CyberPatriot Ubuntu Checklist

**TAKE SCREENSHOTS OFTEN!!!!!!**

**READ THE README BEFORE STARTING!!!!**

**BEFORE STARTING, EDIT THE SCRIPT TO MEET README GUIDELINES!!!!!! - eg. if readme says NGINX is a critical service, make sure the script doesn’t delete NGINX - also the firefox section MUST be edited before running**

**~~Script may break your password (can’t use sudo), temporary workaround is to switch to another admin user and use their account for sudo~~**

**Do NOT use your account in the script!!! Doing so will break your login!!!**

# **>Forensics Questions**

**DO THESE FIRST!!!!!!!!!!!!!!!!!!!!!!!**

**DO THESE FIRST!!!!!!!!!!!!!!!!!!!!!!!**

**DO THESE FIRST!!!!!!!!!!!!!!!!!!!!!!!**

**DO THESE FIRST!!!!!!!!!!!!!!!!!!!!!!!**

**DO THESE FIRST!!!!!!!!!!!!!!!!!!!!!!!**

**FOR CHANGING SYSTEM FILES, USE SUDO GEDIT OR SUDO NANO**

**Anything that has a $ is a placeholder, replace that term with the appropriate name**

**Also, try to make a copy of any files you are editing for when you screw up (cp $filepath $destinationpath)**

# **NOTE ON COPY/PASTE**

**Ctrl+C and Ctrl+V do NOT work in the terminal. Ctrl+C is an interrupt sequence and will kill whatever process is currently running in the terminal.**

**To copy, use Ctrl+Shift+C**

**To paste, use Ctrl+Shift+V**

**If those don’t work right click in the terminal and select copy/paste**

**Outside of the terminal Ctrl+C and Ctrl+V should work normally.**

# >**Scripts**

For VMWare Workstation Pro

In the virtual machine, go to VM tab -> Removable devices -> CD/DVD (SATA) -> Settings

Under device status, select the boxes for Connected and Connect at power on

Under connection, select Use ISO image file and browse to the script ISO

Copy the Linux folder into the /tmp folder

In a terminal, run cd /tmp/Linux, then use chmod +x $script.sh to grant execute permissions

**BEFORE RUNNING SCRIPT GO IN AND EDIT THE FIREFOX SECTION**

**EDIT SCRIPT TO MATCH README REQUIREMENTS IF NECESSARY**

**THEN TAKE SNAPSHOT TO SAVE CURRENT PROGRESS**

To run the script, type sudo ./$script.sh

# >**Updates**

System settings -> Software and Updates -> Updates -> Select boxes for check daily, important security updates, recommended updates, display immediately (for security updates and for other updates) -> choose for long-term support versions for Notify me of a new Ubuntu version -> DON’T click revert -> close

/etc/apt/sources.list - make sure it has http://security.ubuntu.com/ubuntu/

**Start updates early on (but do the steps above this beforehand and verify that the software update sources are correct)**

Make sure services that need to be upgraded are upgraded

sudo apt-get update

sudo apt-get dist-upgrade -y

sudo apt-get install -f -y

sudo apt-get autoremove -y

sudo apt-get autoclean -y

sudo apt-get check

Upgrade kernel - sudo apt-get update && sudo apt-get install linux-image-generic and sudo apt-get update && sudo apt-get install linux-headers-generic

Upgrade bash if it is vulnerable to Shellshock

Check by running this command in Bash (terminal):

env 'VAR=() { :;}; echo Bash is vulnerable!' 'FUNCTION()=() { :;}; echo Bash is vulnerable!' bash -c "echo Bash Test"

If output contains Bash is vulnerable! (injected payload) then Bash needs to be updated

Output looks like this:

Bash is vulnerable!

Bash Test

Also, if there are bash warnings or errors in the output, update Bash to the latest version

If all you see in the output is: Bash Test then Bash is safe from Shellshock

Update Bash with: sudo apt-get update && sudo apt-get install --only-upgrade bash

Check again with: env 'VAR=() { :;}; echo Bash is vulnerable!' 'FUNCTION()=() { :;}; echo Bash is vulnerable!' bash -c "echo Bash Test"

# >**Remove aliases**

View with alias

Remove one with unalias $aliasname

Remove all with unalias -a

Some aliases that may be useful:

alias ls='ls --color=auto'

alias grep='grep --color=auto'

# >**Check user passwords, account type, and if they are on the approved user list**

System Settings -> User Accounts

Delete unauthorized users

Make sure only administrators are given administrative rights

Check directory /etc/sudoers.d and make sure only members of group sudo can sudo (remove unauthorized files from directory and fix insecure configurations)

Remove !authenticate and NOPASSWD from /etc/sudoers and files in /etc/sudoers.d

Check /etc/group and remove non-admins from sudo and admin groups (or do cat /etc/group | grep sudo)

Open up /etc/passwd (cat /etc/passwd) and check which users:

* Are UID 0 (remove if not root)
* Can login
* Are allowed in the readme
* Check UID and GID ($username:x:UID:GID) of users (can compare with /etc/group) make sure none are UID 0, have unique UIDs
* Check bash shell of users (something like /bin/bash at the end of each user’s line)
* Change owner of /etc/passwd (not root? Check permissions)
* Perhaps compare with clean copy of file to verify users

Open up /etc/shadow and check if any users

* Are locked ($username:!$passwordhash:$morestuff)
* Have never set a password ($username:\*:$morestuff)
* Can login with empty password ($username::$morestuff)

Add users from terminal: sudo adduser $username

Delete users from terminal: sudo deluser $username

Optionally, lock users with: sudo usermod -L $username

List all groups with sudo cat /etc/group

Add a group with addgroup $groupname

Add user to group with adduser $username $groupname

Check UID with sudo id -u $username

* UID 0 is root account
* UID 65534 commonly reserved for *nobody* (user with no system privileges), often used for individuals remotely accessing the system via FTP or HTTP
* UID for system accounts/privileged users typically are from 1 to whatever number the non-privileged users start at
* UID for non-privileged users vary between distributions, but is set in /etc/login.defs with the lines UID\_MIN and UID\_MAX

/etc/shadow (password file, not sure what to do with it but seen it on a few lists) - make sure owned by root, permissions should be 644

# >**Check critical services**

Based on README, google “How to harden *servicename*” (ex. how to harden ssh) and follow recommended steps - a few listed near the end of this document

# >**Secure SSH**

Set PermitRootLogin no in /etc/ssh/sshd\_config

Also set PermitEmptyPasswords no and HostbasedAuthentication no

Also change port to nonstandard port (eg. Port 3784)

sudo service ssh restart

Check user home directories for files that allows for authenticated SSH access (ex. /home/$username/.ssh/authorized\_keys) - remove or rename .ssh/ to prevent further SSH authentication capabilities (another suggestion is to remove entries in the authorized\_keys file instead, if present)

Check for established SSH connections:

* who | grep $username
* sudo pkill -f pts/#

If need to restrict SSH access, create group “sshlogin” (or whatever) and add group name to the AllowGroups variable in /etc/ssh/sshd\_config (ex. AllowGroups sshlogin)

Then add permitted SSH users to group “sshlogin” and restart SSH service

* sudo adduser username sshlogin
* sudo systemctl restart sshd.service (or use above command on line 3 of >Secure SSH)

Install SSH Google Authenticator (2FA) if needed

* sudo apt-get install libpam-google-authenticator

Install OpenSSH Server if needed

* sudo apt-get install openssh-server

# >**Disable Guest account**

Go to /etc/lightdm/lightdm.conf and add the line allow-guest=false

Also add greeter-hide-users=true

Restart with sudo restart lightdm

Other lightdm conf files:

/usr/share/lightdm/lightdm.conf.d/\*.conf

/etc/lightdm/lightdm.conf.d/\*.conf

/etc/lightdm/lightdm.conf

# >**Disable root account**

sudo passwd (gives root a password, thus enabling it)

sudo passwd -l root (locks root)

OR sudo usermod -L root

sudo gedit /etc/passwd then change first line (root) to root:x:0:0:root:/root:/sbin/nologin

# >**Use sudo instead of root**

sudo visudo (there should be no “NOPASSWD”)

$username ALL=(ALL) ALL

sudo adduser $username sudo

Check defaults (remove !authenticate and NOPASSWD)

# >**Password policy**

***Edit PAM files last, as it can screw with the machine!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!***

Open /etc/login.defs and change these values

PASS\_MAX\_DAYS 60

PASS\_MIN\_DAYS 1

PASS\_WARN\_AGE 7

(alternately: max 30, min 3, warn 7)

Safe password: H=Fmcqz3M]}&rfC$F>b)

Open /etc/pam.d/common-password

In the line with pam\_unix.so, add minlen=16 and remember=5 to the end of the line

In the line with pam\_cracklib.so, change the line to: password requisite pam\_cracklib.so try\_first\_pass retry=3 difok=4 minlen=16 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1 maxrepeat=2 reject\_username gecoscheck enforce\_for\_root

(If you cannot find pam\_cracklib.so, install it with sudo apt-get install libpam-cracklib)

Alternatively, run: sudo sed -i '1s/^/password requisite pam\_cracklib.so try\_first\_pass retry=3 difok=4 minlen=16 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1 maxrepeat=2 reject\_username gecoscheck enforce\_for\_root\n/' /etc/pam.d/common-password

Open /etc/pam.d/common-auth

Add the line: auth required pam\_tally2.so onerr=fail deny=3 unlock\_time=1800 audit even\_deny\_root\_account silent after the line with pam\_deny.so but before pam\_permit.so; also change the requisite in the line with pam\_deny.so to required

Alternatively, run: sudo sed -i '1 s/^/auth optional pam\_tally.so deny=3 unlock\_time=1800 onerr=fail audit even\_deny\_root\_account silent\n/' /etc/pam.d/common-auth

Add/change line below [pam]: timestamp\_timeout = 86400 in /etc/pam.d/common-auth or /etc/pam.d/common-session

/etc/security/pwquality.conf can set the same values as for pam\_cracklib.so

Also set dictcheck=1

/etc/pam.d/passwd ensure there is the line @include common-password

Use chpasswd to change all passwords to match policy (enter with format $username:$password, with each user on a new line)

Expiration

* sudo chage $username
* sudo chage -l $username (view current status)

# >**Remove prohibited files**

cd /home then sudo ls -R \* then sudo ls -Ra \* | less, look through directories for media files/tools and/or “hacking tools”

List hidden files in a directory with ls -a | grep -e "^\."

ls -a works great for searching within one directory, while ls -la provides more detailed info

Check for hidden messages with filename of “...” with sudo ls -Ra / | grep "\.\.\." (check places like /home, /etc/init.d, /etc/lightdm, /etc/ufw/applications.d, /var/spool/cron/crontabs, etc. but GOOD LUCK finding them since output doesn’t tell you where they are, HINT replace or remove / to search specific directories and go one by one across one layer of the file system to narrow down the locations)

Check with: sudo find / -name ‘\*.shost’ and sudo find / -name ‘\*.rhost’ and remove any found

apt list --installed | less

Media files: .mp3, .mp4, .jpg, .mov, .txt (passwords plain text), .png, .gif

Hacking tools: Zenmap, nmap, Wireshark, Crack, John the Ripper\*, Hydra\*, telnetd, Ophcrack, Medusa, nikto\*, netcat, logkeys (\* Disable Remote Desktop), ettercap, yersinia

Other software: Minetest, Freeciv, Aisleriot, Wesnoth, Nginx, endless sky, ManaPlus, Game Conqueror, deluge

Applications -> Ubuntu Software Center -> Installed Software -> find software and click remove

If removing from terminal use:

* dpkg --list | less (find name of application’s package, using grep instead can also work)
* sudo apt-get --purge remove $package (if apt-get doesn’t work, try sudo aptitude remove $package)

Record file names and locations that were deleted

Script for finding prohibited files, must paste into bash or sh file to run

for suffix in mp3 txt wav wma aac mp4 mov avi gif jpg png bmp img exe msi bat sh

do

sudo find /home -name \*.$suffix

done

List of base software (Ubuntu 16, may still get points for removing some of these)

* Activity Log Manager
* Additional Drivers
* AisleRiot Solitaire
* Archive Manager
* Backups
* Browser
* Calendar
* Character Map
* Checkbox
* Cheese
* Desktop Sharing
* Firefox Web Browser
* GNOME System Monitor
* IBus Preferences
* ImageMagick (display Q16)
* Input Method
* Language Support
* LibreOffice Calc, Draw, Impress, Math, and Writer
* Mahjongg
* Mines
* Onboard
* Online Accounts
* Passwords and Keys
* Personal File Sharing
* Power Statistics
* Printers
* Remmina
* Rhythmbox
* Screen Reader
* Shotwell
* Simple Scan
* Software Updater
* Startup Applications
* Startup Disk Creator
* Sudoku
* System Log
* Terminal
* Thunderbird Mail
* Transmission
* Universal Access
* Vim
* xdiagnose
* XTerm
* Calculator
* Disk Usage Analyzer
* Document Viewer
* Files
* Font Viewer
* gedit
* GNOME Disks
* Help
* Image Viewer
* Screenshot
* Software
* Videos

# >**Check firewall settings**

Enable: sudo ufw enable

Enable logging: sudo ufw logging high or sudo ufw logging on or sudo ufw logging low

Alternatively, use Firestarter for GUI (sudo apt-get install firestarter, go to preferences)

Check firewall rules for unauthorized inbound rules: (ex. Don’t allow telnet through firewall), block all inbound connections except on ports that need to remain listening

sudo ufw status numbered

sudo ufw delete $number

To specify rules use:

* ufw allow $port (optionally add /$protocol to the port)
* Ex. ufw allow 53 OR ufw allow 25/tcp
* Can specify the direction of traffic: ufw allow in http OR ufw reject out smtp

<https://wiki.ubuntu.com/UncomplicatedFirewall?action=show&redirect=UbuntuFirewall>

Iptables - backbone of ufw, can configure this way too but more complicated

# >**Disable services that aren’t critical**

Check with service --status-all MUST READ README

service --status-all | grep "+" (shows programs with a return code of 0 (C/C++ users will understand), which is non-native programs)

sudo service $servicename stop (use start if enabling a service)

sudo apt-get purge $servicename

Quickly look up packages with dpkg --list | grep $packagename

sshd

telnet and telnetd

vsftpd - FTP server

Remote Desktop - rexec, rlogin, rsh, rexecd, rlogind, rshd, xrdp, vnc

FTP

Anonymous FTP

Fingerd and Finger

tftpd

snmp - simple network management protocol

pop3 - email server

icmp - ping

sendmail - email server

dovecot - email server

rstatd

talk, ntalk

apache2

kdump

WorldForge

samba

postgresql

PHP

MySQL

WordPress

proftpd

List of base services in Ubuntu 16 (+ is running, - is off)

* [+] acpid
* [-] alsa-utils
* [-] anacron
* [+] apparmor
* [+] apport
* [+] avahi-daemon
* [-] bluetooth
* [-] bootmisc.sh
* [-] brltty
* [-] checkfs.sh
* [-] checkroot-bootclean.sh
* [-] checkroot.sh
* [+] console-setup
* [+] cron
* [+] cups
* [+] cups-browsed
* [+] dbus
* [-] dns-clean
* [+] grub-common
* [-] hostname.sh
* [-] hwclock.sh
* [+] irqbalance
* [-] kerneloops
* [+] keyboard-setup
* [-] killprocs
* [+] kmod
* [+] lightdm
* [-] lvm2
* [+] lvm2-lvmetad
* [+] lvm2-lvmpolld
* [-] mountall-bootclean.sh
* [-] mountall.sh
* [-] mountdevsubfs.sh
* [-] mountnfs-bootclean.sh
* [-] mountnfs.sh
* [+] network-manager
* [+] networking
* [+] ondemand
* [-] plymouth
* [-] plymouth-log
* [-] pppd-dns
* [+] procps
* [+] rc.local
* [+] resolvconf
* [-] rsync
* [-] rsyslog
* [+] saned
* [-] sendsigs
* [+] speech-dispatcher
* [-] thermald
* [+] udev
* [+] ufw
* [-] umountfs
* [-] umountnfs.sh
* [-] umountroot
* [+] unattended-upgrades
* [+] urandom
* [-] uuidd
* [+] whoopsie
* [-] x11-common

If system is configured with inetd, open /etc/inetd.conf and comment out lines with #, then restart inetd service or reboot

If using xinetd, configuration file is in /etc/xinetd.d

Each file in the directory is a service, add disable = yes to any you want to disable

IF INETD IS NOT NEEDED REMOVE FROM SYSTEM

Alternatively, view and manage with GUI: apt-get install bum

Run with sudo bum

To start a service, right-click it and select “start”

To enable a service, check the box next to it

Started services will have a lit light bulb next to it, while stopped services will have a darkened light bulb

If README says no openssh-server or ftp, run:

sudo apt-get -y purge openssh-server\*

sudo apt-get -y purge vsftpd\*

\* Disable Remote Desktop

# >**Check Firefox settings**

Block pop-ups

Turn on automatic updates (if needed)

Set as default browser (if needed)

Warn when sites try to install add-ons (if possible)

Disable Java? (openJDK and Oracle Java)

Install git with sudo apt-get install git

git clone<https://github.com/pyllyukko/user.js>

In the user.js **folder**, copy the user.js **file** to ~/.mozilla/firefox/XXXXXXXX.your\_profile\_name/

(cp user.js ~/.mozilla/firefox/XXXXXXXX.your\_profile\_name/user.js)

Still in the user.js folder, run make systemwide\_user.js and copy that file to /etc/firefox/syspref.js

(sudo cp systemwide\_user.js /etc/firefox/syspref.js)

# >**Secure ports**

sudo netstat -tulpn or sudo ss -tulpn

If local address is 127.0.0.1:$port, it is loopback and isn’t exposed

sudo lsof -i :$port (more info, not always needed)

whereis $program (copy program listening on port)

Copy where the program is (first one if more than one listed): dpkg -S $location (note the name of the package given to you)

sudo apt-get purge $package (if no package, you can probably delete it with rm $location and then killall -9 $program) MAKE SURE IT IS NOT A CRITICAL PACKAGE BEFORE HITTING Y

sudo netstat -tulpn (make sure the port is actually closed)

ALTERNATE:

sudo netstat -tulnp or sudo ss -tulpn

Use sudo kill $PID

To check the listening ports and applications on Linux:

1. Open a terminal application i.e. shell prompt.
2. Run any one of the following command on Linux to see open ports:  
   sudo lsof -i -P -n | grep LISTEN  
   sudo netstat -tulpn | grep LISTEN  
   sudo lsof -i:22 ## see a specific port such as 22 ##  
   sudo nmap -sTU -O IP-address-Here
3. For the latest version of Linux use the ss command. For example, ss -tulw

# >**Secure network**

Enable firewall - sudo ufw enable

Enable syn cookie protection - sysctl -n net.ipv4.tcp\_syncookies

Disable IPv6 (can be harmful) - echo "net.ipv6.conf.all.disable\_ipv6 = 1" | sudo tee -a /etc/sysctl.conf

Disable IP forwarding - echo 0 | sudo tee /proc/sys/net/ipv4/ip\_forward

Prevent IP spoofing - echo "nospoof on" | sudo tee -a /etc/host.conf

# >**Configure services**

Check service configuration files for required services. Usually a wrong setting in a config file for sql, apache, etc. will be a point. MySQL, SSH, Apache, README software

Ensure all services are legitimate - service --status-all

# >**Check for malware**

ClamAV

* sudo apt-get install clamav (or sudo apt-get install clamtk for GUI, run with sudo clamtk)
* Update signatures with sudo freshclam
* Scan with clamscan -r --bell -i $directory
* Scan whole system (NOT recommended!): clamscan -r --remove /

Alternatively, use AVG (download from<http://free.avg.com/usen/download>, get “.deb” version for Ubuntu)

* dpkg -i $package
* sudo /etc/init.d/avgd start
* sudo avgupdate
* sudo avgscan /

chkrootkit

* sudo apt-get install chkrootkit
* sudo chkrootkit -q

rkhunter

* sudo apt-get install rkhunter
* sudo rkhunter --update
* sudo rkhunter --propupd
* sudo rkhunter -c --enable all --disable none

Check crontab for malware/unauthorized actions (do this as root and as every user) - opens crontab file for you to check

As you: crontab -e

As root: sudo crontab -e

As another user: sudo su - $user then crontab -e

Remove with: crontab -r

Remove entire folder: rm -r /var/spool/cron/crontabs

Check files in /etc/cron.d, /etc/cron.daily, /etc/cron.hourly, /etc/cron.monthly, /etc/cron.weekly, make sure the contents are legitimate

Secure cron by creating an at.allow and at.deny file in /etc if they don’t already exist by using sudo gedit /etc/at.allow and/or sudo gedit /etc/at.deny

Check places like /etc/cron.d, /usr/bin, /boot, /home/$user/.mozilla/firefox/, /usr/lib/firefox-addons/extensions/, /dev/, /etc/rc.local, /etc/rc2.d, /etc/profile.d, /tmp, /proc, ~/.profile, ~/.bashrc, /etc/profile, /usr/share/, etc. for scripts, suspicious files, trojans, rootkits, backdoors, rootkit servers, etc., some of which may be hidden files (our AV and rootkit scans won’t find some as they may be user-created “malware”, also file explorer may good way to visually identify scripts and executables)

Use sudo ls -R | grep "\.sh" | less

# >**Remove world readable permissions to home directory**

sudo chmod 0750 /home/$username (RUN AS USER OF HOME DIRECTORY, NOT ROOT OR ANOTHER USER OR ELSE YOU WILL BE LOCKED OUT OF THOSE FOLDERS)

Check permissions with ls -l /home

# >**Lock/unlock users**

sudo passwd -l $username

sudo passwd -u $username

# >**Check processes**

ps aux

ps -ax

top

htop

System Monitor

Default processes:

See what’s “listening” and/or running on system (watch for remote access programs like Telnet server, netcat, backdoors, etc.)

* lsof -i
* netstat -an
* ps -ef

# >**Logs**

/var/log/messages : General log messages

/var/log/boot : System boot log

/var/log/debug : Debugging log messages

/var/log/auth.log : User login and authentication logs

/var/log/daemon.log : Running services such as squid, ntpd and others log

message to this file

/var/log/kern.log : Kernel log file

View logs with: tail, more, cat, less, grep, GNOME System Log Viewer

Alternatively, search System Log in the Ubuntu menu to see available logs

Four types:

* auth.log - tracks authentication events that prompt for user passwords (ex. sudo)
* dpkg.log - tracks software events (ex. Installations and updates)
* syslog - tracks OS events (ex. Error messages)
* Xorg.0.log - tracks desktop event (ex. Service changes and graphic card errors)

/var should be strictly for logs

# >**Audit policies**

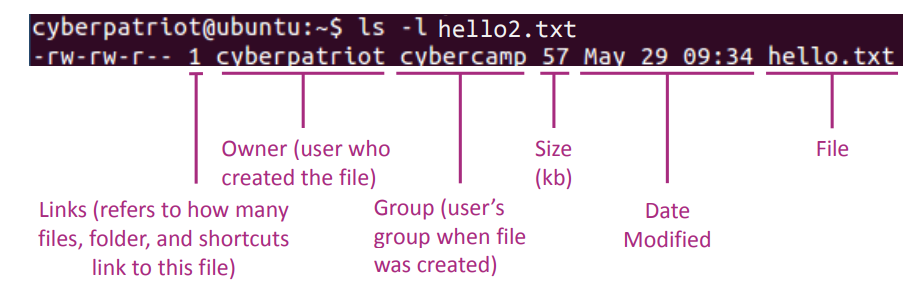
sudo apt-get install auditd

sudo auditctl -e 1 (enable audits)

View and modify policies with sudo gedit /etc/audit/auditd.conf

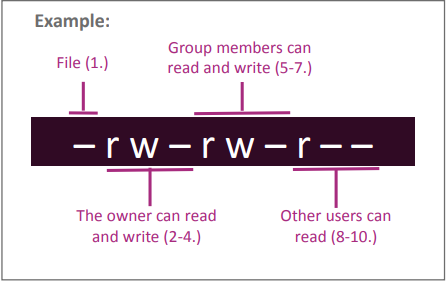
# >**File permissions**

ls -l $filename (check to ensure proper owners)



First items listed are file permissions (split into 10 fields)

Blank fields mean users in that section cannot do that action with the file/directory

r - read permission

w - write/modify permission

x - execute permission

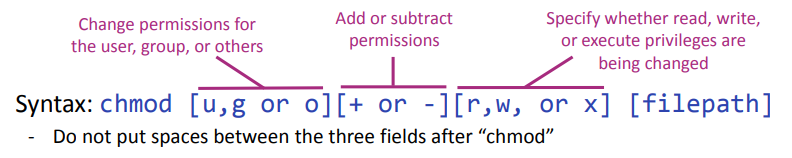
1. Type: “d” means item is a directory, blank means it is a file

2-4. Owner File Permissions

5-7. Group File Permissions

8-10. Other File Permissions

# >**chmod command**

****

Ex. chmod o-r hello2.txt (subtract read permissions from others for file hello2.txt)

Open /etc/login.defs and change the UMASK variable to 077

Check for files with no owner: sudo find / -nouser and remove (or assign owner with sudo chown $username $file

Check for files with no group owner: sudo find / -nogroup and remove (or assign owner with sudo chgrp $groupname $file

Make sure all users have a home directory assigned in /etc/passwd

CREATE\_HOME yes in /etc/login.defs

# Check that the home directory of user is group-owned by the user’s primary GID: check home directory in /etc/passwd and compare with groups in /etc/group

Check system for world-writable directories with sudo find / -perm -2 -type d ! -group sys ! -group root ! -group bin -exec ls -lLd {} \; and ensure those directories are group-owned by root, sys, bin, or an application group (if not, change with chgrp root $directory)

/var/log directory must be group-owned by syslog, check with ls -lad /var/log | cut -d' ' -f4, if the output is not syslog then change with sudo chgrp syslog /var/log

/var/log directory must be owned by root, check with ls -lad /var/log | cut -d' ' -f3, if the output is not root then change with sudo chown root /var/log

/etc/passwd should have permission of 644 (chmod 644 /etc/passwd)

/etc/shadow should have permission of 600 (chmod 600 /etc/shadow)

# >**Shared folders/directories**

NFS:

* exports
* cat /etc/exportfs
* Use to stop NFS
* sudo service portmap stop
* sudo service nfs-kernel-server stop
* OR (instead of the previous two, run these two):
* sudo /etc/init.d/portmap stop
* sudo /etc/init.d/nfs-kernel-server stop
* Delete packages with: sudo apt-get --purge remove nfs-kernel-server nfs-common portmap
* sudo umount /$directory

Samba:

* smbstatus
* smbclient -L host
* smbstatus --shares (shows shared folders)
* testparm
* net rpc user
* net rpc share list -U $username
* sudo umount -a -t cifs -l

Disable file sharing of folders (right-click on folder -> file sharing -> uncheck box)

# >**Screen lock**

Log user out after 10 min, require password before unlocking

Settings -> Lock ON -> Lock screen after 10 min -> Require password

<https://askubuntu.com/questions/1042641/how-to-set-custom-lock-screen-time-in-ubuntu-18-04>

# >**Local security policy**

Modify startup/configuration files

Startup files located in /etc/rc\*

Configuration files are everywhere (depending on application and service)

Check /etc/init.d for bad init files

# >**Harden VSFTPD**

**ONLY DO THIS IF README SAYS SO**

Disable anonymous uploads:

* sudo sed -i '/^anon\_upload\_enable/ c\anon\_upload\_enable no' /etc/vsftpd.conf
* sudo sed -i '/^anonymous\_enable/ c\anonymous\_enable=NO' /etc/vsftpd.conf

FTP user directories use chroot:

* sudo sed -i '/^chroot\_local\_user/ c\chroot\_local\_user=YES' /etc/vsftpd.conf
* sudo service vsftpd restart

# >**Harden MySQL**

<https://www.techrepublic.com/article/how-to-harden-mysql-security-with-a-single-command/>

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

# >**Harden NGINX**

<https://www.acunetix.com/blog/web-security-zone/hardening-nginx/>

<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-on-ubuntu-14-04>

<https://hostadvice.com/how-to/how-to-harden-nginx-web-server-on-ubuntu-18-04/>

<https://www.linuxtechi.com/harden-secure-nginx-web-server-linux/>

<https://www.upguard.com/blog/how-to-build-a-tough-nginx-server-in-15-steps>

# >**Harden bind9**

must be configured to hide its version number

must be configured to not allow zone transfers

# >**Harden FTP**

Root account must not be allowed to login to the FTP server.

The FTP user must not have a root UID.

# >**Check host and nameservers**

sudo gedit /etc/resolv.conf (make sure it looks something like “nameserver x.x.x.x”. Try using 8.8.8.8)

sudo gedit /etc/hosts (make sure your traffic isn’t redirecting)

# >**Run Lynis AV (optional)**

sudo apt-get install lynis

OR

wget https://downloads.cisofy.com/lynis/lynis-2.7.5.tar.gz -O lynis.tar.gz

sudo tar -xzf ./lynis.tar.gz --directory /usr/share/

cd /usr/share/lynis

/usr/share/lynis/lynis update info

/usr/share/lynis/lynis audit system

Other methods of installation:<https://cisofy.com/documentation/lynis/get-started/#installation-package>

Running Lynis:<https://cisofy.com/documentation/lynis/get-started/#first-run>

# >**Secure sysctl**

sudo sysctl -w net.ipv4.tcp\_syncookies=1

sudo sysctl -w net.ipv4.ip\_forward=0

sudo sysctl -w net.ipv4.conf.all.send\_redirects=0

sudo sysctl -w net.ipv4.conf.default.send\_redirects=0

sudo sysctl -w net.ipv4.conf.all.accept\_redirects=0

sudo sysctl -w net.ipv4.conf.default.accept\_redirects=0

sudo sysctl -w net.ipv4.conf.all.secure\_redirects=0

sudo sysctl -w net.ipv4.conf.default.secure\_redirects=0

sudo sysctl -w net.ipv4.icmp\_echo\_ignore\_broadcasts=1

Disable “magic” kernel key - set /proc/sys/kernel/sysrq to 0 or run sudo sysctl -w kernel.sysrq=0

Set /proc/sys/net/ipv4/tcp\_rfc1337 to 1

OR

Edit manually with: sudo gedit /etc/sysctl.conf

# >**Disable CTRL-ALT-DEL**

sudo systemctl mask ctrl-alt-del.target

sudo systemctl daemon-reload

If this command doesn’t work (older versions of Ubuntu?) then use: sudo gedit /etc/init/control-alt-delete.conf and remove everything after the author OR sudo gedit /etc/event.d/control-alt-delete and remove everything after the author OR delete the file entirely

# >**Locked out (password doesn’t work)**

*Might work*

Ctrl+Alt+F1 then type username and password, then run: sudo rm .Xauthority sudo reboot

# >**Disable automatic mounting**

sudo echo “install usb-storage /bin/true” >> /etc/modprobe.d/DISASTIG.conf

sudo systemctl stop autofs - automount service

<https://help.ubuntu.com/community/Mount/USB> - different method

# >**Apparmor**

This probably will already be on the system but if not use the following three lines to install

sudo apt-get install libpam-apparmor

sudo systemctl enable apparmor.service

sudo systemctl start apparmor.service

sudo apparmor\_status (if all loaded profiles are not in “enforce" mode, or there are any profiles in "complain" mode, this is a finding)

# >**Initialization files**

Local initialization files must have mode 0740 or less (check with ls -al /home/$username/.\* | more and fix with chmod 0740 /home/$username/.$initfile)

Check executable search path of init files - they should only resolve to system default or the user’s home directory: grep -i path /home/$username/.\*

If they don’t match the user home directory listed in /etc/passwd or system default, open file with gedit and fix

Local initialization files must not execute world-writable programs, check for world-writable files with sudo find / -perm -002 -type f -exec ls -ld {} \; | more and for all files listed, check for their presence in the local initialization files with grep $filename /home/\*/.\*, set mode on files being executed by local initialization files with chmod 0755 $filename

# >**Set GRUB Password**

grub2-mkpasswd-pbkdf2 or grub-mkpasswd-pbkdf2

Enter password to use

Copy hashed password (looks something like grub.pbkdf2.sha512.10000.somestuff)

sudo gedit /etc/grub.d/00\_header

Paste at the bottom of the file:

cat << EOF

set superusers="admin"

password\_pbkdf2 admin $hash

EOF

Then run sudo update-grub to update GRUB

# >**Other**

Remove bad PATH routing - check /etc/environment

/proc/mounts - edit /etc/fstab to change line with shm to tmpfs /dev/shm tmpfs ro,nosuid,nodev,noexec 0 0

/etc/rc.local should only have exit 0

Encrypt - sudo apt-get install ecryptfs-utils cryptsetup

sudo ecryptfs-migrate-home -u $username - encrypt filesystem

Access.conf?

Lock down sudo su command (gives root shell even if root is disabled) - research

# **Previous competition results**

| **Competition** | **Missed** |
| --- | --- |
| Round 1 | 1 |
| Round 2 | Application Security Settings - 1  Operating System Updates - 1  Service Auditing - 2  Uncategorized Operating System Settings - 1 |
| State (Platinum) | Account Policies - 1  Application Security Settings - 6  Forensics Questions - 1  Local Policies - 3  Prohibited Files - 2  Unwanted Software - 2  Uncategorized Operating System Settings - 1  User Auditing - 2 |
| Semifinals (Platinum) |  |

# **Avenues of further research**

<https://neprisstore.blob.core.windows.net/sessiondocs/doc_362f4940-9202-4477-9f45-b271bc2a9877.pdf> - CIS Ubuntu Linux 14.04 LTS Benchmark

<https://neprisstore.blob.core.windows.net/sessiondocs/doc_8ac75a77-40a4-4e08-a6c0-93b39b92abd8.pdf> - CIS Ubuntu Linux 16.04 LTS Benchmark

<https://github.com/BiermanM/CyberPatriot-Scripts> - checklist + scripts

<https://quizlet.com/366166161/afa-cyber-patriot-scripts-flash-cards/> - scripts

<http://www.lacapnm.org/Cadets/STEM/CyberPatriot/SeasonVIII/CyberPatriot_Linux_CheckList.pdf> - massive checklist

<https://mega.nz/#!UJYVjQDC!s4bY7wkNLBqHmG3ZQCfXGe9uQgMG2g5b2LXKQ8AwFMI> - practice image from reddit

<https://github.com/Ryan-Galligher/CyberPatriot> - practice image w/scoring bots to check against

<https://cisofy.com/checklist/linux-security/> - linux security practices (not cyberpatriot specific)

<https://github.com/JoshuaTatum/cyberpatriot/blob/master/harrisburg-linux.sh> - script

<https://www.howtogeek.com/102009/how-to-password-protect-ubuntus-boot-loader/>

<https://www.cyberciti.biz/faq/linux-add-nodev-nosuid-noexec-options-to-temporary-storage-partitions/> - idk if this is needed but its worth checking

<http://otus.pepsipu.com/> - linux images

<https://support.ca-cyberhub.org/support/solutions/folders/33000201655> - images for every OS

<https://gist.github.com/D4stiny/4ef8fe2ea744b9f149ac34c144bd6a3d#file-cyberpatriot-2016-ubuntu-xml-L196> - actual decrypted scoring engine for platinum state 2016-2017

<https://www.redhat.com/archives/redhat-install-list/2005-October/msg00105.html> - sysctl stuff?

<https://stigviewer.com/> - hardening stuff

<https://github.com/decalage2/awesome-security-hardening> - heck ton of hardening stuff for all sorts of stuff

<https://github.com/pyllyukko/user.js> - firefox hardening

<https://github.com/trimstray/linux-hardening-checklist> - generic hardening

<https://github.com/imthenachoman/How-To-Secure-A-Linux-Server> - generic hardening

Game conqueror

ManaPlus

<https://askubuntu.com/questions/1011368/how-can-i-protect-against-single-user-mode>

Sshd\_config tcpkeepalive no, printlastlog yes