
sudo apt dist-upgrade
```

The `&&` ensure that you upgrade only if update succeeds. Otherwise you may install broken or bad packages

Auto Updates

```
# If files are not present
sudo apt install unattended-upgrades
```

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
APT::Periodic::AutocleanInterval "7";
```

On **older ubuntu** versions, you will do it on `/etc/apt/apt.conf.d/10periodic`

On **newer ubuntu** versions, you will do it on `/etc/apt/apt.conf.d/20auto-upgrades`

CHECK FOR BOTH FILES AND CHANGE ON ALL FILES PRESENT

An easy way to find which file you require is doing

```
grep "APT::Periodic::Update-Package-Lists" -r /etc/apt/apt.conf.d/
```

Getting Apt-Fast

```
sudo add-apt-repository ppa:apt-fast/stable
sudo apt update
sudo apt install curl
sudo apt install realpath
sudo apt update
/bin/bash -c "$(curl -sL https://git.io/vokNn)"
sudo sed -i '/_MAXNUM/c_MAXNUM=20' /etc/apt-fast.conf
```

Do

```
apt-fast upgrade
apt-fast dist-upgrade
```

Installed Packages

Listing

To list all the installed packages, you can do either

```
dpkg -l
apt-mark showmanual
dpkg --get-selections
```

`apt-mark showmanual` is for packages installed manually

Deleting malicious tools

Here is a list of a lot of the bad bad tools to delete:

```
john, abc, sqlmap, aria2
aquisition, bitcomet, bitlet, bitspirit
"endless-sky", "zenmap", "minetest", minetest-server
armitage, crack, apt pureg knocker, aircrack-ng
airbase-ng, hydra, "freeciv"
"wireshark", tshark
hydra-gtk, "netcat", netcat-traditional, netcat-openbsd
netcat-ubuntu, netcat-minimal, "qbittorrent", ctorrent
ktorrent, rtorrent, deluge, transmission-common
"transmission-bittorrent-client", tixati, frostwise, vuse
irssi, transmission-gtk, "utorrent", "kismet"
medusa, "telnet", exim4, telnetd
bind9, crunch, "tcpdump", tomcat
tomcat6, "vncserver", tightvnc, tightvnc-common
tightvncserver, vnc4server, nmdb, dhclient
telnet-server, "ophcrack", cryptcat, cups
cupsd, "tcpspray", "ettercap"
"wesnoth", snort, pryit
weplab, "wireshark", nikto, lcrack
postfix, snmp, icmp, dovecot
pop3, "p0f", "dsniff", "hunt"
ember, nbtscan, rsync, freeciv-client-extras
freeciv-data, freeciv-server, freeciv-client-gtk
```

I highlighted the main ones.

You will need to do this command where `$i` is the name of the file (you could use a for loop to do this)

```
sudo apt purge -y --force-yes $i
```

Delete games folders

```
/usr/lib/games
/usr/local/games
/usr/share/games
/var/games
/var/lib/games

find / type d -iname "games"
```

Services

Securing Services

Securing SSH

We don't want anyone to be able to become root from ssh. We can just use some trusty regex and `sed` to change this in `/etc/ssh/sshd_config`

```
sed -i '/^PermitRootLogin/ c\PermitRootLogin no' /etc/ssh/sshd_config
```

All these settings can be changed in a similar manner

```
Port 222 # Only change this if needed

Protocol 2
PermitRootLogin no
PermitEmptyPasswords no

X11Forwarding no
AllowTcpForwarding no

UsePAM yes
PasswordAuthentication yes
HostBasedAuthentication no
StrictModes yes

UsePrivilegeSeparation yes
PrintLastLog no
PermitUserEnvironment no

LogLevel INFO
MaxAuthTries 4
IgnoreRhosts yes # Force user entering password

Ciphers aes128-ctr,aes192-ctr,aes256-ctr
ClientAliveInterval 300
ClientAliveCountMax 0
Banner /etc/issue.net # Change this file to remove fingerprinting

# Other
```

```
AllowGroups wheel admin
AllowUsers alex ref me@somewhere
DenyUser bad_user1 bad_user_2
```

Edit `/etc/issue.net`

```
echo "Hello" > /etc/issue.net
```

Make sure to restart SSH

```
sudo service ssh restart
```

It may also be called `sshd` or `openssh`. Make sure to use the package name.

More info at: http://bookofzeus.com/harden-ubuntu/hardening/ssh/

Securing Samba

Edit or add the lines `/etc/samba/smb.conf`

```
restrict anonymous = 2
encrypt passwords = True
encrypt passwords = yes
read only = Yes
ntlm auth = no
obey pam restrictions = yes
server signing = mandatory
smb encrypt = mandatory
min protocol = SMB2
protocol = SMB2
guest ok = no
max log size = 24
```

Securing FTP

Config locations:

```
# FTPd
/etc/ftpd.conf

#VsFTPd
/etc/vsftpd/vsftpd.conf

# ProFTPd
/etc/proftpd.conf
/usr/local/proftpd.conf
```

Change anonymous to off - check by trying to login to `localhost`

```
$ ftp localhost

USER: anonymous
PASSWORD: <literally anything>
```

Restart by using

```
service <proftpd/ftpd> restart
```

VsFTPd

```
/etc/vsftpd/vsftpd.conf

# Jail users to home directory (user will need a home dir to exist)
chroot_local_user=YES
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
allow_writeable_chroot=YES # Only enable if you want files to be editable

# Allow or deny users
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO

# General config
anonymous_enable=NO      # disable  anonymous login
local_enable=YES         # permit local logins
write_enable=YES         # enable FTP commands which change the filesystem
local_umask=022          # value of umask for file creation for local users
dirmessage_enable=YES    # enable showing of messages when users first enter a
new directory
xferlog_enable=YES       # a log file will be maintained detailing uploads and
downloads
connect_from_port_20=YES # use port 20 (ftp-data) on the server machine for PORT
style connections
xferlog_std_format=YES   # keep standard log file format
listen=NO                # prevent vsftpd from running in standalone mode
listen_ipv6=YES          # vsftpd will listen on an IPv6 socket instead of an
IPv4 one
pam_service_name=vsftpd  # name of the PAM service vsftpd will use
userlist_enable=YES      # enable vsftpd to load a list of usernames
tcp_wrappers=YES         # turn on tcp wrappers

ascii_upload_enable=NO
ascii_download_enable=NO
```

PureFTPd

```
echo "yes" >> /etc/pure-ftpd/conf/NoAnonymous
echo "yes" >> /etc/pure-ftpd/conf/ChrootEveryone
echo "yes" >> /etc/pure-ftpd/conf/IPV4Only
echo "yes" >> /etc/pure-ftpd/conf/ProhibitDotFilesWrite
echo "2" > /etc/pure-ftpd/conf/TLS
echo 2 |  tee /etc/pure-ftpd/conf/TLS
echo 1 |  tee /etc/pure-ftpd/conf/NoAnonymous
```

ProFTPd

Edit or add the lines in `/etc/proftpd/proftpd.conf`

```
DenyFilter \*.*/
DelayEngine on
UseLastLog on
ServerIndent off
IdentLookups off
TLSEngine on
TLSProtocol SSLv23
TLSRequired on
UseReverseDNS on
```

After applying changes

```
systemctl restart <service>
service <service> restart
```

Securing MySql

Secure Installation

```
apt install mysql-server
sudo mysql_secure_installation
```

Config file: `/etc/mysql/my.cnf`

```
[mysqld]
local-infile=0                # Stop mysql reading files from local file system
skip-show-database            # Lowers database privelages
bind-address=127.0.0.1        # Disable remote access
symbolic-links=0              # Disables symbolic links
default_password_lifetime=90  # Set password expiration

[mysqladmin]                  # Sets root account password
user=root
password=<PASSWORD>
# Packet Restrictions
key_buffer_size=16M
max_allowed_packet=16M
```

Change/Read root password

```
service mysql stop
mysqld_safe --skip-grant-tables &

#Now you can go into mysql as root
mysql -u root

UPDATE mysql.user SET Password=PASSWORD('NEW-PASSWORD') WHERE User='root';
#or
SELECT * from mysql.user;
#Now crack the hash
```

Enable SSL

```
mkdir ~/cert && cd ~/cert

openssl genrsa 2048 > ca-key.pem
openssl req -sha1 -new -x509 -nodes -key ca-key.pem -subj "/CN=certificate-authority" > ca-cert.pem

openssl req -sha1 -newkey rsa:2048 -nodes -keyout server-key.pem -subj "/CN=mysql-server" > server-req.pem
openssl x509 -sha1 -req -in server-req.pem  -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 > server-cert.pem
openssl rsa -in server-key.pem -out server-key.pem

openssl req -sha1 -newkey rsa:2048 -nodes -keyout client-key.pem -subj "/CN=mysql-client" > client-req.pem
openssl x509 -sha1 -req -in client-req.pem -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 > client-cert.pem
openssl rsa -in client-key.pem -out client-key.pem

mkdir -p /etc/mysql/ssl
cp ca-cert.pem server-cert.pem server-key.pem /etc/mysql/ssl
chown -R mysql.mysql /etc/mysql/ssl
chmod -R 700 /etc/mysql/ssl
```

Now edit `/etc/mysql/my.cnf`

```
[mysqld]
ssl-ca=   /etc/mysql/ssl/ca-cert.pem
ssl-cert= /etc/mysql/ssl/server-cert.pem
ssl-key=  /etc/mysql/ssl/server-key.pem
```

Restart and check if it worked

```
$ /etc/init.d/mysql restart
$ mysql -u root -e "SHOW GLOBAL VARIABLES LIKE 'have_%ssl';"
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| have_openssl  | YES   |
| have_ssl      | YES   |
+---------------+-------+
```

Securing Apache

Just try and go to

```
http://localhosts/
```

If this exists, you can go to `/var/www/html` to edit files etc.

Check these things

```
Not running as root user
Apache account has invalid shell (/etc/passwd)

chown -R root:root /etc/apache2
chown -R root:root /etc/apache
```

Disable unnecessary modules

List modules:

```
apache2 -l
grep -r LoadModule /etc/apache2/mods-enabled/*
```

Enable/Disable modules:

```
a2enmod userdir
a2enmod headers
a2dismod imap
a2dismod include
a2dismod info
a2dismod userdir
a2dismod autoindex
```

Mod_security

```
apt install mod_security
service httpd restart
```

Security Configuration

Edit `/etc/apache2/conf-available/security.conf` add or edit the following lines

```
# Enable HTTPOnly and Secure Flags
Header edit Set-Cookie ^(.*)\$ \$1;HttpOnly;Secure

# Clickjacking Attack Protection
Header always append X-Frame-Options SAMEORIGIN

# XSS Protection
Header set X-XSS-Protection "1; mode=block"

# Enforce secure connections to the server
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains"

# MIME sniffing Protection
Header set X-Content-Type-Options: "nosniff"

# Prevent Cross-site scripting and injections
Header set Content-Security-Policy "default-src 'self';"
```

Regular configuration

File: `/etc/apache2/apache2.conf`

Add the lines inside within a previous existing root node if possible

```
HostnameLookups Off
LogLevel warn
ServerTokens Prod
ServerSignature Off
Options all -Indexes
Header unset ETag
Header always unset X-Powered-By
FileETag None
```

```
TraceEnable off
Timeout 60
RewriteEngine On

# Secure /
<Directory />
  Options -Indexes
  AllowOverride None
  Order Deny,Allow
  Options None
  Deny from all
</Directory>

# Secure /var/www/html

<Directory />
  Options -Indexes -Includes
  AllowOverride None
  Order Allow,Deny
  Deny from All
</Directory>

# Disable old protocol (HTTP 1.0)
RewriteEngine On
RewriteCond %{THE_REQUEST} !HTTP/1\.1$
RewriteRule .* - [F]

# Disable SSI (Server Side Inclusion)
# Search for Directory and add Includes in Options directive:
<Directory /path/to/htdocs>
  Options -Indexes -Includes
  Order allow,deny
  Allow from all
</Directory>

# Disable CGI execution
# Similar to SSI, you can disable CGI Execution in the "apache2.conf" by adding
the "-ExecCGI" option.
<Directory /path/to/htdocs>
  Options -Indexes -Includes -ExecCGI
  Order allow,deny
  Allow from all
</Directory>
```

Then restart apache

```
sudo service apache2 restart
```

See more at: http://bookofzeus.com/harden-ubuntu/hardening/apache/

Securing Nginx

Remove default page

```
echo > /var/www/index.html
```

Edit or add these lines in `/etc/nginx/nginx.conf`

```
# Hide nginx version
server_tokens off;

# Remove etags
etag off;

# Strong cipher suites
ssl_prefer_server_ciphers on;
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;

# Set ssl session timeout
ssl_prefer_server_ciphers on;
ssl_session_timeout 5m;
```

Edit or add these lines in `/etc/nginx/sites-available/default`

Inside the `server{...}`

```
# Enable HttpOnly and Secure flags
proxy_cookie_path / "/; HTTPOnly;   Secure";
# Clickjacking Attack Protection
add_header X-Frame-Options DENY;
# XSS Protection
add_header X-XSS-Protection "1; mode=block";
# Enforce secure connections to the server
add_header Strict-Transport-Security "max-age=31536000; includeSubdomains;";
# MIME sniffing Protection
add_header X-Content-Type-Options nosniff;
# Prevent Cross-site scripting and injections
add_header Content-Security-Policy \"default-src 'self';\";
# Set X-Robots-Tag
add_header X-Robots-Tag none;
```

Securing PHP

```
ufw allow php
```

php.ini

Find the file by doing

```
php -i | grep "php.ini"
```

Edit or add the following lines

```
# Safe mode
safe_mode=On
safe_mode_gid=On

# Disable Global variables
register_globals=off

# Disable tracking, HTML, and display errors
track_errors=Off
html_errors=Off
display_errors=Off
expose_php=Off
track_errors=Off
html_errors=Off
display_errors=Off
mail.add_x_header=Off

# Disable Remote File Includes
allow_url_fopen=Off
allow_url_include=Off

# Restrict File Uploads
file_uploads=Off

# Control POST/Upload size
post_max_size=1K
upload_max_filesize=2M

# Protect sessions
session.cookie_httponly=1

# General
magic_quotes_gpc=Off
session.use_strict_mode=On

disable_functions=exec,shell_exec,passthru,system,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source,proc_open,pcntl_exec

max_execution_time=30
max_input_time=30
open_basedir="/home/user/public_html" # -> correct html base dir
memory_limit=40M
```

Suhosin

Install

```
apt install php5-suhosin -y
```

Edit or add the lines in `/etc/php5/conf.d/suhosin.ini`

```
extension=suhosin.so
suhosin.session.encrypt=Off
suhosin.log.syslog=511
suhosin.executor.include.max_traversal=4
suhosin.executor.disable_eval=On
suhosin.executor.disable_emodifier=On
suhosin.mail.protect=2
suhosin.sql.bailout_on_error=On
```

Securing DNS / Hosts

Bind9

If `bind9` exists

Open a Terminal and enter the following :

```
sudo vi /etc/bind/named.conf.options
```

Add the following to the **Options** section :

```
recursion no;
version "Not Disclosed";
```

Restart BIND DNS server. Open a Terminal and enter the following :

```
sudo service bind9 restart
```

(Look in more info for links)

Enumerating Services

Run Level Services

You can change the services that run at each run level (same as ring levels in windows)

Install

```
sudo apt install sysv-rc-conf

sysv-rc-conf
```

And check/uncheck each service at the run levels

Disabling / Enabling Services

Unnecessary services are a thing. Disable / uninstall whatever you don't need.

Obvious services include

```
Email - smtp
Samba - smb
Telnet - telnet
SSH - openssh
FTP - vsftpd
Apache
SNMP - snmpd
Cups - cups
autofs
nfs
```

You can use the command `systemctl stop` to stop services

```
systemctl stop apache2
systemctl stop snmpd
systemctl stop ssh
systemctl stop smbd
systemctl stop vsftpd

service pop3 stop
service icmp stop
service sendmail stop
service autofs stop
```

Otherwise you can use BUM (Boot Up Manager) - clever name btw

Other disabling

These will be checking then action if not `disabled`

```
# Check if service is enabled
systemctl is-enabled <service>

# Check if package is installed
dpkg -s <package>

# Stop service
service <service> stop
systemctl disable <service
systemctl disable cups # Scored
apt purge openbsd-inetd
apt purge xserver-xorg*
systemctl disable avahi-daemon  # Scored
systemctl disabled isc-dhcp-server
systemctl disabled isc-dhcp-server6
systemctl disabled slapd
apt purge ldap-utils
systemctl disable autofs     # Scored
systemctl disable nfs-server # Scored
systemctl disable rpcbind    # Scored
systemctl disable bind9      # Scored
systemctl disable dovecot
systemctl disable squid
systemctl disable rsync
systemctl disable nis
apt purge nis
apt purge talk
apt purge telnet # Scored
```

Enumeration

To list services that may be running you can do the following

```
# Run these commands as root

service --status-all

ps aux  | column
ps Zaux | column
ps -ef  | column
ps aux  | grep python

bum
pspy
LinEnum.sh -t
unhide -m -d -f -v  sys procall brute reverse
```

Processes running as root

```
ps aux | grep root
```

Startup / scheduling tasks

Cronjobs

These are actions run at regular intervals

Locations:

```
ls /var/spool/cron/* 2> /dev/null
ls /etc/crontab 2> /dev/null
ls /etc/cron.d/* 2> /dev/null
ls /etc/cron.hourly/ 2> /dev/null
ls /etc/cron.daily/* 2> /dev/null
ls /etc/cron.weekly/* 2> /dev/null
ls /etc/cron.monthly/* 2> /dev/null
```

Restoring crons to default:

To restore first make a backup

```
crontab -l ~/Desktop/crontab.bak
```

Next you can run the simple command

```
crontab -r
```

Boot-up scripts

Check:

```
/etc/init/
/etc/init.d/
/etc/crontab
/etc/rc.local # Should only have exit 0
/etc/rc.d/rc.local
```

For anything unusual

Systemctl services

Systemctl

This is the top-level process that manages all other processes on the system.

Use this command to see all system services

```
systemctl list-unit-files --type service
```

Systemctl-xinetd

This feature of systemctl will start services all demand. These sockets will show as `listening` using ss, but may have PID of 1.

Use this command to see these

```
systemctl list-unit-files --type socket
```

Networking

Hardening connections

You need to set all of these values

```
# IPv4 TIME-WAIT assassination protection
net.ipv4.tcp_rfc1337=1 # Scored

# IP Spoofing protection, Source route verification
# Scored
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.default.rp_filter=1

# Ignore ICMP broadcast requests
net.ipv4.icmp_echo_ignore_broadcasts=1

# Ignore Directed pings
net.ipv4.icmp_echo_ignore_all=1

# Log Martians
net.ipv4.conf.all.log_martians=1
net.ipv4.icmp_ignore_bogus_error_responses=1

# Disable source packet routing
net.ipv4.conf.all.accept_source_route=0
net.ipv4.conf.default.accept_source_route=0

net.ipv6.conf.all.accept_source_route=0
net.ipv6.conf.default.accept_source_route=0

# Block SYN attacks
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_max_syn_backlog=2048
net.ipv4.tcp_synack_retries=2
net.ipv4.tcp_syn_retries=4 # Try values 1-5


# Ignore ICMP redirects
net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.default.send_redirects=0
net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.default.accept_redirects=0
net.ipv4.conf.all.secure_redirects=0
net.ipv4.conf.default.secure_redirects=0

net.ipv6.conf.all.send_redirects=0 # ignore ?
net.ipv6.conf.default.send_redirects=0 # ignore ?
net.ipv6.conf.all.accept_redirects=0
net.ipv6.conf.default.accept_redirects=0
net.ipv6.conf.all.secure_redirects=0 # ignore ?
net.ipv6.conf.default.secure_redirects=0 # ignore ?

# Note disabling ipv6 means you dont need the majority of the ipv6 settings
# General options
net.ipv6.conf.default.router_solicitations=0
net.ipv6.conf.default.accept_ra_rtr_pref=0
net.ipv6.conf.default.accept_ra_pinfo=0
net.ipv6.conf.default.accept_ra_defrtr=0
net.ipv6.conf.default.autoconf=0
net.ipv6.conf.default.dad_transmits=0
net.ipv6.conf.default.max_addresses=1
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

These can all be found in the file `/etc/sysctl.conf`

Another way to change these to 0 is to use the command

```
sudo sysctl -w <item>=<value>
```

An example of setting these will be

```
# IP Spoofing protection
sudo sysctl -w net.ipv4.conf.all.rp_filter=1
sudo sysctl -w net.ipv4.conf.default.rp_filter=1
```

To reload `sysctl` with the latest changes, enter:

```
# Idk difference try both lol
sudo sysctl --system
sudo sysctl -p
```

Monitoring connections

These commands list all connections incoming and outgoing on the local ports.

```
sudo lsof -i -n -P | column -t

sudo netstat -tulnp | column -t
sudo netstat -pelnut | column -t
sudo netstat -nlpa | column -t

cat /etc/services

sudo ss -l | column -t
sudo ss -nutlpw | column -t
```

The different variations of netstat may be useful in their own aspects

Using `sudo` means that the processes and users may also show up.

The command `column -t` prettifies the input into columns

Have a look at the `/proc` directory!

Nmap

You can install `nmap` to do an active enumeration of all the ports. Make sure to delete afterwards as it may score points on the image

Installing

```
sudo apt install nmap
```

Recon

To run scripts and version detection you can do

```
nmap -T4 -A -sC -sV localhost
```

UDP scan

```
nmap -sU localhost
```

Sys admins recommended scans

Scan your system for open ports with :

```
nmap -v -sN localhost
```

Flags include:

```
-p- : full port scan
-sU : UDP scan
-sC : Script scan
-sV : Version enumeration scan
-sS : TCP SYN scan
-sT : TCP Connect scan
-A  : Aggressive


-T<integer> : wait time
|
|-> 5 is almost no time, 4 is pretty good
```

Firewall

UFW

First you need to make sure UFW is installed

```
sudo apt install -y ufw
```

These are the rules that I found useful in the competition - the names are self-explanatory

```
ufw disable

ufw default deny incoming
ufw default allow outgoing

ufw logging on
ufw logging high

ufw allow 22  # ssh
ufw allow 80  # http
ufw allow 443 # https

ufw deny 23   # telnet
ufw deny 2049 # NFS
ufw deny 515  # printer port
ufw deny 111  # Sun rpc / NFS

ufw enable
ufw status numbered
```

Disable IPV6

Edit `/etc/default/ufw`

```
IPV6=no
```

Add line/Edit `/etc/modprobe.d/blacklist`

```
blacklist ipv6
```

Iptables

```
# Flush/Delete firewall rules (run ufw next)
iptables -F
iptables -X
iptables -Z

# Block null packets (DoS)
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

```
# Block syn-flood attacks (DoS)
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

#Drop incoming packets with fragments
iptables -A INPUT -f -j DROP

# Block XMAS packets (DoS)
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP

# Allow internal traffic on the loopback device
iptables -A INPUT -i lo -j ACCEPT

# Allow ssh access
iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT

# Allow established connections
iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow outgoing connections
iptables -P OUTPUT ACCEPT

# Set default deny firewall policy
iptables -P INPUT DROP

#Block Telnet
iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 23 -j DROP

#Block NFS
iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 2049 -j DROP

#Block X-Windows
iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 6000:6009 -j DROP

#Block X-Windows font server
iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 7100 -j DROP

#Block printer port
iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 515 -j DROP

#Block Sun rpc/NFS
iptables -A INPUT -p udp -s 0/0 -d 0/0 --dport 111 -j DROP

#Deny outside packets from internet which claim to be from your loopback
interface.
iptables -A INPUT -p all -s localhost  -i eth0 -j DROP
```

Save rules

```
# Save rules
iptables-save > /etc/iptables/rules.v4
```

Hosts

The hosts file needs to be checked. You can find it at `/etc/hosts`

```
cat /etc/hosts
```

A default hosts file should look similar to:

```
127.0.0.1 localhost
127.0.1.1 $USER
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

You can put it as hosts by doing the following (make sure to back hosts up before had)

```
cp /etc/hosts /etc/hosts.bak

/etc/hosts

127.0.0.1 localhost
127.0.1.1 $USER
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

The `/etc/hosts.allow` file specifies which IP addresses **are** permitted to connect to the host. It is intended to be used in conjunction with the `/etc/hosts.deny` file.

```
cat /etc/hosts.allow
```

To fix

```
echo "ALL: <net>/<mask>, <net>/<mask>, ..." > /etc/hosts.allow
```

The `/etc/hosts.deny` file specifies which IP addresses are **not** permitted to connect to the host.

```
cat /etc/hosts.deny
```

To fix

```
echo "ALL: ALL" >> /etc/hosts.deny
```

Prevent IP Spoofing

Add or edit the following lines in `/etc/host.conf`

```
order bind,hosts
nospoof on
```

Resolv.conf

You also want to check `/etc/resolv.conf`

```
nameserver <ip>
search localdomain

# Change to Google's DNS
nameserver 8.8.8.8
```

System

Sysctl

`/etc/sysctl.conf`

```
kernel.dmesg_restrict=1         # Scored
fs.suid_dumpable=0 # Core dumps # Scored
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.sysrq = 0
kernel.maps_protect=1
kernel.core_uses_pid=1
kernel.shmmax = 68719476736
kernel.shmall = 4294967296

kernel.exec_shield=1 # Or
echo "kernel.exec-shield = 1" > /etc/sysctl.d/50-exec-shield.conf

kernel.panic=10
kernel.kptr_restrict=2
vm.panic_on_oom=1
fs.protected_hardlinks=1
fs.protected_symlinks=1
kernel.randomize_va_space=2         # Scored ASLR; 2 = full; 1 = semi; 0 = none
kernel.unprivileged_userns_clone=0  # Scored
kernel.ctrl-alt-del=0               # Scored  CTRL-ALT-DEL disable
```

To reload `sysctl` with the latest changes, enter:

```
# Idk difference try both lol
sudo sysctl --system
sudo sysctl -p
# Better to write to file /etc/sysctl.conf
```

Disable CTRL-ALT-DEL

You don't want to reboot your server if you accidentally hit the CTRL+ALT+DELETE key combo.

To disable the reboot action, edit the control-alt-delete.conf file:

```
sudo vim /etc/init/control-alt-delete.conf
```

and delete or comment out the following line:

```
exec shutdown -r now "Control-Alt-Delete pressed"
```

Also can try

```
systemctl mask ctrl-alt-del.target
systemctl daemon-reload
```

File permissions

Set these file permissions correctly (Don't just do the ones that are scored if you have time)

```
chattr -i /etc/passwd          # Scored
chattr -i /etc/group           # Scored
chattr -i /etc/shadow          # Scored
chattr -i /etc/ssh/sshd_config # Scored
chattr -i /etc/lightdm/lightdm.conf
chattr -i /etc/profile
chattr -i /etc/bash.bashrc
chattr -i /etc/login.defs
chattr -i /etc/pam.d/common-auth
chattr -i /etc/pam.d/common-password
chattr -i /etc/host.conf
chattr -i /etc/hosts.deny
chattr -i /etc/hosts.allow
chattr -i /etc/hosts
chattr -i /etc/resolv.conf
chattr -i /etc/default/grub         # Scored
chattr -i /etc/grub.d/40_custom     # Scored
chattr -i /etc/ers
chattr -i ~/.mozilla/firefox/*.default/prefs.js
chattr -i /etc/sysctl.conf
chattr -i /etc/apt/sources.list
chattr -i /etc/lightdm/lightdm.conf.d/50-myconfig.conf
```

```
chown root:root /etc/fstab   # Scored
chmod 644 /etc/fstab         # Scored
chown root:root /etc/group   # Scored
chmod 644 /etc/group         # Scored
chown root:root /etc/shadow  # Scored
chmod 400 /etc/shadow        # Scored
chown root:root /etc/apache2 # Scored
chmod 755 /etc/apache2       # Scored

chmod 0600 /etc/securetty
chmod 644 /etc/crontab
chmod 640 /etc/ftpusers
chmod 440 /etc/inetd.conf
chmod 440 /etc/xinetd.conf
chmod 400 /etc/inetd.d
chmod 644 /etc/hosts.allow
chmod 440 /etc/ers
chmod 640 /etc/shadow           # Scored
chmod 600 /boot/grub/grub.cfg   # Scored
chmod 600 /etc/ssh/sshd_config  # Scored
chmod 600 /etc/gshadow-         # Scored
chmod 600 /etc/group-           # Scored
chmod 600 /etc/passwd-          # Scored

chown root:root /etc/ssh/sshd_config # Scored
chown root:root /etc/passwd-         # Scored
chown root:root /etc/group-          # Scored
chown root:root /etc/shadow          # Scored
chown root:root /etc/securetty
chown root:root /boot/grub/grub.cfg  # Scored

chmod og-rwx /boot/grub/grub.cfg    # Scored
chown root:shadow /etc/shadow-
chmod o-rwx,g-rw /etc/shadow-
chown root:shadow /etc/gshadow-
chmod o-rwx,g-rw /etc/gshadow-
touch /etc/cron.allow
touch /etc/at.allow
chmod og-rwx /etc/cron.allow
chmod og-rwx /etc/at.allow
chown root:root /etc/cron.allow
chown root:root /etc/at.allow
chown root:root /etc/cron.d
chmod og-rwx /etc/cron.d
chown root:root /etc/crontab
chmod og-rwx /etc/crontab
chmod -R g-wx,o-rwx /var/log/*
```

Disable compilers

Disable

```
sudo chmod 000 /usr/bin/byacc
sudo chmod 000 /usr/bin/yacc
sudo chmod 000 /usr/bin/bcc
sudo chmod 000 /usr/bin/kgcc
sudo chmod 000 /usr/bin/cc
sudo chmod 000 /usr/bin/gcc
sudo chmod 000 /usr/bin/*c++
sudo chmod 000 /usr/bin/*g++
```

Re-enable

```
sudo chmod 755 /usr/bin/byacc
sudo chmod 755 /usr/bin/yacc
sudo chmod 755 /usr/bin/bcc
sudo chmod 755 /usr/bin/kgcc
sudo chmod 755 /usr/bin/cc
sudo chmod 755 /usr/bin/gcc
sudo chmod 755 /usr/bin/*c++
sudo chmod 755 /usr/bin/*g++
```

Boot settings

You want to ensure that `/boot/grub/grub.cfg` is only owned by root

```
stat /boot/grub/grub.cfg
```

If this isn't `0`

```
chown root:root /boot/grub/grub.cfg
chown og-rwx /boot/grub/grub.cfg
```

Grub Password

Update by running

```
apt-get install grub-common -y
```

```
user@ubuntu:~/Desktop$ grub-mkpasswd-pbkdf2
Enter password: CyberPatriot1!
Reenter password: CyberPatriot1!
PBKDF2 hash of your password is
```

```
grub.pbkdf2.sha512.10000.097F740E017A137E1AAC5C89AA6F388C56B126AB84C958BB5B6685A
2448ED6B456290C2A2DA106C4FC620320A354ED87781FA6CB6E535E865274D450A71DF586.EF9C62
EBF740690628E70E9FEE5942DF26FB5496F974E546B2F1A245015A23F060C80FF332FD98F3C1E5D5
0F5E045F8A74238B49348A762C5FD9AEF4B7020F8F
```

This is important.

Now open and add the following lines to the end of the file `/etc/grub.d/40_custom`

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.

# New stuff

set superusers="<username>" # set this to root
password_pbkdf2 <username> <long sha512 string>
~
```

Setting passwords

```
password_pbkdf2 <username> <long sha512 string>

# or

password <username> insecure_plain_text_password
```

```
sudo update-grub
```

Setting / Creating other super-users

Edit `/etc/grub.d/10_linux`

Change the line

```
printf “menuentry ‘${title}’
```

to

```
printf “menuentry –users <username> ‘${title}”
```

Process Limit

You might need to protect your system against fork bomb attacks. A simple way to prevent this is by setting up processes limit for your users. All the limits can be configured in the `/etc/security/limits.conf` file.

```
sudo vim /etc/security/limits.conf
```

This file comes with all the help you need. Here's an example:

```
user1 hard nproc 100
@group1 hard nproc 20
```

This will prevent users from a specific group from having a maximum of 20 processes and maximize the number of processes to 100 to user1.

Miscellanous

Secure kernel?

```
echo "* hard core 0" >> /etc/security/limits.conf
```

Harden /proc with hidepid

```
mount -o remount,rw,hidepid=2 /proc
```

Secured shared memory (scored)

Back to normal (Recommended in comp?)

```
echo "tmpfs /run/shm tmpfs rw,noexec,nosuid,nodev 0 0" >> /etc/fstab
```

Read-only (more secure)

```
echo "tmpfs /run/shm tmpfs defaults,ro,noexec,nosuid 0 0" >> /etc/fstab
```

If GUI Doesnt boot

Check run levels

```
runlevel
```

Different run levels

```
0 - System halt; no activity
1 - Single User
2 - Multi-user, no filesystem
3 - Multi-user, commandline only
4 - User definable
5 - Multi-users, GUI
6 - Reboot
```

Change run level

```
telinit <level>
```

Malware

Backdoor enumeration

Processes

```
# As root
ps aux | grep nc
ps aux | grep netcat
ps aux | grep python

# Especially services running as root
```

Look at cronjobs, auto-run scripts

```
sudo su
crontab -l
```

Finding SUID files and SGID files

```
# SUID
find / -perm -4000 -print
# SGID
find / -perm -2000 -print
```

You can also include a `2> /dev/null` at the end to suppress errors

Php backdoors

Look in web server files i.e index.php for code

```
grep'((eval.*(base64_decode|gzinflate|\$_))|\$[0O]
{4,}|FilesMan|JGF1dGhfc|IIIl|die\(PHP_OS|posix_getpwuid|Array\
(base64_decode|document\.write\("\\u00|sh(3(ll|11)))' <path> -lroE --
include=*.php*
```

Anti-malware software

Installing

These are all the ones you'll need

```
sudo apt install -y chkrootkit clamav rkhunter apparmor apparmor-profiles
```

And for Lynis:

```
sudo wget https://downloads.cisofy.com/lynis/lynis-2.7.0.tar.gz -O
~/Desktop/lynis.tar.gz
sudo tar -xzf ~/Desktop/lynis.tar.gz --directory /usr/share
```

Running

Chrootkit

```
chkrootkit -q
```

Rkhunter

```
rkhunter --update
```

Only run this command once:

```
rkhunter --propupd
```

Then finally

```
rkhunter -c --enable all --disable none
```

Lynis

```
/usr/share/lynis/lynis update info
/usr/share/lynis/lynis audit system
```

ClamAV

```
systemctl stop clamav-freshclam
freshclam --stdout
systemctl start clamav-freshclam

clamscan -r -i --stdout --exclude-dir="^/sys"
```

Access Control

AppArmor

Run

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

aa-enforce /etc/apparmor.d/usr.bin.Firefox # Firefox specific
```

SELinux

Installing

Make sure to stop `apparmor` (You may need to remove but make sure to replace it)

```
sudo /etc/init.d/apparmor stop

### DONT NEED TO DO THIS UNLESS NECESSARY
apt purge apparmor
```

Update and reboot

```
apt update && apt upgrade -yuf
reboot
```

Install SELinux

```
apt install selinux
reboot
```

Check it is working

```
setenforce 1

#Check it is on now
getenforce

#Should respond with
> Enforcing
```

Make SELinux persistent

Edit the file `/etc/selinux/config` file to set SELINUX parameter:

```
SELINUX=enforcing
```

The following code does this automatically

```
sed -i "/^SELINUX=.*/ c\SELINUX=enforcing" /etc/selinux/config
```

Ensuring it's not disabled in bootloader config

```
grep "^\s*linux" /boot/grub/grub.cfg
```

Verify that no Linux line has `selinux=0` or `enforcing=0`

To fix:

Edit `/etc/default/grub` and remove all instances of `selinux=0` and `enforcing=0` from `CMDLINE_LINUX` parameters:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""
```

Then run the following to update `grub2` config

```
update-grub
```

Urls / General Info

https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh

https://github.com/DominicBreuker/pspy

http://bookofzeus.com/harden-ubuntu/hardening/ssh/

http://bookofzeus.com/harden-ubuntu/hardening/apache/

https://mirrors.ustc.edu.cn/repogen/

http://www.securityandit.com/security/securing-bind-dns-server/

https://github.com/decalage2/awesome-security-hardening

Different run levels

```
0 - System halt; no activity
1 - Single User
2 - Multi-user, no filesystem
3 - Multi-user, commandline only
4 - User definable
5 - Multi-users, GUI
6 - Reboot
```

Vulnerability Categories

Type	Description
Account Policies	Password Policy, Lockout Policy, etc.
Application Security Settings	Critical Service Settings, Required Application Settings, Application Permission, etc.
Application Updates	Application Updates, Application Automatic Update Settings, etc.
Defensive Countermeasures	Firewall, Anti-virus, Encryption, etc.
Forensics Questions	Forensics Questions
Local Policies	Audit Policy, User Rights Assignment, Security Options -- Security Options include: Network Security Options and Privilege Elevation Authorization, etc
Operating System Updates	Windows Updates, Service Packs, Windows Automatic Update Settings, etc.
Policy Violation: Malware	Backdoors, Remote Administration Tools, Keyloggers, Password Sniffers, etc.
Policy Violation: Prohibited Files	Media Files, Software Archives, Confidential Information, etc.
Policy Violation: Unwanted Software	Games, Servers, Scareware, Adware, PUP, "Hacking" Tools, etc.
Service Auditing	Enable and Disable Services, etc.
Uncategorized Operating System Settings	Remote Access, File Sharing, Screen Locking, Group Policy Settings, Operating System Permissions, etc.
User Auditing	Authorized Users, Groups, and other settings unique to users, etc.

Ubuntu 16

- Account Policies – 4
- Application Security Settings – 5
- Application Updates – 2
- Defensive Countermeasures – 1
- Forensics Questions – 3
- Local Policies – 5
- Operating System Updates – 4
- Policy Violation: Malware – 1
- Policy Violation: Prohibited Files – 1
- Policy Violation: Unwanted Software – 2
- Service Auditing – 2

- Uncategorized Operating System Settings – 1
- User Auditing – 5

Debian 8

- Account Policies – 5
- Application Security Settings – 6
- Application Updates – 2
- Defensive Countermeasures – 1
- Forensics Questions – 3
- Local Policies – 5
- Operating System Updates – 3
- Policy Violation: Malware – 1
- Policy Violation: Prohibited Files – 1
- Policy Violation: Unwanted Software – 4
- Service Auditing – 3
- Uncategorized Operating System Settings – 1
- User Auditing – 5

Extra cautious Check List

1. Forensic Question
2. **Removed unauthorized user [user]**
3. **Removed hidden user**
4. **User [user] is not an administrator**
5. **Changed insecure password for [user]**
6. **Created user group [group]**
7. **Created user account [user]**
8. **Users added to group [group]**
9. **Guest account is disabled**
10. **User cannot login without a password**
11. **[User] password expires**
12. **Root password is no longer blank / Changed insecure Root password**
13. **Password for user is hashed with a secure algorithm**
14. **man - Does not have a login shell**
15. **A default maximum password age is set**
16. **Previous passwords are remembered**
17. **A minimum password length is required**
18. **Extra dictionary based password strength checks**
19. **A secure password hashing algorithm is used**
20. **PASS_MAX_DAYS corrected in login.defs**
21. **Null passwords do not authenticate**
22. **An account lockout policy is configured**
23. **Greeter does not enumerate user accounts**

24. **Ignore broadcast ICMP echo requests enabled**
25. **IPv4 TCP SYN cookies have been enabled**
26. **IPv4 TCP SYN,ACK retries reduced**
27. **IPv4 forwarding has been disabled**
28. **IPv4 sending ICMP redirects disabled**
29. **ASLR is enabled**
30. **/proc/sys/net/ipv4/tcp_rfc1337 - contains 1**
31. **/proc/sys/kernel/unprivileged_userns_clone - contains 0**
32. **Resolver checks for IP spoofing**
33. **Dmesg restrict set to 1**
34. **Sysrq is set to 0(disable sysrq)**
35. **Xserver-allow-tcp disabled**
36. **Sudo requires authentication**
37. **Insecure permissions on shadow file fixed**
38. **GRUB configuration is not world readable**
39. **boot/grub/grub.cfg - contains set superusers= and password_pbkdf2 grub**
40. **Stricter defaults have been enabled for shared memory**
41. **Firewall protection has been enabled**
42. **Minetest service has been disabled or removed**
43. **OpenArena service has been disabled or removed**
44. **DNS service is disabled or removed**
45. **IRC daemon has been stopped and disabled**
46. **Postgresql has been disabled or removed**
47. **Apache2 service has been disabled or removed**
48. **FTP service has been disabled or removed**
49. **SNMP service has been disabled/removed**
50. **Samba service has been disabled/removed**
51. **Rsync service has been disabled/removed**
52. **bind9 service is stopped and removed**
53. **nfs services is stopped and removed**
54. **The system automatically checks for updates daily**
55. **Install updates from important security updates**
56. **The Linux Kernel has been updated**
57. **Bash has been updated**
58. **OpenSSL shared libraries have been updated**
59. **Glibc has been updated**
60. **sources.list has valid lists**
61. **Firefox has been updated**
62. **Samba has been updated**

63. **Apache2 has been updated**
64. **PHP5 has been updated**
65. **WordPress has been updated**
66. **OpenSSH has been updated**
67. **7zip has been updated**
68. **LibreOffice has been updated**
69. **Pure FTP has been updated**
70. **Prohibited MP3 files are removed**
71. **Removed plaintext file containing passwords**
72. **Stellarium has been installed**
73. **Removed {*Prohibited software below*}**
    - **Minetest**
    - **NBTScan**
    - **Nmap, Zenmap**
    - **Wireshark, Tshark**
    - **TCPSpray**
    - **Ettercap**
    - **Dsniff**
    - **p0f**
    - **freeciv**
    - **ophcrack**
    - **Kodi**
    - **uTorrent**
    - **knocker**
    - **kismet**
74. **Removed python backdoor**
75. **Removed perl backdoor**
76. **Removed netcat backdoor**
77. **php backdoor is removed**
78. **SUID backdoor removed**
79. **SSH root login has been disabled**
80. **SSH protocol 1 has been disabled**
81. **SSH only listens on port 222(situational)**
82. **SSH allows only public key authentication**
83. **SSH does not permit empty passwords**
84. **SSH user environment processing is disabled**
85. **Anonymous Samba access is disabled**
86. **Unauthorized Samba share is disabled**
87. **Samba blank passwords are disabled**
88. **Samba SMB1 protocol is disabled**
89. **SMB share is not world writeable**
90. **Encrypt smb traffic is enabled**
91. **ntlm authentication is disabled**

92. FTP local users must log in as anonymous
93. FTP anonymous write commands are disabled
94. FTP PASV security checks enabled
95. FTP anonymous user is not root
96. FTP anonymous access is disabled
97. Insecure permissions on FTP root directory fixed
98. **FTP plain-text authentication disabled**
99. **MySQL remote access is disabled**
100. **SQL is not ran as root**
101. **Removed phpinfo() php file**
102. **PHP expose is Off**
103. **PHP system function is disabled**
104. **php allow url fopen is off**
105. **php session.use strict mode is on**
106. **Apache server signature is disabled**
107. **Apache trace requests disabled**
108. **Apache fileEtags none**
109. **Firefox displays warning on known malware sites**
110. **Firefox warns when sites try to install add-ons**
111. **Firefox - Block dangerous and deceptive content**

Ports and Protocols

Protocol	TCP/UDP	PORT
FTP	TCP	20/21
SSH	TCP	22
Telnet	TCP	23
SMTP	TCP	25
DNS	BOTH	53
DHCP	UDP	67/68
TFTP	UDP	69
HTTP	TCP	80
POP3	TCP	110
NTP	UDP	123
NetBIOS	BOTH	137-139
IMAP	TCP	143
SNMP	BOTH	161/162
BGP	TCP	179
LDAP	BOTH	389
HTTPS	TCP	443
SMB	TCP	445
LDAPS	BOTH	636
SFTP	TCP	989/990

Checklist

Topic	Task	Done
README	Read README x3 Times	
3rd Party	LinEnum.sh -t	
	pspy	
	Webmin	
General	Forensics Questions	
	Remove media files	
	Firefox hardening	
	Check Logs	
	LibreOffice - macros	
Users	Removed extra users	
	Removed sudoers	
	Check /etc/sudoers (visudo) file	
	Changed **all** passwords	
	Added any README users	
	Check passwd file (repeating UID, UID=0)	
	Check shadow file	
	Check group file	
	Remove guest users	
User Policies	PAM	
	Users, roots umask	
	Disable su login as root	
	Password policies	
	Account Policy	
Apt	Check / Replace sources.list	
	Update	
	Upgrade	
	Auto updates	
	Enumerate installed packages	
	Remove "Hacking tools" / Games	
	Shellshock test	

Topic	Task	Done
Services	Secure **SSH**	
	[*] Config	
	[*] File Permissions	
	Secure **FTP**	
	[*] Config for specific type	
	Secure **Mysql**	
	[*] Config	
	S	
	Secure **Samba**	
	[*] Config	
	Secure **Nginx** / **Apache2**	
	[*] Config	
	[*] File permissions	
	[*] Disable / Enable modules	
	Secure **PHP**	
	[*] Config	
	[*] Security Config	
	Secure **DNS**	
	Disable / uninstall **ALL** unnecessary services	
	No services running as root	
	Enumerate Cronjobs	
	Enumerate boot-up scripts	
Networking	Harden network configuration policies	
	Enumerate open ports	
	Nmap localhost	
	Firewall (UFW + IPTABLES)	
	Checked hosts	
	Changed nameserver	
	Prevent IP Spoofing	
System	**Sysctl** config	

Topic	**Task**	**Done**
	Disable CTRL-ALT-DEL	
	Harden permissions	
	Disable compilers	
	Check boot settings	
	Grub password	
	Process Limit	
	Disable compilers	
	Miscellaneous	
Malware	Look for backdoors	
	Install anti-malware tools	
	Run malware tools x3	
Access Control	AppArmor	
To install	**INSTALL REQUIRED / CRITICAL SERVICES**	
	fail2ban	
	bum	
	sysv-rc-conf	
	auditd	
	rsyslog	
	psad	
	aide	
	tcpd	
	mod_security	
	ranger	
	unhide	