



Purple Team

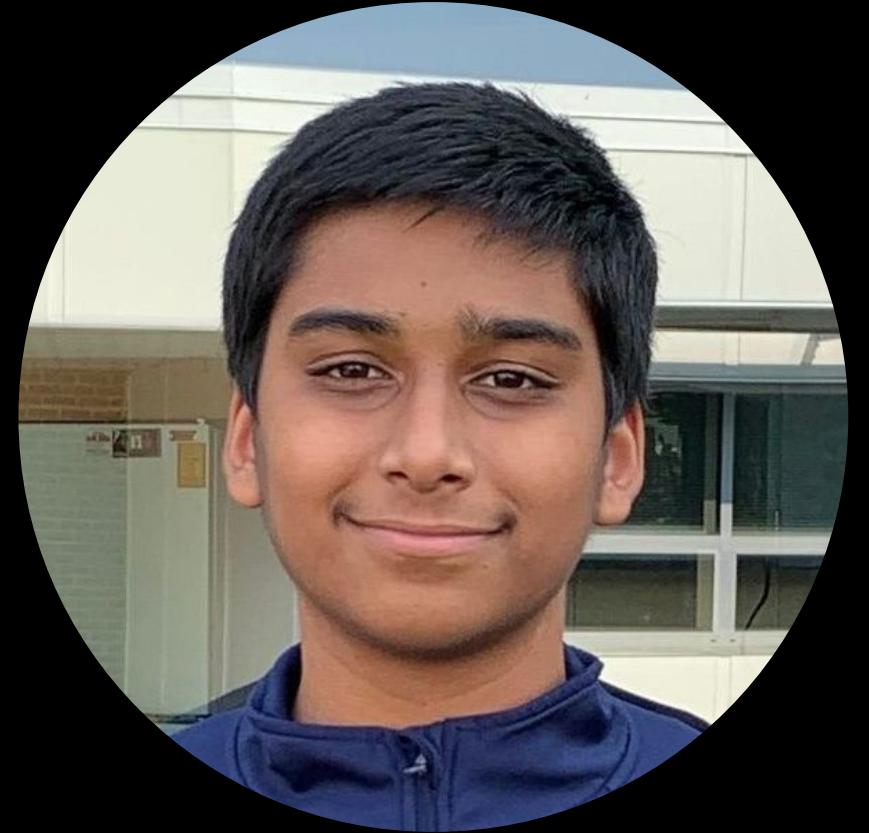
FA2026 • 2026-10-20

System Administration & Hardening

Krishnan Shankar, Pranav Srinivasan

Krishnan Shankar

- SIGPwny Admin
- Computer Engineering '27
- Fun fact: I love planespotting



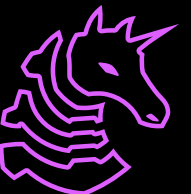
Pranav Srinivasan

- SigPwny Purple Team Lead
- Computer Engineering '28
- Fun fact: I play classical guitar



Table of Contents

- Linux
 - General Hardening
 - Securing Files
 - Process Monitoring
 - Services & Systemd
 - Network Defense & Host Firewalls
- Windows
- Containerization
 - Docker
 - Firejail
 - Bubblewrap



Linux



Current Users and Login Info

- Check currently logged in users with `w` command (or `who`)
- Display recent logins with `last`

```
~ > w
05:46:09 up 1 day, 10:08,  1 user,  load average: 0.81, 0.89, 0.90
USER      TTY      LOGIN@  IDLE   JCPU   PCPU   WHAT
krishnan  tty2     Sun19   34:08m  0.05s  0.05s  /nix/store/y4xj2r3vqln1k6av80s62qjgvvnkpws-plasma-workspace-6.7.5/bin/startplasma-wayland
```

```
~ > last
krishnan  tty2                Sun Oct 18 19:38    still logged in
reboot    system boot        6.18.55            Sun Oct 18 19:37    still running
reboot    system boot        6.18.55            Sat Oct 17 11:07 - 11:08    (00:00)
reboot    system boot        6.18.55            Sat Oct 17 04:21 - 11:05    (06:43)
reboot    system boot        6.18.55            Fri Oct 16 15:56 - 15:57    (00:00)
krishnan  tty2                Wed Oct 14 01:02 - down    (2+14:53)
reboot    system boot        6.18.55            Wed Oct 14 01:01 - 15:56    (2+14:54)
krishnan  tty2                Mon Oct 12 06:41 - down    (1+18:19)
reboot    system boot        6.18.55            Mon Oct 12 06:41 - 01:01    (1+18:20)
krishnan  tty2                Thu Oct  8 11:00 - down    (3+10:33)
reboot    system boot        6.18.55            Thu Oct  8 10:56 - 21:34    (3+10:37)
krishnan  tty2                Tue Oct  6 17:44 - down    (08:39)
reboot    system boot        6.18.54            Tue Oct  6 17:44 - 02:23    (08:39)
krishnan  tty2                Mon Oct  5 15:53 - down    (1+01:49)
reboot    system boot        6.18.54            Mon Oct  5 15:53 - 17:43    (1+01:50)
krishnan  tty2                Mon Oct  5 15:21 - down    (00:31)
reboot    system boot        6.18.54            Mon Oct  5 15:21 - 15:52    (00:31)
```



Users and Group

- User information is stored in `/etc/passwd` and `/etc/shadow`

<code>/etc/passwd</code>	<code>/etc/shadow</code>
Home directory, login shell, and user ID	Hashed password and expiration info



/etc/passwd

user name user ID comment
 section login shell

```
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
```

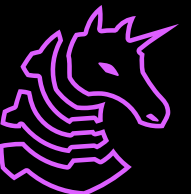
 default group ID home directory

has password in
/etc/shadow

A diagram showing a line from the /etc/passwd file: 'ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash'. Arrows point from labels to specific parts of the line: 'user name' points to 'ubuntu'; 'user ID' points to '1000'; 'comment section' points to 'ubuntu,,,'; 'login shell' points to '/bin/bash'; 'default group ID' points to '1000'; 'home directory' points to '/home/ubuntu'; and 'has password in /etc/shadow' points to 'x'.

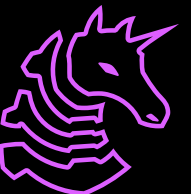
/etc/passwd and /etc/shadow Misconfigs

- Change the default passwords! Always do this!
 - `passwd <user>`
- Unnecessary users
 - More users means a larger attack surface
 - If a user's shell is set to `/bin/false`, `/bin/nologin`, or `/bin/<not-a-shell>`, it's probably a system account and can generally be ignored
 - Delete unneeded accounts
 - `sudo userdel <username>`
 - Add the `-r` flag to delete that user's home directory
 - Add the `-f` flag to force delete a user account, even if they're signed in
 - If unsure about deleting an account, set the login shell to `/bin/false`
- UID/GID of 0 on non-root account
 - Linux will treat this user/group as root with regards to file permissions
 - Change the UID/GID to a unique number if found, or remove account



Sudo

- The root account can do basically anything
- Sudo allows a lesser privileged user to run a command as a different user, including root
 - sudo <command>
 - sudo su
 - Switches to root
- /etc/sudoers contains the rules as to which users can use sudo and under what circumstances



/etc/sudoers

- The sudoers files can be edited with `visudo`
 - Try to NOT use vim/nano/etc., unless you really know what you're doing
- Users can be restricted to only being able to use sudo for specific commands
- If a user or group doesn't need to be able to sudo, they shouldn't be able to

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

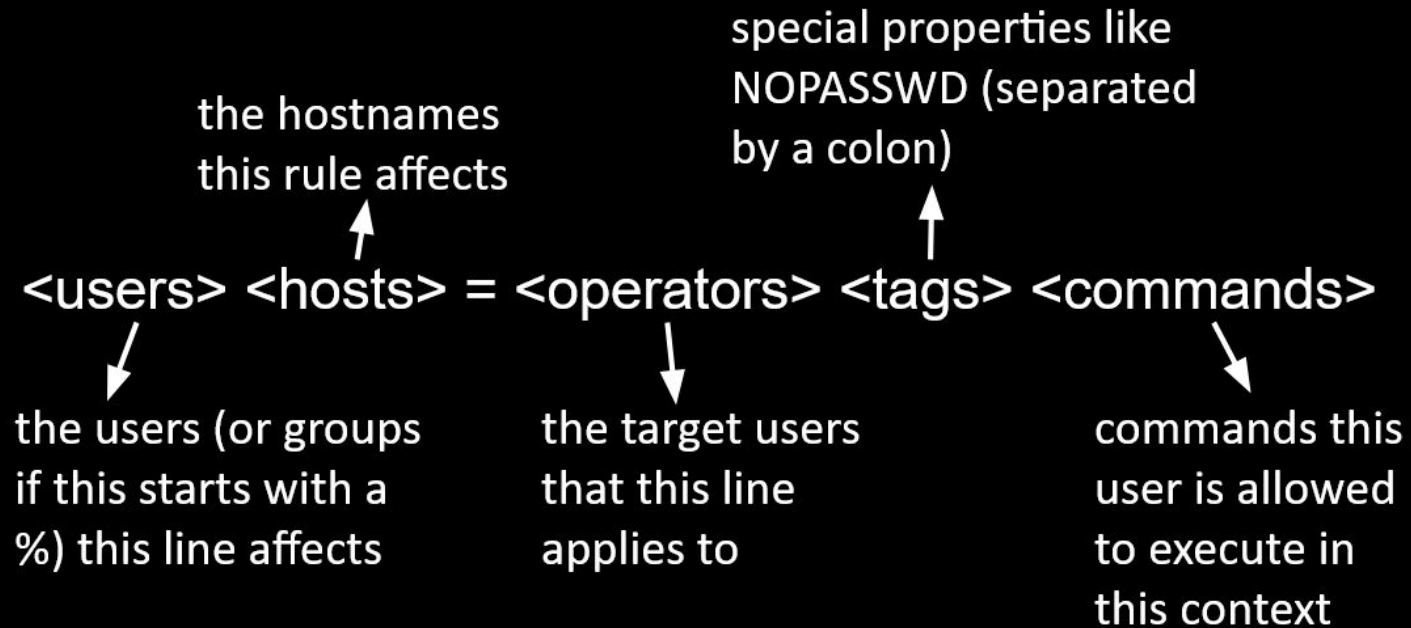
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

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



/etc/sudoers Cont.

- Any of these sections can be replaced with ALL to have it apply to all users, hosts, commands, etc.



/etc/sudoers Cont.

- ALL ALL=(ALL:ALL) ALL
 - If you see this, ANY user can use sudo with any command as any other user without a password
 - This is very bad



File Ownership

- Files are owned an user and a group
- File owners can change file permissions and ownership with `chmod` and `chown`

```
~/NixOS main > ls -la
total 160
drwxr-xr-x 18 krishnan users 4096 Oct 11 03:45 .
drwx----- 74 krishnan users 4096 Oct 19 04:17 ..
drwxr-xr-x  2 krishnan users 4096 Oct  9 23:31 custom
drwxr-xr-x  2 krishnan users 4096 Oct  9 18:56 devshells
drwxr-xr-x  2 krishnan users 4096 Sep  9 21:48 dotfiles
-rw-r--r--  1 krishnan users 59311 Oct  9 18:56 flake.lock
-rw-r--r--  1 krishnan users 2802 Oct 11 05:30 flake.nix
drwxr-xr-x  2 krishnan users 4096 Oct  9 18:56 formatter
drwxr-xr-x  9 krishnan users 4096 Oct 20 06:04 .git
drwxr-xr-x  3 krishnan users 4096 Oct  9 18:56 .github
-rw-r--r--  1 krishnan users    7 Dec  2 2024 .gitignore
-rw-r--r--  1 krishnan users  129 Oct  9 18:56 .gitmodules
drwxr-xr-x  5 krishnan users 4096 Mar 31 2025 .idea
-rw-r--r--  1 krishnan users 1073 Oct  9 18:56 LICENSE
drwxr-xr-x  4 krishnan users 4096 Oct  9 18:56 modules
drwxr-xr-x  7 krishnan users 4096 Oct  9 18:56 profiles
-rw-r--r--  1 krishnan users 6101 Oct 11 03:45 README.md
drwxr-xr-x  2 krishnan users 4096 Oct 11 03:45 .run
drwxr-xr-x  2 krishnan users 4096 Oct  9 18:56 scripts
drwxr-xr-x  2 krishnan users 4096 Oct  7 01:58 secrets
-rw-r--r--  1 krishnan users  745 Oct  9 18:56 .sops.yaml
drwxr-xr-x  8 krishnan users 4096 Oct  9 18:56 systems
drwxr-xr-x  2 krishnan users 4096 Oct 18 20:50 utils
drwxr-xr-x  2 krishnan users 4096 Oct  9 18:56 .vscode
drwxr-xr-x  2 krishnan users 4096 Oct 11 03:45 .zed
```



File System Permissions

Example Listing:

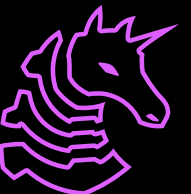
```
drwxrwxr-x  2  ronan  ronan      4096  Sep  29  20:58  Payloads
-rwx-----  1  ronan  ronan 24383064  Sep  29  21:28  shellcode.bin
-rwxrwxrwx  1  ronan  ronan      4096  Nov  23  12:10  /etc/shadow
```



File System Permissions

Changing File Permissions

- Octal for Read (4), Write (2), Execute (1), then repeat for each category
- Use the `chmod` command
- If I want to grant everything for everyone: `chmod 777 file`
- If I want read + write for only me: `chmod 600 file`



File Permissions Cont.

- Follow the principle of least privilege.
- Config files in /etc, scripts that are periodically run by the system (ie. cronjobs), and others should ***not*** be globally writable
- If in doubt (what permissions does a specific file need?):
 - Google it
 - Checking your own system can be helpful

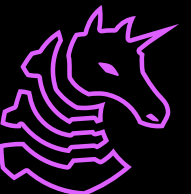


SUID/SGID Bits

- Set User ID (SUID)
 - SUID is a special permission that if set, enables a file to be run from the UID of the file's owner, instead of the user running it
- Set Group ID (SGID)
 - SGID allows the file to be executed as the group that owns the file if set, similar to SUID
 - If set on a directory, any files created in the directory will have their group ownership set to that of the directory owner

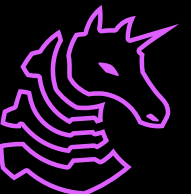
```
[tcarrigan@server ~]$ ls -l /usr/bin/passwd  
-rwsr-xr-x. 1 root root 33544 Dec 13 2019 /usr/bin/passwd
```

Note the **s** where **x** would usually indicate execute permissions for the user.



SUID/SGID Bits Cont.

- When you SUID a file owned by root, the executable runs with root permissions
- Note that some executables, like sudo, need this and removing SUID will prevent everyone from sudoing
- If an executable doesn't need it, it's best to remove it
- Find such files:
 - `find / -perm -u=s -type f 2>/dev/null`
 - `find / -perm -g=s -type f 2>/dev/null`



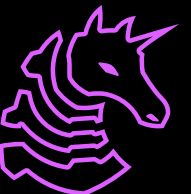
SUID/SGID Bits Cont.

- Find such files:
 - `find / -perm -u=s -type f 2>/dev/null`
 - `find / -perm -g=s -type f 2>/dev/null`

```
krishnan@krishnan-mew ~ > sudo find / -perm -u=s -type f 2>/dev/null
/run/wrappers/wrappers.xbJmhptDlF/unix_chkpwd
/run/wrappers/wrappers.xbJmhptDlF/umount
/run/wrappers/wrappers.xbJmhptDlF/sudoedit
/run/wrappers/wrappers.xbJmhptDlF/sudo
/run/wrappers/wrappers.xbJmhptDlF/su
/run/wrappers/wrappers.xbJmhptDlF/sg
/run/wrappers/wrappers.xbJmhptDlF/passwd
/run/wrappers/wrappers.xbJmhptDlF/newgrp
/run/wrappers/wrappers.xbJmhptDlF/mount
/run/wrappers/wrappers.xbJmhptDlF/firejail
/run/wrappers/wrappers.xbJmhptDlF/chsh

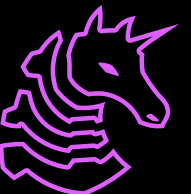
krishnan@krishnan-mew ~ 3s > sudo find / -perm -g=s -type f 2>/dev/null

krishnan@krishnan-mew ~ 3s >
```



File Attributes

- List with `lsattr`
- Modify with `chattr`
- You can set a file to be immutable (even if you `chmod 777` it)
 - `sudo chattr +i /etc/passwd /etc/shadow`
- Also, the append-only attribute may be helpful:
 - `sudo chattr +a /var/log/auth.log`
- Obviously, if the attacker has sudo/root, they can undo this
- (Note: These are examples; filenames/filepathes will vary)



Process Monitoring

- The `ps` command displaying information on currently running processes
 - I personally like `ps -auxf`
- Monitor the system for currently running process and kill them with `kill` or `pkill`
 - However, this is merely treating a symptom because the attacker likely has persistence and can restart malicious activity easily
 - Document information about the process and continue looking to find the root issue
 - In competitions, this can sometimes buy you time (so it's generally worth doing) but does NOT solve the problem and will not be enough for an IR

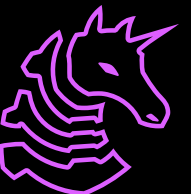


Process Monitoring

```
root      169058 ?        \_ [kworker/13:2]
root      169362 ?        \_ [kworker/u64:2-events_unbound]
root      169374 ?        \_ [kworker/u64:5-events_unbound]
root      1 ?        /run/current-system/systemd/lib/systemd/systemd --switched-root --system --deserialize=47
root      452 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd-journald
systemd+  454 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd-oomd
systemd+  478 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd-timesyncd
root      485 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd-udevd
systemd+  498 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd-resolved
message+  745 ?        \_ /nix/store/jn6ykk9k52jxmjrg95vwaq7w5czi4bpg-dbus-broker-37/bin/dbus-broker-launch --scope system --audit
message+  746 ?        | \_ dbus-broker --log 11 --controller 10 --machine-id 79848c5f8fad4db99b68e3fb1d58ba17 --max-bytes 536870912 --max-fds 4096 --max-match
root      748 ?        \_ /nix/store/6qz6q8j2r2cirh33f67mpdsxnqf8d680-networkmanager-1.58.0/sbin/NetworkManager --no-daemon
nsd       753 ?        \_ /nix/store/yv1diivazrv3pf79lgm14awd3kb8738m-nsd-1.5.2/bin/nsd
root      756 ?        \_ /nix/store/cb4wqc0gxzk9pzdwdgpy6nfba7p6d5ca-qemu-host-cpu-only-11.1.0-ga/bin/qemu-ga --statedir /run/qemu-ga
root      792 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd-logind
wpa_sup+  836 ?        \_ wpa_supplicant -s -u -Dnl80211,wext -c /etc/wpa_supplicant/imperative.conf
root      837 ?        \_ nix-daemon --daemon
root      844 ?        \_ sshd: /nix/store/ggcm0s524bsly3qc7kww945qs7003psf-openssh-10.5p1/bin/sshd -D -f /etc/ssh/sshd_config [listener] 0 of 10-100 startups
root      845 ?        \_ /nix/store/lfqbsp1wh59izfbs39sf9yds7dp0jwdj-tailscale-1.102.3/bin/tailscaled --state=/var/lib/tailscale/tailscaled.state --socket=/run/
root      169266 pts/0    | \_ /run/current-system/sw/bin/login -f -h 100.95.242.59 -p
krishnan  169273 pts/0    | \_ -zsh
krishnan  170138 pts/0    | \_ ps -axfo user,pid,TTY,cmd --cols 165
redis     860 ?        \_ /nix/store/1pnhssld9r4s4qxmsm9fbqvab9h2fyx3-redis-8.10.1/bin/redis-server 127.0.0.1:6379
root      1045 tty1      \_ /nix/store/qikidl3z4p1i05wri3y53ya4gq2nb8as-shadow-4.20.2/bin/login --
krishnan  1074 tty1      | \_ -zsh
krishnan  1052 ?        \_ /nix/store/sm8d6jpilwdy3bw3yq2lv8rr8jld26pb-systemd-261.2/lib/systemd/systemd --user
krishnan  1054 ?        | \_ (sd-pam)
krishnan  1203 ?        | \_ /nix/store/nvh2r6adpxhkl37wsxglz4w9cbsz506h-gnupg-2.4.9/bin/gpg-agent --supervised
krishnan  1210 ?        | | \_ sddm --multi-server
krishnan  1219 ?        | \_ /nix/store/jn6ykk9k52jxmjrg95vwaq7w5czi4bpg-dbus-broker-37/bin/dbus-broker-launch --scope user
krishnan  1220 ?        | \_ dbus-broker --log 11 --controller 10 --machine-id 79848c5f8fad4db99b68e3fb1d58ba17 --max-bytes 1000000000000000 --max-fds 250000
krishnan  1101 tty1      \_ -zsh
krishnan  1126 tty1      | \_ /nix/store/csixxp50861bxiysrihlqgw5gjdwz1-gitstatus-1.5.5/bin/gitstatusd -G v1.5.5 -s -1 -u -1 -d -1 -c -1 -m -1 -v FATAL -t 32
krishnan  1123 tty1      \_ -zsh
krishnan  1124 tty1      \_ -zsh
krishnan  169299 pts/0    \_ -zsh
krishnan  169323 pts/0    | \_ /nix/store/csixxp50861bxiysrihlqgw5gjdwz1-gitstatus-1.5.5/bin/gitstatusd -G v1.5.5 -s -1 -u -1 -d -1 -c -1 -m -1 -v FATAL -t 32
krishnan  169320 pts/0    \_ -zsh
krishnan  169321 pts/0    \_ -zsh
```

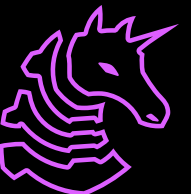
Process Monitoring Cont.

- The `/proc` filesystem contains one subdirectory for each process running on the system named after the process ID. ([docs](#))
- `/proc` also hosts files that offer system-wide information, including:
 - `/proc/cpuinfo`: Details about the system's CPUs
 - `/proc/meminfo`: Information about memory usage and statistics.
 - `/proc/devices`: Lists character and block devices.
 - `/proc/mounts`: Information about mounted filesystems.
 - `/proc/version`: Linux kernel version details.
- We can find memory maps, open files, process binaries, and other useful information in here



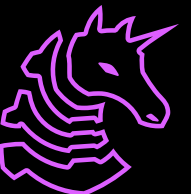
Services

- A lot of Linux distros have their own ways of managing services/init
- The most common is **systemd**
- Some others are: **openrc**, dinit, runit, s6-rc



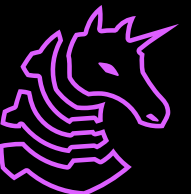
Services

- A lot of Linux distros have their own ways of managing services/init
- The most common is **systemd**
- Some others are: **openrc**, dinit, runit, s6-rc
- A few notes:
 - Alpine Linux (which has been present in past CCDC Invitationals!) uses **openrc**! Do not assume systemd, do your research beforehand
 - VyOS (the router) rolls its own custom thing (with commands like **configure**; **set service**; **delete service**; **commit**; **save**)
 - Firewalls are also weird; PanOS requires the GUI and Cisco FTD additionally uses **pmtool** (process manager tool) for core services



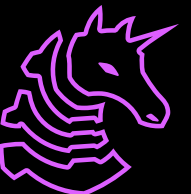
Systemctl

- Systemd does a lot of things!
- The core of systemd is managed by systemctl (system controller)
- Common commands:
 - `systemctl list-units | grep service`
or... `systemctl list-units --type service`
 - `systemctl status sshd.service`
 - `systemctl start sshd.service`
 - `systemctl stop sshd.service`
 - `systemctl restart sshd.service`
 - `systemctl enable sshd.service`
 - `systemctl disable sshd.service`



Systemd

- Systemd does a lot of things!
 - Acts as the init system: Starts core services needed for the PC to boot
 - Handles services, targets (groups of services), mounts, processes, slices (groups of processes), sockets, swapfiles, timers, tmpfiles, ...
 - systemd-journald (journalctl) handles logs
 - systemd-networkd (networkctl) handles networking (like NetworkManager)
 - systemd-resolved (resolvectl) handles DNS (see also /etc/resolv.conf)
 - systemd-timesyncd (timedatectl) handles NTP
 - systemd-logind (loginctl) handles login sessions (like the `last` command)
 - There's a whole thing for user services: `systemctl --user ...`
 - systemd-oomd, systemd-udev, systemd-nspawn, ...



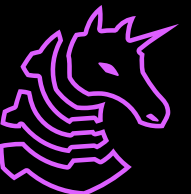
Systemd

- Most systemd configuration files are in `/etc/systemd/system`
 - Note the file extensions (`.service`, `.target`, `.timer`, `.socket`, `.slice`, ...)
 - Many services can be hidden in nested directories (like `sysinit.target.wants`, `multi-user.target.wants`, `graphical.target.wants`, etc.)
- `systemd show` can be used to view all parameters of a service (or just `cat /etc/systemd/system/<name>.service`)
- Some more helpful commands:
 - Live feed of logs: `journalctl -xef`
 - View logs of a systemd service: `journalctl -u sshd.service`
 - View/manage DNS: `resolvectl`; `resolvectl flush-caches`



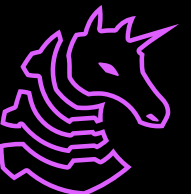
Network Monitoring

- Besides assumed-breach, open ports are generally the way attackers get into a system
- How do we check open ports from a system administration POV?



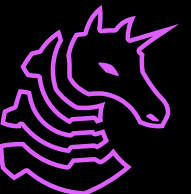
Network Monitoring

- Besides assumed-breach, open ports are generally the way attackers get into a system
- How do we check open ports from a system administration POV?
 - We can just portscan from another computer!



Network Monitoring

- Besides assumed-breach, open ports are generally the way attackers get into a system
- How do we check open ports from a system administration POV?
 - We can just portscan from another computer!
- Or, use some nice Linux utilities from the same system (you might remember this from service enumeration)
 - `sudo ss -tunlp`
 - `sudo netstat -tunlp`



Network Monitoring

- Or, use some nice Linux utilities from the same system (you might remember this from service enumeration)
 - `sudo ss -tunlp`
 - `sudo netstat -tunlp`

```
krishnan@krishnan-mew ~ > sudo ss -tunlp
06:46:06 AM
Netid      State      Local Address:Port      Peer Address:Port      Process
udp        UNCONN     127.0.0.54:53            0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=24))
udp        UNCONN     127.0.0.53%lo:53         0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=22))
udp        UNCONN     0.0.0.0:5353             0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=17))
udp        UNCONN     0.0.0.0:5355             0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=13))
udp        UNCONN     0.0.0.0:41641            0.0.0.0:*               users:(( ".tailscaled-wra",pid=845,fd=20))
udp        UNCONN     [::]:5353                [::]:*                  users:(( "systemd-resolve",pid=498,fd=18))
udp        UNCONN     [::]:5355                [::]:*                  users:(( "systemd-resolve",pid=498,fd=15))
udp        UNCONN     [::]:41641               [::]:*                  users:(( ".tailscaled-wra",pid=845,fd=18))
tcp        LISTEN     127.0.0.53%lo:53         0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=23))
tcp        LISTEN     127.0.0.54:53            0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=25))
tcp        LISTEN     0.0.0.0:22               0.0.0.0:*               users:(( "sshd",pid=844,fd=6))
tcp        LISTEN     0.0.0.0:5355             0.0.0.0:*               users:(( "systemd-resolve",pid=498,fd=14))
tcp        LISTEN     127.0.0.1:6379           0.0.0.0:*               users:(( "redis-server",pid=860,fd=9))
tcp        LISTEN     100.124.137.41:62280     0.0.0.0:*               users:(( ".tailscaled-wra",pid=845,fd=28))
tcp        LISTEN     [fd7a:115c:a1e0::2a2d:892a]:58997 [::]:*                  users:(( ".tailscaled-wra",pid=845,fd=29))
tcp        LISTEN     [::]:22                  [::]:*                  users:(( "sshd",pid=844,fd=7))
tcp        LISTEN     [::]:5355                [::]:*                  users:(( "systemd-resolve",pid=498,fd=16))
```

Linux Firewall

- **iptables**: Provides various functionalities to control network traffic
- **nftables**: A successor to the “iptables” utility, with enhanced packet filtering and NAT capabilities
- **firewalld**: Has predefined rule sets, works a bit differently from the others and comes with different pre-built network zone configurations.
- **ufw**: “Uncomplicated Firewall” common on Debian/Ubuntu

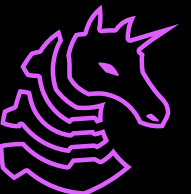


Linux Firewall

- iptables
- nftables
- firewalld
- ufw

Default Front-End Firewalls by Distribution

- **UFW (Uncomplicated Firewall):** Default user-friendly front-end used by **Ubuntu**, **Debian**, and their derivatives (like Linux Mint). 
- **Firewalld:** Default dynamic firewall management tool used by **Red Hat Enterprise Linux (RHEL)**, **Fedora**, **CentOS**, **AlmaLinux**, **Rocky Linux**, and **openSUSE**.  Akamai +2
- **nftables / iptables (Raw/Direct):** Some minimal or server-focused installations manage rules directly via the `nft` or `iptables` CLI tools rather than a high-level wrapper.



UFW (Uncomplicated Firewall)

- Status: `sudo ufw status`
- Enable: `sudo ufw enable`
- Disable: `sudo ufw disable`
- Block a specific port: `sudo ufw deny 80`
- Block a specific IP: `sudo ufw deny from X.X.X.X to any`
- Delete a rule: `sudo ufw delete ...`
- Reset rules: `sudo ufw reset`
- *(It's actually pretty "uncomplicated" - esp. for a Canonical product)*



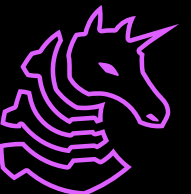
firewalld

- Linux utilizes “daemons”, background processes
- firewalld, when enabled, can be configured in CLI
- [/usr/lib/firewalld/zones/](#)
- Firewall configuration allows for separating a network into zones, provided a default zone, here is where you add interface zones and more



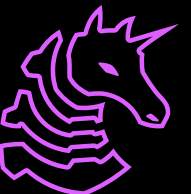
Host Firewall Best-Practices

- Generally, the NGFW (entrypoint to the network) should be prioritized over host-level firewalls
- However, in assumed-breach scenarios, host firewalls can be very helpful in deterring lateral movement
- It's key to have a good idea of the services/ports you expect to be running/open on a given box
- Ideally, set up default-deny (whitelist), with manual exclusions for ^
- Blacklisting sucks, but I guess it can buy you time



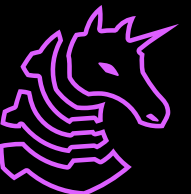
IP Banning

- Disclaimers:
 - This is temporary! Attackers can switch IPs/IP ranges pretty easily. Only use when you want to buy yourself some time.
 - Again, prioritize the NGFW (banning across the network is more effective)



Host IP Banning

- Disclaimers:
 - This is temporary! Attackers can switch IPs/IP ranges pretty easily. Only use when you want to buy yourself some time.
 - Again, prioritize the NGFW (banning across the network is more effective)
- Using iptables (my preference):
 - Add banned IP: `sudo iptables -A INPUT -s xxx.xxx.xxx.xxx -j DROP`
 - List banned IPs: `sudo iptables -L -v -n`
 - Remove banned IP: `sudo iptables -D INPUT -s xxx.xxx.xxx.xxx -j DROP`
- Using ufw:
 - `sudo ufw deny from X.X.X.X to any`



Windows



Containerization



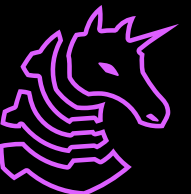
Scenario

- You want to run an **untrusted application**
 - The last cybersecurity team was incompetent, and built/deployed a bunch of insecure applications that you want to quickly patch up
 - You're told (e.g., by an inject) to deploy an application that they provide, that you don't have time to properly verify and fix security vulnerabilities for
- The application has to run as root, since it needs to use files in /etc/my-application (this is okay)
- What if, as soon as the application starts, it also reads /etc/shadow and immediately sends it to a remote server? (this is not okay)



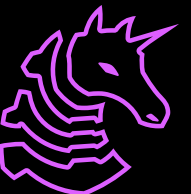
Docker

- Isolates applications into “containers” which are (for the most part) independent
- Applications specifically devoted to the container are quick to run
- The idea is you build containers “once” and you run them anywhere
- We use this in security to quickly sandbox applications
 - Providing us further isolation from a host’s operating system



Containers vs. Virtual Machines (VMs)

- Virtual Machines fully emulate hardware, software, and network components otherwise found in a physical machine and are drawn from a single physical server
- Containers run from the hardware level and borrow overhead from a host's operating system to present a lightweight option



Containers vs. VMs

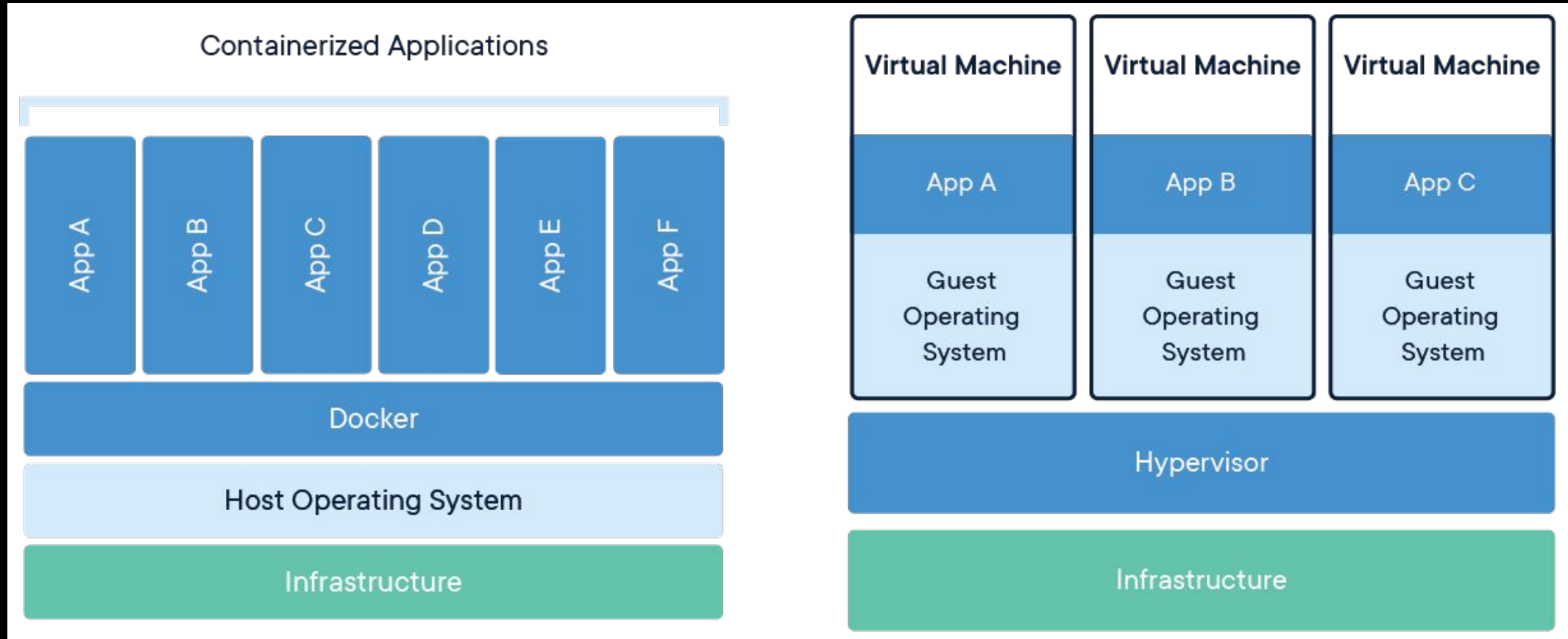


Image Credit: [Docker](#)



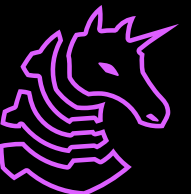
Docker Security

- [Docker Security Tips](#)
- One primary risk with running Docker containers is that the default set of capabilities and mounts given to a container may provide **incomplete isolation**, either independently, or when used in combination with kernel vulnerabilities
- Docker will most likely “work” for a quick sandbox
- However, it’s far from perfect (when it comes to security)



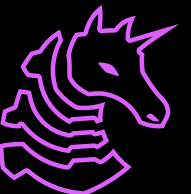
Docker Security

- <https://docs.docker.com/engine/security/>
- Running containers (and applications) with Docker implies running the Docker daemon. This daemon requires root privileges unless you opt-in to Rootless mode...
- Docker allows you to share a directory between the Docker host and a guest container; and it allows you to do so without limiting the access rights of the container. This means that you can start a container where the /host directory is the / directory on your host; and the container can alter your host filesystem without any restriction



Firejail

- A security-focused sandboxing tool to run untrusted applications
- Provides fine-grained control over what the application can do
- This is done through “profiles,” which are just files with some custom syntax
- For example, `my-app.profile`



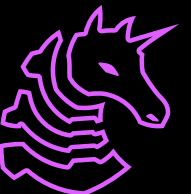
Firejail Profiles: Blacklist

```
blacklist /etc/shadow # No access to /etc/shadow
```



Firejail Profiles: Blacklist

```
blacklist /etc/shadow # No access to /etc/shadow
blacklist /etc/passwd
blacklist /etc/pam.d
blacklist /var/log
blacklist /bin/sh
blacklist /bin/bash
blacklist /bin/zsh
blacklist /usr/bin/sh
...
```



Firejail Profiles: Whitelist

```
blacklist /  
whitelist /etc/my-application
```

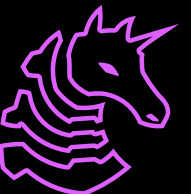


Firejail Profiles: Whitelist

```
blacklist /
```

```
whitelist /etc/my-application
```

- This will create a new temporary in-memory filesystem (tmpfs)
- It will “bind mount” the whitelisted directory into the tmpfs
 - This is similar to Docker’s bind mount
- Any edits are copied over to the host filesystem in real time



Firejail Profiles: Read-Only?

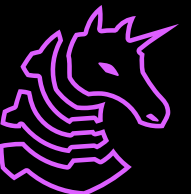
```
blacklist /  
whitelist /etc/my-application  
read-only /etc/my-application
```



Firejail Profiles: Read-Only?

```
blacklist /  
whitelist /etc/my-application  
read-only /etc/my-application
```

- This will still create a tmpfs and bind mount the directory
- However, the directory will be mounted read-only

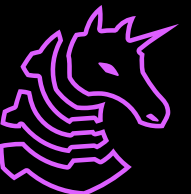


Firejail Profiles: Read-Only?

```
blacklist /  
whitelist /etc/my-application  
read-only /etc/my-application
```

- This will still create a tmpfs and bind mount the directory
- However, the directory will be mounted read-only

What if the application needs to write to its directory, but we don't want it to affect the host filesystem?



Firejail Profiles: Private

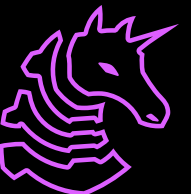
```
blacklist /  
private-etc my-application
```

- This will still create a tmpfs, but will not bind-mount the directory
- Instead, the directory will be “copied” to the tmpfs
- Any modifications to the tmpfs will be discarded at the end (so the host filesystem isn’t affected at all)
- Very common in real-world firejail profiles



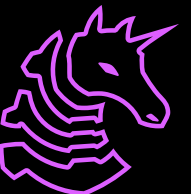
Firejail Profiles: Resource Limits

```
rlimit-as 4000000000 # Memory limit: 4GB  
rlimit-fsize 31457280 # Maximum created file size: 30MB  
rlimit-nofile 500 # Maximum number of open files  
rlimit-nproc 10 # Maximum number of spawned processes
```



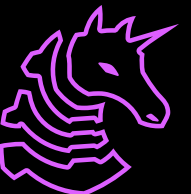
Firejail Profiles: Other/Miscellaneous

```
caps.drop all    # Disable all “Linux Kernel” capabilities
newprivs         # Disable use of SUID binaries (sudo)
nogroups         # Disable groups (sudo, wheel, dialout, ...)
noroot           # The root account no longer exists (in sandbox)
x11 none         # No control of the display manager (X11)
nodvd
nosound
notv
novideo
no3d
```



Another Option: Bubblewrap

```
bwrap --ro-bind /usr /usr
      --dir /tmp
      --dir /var
      --symlink ../tmp var/tmp
      --proc /proc
      --dev /dev
      --ro-bind /etc/resolv.conf /etc/resolv.conf
      --symlink usr/lib /lib
      --symlink usr/lib64 /lib64
      --symlink usr/bin /bin
      --symlink usr/sbin /sbin
      --chdir /
      --unshare-all
      --share-net
      --die-with-parent
      my-application
```



Windows System Hardening

Document is in Purple general



Next Meetings

2026-10-22 • This Thursday

- Red vs. Blue Exercise!
- Practice your system administration skills



Meeting content can be found at
sigpwny.com/meetings.

