Linux + Windows

CCDC Fall Bootcamp Week 2

Sign-in:
jessh.zip/2026ccdcfallweek2

Weekly Schedule

Date	CCDC (1PM-4PM)
~~Aug 29~~	~~Intro, Business, and Networking~~
Sep 5	Securing Linux and Windows
Sep 12	Common services + Q&A
Sep 19-20	CCDC Tryouts (1-5 PM)

whoami

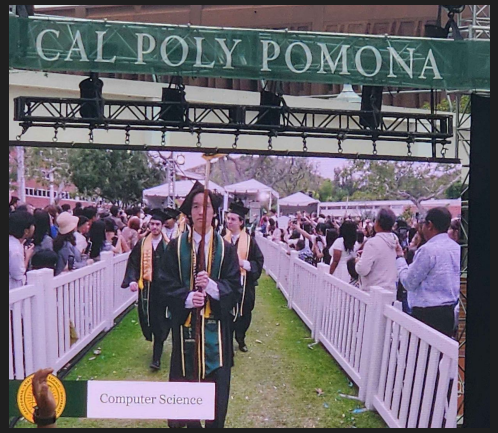
Adam Cheung | @adumb

IST Sysadmin Intern @ Esri

CCDC Co-Captain, Linux Lead 2026-2027
CCDC Linux Database 2025-2026

BS CS, Cyber Minor; MSIS 1st year
Sec+, Network+

slay da spire



whoami

Anson Ng | @2 acc

Cybersecurity Intern @ Varda

CCDC Linux Lead 2025-2026
Taco Bell IT 2024-2025
NCAE Cyber Games Nationals 2024-2025

senior citizen student (will be doing MSIS @ cpp)

monkey brain

whoami

Anthony Rivera | @4nthvny

Security Analyst Intern @ SoFi

CCDC Threat Hunter 2025-2027
CPTC Captain (2026-2027)
CPTC Windows (2025-2026)

Grad student (InfoSec)

Linux

Fundamentals, Administration, and Security

01 Linux Fundamentals

02 Linux Administration

03 Linux Security and Networking

04 Firewall

1.1

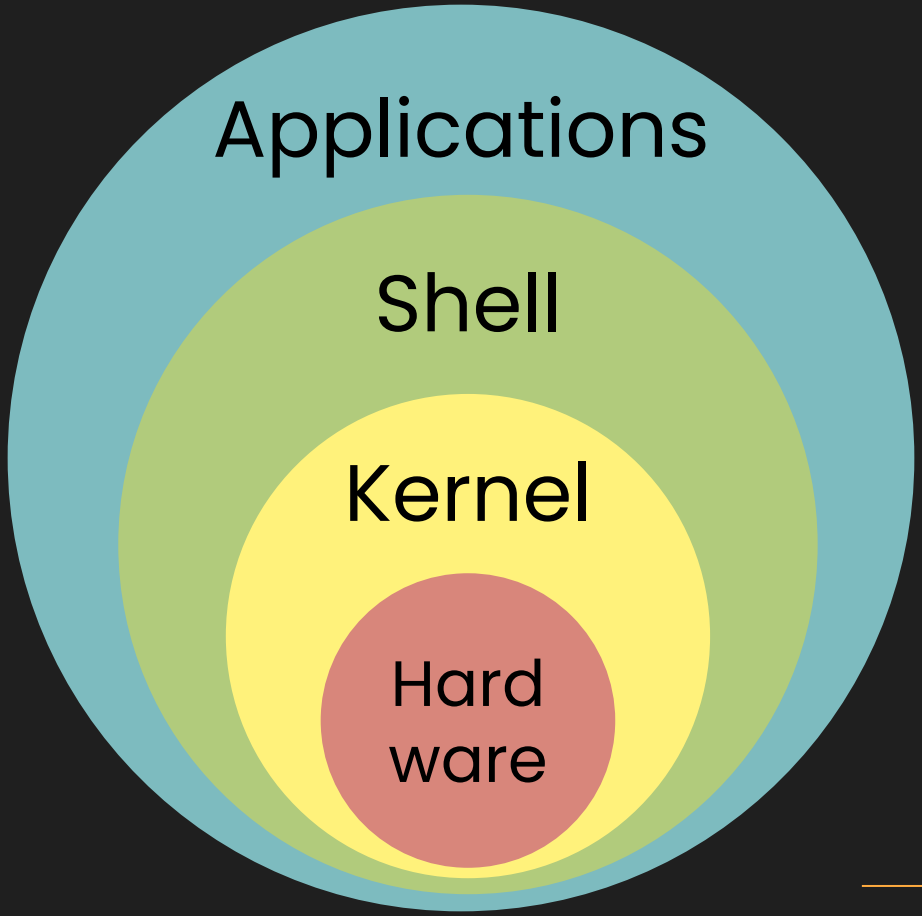
Linux Fundamentals

What is Linux

- **Not** an operating system
  - Distributions or flavors of Linux
- Free & open-source **kernel**
- Built on **Unix** (unix-like)

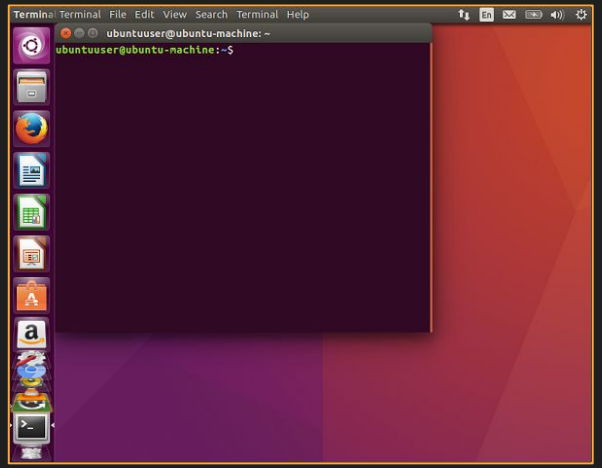





Applications
Shell
Kernel
Hard
ware

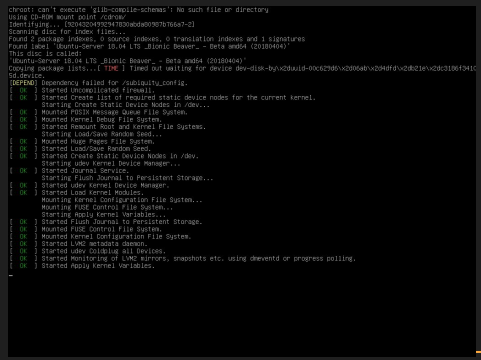
Terminals

Terminal Emulator



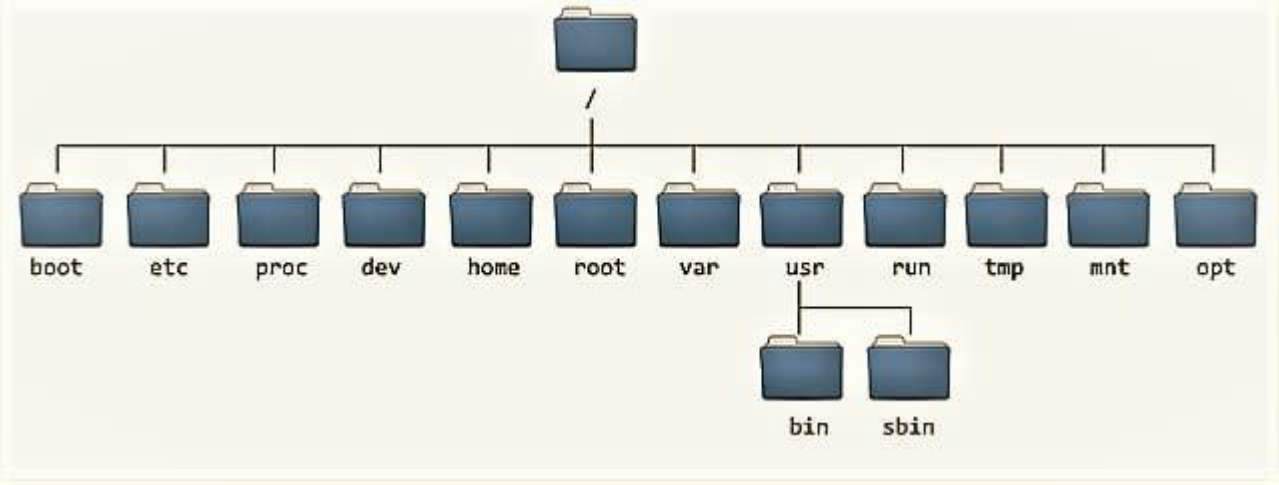
"Desktop" Version

"Server" Version



Linux File System

- Everything can be categorized into a File, Directory, or Link



Paths and Directories

Absolute Path

Starts with /

Relative Path

Starts with pwd (print working directory)

Current Directory

Use .

Previous Directory

Use ..

Examples

cd /home/user/Desktop/	cd ..
cd /var/www/html/	cd ~/Downloads
/etc/ssh/sshd_config	var/www/html/

root vs /root vs /

- root user (uid 0) = admin
- root (/) directory = start of file system
- root user's home = /root

Shell and Syntax

command -options arguments

- EXAMPLE: ls
- EXAMPLE: cd /home/user1
- EXAMPLE: ls -la user1/Downloads
- EXAMPLE: iptables -L --line-numbers

Listing Files

ls (list) → **ls [flags] [filepath]**

Flags:

-l (more detailed view)

-a (show all files and directories)

```
jamie@debian:~$ ls -l
total 19208
-rwxr-xr-x 1 root  root   4703728 Dec 17 07:01 Battle.net-Setup.exe
drwxr-xr-x 3 jamie jamie     4096 Nov  5 03:30 Desktop
drwxr-xr-x 2 jamie jamie     4096 Jun  5  2018 Documents
drwxr-xr-x 3 jamie jamie     4096 Dec 17 06:49 Downloads
-rw-r--r-- 1 jamie jamie   179765 Dec 17 07:01 Linux_for_beginners.pdf
-rw-r--r-- 1 jamie jamie   458980 Dec 17 06:59 metamorphose2_0.8.2-1_all.deb
drwxr-xr-x 2 jamie jamie     4096 Apr 30  2018 Music
-rw-r--r-- 1 jamie jamie     1520 Dec 17 07:01 Neofetch
-rw-r--r-- 1 root  root  13902480 Dec 17 07:01 pdfsam_3.3.6-1_all.deb
-rw------- 1 root  root    375728 Dec 17 07:01 PDFsam_merge.pdf
drwxr-xr-x 2 jamie jamie     4096 Apr 30  2018 Pictures
drwxr-xr-x 2 jamie jamie     4096 Apr 30  2018 Public
drwxr-xr-x 2 jamie jamie     4096 Apr 30  2018 Templates
drwxr-xr-x 2 jamie jamie     4096 Apr 30  2018 Videos
```

Creating Files

touch → **touch [flags] [filename/filepath]**

[favorite text editor] [filename]

nano, vim, vi

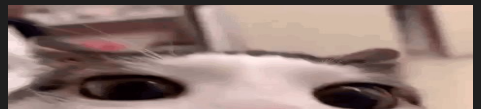
...many more for later...

Commands You Would Find with a Right-Click

cp (copy) - **cp [source] [destination]**

mv (move) - **mv [source] [destination]**

cat (concatenate) - **cat [filename]**



rm (remove) - **rm [flags] [filepath/filename]**

Flags :
-f or --force, do not prompt, only remove
-r or --recursive, remove content and subdirectories

mkdir (make directory) - **mkdir [flags] [path/directory_name]**

Flags:
-p or --parent

1.2

Linux Administration

Environment Variables

$PATH

An environment variable for the directory search order for commands you call

```
echo $PATH
```

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin

env

Command to list all currently set environment variables

Aliases

```
ccdc@ubuntu22:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && e
cho terminal || echo error)" "$(history|tail -n1|sed -e '\''
s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
alias ccdc='echo ceeceeDc'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
ccdc@ubuntu22:~$ alias ccdc='echo ceeceeDc'
ccdc@ubuntu22:~$ ccdc
ceeceeDc
ccdc@ubuntu22:~$
```

Strings n' Stuff

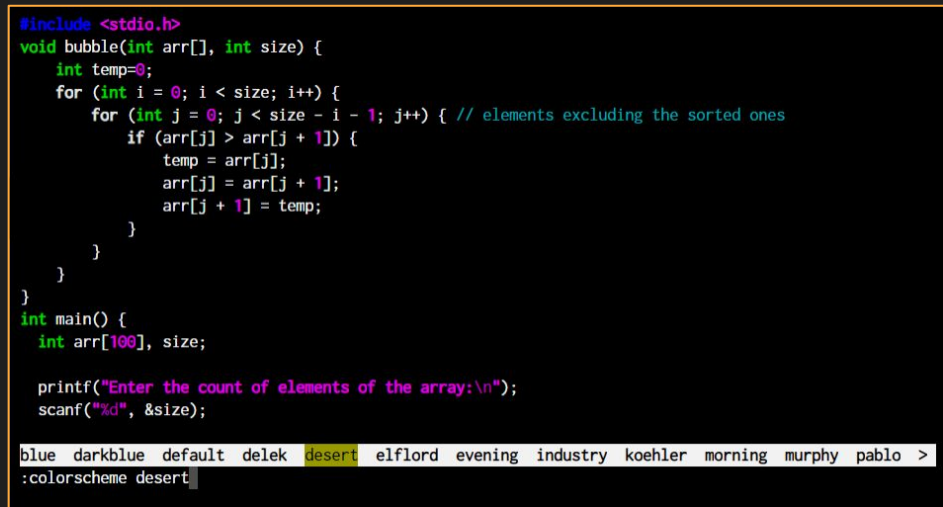
NANO

- **nano** <filename>
- installed by default mostly
- very basic
- CTRL+X to exit "Y" to save as same name


```
  GNU nano 2.0.9            File: txt_files/testfile                  Modified

Learn how to use nano to boost your terminal confidence!
Edit config files like a pro!
Make easy to-do lists and notes in a text-only format!
Do it via SSH from a smartphone or other computer!

# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
# / was on /dev/sdb1 during installation
                              [ Read 17 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

VIM

- **vim** <filename>
- sometimes not installed by default
- extremely customizable
- :wq to close and save file
- 5 modes
- can run commands in the editor
- vimtutor to get started



SED

sed <script> <filename>

sed -i 's/pattern/replace/g' file.txt

Good for scripting out file changes

Can use Regex for pattern matches

```
[Jul 14, 2024 - 19:24:46 (PDT)] exegol-attack bootcamps # cat file
the quick brown fox jumps over the lazy dog
[Jul 14, 2024 - 19:24:47 (PDT)] exegol-attack bootcamps # sed -i 's/fox/dog/' file; cat file
the quick brown dog jumps over the lazy dog
[Jul 14, 2024 - 19:24:50 (PDT)] exegol-attack bootcamps # sed -i 's/dog/wolf/' file; cat file
the quick brown wolf jumps over the lazy dog
[Jul 14, 2024 - 19:25:04 (PDT)] exegol-attack bootcamps # sed -i 's/the/a/g' file; cat file
a quick brown wolf jumps over a lazy dog
```

AWK

- **awk** <options> <script> <input>
- awk -F ',' '{print $2}' data.csv
- Good for filtering specific fields
- Similar to running a for loop on each line

```
ubuntu-user@HP:~$ awk '{print $1, $2}' employees.txt
John Manager
Stacy CEO
Duke Manager
Kate CEO
Sunil Manager
Duke CEO
Kate Manager
Sunil CEO
ubuntu-user@HP:~$
```

GREP

grep <options> "string to search for" <file>

cat <file> | grep "string"

Good for finding strings in a file

Lots of options

```
zunaid@Ubuntu:~$ grep "error" sample.txt
The word "error" appears here as lowercase.
This line mentions terror but not the actual error.
zunaid@Ubuntu:~$ grep -w "error" sample.txt
The word "error" appears here as lowercase.
This line mentions terror but not the actual error.
zunaid@Ubuntu:~$
```

Moving Strings Around

- **STDIN (Standard Input Stream)** - takes strings as input
  - **"<" –** Redirects STDIN

```
user@                    :~$ cat < example1.txt
Goodluck at tryouts!
user@                    :~$
```

- **Pipes** - output of one command used for another.
  - **"|"** → not an L

```
user@                    :~$ cat favoriteThings.txt | sort
Buttered Chicken
Cheeseburger
Computers
Food
Food
Iphone
Penguins
```

- **STDOUT (Standard Input Stream)** - output strings from command
  - **">" –** Redirects STDOUT

```
user@                    :~$ echo "Hello from Texas" > example1.txt
user@                    :~$ cat example1.txt
Hello from Texas
user@                    :~$
```

Command line bash scripting

- bash scripts can be run using:

```
./file_name       Ex: ./my_script.sh

bash file_name    Ex: bash my_script.sh
```

NOTE: ensure your file has the executable (x) permission!

- With a bash shell, you can use bash code in the command line!

```
for line in $(cat poem);
do echo $line; done
```

Combining all the tools

Discuss among yourselves!

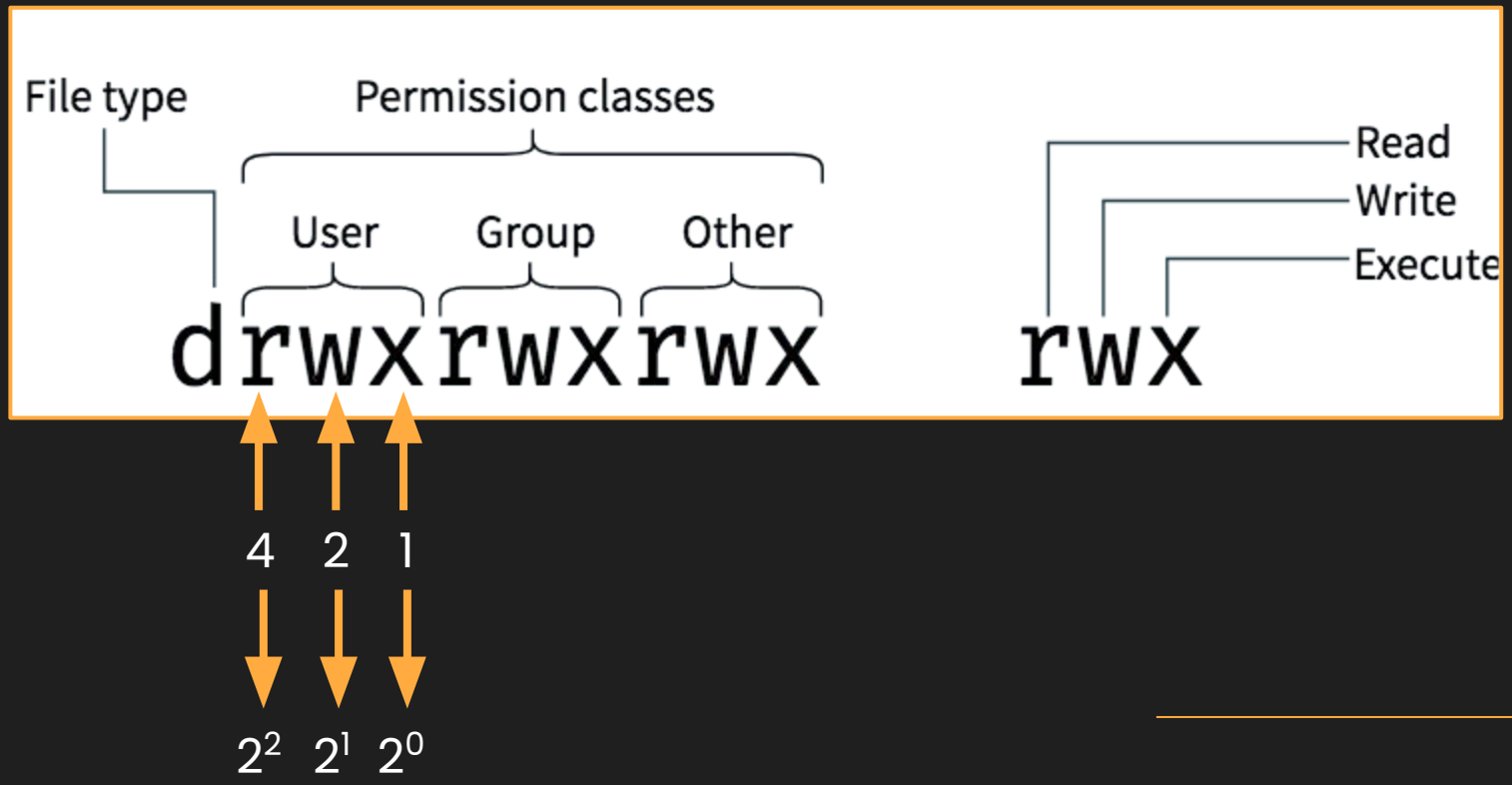
- What does the following command do?

```
cat /etc/ssh/sshd_config | grep "^PermitRootLogin" | awk
'{print $2}'
```

- TIP: Separate the command into segments split by pipes → | and process each segment one by one!

File Permissions

Linux File Permissions



Convert to octal

rwxr--r--

Convert to rwx

644

Convert to octal

r-x-w---x

Convert to rwx

755

Changing File Permissions

chmod to change permissions

chown to change file owner
Ex:
chown user1:group1 <file>
chown root:root notes.txt

CHMOD is used to change permissions of a file.

PERMISSION			COMMAND
U	G	W	
rwx	rwx	rwx	chmod 777 filename
rwx	rwx	r-x	chmod 775 filename
rwx	r-x	r-x	chmod 755 filename
rw-	rw-	r--	chmod 664 filename
rw-	r--	r--	chmod 644 filename
User	Group	World	

r = Readable
w = Writable
x = Executable
- = None

```
-bash-5.0$ chmod 777 file1
-bash-5.0$ chmod a+rwx file2
-bash-5.0$ ls -l
total 0
-rwxrwxrwx 1 nigerald nigerald 0 Jul 19 01:45 file1
-rwxrwxrwx 1 nigerald nigerald 0 Jul 19 01:45 file2
-bash-5.0$ chmod 744 file1
-bash-5.0$ chmod go+r file2
-bash-5.0$ ls -l
total 0
-rwxr--r-- 1 nigerald nigerald 0 Jul 19 01:45 file1
-rwxrwxrwx 1 nigerald nigerald 0 Jul 19 01:45 file2
```

Immutability

Make file immutable

chattr +i <file>

Check for immutable bit

lsattr <file>

Remove immutable bit

chattr -i <file>

this is an i

I am ~~g~~root

sudo <command>

✨**sudo <command>**✨

sudo -i

sudo su

su <user>

su root

su -

Adding Users

adduser

wrapper for useradd

less clunky

prompts for password

useradd

much less efficient

doesn't create home directories

manually set password

Managing Users

Group Management

not group policy

groups users together

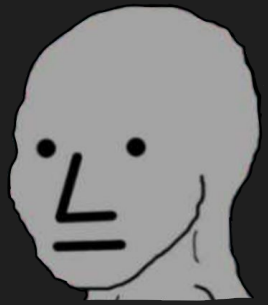
✨**usermod**✨

✨**id**✨

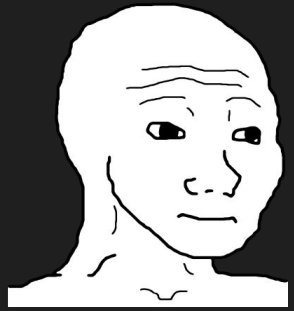
User IDs and Group IDs

root = 0

services < 1000



users > 999



Password Management

passwd

- passwd (changes for current user)
- passwd user2 (changes for user2)

chpasswd

- Can be used for automation
- echo "user2:password" | chpasswd
- Alternatively use the format above and finish with Ctrl + D

```
root@ubuntu22:/home/ccdc# chpasswd
user2:secure_password
root@ubuntu22:/home/ccdc# passwd user2
New password:
Retype new password:
passwd: password updated successfully
root@ubuntu22:/home/ccdc# echo "user2:cool" | chpasswd
```

Different Distros

Debian-based

apt update

apt upgrade

apt install

apt purge/remove

RHEL-based

yum update

yum upgrade

yum install

yum remove/erase

Other

suffering

apk

pacman

solaris

Services

Process running in the background managed by the system

systemctl

On most modern distros

Simple

systemctl start sshd

:)

service

Usually works if systemctl doesn't

Simple

service sshd status

:)

rc.d / init.d

Systems without systemd

Pain. Location may vary

/etc/init.d/sshd start

:(

Services

Process running in the background managed by the system

systemctl <action> <service>

- systemctl status httpd
- systemctl start ssh
- systemctl enable nginx

service <service> <action>

- service httpd status
- service ssh start
- service nginx enable

status, start, enable, stop, disable, restart, mask

Processes

Program running on the computer

ps - List Processes for this terminal

- Often use "ps aux" instead for ALL processes

kill -9 <Process ID (PID)> - Kill process by ID

pgrep - Find PID from process name

Get Some Help

- Man pages
- Find and grep command
- --help parameter
- less or more
- head or tail
- tmux

Tmux Cheatsheet

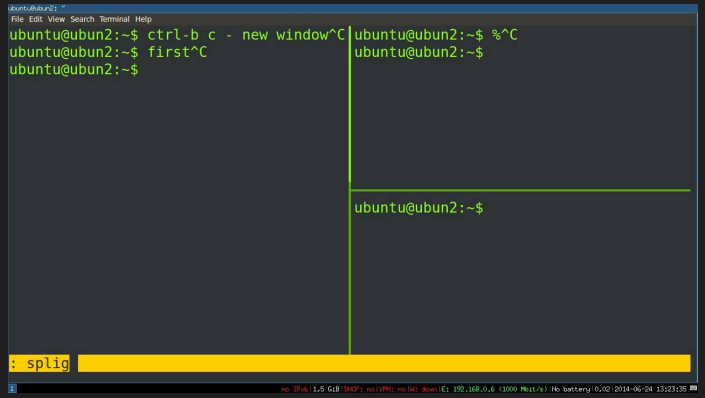
Prefix: ctrl + b

Windows

- New Window: **prefix** + c
- Switch between Windows: **prefix** + [number] OR (p)revious OR (n)ext
- Delete Window: **prefix** + &

Panes

- Split Horizontally: **prefix** + "
- Split Vertically: **prefix** + %
- Switch between panes: **prefix** + [arrow key]

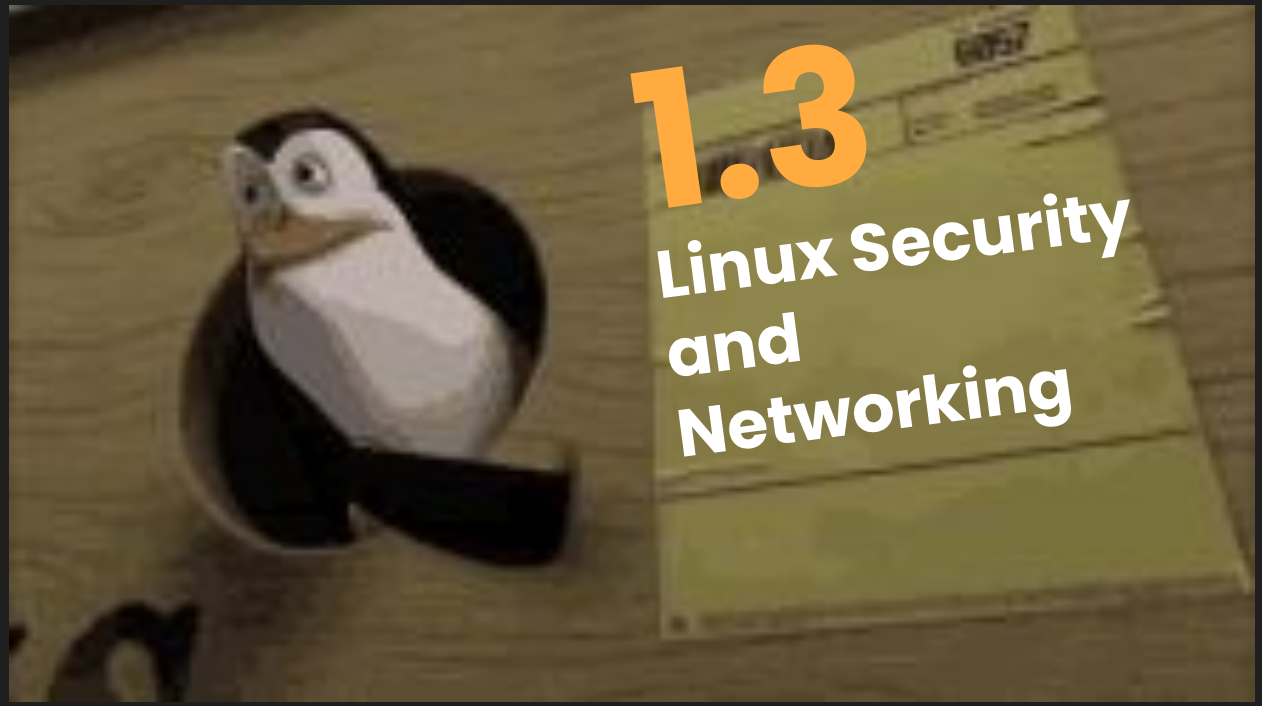


Other

- New Session: tmux
- Detach: **prefix** + d
- Reattach: tmux + a
- Fullscreen: **prefix** + z

Linux Shortcuts

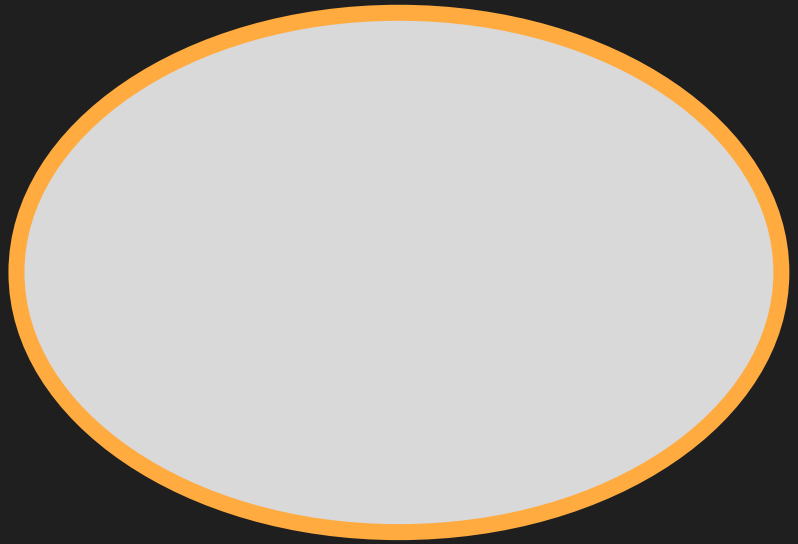
- (ctrl + a) - go to the start of the command line
- (ctrl + e) - go to the end of the command line
- (ctrl + l) - clear screen
- (ctrl + left arrow) - go to the left most word
- (ctrl + right arrow) - go to the right most word
- (ctrl + r) - reverse search history based on typed letters
- (ctrl + p) - move back a command in history list
- (ctrl + n) - move forward a command in history list
- (ctrl + g) - exit reverse search mode
- (ctrl + u) - cut all text to left of cursor
- (ctrl + k) - cut all text to right of cursor
- (ctrl + w) - cut the word immediately preceding the cursor
- (Alt + d) - delete word after cursor
- (ctrl + y) - paste from clipboard (after cut)
- (!!) - repeat the last command
- (!$),(Alt + .) - insert the previous commands last argument


1.3
Linux Security
and
Networking

The Holy Square of User Management

/etc/passwd

/etc/group



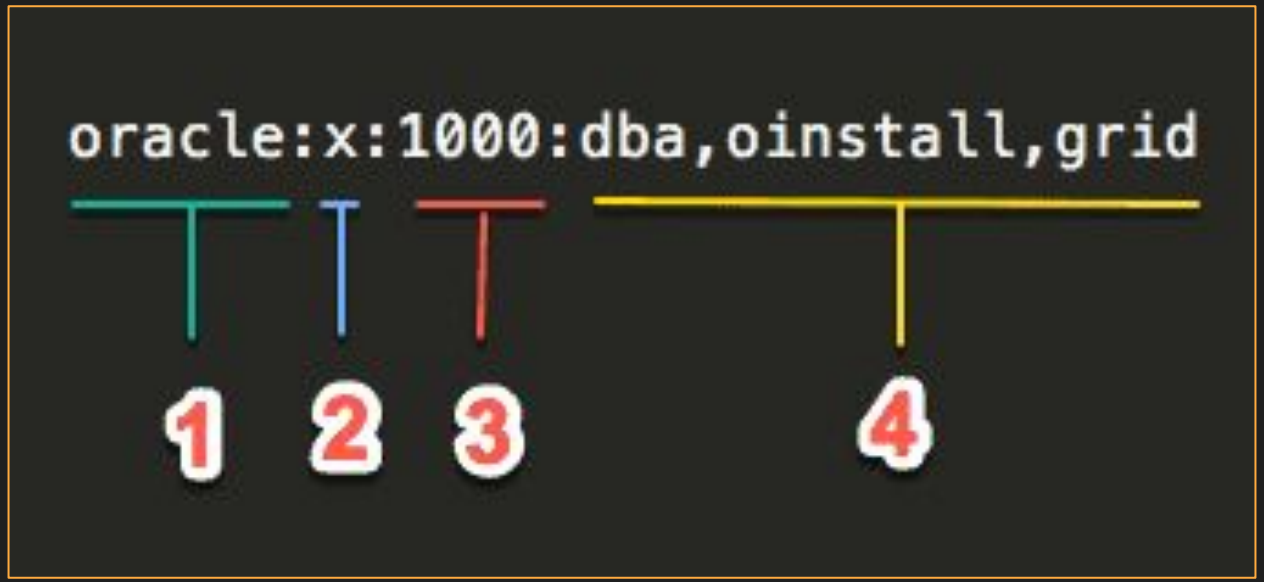
/etc/shadow

/etc/sudoers

/etc/passwd

```
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
```

/etc/group



1: username 2: password 3: GID 4: Members of Group

/etc/shadow

1: username

2: password hash
different algorithms

3: last changed time (epoch)
4: minimum days between password changes
5: maximum days password is valid

/etc/sudoers

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

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

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

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

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

What is PAM?

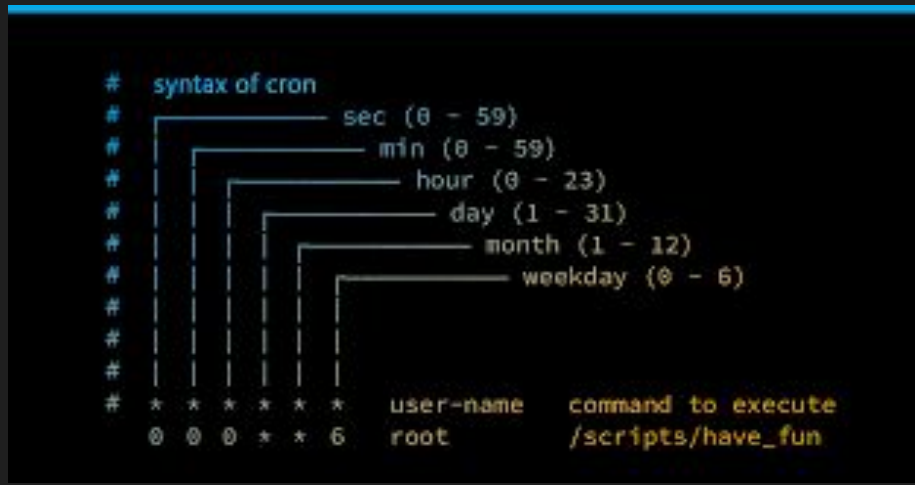
- pluggable authentication module
- manages authentication
- common-auth (Debian)
  system-auth and password-auth (RHEL)

common-auth

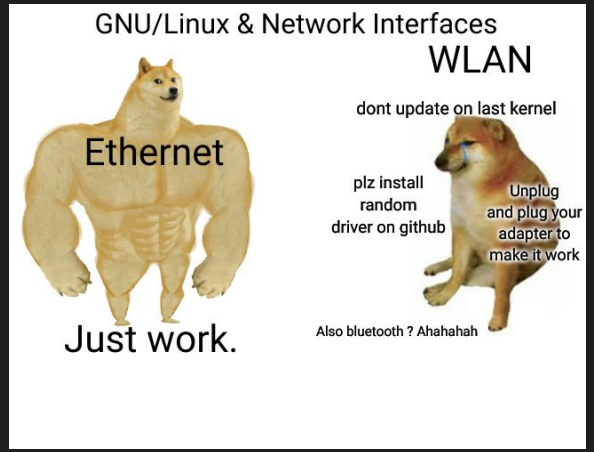
```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    [success=1 default=1]           pam_unix.so nullok
# here's the fallback if no module succeeds
auth    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth    optional                        pam_cap.so
# end of pam-auth-update config
```

cronjobs

- Tasks that run at regular intervals
  - Every day/week/month at certain times
- Can be good or bad
  - Always check for weird scripts!
  - crontab -e
  - /etc/cron.d
- If you have no use for it...
  - Kill it!
  - systemctl stop, disable, mask



Linux Networking



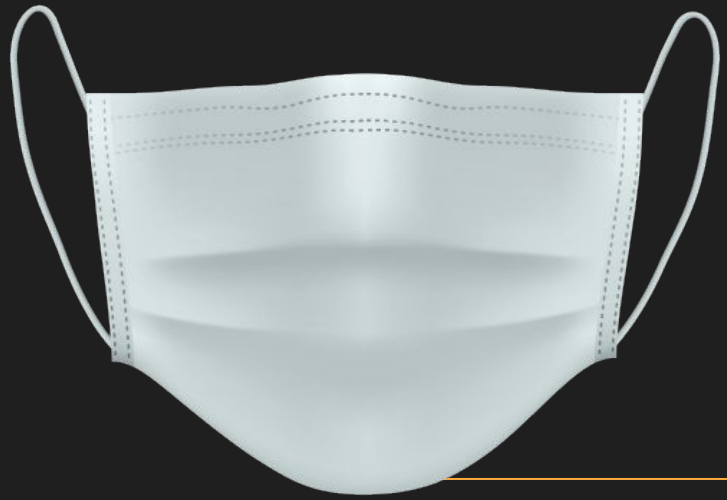
NetworkManager

(And other similar services)

NetworkManager, systemd-networkd, netplan

Manages network connections

Make sure it's running (systemctl)



Network Connections

Get NIC info with "ip a" or "ifconfig"
View connections via netstat (may need net-tools) or ss

```
root@ubuntu20:/home/ccdc# ss -tulpn
Netid     State      Recv-Q     Send-Q               Local Address:Port          Peer Address:Port     Process
udp       UNCONN     0          0                    127.0.0.53%lo:53                 0.0.0.0:*         users:(("systemd-resolve",pid=766,fd=12))
udp       UNCONN     0          0             192.168.30.132%ens33:68                 0.0.0.0:*         users:(("systemd-network",pid=764,fd=19))
tcp       LISTEN     0          4096                 127.0.0.53%lo:53                 0.0.0.0:*         users:(("systemd-resolve",pid=766,fd=13))
tcp       LISTEN     0          128                        0.0.0.0:22                 0.0.0.0:*         users:(("sshd",pid=1699,fd=3))
tcp       LISTEN     0          5                          0.0.0.0:8080               0.0.0.0:*         users:(("python3",pid=2214,fd=3))
tcp       LISTEN     0          128                           [::]:22                    [::]:*         users:(("sshd",pid=1699,fd=4))
tcp       LISTEN     0          511                              *:80                       *:*         users:(("apache2",pid=842,fd=4),("apache2",pid=841,fd=4),("apache2",pid=838,fd=4))
root@ubuntu20:/home/ccdc# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      766/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1699/sshd: /usr/sbi
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      2214/python3
tcp6       0      0 :::22                   :::*                    LISTEN      1699/sshd: /usr/sbi
tcp6       0      0 :::80                   :::*                    LISTEN      838/apache2
udp        0      0 127.0.0.53:53           0.0.0.0:*                           766/systemd-resolve
udp        0      0 192.168.30.132:68       0.0.0.0:*                           764/systemd-network
root@ubuntu20:/home/ccdc#
```

NMAP

Quickly identify open ports on the network

```
┌──(root㉿kali)-[~]
└─# [07/25/24 7:37:13] nmap 172.16.127.31
Starting Nmap 7.94 ( https://nmap.org ) at 2024-07-25 19:37 PDT
Nmap scan report for 172.16.127.31
Host is up (0.00029s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:50:56:97:F2:30 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.14 seconds
```

1.4

Firewalls (but linux)

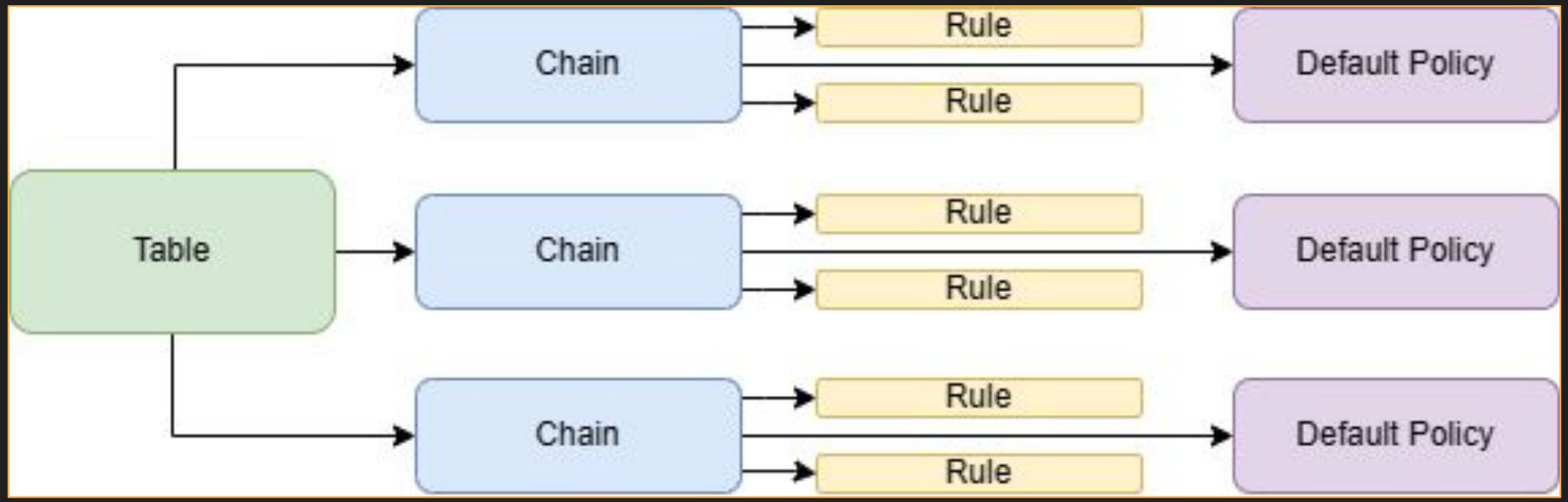
Firewalls

More ports = larger attack surface

Firewalls should operate with the **Implicit Deny** principle

Block by default, allow by exception

iptables - Overview



iptables - Filter Table

3 Chains:

- INPUT
- OUTPUT
- FORWARD

Default Policy:

iptables -P INPUT DROP
iptables -P OUTPUT DROP*
iptables -P FORWARD DROP

Flush Rules:

iptables -F

List Rules:

iptables -L --line-numbers

iptables - Allowing HTTP Example

Allow incoming on 80

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Drop incoming packets if they do not match a rule

```
iptables -P INPUT DROP
```

Allow outgoing responsive connections

```
iptables -A OUTPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Drop outgoing packets if they do not match a rule

```
iptables -P OUTPUT DROP
```

Wildin with Windows

Table of Contents

01
Navigating

GUI vs CLI

- Easier to read information
- Search menu
- Beginner friendly

- Relatively simple
- Quick
- Very extensive for a scripting language

C:\>

- Outdated but consistent Across OS Versions

Powershell😍

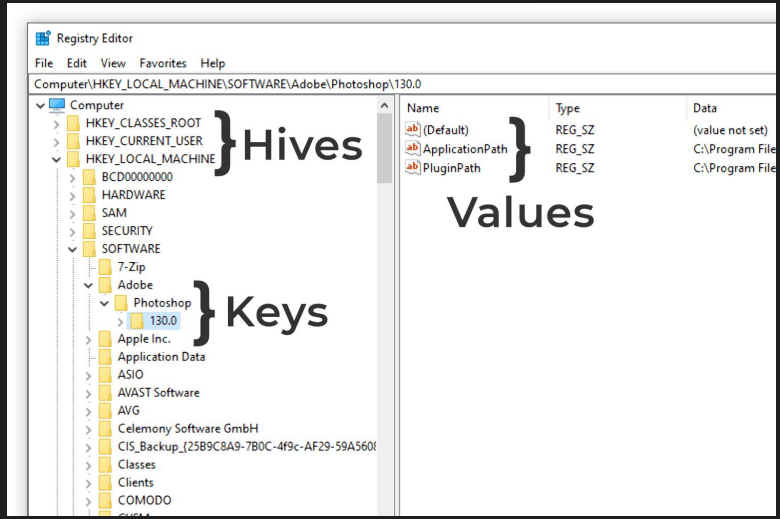
- Supported and actively developed
  - Varying version present on various OSes
- Verb-Noun syntax
  - Get-ChildItem
  - Set-Content
  - Invoke-Expression
- Integrated with the Windows API
  - Can manage pretty much anything with it
    - Users, Services, Apps, Registry Keys,
- Everything is treated as an object
  - Scripting and using command output MUCH easier than BASH

GUI Panes to Know

- Regedit
  - Registry Keys
- File Explorer
  - Filesystem
- Compmgmt.msc
  - Scheduled Tasks, Event Viewer, Shared folders, and Local Users
- Services.msc
  - Services
- Control Panel
  - System settings
- Task Manager
- Server Manager
  - Roles and features
- Gpedit.msc
  - Group policy editor

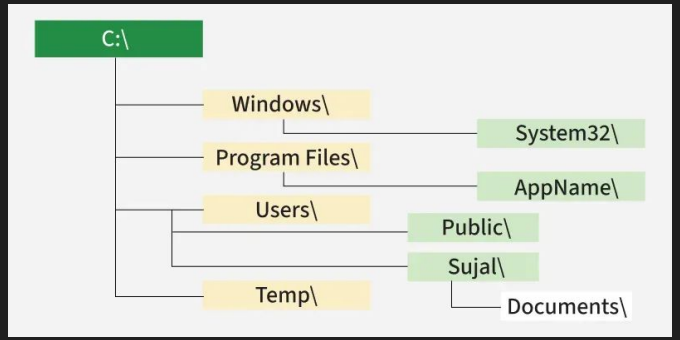
Registry (Regedit)

- Most configurations link back to a registry key
  - Windows' internal database of configs
- Can be modified with CLI
  - Reg add
  - Set-ItemProperty
- Can be used to gain persistence
  - Run program on "x" keys

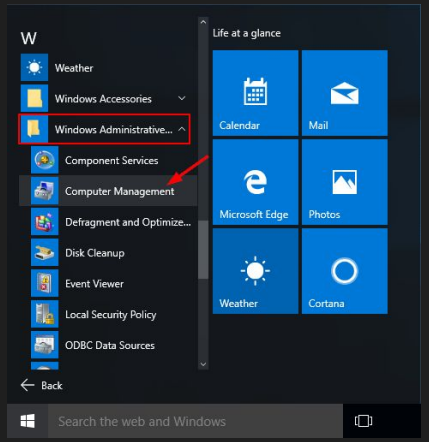


Filesystem

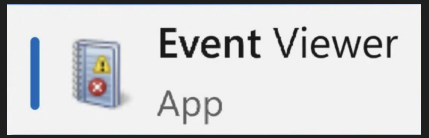
- Holds all data
  - Even the registry
- Paths to know
  - C:\Windows\System32
    - System binaries and libraries
  - C:\Users
    - All user files
  - C:\Program Files | C:\Program Files (x86) | C:\ProgramData
    - Application binaries, libs, and configuration files
- Highly configurable permissions
  - Big reason enterprises use Windows in the first place



Computer Management



- Audit Users
  - Delete/disable unauthorized
  - Set group membership/privileges
- Check file shares
  - C$ and in some cases ADMIN$
  - Domain Controllers have more default ones
- Scheduled Tasks
  - Can run executables on "x" condition
    - Time interval
    - On start
    - On execution of other program
    - etc...



- Event Viewer can troubleshoot issues and identify malicious activity
  - See event ID **4625** for failed login attempts
  - **7045** for service creation

Services

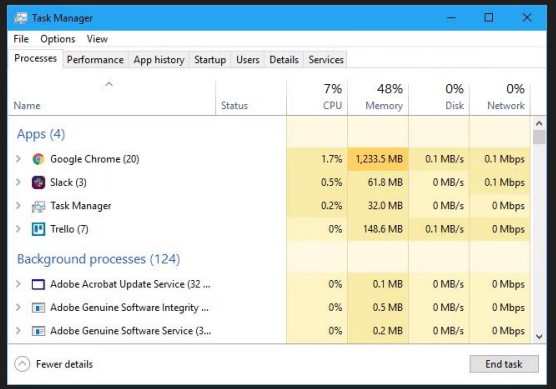
- Binaries (executables) designed to run in the background to serve some sort of OS or 3rd party functionality
  - Filezilla Server service - 3rd party FTP server
  - Bitlocker - Native drive encryption functionality
- Services are identified by one of two things
  - Display name - Simple to understand
    - World Wide Web Publishing Service
  - Service name - Shorter and used to refer to service internally
    - W3SVC
- Some attacks make use of temporary services to spawn a shell as the SYSTEM user

Control Panel

- Manage Firewall
  - Enable it, create rules to allow desired inbound (to services) and outbound (internet/dependencies) traffic
- Enable/Disable Remote Desktop
  - Disable if unused
- Enable User Account Control
  - Runs processes with lower privs if possible, introduces popups that make you explicitly elevate.
- Manage network adapters
  - Change IPv4 settings

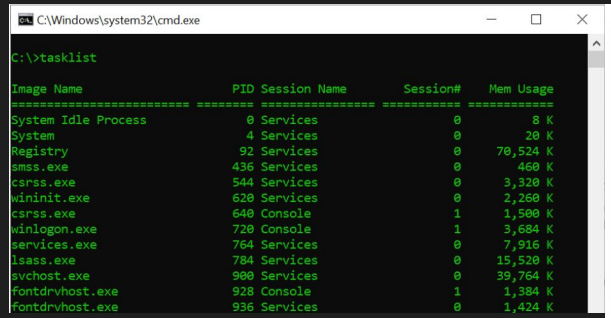
Processes (Task Manager)

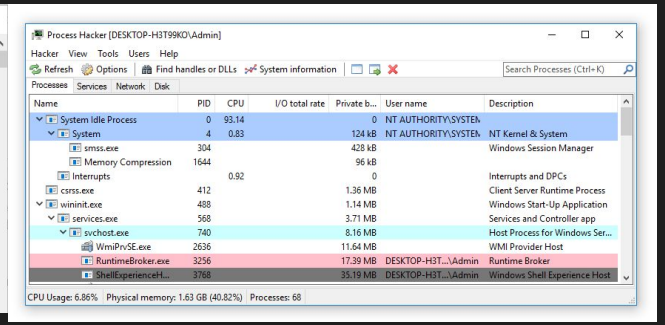
- Processes originate from executable files
  - Apps
    - Discord
  - System binaries
    - Winlogon
- Can create them and kill them (mostly)
- Types
  - App
    - Can be terminated by user
  - Background
    - No user interaction
  - Windows
    - System level and are auto launched

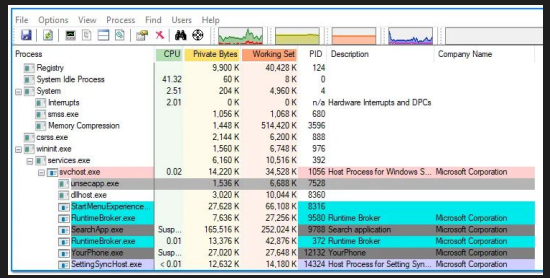


Task Manager Alternatives

- Tasklist or Get-Process
  - Cmd and powershell commands, can supply arguments for more info
- Process Hacker
  - Very thorough and detailed if desired
- Process Explorer
  - Color coded, similar to process hacker, can scan all processes with virustotal





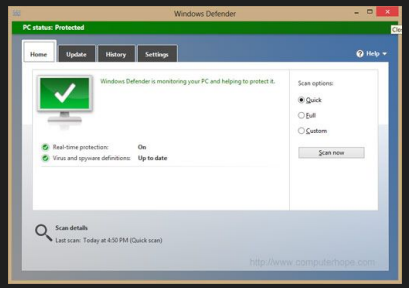


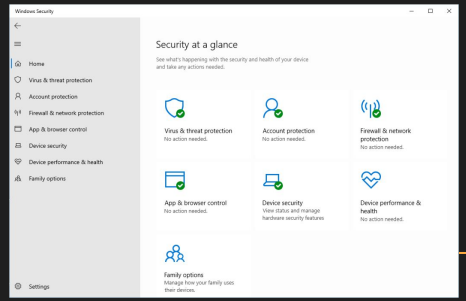
Server Manager

- Only available on "Server" editions of Windows
- Easy GUI access to some tools
  - Can manage roles and features for server editions in this server manager window
    - Roles and Features are optional add ons for the OS
      - IIS web server
      - Active Directory Domain Services
      - Microsoft Defender

Defender

- Installed by default on every OS except Server 2012
  - Can be uninstalled on Server editions through server manager
- Effectiveness depends on windows updates and OS
- Scan, Exclude, and Remediate malicious files
  - Newer defender has fancy capabilities like core isolation, attack surface reduction rules, exploit mitigations (DEP, ASLR, SEHOP, etc)





Gpedit

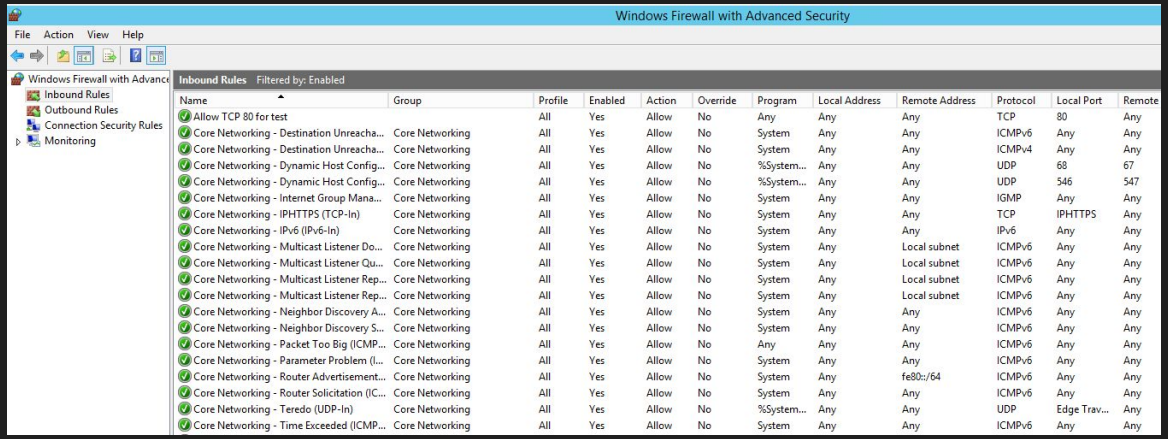
- Control IT and security specific OS policies
- Secpol
  - Windows Settings > Security Settings
    - Account Policies
    - Local Policies
- Group Policy Objects (aka just more configs)
  - Administrative Templates
    - Everything here is considered a GPO
- What exactly do I set here?
  - https://www.stigviewer.com/stig/windows_server_2019/

Windows Firewall

Stateful

Windows Firewall Rules

- GUI
  - more user friendly
  - Control Panel > System and Security > Windows Defender Firewall > Advanced settings



Windows Firewall Rules

- PowerShell
  - Faster/allows for scripting and automation
  - Very intuitive/readable
- View firewall rules
  - Get-NetFirewallRule
- Create a new rule
  - Can allow/block traffic on ports or coming from programs
    - Use -LocalPort or -Program
  - New-NetFirewallRule -DisplayName [Rule Name] -Direction [Inbound/Outbound] -Action [Allow/Block]

Windows Firewall Rules

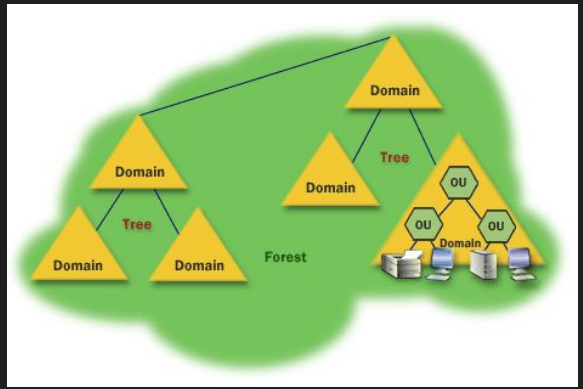
- Netsh command-line
  - Consistent across windows versions
  - Can be faster to type with shorthand
- View firewall rules
  - Netsh advfirewall firewall show rule name=all
- Create a new rule
  - Netsh advfirewall firewall add rule name="[name]" dir=[in/out] action=[allow/deny] protocol=[tcp/udp] localport=[port #]
  - Netsh a f a r n="ssh" dir=in action=allow protocol=tcp localport=22

02
Active Directory

Domains?

Explanation

- Simply put, it's a way to centralize management of policies, computers, and users and store this data in a database
- How does your CPP login work everywhere? (library, canvas, broncodirect)
- Forrest (nebula.lan)
  - Domain (nebula.lan or nebula-na.lan)
    - Possibly child domains (us.nebula-na.lan)



Explanation Cont.

- A forrest is just an overall cabinet containing multiple draws (domains)
- Domains are headed by machines designated as Domain controllers (DC)
    - Member servers and workstations join the domain And are subject to management from the DC

Explanation Cont.

- Within Domains you have objects
  - Organizational Units (OUs)
  - Groups
  - Users
    - Can log into to every domain joined computer
  - Misc AD data
- Lightweight Directory Access Protocol
  - Query any and all information quickly
  - Can use any LDAP client
    - Dedicated programs
    - Internal windows tools
    - Powershell

LDAP

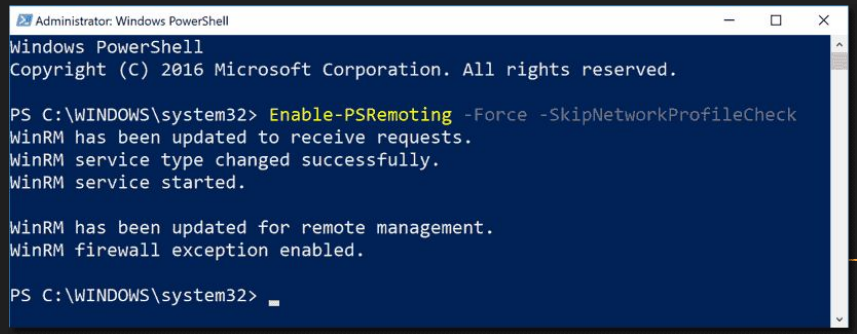
- Lightweight Directory Access Protocol
- Used in Active Directory (AD)
- Allows programs to find information quickly
- Interacts with, stores, and extracts objects
- Common uses:
  - User authentication
  - Looking up user info
  - User authorization

Domain Computer Management

- Know so far:
  - Login to any computer
  - Synced policies
    - But how?
- Deploy Policies via GPMC.msc
  - Default Domain Policy
    - Applies to all systems in the domain
  - Default Domain Controller Policy
    - Applies to all DCs
- WinRM (PS Remoting)
  - Enabled by default when you join a domain
    - It's how a lot of the behind the scenes management goes on
  - PS remoting lets us run powershell commands and scripts across the domain

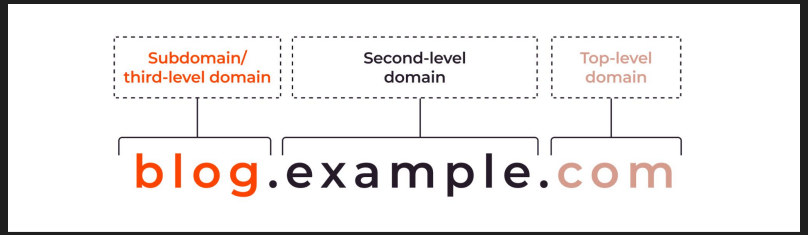
PS Remoting

- By default servers will only accept connections originating within the domain and from admin users
- WinRM is enabled by default but PS Remoting isn't
  - Enable-PSRemoting -force
- Example commands
  - Invoke-Command -ScriptBlock {whoami} -ComputerName WEBSRV1
  - Enter-PSSession -ComputerName Server01
    - Interactive



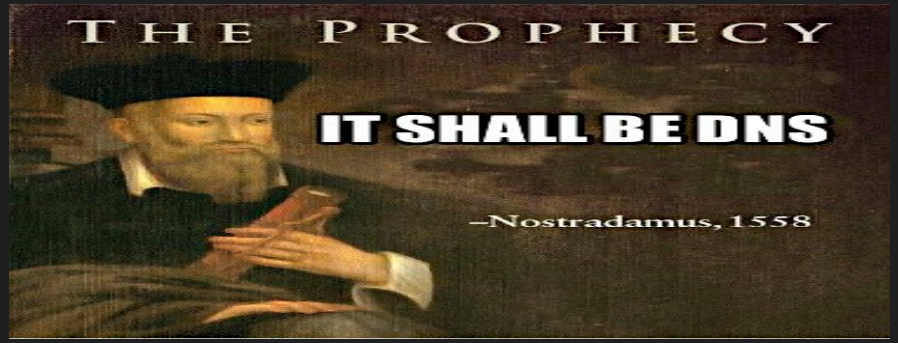
Domain Names?

- Previous slide used the hostname of machines to connect. How does this work?
- Domain Name System (DNS)
  - Attach words to ips
    - User friendly
  - When IPs change, domain names stay the same but adjust to reflect the new IP
    - Good for avoiding hiccups with things like DHCP
- Dns.google -> 8.8.8.8
  - Server01.nebula.lan -> 192.168.1.25
- Uses UDP for queries
  - TCP used for other operations



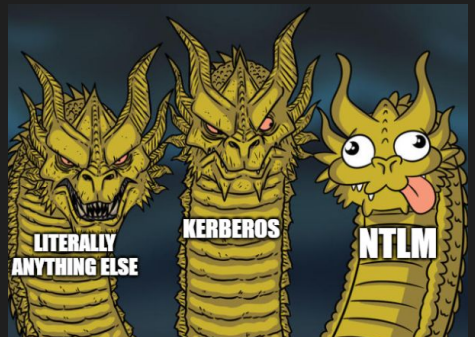
DNS Service

- DNS Server role can be installed on Windows Server editions
- DNS Server
  - Finds domain controllers
    - User with AD user account logs in to an AD domain
    - DNS Service queries DNS server to locate domain controller
    - DNS server responds with DC's IP address
  - Converts computer names to IP Addresses
    - User requests a name



Authentication

- 2 main ones, NTLM/NetNTLM and Kerberos
- NTLM/NetNTLM
  - NTLM is used locally on each machine to verify **local** users
  - NetNTLM is the **network** authentication protocol
  - More prone to cracking
- Kerberos
  - Uses the concept of tickets
  - Tickets have a lifetime (10 hours)
  - Derived from the user's password and the krbtgt account password



Service Security Descriptors

- Every AD object has a security descriptor
  - Ex. user objects, group objects, printers, shared folders
- Security descriptors hold security information including:
  - Owner security identifier (SID)
    - Unique value assigned to the account/group that owns the service
  - Discretionary access control list (DACL)
    - Identifies users that are allowed/denied access to the service
  - System access control list (SACL)
    - Allows administrators to log access attempts to the service

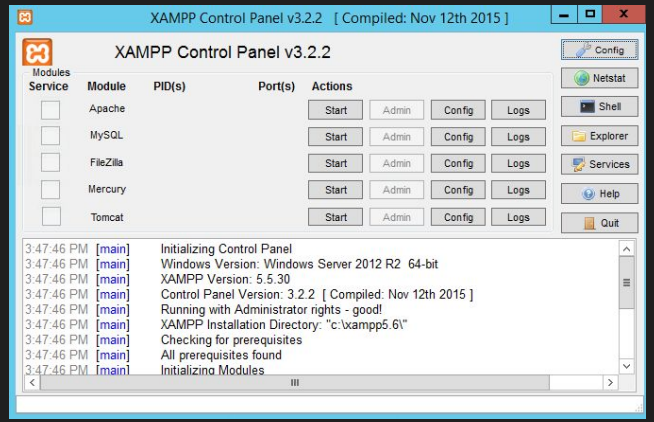
03
Other Services

Internet Information Services

- HTTP Web server built into Windows
  - Role you have to add
- Supports .Net (aspx) out of the box, PHP also configurable
- Very plain and boring out of the box

XAMPP

- Packaged installation of parts of a web framework
- Apache serving HTTP
- MySQL hosting database
- Filezilla for FTP
- Tomcat is Apache but in java so worse



(Content Management System) CMS

- Actual apps that run on the framework
  - Wordpress, OpenCart, Drupal, MediaWiki
  - Blogs, Ecommerce sites, General management system, wiki
- All setup and work the same
  - HTTP - host the actual files allowing network access
    - Apache or IIS
  - Scripting Engine - Processes logic and renders content on HTTP pages
    - PHP
  - Database - contains all site data like posts, users, etc
    - MySQL

1. Put CMS files in webroot
2. Setup database in MySQL
3. Configure db credentials in CMS config file
4. Browse to http:\\localhost to run through the installer

HW

Troubleshooting

Homework !!

Troubleshooting Services

Due September 11th @ 11:59 PM
jessh.zip/2026ccdcfallhw2

Deploy a pod

- Connect to the VPN and visit kamino.sdc.cpp
- Navigate to Pods > Click "CCDC Fall Week 2" > Clone
- **Please delete your pod after you finish the lab!**

Thanks!

Any questions? Please ask questions; we are available to help :D