



**CSCE 5585: Advanced Network Security**

# **Part A - Attack Vectors and Threat Intelligence: Modern Challenges and Practical Defenses**

Cybersecurity Research Overview

**Project By:**

Group 9

Aparna Singh | Durga Shankar Dalayi | Nikhil Ethamukkala

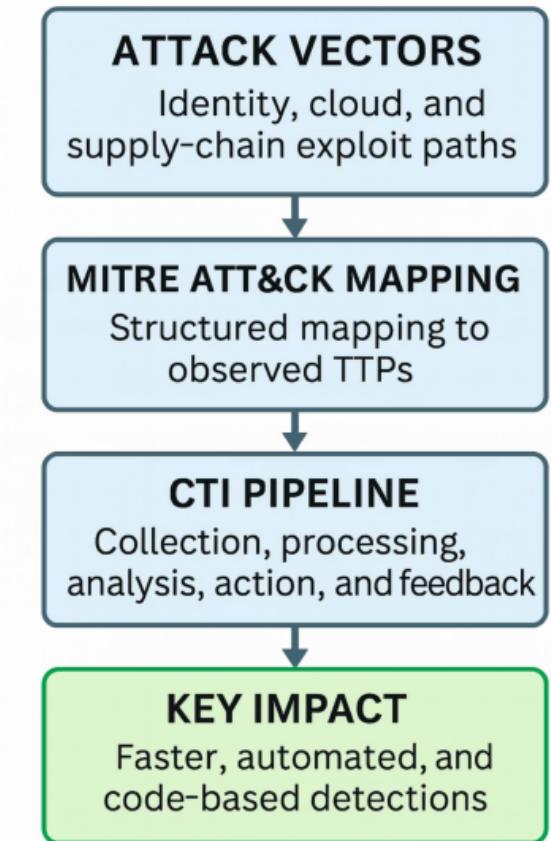
University of North Texas

# Introduction & Problem Statement

Organizations of today are dealing with changing attack vectors which include sophisticated phishing attacks as well as cloud-native privilege abuse and compromise of IOT devices. Static defense methods have proven ineffective in defending against the dynamic nature of an ever-changing attacker.

The evolution of Cyber Threat Intelligence (CTI), is beyond the use of unstructured, ad hoc indicators. CTI now is structured and can be automated for effective mapping of adversaries' behaviors to the MITRE ATT&CK framework for the benefit of a more comprehensive defense strategy

## OVERVIEW



## ATTACK INTELLIGENCE

# Major Attack Vectors (2020-2025)

## Phishing, BEC & Social Engineering

- Credential harvesting
- OAuth abuse
- MFA fatigue
- Deepfake voice
- QR-code phishing

## Ransomware & Double/Triple Extortion

- Initial access brokers
- Lateral movement
- Data exfiltration

## Supply-Chain Compromise

- Software update channels
- CI/CD credentials
- Third-party components

## Cloud-Native Abuse

- Misconfiguration
- Token theft
- IMDS abuse
- Privilege escalation

## Web & Application Exploits

- RCEs
- Deserialization
- SSRF
- SQLi

## IoT/OT & Edge

- Weak authentication
- Unpatched firmware
- Flat networks

# CTI Lifecycle and Pipeline



## 1. Requirements & Collection

- Define decision-makers and use-cases
- Gather from internal telemetry, honeypots, external OSINT, commercial feeds, ISACs/ISAOs, CISA KEV

## 5. Feedback & Measure

- Monitor SOC analyst feedback
- Track MTTD/MTTR
- Measure ATT&CK coverage

## 2. Processing & Normalization

- Remove duplicates
- Enrich with WHOIS/pассиве DNS/ASN/geolocation
- Score indicators
- Convert to STIX objects

## 3. Analysis & Context

- Connect indicators to campaigns, malware families, ATT&CK techniques
- Extract TTPs from reports

## 4. Dissemination & Action

- Publish to TIP
- Distribute to SIEM/XDR, EDR/NGAV, SOAR, Cloud controls

# Reference Architecture

## CTI-Driven Security Architecture

- **Unified Threat Intelligence:** Centralizes CTI into SIEM, EDR/XDR, SOAR, and cloud tools for ATT&CK-aligned detections.
- **Multi-Source Visibility:** Collects DNS, EDR, identity, email, cloud logs + OSINT & KEV feeds for complete attack coverage.
- **Threat Intelligence Platform (TIP):** Automates STIX/TAXII ingest, IOC scoring/decay, and maps indicators to TTPs.
- **Analytics & Automation:** SIEM correlation + UEBA identity detection + automated SOAR playbooks for fast containment.
- **Cloud & Identity Security:** CSPM/CNAPP, JIT access, workload identities, and secrets scanning reduce cloud attack surface.
- **Outcome:** Faster detection (low MTTD), rapid response (low MTTR), and reduced overall attack blast radius.

# Conclusion & Future Directions

## Key Findings

- Attack vectors have shifted toward identity and supply-chain leverage while maintaining social engineering as dominant entry point.
- Effective defense operationalizes CTI, converting raw indicators into mapped behaviors, detections-as-code, and repeatable response playbooks.
- Measurable outcomes anchor programs to risk and reveal investment priorities.



## Future Work

- Graph-based campaign clustering and infrastructure lineage identification.
- Automated SBOM + VEX integration in build/deploy pipelines.
- Federated/privacy-preserving sharing among peers and ISACs.
- Machine Learning/LLM assistance for report triage and enrichment.
- Transition to memory-safe languages and hardware-backed isolation.



# References

1. Tounsi, W., & Rais, H. (2018). *A survey on technical threat intelligence in the age of sophisticated cyber attacks*. Computers & Security, 72, 212–233. <https://doi.org/10.1016/j.cose.2017.09.001>
2. Wagner, T. D., Mahbub, K., Palomar, E., & Abdallah, A. E. (2019). *Cyber threat intelligence sharing: Survey and research directions*. Computers & Security, 87, Article 101589 <https://doi.org/10.1016/j.cose.2019.101589>
3. Verizon. (2025). *2025 Data Breach Investigations Report*. Verizon Enterprise Solutions.  
<https://www.verizon.com/business/resources/reports/dbir/>
4. European Union Agency for Cybersecurity. (2023, October 19). *ENISA Threat Landscape 2023*.  
<https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023>
5. National Institute of Standards and Technology. (2024, February 26). *The NIST Cybersecurity Framework (CSF) 2.0*.  
<https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf>



# **CSCE 5585: Advanced Network Security**

## **Part B - Secure Network Design and Implementation**

**Project By:**

**Group 9**

Aparna Singh | Durga Shankar Dalayi | Nikhil Ethamukkala

University of North Texas

# Project Overview

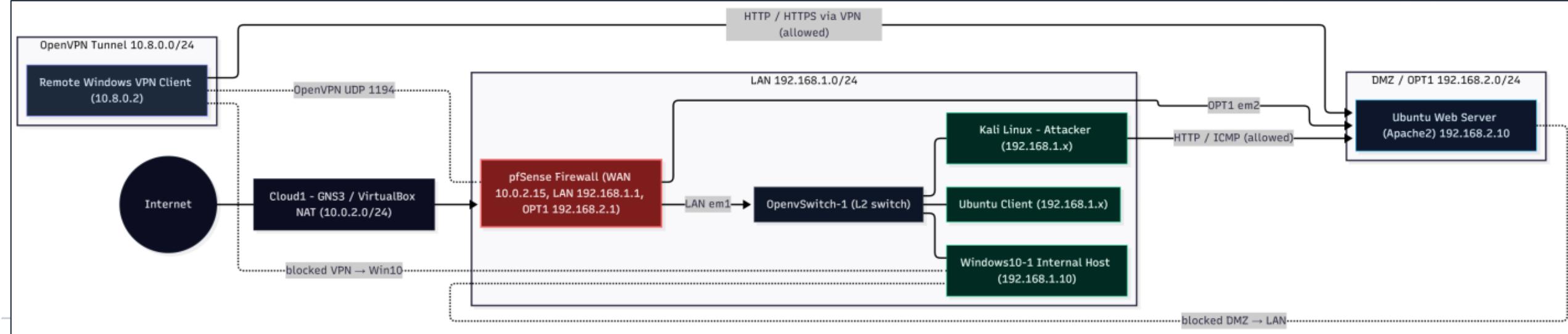
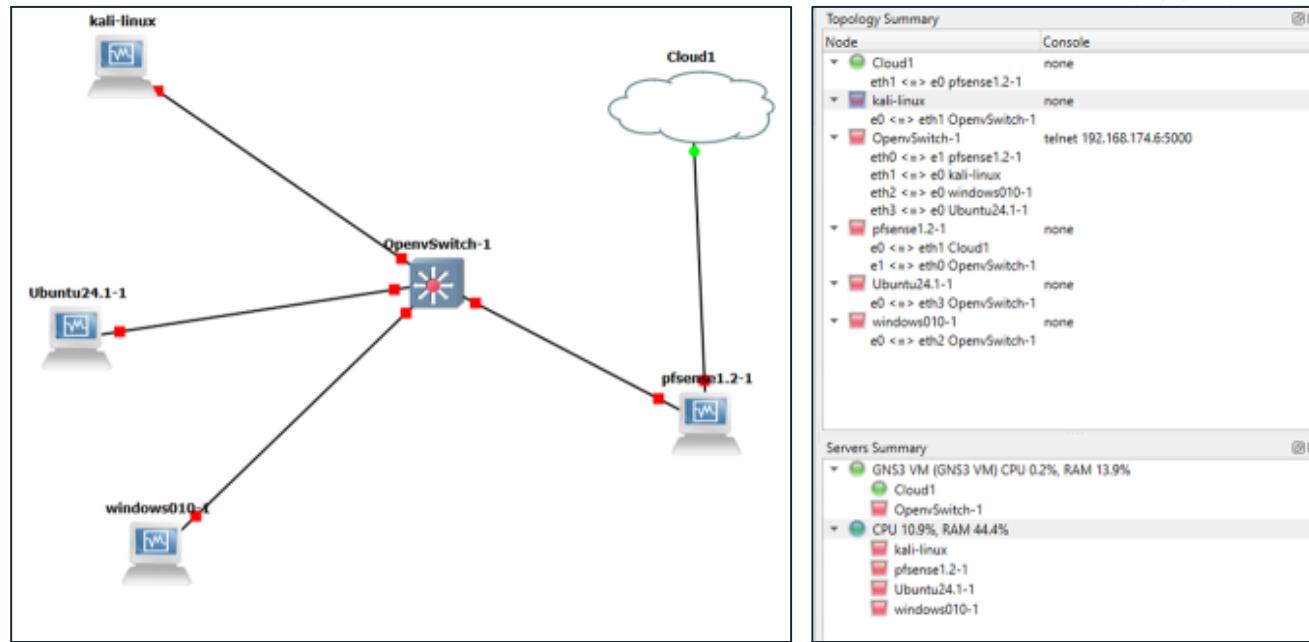
- Designed a secure, multi-segment network using pfSense with **LAN**, **DMZ**, and **WAN** separation.
- Configured **stateful firewall rules**, Hybrid NAT, and strict segmentation to control traffic between networks.
- Deployed **Suricata IDS/IPS** in inline mode on LAN & DMZ to monitor and block malicious activity.
- Implemented **OpenVPN remote-access** solution using certificate-based authentication and a dedicated tunnel network.
- Validated VPN functionality by accessing DMZ services while ensuring LAN access remained restricted.
- Performed **attack simulations** (SYN flood, ICMP flood, Slowloris, Nmap scanning, Hydra brute-force) to evaluate defensive controls.
- Verified detection & alerting via Suricata and confirmed enforcement of firewall rules and segmentation policies.

## Technology Stack



# Deployed Network Topology

- Built a multi-segment virtual network in GNS3 consisting of **Kali**, **Ubuntu (DMZ server)**, **Windows 10**, and **pfSense**.
- All internal hosts connect through **Open vSwitch**, forming the 192.168.1.0/24 LAN.
- **pfSense** serves as the security gateway with **WAN ↔ Cloud** connectivity and **LAN/OPT1 (DMZ) segmentation**.
- The Ubuntu server is positioned in the DMZ, while Kali and Windows reside in the LAN for internal and attacker-role testing.
- This topology mirrors a real enterprise layout, supporting segmentation between LAN, DMZ, and external networks.



# Firewall Configuration

```
.168.1.100 (Local Database)

FreeBSD/amd64 (pfSense.lab.local) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 03259eca7731c596fd09

*** Welcome to pfSense 2.8.1-RELEASE (amd64) on pfSense ***

WAN (wan)  -> em0 -> v4/DHCP4: 10.0.2.15/24
LAN (lan)  -> em1 -> v4: 192.168.1.1/24
OPT1 (opt1) -> em2 -> v4: 192.168.2.1/24

0) Logout / Disconnect SSH          9) pfTop
1) Assign Interfaces                10) Filter Logs
2) Set interface(s) IP address      11) Restart GUI
3) Reset admin account and password 12) PHP shell + pfSense tools
4) Reset to factory defaults        13) Update from console
5) Reboot system                   14) Enable Secure Shell (sshd)
6) Halt system                     15) Restore recent configuration
7) Ping host                       16) Restart PHP-FPM

Enter an option: █
```

- pfSense automatically detected and initialized three interfaces: **WAN (em0)**, **LAN (em1)**, and **OPT1 (em2)** during boot.
- The WAN interface was assigned a DHCP address (**10.0.2.15/24**) by the VirtualBox NAT adapter, giving the firewall external connectivity.
- We configured the LAN interface as **192.168.1.1/24** to serve as the default gateway for all internal hosts.
- We also configured OPT1 as **192.168.2.1/24**, designating it as the DMZ segment in our network.
- The console view confirms successful interface assignment and establishes the base routing required for further firewall, NAT, and segmentation setup.

# Firewall Rules

## WAN Rules

| States  | Protocol | Source                        | Port | Destination           | Port | Gateway | Queue | Schedule | Description                        | Actions |
|---------|----------|-------------------------------|------|-----------------------|------|---------|-------|----------|------------------------------------|---------|
| ✗ 0/0 B | *        | RFC 1918 networks             | *    | *                     | *    | *       | *     | *        | Block private networks             |         |
| ✗ 0/0 B | *        | Reserved Not assigned by IANA | *    | *                     | *    | *       | *     | *        | Block bogon networks               |         |
| ✓ 0/0 B | IPv4 UDP | *                             | *    | WAN address (OpenVPN) | 1194 | *       | none  |          | Allow OpenVPN                      |         |
| ✓ 0/0 B | IPv4 UDP | *                             | *    | WAN address (OpenVPN) | 1194 | *       | none  |          | OpenVPN OpenVPN-Server-Cert wizard |         |

## LAN Rules

| States         | Protocol | Source | Port        | Destination | Port | Gateway | Queue | Schedule | Description                        | Actions |
|----------------|----------|--------|-------------|-------------|------|---------|-------|----------|------------------------------------|---------|
| ✓ 3/1.30 MiB   | *        | *      | *           | LAN Address | 443  | *       | *     |          | Anti-Lockout Rule                  |         |
|                |          |        |             |             | 80   |         |       |          |                                    |         |
| ✓ 36/22.88 MiB | IPv4     | *      | LAN subnets | *           | *    | *       | *     | none     | Default allow LAN to any rule      |         |
| ✓ 0/0 B        | IPv6     | *      | LAN subnets | *           | *    | *       | *     | none     | Default allow LAN IPv6 to any rule |         |

## OPT1 Rules

| States       | Protocol  | Source       | Port         | Destination          | Port        | Gateway | Queue | Schedule | Description                      | Actions |
|--------------|-----------|--------------|--------------|----------------------|-------------|---------|-------|----------|----------------------------------|---------|
| ✓ 0/47 KiB   | IPv4 TCP  | OPT1 subnets | *            | This Firewall (self) | 53 (DNS)    | *       | none  |          | Allow DMZ DNS                    |         |
| ✓ 0/5 KiB    | IPv4 ICMP | OPT1 subnets | *            | This Firewall (self) | *           | *       | none  |          | Allow DMZ ping to firewall       |         |
| ✓ 0/1.70 MiB | IPv4      | *            | OPT1 subnets | *                    | *           | *       | *     | none     | Allow DMZ Outbound internet      |         |
| ✗ 0/840 B    | IPv4      | *            | OPT1 subnets | *                    | LAN subnets | *       | *     | none     | block DMZ access to LAN          |         |
| ✓ 0/0 B      | IPv4 TCP  | LAN subnets  | *            | 192.168.2.10         | 80 (HTTP)   | *       | none  |          | Allow LAN to DMZ Web Server HTTP |         |

### WAN Rules:

- Blocks all unsolicited inbound traffic;
- RFC1918 & Bogon filtering enabled;
- OpenVPN 1194 allowed.

### LAN Rules:

- LAN fully allowed outbound;
- Anti-Lockout on 443;
- LAN → DMZ webserver (192.168.2.10:80) permitted.

## Outbound NAT Rules

| Mode   | Automatic outbound NAT rule generation.<br>(IPsec passthrough included) | Hybrid Outbound NAT rule generation.<br>(Automatic Outbound NAT + rules below) | Manual Outbound NAT rule generation.<br>(AON - Advanced Outbound NAT) | Disable Outbound NAT rule generation.<br>(No Outbound NAT rules) |             |          |             |                  |                              |
|--|---|--|---|--|-------------|----------|-------------|------------------|------------------------------|
| <input checked="" type="radio"/>                     |   | <input checked="" type="radio"/>   |   |  |             |          |             |                  |                              |
| <b>Save</b>  |   |  |   |  |             |          |             |                  |                              |
| <b>Mappings</b>                                      |   |  |   |  |             |          |             |                  |                              |
| Interface  | Source  | Source Port  | Destination   | Destination Port   | NAT Address | NAT Port | Static Port | Description      | Actions                      |
| <input checked="" type="checkbox"/> WAN              | 192.168.2.0/24  | *  | *   | *  | WAN address | *        | *           | DMZ outbound NAT |                              |
|  |   |  |   |  |             |          |             |                  |                              |
| <b>Automatic Rules</b>                               |   |  |   |  |             |          |             |                  |                              |
| Interface  | Source  | Source Port  | Destination   | Destination Port   | NAT Address | NAT Port | Static Port | Description      |                              |
| <input checked="" type="checkbox"/> WAN              | 127.0.0.0/8 ::1/128   | 192.168.1.0/24   | 192.168.2.0/24  | *  | *           | 500      | *           | ✓                | Auto created rule for ISAKMP |
| <input checked="" type="checkbox"/> WAN              | 127.0.0.0/8 ::1/128   | 192.168.1.0/24   | 192.168.2.0/24  | *  | *           | *        | *           | ✓                | Auto created rule for ISAKMP |
| Activate Window<br>Go to Settings to activate Window |   |  |   |  |             |          |             |                  |                              |

### DMZ (OPT1) Rules:

- DMZ allowed DNS & ICMP;
- outbound internet via NAT;
- DMZ → LAN traffic explicitly blocked.

### Outbound NAT:

- Hybrid NAT;
- DMZ subnet 192.168.2.0/24 translated to WAN;
- auto-rules support VPN/ISAKMP.

# Setup Verification

```
ubuntu@Aparna:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=13.7 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=6.53 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=2.74 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=3.16 ms
64 bytes from 192.168.2.1: icmp_seq=5 ttl=64 time=6.94 ms
^C
--- 192.168.2.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4221ms
rtt min/avg/max/mdev = 2.742/6.617/13.723/3.939 ms
ubuntu@Aparna:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=23.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=36.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=25.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=109 time=25.4 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=109 time=22.2 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4011ms
rtt min/avg/max/mdev = 22.198/26.661/36.659/5.154 ms
ubuntu@Aparna:~$
```

```
Session Actions Edit View Help
[kali㉿ kali)-[~]
$ ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
64 bytes from 192.168.2.10: icmp_seq=1 ttl=63 time=6.36 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=63 time=6.20 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=63 time=7.26 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=63 time=9.73 ms
64 bytes from 192.168.2.10: icmp_seq=5 ttl=63 time=6.43 ms
^C
--- 192.168.2.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 6.204/7.196/9.731/1.319 ms

[kali㉿ kali)-[~]
$ curl -v http://192.168.2.10
* Trying 192.168.2.10:80 ...
* Connected to 192.168.2.10 (192.168.2.10) port 80
* using HTTP/1.x
> GET / HTTP/1.1
> Host: 192.168.2.10
> User-Agent: curl/8.15.0
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Date: Fri, 28 Nov 2025 08:13:38 GMT
< Server: Apache/2.4.58 (Ubuntu)
< Last-Modified: Thu, 27 Nov 2025 23:16:14 GMT
< ETag: "29af-6449bb302d469"
< Accept-Ranges: bytes
< Content-Length: 10671
< Vary: Accept-Encoding
< Content-Type: text/html
<
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
<!--
Modified from the Debian original for Ubuntu
Last updated: 2022-03-22
-->
```

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should [replace this file](#) (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the [manual](#) if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|   '-- mods-enabled
|       '-- *.load
|       '-- *.conf
|   '-- conf-enabled
|       '-- *.conf
|-- sites-enabled
    '-- *.conf
```

\* apache2.conf is the main configuration file. It puts the pieces together by including all remaining

## Key Checks Completed:

### ➤ DMZ Gateway Reachability:

Successfully pinged 192.168.2.1, confirming the Ubuntu webserver is correctly connected to the DMZ network.

### ➤ Internet Connectivity from DMZ:

Ping to 8.8.8.8 verified outbound NAT and internet access.

### ➤ Kali → DMZ Host Access:

Kali machine pinged 192.168.2.10 and reached the Apache server over HTTP.

### ➤ Webserver Functionality:

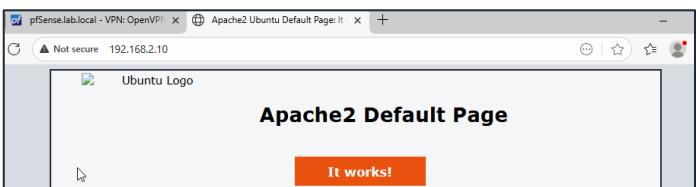
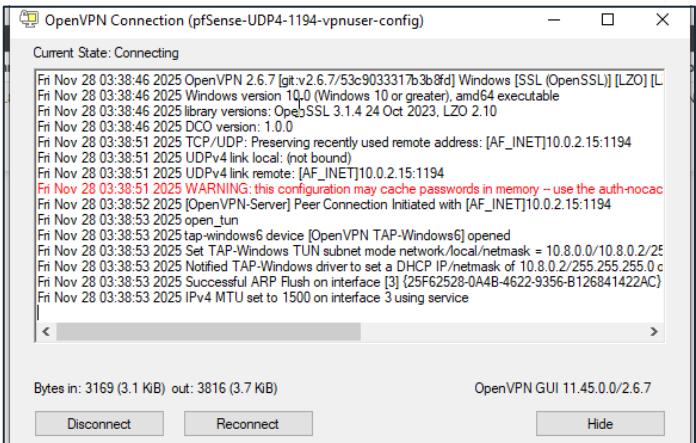
HTTP request to <http://192.168.2.10> returned the Apache2 default page, confirming service availability.

### ➤ Firewall Verification (Implicit):

Traffic flow between LAN, DMZ, and WAN confirmed that pfSense rules and NAT configurations are functioning as intended.

# OpenVPN Configuration & Verification

- OpenVPN server configured on pfSense (UDP/1194, tunnel network: **10.8.0.0/24**).
- VPNuser profile exported via Client Export Utility → confirms certificate + config provisioning.
- Windows client successfully connected, OpenVPN tunnel established, TAP adapter assigned **10.8.0.2**.
- Remote access to DMZ confirmed:
  - VPN user pinged **192.168.2.10** (DMZ webserver) successfully.
  - Apache webpage loaded via **VPN tunnel**, proving routing + firewall allowance.
- LAN access blocked by design:
  - Ping / curl to **192.168.1.10** failed → confirms segmentation & least-privilege policy.
- Firewall rules working as intended:
  - VPN → DMZ allowed
  - VPN → LAN denied
  - NAT & routing correctly forwarding VPN traffic.



```
Aparna>ping 192.162.2.10  
Pinging 192.162.2.10 with 32 bytes of data:  
Reply from 192.162.2.10: bytes=32 time=467ms TTL=229  
Reply from 192.162.2.10: bytes=32 time=500ms TTL=229  
Reply from 192.162.2.10: bytes=32 time=331ms TTL=229  
Reply from 192.162.2.10: bytes=32 time=289ms TTL=229  
Ping statistics for 192.162.2.10:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 289ms, Maximum = 500ms, Average = 396ms  
Aparna>  
Aparna>ping 192.162.1.10  
Pinging 192.162.1.10 with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
Ping statistics for 192.162.1.10:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
Aparna>
```

| OpenVPN Servers |                      |                |  |                     |         |
|-----------------|----------------------|----------------|--|---------------------|---------|
| Interface       | Protocol / Port      | Tunnel Network | Mode / Crypto  | Description         | Actions |
| WAN             | UDP4 / 1194<br>(TUN) | 10.8.0.0/24    | Mode: Remote Access ( SSL/TLS + User Auth )<br>Data Ciphers: AES-256-GCM, AES-128-GCM, CHACHA20-POLY1305, AES-256-CBC<br>Digest: SHA256<br>D-H Params: 2048 bits | OpenVPN-Server-Cert |         |

# Configuring Suricata for Intrusion Detection & Prevention

## Suricata IPS Deployment & Rule Activation

- Suricata enabled in **INLINE IPS mode on LAN and DMZ (OPT1)**
- **Loaded ETOpen Emerging Threats rule set**
- Rules updated successfully and IPS is actively blocking malicious traffic
- System ready to inspect SYN, ICMP, HTTP, Slowloris, Nmap, Hydra traffic

Services / Suricata

Interfaces Global Settings Updates Alerts Blocks Files Pass Lists Suppress Logs View Logs Mgmt SID Mgmt

Sync IP Lists

**Interface Settings Overview**

| Interface  | Suricata Status | Pattern Match | Blocking Mode | Description | Actions |
|------------|-----------------|---------------|---------------|-------------|---------|
| LAN (em1)  |                 | AUTO          | INLINE IPS    | LAN         |         |
| OPT1 (em2) |                 | AUTO          | INLINE IPS    | OPT1        |         |

Services / Suricata / Updates

Interfaces Global Settings Updates Alerts Blocks Files Pass Lists Suppress Logs View Logs Mgmt SID Mgmt

Sync IP Lists

**INSTALLED RULE SET MD5 SIGNATURES**

| Rule Set Name/Publisher          | MD5 Signature Hash               | MD5 Signature Date              |
|----------------------------------|----------------------------------|---------------------------------|
| Emerging Threats Open Rules      | 7ca1ee0a2dbaca189bd1cbb565251137 | Tuesday, 18-Nov-25 16:47:16 CST |
| Snort Subscriber Rules           | Not Enabled                      | Not Enabled                     |
| Snort GPLv2 Community Rules      | 336d8584f0fe0daf071da84b66b6819b | Tuesday, 18-Nov-25 16:47:17 CST |
| Feodo Tracker Botnet C2 IP Rules | Not Enabled                      | Not Enabled                     |
| ABUSE.ch SSL Blacklist Rules     | Not Enabled                      | Not Enabled                     |

Services / Suricata / Global Settings

Interfaces Global Settings Updates Alerts Blocks Files Pass Lists Suppress Logs View Logs Mgmt SID Mgmt

Sync IP Lists

**Please Choose The Type Of Rules You Wish To Download**

**Install ETOpen Emerging Threats rules**  ETOpen is a free open source set of Suricata rules whose coverage is more limited than ETPro.  Use a custom URL for ETOpen downloads

Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETOpen rules.

**Install ETPro Emerging Threats rules**  ETPro for Suricata offers daily updates and extensive coverage of current malware threats.  Use a custom URL for ETPro rule downloads

The ETPro rules contain all of the ETOpen rules, so the ETOpen rules are not required and are disabled when the ETPro rules are selected. Sign Up for an ETPro Account. Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETPro rules.

**Install Snort rules**  Snort free Registered User or paid Subscriber rules  Use a custom URL for Snort rule downloads

[Sign Up for a free Registered User Rules Account](#) [Sign Up for paid Snort Subscriber Rule Set \(by Talos\)](#)

Enabling the custom URL option will force the use of a custom user-supplied URL when downloading Snort Subscriber rules.

**Install Snort GPLv2 Community rules**  The Snort Community Ruleset is a GPLv2 Talos-certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions.  Use a custom URL for Snort GPLv2 rule downloads

This ruleset is updated daily and is a subset of the subscriber ruleset. If you are a Snort Subscriber Rules customer (paid subscriber), the community ruleset is already built into your download of the Snort Subscriber rules, and there is no benefit in adding this rule set separately.

**Install Feodo Tracker Botnet C2 IP rules**  The Feodo Botnet C2 IP Ruleset contains Dridex and Emotet/Heodo botnet command and control servers (C&Cs) tracked by Feodo Tracker.

# Attack Simulations on the Network

**SYN Flood**

**ICMP Flood**

**Nmap Full Scan**

**Slowloris**

**SlowHTTPTest**

**Packets: 14787 - Displayed: 12 (0.1%)**

**Profile: Default**

# Suricata Detection results

- Suricata successfully detected **multiple attack types** generated from Kali.
- Alerts triggered for:
  - **SYN traffic anomalies**
  - **ICMP flood patterns**
  - **HTTP protocol irregularities (Slowloris indicators)**
- Each alert includes **source, destination, ports, protocol, and GID:SID** for rule mapping.
- IPS (Inline mode) inspected traffic in real time and flagged packets violating protocol behavior.
- Results confirm Suricata is actively monitoring and enforcing security at both **LAN** and **DMZ** boundaries.

| Alert Log View Filter  |        |     |       |                                 |               |       |               |       |
|--|--------|-----|-------|---------------------------------|---------------|-------|---------------|-------|
| Last 250 Alert Entries. (Most recent entries are listed first) |        |     |       |                                 |               |       |               |       |
| Date   | Action | Pri | Proto | Class                           | Src           | SPort | Dst           | DPort |
| 11/28/2025<br>04:16:59   | ⚠️     | 3   | TCP   | Generic Protocol Command Decode | 192.168.2.10  | 80    | 192.168.1.101 | 34054 |
| 11/28/2025<br>04:16:57   | ⚠️     | 3   | ICMP  | Generic Protocol Command Decode | 192.168.2.10  | 0     | 192.168.1.101 | 9     |
| 11/28/2025<br>04:16:57   | ⚠️     | 3   | ICMP  | Generic Protocol Command Decode | 192.168.1.101 | 8     | 192.168.2.10  | 9     |
| 11/28/2025<br>04:16:55   | ⚠️     | 3   | ICMP  | Generic Protocol Command Decode | 192.168.2.10  | 0     | 192.168.1.101 | 9     |
| 11/28/2025<br>04:16:55   | ⚠️     | 3   | ICMP  | Generic Protocol Command Decode | 192.168.1.101 | 8     | 192.168.2.10  | 9     |
| 11/28/2025<br>04:10:44   | ⚠️     | 3   | TCP   | Generic Protocol Command Decode | 20.25.227.174 | 443   | 192.168.1.100 | 49843 |
| 11/28/2025<br>04:10:32   | ⚠️     | 3   | TCP   | Generic Protocol Command Decode | 20.25.227.174 | 443   | 192.168.1.100 | 49843 |



# Key Findings



## Network Segmentation

VPNs isolated from DMZ, LAN resources protected.



## Firewall Rules

DMZ reachable, LAN blocked, protected ports monitored.



## Weaklink Captures

Attack behavior detected (VPN Scan, ICMP floods, HTTP requests).



## Hydra Mitigation

SSH brute-force vulnerabilities blocked, credential attacks prevented.



## OpenVPN Validated

Secure remote access and correct IP assignment.



## IPS Active

Malicious traffic inspected, protected hosts secured.



## No IPS Bypasses

Malicious traffic blocked, alerts confirmed in Suricata.



## Rule Updates

Current threat intelligence successfully loaded.



**Overall Assessment:** The environment demonstrates strong isolation, effective threat detection, and reliable VPN security. Firewall and IPS configurations worked cohesively to enforce policy and detect adversarial behavior.

# Conclusion

- The security architecture worked as intended, enforcing clear separation between LAN, DMZ, and VPN networks.
- OpenVPN provided secure, encrypted access while maintaining strict least-privilege controls.
- Suricata effectively detected and alerted on multiple attack types, proving IPS visibility and responsiveness.
- Firewall rules successfully blocked unauthorized access attempts and allowed only permitted flows.
- The overall setup demonstrated a **defense-in-depth** approach with layered protection and validated the network's ability to withstand common attack techniques.

## Challenges Faced

- Creating and isolating LAN, DMZ, and VPN networks was difficult, especially aligning routes, gateways, and access rules without breaking connectivity.
- VMs on Ubuntu and Windows repeatedly lost internet access due to NAT conflicts and pfSense interface instability, requiring multiple fixes and restarts.
- Running Suricata alongside heavy scans (Nmap), floods (ICMP/SYN), and Slowloris attacks caused significant CPU spikes on pfSense, making the firewall slow or unresponsive.
- pfSense occasionally froze, failed to apply rules correctly, or stopped routing properly, forcing full system reboots to restore functionality.



**Thank You!**