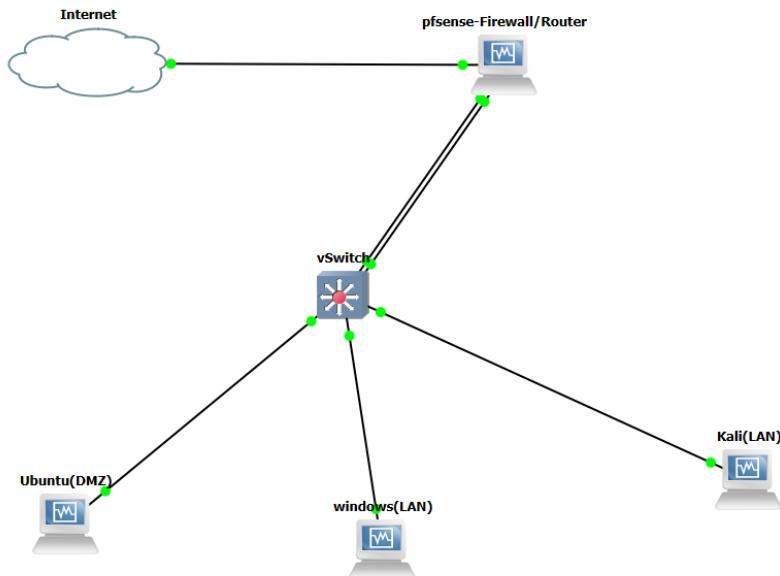


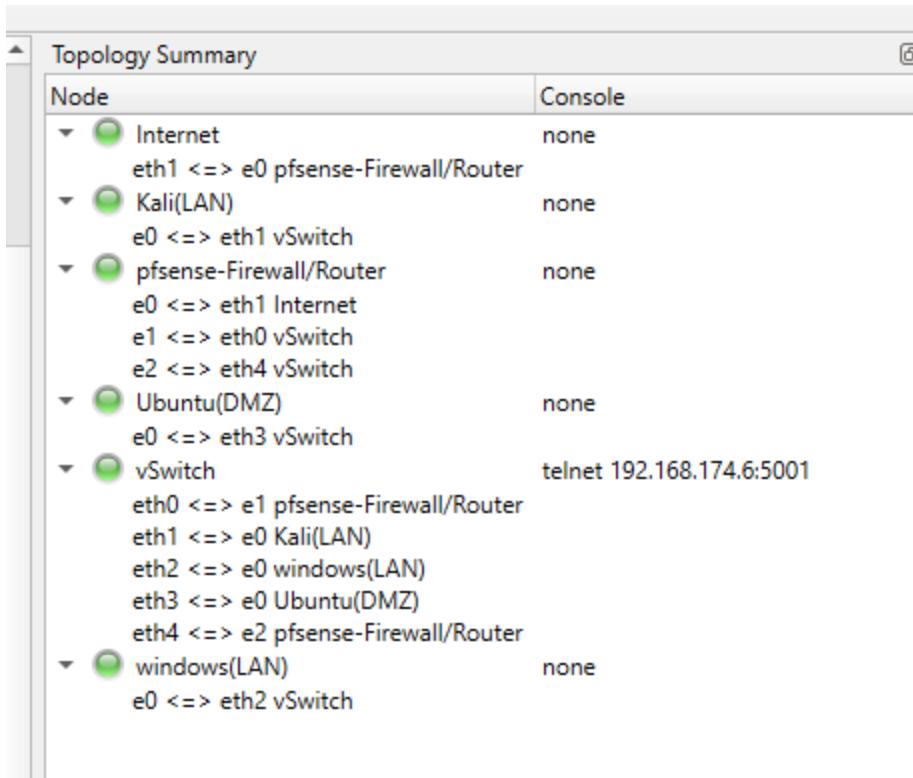
## SECTION 1 — Network Environment Setup

### GNS3 Topology Diagram

#### 1. GNS3 Network Diagram

The GNS3 Network Diagram is an example of a properly designed lab topology used to simulate a small enterprise network comprising of a pfSense router/firewall at the centre of the topology with three separate interfaces; The WAN interface connects the pfSense router to the internet as well as private IP addresses located on the LAN segment, which contains all of the internal devices connected to the LAN segment, including two Kali Linux and Windows Workstations, and one of these is configured with a DMZ interface, which connects to an external user via an Internet WAN cloud. In addition, Virtual Switching provides connectivity between all devices within the topology and verifies that all interfaces are properly configured (i.e. eth0 → WAN, eth1 → LAN, and eth2 → DMZ) through the Topology Summary tab in GNS3. Furthermore, as reflected in this topology design, all public-facing services, Internal users and the Internet are clearly separated, providing the ideal basis for creating and testing a Secure





## SECTION 2 — Network Design & Segmentation

The console menu of pfSense reflects a WAN interface connected to Internet using DHCP (em0), LAN interface using static ip of 192.168.10.0/24, and an Optional interface (em2) of 192.168.20.0/24. Later, Optional interface was named DMZ to represent fact that it is now holding devices with lower-trust levels devices connected to LAN. With three network interfaces, pfSense is able to enforce granular policies by segmenting trusted LANs, semi-trusted DMZs and untrusted WANS, which reflects best practices of network zoning.

In addition, the DMZ contains a terminal output of Ubuntu Server confirming it is receiving the static IP address of 192.168.20.10/24 and pfSense's DMZ Gateway is configured to default route of 192.168.20.1. indicates that either the static configuration or DHCP configuration completes within isolated DMZ.

Lastly, Ubuntu Server in DMZ has achieved multiple successful Ping results across multiple segments and Internet connectivity by successfully pinging pfSense DMZ Gateway (192.168.20.1) and Google Public DNS (8.8.8.8). This confirms outbound Internet access from DMZ. This test confirms that IP reachability is working properly, NAT is working correctly, and that Kali system has successfully pinged a Windows host on the LAN.

## pfSense Interface Assignments

```
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: 10.0.3.16/24
LAN (lan)  -> em1 -> v4: 192.168.10.1/24
OPT1 (opt1) -> em2 -> v4: 192.168.20.1/24

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
8) Shell

Enter an option: █
```

## Ubuntu DMZ IP (ip a)

Showing: 192.168.20.10/24

```
ubuntu@Dheeraj:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e5:3e:ad brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.10/24 brd 192.168.20.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe5:3ead/64 scope link
        valid_lft forever preferred_lft forever
ubuntu@Dheeraj:~$
```

Kali LAN

```
(kali㉿Raj)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:bf:23 brd ff:ff:ff:ff:ff:ff
        inet 192.168.10.100/24 brd 192.168.10.255 scope global dynamic noprefixroute eth0
            valid_lft 6715sec preferred_lft 6715sec
        inet6 fe80::dc08:831:24f:12cc/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

## Windows LAN IP

## Ping tests between segments:

Ubuntu → pfSense DMZ (192.168.20.1)

```
ubuntu@Dheeraj:~$ ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=64 time=71.5 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=64 time=35.3 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=64 time=99.6 ms
64 bytes from 192.168.20.1: icmp_seq=4 ttl=64 time=64.1 ms
64 bytes from 192.168.20.1: icmp_seq=5 ttl=64 time=22.3 ms
^C
--- 192.168.20.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 22.331/58.551/99.595/27.333 ms
ubuntu@Dheeraj:~$
```

Ubuntu → Internet (8.8.8.8)

```
ubuntu@Dheeraj:~$ 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=113 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=78.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=130 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=109 time=85.2 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=109 time=44.2 ms

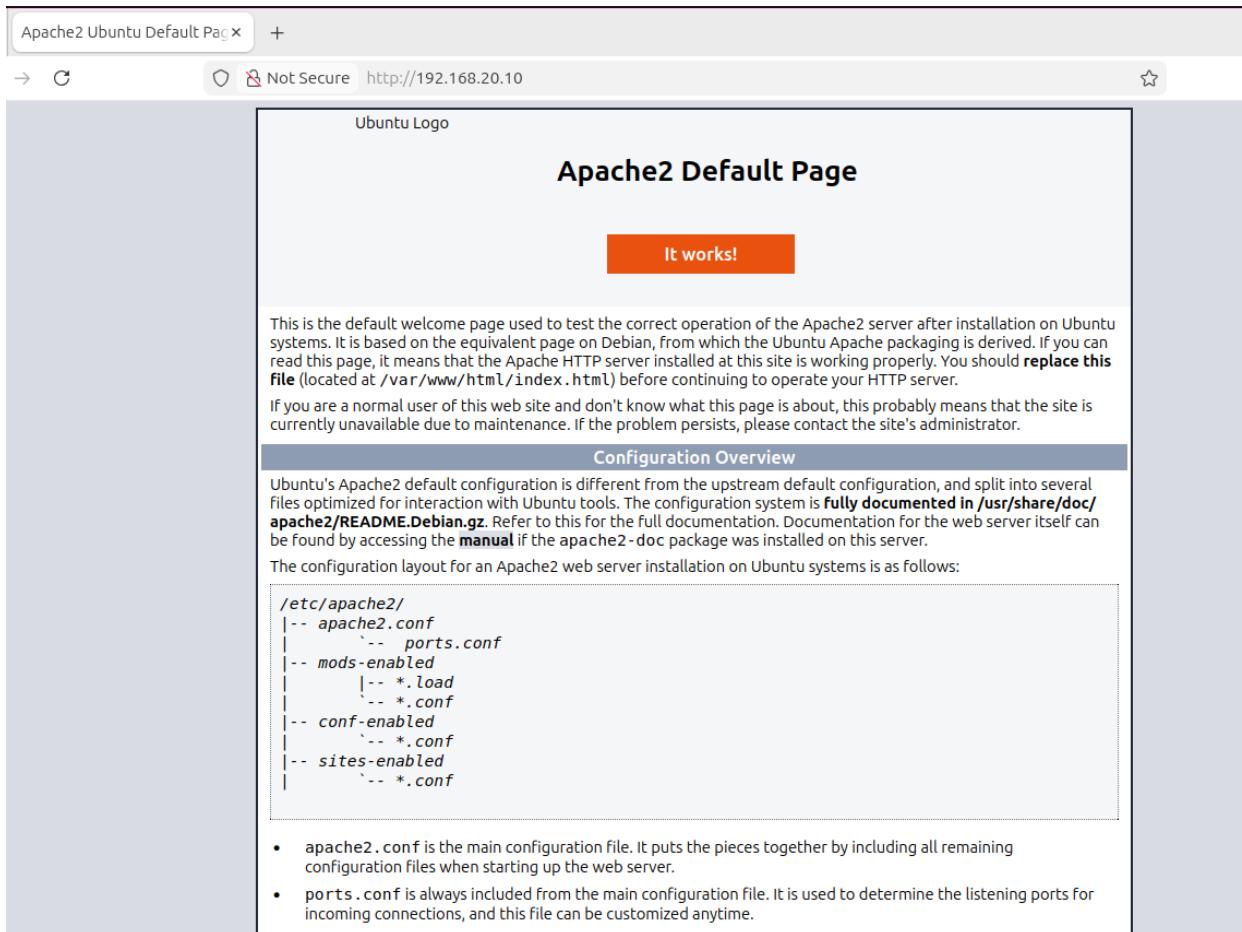
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 44.203/90.202/130.321/29.677 ms
^Cubuntu@Dheeraj:~$
```

Kali (LAN) → Windows (LAN)

```
(kali㉿Raj)-[~]
$ ping 192.168.10.101
PING 192.168.10.101 (192.168.10.101) 56(84) bytes of data.
64 bytes from 192.168.10.101: icmp_seq=1 ttl=128 time=4.56 ms
64 bytes from 192.168.10.101: icmp_seq=2 ttl=128 time=4.01 ms
64 bytes from 192.168.10.101: icmp_seq=3 ttl=128 time=6.36 ms
64 bytes from 192.168.10.101: icmp_seq=4 ttl=128 time=4.27 ms
64 bytes from 192.168.10.101: icmp_seq=5 ttl=128 time=5.41 ms
64 bytes from 192.168.10.101: icmp_seq=6 ttl=128 time=7.04 ms
64 bytes from 192.168.10.101: icmp_seq=7 ttl=128 time=8.00 ms
^C
--- 192.168.10.101 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 7332ms
rtt min/avg/max/mdev = 4.012/5.664/8.002/1.404 ms
```

```
(kali㉿Raj)-[~]
$
```

## Test HTTP



Apache2 Ubuntu Default Page

Not Secure http://192.168.20.10

Ubuntu Logo

**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 configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

## SECTION 3 — Firewall Configuration

### WAN Rule

The rules configured under Firewall>Rules>WAN interface for receiving a connection over the Internet has the two top rules clearly set to block all RFC1918 private as well as bogon networks (prevents spoofing). this has to be done prevent anyone a being able the spoof connections of WAN side are internal network. After a those two top rules, the next rule permits OpenVPN traffic coming in on UDP1194 from any source (the OpenVPN protocol by default uses UDP port 1194). The last rule is the anti-lockout rule, preventing administrative access being cut off. This is an example of secure configuration for hardening the WAN side.

### LAN Rule

The rules configured on the LAN interface are examples of a default permissive policy. After the anti-lockout rule there are two very broad rules that permit "any to any" traffic (IPv4 and IPv6) to flow from the general public Internet to the trusted LAN segment and are considered safe for internal users.

### OPT (DMZ) Rule

On the OPT1/DMZ interface the outbound rules are very strict and follow a least-privilege policy. The DMZ hosts are permitted to resolve DNS to the firewall; health-monitoring traffic can be sent to the firewall, outbound access, limited HTTP/HTTPS access (to the general Internet). But unless absolutely necessary, they are explicitly prohibited from making any connections back to the LAN. There is also a rule in place to restrict DMZ-originating connections to any of the internal LAN subnets.

### NAT Configuration

The Firewall > NAT > Outbound Page has NAT set to Hybrid mode with 2 manual outbound rules. One rule translates the LAN address (192.168.10.0/24) and second rule translates DMZ address (192.168.20.0/24) so both will have the WAN\_iface.Ip Address (public IP).

### Blocked packets from DMZ to LAN

In the Status > System Logs > Firewall logs, number of attempts the DMZ host (192.168.20.10) access resources the LAN (default gateways vs 192.168.10.x) is recorded. firewall are working for prevent the DMZ hosts are accessing trusted LAN through actively enforcing DMZ to LAN block rule; thus, it confirms successful prevention of lateral moves from less-trusted DMZ to more trusted LAN.

## WAN Rules

Firewall / Rules / WAN

Floating WAN LAN OPT1 OpenVPN

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	1194 (OpenVPN)	*	none		Allow OpenVPN	
0/0 B	IPv4 UDP	*	*	WAN address	1194 (OpenVPN)	*	none		OpenVPN OpenVPN-Server-Cert wizard	

Add Add Delete Toggle Copy Save Separator

## LAN Rule

Firewall / Rules / LAN

Floating WAN LAN OPT1 OpenVPN

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
3/1.48 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
37/5.55 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	

Add Add Delete Toggle Copy Save Separator

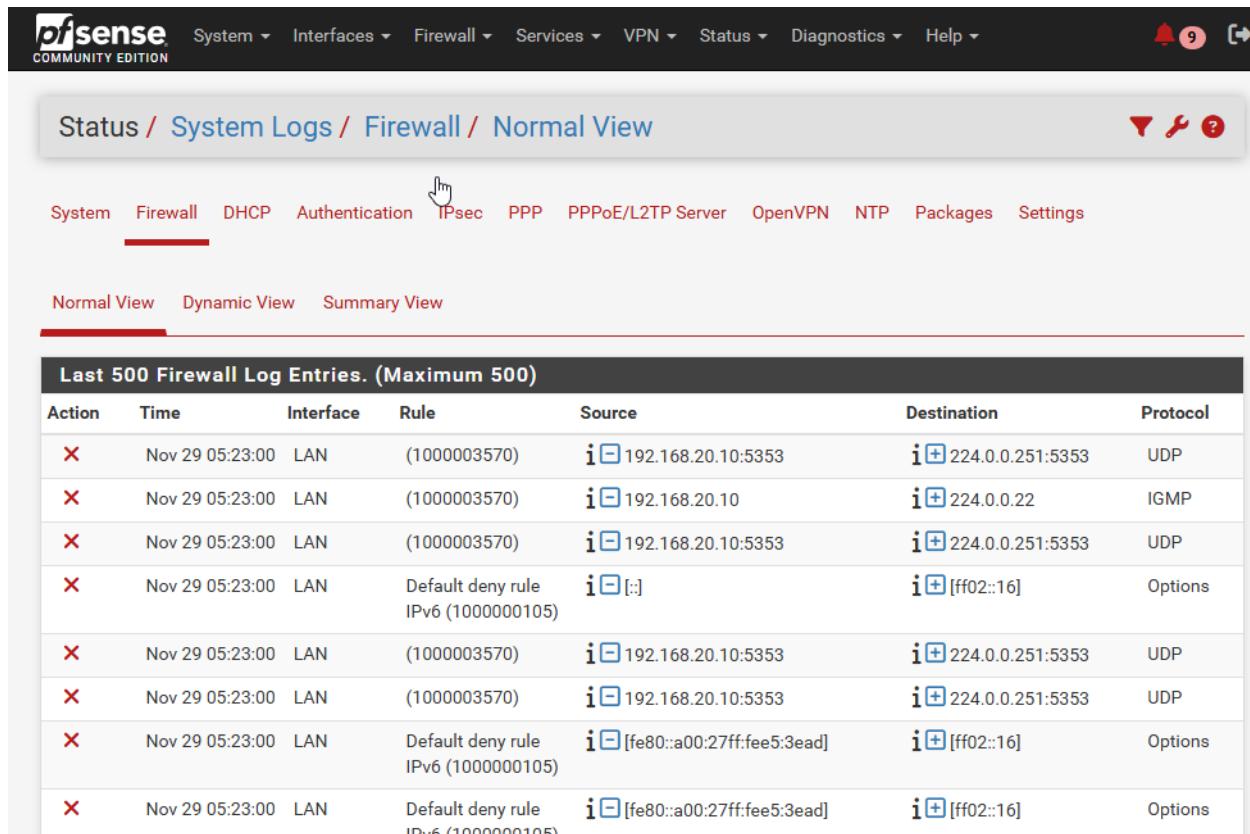
## OPT(DMZ) Rule:

Rules (Drag to Change Order)											Actions
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 TCP	OPT1 subnets	*	This Firewall (self)	53 (DNS)	*	none		Permit DNS Queries from DMZ hosts to firewall resolver	
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/1 KIB	IPv4 ICMP <u>echoreq</u>	OPT1 subnets	*	This Firewall (self)	*	*	none		Permit ICMP echo requests from DMZ for health Monitoring	
<input type="checkbox"/>	<input checked="" type="checkbox"/> 79/1.43 MIB	IPv4 *	OPT1 subnets	*	*	*	*	none		Authorize general outbound traffic from DMZ to external networks	
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 *	OPT1 subnets	*	LAN subnets	*	*	none		Restrict all DMZ-origin traffic from reached LAN segment	
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0/0 B	IPv4 TCP	LAN subnets	*	192.168.10.10	80 (HTTP)	*	none		Allow LAN client to access DMZ web server over HTTP	

## NAT Configuration

Outbound NAT Mode																		
Mode	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>														
Automatic outbound NAT rule generation. (IPsec passthrough included)			Hybrid Outbound NAT rule generation. (Automatic Outbound NAT + rules below)			Manual Outbound NAT rule generation. (AON - Advanced Outbound NAT)			Disable Outbound NAT rule generation. (No Outbound NAT rules)									
Mappings																		
	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description	Actions								
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	192.168.10.0/24	*	*	*	WAN address	*		LAN to outbound NAT rule									
<input type="checkbox"/>	<input checked="" type="checkbox"/> WAN	192.168.20.0/24	*	*	*	WAN address	*		DMZ to outbound NAT Rule									

Showing blocked DMZ→LAN packets



The screenshot shows the pfSense Firewall Log Entries page. The top navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. A notification bell icon shows 9 pending notifications. The main title is "Status / System Logs / Firewall / Normal View". Below the title, a sub-navigation bar has "Firewall" selected. The page displays a table of "Last 500 Firewall Log Entries. (Maximum 500)". The table columns are: Action, Time, Interface, Rule, Source, Destination, and Protocol. The data shows several entries with an "X" in the Action column, indicating they were blocked. The "Rule" column shows various rules, including "Default deny rule IPv6 (1000000105)". The "Source" and "Destination" columns show IP addresses, and the "Protocol" column shows UDP, IGMP, and Options.

Last 500 Firewall Log Entries. (Maximum 500)						
Action	Time	Interface	Rule	Source	Destination	Protocol
✗	Nov 29 05:23:00	LAN	(1000003570)	i [192.168.20.10:5353]	i [224.0.0.251:5353]	UDP
✗	Nov 29 05:23:00	LAN	(1000003570)	i [192.168.20.10]	i [224.0.0.22]	IGMP
✗	Nov 29 05:23:00	LAN	(1000003570)	i [192.168.20.10:5353]	i [224.0.0.251:5353]	UDP
✗	Nov 29 05:23:00	LAN	Default deny rule IPv6 (1000000105)	i [:]	i [ff02::16]	Options
✗	Nov 29 05:23:00	LAN	(1000003570)	i [192.168.20.10:5353]	i [224.0.0.251:5353]	UDP
✗	Nov 29 05:23:00	LAN	(1000003570)	i [192.168.20.10:5353]	i [224.0.0.251:5353]	UDP
✗	Nov 29 05:23:00	LAN	Default deny rule IPv6 (1000000105)	i [fe80::a00:27ff:fee5:3ead]	i [ff02::16]	Options
✗	Nov 29 05:23:00	LAN	Default deny rule IPv6 (1000000105)	i [fe80::a00:27ff:fee5:3ead]	i [ff02::16]	Options

## SECTION 4 — VPN

OpenVPN Server Settings Page

Configuration OpenVPN Server

Under VPN > OpenVPN > Server tab, an Open VPN Server to set up and to accept Incoming connections via UDP port 1194 through WAN interface. VPN Server is assigned for IP Address 10.8.0.0 it has been used for Tunnel Network communications, which is encrypted using AES 256 GCM, and Authentication via TLS. In addition of above, the PFS (Perfect Forward Secrecy) will also guarantee the Impossible Decrypt Sessions saved in the Past, thus giving Remote Users the Most Secure Method of Accessing the Network.

Firewall Configuration for OpenVPN

The Open VPN Rule page has been configured with a Custom Rule granting Authenticated VPN End Users Unrestricted Access to the Entire Internal LAN, (i.e., 192.168.10.0/24). This Custom Rule has been designed to Function in Conjunction with the Any-to-Any Rule that was created during the wizard Setup so as to give Remote Users the Same level of

Resource Access as if they were Physically on the Same Local Area Network (LAN) as Local Clients.

VPN / OpenVPN / Servers

Servers Clients Client Specific Overrides Wizards Client Export

**OpenVPN Servers**

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

**Add**

## OpenVPN Client Export Page

OpenVPN / Client Export Utility

Server Client Client Specific Overrides Wizards Client Export

**OpenVPN Server**

Remote Access Server	OpenVPN-Server-Cert UDP4:1194
----------------------	-------------------------------

**Client Connection Behavior**

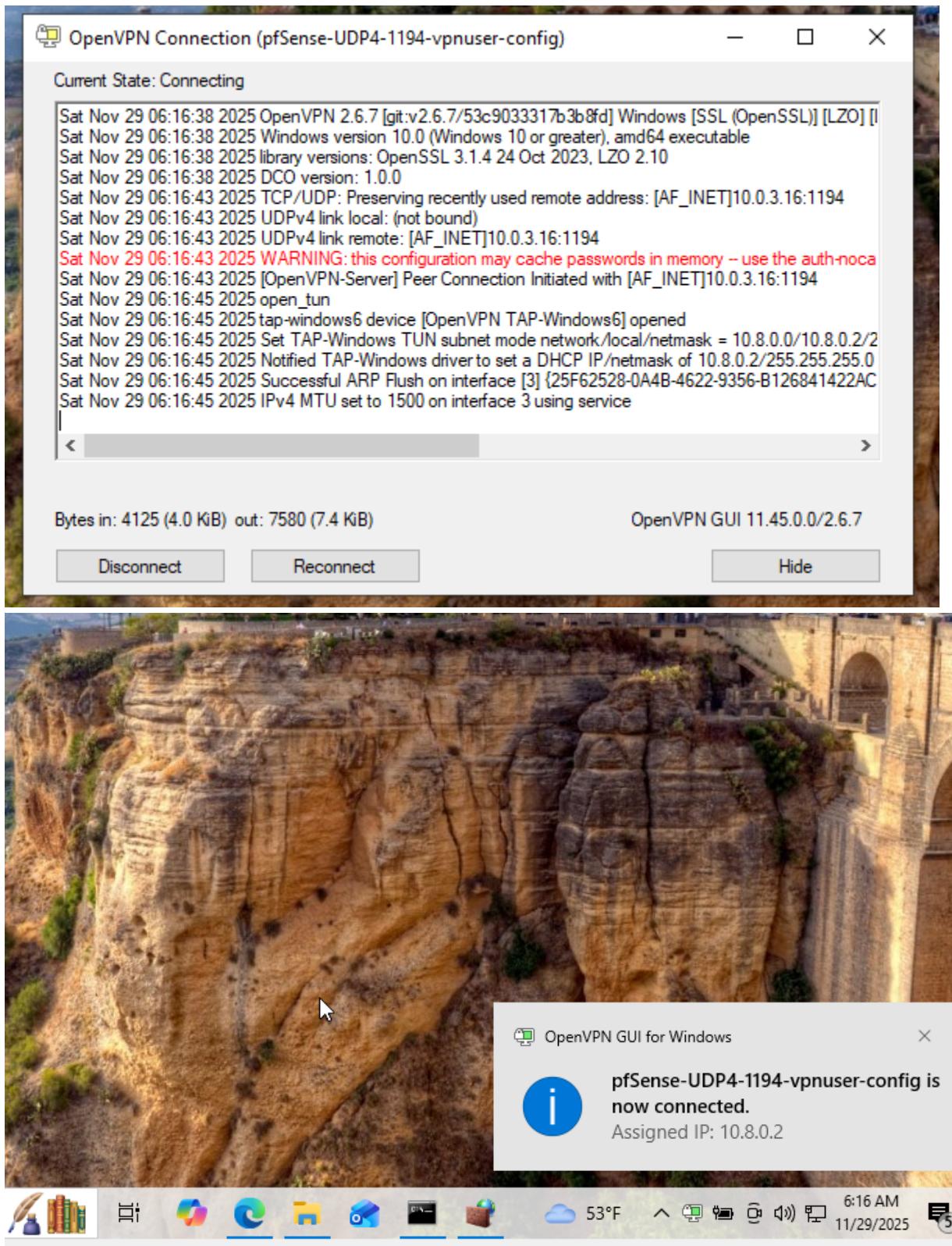
Host Name Resolution	Interface IP Address
Verify Server CN	Automatic - Use verify-x509-name where possible
Optionally verify the server certificate Common Name (CN) when the client connects.	
Block Outside DNS	<input type="checkbox"/> Block access to DNS servers except across OpenVPN while connected, forcing clients to use only VPN DNS servers. Requires Windows 10 and OpenVPN 2.3.9 or later. Only Windows 10 is prone to DNS leakage in this way, other clients will ignore the option as they are not affected.
Legacy Client	<input type="checkbox"/> Do not include OpenVPN 2.5 and later settings in the client configuration. When using an older client (OpenVPN 2.4.x), check this option to prevent the exporter from placing known-incompatible settings into the client configuration.

**OpenVPN Clients**

User	Certificate Name	Export
openuser	VPNUSERs	<ul style="list-style-type: none"> <li>- Inline Configurations:           <ul style="list-style-type: none"> <li> Most Clients</li> <li> Android</li> <li> OpenVPN Connect (iOS/Android)</li> </ul> </li> <li>- Bundled Configurations:           <ul style="list-style-type: none"> <li> Archive</li> <li> Config File Only</li> </ul> </li> <li>- Current Windows Installers (2.6.7-1x001):           <ul style="list-style-type: none"> <li> 64-bit</li> <li> 32-bit</li> </ul> </li> <li>- Previous Windows Installers (2.5.9-1x601):           <ul style="list-style-type: none"> <li> 64-bit</li> <li> 32-bit</li> </ul> </li> <li>- Legacy Windows Installers (2.4.12-1x601):           <ul style="list-style-type: none"> <li> 10/2016/2019</li> <li> 7/8/8/1/2012/2</li> </ul> </li> <li>- Viscosity (Mac OS X and Windows):           <ul style="list-style-type: none"> <li> Viscosity Bundle</li> <li> Viscosity Inline Config</li> </ul> </li> </ul>

Only OpenVPN-compatible user certificates are shown

## Client Connected Screenshot



## Windows VPN assigned IP

```
Windows IP Configuration

Unknown adapter OpenVPN Wintun:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . .

Ethernet adapter Ethernet:
  Connection-specific DNS Suffix . . . : lab.local
  Link-local IPv6 Address . . . . . : fe80::fd56:f43b:ec58:4df5%6
  IPv4 Address. . . . . : 192.168.10.101
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.10.1

Unknown adapter OpenVPN TAP-Windows6:
  Connection-specific DNS Suffix . . .
  Link-local IPv6 Address . . . . . : fe80::23d2:64df:a96a:685c%3
  IPv4 Address. . . . . : 10.8.0.2
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . :

Unknown adapter OpenVPN Data Channel Offload:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . .

C:\Windows\system32>                               Valid From: Tue 18 Nov 2025 19:00:52 -0600
```

Assigned IP: 10.8.0.2

Windows can ping pfSense via VPN IP

```
C:\Windows\system32>ping 10.8.0.1

Pinging 10.8.0.1 with 32 bytes of data:
Reply from 10.8.0.1: bytes=32 time=90ms TTL=64
Reply from 10.8.0.1: bytes=32 time=14ms TTL=64
Reply from 10.8.0.1: bytes=32 time=55ms TTL=64
Reply from 10.8.0.1: bytes=32 time=80ms TTL=64

Ping statistics for 10.8.0.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 90ms, Average = 59ms

C:\Windows\system32>
```

## Windows can ping LAN network over VPN

```
C:\Windows\system32>ping 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:
Reply from 192.168.10.1: bytes=32 time=47ms TTL=64
Reply from 192.168.10.1: bytes=32 time=23ms TTL=64
Reply from 192.168.10.1: bytes=32 time=63ms TTL=64
Reply from 192.168.10.1: bytes=32 time=168ms TTL=64

Ping statistics for 192.168.10.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 23ms, Maximum = 168ms, Average = 75ms

C:\Windows\system32>ping 192.168.10.100

Pinging 192.168.10.100 with 32 bytes of data:
Reply from 192.168.10.100: bytes=32 time=4ms TTL=64
Reply from 192.168.10.100: bytes=32 time=5ms TTL=64
Reply from 192.168.10.100: bytes=32 time=4ms TTL=64
Reply from 192.168.10.100: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.10.100:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 4ms, Maximum = 5ms, Average = 4ms

C:\Windows\system32>ping 192.168.10.101

Pinging 192.168.10.101 with 32 bytes of data:
Reply from 192.168.10.101: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.10.101:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Windows\system32>
```

Windows can ping DMZ network over VPN

```
C:\Windows\system32>ping 192.168.20.10

Pinging 192.168.20.10 with 32 bytes of data:
Reply from 192.168.20.10: bytes=32 time=191ms TTL=63
Reply from 192.168.20.10: bytes=32 time=116ms TTL=63
Reply from 192.168.20.10: bytes=32 time=143ms TTL=63
Reply from 192.168.20.10: bytes=32 time=86ms TTL=63

Ping statistics for 192.168.20.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 86ms, Maximum = 191ms, Average = 134ms

C:\Windows\system32>
```

OpenVPN status on pfSense

ovpn1: OpenVPN-Server-Cert UDP4:1194 / Client Connections: 1							
Common Name	Real Address	Virtual Address	Last Change	Bytes Sent	Bytes Received	Cipher	Actions
admin	192.168.10.101:49753	10.8.0.2	2025-11-29 06:16:45	6 KiB	10 KiB	AES-256-GCM	 
admin							  

OpenVPN firewall rule

Firewall / Rules / OpenVPN											
Rules (Drag to Change Order)											
□	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	LAN subnets	*	*	none		Allows VPN to LAN Network	  
<input type="checkbox"/>	✓ 0/480 B	IPv4 *	*	*	*	*	*	none		OpenVPN-Server-Cert wizard	  

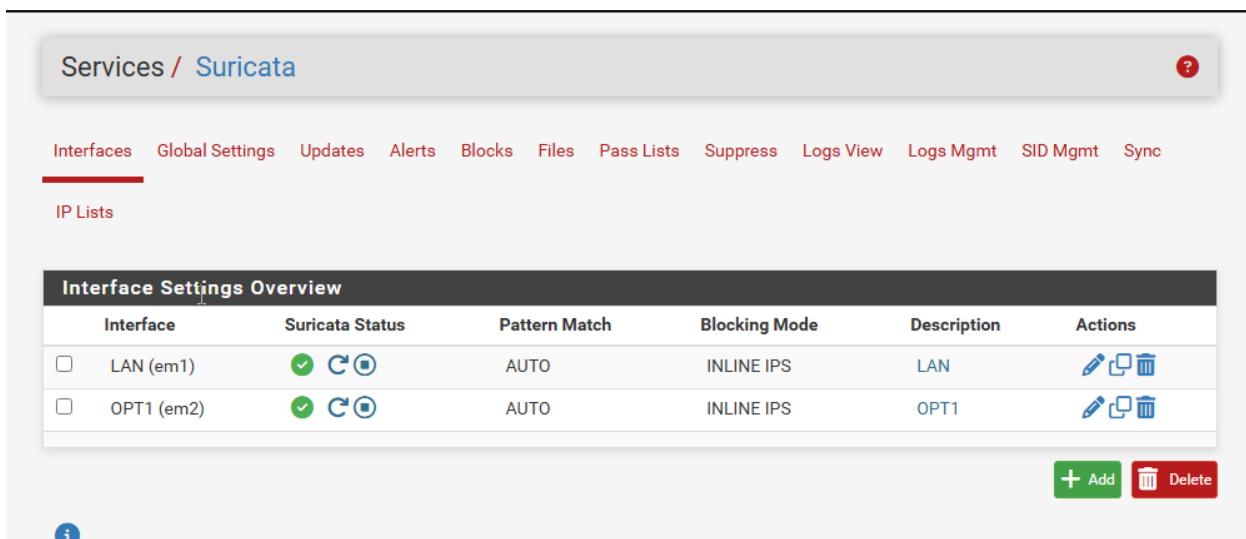
## Section 5: Configure IDS/IPS

### Suricata Interface Installation

Services → Suricata → Interfaces provides proof for successful Suricata installations and indicates active running inline IPS functionality on the following interfaces: Local Area Network (LAN) or via Ethernet port em1, and the Optional Interface 1 (OPT1) or via Ethernet port em2, which is designated as the De-Militarized Zone (DMZ). Green status lights indicate that both LAN and DMZ segments have complete operational capabilities to detect and prevent intrusions by utilizing Emerging Threats Open rulesets. The Suricata rule categories shown enabled for both LAN and DMZ interfaces are the Malware, Exploit Kit, Botnet C2 and others. This demonstrates that there is a full range of threat detection available through Suricata, with rules that have been developed to specifically protect both internal users and publicly available DMZ services from cyber threats.

I installed Suricata

Suricata Installed Interfaces Page



Interface Settings Overview					
Interface	Suricata Status	Pattern Match	Blocking Mode	Description	Actions
<input type="checkbox"/> LAN (em1)	<span>✓</span>   	AUTO	INLINE IPS	LAN	  
<input type="checkbox"/> OPT1 (em2)	<span>✓</span>   	AUTO	INLINE IPS	OPT1	  

+ Add Delete

## Suricata → LAN Rule Categories Page

State	Action	GID	SID	Proto	Source	SPort	Destination	DPort	Message
✓	⚠	1	2260000	ip	any	any	any	any	SURICATA Applayer Mismatch protocol both directions
✓	⚠	1	2260001	ip	any	any	any	any	SURICATA Applayer Wrong direction first Data
✓	⚠	1	2260002	ip	any	any	any	any	SURICATA Applayer Detect protocol only one direction
✓	⚠	1	2260003	ip	any	any	any	any	SURICATA Applayer Protocol detection skipped
✓	⚠	1	2260004	tcp	any	any	any	any	SURICATA Applayer No TLS after STARTTLS
✓	⚠	1	2260005	tcp	any	any	any	any	SURICATA Applayer Unexpected protocol

## Suricata → OPT1(DMZ) Rule Categories Page

State	Action	GID	SID	Proto	Source	SPort	Destination	DPort	Message
✓	⚠	1	2260000	ip	any	any	any	any	SURICATA Applayer Mismatch protocol both directions
✓	⚠	1	2260001	ip	any	any	any	any	SURICATA Applayer Wrong direction first Data
✓	⚠	1	2260002	ip	any	any	any	any	SURICATA Applayer Detect protocol only one direction
✓	⚠	1	2260003	ip	any	any	any	any	SURICATA Applayer Protocol detection skipped
✓	⚠	1	2260004	tcp	any	any	any	any	SURICATA Applayer No TLS after STARTTLS
✓	⚠	1	2260005	tcp	any	any	any	any	SURICATA Applayer Unexpected protocol

## Suricata LAN Alerts:

Services / Suricata / Alerts

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

Sync IP Lists

**Alert Log View Settings**

Instance to View: (LAN) LAN  
Choose which instance alerts you want to inspect.

Save or Remove Logs: [Download](#) [Clear](#)  
All alert log files for selected interface will be downloaded.

Save Settings: [Save](#)  Refresh  
Save auto-refresh and view settings Default is ON

Number of alerts to display. Default is 250

**Alert Log View Filter**

**Last 250 Alert Entries. (Most recent entries are listed first)**

Note: Alerts triggered by DROP rules that resulted in dropped (blocked) packets are shown with highlighted rows below.

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
11/29/2025 06:30:50	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:46	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:37	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:33	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum

**Last 250 Alert Entries. (Most recent entries are listed first)**

Note: Alerts triggered by DROP rules that resulted in dropped (blocked) packets are shown with highlighted rows below.

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
11/29/2025 06:30:50	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:46	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:37	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:33	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:24	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:18	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:12	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:30:06	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:29:59	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:29:52	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:29:47	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:29:39	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:29:34	⚠️	3	TCP	Generic Protocol Command Decode	23.198.7.177	443	192.168.10.101	55027	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:29:25	⚠️	3	TCP	Generic Protocol Command Decode	92.223.96.6	80	192.168.10.101	55019	1:2200074	SURICATA TCPv4 invalid checksum

## Suricata OPT1(DMZ) Alerts:

Instance to View: (OPT1) OPT1

Choose which instance alerts you want to inspect.

Save or Remove Logs: [Download](#) [Clear](#)

All alert log files for selected interface will be downloaded. Clear the currently active Alerts log file.

Save Settings: [Save](#)  Refresh

Save auto-refresh and view settings. Default is ON.

250: Number of alerts to display. Default is 250.

**Alert Log View Filter**

**Last 250 Alert Entries. (Most recent entries are listed first)**

Note: Alerts triggered by DROP rules that resulted in dropped (blocked) packets are shown with highlighted rows below.

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
11/29/2025 06:31:57	⚠️	3	TCP	Generic Protocol Command Decode	151.101.193.91	443	192.168.20.10	60254	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:55	⚠️	3	TCP	Generic Protocol Command Decode	91.189.91.82	80	192.168.20.10	33168	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:45	⚠️	3	TCP	Generic Protocol Command Decode	151.101.193.91	443	192.168.20.10	60254	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:43	⚠️	3	TCP	Generic Protocol Command Decode	91.189.91.82	80	192.168.20.10	33168	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:32	⚠️	3	TCP	Generic Protocol Command Decode	151.101.193.91	443	192.168.20.10	60254	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:30	⚠️	3	TCP	Generic Protocol Command Decode	91.189.91.82	80	192.168.20.10	33168	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:19	⚠️	3	TCP	Generic Protocol Command Decode	151.101.193.91	443	192.168.20.10	60254	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:17	⚠️	3	TCP	Generic Protocol Command Decode	91.189.91.82	80	192.168.20.10	33168	1:2200074	SURICATA TCPv4 invalid checksum
11/29/2025 06:31:07	⚠️	3	TCP	Generic Protocol Command Decode	151.101.193.91	443	192.168.20.10	60254	1:2200074	SURICATA TCPv4 invalid checksum

## pfSense Firewall Logs showing Suricata blocks

Status / System Logs / Firewall / Normal View

System Firewall DHCP Authentication IPsec PPP PPPoE/L2TP Server OpenVPN NTP Packages Settings

Normal View [Dynamic View](#) [Summary View](#)

**Last 500 Firewall Log Entries. (Maximum 500)**

Action	Time	Interface	Rule	Source	Destination	Protocol
✗	Nov 29 05:34:13	OPT1	Block IPv4 link-local (1000000101)	i [169.254.85.98:137]	i [169.254.255.255:137]	UDP
✗	Nov 29 05:34:13	OPT1	Block IPv4 link-local (1000000101)	i [169.254.85.98:53792]	i [224.0.0.252:5355]	UDP
✗	Nov 29 05:34:13	OPT1	(1000002520)	i [192.168.10.101]	i [224.0.0.22]	IGMP
✗	Nov 29 05:34:13	OPT1	(1000002520)	i [192.168.10.101]	i [224.0.0.22]	IGMP
✗	Nov 29 05:34:13	OPT1	(1000002520)	i [192.168.10.101:137]	i [192.168.10.255:137]	UDP
✗	Nov 29 05:34:13	OPT1	(1000002520)	i [192.168.10.101:137]	i [192.168.10.255:137]	UDP
✗	Nov 29 05:34:14	OPT1	(1000002520)	i [192.168.10.101]	i [224.0.0.22]	IGMP
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options
✗	Nov 29 05:34:14	OPT1	(1000002520)	i [192.168.10.101]	i [224.0.0.22]	IGMP
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options
✗	Nov 29 05:34:14	OPT1	Default deny rule IPv6 (1000000105)	i [fe80::fd56:f43b:ec58:4df5]	i [ff02::16]	Options

## SECTION 6 — Testing

The NMAP Basic Port Scan that conducted Kali Linux machine for Local Area Network (LAN) of Ubuntu machine to the DMZ proved is successful according to the screen output of the simple command nmap -F 192.168.20.10. Basic Port Scan confirmed for two open ports available Ubuntu machine: port 22 (SSH) and port 80 (HTTP). purpose is Basic Port Scan verify that no unintentional services are exposed DMZ of the LAN, and that firewall restricts intended traffic from the LAN to the Ubuntu machine.

The NMAP Aggressive OS & Service Detection Scan executed from the Kali Linux machine within the LAN towards the Ubuntu DMZ machine provided detailed service versioning and operating system fingerprinting using the command nmap -A 192.168.20.10. Service versioning showed that both versions of OpenSSH (8.9p1) and Apache (2.4.52) were available from the scan, as well as that the operating system running on the Ubuntu DMZ machine is Ubuntu Linux 22.04. The results of this aggressive scan prove that the pfSense firewall allows legitimate and thorough scanning of the DMZ machine from the internal trusted LAN, making it a very useful tool for conducting internal security assessments of the company, as any attempts to conduct a similar detailed probe from outside the company via the internet would be blocked and/or limited by the pfSense firewall.

An example of a SYN stealth scan executed from the Kali Linux machine targeting a Windows PC on the same LAN segment returns a number of open ports for this Windows PC within a very short time frame (generally, this is 135, 139, 445 and 3389 for RDP). As both Kali machine and Windows PC are found within same trusted LAN, and since both machines have no specific restrictions, SYN stealth scan completed quickly and without detection by traditional firewalls. provides an example of why it is critical for companies to implement some type host-based or intrusion detection/prevention system, as traditional firewalls offer no protection to organization.

### Examining Suricata for Detection of Scans via Firewall/IDS:

As a result of NMAP scanning, Suricata Alerts for the LAN and DMZ Interfaces exhibited numerous alerts for “SCAN Nmap – Timing Ping”; “SCAN Nmap – OS Detection Probe”; and “ET SCANT – Aggressive Scanning with full proxying”.

All indications of successful identification by Suricata are shown by the green block icon, which also shows when Suricata blocked the identified activity. Thus Suricata was able to effectively detect and prevent an intrusion.

### Denial of Service (DoS) Attack Simulation from Kali Linux to Ubuntu DMZ:

After issuing the command hping3 --flood -S --rand-source -p 80 from Kali Linux to initiate an SYN flood attack against the web server (Linux DMZ) with an IP of 192.169.20.10. Suricata triggered several alerts as a result, showing alerts for both “DoS Attempt” and “SYN Flood.” The logs from pfSense also included logs showing packets being dropped during the attack; therefore, indicating that Suricata was successfully utilizing its inline IPS to prevent the attack. The Ubuntu DMZ web server experienced no increase in CPU usage as the attack was ongoing; therefore Suricata’s real-time size detection and prevention system was clearly effective against the DoS attack.

#### SSH Brute-force Attack from Kali to Ubuntu DMZ (192.168.20.10)

perform a brute-force attack against SSH on DMZ server, an attack using Hydra or Medusa will be accomplished. Repeated alerts Suricata for “BRUTEFORCE SSH and ET SCAN Potential SSH Brute Fo0rce” will appear in the intrusion detection system (IDS) log as attacks build up their attempts. Intrusion Prevention System (IPS) of Suricata automatically block Kali Attacker device's IP address after a defined threshold has been reached ( threshold typically includes 15-20 failed login attempts within a timeframe of 60 seconds). Once this threshold has been recorded by Suricata, future attempts to connect from Kali to Port 22 via SSH will be blocked or denied by firewall. The forensic evidence of the blocked attempts from Kali can be seen in both the Suricata block logs, and the immediate receipt of an RST packet or timeout response after the third attempt.

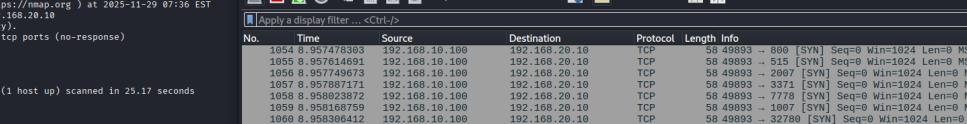
#### Final Summary of the Block Logs from the pfSense Firewall in unison with Suricata

When combined, the pfSense Firewall Block Logs and Suricata Block Logs illustrate a flow of attacks detected as they move through the security infrastructure: Initial reconnaissance alerts, Aggressive Scanning Denial of Services (DoS), Mitigation, Brute-Force Blocking. Of the many confirmed detections in the logs, the numerous Red Block Log Entries from Suricata indicate all the attacks performed against the DMZ were stopped using a combination of Stateful Firewall Rules and Suricata Inline IPS. The logs also show many Policy Descriptions from Suricata, i.e. “ET POLICY SUSPICIOUS INBOUND SSH” and “INLINE BLOCK”. Throughout the entire process all types of attacks were detected, alerted to, and stopped while at the same time all legitimate data was processed normally. Validation of the Recent Comprehensive Testing Phase confirms that the segmented network, very restrictive Firewall Policies, and properly configured Suricata IDS/IPS of pfSense can effectively prevent successful reconnaissance activities; and Deny, Block or Mitigate exploitation attacks against the enterprise and Denial of Availability Attacks

## NMAP SCAN FROM KALI → UBUNTU DMZ

## Basic port scan

```
(kali㉿kali)-[~]
$ nmap 192.168.20.10
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-29 07:36 EST
Nmap scan report for 192.168.20.10
Host is up (0.25s latency).
Nmap done: 1 IP address (1 host up) scanned in 25.17 seconds
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 25.17 seconds
(kali㉿kali)-[~]
$
```



## Aggressive OS & service detection

## NMAP SCAN FROM KALI → WINDOWS LAN PC

## Stealth scan

## Detect firewall

## DoS Attack Simulation

## From Kali → DMZ

## SSH Brute-Force

Target: **Ubuntu DMZ (192.168.20.10)**