SIGPwny

Purple Team FA2025 • 2025-09-25

Firewalls, Web Defense, Containerization

Suchit Bapatla, Krishnan Shankar, Michael Khalaf

Announcements

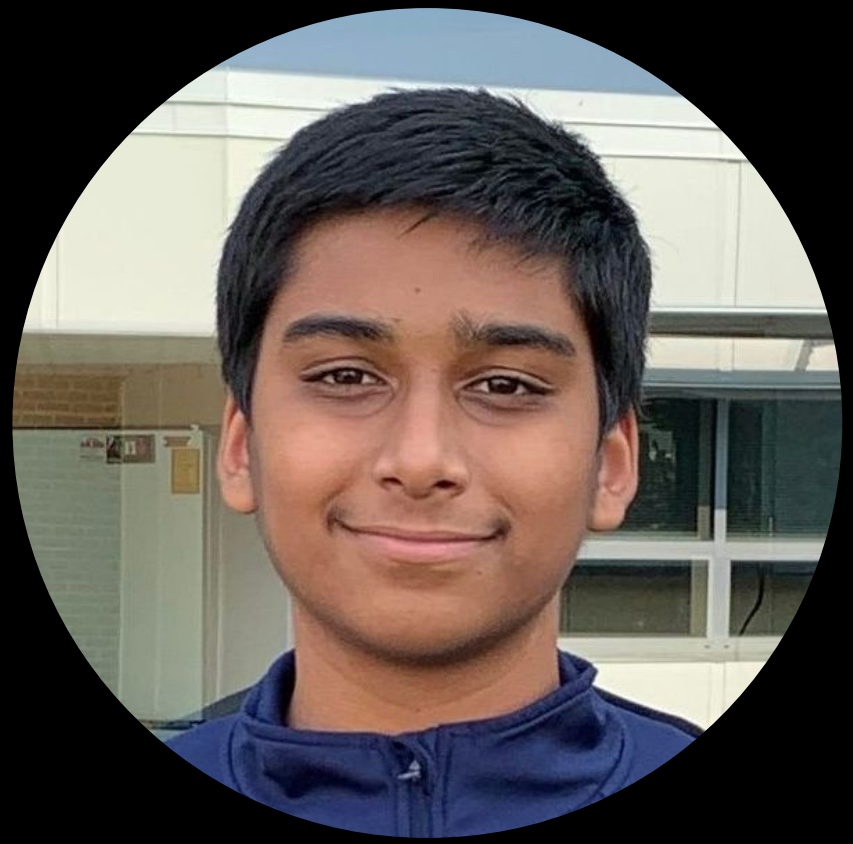
- CyberForce 2025: how many of you would like to opt in as a member for a possible 2nd team?
- If so, contact Michael immediately.
- Send me:
  - Name, .edu email, non .edu email, phone number, year in school, major, minor (if applicable), shirt size, dietary restrictions

Suchit Bapatla

- Helper
- CS MCS
- I did Mock Trial in HS

Krishnan Shankar

- SIGPwny Helper
- Computer Engineering ‘28
- Fun fact: I love planespotting

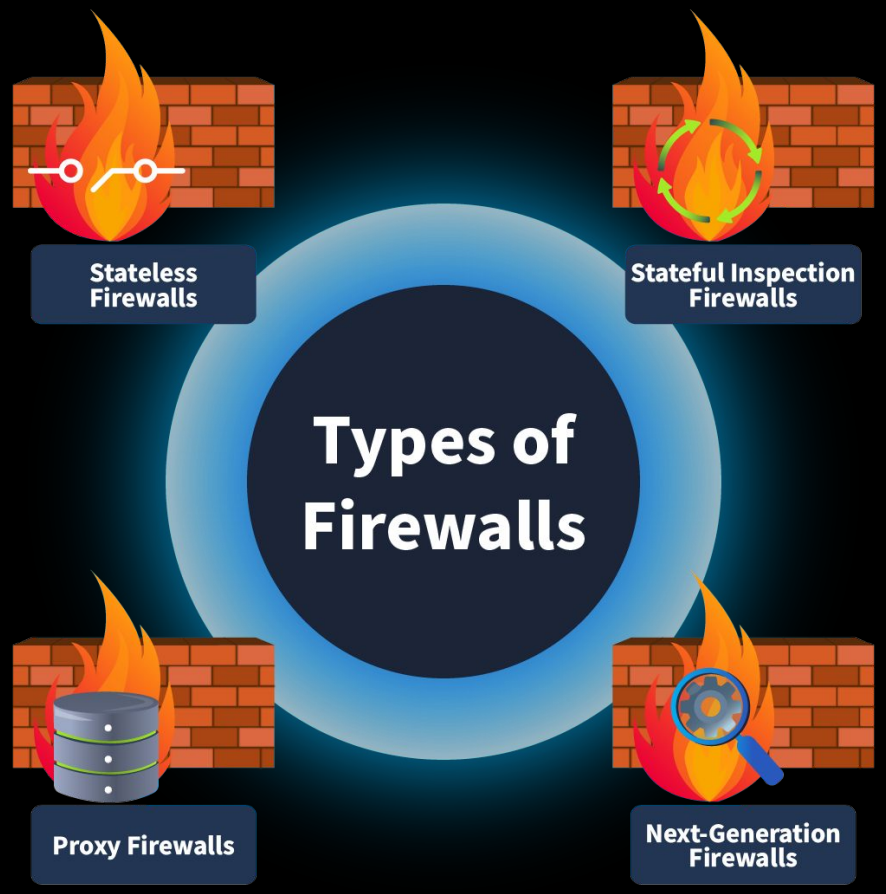


ctf.sigpwny.com

sigpwny{fiya_wall_bl0cked_ya}

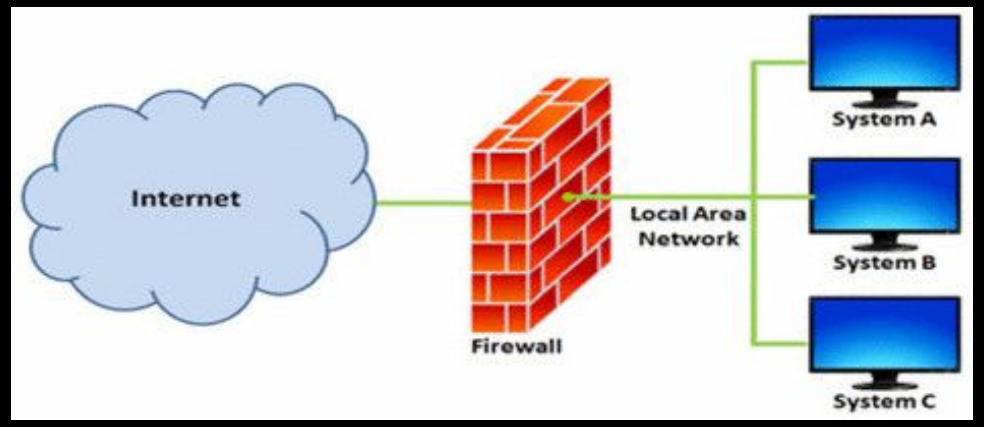
What is a Firewall

- Monitors incoming and outgoing network traffic to look for any suspicious activities
- Proxy firewalls operate on layer 7
- Next-gen firewalls operate on levels 3 and 7 with heuristic analysis



What is a Firewall

- When asking the question, “how does a blue teamer defend against malicious traffic in a live setting” there is one reliable option for live detection
- Firewalls are the technical utility to achieve this
- They are locks on doors between (hopefully) segmented networks (in many cases, VLANs)
- They are both GUI and command line accessible
- Capable of inspection of packets before forwarding them to a LAN system (maybe a gateway) prior to a dedicated recipient client LAN



Firewall Diagram

How does it work?

- Generally speaking, a firewall implemented within a network is responsible for enforcement of access to other portions of a network
- Prioritize rules at the top for those traffic types you will have more frequently over a network
- Firewalls process rules from top to bottom
- Implicitly firewalls are set to deny any-any, must remember to validate this
- Rules are written into the interface
- Firewalls can have multiple interfaces, just like a router

Firewall Rules

- Source and Destination Addresses
- Ports
- Protocol
- Action
- Direction

- Rules help monitoring and filtering of packets both ingress and egress traffic

Action	Source	Destination	Protocol	Port	Direction
Allow	192.168.1.0/24	Any	TCP	80	Outbound

Linux Firewall

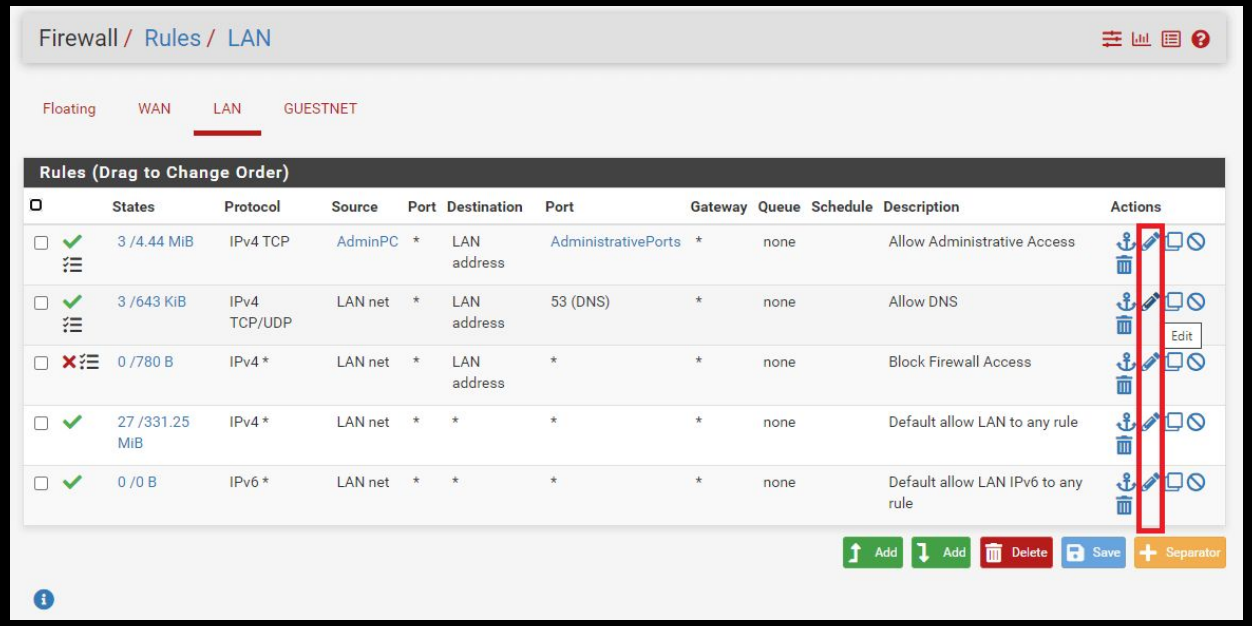
- Netfilter-based
  - iptables: Provides various functionalities to control network traffic
  - nftables: A successor to the “iptables” utility, with enhanced packet filtering and NAT capabilities
  - firewalld: Has predefined rule sets, works a bit differently from the others and comes with different pre-built network zone configurations.
- Uncomplicated Firewall (ufw)
  - Enable UFW: `sudo ufw enable`
  - Disable UFW: `sudo ufw disable`
  - Reset UFW rules: `sudo ufw reset`
  - Block a specific port: `sudo ufw deny 80`

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

Pfsense

- A free open-source firewall and router software
- Based on FreeBSD
- Can prioritize different types of traffic, top (high) to bottom (low)
- Walks through a step by step process
- You care about WAN and LAN for public facing and private facing networks (your 192s, 172s, 10s, etc)



Slightly advanced attacks using NMAP

- `nmap -sS -Pn -D 10.10.10.1,10.10.10.2,ME -F MACHINE_IP`
- `-sS` is stealth mode `-Pn` continues even if no reply received, `-F` scans the hundred most used ports
- `-D` uses decoys and is exceptionally useful as it alternates between provided IP's so there is less detection by a firewall due to multiple IP's
- `nmap -sS -Pn -D RND,RND,ME -F MACHINE_IP` assigns random decoy IP addresses
- Note: the amount of decoys you use increases the number of messages sent
- `nmap -sS -Pn --proxies PROXY_URL -F MACHINE_IP`
  - This uses a proxy so the source IP is hidden

How to detect advanced attacks

- Firewalls won’t always cut it so more advanced Intrusion Detection Systems are needed like Suricata, Security Onion, and Snort
- More on those tools later but they can be used to look for outliers in data to find anomalies like port scans
- A layered defense will almost always lay more reliance on firewalls, a severely underrated defense mechanism
  - But when paired with SIEM and IDS tools + logging, you not only have a way to respond to suspected malicious traffic, you can further analyze it
  - The only drawback: your defense is as good as your ability to configure it and account for traffic types you want or don’t want
  - Rely on your default deny

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

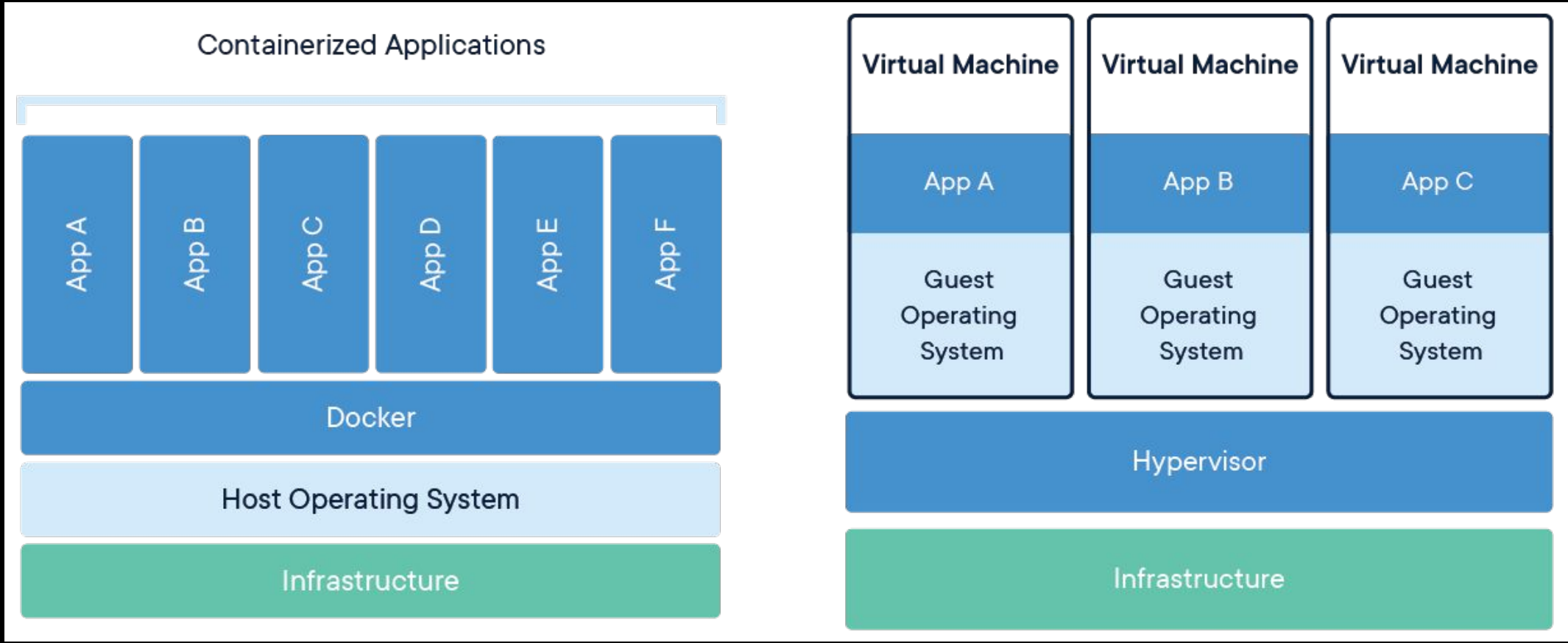


Image Credit: Docker

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
nonewprivs  # 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
```

Demonstration (plan)

Scripting IPTables and dropping a ip based on detection

Plan: find a malicious IP

5 minutes

→ IP blacklisting ←

3 minutes

→ Attempt communication after iptables DROP

2 minutes

→ Explain firewall’s role in live detection to reduce risk

Next Meetings

2025-09-30 • This Tuesday

- Linux & Linux Privilege Escalation
- Navigate Linux and learn how privileges escalate.

2025-10-02 • Next Thursday

- Linux & Linux Forensics
- Checking out investigative aspects of Linux

2025-10-07 • Next Tuesday

- Windows & Windows Privilege Escalation
- Navigate Windows and how privileges escalate

ctf.sigpwny.com

sigpwny{fiya_wall_bl0cked_ya}

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

SIGPwny