

pfSense Virtual Firewall Lab Report

OBJECTIVE

Enhance your cybersecurity lab by placing a pfSense virtual firewall between your Kali Linux and Windows 10 VMs to control, inspect, and secure internal traffic

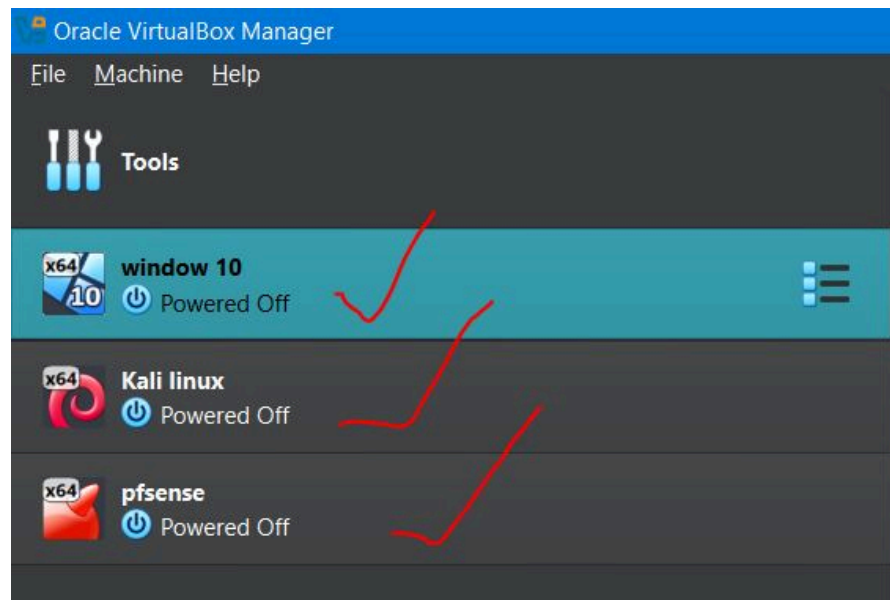
Tools Used

VirtualBox

pfSense CE ISO (v2.7.2)

Kali Linux VM

Windows 10 VM



pfSense Configuration

1. Network Interfaces

WAN (em0) – Connected to NAT (Internet access)

LAN (em1) – Connected to Internal Network
LAN IP: 192.168.1.1/24

```

pfSense [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Do you want to proceed [y/n]? y
Writing configuration...done.
One moment while the settings are reloading... done!
VirtualBox Virtual Machine - Netgate Device ID: a74de5a5bc4adc511058

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.2.15/24
                v6/DHCP6: fd17:625c:f037:2:a00:27ff:fed9:8d73/
LAN (lan)      -> em1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator 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: 
```

2. Firewall Rules

ICMP Rule (Allow Ping):

Interface: LAN

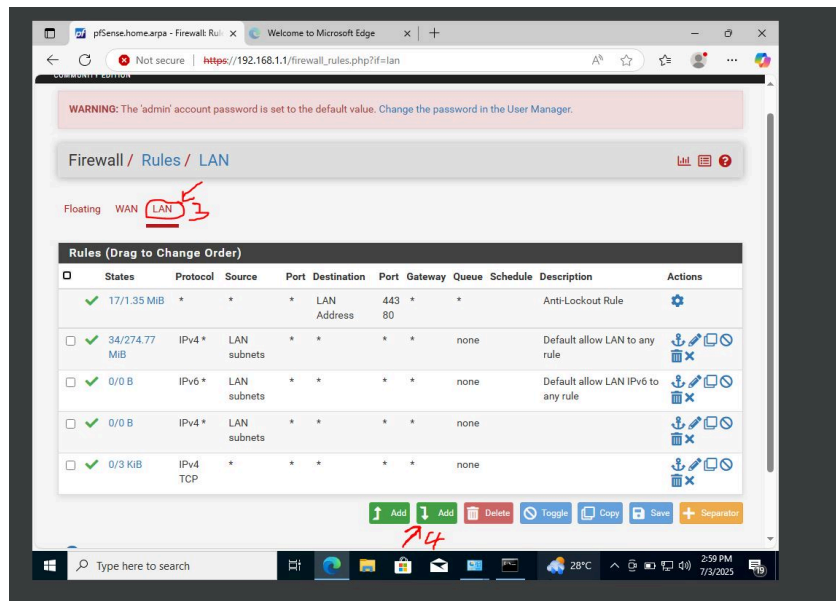
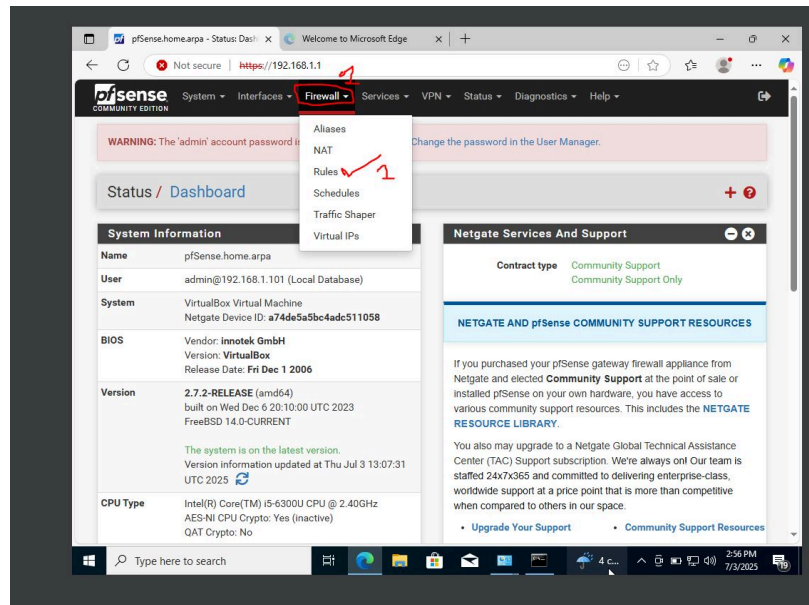
Protocol: ICMP

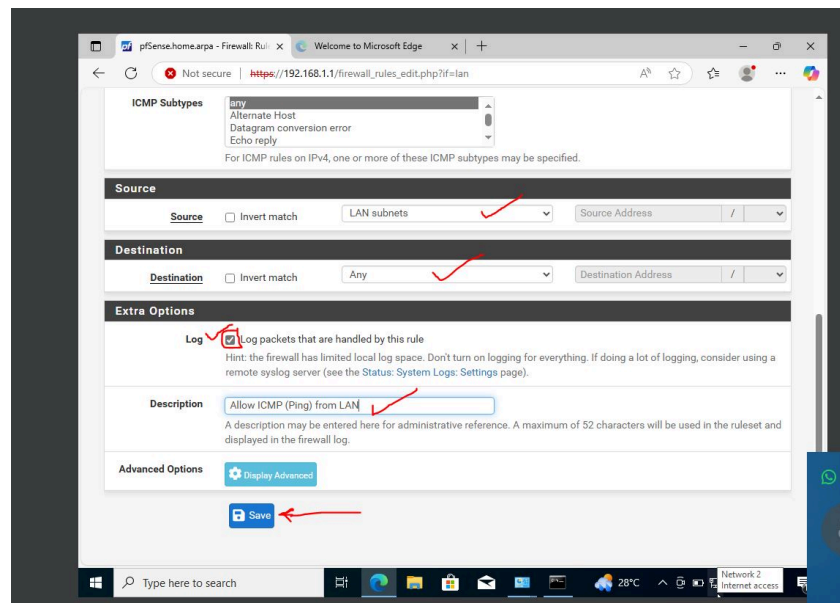
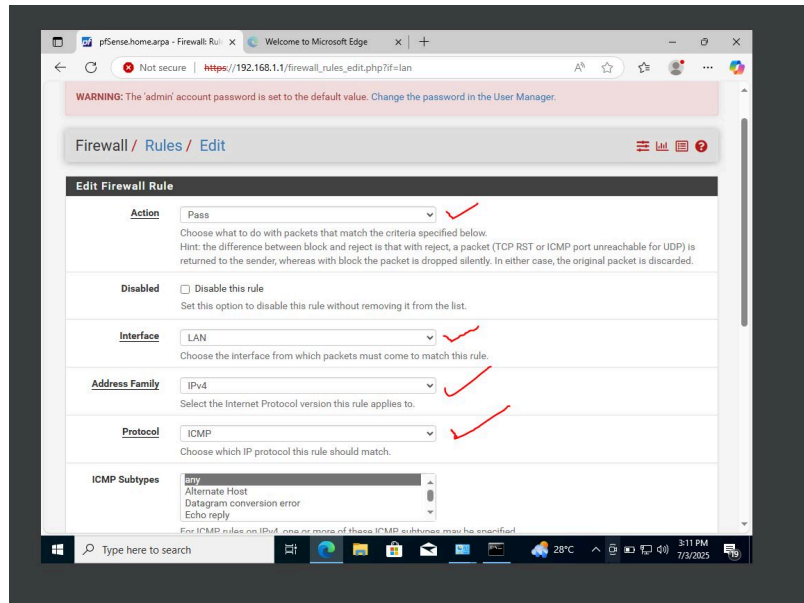
Source: LAN Subnet

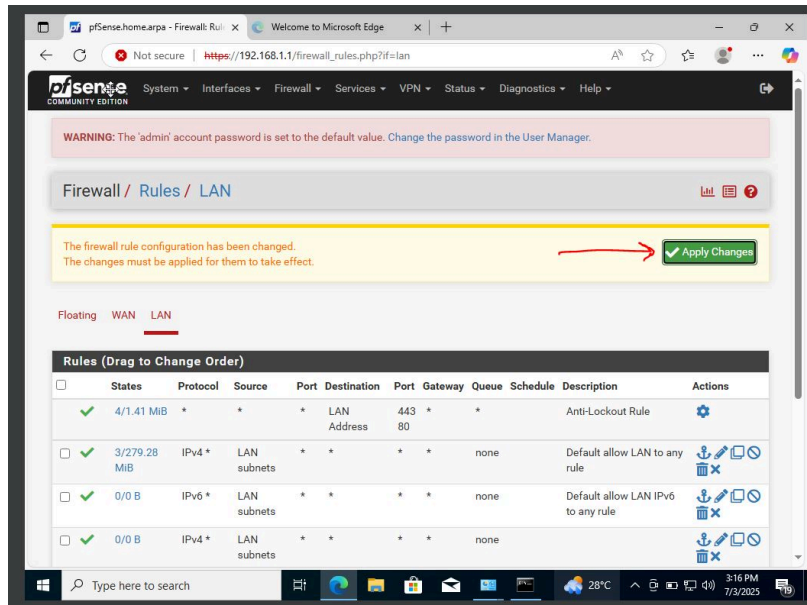
Destination: any

Description: Allow ICMP Ping

Log packets: ✓







DNS Rule (Allow DNS):

Interface: LAN

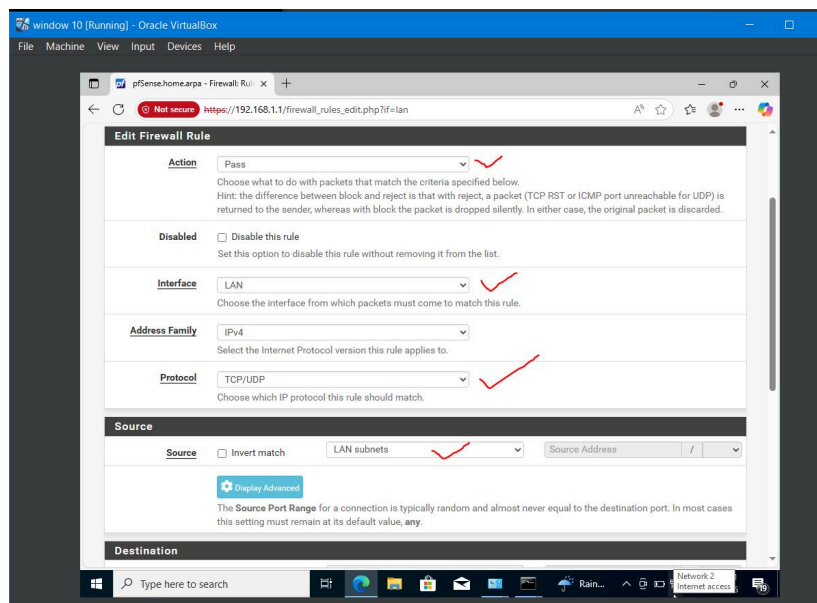
Protocol: TCP/UDP

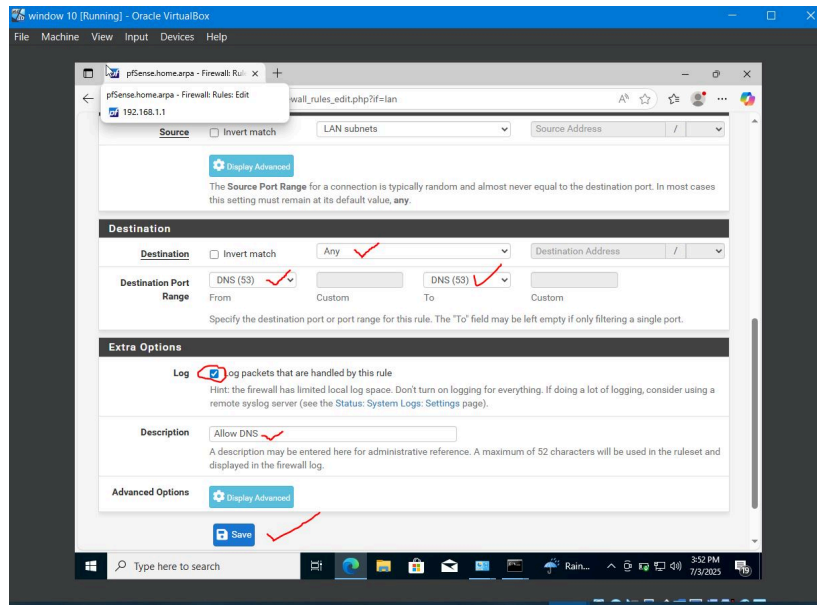
Source: LAN Subnet

Destination Port Range: DNS (53) to DNS (53)

Description: Allow DNS

Log packets: ☒

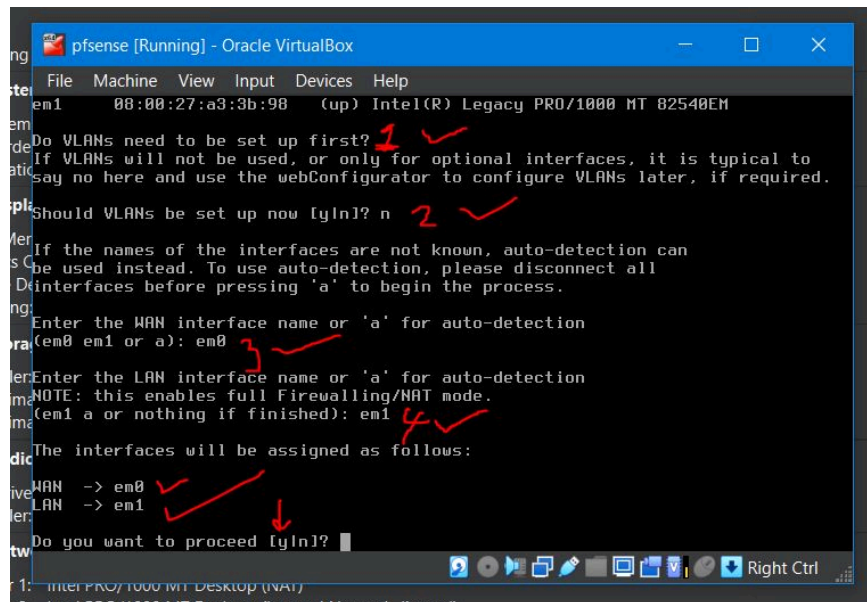




1. NAT Configuration

Mode: Automatic Outbound NAT (default)

Translates LAN traffic to WAN for internet access

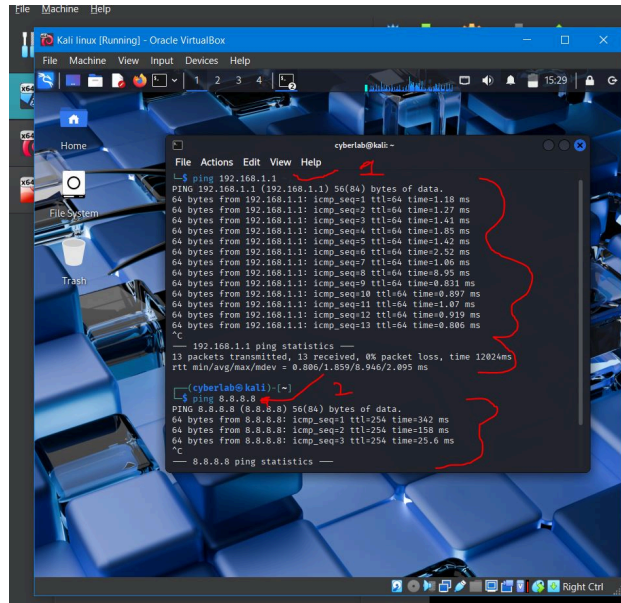


TESTING & VALIDATION

Ping Tests

From To Result

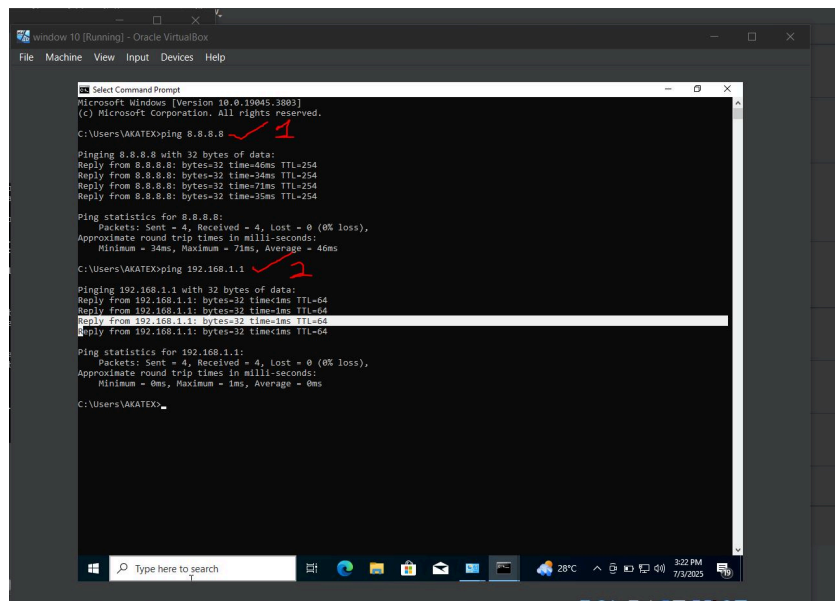
Kali	192.168.1.1	Success	Windows	192.168.1.1	Success	Kali	8.8.8.8
Success	Windows	8.8.8.8	Success				



```

cyberlab@kali:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=1.27 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=1.41 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.25 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=1.42 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=2.52 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=1.96 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=0.95 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=64 time=0.831 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=64 time=0.897 ms
64 bytes from 192.168.1.1: icmp_seq=11 ttl=64 time=1.07 ms
64 bytes from 192.168.1.1: icmp_seq=12 ttl=64 time=0.919 ms
64 bytes from 192.168.1.1: icmp_seq=13 ttl=64 time=0.886 ms
^C
192.168.1.1 ping statistics:
 13 packets transmitted, 13 received, 0% packet loss, time 12024ms
rtt min/avg/max/mdev = 0.886/1.859/8.946/2.895 ms

cyberlab@kali:~$ 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=254 time=342 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=254 time=158 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=254 time=25.6 ms
^C
8.8.8.8 ping statistics:
 3 packets transmitted, 3 received, 0% packet loss, time 200ms
rtt min/avg/max/mdev = 158.000/158.000/342.000/158.000 ms
    
```



```

C:\Users\AKATEX>ping 8.8.8.8
Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=46ms TTL=254
Reply from 8.8.8.8: bytes=32 time=34ms TTL=254
Reply from 8.8.8.8: bytes=32 time=71ms TTL=254
Reply from 8.8.8.8: bytes=32 time=35ms TTL=254

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 34ms, Maximum = 71ms, Average = 46ms

C:\Users\AKATEX>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
    
```


✓ DNS RESOLUTION

Kali:

dig google.com → Returned IP

```
cyberlab@kali:~$ dig google.com

; <div>Example Domain</div>
</div>
</div>
```

Windows:

nslookup google.com → Returned IPv6 address (success)

```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AKATEX>nslookup google.com
Server: pfSense.home.arpa
Address: 192.168.1.1

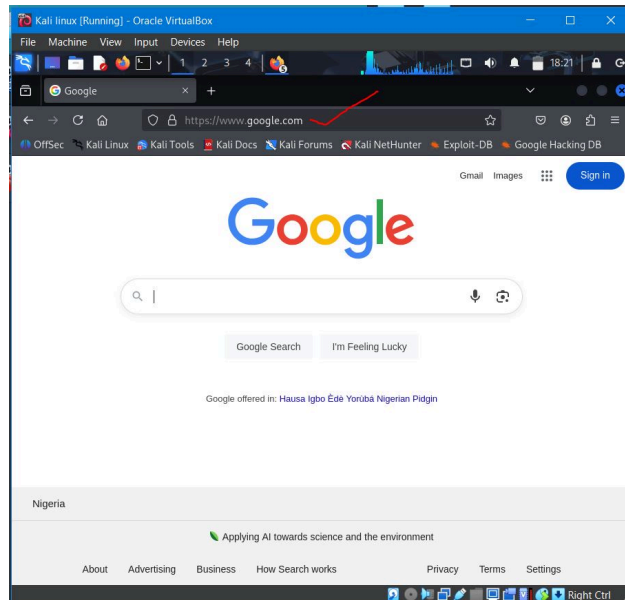
DNS request timed out.
timeout was 2 seconds.
Name: google.com
Address: 2a00:1450:4003:819::200e

C:\Users\AKATEX>
```

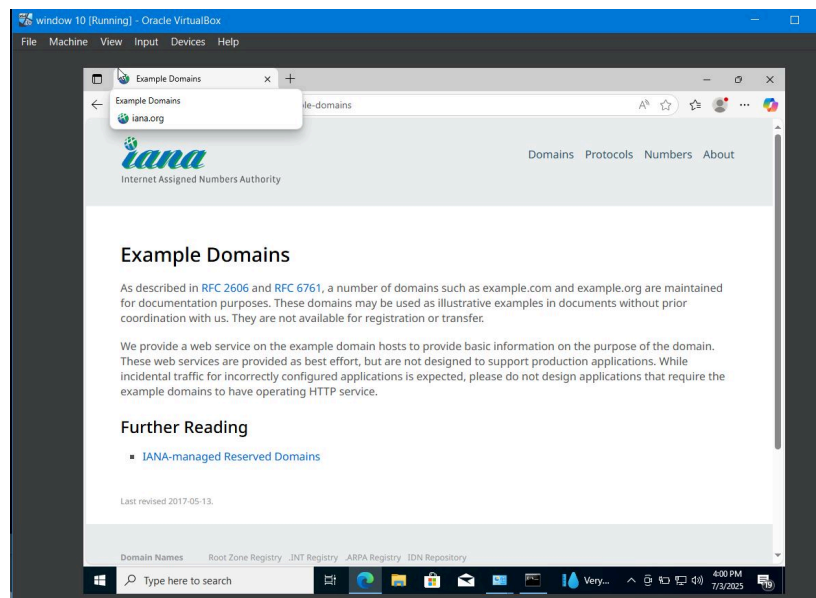

✓ Web Access

OS Website Result

Kali https://google.com Loaded



Windows https://example.com Loaded



🚩 CONCLUSION

You've successfully:

Installed and configured pfSense

Routed Kali and Windows VMs through it

Created custom firewall rules (ICMP, DNS)

Verified NAT and DNS functionality

Logged and tested all network paths

Your virtual firewall is fully operational! This setup can now be extended for advanced testing like VLAN segmentation, VPN tunneling, IDS/IPS with Suricata, or full network monitoring.