Renée's Curated Checklist

Checklists referenced in creation:
-my old one
-Forty bot's checklist
-MDC3 2012 checklist (version with 2014 revisions)
-Mike's Ultraswag checklist
-random others from the old cyberpatriot archive
-cyberpatriot powerpoints
Also includes some new stuff from my own research
Color coding:
Done in Renee's script
Used for emphasis of points
Are things I am highly doubtful will get points

Tips/Notes

- Netcat is installed by default in ubuntu. You will most likely not get points for removing this version. (but still do just in case)
- Some services (such as ssh) may be required even if they are not mentioned in the readme. Others may be points even if they are explicitly mentioned in the readme

Prechecks

What commands have been run on the machine so far?

- *cat /home/* /.bash_history*
- *cat /home/* /.sh_history*
- It probably won't produce any results, but if it does, this was historically what commands were run by the users. However, in real cyber attacks almost no one would be dumb enough to leave these intact

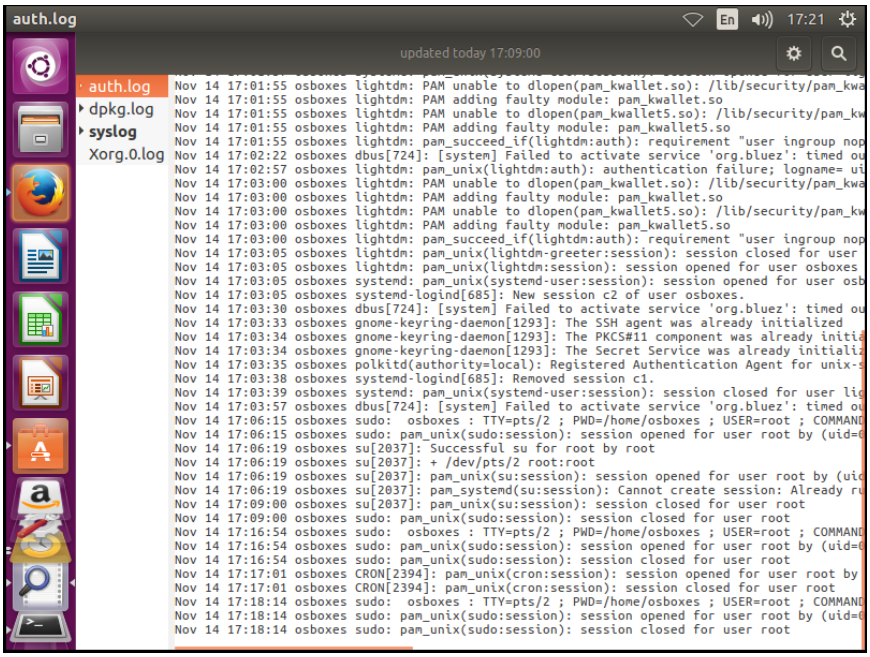
What packages have been installed on machine so far?

- In GUI:
    - Software Center
    - Installed Software

- History
    - Google the things they installed not on the competition day

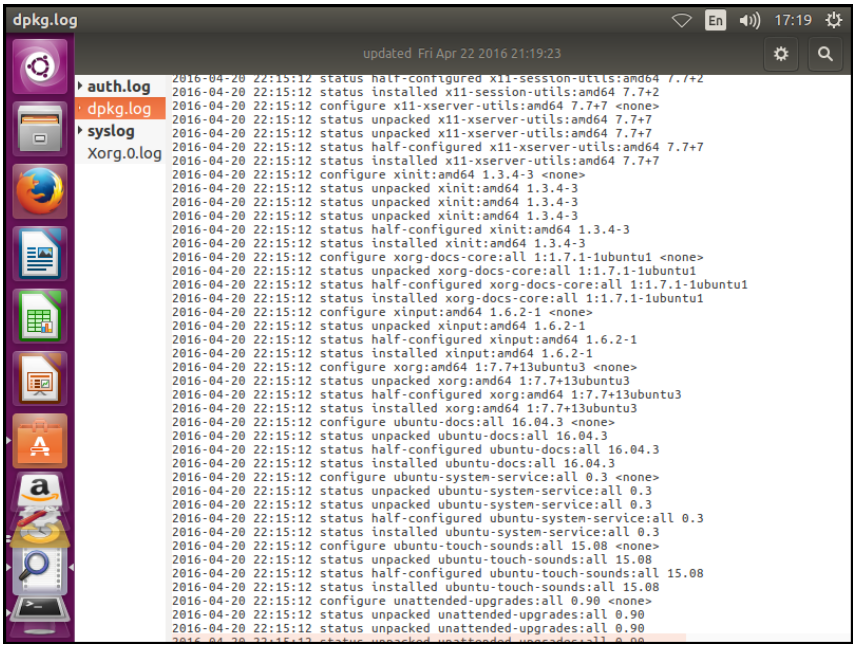
General overview of what's been done to the machine:

- In GUI, search System Log
- Check each of the logs available for weird stuff
    - **auth.log**: Tracks authentication events that prompt for user passwords (e.g., uses of PAM files and sudo)
        - Specifically check for PAM things, adding/deleting of users, etc.



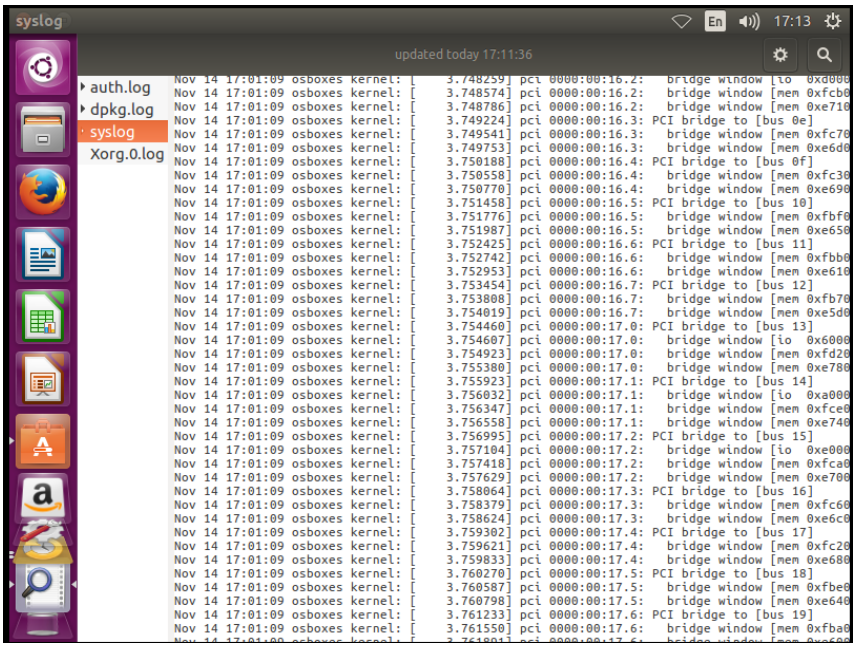
Example auth.log picture

- **DPKG.log**: Tracks software events (e.g., installations and updates)
    - Installations are important



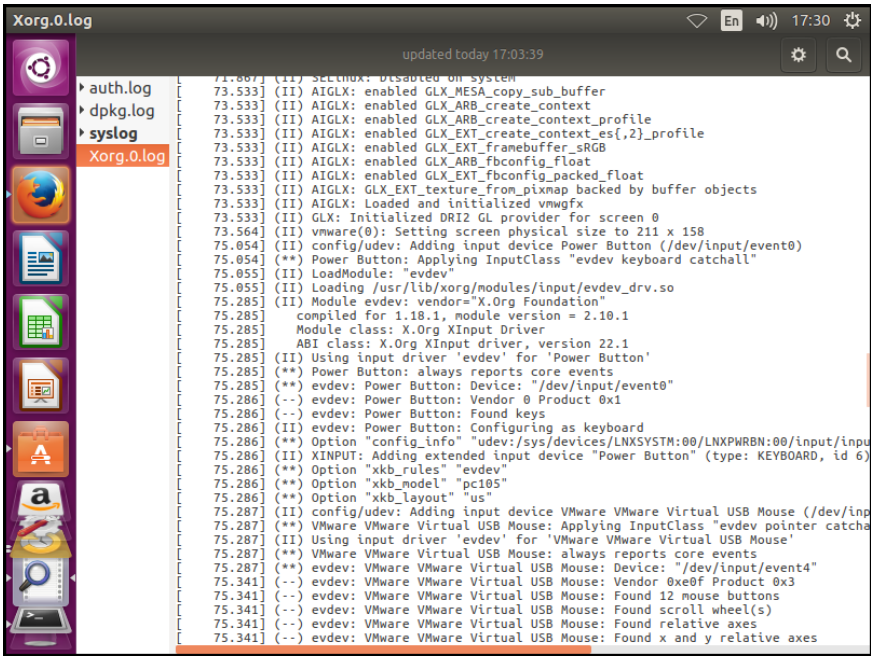
Dpkg.log picture

- **syslog**: Tracks operating system events (e.g. error messages)
  - Can most likely ignore



Syslog picture

- **Xorg.0.log**:Tracks desktop events (e.g., service changes and graphic card errors.
  - Sometimes useful for services



Example of Xorg.0.log

Are there any malicious scripts scheduled to execute?

- Check in */var/spool/cron/crontabs*
  - Look for bad stuff

```
root@osboxes:/var/spool/cron/crontabs# ls
root@osboxes:/var/spool/cron/crontabs#
```

Example of an empty crontabs directory

- Check in */var/spool/anacron*

```
root@osboxes: /var/spool/anacron
root@osboxes:/var/spool/anacron# ls
cron.daily  cron.monthly  cron.weekly
root@osboxes:/var/spool/anacron# cat cron.daily
20171114
root@osboxes:/var/spool/anacron# cat cron.monthly
20171114
root@osboxes:/var/spool/anacron# cat cron.weekly
20171114
root@osboxes:/var/spool/anacron#
```

Example anacron checks (cron.daily, cron.monthly, cron.weekly, etc.)

User/Admin/Root

Gui stuff

- Disable automatic logins
- Check for incorrect admins and users
- Change Passwords

Groups

- *cat /etc/group*

Example of Group file (from cyberpatriot ppts)

```
root@ubuntu: /home/cyberpatriot
root@ubuntu:/home/cyberpatriot# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,cyberpatriot
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:cyberpatriot
floppy:x:25:
tape:x:26:
sudo:x:27:cyberpatriot
audio:x:29:pulse
dip:x:30:cyberpatriot
www-data:x:33:
backup:x:34:
test:x:1002:cyberpatriot,guest
cybercamp:x:1003:cyberpatriot
root@ubuntu:/home/cyberpatriot#
```

- To add user to group:
  - sudo adduser [username] [groupname]

Adding users and groups

- To add a group:
  - *sudo addgroup [groupname]*
- To add a user:
  - *sudo adduser [username]*

Delete Users/Groups

- sudo userdel -r username
  - Try users (unless you actually need them): *toor, admin, r00t, adm1n, adm, ftp*
  - Don't just disable because apparently this doesn't get points anymore
- sudo groupdel username

/etc/passwd

LINK: [Format of /etc/passwd]

User Permissions

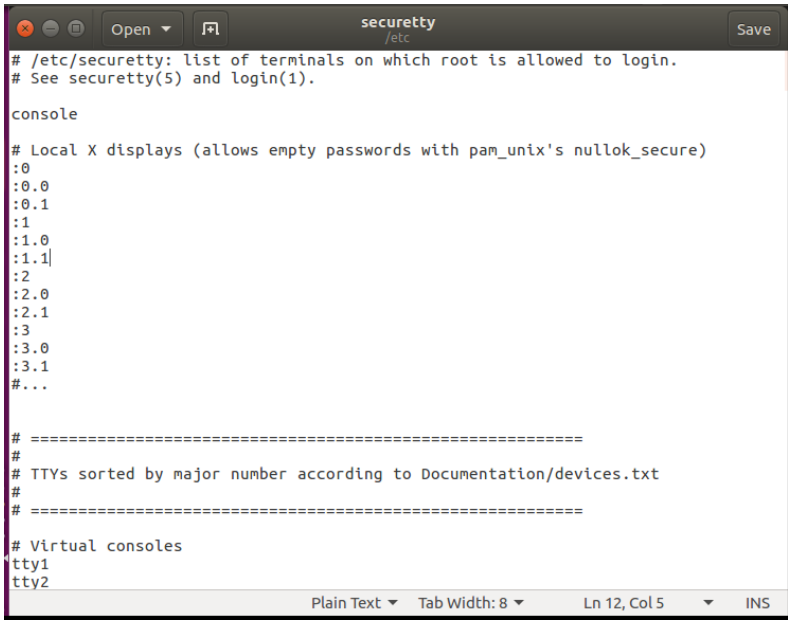
- *awk -F: '($3 == "0") {print}' /etc/passwd*
  - Prints all with uid zero, should only return *root:x:0:0:root:/root:/bin/bash*
- sudo gedit */etc/passwd*
  - Delete uid zeros/check for weird things

/etc/passwd Permissions

- *ls -l /etc/passwd*
- Should be similar to: *-rw-r--r-- 1 root root 2659 Sep 17 01:46 /etc/passwd* (read only to users, root as owner)

Remove Root Logon Access

- *sudo gedit /etc/securetty*
- Keep only console and virtual console and device lines. Usually begin with tty



Example of an okay securetty file

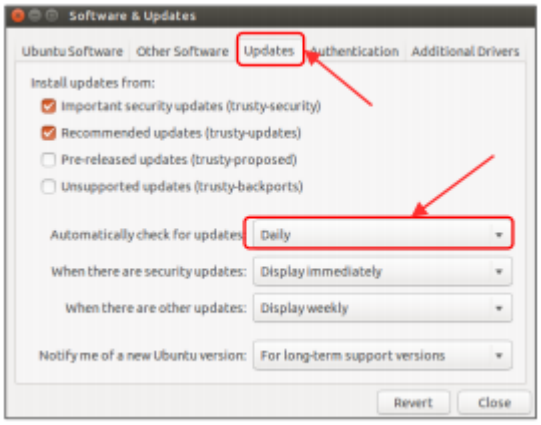
Disable Guest

- *sudo gedit /etc/lightdm/lightdm.conf*
- Add *allow-guest=false*

Steps for Lunch

Updates

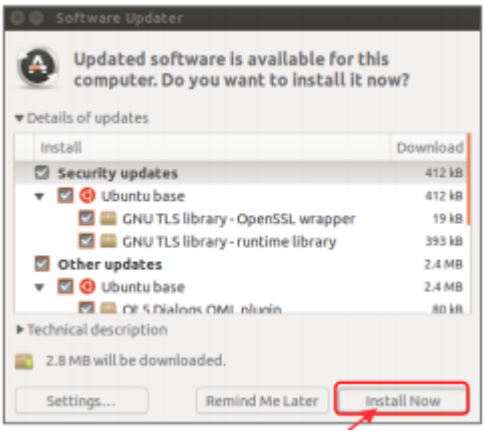
Gui stuff



- Applications->System(?)->Administration->Update(s) Manager
    - Check at least the following
        - Important security updates (precise-security)
        - Recommended updates (precise-updates)
    - Choose the following configurations
        - Automatically check for updates: Daily
        - When there are security updates: Display Immediately
        - When there are other updates: Display Weekly
        - Notify me of a new Ubuntu version: For long-term support versions
- Software Sources
    - Under Ubuntu Software
        - Check at least the following lines
            - Canonical-supported free and open-source software (main)
            - Community-maintained free and open-source software (universe)
        - If there's no other reason (basically in the readme), just check
            - Proprietary drivers for devices (restricted)
            - Software restricted by copyright or legal issues (multiverse)
            - Source Code as well
    - Under Other Software
        - Check at least the following lines
            - Canonical Partners
            - Canonical Partners (Source Code)
        - If there's no other reason, check
            - Independent
            - Independent (Source Code)
    - Under Authentication
        - Only <ftpmaster@ubuntu.com> and <cdimage@ubuntu.com> keys should exist. Removing any others is a judgment call.

Install updates

- In the main software updater menu click install now



- *apt-get update && apt-get upgrade*

Readme Updates

- Update services listed in the readme (google install instructions)

Steps in Terminal

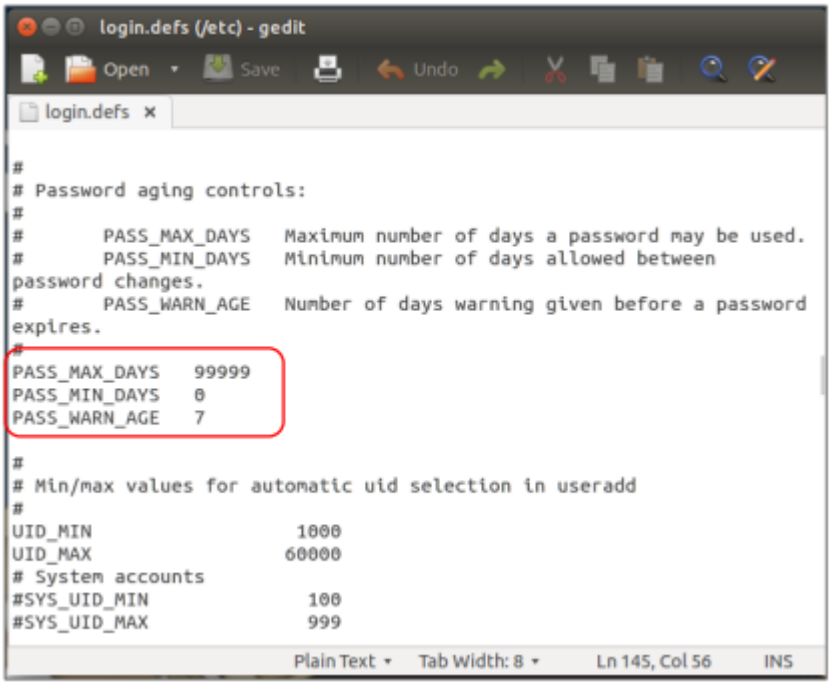
Passwords

Root password

- *sudo passwd root*

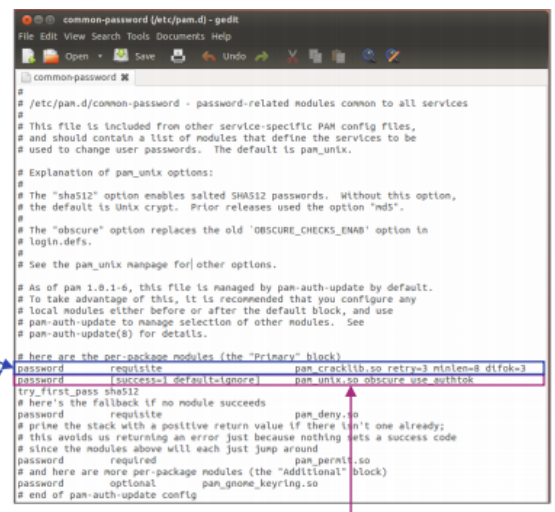
Enforce Complex passwords

- *sudo apt-get install libpam-cracklib --force-yes -y*
- Go to */etc/login.defs*
    - *PASS_MIN_DAYS 10*
    - *PASS_MAX_DAYS 90*
    - *PASS_WARN_AGE 7*



- Password Length: go to /etc/pam.d/common-password
  - Add *remember=5* to the end of the line that has *pam_unix.so* in it.
  - Add *minlen=8* to the end of the line that has *pam_unix.so* in it.
  - Add *ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1* to the end of the line with *pam_cracklib.so* in it

- Type `gedit /etc/pam.d/common-password`
- Lines in the file starting with "#" are comments to help the user understand the file. They do not enforce any policies.
- After making changes, save the file and close it.



1. To enforce password history of 5 :
Add "`remember=5`" to the end of the line that has "pam_unix.so" in it.

2. To enforce Password length of 8:
Add "`minlen=8`" to the end of the line that has "pam_unix.so" in it

3. To enforce password complexity with one of each type of character:*
Add "`ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1`" to the end of the line with "pam_cracklib.so" in it.**
*ucredit = upper case, lcredit=lower case, dcredit = number and ocredit = symbol
**cracklib may need to be installed before enforcing password complexity

Source: http://www.deer-run.com/~hal/sysadmin/pam_cracklib.html

- ACCOUNT LOCKOUT: WARNING WARNING

        - Open */etc/pam.d/common-auth*
        - At the end of the file add:
            - *auth required pam_tally2.so deny=5 onerr=fail unlock_time=1800*

Immutable Bits

- Add:
    - *sudo chattr +i /etc/shadow*
    - *sudo chattr +i /etc/passwd*
- Or to remove:
    - *sudo chattr -i (file path)*

Apt

Manually confirm APT settings

- Run *cd /etc/apt/apt.conf.d/* to traverse to APT's configurations.
    - *sudo gedit 10periodic* to edit the periodic actions file
        - Edit the *APT::Periodic:Update-Package-Lists* line from 0 to 1, then *APT::Periodic::AutoCleanInterval* and *APT::Periodic::Download-Upgradeable-Packages* to 1 as well

Firewall (click here for gui version)

- *sudo apt-get install ufw*
- *sudo ufw enable/disable*
- *sudo ufw status* (is it working?)
- *sudo ufw status verbose* (is it working but fancier)
- *sudo ufw app list* (list of applications available for rules)
- Allow required services (only if required!)
    - *sudo ufw allow ssh*
    - *sudo ufw allow http*
    - *sudo ufw allow https*
    - *sudo ufw allow (whatever you want)*
- Enable logging
    - ufw logging on

Turn off CTRL+ALT+DEL shutdown

- *sudo cat /etc/init/control-alt-delete.conf* and if there's anything starting with exec, esp something that looks bad
- sudo gedit */etc/init/control-alt-delete.conf* and edit it to exec false

Virus/Rootkits/bad stuff

Chkrootkit/Rkhunter

- *sudo apt-get install rkhunter chkrootkit*
- *sudo chkrootkit*
- *sudo rkhunter --check*
- Note there probably will be false positives. A pretty normal false positive (in my experience) is unhide.rb but there are others as well.

Hacking tools/unnecessary

- Check for bad things. They are usually in /etc folder
- Tools that I think are especially loved by cyberpatriot are highlighted. Many in my list have not been used by cyberpatriot as of yet.These may include: Link

User Directories

- *sudo ls -Ra **
  - Lists all files (including hidden ones) recursively through all subdirectories
- Look for:
  - Unauthorized media files
  - Unauthorized tools/hacking tools
  - *.ssh/authorized_keys file*
    - Remove if not supposed to log in remotely

Prohibited Media

- *sudo find / -name "*.extension" -type f* to search the file system for different media where .extension is an extension you are looking for
- *sudo find /home -name "*.extension" -type f* to search user home directories
- *sudo rm -f (full path and filename)* to delete the media

- In root, mp3, wav, wmv, mp4, mov avi and mpeg are all worth searching for
- In home, jpeg, jpg, png, gif, tif, and tiff are all worth searching for
    - Add extra extensions if you think of them

Sudoers

- Check contents of files to make sure only required groups (prob the sudo group) can use sudo and look for anything that says NOPASSWD
    - *sudo cat /etc/sudoers*
    - *cd /etc/sudoers.d && ls -la*
- Check the files in */etc/group* and remove non-admins from sudo and admin groups

Audit Policies

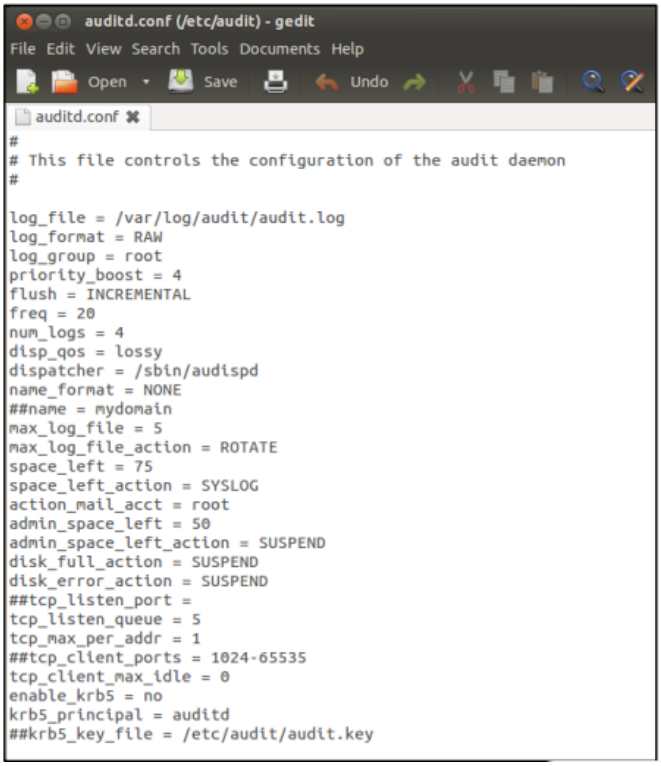
This is specifically from cyberpatriot official powerpoints, though doesn't seem to be in many other checklists.

- Enable Auditing
    - *apt-get install auditd*
    - *auditctl -e 1*
- View and edit policies:
    - *sudo gedit /etc/audit/auditd.conf*

View failed attempts to log in through ssh:

*grep sshd.*Failed /var/log/auth.log | less*

It looks like this:



Check HOSTS File

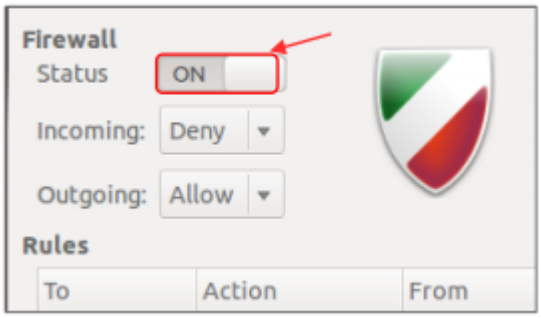
- *sudo gedit /etc/hosts*
- Should only contain the following lines:
    - Lines preceded by #
    - These lines:
        - *127.0.0.1 localhost*
        - *127.0.1.1 ubuntu*
        - *::1 ip6-localhost ip6-loopback*
        - *fe00::0 ip6-localnet*
        - *ff00::0 ip6-mcastprefix*
        - *ff02::1 ip6-allnodes*
        - *ff02::2 ip6-allrouters*

Steps in GUI

Firewall (click here for terminal version)

With Gufw (cyberpatriot powerpoints)

- Download from software center if not already installed
- In GUI go to Firewall Configuration
- Click the Unlock button on the Gufw window → Enact root permissions by authenticating → Turn Firewall Status On
- Deny Incoming and Allow outgoing



- Add rules based on readme and things I say in terminal section

With Firestarter

- *sudo apt-get install firestarter*
- Preferences

Packages/Services/Processes/Ports config

Service Configuration

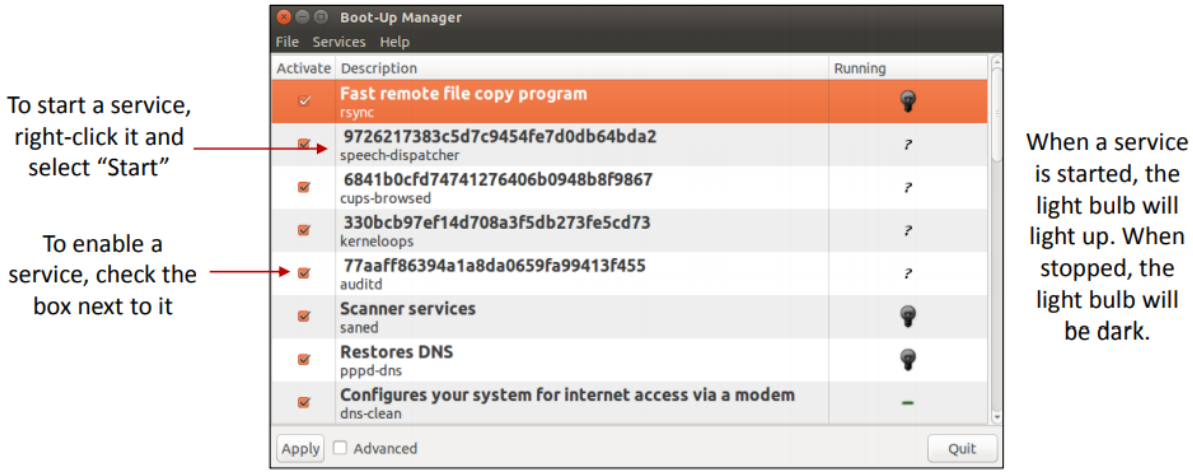
Services are basically just files that run on startup. Located in /etc/init.d

Ensure all services are legitimate.

In GUI

Cyberpatriot says use bum (lol)

- *sudo apt-get install bum*
- *bum* to run the bum :>)



In Terminal

- *service --status-all*
    - Tells what services are running
- *service stop <service name>*
    - Stops a service

What to do

- Check service configuration files for required services. Usually a wrong setting in a config file for sql, apache, etc. will be a point.
- If not required then remove dovecot, postgreys, apache, mysql, and samba (samba is usually listed in readme under required service as 'smb service')

Packages

In GUI:

- *sudo apt-get install synaptic*
- In Gui:
    - *System > Administration > "Synaptic Package Manager"*
    - More info on how to use

In terminal:

- Check the installed packages for "hacking tools," such as password crackers
- *lsof -i -n -P*
- Cross check against the readme
    - Stay safe, if you aren't sure google it and if you are still not sure leave it.
    - Required to keep include (but aren't limited to):
        - DHCP
        - DNS
        - Most of the time Avahi
- Remove bad packages
    - *apt-get purge (name)*
- *dpkg-query -l | grep ftp*
  dpkg-query -l | grep apache
  dpkg-query -l | grep torrent
    - Search for sometimes bad things

Processes

- To view all with permissions:
    - *ls -l /proc*
- *ps -ef | grep nc*
  ps -ef | grep ftp
  ps -ef | grep ssh
    - Search for possibly bad things
- To kill a process, look at the second number on the line outputted by ps -ef describing the process and run sudo kill -KILL [NUM].
    - *sudo service vsftpd stop*
    - *sudo service sshd stop*
    - *sudo service apache2 stop*

Ports

- You have to keep:
    - 80 (http)
    - 443(https)
- Depending on README, may want to remove:
    - 20-21 (ftp)
    - 23 (telnet)
    - 135 (rpc/remote stuff)
    - 411-412 (direct connect peer-to-peer)

- sudo *ss -ln*
- If a port has *127.0.0.1:$port* in its line, that means it's connected to loopback and isn't exposed. Otherwise, there should only be ports which are specified in the readme open (but there probably will be tons more).
- For each open port which should be closed:
    - *sudo lsof -i :$port*
    - Copy the program which is listening on the port. *whereis (name of program)*
    - Copy where the program is (if there is more than one location, just copy the first one). *dpkg -S (location of program)*
    - This shows which package provides the file (If there is no package, that means you can probably delete it with *rm (location of program); killall -9 (name of program)*). *sudo apt-get purge (name of package)*
    - Check to make sure you aren't accidentally removing critical packages before hitting "y".
    - *sudo ss -l* to make sure the port actually closed.

Linux Server Config

Apache

- Config files usually located in */etc/apache/apache2.conf* (in ubuntu)
- Before starting check out var/www folder and see whats inside

In the file apache2.conf

- *TraceEnable off*
    - Leaving on could allow hacker to steal cookie info
- *User apache*
    - Don’t let apache run as root
- *Group apache*
    - Don’t let apache run as root
- *ServerSignature Off*
- *ServerTokens Prod*
- *<Directory /var/www/html>*
  Options -Indexes
  </Directory>
- *Options -FollowSymLinks*
- *Options -Includes*
- *Options -ExecCGI*

MySQL

- Config file in *etc/my.cnf* or */etc/mysql/my.cnf*

SQL commands to use for configuring (Use in mysql)

- ***mysql>*** *drop database test;*
    - drop default table that can be accessed anonymously
- ***mysql>*** *select * from mysql.user where user="";*
    - check for anonymous users without passwords. secure system should not return anything.
- ***mysql>*** *DROP USER "";*
    - remove account from last account
- file permissions should be owned by user and group mysql then only accessible by user mysql and root
    - ***shell>****ls -l /var/lib/mysql*
    - ***shell>****ls -l /usr/bin/my**

vsftpd

dovecot

VNC

Inetd

These probably won't get point but idrc

- Under each home directory:
    - *chmod 600 .rhosts && chmod 600 hosts.equiv*

Find all world writable files

Permissions formatting (from cyberpatriot ppts):

```
cyberpatriot@ubuntu:~$ ls -l hello2.txt
-rw-rw-r-- 1 cyberpatriot cybercamp 57 May 29 09:34 hello.txt
```

Owner (user who created the file)

Size (kb)

File

Links (refers to how many files, folder, and shortcuts link to this file)

Group (user's group when file was created)

Date Modified

- File permissions are the first items noted when using the `ls` command with the `-l` option
- File permissions are split into the 10 fields outlined below
- If any fields are blank, the users in that section cannot do that action with the file

1. Type: if this says "d," the item in question is a directory. A blank means it is a file.

2-4. Owner File Permissions: what the user can do with the file or directory

(Blank 2) Read - r

(Blank 3) Write/modify - w

(Blank 4) Execute – x

5-7. Group File Permissions

(Blank 2) Read - r

(Blank 3) Write/modify - w

(Blank 4) Execute – x

8-10. Other File Permissions

(Blank 2) Read - r

(Blank 3) Write/modify - w

(Blank 4) Execute – x

Example:

File (1.)

Group members can read and write (5-7.)

– r w – r w – r – –

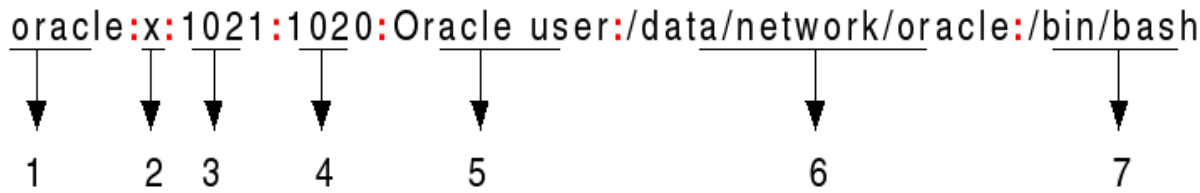
The owner can read and write (2-4.)

Other users can read (8-10.)

- *find / -perm 777 -type f -print*
  - Use chmod for files with odd permissions
- *find PART -xdev -type d -perm -0002 -uid +500 -print*
  - Locates any directories in local partitions which are world-writable and ensure that they are owned by root or another system account.
  - If this command produces any output, investigate why the current owner is not root or another system account.
- *find PARTITION -xdev -type d \(-perm -0002 -a ! -perm -1000 \) -print*
  - Finds any world writable directories without sticky bits
  - To add sticky bits: *chmod +t /dir*

Links from other parts of the checklist:

/etc/passwd format



1. Username: It is used when user logs in. It should be between 1 and 32 characters in length.
2. Password: An x character indicates that encrypted password is stored in /etc/shadow file. Please note that you need to use the passwd command to computes the hash of a password typed at the CLI or to store/update the hash of the password in /etc/shadow file.
3. User ID (UID): Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
4. Group ID (GID): The primary group ID (stored in /etc/group file)
5. User ID Info: The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. This field use by finger command.
6. Home directory: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
7. Command/shell: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. Please note that it does not have to be a shell.

Hacking Tools

- Password Crackers
  - John (short for John the ripper)
    - To remove:
      - *sudo apt-get purge --auto-remove john*
      - *sudo apt-get remove --auto-remove john*
    - Check home directory configuration files to ensure it is gone if you didn't get points
  - Oclhashcat (also known as just hashcat)
    - To remove:

            - *sudo apt-get remove --auto-remove hashcat*
            - *sudo apt-get purge --auto-remove hashcat*
- Network Hacking Tools
    - NMap
        - To remove:
            - *sudo apt-get remove --auto-remove nmap*
            - *sudo apt-get purge --auto-remove nmap*
    - Wireshark
        - To remove:
            - *sudo apt-get remove --purge wireshark*
            - *sudo apt-get autoremove*
    - Netcat
        - To remove:
            - *apt-get purge netcat netcat-openbsd netcat-traditional -y*
            - *find / -name netcat -o -name nc* to look for netcat in the system. Remove anything explicitly named nc or netcat.

- Pentesting tools

    - Metasploit
        - To remove: *sudo rm -rf /opt/metasploit*
    - W3af
        - To remove:
            - *sudo apt-get remove --auto-remove w3af*
            - *sudo apt-get purge --auto-remove w3af*

So I got a bit lazy and didn't want to add everything so here is a list of sometimes bad stuff I found online:

- Web Vulnerability Scanners – Burp Suite, Firebug, AppScan, OWASP Zed, Paros Proxy, Nikto, Grendel-Scan
- Vulnerability Exploitation Tools – Netsparker, sqlmap, Core Impact, WebGoat, BeEF
- Forensic Tools – Helix3 Pro, EnCase, Autopsy
- Port Scanners – Unicornscan, NetScanTools, Angry IP Scanner
- Traffic Monitoring Tools – Nagios, Ntop, Splunk, Ngrep, Argus
- Debuggers – IDA Pro, WinDbg, Immunity Debugger, GDB
- Rootkit Detectors – DumpSec, Tripwire, HijackThis
- Encryption Tools – KeePass, OpenSSL, OpenSSH/PuTTY/SSH, Tor
- Password Crackers – John the Ripper, Aircrack, Hydra, ophcrack

---

Advanced Security

Recommended Permissions on Directories/Files and Definitions

Not all dirs will exist in every image

Root Directory

- /tmp dir should be world writable
  - /tmp is used for temporary storage by all users and some applications
- /var dir
  - /var is used by daemons and other system services to temporarily store dynamic data
- /home dir
  - /home is used to support disk storage needs of users (personal files)
- /dev dir
  - /dev is used as the location of special or device files
- /etc dir
  - /etc contains all system related configuration files in here or in its sub-directories. A "configuration file" is defined as a local file used to control the operation of a program; it must be static and cannot be an executable binary
- /boot dir
  - /boot holds files used in booting the operating system

/var

- /var/tmp dir should be world writable
  - Used for temporary storage by all users and some applications (similar to /tmp in the root directory)
- /var/log dir
  - Used by system services to store log data

/var/log

- /var/log/audit dir
  - Logs of all data by the linux audit framework (auditd) are stored here

/var/log/audit

- /var/log/audit/audit.log file

/home

/dev

- /dev/shm dir
    - /dev/shm is used for passing data between programs
- /dev/kmem and /dev/mem files
    - Provide privileged virtual memory read and write access

/boot

- /boot/grub or /boot/grub2
    - Contains grub.conf , grub.cfg , or menu.lst. These should be Root read and write only
        - contain information on boot settings and passwords for unlocking boot options
        - *chown root:root /boot/grub/grub.cfg* or whatever the config file is
        - *chmod og-rwx /boot/grub/grub.cfg* or whatever the config file is
    -

/etc

- /etc/grub.conf symlink
    - Commonly links to file in /boot/grub or /boot/grub2

For Ubuntu 14.04

Adding sticky bits

Setting the sticky bit on world writable directories prevents users from deleting or renaming files in that directory that are not owned by them.

1. Verify no world writable dirs exist without sticky bits

    a. *df --local -P | awk if (NR!=1) print $6 | xargs -I '{}' find '{}' -xdev -type d \(-perm -0002 -a ! -perm -1000 \) 2>/dev/null*
2. Set sticky bits on all world writable dirs
    a. *df --local -P | awk if (NR!=1) print $6 | xargs -I '{}' find '{}' -xdev -type d -perm -0002 2>/dev/null | chmod a+t*
3. If that didn't work:
    a. *chmod +t /dir*
        i. To each world writable dir

Disable automounting

autofs allows automatic mounting of devices, typically including CD/DVDs and USB drives. With automounting enabled anyone with physical access could attach a USB drive or disc and have its contents available in system even if they lacked permissions to mount it themselves.

1. Remove or comment out start lines in */etc/init/autofs.conf*:
    a. *#start on runlevel [2345]*

Changing ownership of files

Grub:

- chown root:root /boot/grub/grub.cfg or whatever the config file is
- chmod og-rwx /boot/grub/grub.cfg or whatever the config file is

Motd:

- chown root:root /etc/motd
- chmod 644 /etc/motd

Issue:

- chown root:root /etc/issue
- chmod 644 /etc/issue

Issue.net:

- chown root:root /etc/issue.net
- chmod 644 /etc/issue.net

Hosts.allow

- chown root:root /etc/hosts.allow
- chmod 644 /etc/hosts.allow

Hosts.deny

- chown root:root /etc/hosts.deny
- chmod 644 /etc/hosts.deny

/etc/mtab

- chmod 0700 /etc/mtab

/etc/utmp

- chmod 0700 /etc/utmp

Wtmp

- chmod 0700 /var/adm/wtmp
- chmod 0700 /var/log/wtmp

Syslog.pid

- chmod 0700 /etc/syslog.pid
- chmod 0700 /var/run/syslog.pid

/var/log files

- chmod -R g-wx,o-rwx /var/log/*

/etc/ssh/sshd_config

- chown root:root /etc/ssh/sshd_config
- chmod og-rwx /etc/ssh/sshd_config

/etc/passwd-

- chown root:root /etc/passwd-
- chmod 600 /etc/passwd-

/etc/shadow-

- chown root:root /etc/shadow-
- chmod 600 /etc/shadow-

/etc/group-

- chown root:root /etc/group-
- chmod 600 /etc/group-

/etc/gshadow-

- chown root:root /etc/gshadow-
- chmod 600 /etc/gshadow-

/etc/cron and /var/spool/cron

- *chown -R root:root /etc/*cron**
- *chmod -R 600 /etc/*cron**
- *chown -R root:root /var/spool/cron*
- *chmod -R 600 /var/spool/cron*

/etc/Passwd and /etc/Shadow

- *chown root:root /etc/passwd /etc/shadow /etc/group /etc/gshadow*
- *chmod 644 /etc/passwd /etc/group*
- *chmod 400 /etc/shadow /etc/gshadow*

/etc/sysctl.conf

- chmod 700 /etc/sysctl.conf

/etc/inittab

- chmod 700 /etc/inittab

/etc/fstab

- chmod 644 /etc/fstab
- *chown root:root /etc/fstab*

Services

Inetd

inetd is a super-server daemon that provides internet services and passes connections to configured services.

1. Disable chargen

chargen is a network service that responds with 0 to 512 ASCII characters for each connection it receives. This service is intended for debugging and testing purposes. It is recommended that this service be disabled.

   a. Set *disable* = *yes* on all chargen services in */etc/xinetd.conf* and */etc/xinetd.d/**

2. Disable daytime
   a. daytime is a network service that responds with the server's current date and time. This service is intended for debugging and testing purposes. It is recommended that this service be disabled.
   b. Comment out or remove any lines starting with daytime from */etc/inetd.conf* and */etc/inetd.d/**
   c. Set disable = yes on all daytime services in /etc/xinetd.conf and /etc/xinetd.d/*.

RSH Server

The Berkeley rsh-server (rsh, rlogin, rexec) package contains legacy services that exchange credentials in clear-text. Rsh has been replaced with ssh.

1. Comment out or remove any lines starting with shell, login, or exec from /etc/inetd.conf and /etc/inetd.d/*.
2. Set disable = yes on all rsh, rlogin, and rexec services in /etc/xinetd.conf and /etc/xinetd.d/*
3. apt-get remove rsh-client rsh-redone-client

Talk Server

The talk software makes it possible for users to send and receive messages across systems through a terminal session. The talk client (allows initiate of talk sessions) is installed by default.

1. Comment out or remove any lines starting with talk or ntalk from /etc/inetd.conf and /etc/inetd.d/*.
2. Set disable = yes on all talk services in /etc/xinetd.conf and /etc/xinetd.d/*.
3. apt-get remove talk

telnet server

The telnet-server package contains the telnet daemon, which accepts connections from users from other systems via the telnet protocol

- Comment out or remove any lines starting with telnet from /etc/inetd.conf and /etc/inetd.d/*.
- Set disable = yes on all telnet services in /etc/xinetd.conf and /etc/xinetd.d/*
- apt-get remove telnet

tftp server

Trivial File Transfer Protocol (TFTP) is a simple file transfer protocol, typically used to automatically transfer configuration or boot machines from a boot server. The packages tftp and atftp are both used to define and support a TFTP server

- Comment out or remove any lines starting with tftp from /etc/inetd.conf and /etc/inetd.d/*.
- Set disable = yes on all tftp services in /etc/xinetd.conf and /etc/xinetd.d/*

FTP

The File Transfer Protocol (FTP) provides networked computers with the ability to transfer files.

- Remove or comment out start lines in /etc/init/vsftpd.conf: #start on runlevel [2345] or net-device-up IFACE!=lo

Apache2

HTTP or web servers provide the ability to host web site content

- *update-rc.d apache2 disable*

Samba

The Samba daemon allows system administrators to configure their Linux systems to share file systems and directories with Windows desktops. Samba will advertise the file systems and directories via the Small Message Block (SMB) protocol. Windows desktop users will be able to mount these directories and file systems as letter drives on their systems

- Remove or comment out start lines in /etc/init/smbd.conf:
- *#start on (local-filesystems and net-device-up)*

SNMP

The Simple Network Management Protocol (SNMP) server is used to listen for SNMP commands from an SNMP management system, execute the commands or collect the information and then send results back to the requesting system.

- *update-rc.d snmpd disable*

Rsync

The rsyncd service can be used to synchronize files between systems over network links. The rsyncd service presents a security risk as it uses unencrypted protocols for communication.

- Edit the /etc/default/rsync file and set RSYNC_ENABLE to false:
- *RSYNC_ENABLE=false*

NIS Server

The Network Information Service (NIS) (formally known as Yellow Pages) is a client-server directory service protocol for distributing system configuration files. The NIS server is a collection of programs that allow for the distribution of configuration files.

1. Remove or comment out start lines in /etc/init/ypserv.conf:
    a. #start on (started portmap ON_BOOT=
    b. # or (started portmap ON_BOOT=y
    c. # and ((filesystem and static-network-up) or failsafe-boot)))
2. *apt-get remove nis*

LDAP Client

The Lightweight Directory Access Protocol (LDAP) was introduced as a replacement for NIS/YP. It is a service that provides a method for looking up information from a central database.

- apt-get remove ldap-utils

IPV4

Edit /etc/sysctl.conf to include the following:

```
# IP Spoofing protection
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

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

# Ignore send redirects
net.ipv4.conf.all.send_redirects = 0
```

```
net.ipv4.conf.default.send_redirects = 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 = 5

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

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

# Ignore Directed pings
net.ipv4.icmp_echo_ignore_all = 1
```

SSH

SSH is a secure, encrypted replacement for common login services such as telnet, ftp, rlogin, rsh, and rcp. It is strongly recommended that sites abandon older clear-text login protocols and use SSH to prevent session hijacking and sniffing of sensitive data off the network.

In: /etc/ssh/sshd_config

1. Protocol 2
2. LogLevel INFO
3. X11Forwarding no
4. IgnoreRhosts yes
5. HostbasedAuthentication no
6. PermitRootLogin no
7. PermitEmptyPasswords no
8. LoginGraceTime 60
9. UsePAM yes
10. StrictModes yes

Ensure System accounts are non-login

- Run the following command:
- *egrep -v "^\+" /etc/passwd | awk -F: '($1!="root" && $1!="sync" && $1!="shutdown" && $1!="halt" && $3<1000 && $7!="/usr/sbin/nologin" && $7!="/bin/false") {print}'*

- Set the shell for any accounts returned by the audit script to /usr/sbin/nologin:
- *usermod -s /usr/sbin/nologin <user>*

Ensure group ID of root is GID 0

1. *usermod -g 0 root*

Ensure default user unmask

- Edit the /etc/bash.bashrc and /etc/profile files (and the appropriate files for any other shell supported on your system) and add or edit any umask parameters as follows:
- *umask 027*

Restrict access to su

- Add the following line to the /etc/pam.d/su file:
    - auth required pam_wheel.so use_uid
- In /etc/group
    - wheel:x:10:root,<user list>

Ensure no world writable files exist

- df --local -P | awk {'if (NR!=1) print $6'} | xargs -I '{}' find '{}' -xdev -type f -perm -0002
- For each incorrect one, chmod o-w <filename>

Ensure no unowned files or directories exist

- *df --local -P | awk {'if (NR!=1) print $6'} | xargs -I '{}' find '{}' -xdev -nouser*
- Locate files that are owned by users or groups not listed in the system configuration files, and reset the ownership of these files to some active user on the system as appropriate.

Ensure no ungrouped files or directories exist

- *df --local -P | awk {'if (NR!=1) print $6'} | xargs -I '{}' find '{}' -xdev -nogroup*
- Locate files that are owned by users or groups not listed in the system configuration files, and reset the ownership of these files to some active user on the system as appropriate

Audit SUID executables

- df --local -P | awk {'if (NR!=1) print $6'} | xargs -I '{}' find '{}' -xdev -type f -perm -4000
- Ensure that no rogue SUID programs have been introduced into the system. Review the files returned by the action in the Audit section and confirm the integrity of these binaries.

Audit SGID executables

- df --local -P | awk {'if (NR!=1) print $6'} | xargs -I '{}' find '{}' -xdev -type f -perm -2000
- Ensure that no rogue SGID programs have been introduced into the system. Review the files returned by the action in the Audit section and confirm the integrity of these binaries

Ensure no legacy "+" entries exist in /etc/passwd

- grep '^+:' /etc/passwd
- Remove any legacy '+' entries from /etc/passwd if they exist.

Ensure no legacy "+" entries exist in /etc/shadow

- grep '^+:' /etc/shadow
- Remove any legacy '+' entries from /etc/shadow if they exist.

Ensure no legacy "+" entries exist in /etc/group

- grep '^+:' /etc/group
- Remove any legacy '+' entries from /etc/group if they exist

Ensure shadow group is empty

The shadow group allows system programs which require access the ability to read the /etc/shadow file. No users should be assigned to the shadow group

- grep ^shadow:[^:]*:[^:]*:[^:]+ /etc/group
- awk -F: '($4 == "<shadow-gid>") { print }' /etc/passwd
- Remove all users from the shadow group, and change the primary group of any users with shadow as their primary group

Secure shared memory

- gedit /etc/fstab
- Add to the bottom of the file:
    - *tmpfs /run/shm tmpfs defaults,noexec,nosuid 0 0*

Prevent IP spoofing

- gedit /etc/host.conf
- Should look like this:

The "order" line is only used by old versions of the C library.
order hosts,bind
multi on

- Change to this:

The "order" line is only used by old versions of the C library.
order bind,hosts
nospoof on