Ultimate Ubuntu 22 Checklist

ALWAYS ANSWER FORENSICS QUESTIONS FIRST, THE SETTINGS IN THIS CHECKLIST MAY MESS UP PRECONFIGURED SETTINGS

Initial Setup:

- ☐ **Disable cramfs with the following script**
- ☐ **Disable squashfs with the following script**
- ☐ **Disable udf with the following script**
- ☐ **Ensure /tmp is a separate partition**
  - ☐ Run this command to see if it is enabled: # systemctl is-enabled tmp.mount
  - ☐ If not, run the following commands:
    - ☐ # systemctl unmask tmp.mount

- ☐ **Configure the /etc/fstab file:**
  - ☐ Edit the /etc/fstab file and add **nodev,noexec,rw,nosuid,relatime,seclabel** to the fourth field (mounting options) for the /tmp partition so it looks like this:

```
rw,nosuid,nodev,noexec,relatime,seclabel
```

- ☐ **Ensure package manager repositories are configured:**
  - ☐ Run the following command and verify they are configured correctly:
    - ☐ # apt-cache policy

- ☐ **Ensure GPG keys are configured:**
  - ☐ Run the following command and verify they are configured correctly:
    - ☐ # apt-key list

- ☑ ~~**Ensure AIDE is installed:**~~
  - ☑ ~~Run the following command:~~
    - ☑ ~~# apt install aide aide-common~~
  - ☐ Run these commands to initialize aide:
    - ☐ # aideinit
    - ☐ # mv /var/lib/aide/aide.db.new /var/lib/aide/aide.db

- ☐ **Ensure file system integrity is regularly checked:**
  - ☐ Edit the file to enable this:
    - ☐ # sudo pico /etc/systemd/system/aidecheck.service
    - ☐ Add the following lines:

```
[Unit]
Description=Aide Check

[Service]
Type=simple
ExecStart=/usr/bin/aide.wrapper --config
/etc/aide/aide.conf --check

[Install]
WantedBy=multi-user.target
```

    - ☐ # sudo pico /etc/systemd/system/aidecheck.timer
    - ☐ Add the following lines:

```
[Unit]
Description=Aide check every day at 5AM
```

```
[Timer]
OnCalendar=*-*-* 05:00:00 Unit=aidecheck.service

[Install]
WantedBy=multi-user.target
```

- ☐ Lastly, run the following commands:

```
# chown root:root /etc/systemd/system/aidecheck.*
# chmod 0644 /etc/systemd/system/aidecheck.*

# systemctl daemon-reload

# systemctl enable aidecheck.service
# systemctl --now enable aidecheck.timer
```

- ☐ **Ensure bootloader password is set:**
  - ☐ Run the following commands:

```
# grub-mkpasswd-pbkdf2
Enter password:
Reenter password:
PBKDF2 hash of your password is <encrypted-password>
```

  - ☐ Next add the following into a custom /etc/grub.d file:

```
cat <<EOF
set superusers="<username>"
password_pbkdf2 <username> <encrypted-password>
EOF
```

  - ☐ And lastly, run the following command:

```
# update-grub
```

- ☑ ~~**Ensure permissions on bootloader config are configured correctly:**~~
  - ☑ ~~Run the following commands:~~

```
# chown root:root /boot/grub/grub.cfg
# chmod u-wx,go-rwx /boot/grub/grub.cfg
```

- ☑ ~~**Ensure authentication required for single user mode:**~~
  - ☑ ~~Run the following commands:~~

```
# sudo passwd root
```

- ☐ **Ensure ASLR is enabled:**
  - ☐ Set the following parameter in **/etc/sysctl.conf** or **/etc/sysctl.d/***

```
kernel.randomize_va_space = 2
```

- ☑ ~~**Ensure prelink is not installed:**~~
  - ☑ ~~Run the following commands:~~

```
# prelink -ua
# apt purge prelink
```

- ☑ ~~**Ensure Automatic Error Reporting is disabled:**~~
  - ☑ ~~Run the following two commands to ensure it is disabled:~~

```
# dpkg-query -s apport > /dev/null 2>&1 && grep -Psi --
'^\h*enabled\h*=\h*[^0]\b' /etc/default/apport

# systemctl is-active apport.service | grep '^active'
```

  - ☑ ~~Nothing should be returned on either of these commands, if something is, run the following commands to disable:~~

```
# apt purge apport
```

  - ☐ If the service is marked necessary, disable like this:

```
# sudo pico /etc/default/apport
```

  - ☐ Edit the “enabled” parameter to equal 0:

```
enabled=0
```

- ☐ Then run these commands to stop the service:

```
# systemctl stop apport.service
# systemctl --now disable apport.service
```

- ☐ **Ensure core dumps are restricted:**
  - ☐ Run the following commands:

```
# sudo pico /etc/security/limits.conf
```

  - ☐ If that doesn't work, do this command instead:

```
# sudo pico /etc/security/limits.d/*
```

  - ☐ Next add the following line to whichever one of those files opened:

```
* hard core 0
```

  - ☐ Run the following commands:

```
# sudo pico /etc/sysctl.conf
```

  - ☐ If that doesn't work, do this command instead:

```
# sudo pico /etc/sysctl.d/*
```

  - ☐ Next add the following line to whichever one of those files opened:

```
fs.suid_dumpable = 0
```

  - ☐ Run the following command to set the active kernel parameter:

```
# sysctl -w fs.suid_dumpable=0
```

  - ☐ If systemd-coredump is installed, refer to the picture below:

edit `/etc/systemd/coredump.conf` and add/modify the following lines:

```
Storage=none
ProcessSizeMax=0
```

Run the command:

```
systemctl daemon-reload
```

- ☐ **Configuring/Installing AppArmor:**
  - ☑ ~~Install AppArmor with the following command:~~

```
# apt install apparmor
```

  - ☐ Run the following command:

```
# sudo pico /etc/default/grub
```

  - ☐ Add the following parameters to the GRUB_CMDLINE_LINUX= line so it looks like this:

```
GRUB_CMDLINE_LINUX="apparmor=1 security=apparmor"
```

  - ☐ Run the following command to update the configuration:

```
# update-grub
```

  - ☐ Run the following command to set all profiles to enforce:

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

  - ☐ Next run the following command and verify no processes are unconfined:

```
# apparmor_status | grep processes
```

- ☐ **Ensure message of the day is configured properly:**
  - ☐ Run the following command and verify no results are returned:

```
# grep -Eis "(\\v|\\r|\\m|\\s|$(grep '^ID='
/etc/os-release | cut -d= -f2 | sed -e 's/"//g'))"
/etc/motd
```

- ☐ If something is returned, remove the message with the following command:

```
# sudo pico /etc/motd
```

- ☐ **Ensure local login warning banner is configured:**
  - ☐ Run the following command:

```
# sudo pico /etc/issue
```

  - ☐ Change the warning banner to the following:

```
"Authorized uses only. All activity may be monitored and reported."
```

- ☐ **Ensure local login warning banner is configured:**
  - ☐ Run the following command:

```
# sudo pico /etc/issue.net
```

  - ☐ Change the warning banner to the following:

```
"Authorized uses only. All activity may be monitored and reported."
```

- ☐ **Ensure permissions on /etc/motd are configured:**
  - ☐ Run the following commands to set permissions:

```
# chown root:root $(readlink -e /etc/motd)
# chmod u-x,go-wx $(readlink -e /etc/motd)
```

- ☐ **Ensure permissions on /etc/issue are configured:**
  - ☐ Run the following commands to set permissions:

```
# chown root:root $(readlink -e /etc/issue)
# chmod u-x,go-wx $(readlink -e /etc/issue)
```

- ☐ **Ensure permissions on /etc/issue.net are configured:**

- ☐ Run the following commands to set permissions:

```
# chown root:root $(readlink -e /etc/issue.net)
# chmod u-x,go-wx $(readlink -e /etc/issue.net)
```

- ☑ ~~**Disable automounting:**~~
  - ☑ ~~Run the following command:~~

```
# apt purge autofs
```

- ☐ **Disable USB storage:**
  - ☐ Run the following script.

- ☑ ~~**Ensure a single time synchronization daemon is in use:**~~
  - ☑ ~~Run the following commands:~~

```
# apt install chrony

# systemctl stop systemd-timesyncd.service

# systemctl -now mask systemd-timesyncd.service

# apt purge ntp
```

- ☐ **Configuring chrony:**
  - ☐ Run the following commands:

```
# sudo pico /etc/chrony/chrony.conf
```

  - ☐ If that doesn't work

```
# sudo pico /etc/chrony/sources.d/
```

  - ☐ and add or edit server or pool lines as appropriate according to local site policy:

```
<[server|pool]> <[remote-server|remote-pool]>
```

- ☐ pool directive:

```
pool time.nist.gov iburst maxsources 4 #The maxsources
option is unique to the pool directive
```

- ☐ server directive:

```
server time-a-g.nist.gov iburst server 132.163.97.3 iburst
server time-d-b.nist.gov iburst
```

- ☐ Lastly restart chrony:

```
# systemctl restart chronyd

# chronyc reload sources
```

- ☐ **Ensure chrony is running as user _chrony**
  - ☐ Run the following command:

```
# sudo pico /etc/chrony/chrony.conf
```

- ☐ If that doesn't work;

```
# sudo pico /etc/chrony/conf.d/
```

- ☐ Add the following line to either file (or both)

```
user _chrony
```

- ☐ **Ensure chrony is running**
  - ☐ Run the following commands

```
# systemctl unmask chrony.service

# systemctl -now enable chrony.service
```

****IF YOU LOSE POINTS FOR STOPPING SYSTEMD/NTP REINSTALL WHICHEVER, AND CONFIGURE IT ACCORDING TO CIS BENCHMARKS****

- ☑ ~~**Ensure X Window System isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge xserver-xorg*
```

- ☑ ~~**Ensure Avahi Server isn't installed**~~
  - ☑ ~~Run the following commands:~~

```
# systemctl stop avahi-daemon.service
# systemctl stop avahi-daemon.socket
```

```
# apt purge avahi-daemon
```

- ☑ ~~**Ensure CUPS isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge cups
```

- ☑ ~~**Ensure DHCP Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge isc-dhcp-server
```

- ☑ ~~**Ensure LDAP Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge slapd
```

- ☑ ~~**Ensure NFS isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge nfs-kernel-server
```

- ☑ ~~**Ensure DNS Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge bind9
```

- ☑ ~~**Ensure FTP Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge vsftpd
```

- ☑ ~~**Ensure HTTP Server isn't installed (If not listed as an essential service)**~~
  - ☑ ~~Run the following command:~~

```
# apt purge apache2
```

- ☑ ~~**Ensure IMAP and POP3 Server aren't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge dovecot-imapd dovecot-pop3d
```

- ☑ ~~**Ensure Samba Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge samba
```

- ☑ ~~**Ensure HTTP Proxy Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge squid
```

- ☑ ~~**Ensure SNMP Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge snmp
```

- ☑ ~~**Ensure NIS Server isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge nis
```

- ☑ ~~**Ensure MariaDB isn't installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge mariadb
```

- ☐ **Ensure mail transfer agent is configured**
  - ☐ Run the following command:

```
# sudo pico /etc/postfix/main.cf
```

  - ☐ If it doesn't work skip this section, if it does find the Receiving Mail Section

```
inet_interfaces = loopback-only
```

  - ☐ Then restart the service

```
# systemctl restart postfix
```

- ☑ ~~**Ensure rsync is not installed**~~
  - ☑ ~~Run the following command:~~

```
# apt purge rsync
```

- ☑ ~~**Ensure RSH Client isn't installed:**~~
  - ☑ ~~Run the following command:~~

```
# apt purge rsh-client
```

- ☑ ~~**Ensure Talk Client isn't installed:**~~
  - ☑ ~~Run the following command:~~

```
# apt purge talk
```

- ☑ ~~**Ensure Telnet Client isn't installed:**~~
  - ☑ ~~Run the following command:~~

```
# apt purge telnet
```

- ☑ ~~**Ensure LDAP Client isn't installed:**~~
  - ☑ ~~Run the following command:~~

```
# apt purge ldap-utils
```

- ☑ ~~**Ensure RPC isn't installed:**~~
  - ☑ ~~Run the following command:~~

```
# apt purge rpcbind
```

- ☐ **Ensure nonessential services are disabled:**
  - ☐ Run the following command to see all services on the system:

```
# lsof -i -P -n | grep -v "(ESTABLISHED)"
```

  - ☐ Disable any with the following command:

```
# apt purge <package_name>
```

- ☐ **Sysctl configuration:**
  - ☐ Run the following command:

```
# sudo pico /etc/sysctl.conf
```

  - ☐ Add the following parameters to the bottom of the file:

```
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.route.flush=1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.ip_forward = 0
net.ipv6.conf.all.forwarding = 0
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
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1
net.ipv4.icmp_echo_ignore_broadcasts = 1
icmp_ignore_bogus_error_responses = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_syncookies = 1
net.ipv6.conf.all.accept_ra = 0
net.ipv6.conf.default.accept_ra = 0
net.ipv4.tcp_rfc1337 = 1
kernel.sysrq=0
kernel.dmesg_restrict=1
kernel.unprivileged_userns_clone=0
```

- ☐ **Disable any wireless interfaces:**
  - ☐ Run the following script:

- ☐ **Disable grub configuration world readability:**
  - ☐ Run the following command:

```
# sudo pico /etc/grub.d/grub.conf
```

  - ☐ If that doesn't work, run the following:

```
# sudo pico /etc/default/grub
```

  - ☐ Add the following parameter to either (or both)

```
GRUB_TIMEOUT_STYLE=hidden
```

- ☐ **Mount /tmp securely:**
  - ☐ Run the following command:

```
# mount -o loop,noexec,nosuid,rw /var/tempFS /tmp
```

- ☐ **Enable stricter defaults for shared memory:**
  - ☐ By default, runshm is mounted as read/write, change it to read only with the following command:

```
# sudo pico /etc/fstab
```

  - ☐ Set the parameter to the below:

```
none     /run/shm     tmpfs     rw,noexec,nosuid,nodev
```

- ☐ **Remove phpinfo() file:**
  - ☐ Run the following command:

```
# sudo rm -rf /var/www/phpinfo.php
```

- ☐ **Disable SCTP:**
  - ☐ Run the following script to ensure sctp is disabled:

- ☐ **Disable RDS:**
  - ☐ Run the following script to ensure rds is disabled:

- ☐ **Disable TIPC:**
  - ☐ Run the following script to ensure tipc is disabled:

- ☐ **Install ufw:**
  - ☐ Run the following command:

```
# apt install ufw
```

- ☐ **Ensure iptables-persistent is not installed with ufw:**
  - ☐ Run the following command:

```
# apt purge iptables-persistent
```

- ☐ **Ensure the ufw service is enabled:**
  - ☐ Run the following commands:

```
# systemctl unmask ufw.service
# systemctl -now enable ufw.service
# ufw enable
```

- ☐ **Ensure ufw loopback traffic is configured:**
  - ☐ Run the following commands:

```
# ufw allow in on lo
# ufw allow out on lo
# ufw deny in from 127.0.0.0/8
# ufw deny in from ::1
```

- ☐ **Ensure ufw outbound connections are configured:**
  - ☐ Run the following command:

```
# ufw allow out on all
```

- ☐ **Ensure ufw default deny firewall policy:**
  - ☐ Run the following commands:

```
# ufw default deny incoming
# ufw default deny outgoing
# ufw default deny routed
```

- ☐ **Ensure auditd is installed:**
  - ☐ Run the following command:

```
# apt install auditd audispd-plugins
```

- ☐ **Ensure auditd service is enabled and active:**

- ☐ Run the following command:

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

- ☐ **Ensure auditing for processes that start prior to auditd is enabled:**
  - ☐ Run the following command:

```
# sudo pico /etc/default/grub
```

  - ☐ Set the following parameter:

```
GRUB_CMDLINE_LINUX="audit=1"
```

  - ☐ Run the following command:

```
# update-grub
```

- ☐ **Ensure audit_backlog_limit is sufficient:**
  - ☐ Run the following command:

```
# sudo pico /etc/default/grub
```

  - ☐ Set the following parameter:

```
GRUB_CMDLINE_LINUX="audit_backlog_limit=8192"
```

  - ☐ Run the following command:

```
# update-grub
```

- ☐ **Ensure audit log storage size is configured:**
  - ☐ Run the following command:

```
# sudo pico /etc/audit/auditd.conf
```

  - ☐ Set the following parameter:

```
max_log_file = 64
```

- ☐ **Ensure audit logs are not automatically deleted:**

- ☐ Run the following command:

```
# sudo pico /etc/audit/auditd.conf
```

- ☐ Set the following parameter:

```
max_log_file_action = keep_logs
```

- ☐ **If desperate for points, go through all audit/collection/log settings on the Ubuntu 22 CIS benchmark.**

- ☐ **Ensure audit configuration files are owned by root:**
  - ☐ Run the following command:

```
# find /etc/audit/ -type f \( -name '*.conf' -o -name
'*.rules' \) ! -user root -exec chown root {} +
```

- ☐ **Ensure audit configuration files belong to root group:**
  - ☐ Run the following command:

```
# find /etc/audit/ -type f \( -name '*.conf' -o -name
'*.rules' \) ! -group root -exec chgrp root {} +
```

- ☐ **Ensure systemd-journal-remote is installed:**
  - ☐ Run the following command:

```
# apt install systemd-journal-remote
```

- ☐ **Ensure systemd-journal-remote is configured:**
  - ☐ Run the following command:

```
# sudo pico /etc/systemd/journal-upload.conf
```

  - ☐ Set the following parameters:

```
URL=192.168.50.42
ServerKeyFile=/etc/ssl/private/journal-upload.pem
```

```
ServerCertificateFile=/etc/ssl/certs/journal-upload.pem
TrustedCertificateFile=/etc/ssl/ca/trusted.pem
```

- ☐ Ensure it cannot receive logs from a remote client:

```
# systemctl --now disable systemd-journal-remote.socket
```

- ☐ More configuration:

```
# sudo pico /etc/systemd/journald.conf
```

- ☐ Add the following parameters:

```
Compress=yes
Storage=persistent
ForwardToSyslog=yes
```

- ☐ Restart the service:

```
# systemctl restart systemd-journal-upload
```

- ☐ **Ensure cron daemon is enabled and running:**
  - ☐ Run the following command:

```
# systemctl -now enable cron
```

- ☐ **Ensure permissions on cron are configured:**
  - ☐ Run the following commands:

```
# chown root:root /etc/crontab
# chmod og-rwx /etc/crontab
# chown root:root /etc/cron.hourly/
# chmod og-rwx /etc/cron.hourly/
# chown root:root /etc/cron.daily/
# chmod og-rwx /etc/cron.daily/
# chown root:root /etc/cron.weekly/
# chmod og-rwx /etc/cron.weekly/
# chown root:root /etc/cron.monthly/
# chmod og-rwx /etc/cron.monthly/
# chown root:root /etc/cron.d/
```

```
# chmod og-rwx /etc/cron.d/
```

```
# rm /etc/cron.deny
# touch /etc/cron.allow
# chmod g-wx,o-rwx /etc/cron.allow
# chown root:root /etc/cron.allow
```

- ☐ If **SSH** is a critical service, refer to the Ubuntu 22 CIS benchmark section 5.2 if not, run the following command:

```
# apt purge openssh-server
```

- ☐ **Ensure sudo is installed:**
  - ☐ Run the following command:

```
# apt install sudo
```

- ☐ **Ensure a sudo log file exists:**
  - ☐ Run the following command:

```
# sudo pico /etc/sudoers
```

  - ☐ Add the following parameter to the file:

```
Defaults logfile="/var/log/sudo.log"
```

- ☐ **Ensure access to the su command is restricted:**
  - ☐ Run the following commands:

```
# groupadd sugroup
# sudo pico /etc/pam.d/su
```

  - ☐ Add/edit the following parameter:

```
auth required pam_wheel.so use_uid group=sugroup
```

- ☐ **Installing and configuring PAM:**

- ☐ Run the following commands:

```
# apt install libpam-pwquality
```

- ☐ Next, edit the password policy:

```
# sudo pico /etc/security/pwquality.conf
```

- ☐ Paste the following starting from where and including “minlen = 3”

```
minlen = 14 minclass = 4 dcredit =-1 ucredit =-1 ocredit
=-1 lcredit =-1 maxrepeat=5 gecoscheck=1 dictcheck=1
enforce_for_root
```

- ☐ Next run the following command:

```
# sudo pico /etc/pam.d/common-account
```

- ☐ Ensure that “account required pam_faillock.so” is at the end of the file
- ☐ Run the following command:

```
# sudo pico /etc/security/faillock.conf
```

- ☐ Configure the following parameters:

```
deny = 4
fail_interval = 900
unlock time = 600
```

- ☐ Ensure password reuse is limited:

```
# sudo pico /etc/pam.d/common-password
```

- ☐ Edit the existing parameters so it looks like this:

```
password [success=1 default=ignore] pam_unix.so obscure
use_authtok try_first_pass yescrypt remember=28
```

- ☐ Ensure the hashing algorithm is up to date:

```
# sudo pico /etc/login.defs
```

- ☐ Ensure that “ENCRYPT_METHOD” is set to “yescrypt”

☐ **Enforcing the newest password hashing algorithm:**

- ☐ Run the following script:

- ☐ **Securing password changes:**
  - ☐ Run the following command:

```
# sudo pico /etc/login.defs
```

  - ☐ Change the following parameters to below:

```
PASS_MIN_DAYS 1
PASS_MAX_DAYS 45
PASS_WARN_AGE 7
```

  - ☐ Next enforce this for all users:

```
# sudo pico /etc/shadow
```

  - ☐ Change the “0”s to “1”s and “99999”s to “45”s for all users

- ☐ **Ensure password inactivity is set:**
  - ☐ Run the following command:

```
# useradd -D -f 30
```

- ☐ **Change all users passwords:**
  - ☐ Run the following command for all users in the readme file and “root”:

```
Passwd {USER} Cyb3Rp4tr!0t$22!
```

- ☐ **Ensure default group for the root account is GID 0:**
  - ☐ Run the following command:

```
# usermod -g 0 root
```

- ☐ **Ensure all users have the right GID:**
  - ☐ Run the following command:

```
# sudo pico /etc/passwd
```

  - ☐ Look next to a username and make sure the number on

the left (UID) matches the number on the right (GID)

- ☐ **Ensure only authorized users exist on the system:**
  - ☐ Check the ReadMe file for a list of authorized users.
  - ☐ Compare this list to the users shown in /etc/passwd
  - ☐ Remove any unauthorized users with the following command:

```
# sudo deluser {USER}
```

  - ☐ Add any users with the following command:

```
# sudo useradd {USER}
```

  - ☐ Check for unauthorized administrators through the GUI:

```
Press the Windows key and search "Users"
```

  - ☐ Open the settings for user management and press unlock
  - ☐ After unlocking, go through each user and toggle the "Administrator" button according to the ReadMe file

- ☐ **Lock the root account:**
  - ☐ Run the following command:

```
# sudo passwd -l root
```

- ☐ **Configuring Permissions:**
  - ☐ Run the following commands to ensure all permissions on sensitive files are correctly set:

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

- ☐ **Ensure no world writable files exist:**
  - ☐ Run the following command to find these files:

```
# find <partition> -xdev -type f -0002
```

  - ☐ Remove these files with the following command:

```
# chmod o-w <filename>
```

- ☐ **Ensure no unowned files exist:**
  - ☐ Run the following command to find these files:

```
# find <partition> -xdev -nouser
```

  - ☐ Then run this command if any exist:

```
# chown root:root <filename>
```

- ☐ **Ensure no ungrouped files exist:**
  - ☐ Run the following command to find these files:

```
# find <partition> -xdev -nogroup
```

  - ☐ Then run this command if any exist:

```
# chown root:root <filename>
```

- ☐ **Ensure accounts in /etc/passwd use shadowed passwords**
  - ☐ Run the following command:

```
# sed -e 's/^\([a-zA-Z0-9_]*\):[^:]*:/\1:x:/' -i
/etc/passwd
```

- ☐ **Ensure all groups in /etc/passwd exist in /etc/group**
  - ☐ Run the following script
  - ☐ Ensure nothing is returned

- ☐ **Ensure shadow group is empty**
  - ☐ Run the following command:

```
# sed -ri 's/(^shadow:[^:]*:[^:]*:)([^:]+$)/\1/' /etc/group
```

- ☐ **Ensure root PATH Integrity:**
  - ☐ Run the following script
  - ☐ Ensure nothing is returned, if something is, fix it

- ☐ **Ensure root is the only UID 0 account:**
  - ☐ Run the following command:

```
# awk -F: '($3 == 0) { print $1 }' /etc/passwd
```

  - ☐ If anything other than root is returned, fix it in /etc/passwd by setting the UID to the GID

- ☐ **Ensure all user's home directories exist:**
  - ☐ Run the following script

- ☐ **Ensure no users have .netrc directories:**
  - ☐ Run the following script

- ☐ **Ensure no users have .ssh directories:** (Only run this script if there is no mention of SSH being necessary)
  - ☐ Run the following script

- ☐ **Ensure no users have .forward directories:**
  - ☐ Run the following script

- ☐ **Ensure no users have .rhosts directories:**
  - ☐ Run the following script

- ☐ **Check for malicious services running:**
  - ☐ Run the following command to see all services running:

```
# systemctl list-units -type=service -all
```

  - ☐ Look through all of the running services and disable anything that doesn't seem essential

- ☐ **Check for backdoors:**
  - ☐ Run the following command to install netstat:

```
# apt install net-tools
```

  - ☐ Run the following command to see all open ports:

```
# netstat -ntlp or #netstat -a
```

- ☐ Remove any malicious looking connections:

```
# kill -9 <PID>
```

- ☐ **Update ubuntu:**
  - ☐ Run the following commands:

```
# apt-get update
# apt-get upgrade
```

- ☐ **Update an application:** (Do this for all installed applications in the readme and web browsers)
  - ☐ Run the following commands:

```
# apt-get upgrade <application name>
# apt-get dist-upgrade <application name>
```

Firefox Configuration:

- ☐ **Import firefox settings:**
  - ☐ First, download both files from this folder:
  - ☐ Next, go to the firefox preferences located at:

```
C:\Program Files\Mozilla Firefox\defaults\pref
```

  - ☐ Paste the file “local-settings.js” into this folder
  - ☐ Then go to the firefox installation directory:

```
C:\Program Files\Mozilla Firefox\
```

  - ☐ Paste the other file, “mozilla.cfg”
  - ☐ Restart firefox, then ensure it says “Some settings managed by your domain”
  - ☐ For the settings that you’re still able to set, set them

- ☐ **Remove prohibited firefox plugins/themes:**
  - ☐ Open firefox and click the three lines
  - ☐ Click settings, then plugins
  - ☐ Remove any plugins and themes

- ☐ **Ensure no prohibited files exist on the system:**
  - ☐ Run the following command:

```
# apt install plocate
```

  - ☐ Run the following command for all media types:

```
# locate <extension> (.mp3 for example)
```

- ☐ **Ensure GRUB uses encrypted passwords:**
  - ☐ Set up grub passwords with this command:
  - ☐ # grub-crypt

- ☐ **If still out of points refer to this file or the CIS benchmarks**