Lab 3: Perimeter Firewalling with Pfsense, and Snort IDS

Rich Macfarlane Bill Buchanan

Aim:

The aim of this lab is to introduce the PfSense stateful firewall, using the Web Application for remote administration, exploring the stateful firewalling, and Snort IDS to explore the building of a secure network infrastructure, and to start using the Kali pen testing distribution for security testing.

Time to Complete:

Approx. 2-5 hours

Activities:

- **Complete Lab 3:** Creating Secure Architectures with Pfsense and Snort IDS

Learning activities:

At the end of this lab, you should understand:

- How to remotely configure a Pfsense firewall, and set up and test the firewalling.
- How stateful firewalling operates.
- How to use security audit features available in Pfsense, such as packet capture and logging.
- How to use Snort IDS to detect various policy violations and intrusions.
- How to use network security tools to scan networks and hosts, and to craft packets for testing.

Reflective statements (end-of-exercise):

•Reflect on the differences between the Vyatta and Pfsense firewalls.

•What are some of the additional features available in Pfsense that have not been covered in this lab?

References:

- Online References throughout the lab notes
- Pfsense online documentation:
  https://doc.pfsense.org/index.php/Main_Page

Lab Overview

Our challenge is to setup **a perimeter network with an inside private subnet and a DMZ subnet**, and test from inside and outside your perimeter network. Each of you will be allocated a network and hosts to configure and get on-line (Figure 1). For this you will be allocated your own network which you can access from the vSphere infrastructure (vSOC.napier.ac.uk).

You have a pfSense firewall, a Linux host, and a Windows host to achieve your objectives.

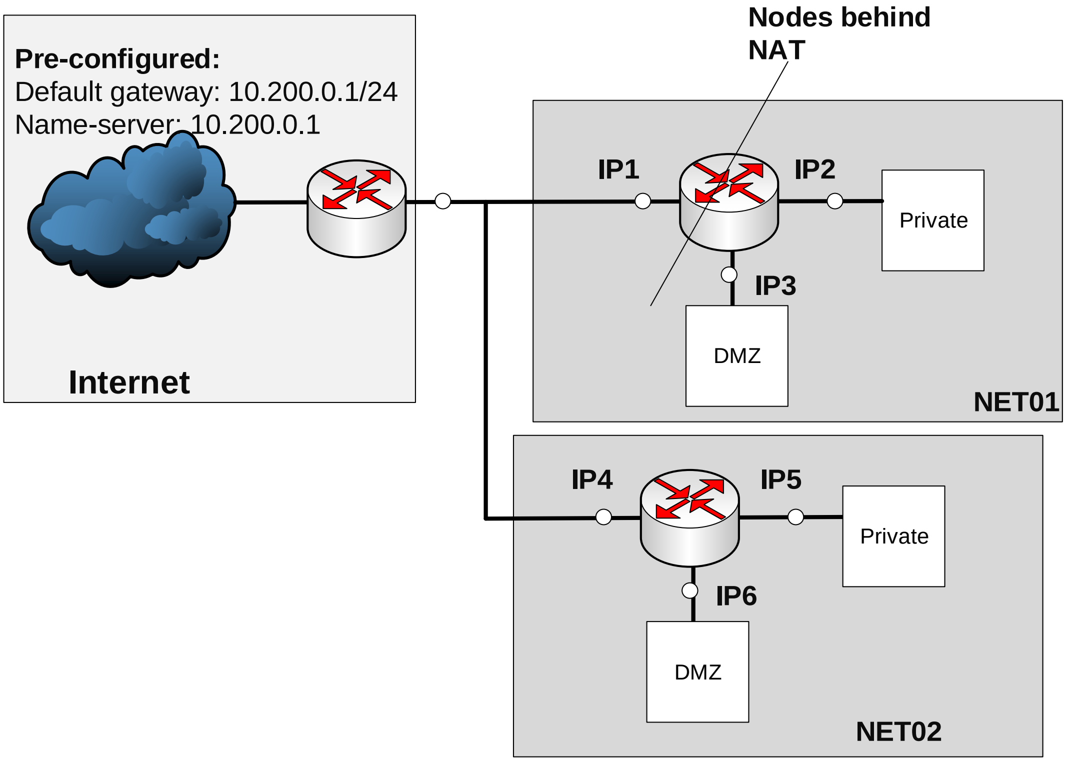


Figure 1: Lab architecture for two students networks

Setting up VMs and the Student Network

Your networks will be: 192.168.*x*.0/24 192.168.*y*.0/24 **From**
Note:

First log into vCenter (vSOC.napier.ac.uk), and then select your network infrastructure addressing from a**ddressing Allocation A**. Use this to configure the network addresses for the rest of the lab.

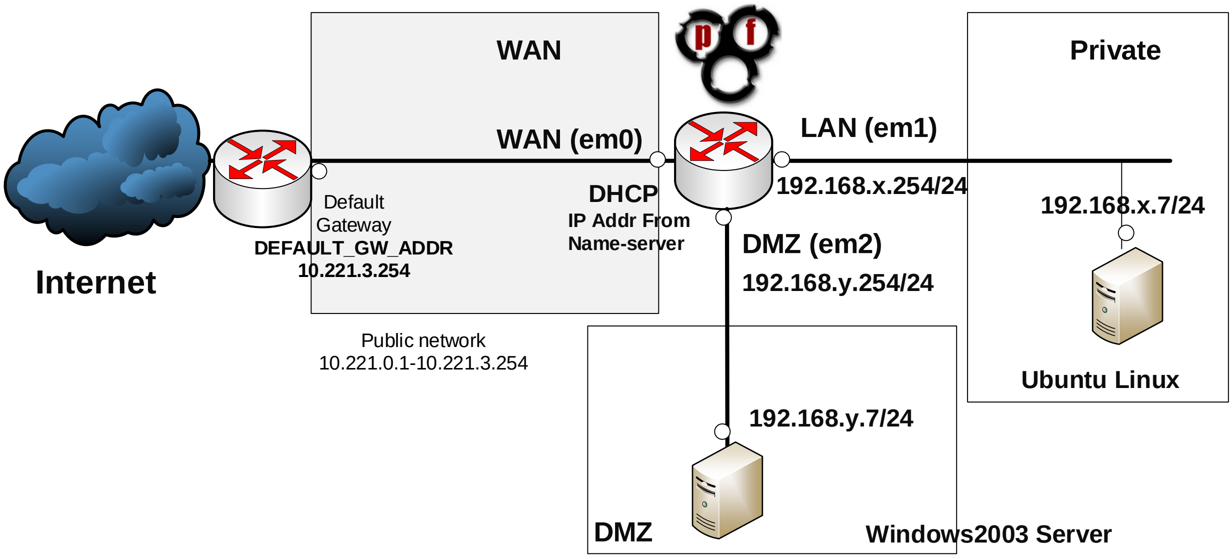


Figure 2: Individual Student Lab setup

Note: Sometimes the network interface cards are different on VM's to what you might expect, such as Eth3, Eth4 and Eth5. Firewall typically have the first network name as the Outside/Public network, the second is the Inside/Private network, and the third/fourth etc are the DMZs (typical for most firewalls)

Complete the addressing on the network diagram on the next page, filling in the boxes with your addressing - the allocated networks, subnets, and IP addresses, and use as reference, as you complete the lab.

Credentials: Users/Passwords for VMs

Windows 2003:	User: Administrator, Password: napier
Ubuntu:	User: root, password: napier123
Pfsense	User: admin, Password: pfsense
Kali	User: root Passrd:toor

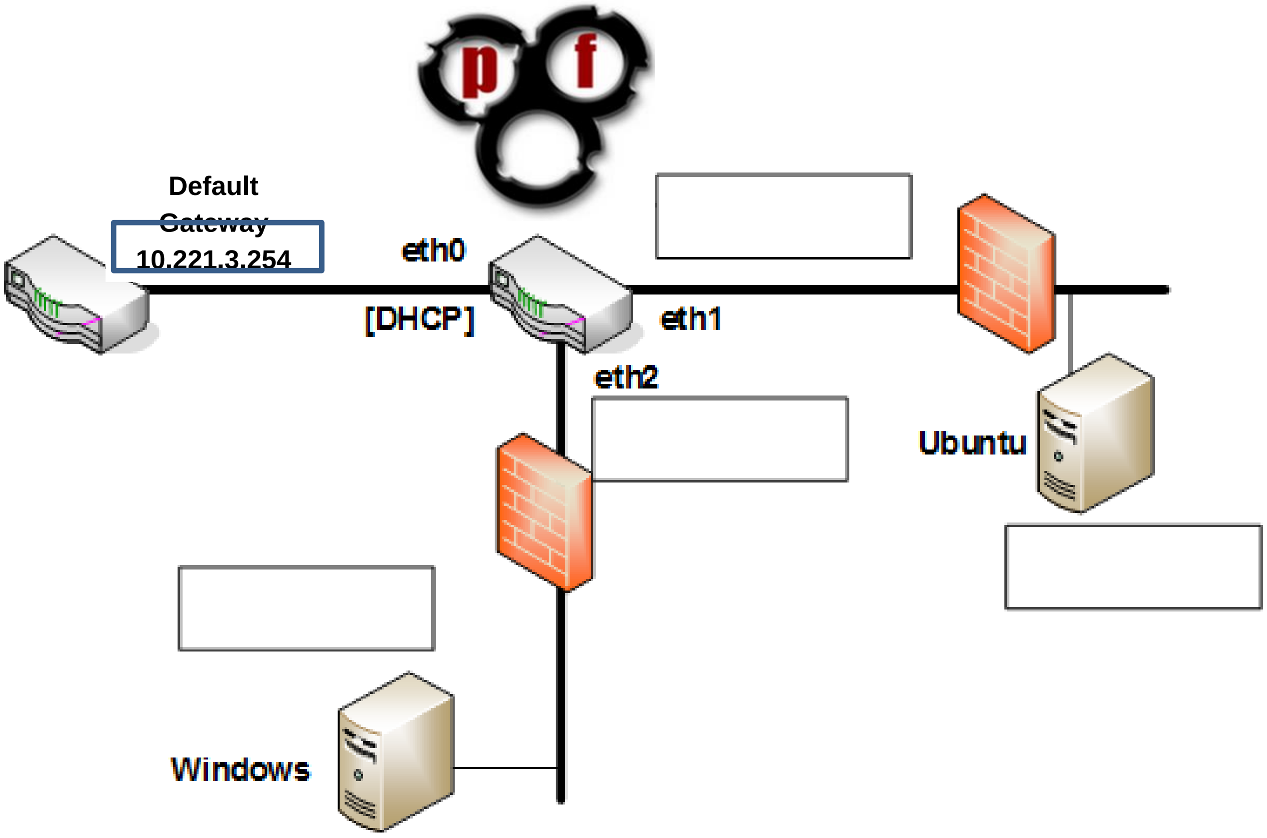
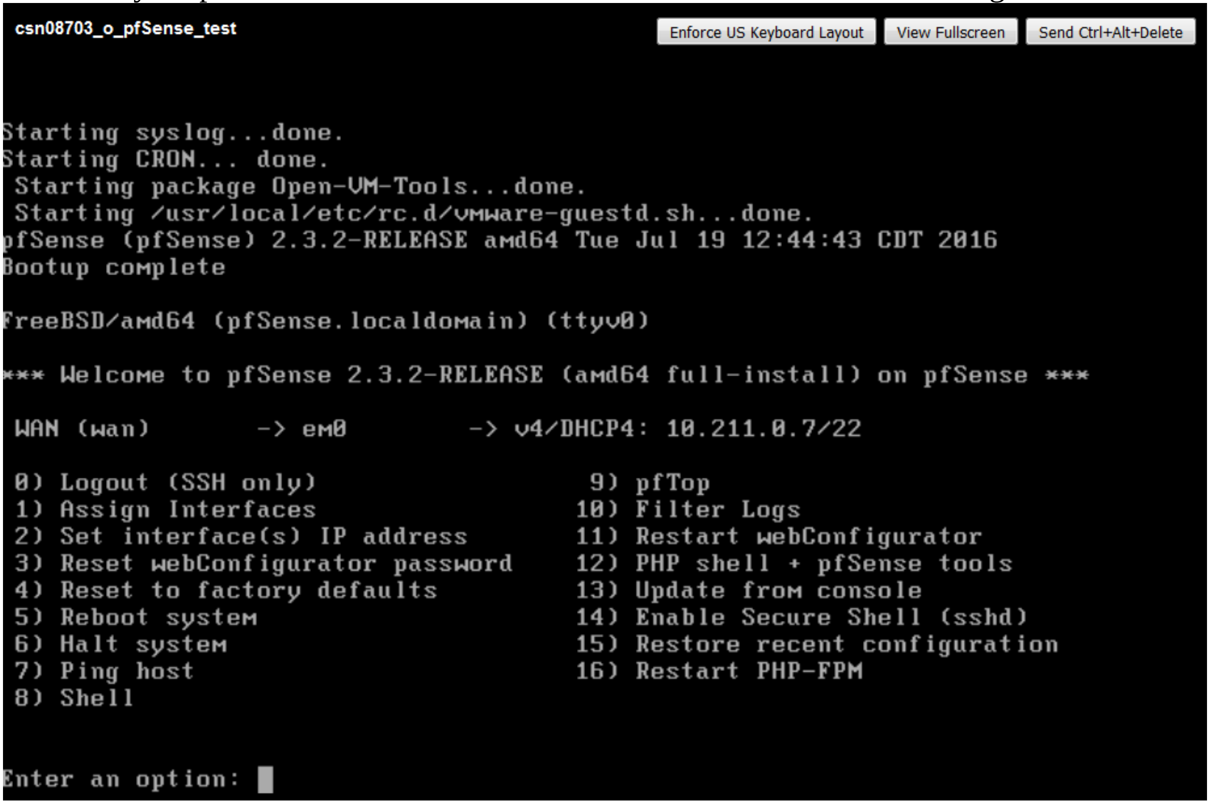


Figure 3: Each student or group’s network – complete the diagram with your own IP Addressing from table and refer back to this while doing the lab!!

Initial Configuration of the PFSense Firewall

Ref for configuring the PFSense firewall:
https://doc.pfsense.org/index.php/Main_Page

Power on your pfsense firewall. You should see the menu driven initial configuration screen:



```
Starting syslog...done.
Starting CRON... done.
 Starting package Open-VM-Tools...done.
 Starting /usr/local/etc/rc.d/vmware-guestd.sh...done.
pfSense (pfSense) 2.3.2-RELEASE amd64 Tue Jul 19 12:44:43 CDT 2016
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 10.211.0.7/22

 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: █
```

Note: at any point to get the mouse focus from the VM use CTRL+ALT)

Networking - Create Interface Names/ Config Interface IP Addressing

To create the interface names for the outside, inside and DMZ network interfaces, select **1) Assign Interfaces** (saying no to setting up VLANs just now)

Assign the firewall's first 3 network interfaces:
Public **WAN** interface to **em0**,
Private **LAN** interface to **em1**
Optional interface **OPT1** to **em2** (DMZ network).

Q. Has the WAN interface been given an IP address?
Q. What is the IP Address?

Q. How did this get an IP Address? And where from?

Test Connectivity

Q. From pfsense firewall, can you ping the WAN interface? Yes/No

Q. Can you ping the lab gateway (DEFAULT_GW_ADDR from Moodle)? Yes/No

Now we want to setup your Private Inside Network gateway.

Select **2) Set Interface IP Address** option to change the IP addresses on the interfaces.

Setup a static IP address for the em1 LAN interface to **192.168.*x*.254/24. (no ipv6, no DHCP)** Use **HTTP** and not HTTPS.

Firewall Remote Administration

PfSense firewalls, also have web-based configuration service running. Your firewall should now be displaying the URL that you can configure your firewall with from with the inside private LAN.

Q. What is the URL for connecting to the **webConfigurator** client?

Q. What is the protocol being used? Is this encrypted? Yes/No

Test Connectivity

Q. Can you ping the LAN interface? Yes/No

We will now configure the hosts and further firewall configuration can then be done via a Web browser.

Configure Host Networking

Now we will configure the hosts on the Private and DMZ LAN networks.

Setup the Linux host with an IP Address of 192.168.*x*.7/24 and a default gateway of your firewall LAN interface (192.168.*x*.254/24).

```
sudo ip addr add 192.168.x.7/24 dev ethx
sudo ip route add default via 192.168.x.254
```

If not set up already, setup the DNS name server on the Linux host by editing the /etc/resolv.config and adding a nameserver:

```
sudo nano /etc/resolv.conf
```

Add the nameserver `DEFAULT_GW_ADDR`

On the Windows server modify the static address on the interface with:

IP: 192.168.*y*.7
Subnet mask: 255.255.255.0
Gateway: 192.168.*y*.254
DNS: DEFAULT_GW_ADDR

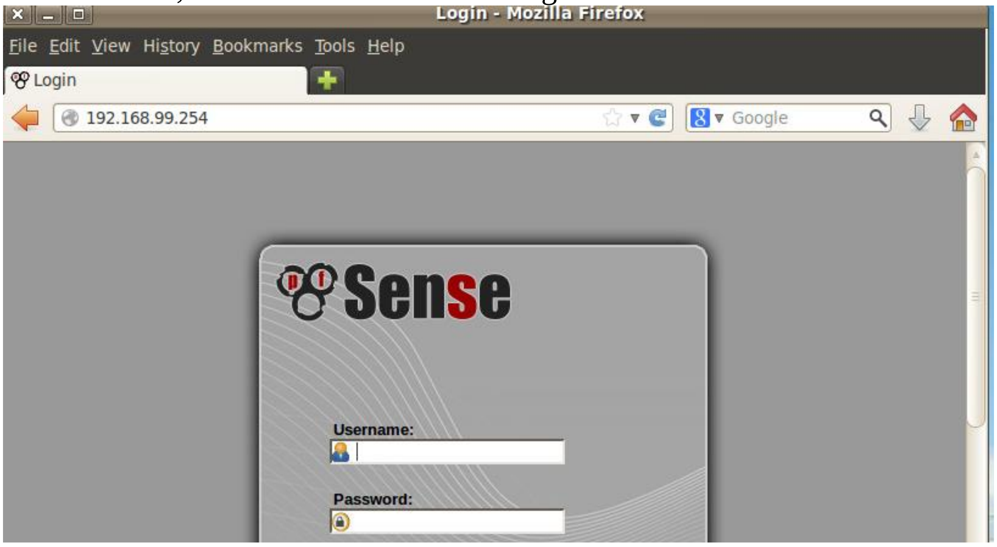
Test Connectivity

Using the **ping** command to check the local host interfaces are up and running and then connectivity from one system to another, by pinging each interface, starting with the local machine interface and working to the other hosts one hop at a time.
Note: the DMZ LAN network does not have a gateway so this will be limited to the host.

Firewall Remote Administration

Now we configure the firewall, via the webConfigurator web service running on the firewall:

From Ubuntu, connect to the firewall using a web browser:



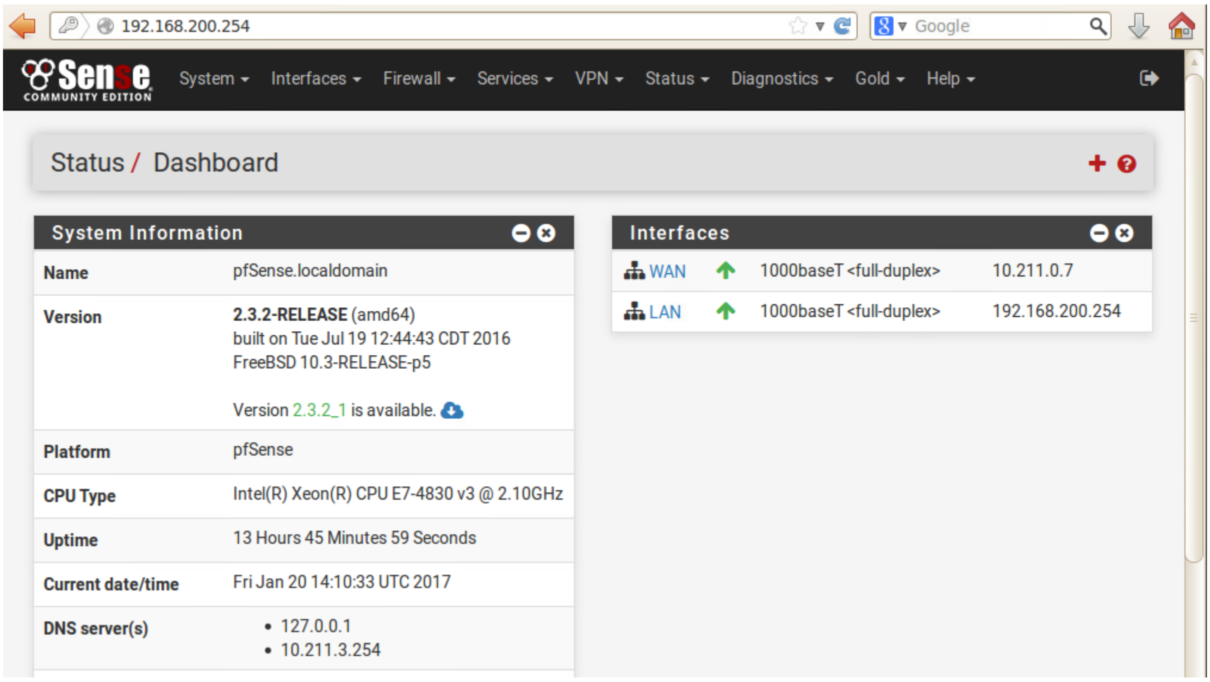
192.168.99.254

Username:

Password:

Login with the pfsense credentials

The default page is the **Status>Dashboard** which shows general information on the current state of the device.


Google
pfSense COMMUNITY EDITION
System Interfaces Firewall Services VPN Status Diagnostics Gold Help

192.168.200.254

Status / Dashboard

System Information

Name	pfSense.localdomain
Version	2.3.2-RELEASE (amd64) built on Tue Jul 19 12:44:43 CDT 2016 FreeBSD 10.3-RELEASE-p5 Version 2.3.2_1 is available.
Platform	pfSense
CPU Type	Intel(R) Xeon(R) CPU E7-4830 v3 @ 2.10GHz
Uptime	13 Hours 45 Minutes 59 Seconds
Current date/time	Fri Jan 20 14:10:33 UTC 2017
DNS server(s)	• 127.0.0.1 • 10.211.3.254

Interfaces

WAN	↑	1000baseT <full-duplex>	10.211.0.7
LAN	↑	1000baseT <full-duplex>	192.168.200.254

Q. From the dashboard, which interfaces are currently up?

Private LAN Network Connectivity

Layer 2 ARP

From the firewall web app, use the **Diagnostics menu**, to view the ARP table.

Q. Which physical MAC addresses are in the cache? For which IP Addresses?

From Ubuntu, check the physical MAC address using ip addr

Q. Does it match the entry for the Ubuntu IP Address in the Firewall ARP table? Yes/No

Layer 3 IP

From the firewall web app, use the **Diagnostics menu**, to check connectivity to the private network by using the web ping tool
Send each of the 192.168.*x*.254 and 192.168.x.7 interfaces some ICMP packets.

Q. Can you ping them? Yes/No

From the Ubuntu VM, use ping to check connectivity to the firewall interface and the lab DEFAULT_GW_ADDR

Q. Can you ping them? Yes/No

Configure DMZ Interface

From the web app, use the menu **Interfaces>OPT1** to edit the Optional 1 interface, **Enabling it**, renaming it to **DMZ**, and configure a static IP Address of **192.168.*y*.254** with a **24 bit subnet mask**.
Save and Apply (commit) the changes!

Check your configuration against your network diagram, using **Status>Interfaces**
Go back to the Dashboard **Status > Dashboard** and check the interface is up

Q. Is the DMZ interface shown as up? Yes/No

DMZ Network Connectivity

Layer 2 ARP

From the firewall web app, use the **Diagnostics menu**, to view the ARP cache.
Have the MAC addresses been added for the DMZ? Yes/No

On the Windows host, check the physical MAC address using **ipconfig /all**
Does it match the entry for the Windows IP Address in the Firewall ARP table? Yes/No
If the ARP entry does not appear for the DMZ, ping your Windows host from Ubuntu and refresh.

Q. Does it match the entry for the Windows IP Address in the Firewall ARP table now?
Yes/No

Routing

Now we will investigate the routing table on the firewall.

On the firewall, investigate the Routing Table (in Diagnostics), and identify how the device makes decisions on the routing of data packets it receives from the connected networks.

Q. What is the firewall's default gateway?

Default Stateful Firewalling

Test the default firewalling which PFSense automatically puts in place. It's based on the interfaces being connected to the different types of networks. Outside untrusted, optional DMZ less trusted, and inside most trusted. i.e. carry out due diligence on the default state!

Firewall Connectivity

From pfsense, using the Diagnostic>Ping page test connectivity to Windows and Ubuntu.

Q. Are they both responding? Yes/[✓ No]

If not run Wireshark and try again.

Q. Is the Windows host receiving/responding to the ICMP pkts from the Ping? [✓ Yes]/No

Host Network Connectivity

From the Ubuntu host, can you ping the DMZ – try the firewall DMZ interface and then the Windows host itself.

Q. Can you ping them? [✓ Yes]/No

From the Windows host, check the local interface is working using ping 192.168.*y*.7 and then check connectivity to the firewall interface at 192.168.*y*.254.

Q. Are they both responding? Yes/[✓ No]

Q. Why is the 192.168.y.254 interface dropping the ICMP packets? What does this mean in terms of default firewalling for PfSense? If you are not sure, go on to the next section.

Q. Similarly, are you able to browse the web from your Ubuntu VM? Yes/No

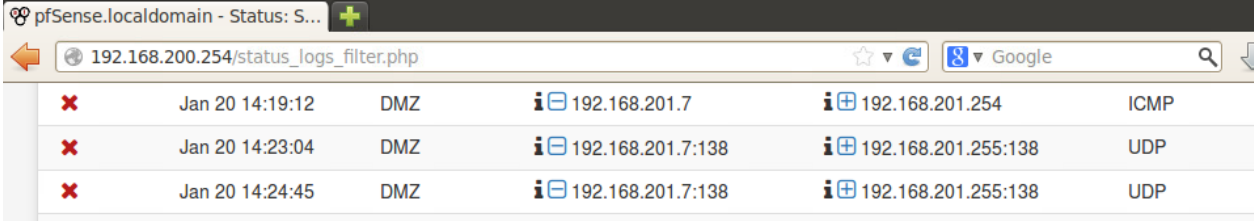
Q. What about from your Windows VM? Yes/No

Firewall Logs

From pfsense, check the firewall log from the **System>Logs** menu and then **Firewall tab**

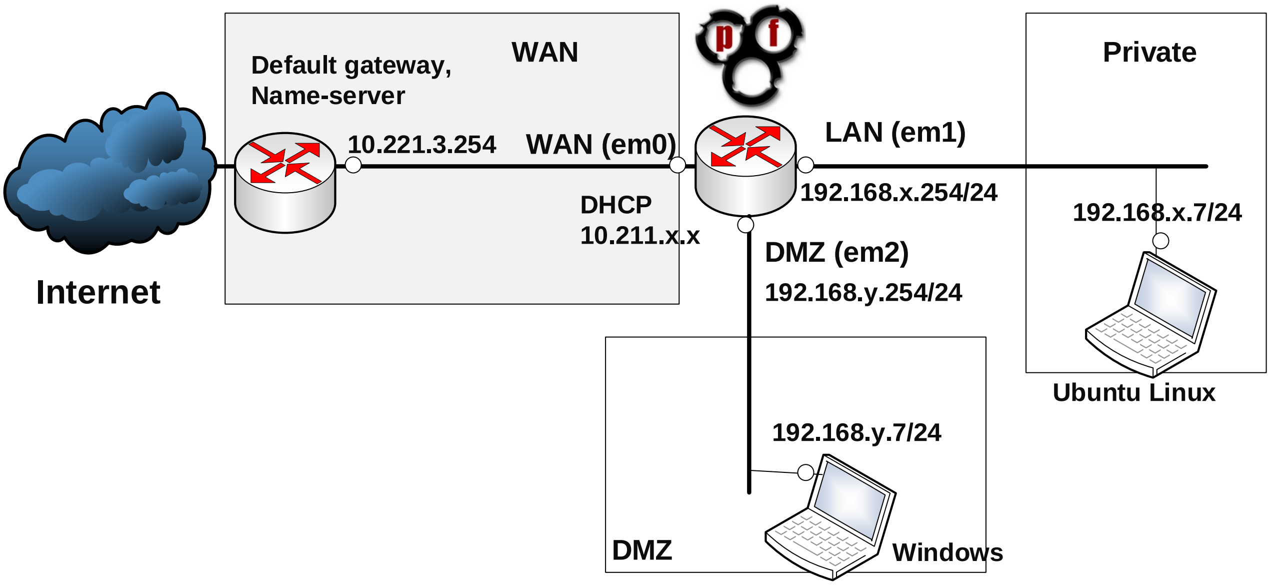
Q. Can you see the log entries for the pfsense dropping the ICMP packets at the 192.168.y.254 interface? Yes/No

You should see some log entries, such as


pfSense.localdomain - Status: S...
192.168.200.254/status_logs_filter.php
Google

✖	Jan 20 14:19:12	DMZ	192.168.201.7	192.168.201.254	ICMP
✖	Jan 20 14:23:04	DMZ	192.168.201.7:138	192.168.201.255:138	UDP
✖	Jan 20 14:24:45	DMZ	192.168.201.7:138	192.168.201.255:138	UDP

Note: not all default rule packets are logged, not all dropped packets are logged, and this also changes across different firewalls.



In the above diagram, draw on a copy of the pdf (or take a snip and save and edit), the **allowed traffic flows** of the current PfSense firewalling setup. Use arrows to indicate how the traffic is allow from/to where.
Maybe red for blocked, green for allowed, and maybe dotted for return traffic!

Default Firewall Rules

From the firewall web app, review the current rules applied to the 3 firewall interfaces using **Firewall>Rules (and then the interface tabs)**

Q. WAN What is the security stance being applied to the outside WAN interface?
Open/Closed

Q. Is any traffic allowed from the outside network through the WAN eth0 interface by default? Yes/No

Q. Would a ping to the WAN interface from another student be responded to? Yes/No
Ask another student to ping your WAN interface from their firewall/Ubuntu host to check

Q. LAN What is the security stance being applied to the inside LAN interface?
Open/Closed

Q. If the ***** means **any**, what type of traffic from which hosts is allowed from the LAN network to the DMZ?

Q. DMZ What is the security stance being applied to the DMZ interface?
Open/Closed

Q. If we wanted to allow ping tool to check connectivity from DMZ to other networks what would we have to do?

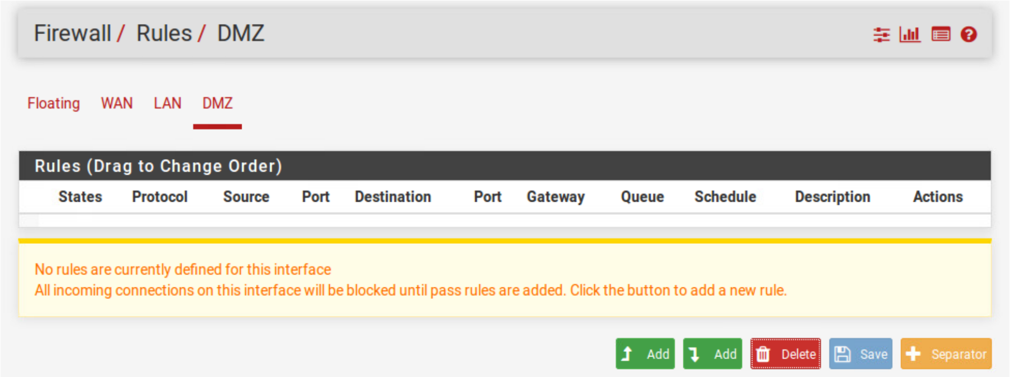
Configuring Stateful Firewalling

Create Firewall Rules

Now we will investigate configuration of the stateful firewalling which PfSense firewall provides.

Firewall Rule to allow ICMP from DMZ

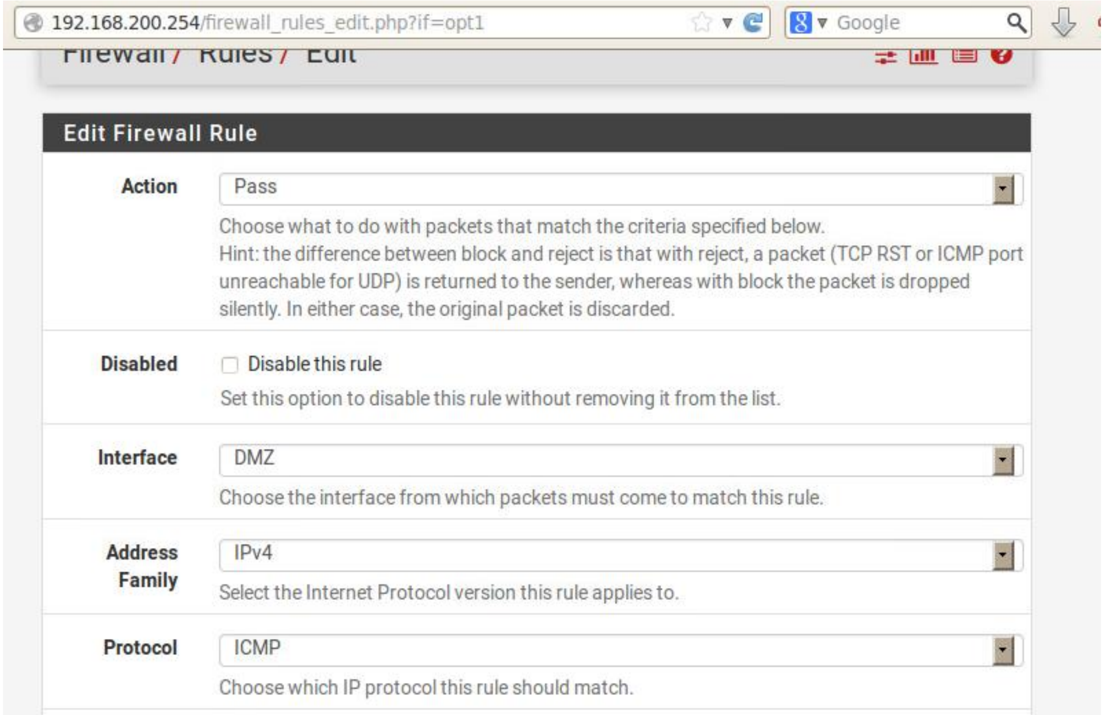
On the firewall, under the DMZ firewall rules, use the **Add button** create a rule which allows any host on the DMZ to use ICMP to any destination network address.



Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions

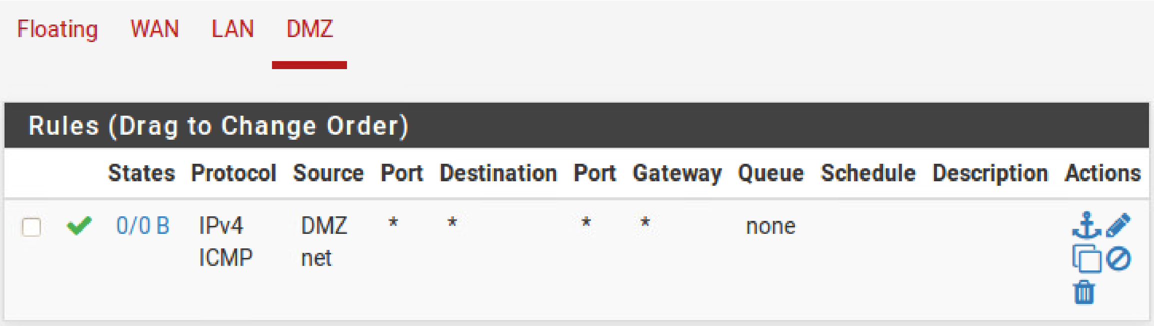
No rules are currently defined for this interface
All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule.



Edit Firewall Rule

Action	Pass Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled	□ Disable this rule Set this option to disable this rule without removing it from the list.
Interface	DMZ Choose the interface from which packets must come to match this rule.
Address Family	IPv4 Select the Internet Protocol version this rule applies to.
Protocol	ICMP Choose which IP protocol this rule should match.

Save, review the rule, and Apply if it looks ok.



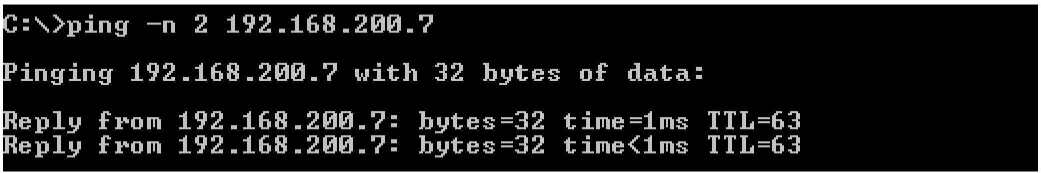
Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
□	✓	0/0 B	IPv4 ICMP	DMZ net	*	*	*	*	none			

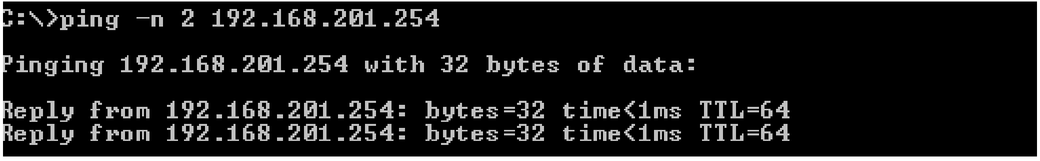
On the Windows host check that you now have connectivity to the LAN network, ping 192.168.*x*.254 and 192.168.*x*.7 interfaces.

Q. Do you have connectivity with the LAN Network interfaces? Yes/No

Q. How is this different from the Vyatta rules we had to create? (Hint: how many rules would we need on Vyatta?)



```
C:\>ping -n 2 192.168.200.7

Pinging 192.168.200.7 with 32 bytes of data:

Reply from 192.168.200.7: bytes=32 time=1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
```



```
C:\>ping -n 2 192.168.201.254

Pinging 192.168.201.254 with 32 bytes of data:

Reply from 192.168.201.254: bytes=32 time<1ms TTL=64
Reply from 192.168.201.254: bytes=32 time<1ms TTL=64
```

Firewall Rule to allow ICMP from Public WAN Network

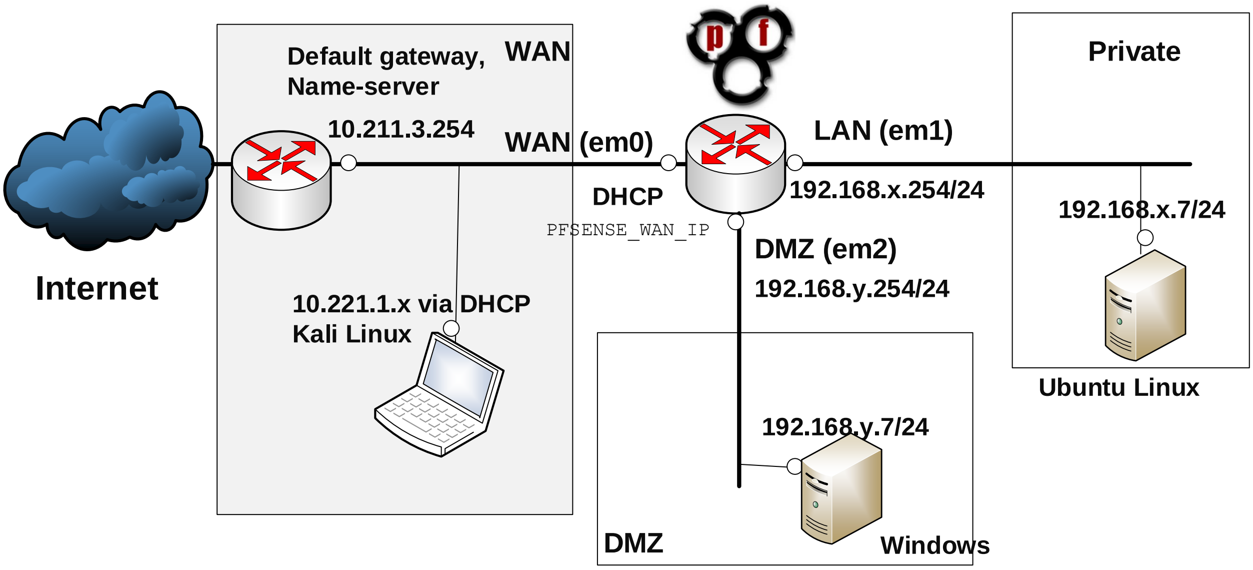
Test current setup using **Diagnostics>Ping** but use the f/w WAN interface as the Source Address to mimic ICMP traffic from an external network.

While doing this run Wireshark on the hosts to check if packets are getting through to them!

Q. From the firewall, can you ping the hosts in the DMZ and Private network from the WAN port?   Yes/No

Test current setup from Kali Linux VM

We will now use our Kali pen testing Linux VM on the external network to test this also.



Kali VM on the WAN network

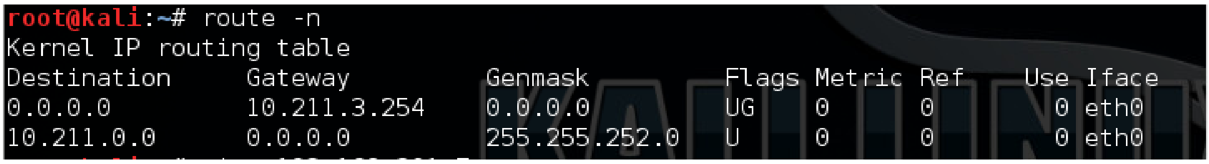
Power on, and open a console to the Kali VM. Log in to the VM.

On Kali VM, check it has received an IP address via DHCP.

On your Kali box, check the routing table:

```
        route -n
```

Q. what is the default root set up? Yes/No
Q.



```
root@kali:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.211.3.254    0.0.0.0         UG    0      0        0 eth0
10.211.0.0      0.0.0.0         255.255.252.0   U     0      0        0 eth0
```

Network Scan the Firewall WAN Interface

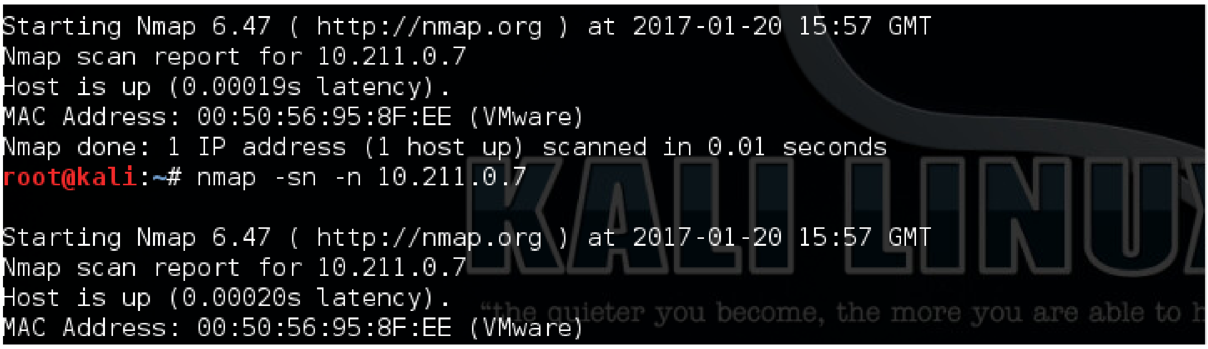
On Kali, run wireshark to monitor packets we send using our pen testing tools.

From Kali, use **nmap** to run a Host Discovery Scan on the pfsense WAN interface

```
        nmap -sn -n PFSENSE_WAN_IP
```

Note: syntax for host scan is slightly diff from Ubuntu nmap version (newer tool)

Q. Is nmap reporting the host is up? Yes/No
Q. Which protocol did it use to send packets? Why?



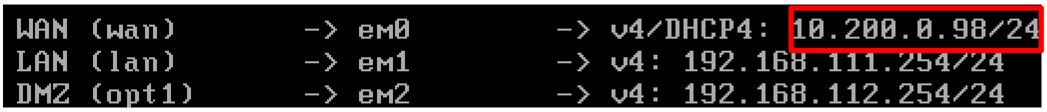
```
Starting Nmap 6.47 ( http://nmap.org ) at 2017-01-20 15:57 GMT
Nmap scan report for 10.211.0.7
Host is up (0.00019s latency).
MAC Address: 00:50:56:95:8F:EE (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.01 seconds
root@kali:~# nmap -sn -n 10.211.0.7

Starting Nmap 6.47 ( http://nmap.org ) at 2017-01-20 15:57 GMT
Nmap scan report for 10.211.0.7
Host is up (0.00020s latency).
MAC Address: 00:50:56:95:8F:EE (VMware)
```

Connect to Internal Networks

As we don't have WAN IP Addresses set up for our internal machines, we will need to point our Kali machine directly at the firewall WAN interface so it can connect.

On Kali VM, change the default gateway to your PfSense WAN interface, e.g.:

```
ip route change default via PFSENSE_WAN_IP
```

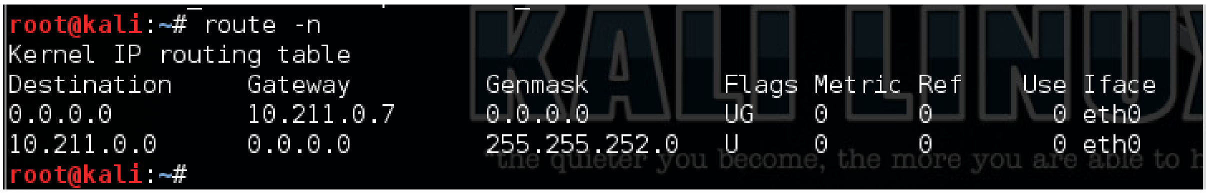


```
WAN (wan)        -> em0        -> v4/DHCP4: 10.200.0.98/24
LAN (lan)        -> em1        -> v4: 192.168.111.254/24
DMZ (opt1)       -> em2        -> v4: 192.168.112.254/24
```

On your Kali box, check the routing table:

```
route -n
```

You should see an output similar to the following with the default gateway set to your firewall outside interface:



```
root@kali:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.211.0.7      0.0.0.0         UG    0      0        0 eth0
10.211.0.0      0.0.0.0         255.255.252.0   U     0      0        0 eth0
root@kali:~#
```

From the Kali VM, ping both the Windows and Ubuntu VMs.

Q. Are you able to ping from Kali to your Windows server? Yes/No

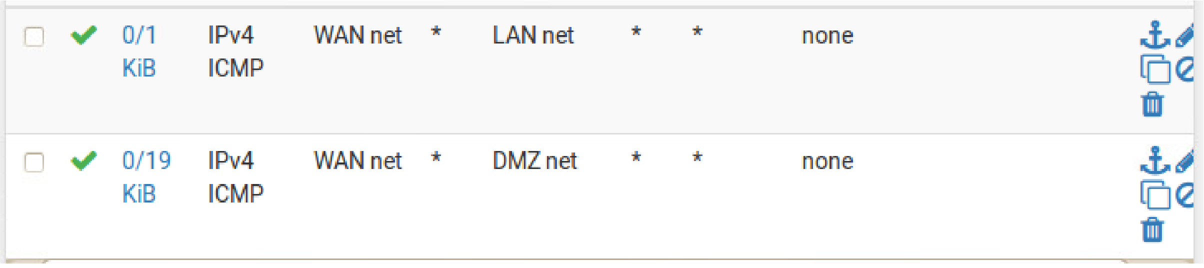
Q. Are you able to ping from Kali to your Ubuntu VM? Yes/No

Check the firewall log, and you should see these being blocked.

Add ICMP Firewall Rules for WAN>DMZ and WAM>LAN

On the firewall, create a rule which **allows** the Public (WAN) network to be able to ping (send ICMP traffic) both to the DMZ and Private network.

Rules should be similar to:



□	✓	0/1 KiB	IPv4 ICMP	WAN net	*	LAN net	*	*	none
□	✓	0/19 KiB	IPv4 ICMP	WAN net	*	DMZ net	*	*	none

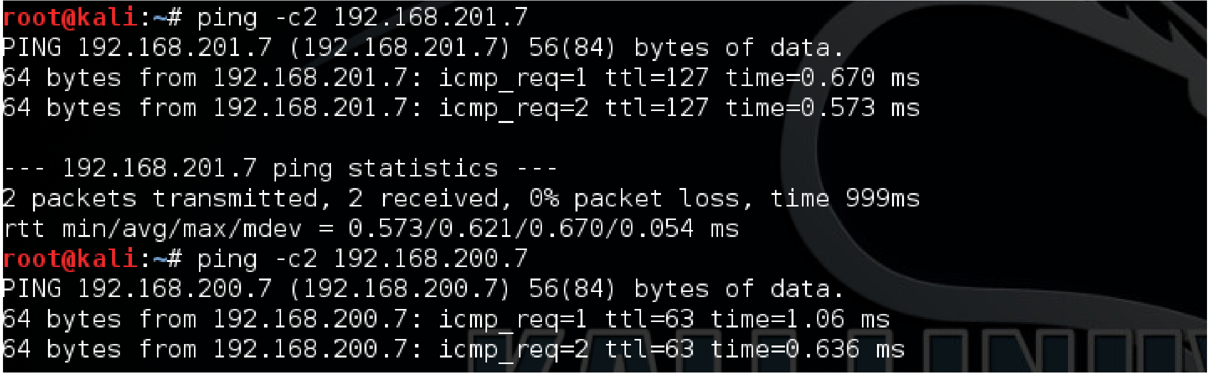
Test Rules

From the Kali VM, ping both the Windows and Ubuntu VMs.

Q. Are you able to ping from Kali to your Windows server? Yes/No

Q. Are you able to ping from Kali to your Ubuntu VM? Yes/No

You should be able to ping because rules should now allow for this.



```
root@kali:~# ping -c2 192.168.201.7
PING 192.168.201.7 (192.168.201.7) 56(84) bytes of data.
64 bytes from 192.168.201.7: icmp_req=1 ttl=127 time=0.670 ms
64 bytes from 192.168.201.7: icmp_req=2 ttl=127 time=0.573 ms

--- 192.168.201.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.573/0.621/0.670/0.054 ms
root@kali:~# ping -c2 192.168.200.7
PING 192.168.200.7 (192.168.200.7) 56(84) bytes of data.
64 bytes from 192.168.200.7: icmp_req=1 ttl=63 time=1.06 ms
64 bytes from 192.168.200.7: icmp_req=2 ttl=63 time=0.636 ms
```

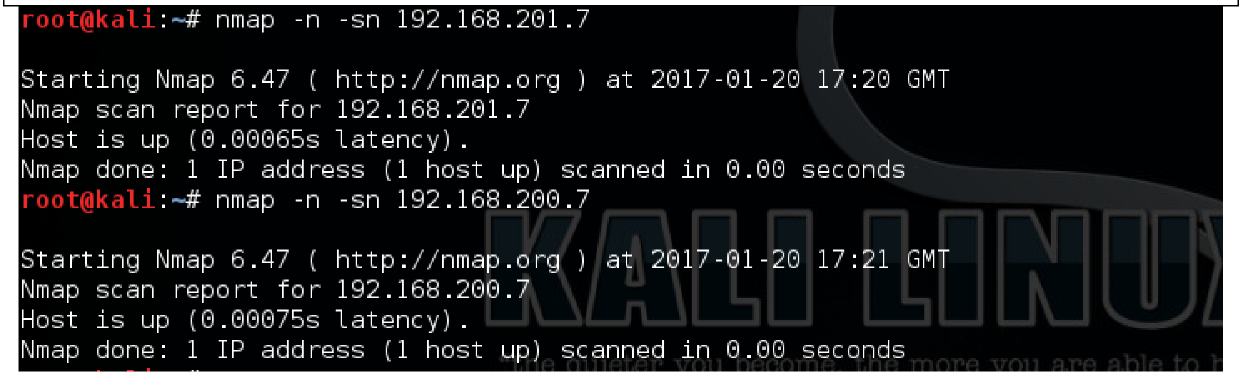
Network Scan the Firewall WAN Interface

On Kali, run wireshark to monitor packets we send using our pen testing tools.

From Kali, use **nmap** to run a Host Discovery Scan on the host VMs

```
        nmap –sn –n 192.168.x.7

        nmap –sn –n 192.168.y.7
```

Q. Is nmap reporting the host is up? Yes/No
Q. Which protocol did it successfully use to find the host? Why?



```
root@kali:~# nmap -n -sn 192.168.201.7

Starting Nmap 6.47 ( http://nmap.org ) at 2017-01-20 17:20 GMT
Nmap scan report for 192.168.201.7
Host is up (0.00065s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.00 seconds
root@kali:~# nmap -n -sn 192.168.200.7

Starting Nmap 6.47 ( http://nmap.org ) at 2017-01-20 17:21 GMT
Nmap scan report for 192.168.200.7
Host is up (0.00075s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.00 seconds
```

Test WAN>DMZ Web Server

From your DMZ Windows server check a web server is running using **netstat –ap tcp**

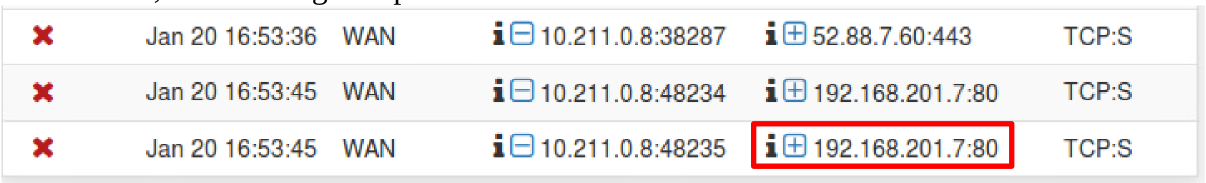
Q. Is a web server running? Yes/No **Q.** What port is it running on? (–n flag might help)

From Kali, attempt to connect to your DMZ Windows web server (port 80).

Q. Is it successful? Yes/No

Check the Firewall Logs

In your firewall web interface, navigate to **Status>System logs**, and if necessary in the **firewall** tab, filter the logs for port 80.

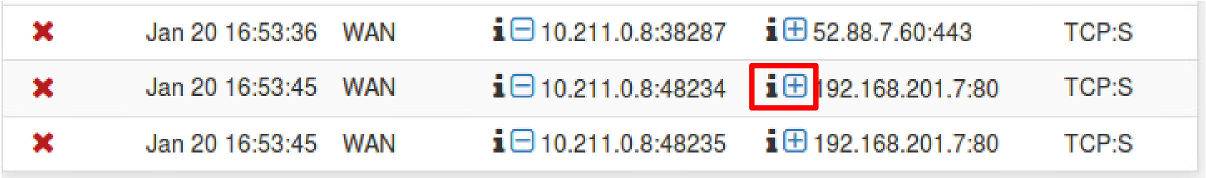


✖	Jan 20 16:53:36	WAN	i ⊟ 10.211.0.8:38287	i ⊞ 52.88.7.60:443	TCP:S
✖	Jan 20 16:53:45	WAN	i ⊟ 10.211.0.8:48234	i ⊞ 192.168.201.7:80	TCP:S
✖	Jan 20 16:53:45	WAN	i ⊟ 10.211.0.8:48235	i ⊞ 192.168.201.7:80	TCP:S

Q. Can you identify the web traffic destined for the DMZ server that has been dropped by the firewall? Yes/No

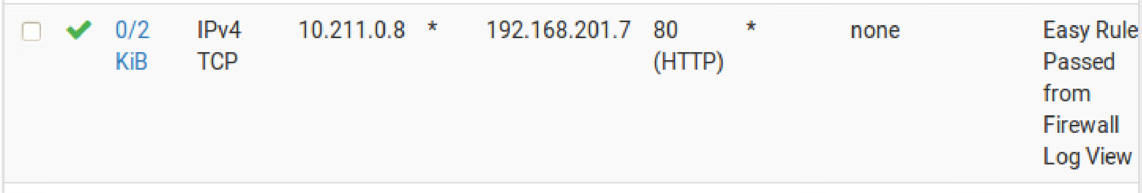
Create a WAN>DMZ Web Server Firewall Rule Automatically

Now click on the little "plus" sign next to the destination IP address, highlighted above. This will add a rule automatically to pass this particular type of connection.



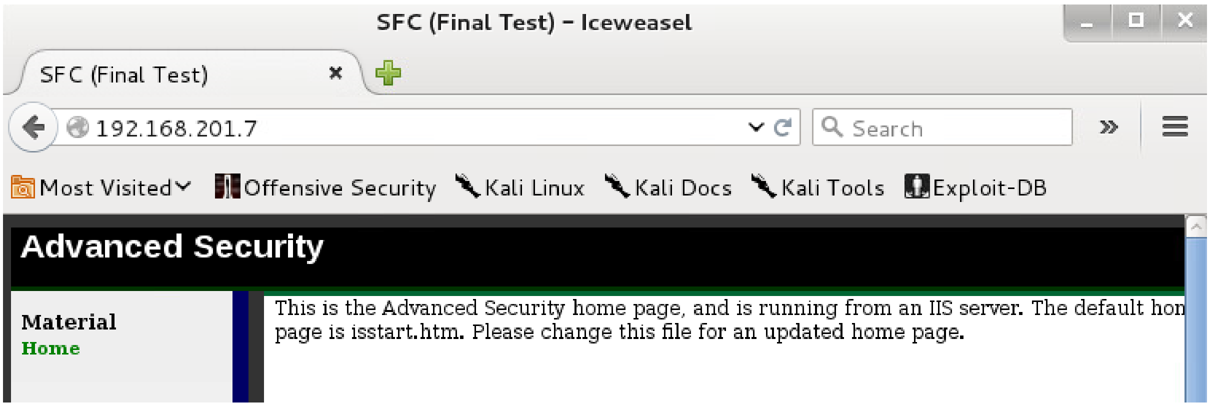
✖	Jan 20 16:53:36	WAN	i ⊟ 10.211.0.8:38287	i ⊞ 52.88.7.60:443	TCP:S
✖	Jan 20 16:53:45	WAN	i ⊟ 10.211.0.8:48234	i ⊞ 192.168.201.7:80	TCP:S
✖	Jan 20 16:53:45	WAN	i ⊟ 10.211.0.8:48235	i ⊞ 192.168.201.7:80	TCP:S

Review the rule; it should be similar to:



☐	✓	0/2 KiB	IPv4 TCP	10.211.0.8	*	192.168.201.7	80 (HTTP)	*	none	Easy Rule Passed from Firewall Log View

Test by connecting from your Kali VM to your Windows web server.
Q. Does it now work? Yes/No



Advanced Security

Material
Home

This is the Advanced Security home page, and is running from an IIS server. The default hon page is isstart.htm. Please change this file for an updated home page.

From Kali, use **nmap** to run a Port Scan on the host VM to check what the firewall is now allowing the outside Kali system to have access to

```
        nmap -sS -n 192.168.x.7
```

Q. Which services are open on the DMZ Windows server?

On pfsense, review the firewall rule that was automatically added to allow access.

Q. Will this allow any host to connect to our DMZ webserver?

To test, we could try to connect by spoofing the source IP Address:

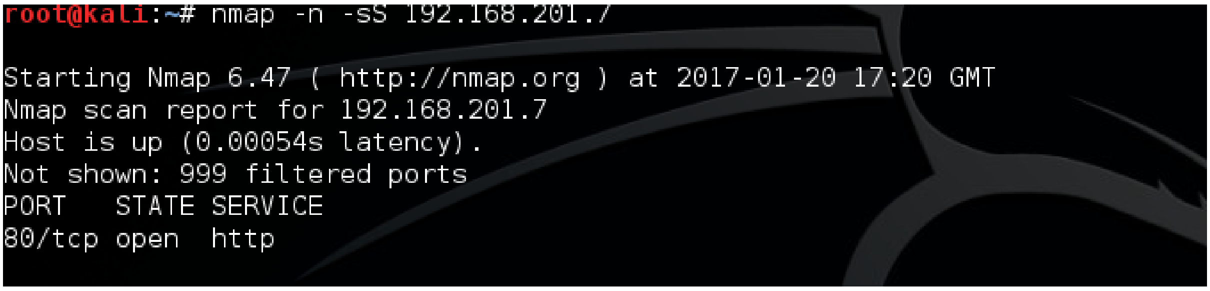
```
        nmap -n -e eth0 -S 10.211.3.KALI+50 192.168.y.7
```

Q. Was the spoofed source IP scan able to check the DMZ webserver port being open?

Yes/No

Q. Why did the spoofed source scan not work?

The **non-spoofed port scan** output should be similar to:



```
root@kali:~# nmap -n -sS 192.168.201.7

Starting Nmap 6.47 ( http://nmap.org ) at 2017-01-20 17:20 GMT
Nmap scan report for 192.168.201.7
Host is up (0.00054s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
80/tcp open  http
```

Instead, you could get another student to compare with a normal port scan of your DMZ sever from their Kali VM. (they would have to change their Kali default gw)

Q. What is the difference with their output?

WAN>DMZ Web Server Firewall Rule - Any Hosts

Typically you want to allow any machine from outside your network to access the forward facing DMZ web server.

Edit the rule to allow **any source IP Address** to access the web server, rather than just the Kali IP Address.

Get another student to try a normal port scan of your DMZ sever from their Kali VM.

Get them to connect to your web server with a browser also.

Q. Did the port scan show the web server port open? **Q.** Was the other students Kali browser able to connect now?

(it should now connect if your rules are correct)

WAN>DMZ - Allow Other Public Services

From the Windows VM, check the FTP and Telnet services are running with **netstat** and on which ports.

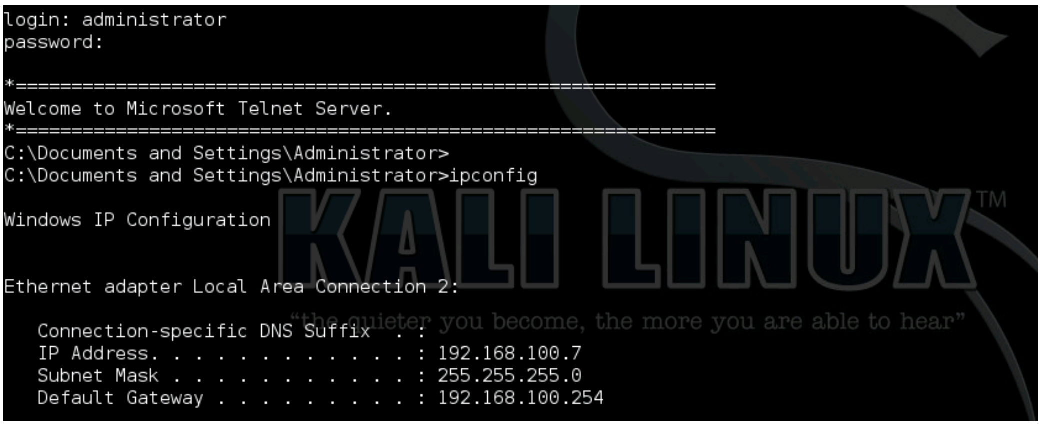
Q. Are the services running? Yes/No **Q.** What ports are they running on?

Add rules to allow **DMZ server FTP and Telnet** from **only the Kali VM** (perhaps that is a sys admin machine out on the Internet)

Test each of these services from the WAN network using your Kali Linux VM. Telnet and FTP can be tested from the command line clients, and check with a port scan using nmap!

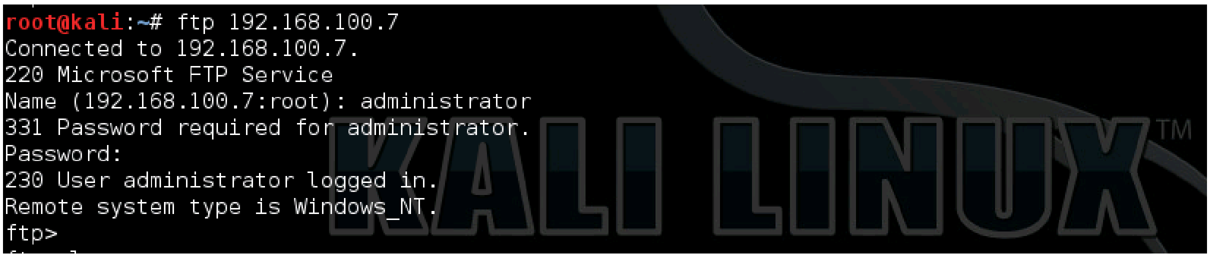
Q. Is Telnet working? Yes/No

Telnet access from Kali should look similar to:



```
login: administrator
password:

*===============================================================
Welcome to Microsoft Telnet Server.
*===============================================================
C:\Documents and Settings\Administrator>
C:\Documents and Settings\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . :
   IP Address. . . . . . . . . . . . : 192.168.100.7
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254
```

FTP access from Kali should look similar to:



```
root@kali:~# ftp 192.168.100.7
Connected to 192.168.100.7.
220 Microsoft FTP Service
Name (192.168.100.7:root): administrator
331 Password required for administrator.
Password:
230 User administrator logged in.
Remote system type is Windows_NT.
ftp>
```

Muli-channel FTP Protocol - Optional Challenge (only if you have extra time)

While connected to FTP server, try the **help** command. Try listing the current FTP directory using the **ls** command.

Q. Does the FTP list **ls** command return anything? Yes/No

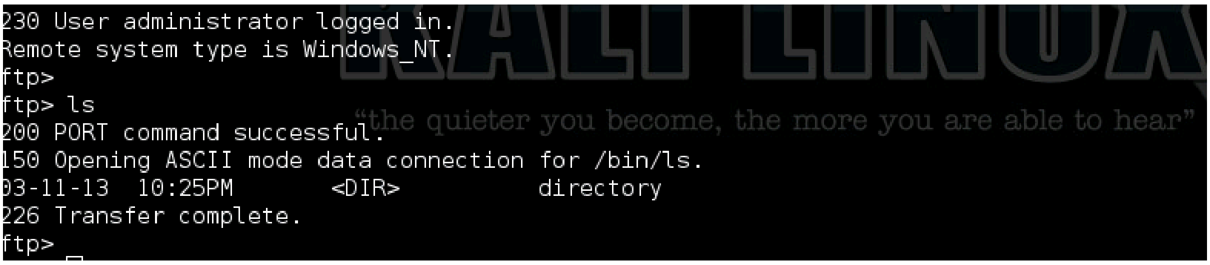
It shouldn’t as FTP opens a 2nd channel to send data (src port 20)

Run Wireshark and try to analyse the traffic, and also try reviewing the firewall logs and find traffic from the DMZ server to the Kali VM with source port 20.

Now add a rule to allow the data channel.

Q. Is FTP ls working? Yes/No
Q. Outline the rule:

The working ls command should look like:



```
230 User administrator logged in.
Remote system type is Windows_NT.
ftp>
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
03-11-13  10:25PM       <DIR>          directory
226 Transfer complete.
ftp> _
```

NAT

Now we will investigate the NAT setup on the Pfsense firewall.

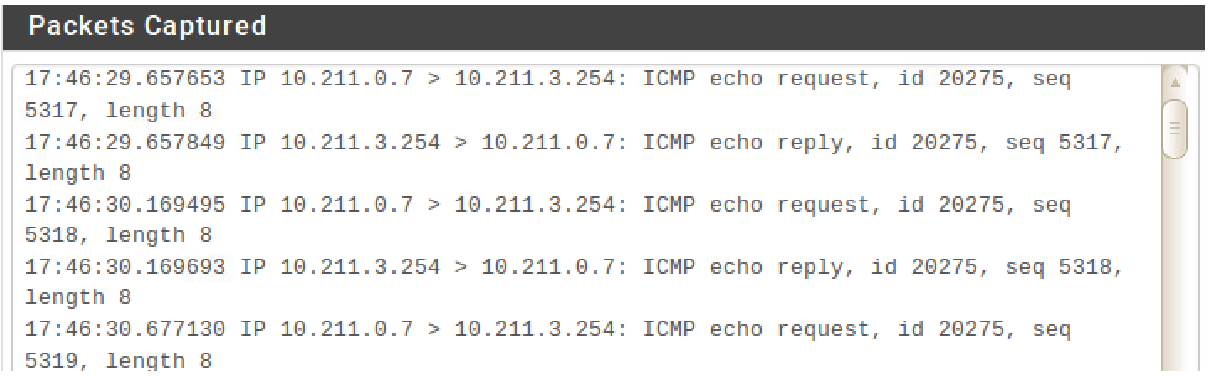
Run the packet capture tool on the firewall using **Diagnostics > Packet Capture** select the WAN interface to capture on.

Ping the public network gateway at DEFAULT_GW_ADDR from both the Windows host and the Linux host.

Stop the trace, and view the captured packets.

Q. Which IP address is shown as the source IP Address in the captured packet output?

Q. Why?



Packets Captured

```
17:46:29.657653 IP 10.211.0.7 > 10.211.3.254: ICMP echo request, id 20275, seq
5317, length 8
17:46:29.657849 IP 10.211.3.254 > 10.211.0.7: ICMP echo reply, id 20275, seq 5317,
length 8
17:46:30.169495 IP 10.211.0.7 > 10.211.3.254: ICMP echo request, id 20275, seq
5318, length 8
17:46:30.169693 IP 10.211.3.254 > 10.211.0.7: ICMP echo reply, id 20275, seq 5318,
length 8
17:46:30.677130 IP 10.211.0.7 > 10.211.3.254: ICMP echo request, id 20275, seq
5319, length 8
```

Pfsense has some default NAT configured!

Similarly, from Kali, run Wireshark from the menu, or the command prompt:

```
        wireshark &
```

From your DMZ or inside VM, ping your Kali VM.

From Kali, to anlayse our Wireshark capture, add the following display filter so we filter only traffic to and from our Kali VM:

```
        ip.addr==IP_Address_Kali
```

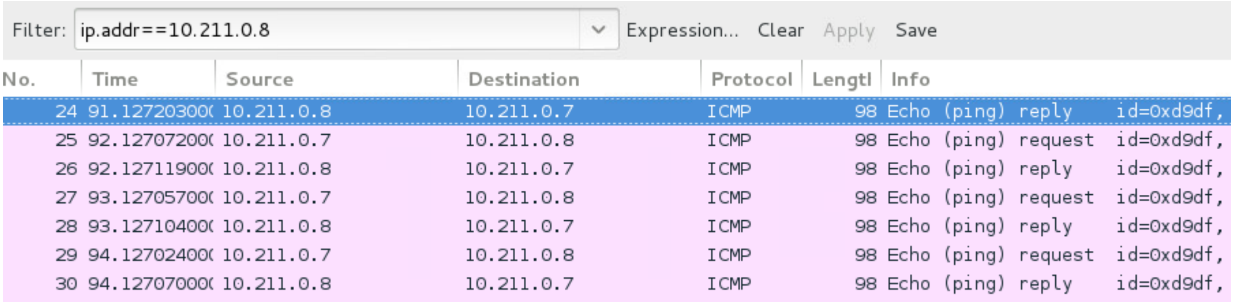
Q. Why is it just a single address?

Q. What type of NAT is this?

Static / Masquerade NAT

Q. Reviewing your packet captures, can you explain the high number of ICMP packets being sent from your PfSense WAN interface?

Kali Wireshark analysis:



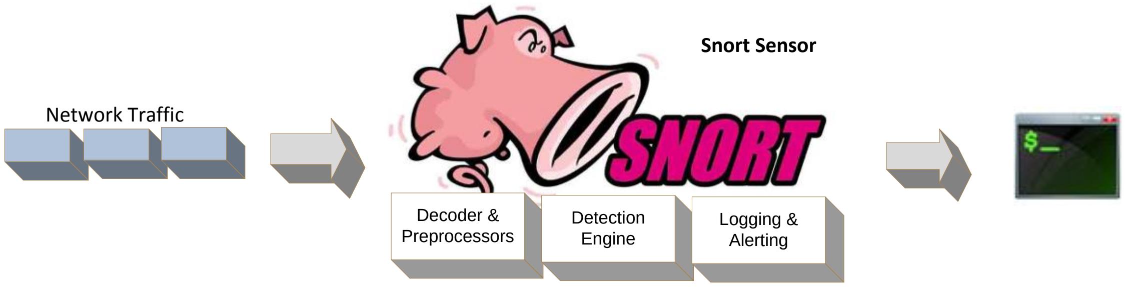
Filter: ip.addr==10.211.0.8

No.	Time	Source	Destination	Protocol	Lengtl	Info
24	91.12720300(	10.211.0.8	10.211.0.7	ICMP	98	Echo (ping) reply id=0xd9df,
25	92.12707200(	10.211.0.7	10.211.0.8	ICMP	98	Echo (ping) request id=0xd9df,
26	92.12711900(	10.211.0.8	10.211.0.7	ICMP	98	Echo (ping) reply id=0xd9df,
27	93.12705700(	10.211.0.7	10.211.0.8	ICMP	98	Echo (ping) request id=0xd9df,
28	93.12710400(	10.211.0.8	10.211.0.7	ICMP	98	Echo (ping) reply id=0xd9df,
29	94.12702400(	10.211.0.7	10.211.0.8	ICMP	98	Echo (ping) request id=0xd9df,
30	94.12707000(	10.211.0.8	10.211.0.7	ICMP	98	Echo (ping) reply id=0xd9df,

Snort IDS

On the DMZ Windows server, check the interfaces available to Snort with:
  C:\Snort\bin> **snort -W**

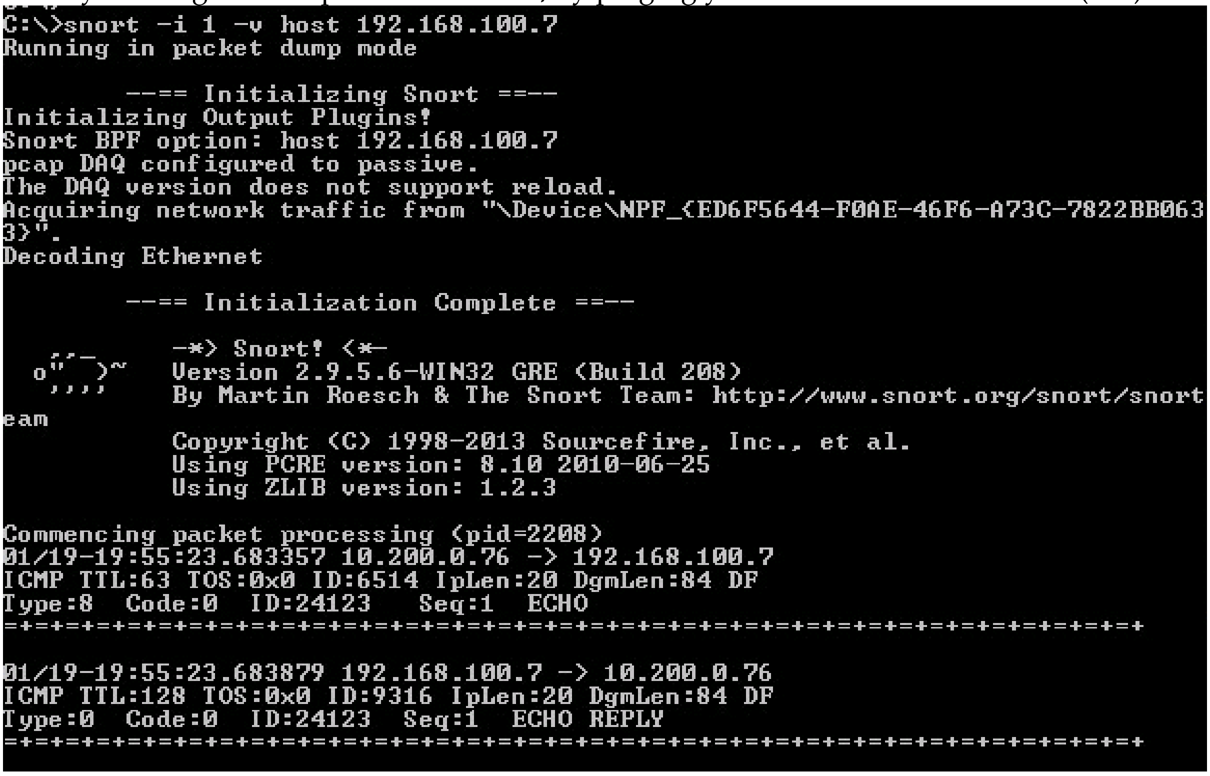
Run the **Snort IDS sensor as a packet sniffer** to test it can inspect traffic



Start by using **–v to only display the TCP/UDP/ICMP headers** of packets captured:
(and adding the filter of host IP_Adress to only display traffic to/from your local interface)

```
snort -v -i 3  host 192.168.y.7
```

Test by sending 1 ICMP packet from **Kali,** by pinging your DMZ Windows server. (-c 1)



```
C:\>snort -i 1 -v host 192.168.100.7
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Snort BPF option: host 192.168.100.7
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{ED6F5644-F0AE-46F6-A73C-7822BB063
3}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.5.6-WIN32 GRE (Build 208)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort
eam
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=2208)
01/19-19:55:23.683357 10.200.0.76 -> 192.168.100.7
ICMP TTL:63 TOS:0x0 ID:6514 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:24123   Seq:1  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/19-19:55:23.683879 192.168.100.7 -> 10.200.0.76
ICMP TTL:128 TOS:0x0 ID:9316 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:24123  Seq:1  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

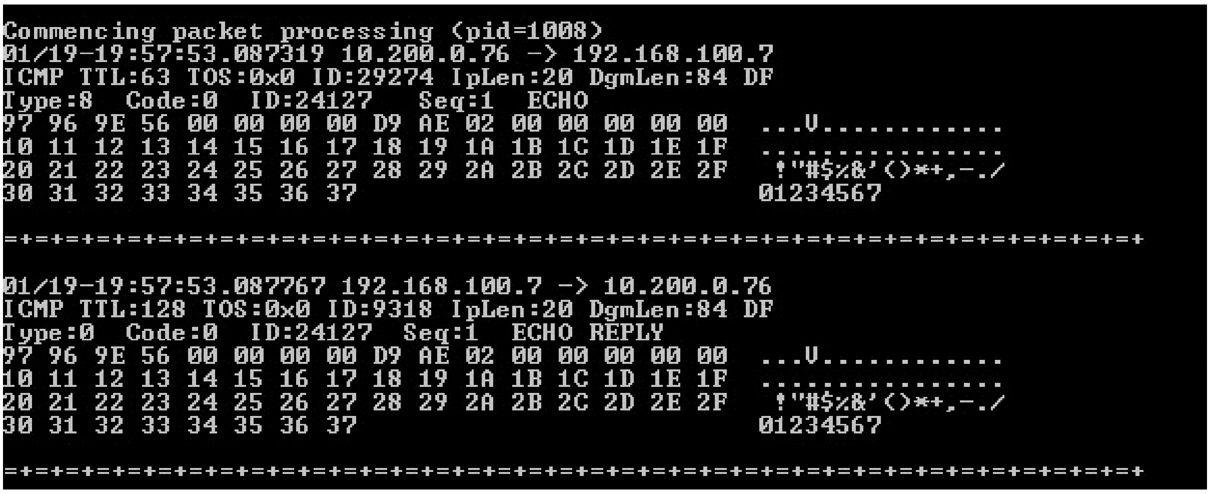
Q. What is the contents of the payload of an ping ICMP packet from Kali Linux?

(cant see it? Move on to next question)

Now try using **–vd** to display the headers and the packet payloads, and test with another single ping from Kali.

Q. What is the contents of the payload of an ping ICMP packet from Kali Linux?

Q. Can you identify the HEX values for 01234?



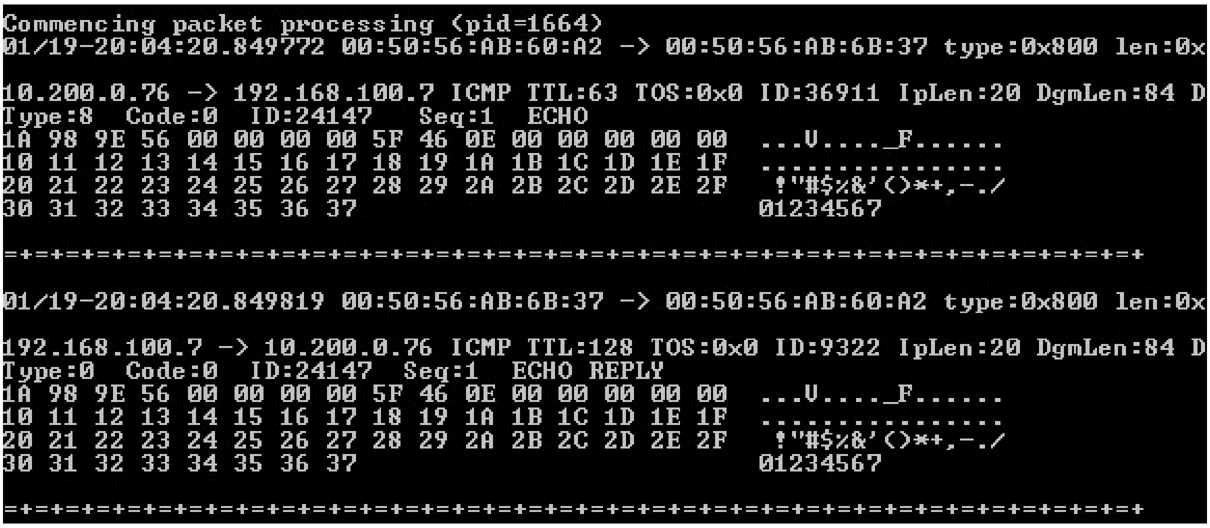
```
Commencing packet processing (pid=1008)
01/19-19:57:53.087319 10.200.0.76 -> 192.168.100.7
ICMP TTL:63 TOS:0x0 ID:29274 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:24127   Seq:1  ECHO
97 96 9E 56 00 00 00 00 D9 AE 02 00 00 00 00 00  ...V............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/19-19:57:53.087767 192.168.100.7 -> 10.200.0.76
ICMP TTL:128 TOS:0x0 ID:9318 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:24127  Seq:1  ECHO REPLY
97 96 9E 56 00 00 00 00 D9 AE 02 00 00 00 00 00  ...V............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Now try using **–vde** to display the headers and the packet payloads, and the layer 2 MAC Addresses,

Test with another single ping from Kali.

Q. Can you identify the DMZ Windows server MAC Address in the output?   Yes/No

Q. Can you identify the Kali MAC Address in the output?   Yes/No
Q. Why not? And what is the dest MAC address otherwise?

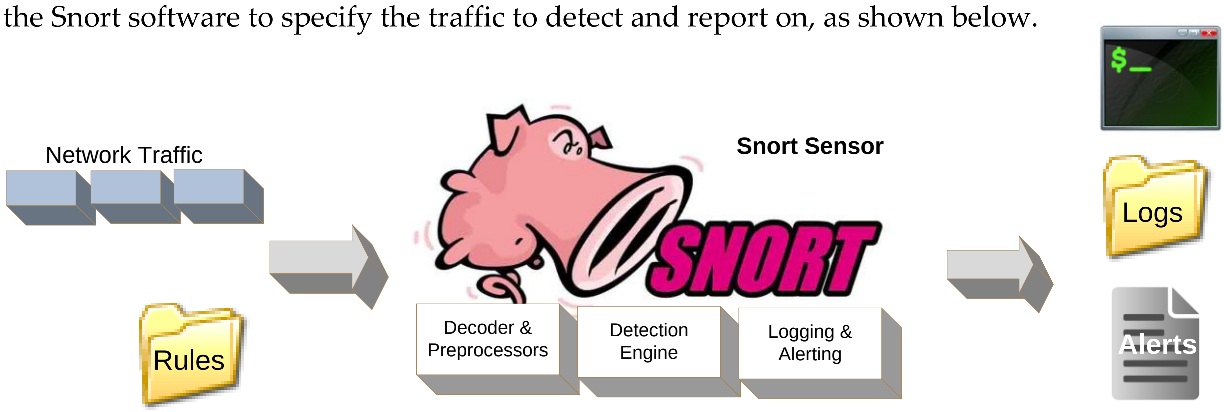


```
Commencing packet processing (pid=1664)
01/19-20:04:20.849772 00:50:56:AB:60:A2 -> 00:50:56:AB:6B:37 type:0x800 len:0x

10.200.0.76 -> 192.168.100.7 ICMP TTL:63 TOS:0x0 ID:36911 IpLen:20 DgmLen:84 D
Type:8  Code:0  ID:24147   Seq:1  ECHO
1A 98 9E 56 00 00 00 00 5F 46 0E 00 00 00 00 00  ...V...._F......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/19-20:04:20.849819 00:50:56:AB:6B:37 -> 00:50:56:AB:60:A2 type:0x800 len:0x

192.168.100.7 -> 10.200.0.76 ICMP TTL:128 TOS:0x0 ID:9322 IpLen:20 DgmLen:84 D
Type:0  Code:0  ID:24147  Seq:1  ECHO REPLY
1A 98 9E 56 00 00 00 00 5F 46 0E 00 00 00 00 00  ...V...._F......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

You should be able to confirm your answer by checking with Wireshark and a ping from Windows VM, or check the interface config on the pfsense f/w.

Signature-based IDS Sensor

To run Snort as a signature-based IDS Sensor, a *Snort Detection Rules file* is used as input to the Snort software to specify the traffic to detect and report on, as shown below.



Snort can be used as a signature-based IDS, with signatures defined in the rules. Signature-based IDS can compare signatures against each packets application protocol data (the packet payload). This is sometimes called *deep packet inspection*.

Firewalls tend to filter at lower levels as inspecting deep inside each packet can drastically reduce throughput. As IDS sensors are typically inspecting copies of the packets off-line, there is no effect on throughput. This means IDS sensors can inspect packets much more thoroughly than most firewalls.

Create DMZ Recon IDS Detection Rule

On the DMZ Windows server using Windows Explorer and a text editor, create a Snort Detection Rules file **`c:\snort\rules\rules.txt`** where our Snort rules will be created.

To detect possible reconnaissance against the DMZ servers, a simple IDS detection rule can be created:

```
alert icmp any any -> any any ( msg: "Possible Recon detected"; sid:10000;)
```

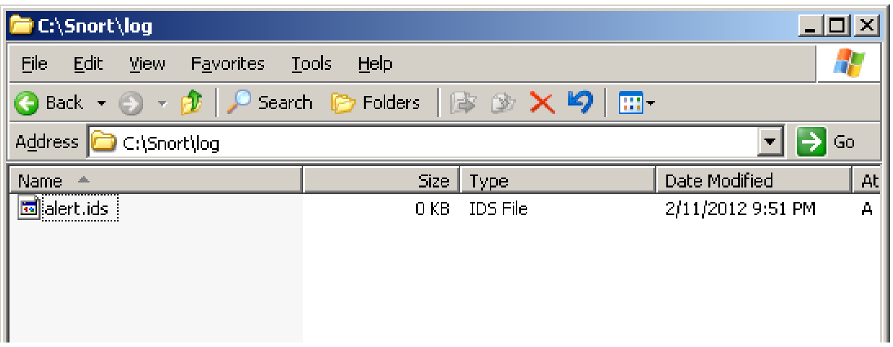
Test Detection Rule

From Windows VM, navigate to **`C:\rules`**
Create a log folder, using mkdir command
Run the Snort IDS sensor using the rules.txt file as its detection signatures, and logging alerts to a Snort log folder, using:

```
snort -dev -i INDEX -p -c rules\rules.txt -l log -K ascii -k none
```

(you can use a filter by adding to reduced output by adding on the end: **`host 192.168.y.7`**)

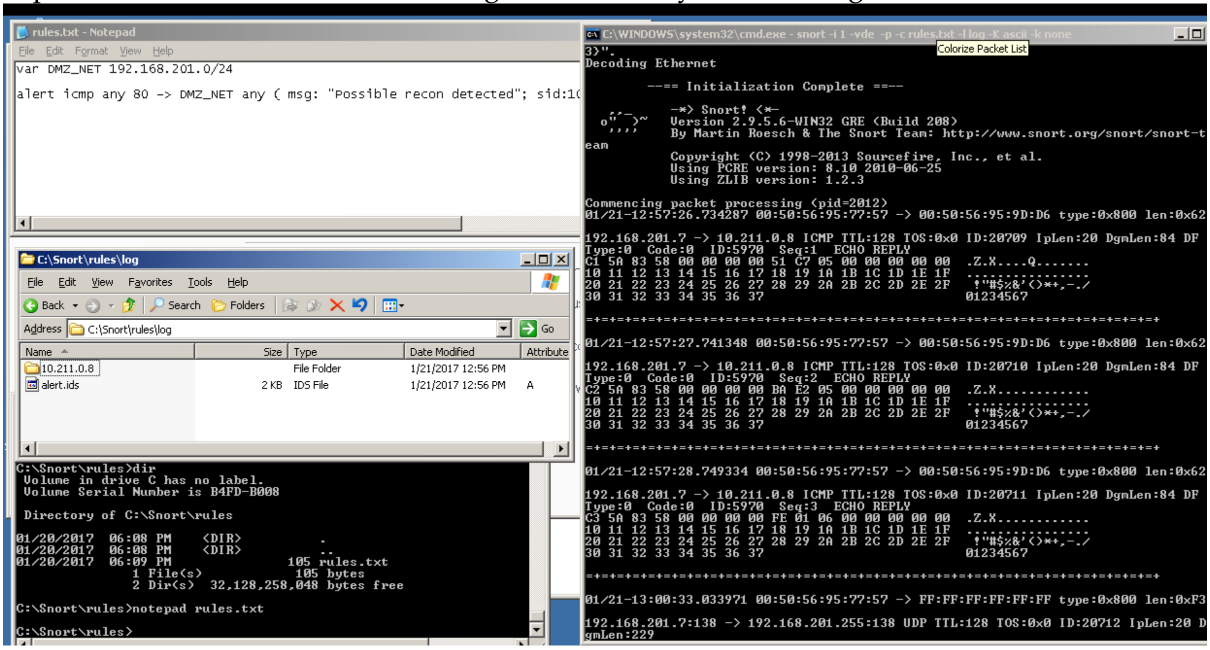
Now monitor the output `log` folder with Windows Explorer. Right click the file panel and select **View>Details**, as shown below. The timestamp and size of the file can now be viewed, and used to check if a rule has caused an alert to be raised, and packet details to be logged.



Address: C:\Snort\log

Name	Size	Type	Date Modified	At
alert.ids	0 KB	IDS File	2/11/2012 9:51 PM	A

Resize/rearrange your windows, so you can see the Snort rules, output alert.ids file in explorer, the console window running snort, and any other testing windows.



```
var DMZ_NET 192.168.201.0/24

alert icmp any 80 -> DMZ_NET any ( msg: "Possible recon detected"; sid:1(
```

Address: C:\Snort\rules\log

Name	Size	Type	Date Modified	Attribute
10.211.0.8		File Folder	1/21/2017 12:56 PM	
alert.ids	2 KB	IDS File	1/21/2017 12:56 PM	A

```
C:\Snort\rules>dir
 Volume in drive C has no label.
 Volume Serial Number is B4FD-B008

 Directory of C:\Snort\rules

01/20/2017  06:08 PM    <DIR>          .
01/20/2017  06:08 PM    <DIR>          ..
01/20/2017  06:09 PM               105 rules.txt
               1 File(s)            105 bytes
               2 Dir(s)  32,128,258,048 bytes free

C:\Snort\rules>notepad rules.txt

C:\Snort\rules>
```

C:\WINDOWS\system32\cmd.exe - snort -i 1 -vde -p -c rules.txt -l log -K ascii -k none

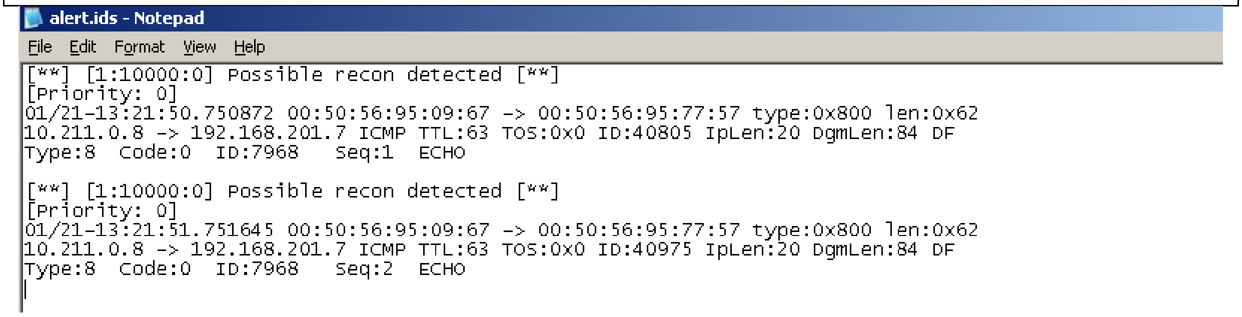
```
3}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.5.6-WIN32 GRE (Build 208)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
eam
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=2012)
01/21-12:57:26.734287 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x62

192.168.201.7 -> 10.211.0.8 ICMP TTL:128 TOS:0x0 ID:20709 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:5970  Seq:1  ECHO REPLY
C1 5A 83 58 00 00 00 00 51 C7 05 00 00 00 00 00  .Z.X....Q.......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/21-12:57:27.741348 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x62

192.168.201.7 -> 10.211.0.8 ICMP TTL:128 TOS:0x0 ID:20710 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:5970  Seq:2  ECHO REPLY
C2 5A 83 58 00 00 00 00 BA E2 05 00 00 00 00 00  .Z.X............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/21-12:57:28.749334 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x62

192.168.201.7 -> 10.211.0.8 ICMP TTL:128 TOS:0x0 ID:20711 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:5970  Seq:3  ECHO REPLY
C3 5A 83 58 00 00 00 00 FE 01 06 00 00 00 00 00  .Z.X............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/21-13:00:33.033971 00:50:56:95:77:57 -> FF:FF:FF:FF:FF:FF type:0x800 len:0xF3

192.168.201.7:138 -> 192.168.201.255:138 UDP TTL:128 TOS:0x0 ID:20712 IpLen:20 D
gmLen:229
```

From Kali, test the rule by sending 2 ICMP packets to your DMZ Windows server, using

```
ping -c 2 192.168.y.7
```

Check if an alert has been generated, by checking the log folder, and the timestamp/size of file being updated. Stop Snort running with <CTRL+C>, and open the alert file in an editor.

Q. Has an alert been generated for the recon traffic?   Yes/No



```
[**] [1:10000:0] Possible recon detected [**]
[Priority: 0]
01/21-13:21:50.750872 00:50:56:95:09:67 -> 00:50:56:95:77:57 type:0x800 len:0x62
10.211.0.8 -> 192.168.201.7 ICMP TTL:63 TOS:0x0 ID:40805 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:7968   Seq:1  ECHO

[**] [1:10000:0] Possible recon detected [**]
[Priority: 0]
01/21-13:21:51.751645 00:50:56:95:09:67 -> 00:50:56:95:77:57 type:0x800 len:0x62
10.211.0.8 -> 192.168.201.7 ICMP TTL:63 TOS:0x0 ID:40975 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:7968   Seq:2  ECHO
```

Try using ping to send ICMP packets from Windows VM to another machine, such as the Ubuntu VM on the inside network.

Q. Has an alert been generated for this traffic?   Yes/No

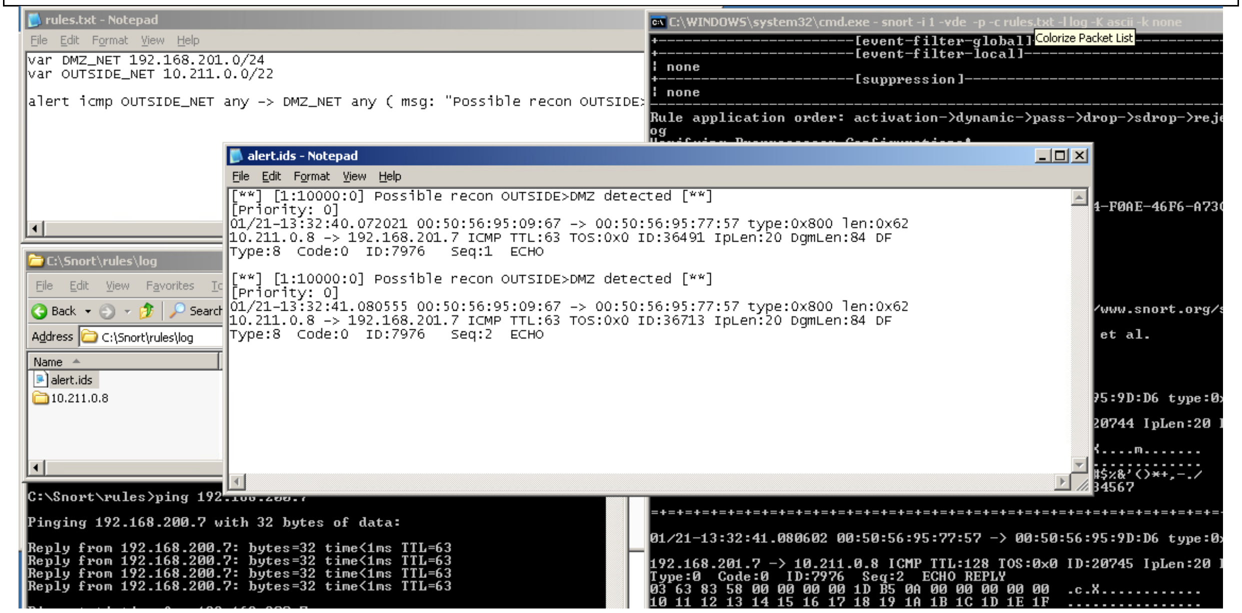
Refine Rule for Networks

We can improve the rule by specifying networks the recon comes from and where it is aimed at.

```
var DMZ_NET 192.168.y.0/24
var OUTSIDE_NET 10.211.0.0/22

alert icmp any any -> DMZ_NET any ( msg: "Possible Recon OUTSIDE>DMZ net
detected"; sid:10000;)
```

Note: if you have to rerun tests, its easiest to stop Snort, delete the alert file, and restart Snort.

Q. Has an alert been generated for the recon traffic?   Yes/No


rules.txt - Notepad
File Edit Format View Help
C:\WINDOWS\system32\cmd.exe - snort -i 1 -vde -p -c rules.txt -l log -K ascii -k none
Colorize Packet List
alert.ids - Notepad
C:\Snort\rules\log
File Edit View Favorites
Back
Search
Address C:\Snort\rules\log
Name
alert.ids
10.211.0.8

```
var DMZ_NET 192.168.201.0/24
var OUTSIDE_NET 10.211.0.0/22

alert icmp OUTSIDE_NET any -> DMZ_NET any ( msg: "Possible recon OUTSIDE>
```

```
+------------------------[event-filter-global]
+------------------------[event-filter-local]--------------------------
| none
+------------------------[suppression]---------------------------------
| none
------------------------------------------------------------------------
Rule application order: activation->dynamic->pass->drop->sdrop->reje
og
```

```
[**] [1:10000:0] Possible recon OUTSIDE>DMZ detected [**]
[Priority: 0]
01/21-13:32:40.072021 00:50:56:95:09:67 -> 00:50:56:95:77:57 type:0x800 len:0x62
10.211.0.8 -> 192.168.201.7 ICMP TTL:63 TOS:0x0 ID:36491 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:7976   Seq:1  ECHO

[**] [1:10000:0] Possible recon OUTSIDE>DMZ detected [**]
[Priority: 0]
01/21-13:32:41.080555 00:50:56:95:09:67 -> 00:50:56:95:77:57 type:0x800 len:0x62
10.211.0.8 -> 192.168.201.7 ICMP TTL:63 TOS:0x0 ID:36713 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:7976   Seq:2  ECHO
```

```
C:\Snort\rules>ping 192.

Pinging 192.168.200.7 with 32 bytes of data:

Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

01/21-13:32:41.080602 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x

192.168.201.7 -> 10.211.0.8 ICMP TTL:128 TOS:0x0 ID:20745 IpLen:20
Type:0  Code:0  ID:7976  Seq:2  ECHO REPLY
03 63 83 58 00 00 00 00 1D B5 0A 00 00 00 00 00  .c.X............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
```

Try using ping to send ICMP packets from Windows VM to another machine, such as the Ubuntu VM on the inside network.

Q. Has an alert been generated for this traffic?   Yes/No

Stop the Snort sensor, delete the alert files, start Snort again.

From Kali, use **nmap** to run a Host Discovery Recon Scan on the DMZ VM

```
        nmap -sn -n 192.168.y.7
```

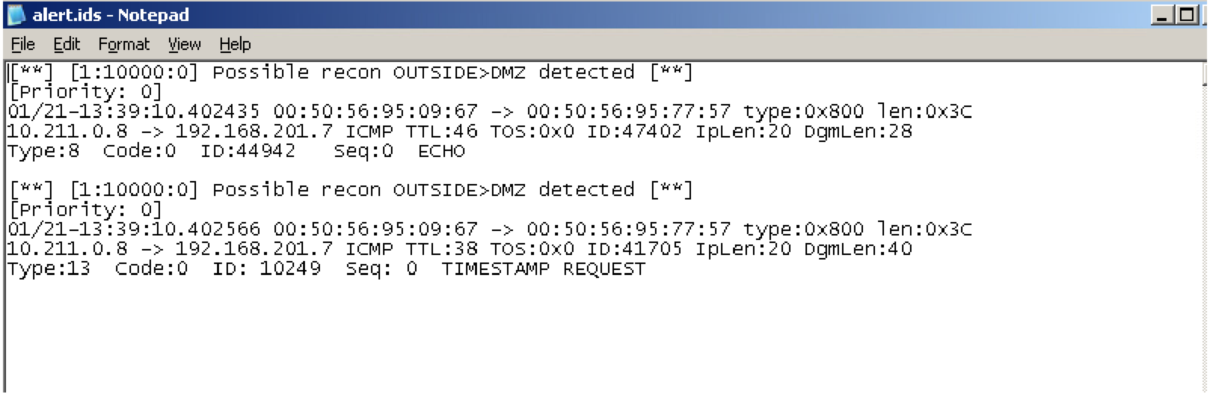
Q. Is nmap reporting the host is up? Yes/No

Q. Has an alert been generated for the recon traffic? Yes/No

Review the alerts raised.

Q. Which types of ICMP packets have nmap sent in its Host Discovery recon traffic?

Nmap uses various different packets to look for hosts:



```
[**] [1:10000:0] Possible recon OUTSIDE>DMZ detected [**]
[Priority: 0]
01/21-13:39:10.402435 00:50:56:95:09:67 -> 00:50:56:95:77:57 type:0x800 len:0x3C
10.211.0.8 -> 192.168.201.7 ICMP TTL:46 TOS:0x0 ID:47402 IpLen:20 DgmLen:28
Type:8  Code:0  ID:44942   Seq:0  ECHO

[**] [1:10000:0] Possible recon OUTSIDE>DMZ detected [**]
[Priority: 0]
01/21-13:39:10.402566 00:50:56:95:09:67 -> 00:50:56:95:77:57 type:0x800 len:0x3C
10.211.0.8 -> 192.168.201.7 ICMP TTL:38 TOS:0x0 ID:41705 IpLen:20 DgmLen:40
Type:13  Code:0  ID: 10249  Seq: 0  TIMESTAMP REQUEST
```

Snort Rule Creation Methodology

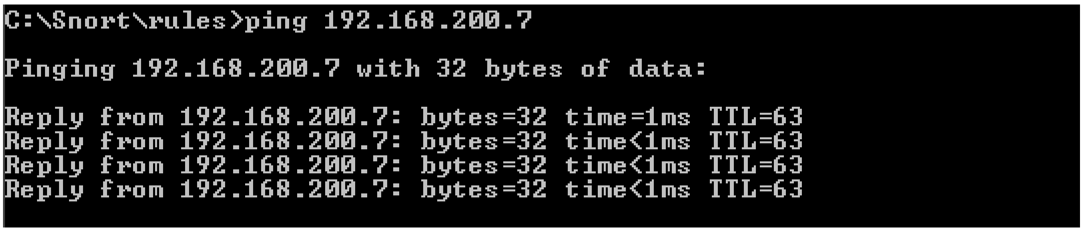
To develop your own Snort rules, it is recommended that the vulnerability be researched or the intrusion traffic generated in a controlled environment while Snort or another packet sniffer capture the network traffic. The traffic should then be analysed, along with researching any protocols involved, and a Snort rule(s) created based in any triggering conditions (Roesch, 1999).

1- Analysis of Intrusion Traffic

Before creating rules, we should analyse the intrusion and the traffic we want to detect. We will now look to detect recon coming from the DMZ server.

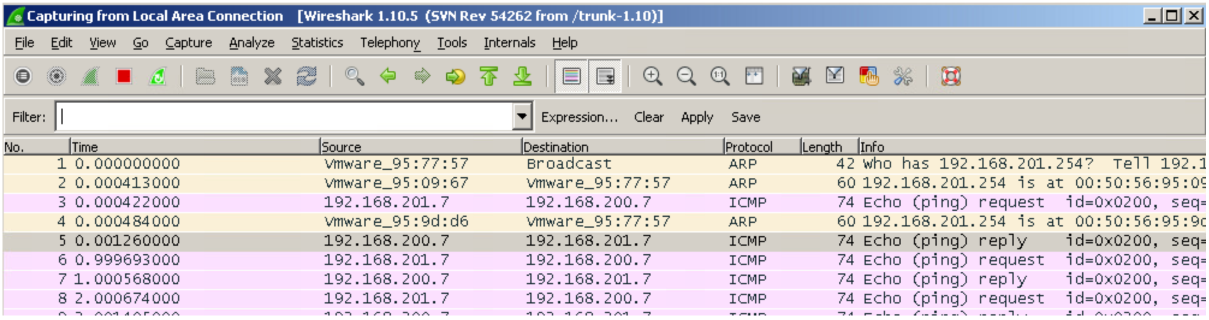
To capture traffic, run Wireshark on the Windows VM, and listen on the interface which you can send packets through.

From the local command line, open a new command windows, and Ping another machine, an outside network server, or inside VM, as shown below.



```
C:\Snort\rules>ping 192.168.200.7

Pinging 192.168.200.7 with 32 bytes of data:

Reply from 192.168.200.7: bytes=32 time=1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
Reply from 192.168.200.7: bytes=32 time<1ms TTL=63
```

Stop the wireshark capture, and filter out the IP of the Windows VM. Find the Ping packets, as shown.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Vmware_95:77:57	Broadcast	ARP	42	Who has 192.168.201.254? Tell 192.1
2	0.000413000	Vmware_95:09:67	Vmware_95:77:57	ARP	60	192.168.201.254 is at 00:50:56:95:09
3	0.000422000	192.168.201.7	192.168.200.7	ICMP	74	Echo (ping) request id=0x0200, seq=
4	0.000484000	Vmware_95:9d:d6	Vmware_95:77:57	ARP	60	192.168.201.254 is at 00:50:56:95:9c
5	0.001260000	192.168.200.7	192.168.201.7	ICMP	74	Echo (ping) reply id=0x0200, seq=
6	0.999693000	192.168.201.7	192.168.200.7	ICMP	74	Echo (ping) request id=0x0200, seq=
7	1.000568000	192.168.200.7	192.168.201.7	ICMP	74	Echo (ping) reply id=0x0200, seq=
8	2.000674000	192.168.201.7	192.168.200.7	ICMP	74	Echo (ping) request id=0x0200, seq=

Select an ICMP Ping packet and review the details.

Q. What is the protocol of the packets generated by the ping tool?

Q. What is the payload (data) within the packet?

Q. Is this the same for all the Ping packets? Yes/No

Q. What are the 2 types of ICMP packets associated with the ping? Yes/No

2- Create Snort Detection Rule

Create a new Snort rule (add it to the rules.txt file with a new SID) to detect the DMZ server outgoing Ping packets only. The alert should be "DMZ Ping Detected – Possible Reconnaissance"

3- Test the rule

First delete everything in the log folder with Windows Explorer.
Run Snort sensor again, while monitoring the output log directory as before.
Generate traffic using ping from the command line, while monitoring the log file for alerts.

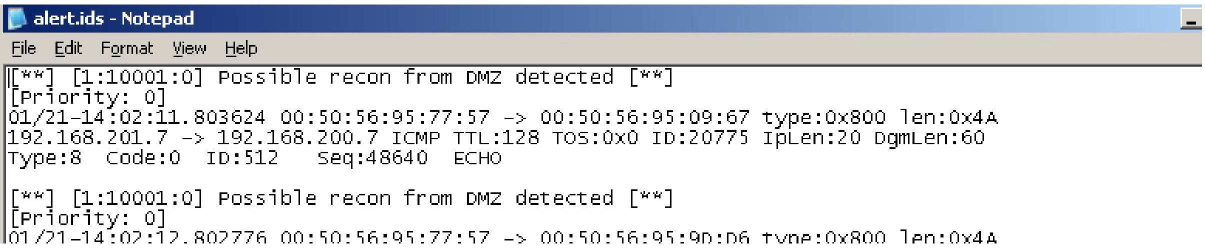
Q. What is the working Snort rule?

Q. How might this signature differ for Ping traffic from the DMZ server only to the DMZ network?

Q. How many alerts did the ping generate?

Q. Why this many alerts?

The alert file should look similar to the following:



```
[**] [1:10001:0] Possible recon from DMZ detected [**]
[Priority: 0]
01/21-14:02:11.803624 00:50:56:95:77:57 -> 00:50:56:95:09:67 type:0x800 len:0x4A
192.168.201.7 -> 192.168.200.7 ICMP TTL:128 TOS:0x0 ID:20775 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:48640  ECHO

[**] [1:10001:0] Possible recon from DMZ detected [**]
[Priority: 0]
01/21-14:02:12.802776 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x4A
```

4- Refine the Rule (repeat)

Change the Snort ICMP detection rule to only alert on Pings which are outgoing from your host system and to only the DMZ network. The alert should be “Possible Recon Pivoting Detected - DMZ>DMZ servers”

Test the changed rule by pinging the pfsense 192.168.y.254 gateway address.

Q. How many alerts did the ping generate?

Q. What is the working Snort rule?

Add a new Snort ICMP detection rule to only alert on ICMP recon which are generated within the local DMZ network, but are destined for another network . (Hint: the negation operator ! might help)

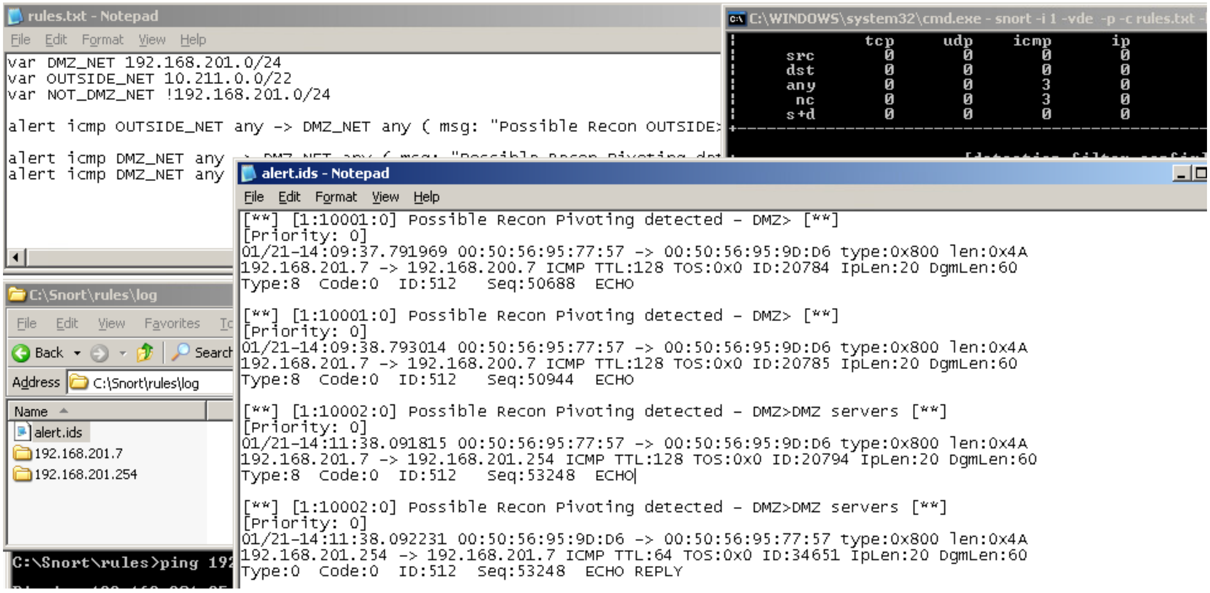
The alert should be “Possible Recon Pivoting Detected - DMZ>NON_DMZ_NETWORK”

Test the rule **DOES NOT** fire for Pings on the local network first.

Now test the rule for Pings from the Windows VM to an external network. Ping server on the inside network, then a web server on the internet, such as **google.com**.

Q. What is the working Snort rule?

The alert file should look similar to the following:



rules.txt - Notepad
File Edit Format View Help
C:\WINDOWS\system32\cmd.exe - snort -i 1 -vde -p -c rules.txt -
C:\Snort\rules\log
File Edit View Favorites
Back Search
Address C:\Snort\rules\log
Name
alert.ids
192.168.201.7
192.168.201.254
alert.ids - Notepad
File Edit Format View Help

```
var DMZ_NET 192.168.201.0/24
var OUTSIDE_NET 10.211.0.0/22
var NOT_DMZ_NET !192.168.201.0/24

alert icmp OUTSIDE_NET any -> DMZ_NET any ( msg: "Possible Recon OUTSIDE>

alert icmp DMZ_NET any
alert icmp DMZ_NET any
```

```
|             tcp      udp     icmp       ip
|     src       0        0        0        0
|     dst       0        0        0        0
|     any       0        0        3        0
|      nc       0        0        3        0
|     s+d       0        0        0        0
+------------------------------------------------
```

```
C:\Snort\rules>ping 192
```

```
[**] [1:10001:0] Possible Recon Pivoting detected - DMZ> [**]
[Priority: 0]
01/21-14:09:37.791969 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x4A
192.168.201.7 -> 192.168.200.7 ICMP TTL:128 TOS:0x0 ID:20784 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:50688  ECHO

[**] [1:10001:0] Possible Recon Pivoting detected - DMZ> [**]
[Priority: 0]
01/21-14:09:38.793014 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x4A
192.168.201.7 -> 192.168.200.7 ICMP TTL:128 TOS:0x0 ID:20785 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:50944  ECHO

[**] [1:10002:0] Possible Recon Pivoting detected - DMZ>DMZ servers [**]
[Priority: 0]
01/21-14:11:38.091815 00:50:56:95:77:57 -> 00:50:56:95:9D:D6 type:0x800 len:0x4A
192.168.201.7 -> 192.168.201.254 ICMP TTL:128 TOS:0x0 ID:20794 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512   Seq:53248  ECHO

[**] [1:10002:0] Possible Recon Pivoting detected - DMZ>DMZ servers [**]
[Priority: 0]
01/21-14:11:38.092231 00:50:56:95:9D:D6 -> 00:50:56:95:77:57 type:0x800 len:0x4A
192.168.201.254 -> 192.168.201.7 ICMP TTL:64 TOS:0x0 ID:34651 IpLen:20 DgmLen:60
Type:0  Code:0  ID:512   Seq:53248  ECHO REPLY
```

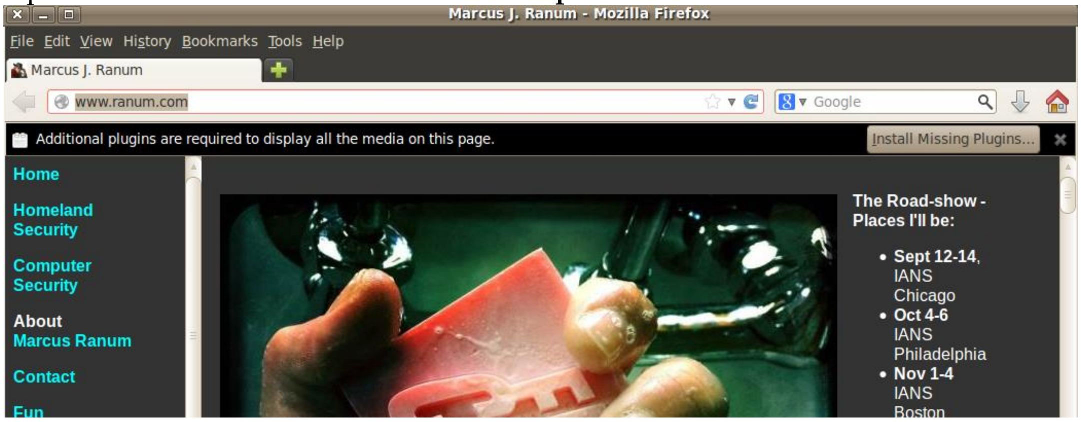
User Policy Breach IDS Detection Rule

Now we will set up detection on the Linux VM on the inside network, initially for some user misuse. The users from the inside network can access the Internet but are not allowed to browse certain content. First we will detect policy breaches of looking at websites with the **inappropriate content** about **cyber warfare**.

1- Analysis of Intrusion Traffic

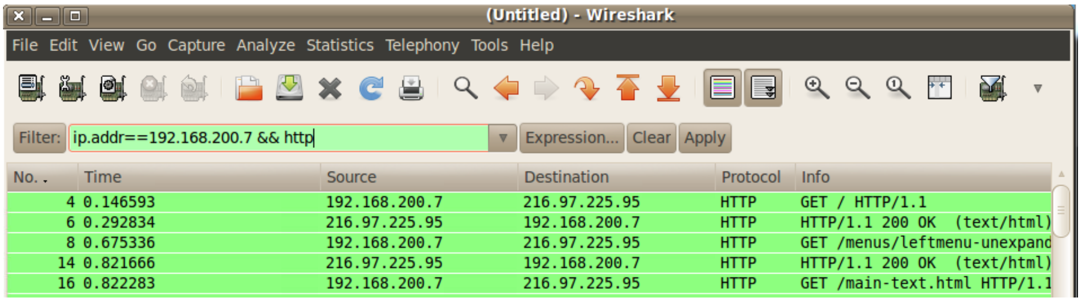
To capture traffic, run Wireshark on the Linux VM, and listen on the interface which you can send packets through.

Open a browser and access the website **http://www.ranum.com**



Marcus J. Ranum - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Marcus J. Ranum
www.ranum.com
Google
Additional plugins are required to display all the media on this page.
Install Missing Plugins...
Home
Homeland Security
Computer Security
About Marcus Ranum
Contact
Fun
The Road-show - Places I'll be:
Sept 12-14, IANS Chicago
Oct 4-6 IANS Philadelphia
Nov 1-4 IANS Boston

Stop the wireshark capture, and Filter out the http traffic for your Linux VM.



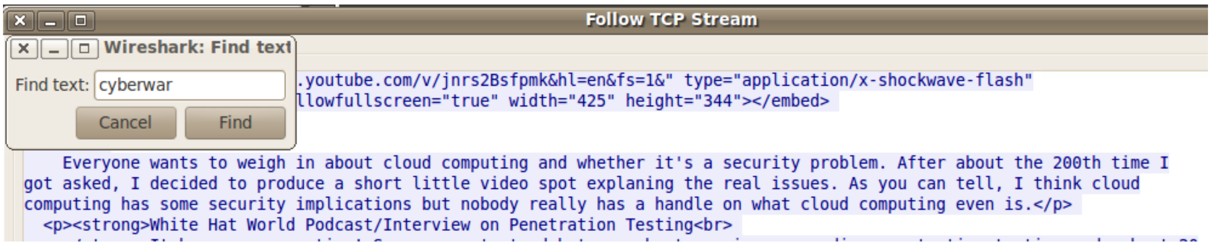
(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Expression... Clear Apply

Filter: ip.addr==192.168.200.7 && http

No.	Time	Source	Destination	Protocol	Info
4	0.146593	192.168.200.7	216.97.225.95	HTTP	GET / HTTP/1.1
6	0.292834	216.97.225.95	192.168.200.7	HTTP	HTTP/1.1 200 OK (text/html)
8	0.675336	192.168.200.7	216.97.225.95	HTTP	GET /menus/leftmenu-unexpand
14	0.821666	216.97.225.95	192.168.200.7	HTTP	HTTP/1.1 200 OK (text/html)
16	0.822283	192.168.200.7	216.97.225.95	HTTP	GET /main-text.html HTTP/1.1

Scroll down, or search on the webpage for the section on 'cyberwar'. This is an example of a policy breach.

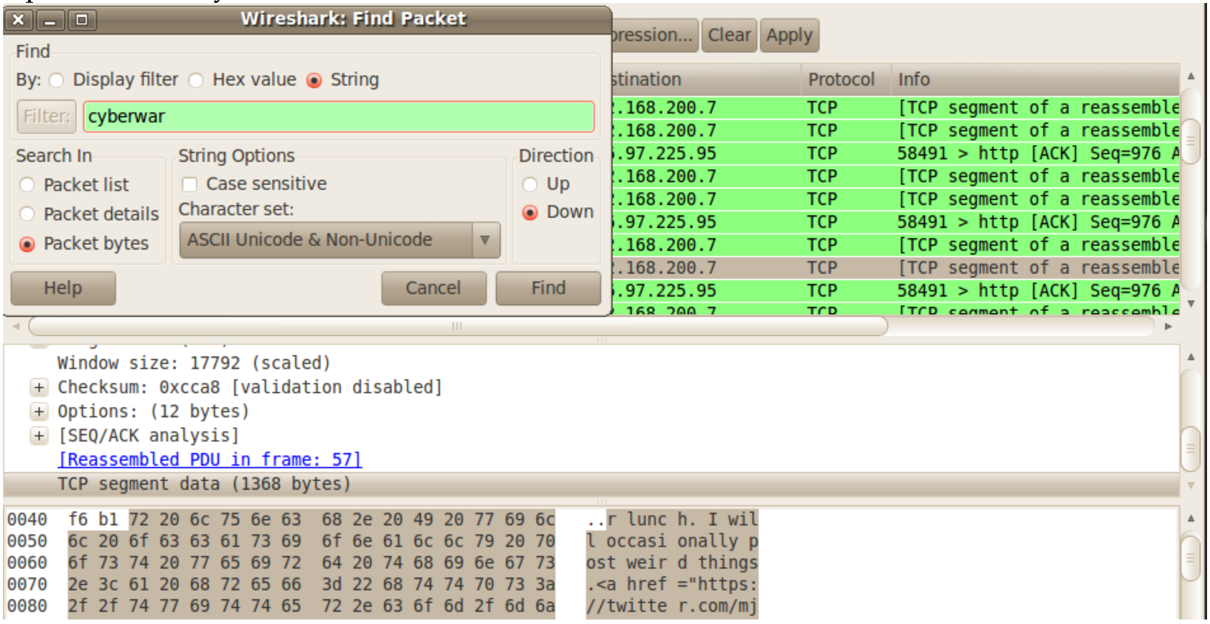
In Wireshark, follow the http stream containing the website content, and search for the string 'cyberwar'



Find text: cyberwar

```
.youtube.com/v/jnrs2Bsfpmk&hl=en&fs=1&" type="application/x-shockwave-flash"
llowfullscreen="true" width="425" height="344"></embed>

    Everyone wants to weigh in about cloud computing and whether it's a security problem. After about the 200th time I
got asked, I decided to produce a short little video spot explaning the real issues. As you can tell, I think cloud
computing has some security implications but nobody really has a handle on what cloud computing even is.</p>
  <p><strong>White Hat World Podcast/Interview on Penetration Testing<br>
```

Use the packet content search tool under Edit>Find Packet. Select the string search and find a packet with 'cyberwar' in.



Find

By: ○ Display filter ○ Hex value ● String

Filter: cyberwar

Search In	String Options	Direction
○ Packet list	☐ Case sensitive	○ Up
○ Packet details	Character set:	● Down
● Packet bytes	ASCII Unicode & Non-Unicode	

```
stination      Protocol  Info
.168.200.7     TCP       [TCP segment of a reassemble
.168.200.7     TCP       [TCP segment of a reassemble
.97.225.95     TCP       58491 > http [ACK] Seq=976 A
.168.200.7     TCP       [TCP segment of a reassemble
.168.200.7     TCP       [TCP segment of a reassemble
.97.225.95     TCP       58491 > http [ACK] Seq=976 A
.168.200.7     TCP       [TCP segment of a reassemble
.168.200.7     TCP       [TCP segment of a reassemble
.97.225.95     TCP       58491 > http [ACK] Seq=976 A
```

```
    Window size: 17792 (scaled)
  + Checksum: 0xcca8 [validation disabled]
  + Options: (12 bytes)
  + [SEQ/ACK analysis]
    [Reassembled PDU in frame: 57]
    TCP segment data (1368 bytes)
```

```
0040  f6 b1 72 20 6c 75 6e 63  68 2e 20 49 20 77 69 6c   ..r lunc h. I wil
0050  6c 20 6f 63 63 61 73 69  6f 6e 61 6c 6c 79 20 70   l occasi onally p
0060  6f 73 74 20 77 65 69 72  64 20 74 68 69 6e 67 73   ost weir d things
0070  2e 3c 61 20 68 72 65 66  3d 22 68 74 74 70 73 3a   .<a href ="https:
0080  2f 2f 74 77 69 74 74 65  72 2e 63 6f 6d 2f 6d 6a   //twitte r.com/mj
```

Look at the web packet and review the details.

Q. What is the application protocol of the web packets?

Q. What is the port number and transport protocol used for HTTP?

2-Create Snort Detection Rule

To detect employees browsing the website with the bad content, a simple IDS detection rule can be created which will detect the text "cyberwar" in the HTTP packets being downloaded from a web server.

Create and edit a Snort IDS rules such as **detect_sigs** as previously.

Edit the file, and create a Snort Detection Rule in the file.

```
alert tcp any 80 -> any any (content: "cyberwar"; msg:"HTTP
inappropriate content detected - cyberwar"; sid:10000;)
```

3- Test the rule

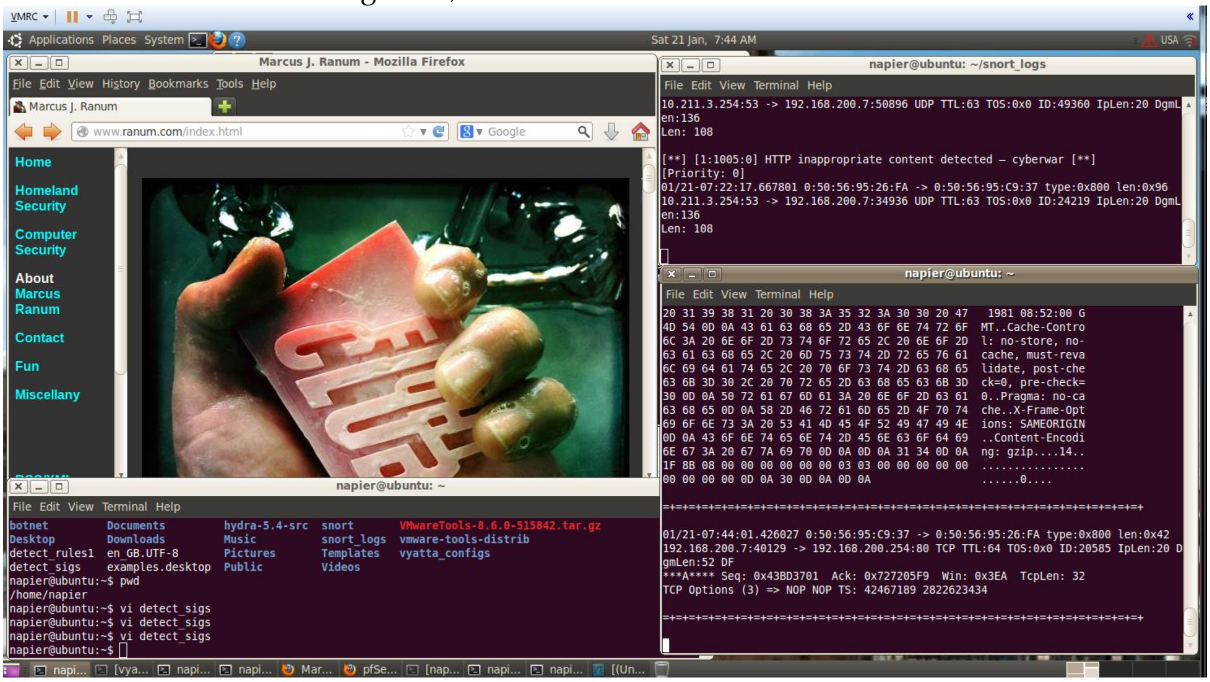
First delete the any residual **alert** file in the snort_logs folder with
To monitor the alerts being generated by the Snort IDS Sensor The output file can be checked for any lines being appended to it using the `tail` command, as shown below.

```
sudo tail -f snort_logs/alert
```

Run Snort sensor again, while monitoring the output log directory as before.

To test, run a web browser, and navigate to the **Marcus Ranum blog** page.

Resize/rearrange your windows, so you can see the monitoring of the Snort output alert file, the console window running snort, and the web browser window etc.


VMRC
Applications Places System
Sat 21 Jan, 7:44 AM
USA
Marcus J. Ranum - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Marcus J. Ranum
www.ranum.com/index.html
Google
Home
Homeland Security
Computer Security
About Marcus Ranum
Contact
Fun
Miscellany
napier@ubuntu: ~/snort_logs
File Edit View Terminal Help
napier@ubuntu: ~
File Edit View Terminal Help
napier@ubuntu: ~
File Edit View Terminal Help
napi... [vya... napi... napi... Mar... pfSe... [nap... napi... napi... [(Un...

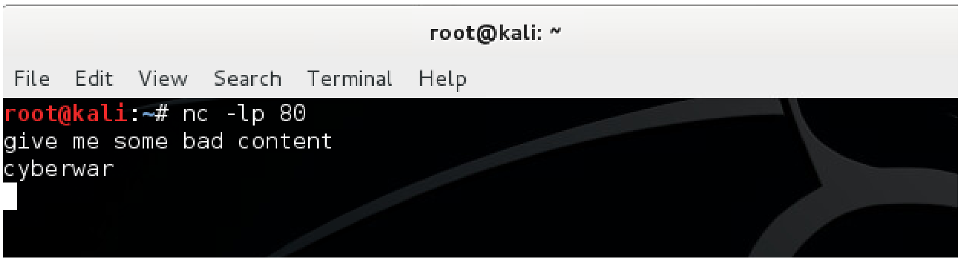
```
10.211.3.254:53 -> 192.168.200.7:50896 UDP TTL:63 TOS:0x0 ID:49360 IpLen:20 DgmL
en:136
Len: 108

[**] [1:1005:0] HTTP inappropriate content detected - cyberwar [**]
[Priority: 0]
01/21-07:22:17.667801 0:50:56:95:26:FA -> 0:50:56:95:C9:37 type:0x800 len:0x96
10.211.3.254:53 -> 192.168.200.7:34936 UDP TTL:63 TOS:0x0 ID:24219 IpLen:20 DgmL
en:136
Len: 108
```

```
20 31 39 38 31 20 30 38 3A 35 32 3A 30 30 20 47   1981 08:52:00 G
4D 54 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F  MT..Cache-Contro
6C 3A 20 6E 6F 2D 73 74 6F 72 65 2C 20 6E 6F 2D  l: no-store, no-
63 61 63 68 65 2C 20 6D 75 73 74 2D 72 65 76 61  cache, must-reva
6C 69 64 61 74 65 2C 20 70 6F 73 74 2D 63 68 65  lidate, post-che
63 6B 3D 30 2C 20 70 72 65 2D 63 68 65 63 6B 3D  ck=0, pre-check=
30 0D 0A 50 72 61 67 6D 61 3A 20 6E 6F 2D 63 61  0..Pragma: no-ca
63 68 65 0D 0A 58 2D 46 72 61 6D 65 2D 4F 70 74  che..X-Frame-Opt
69 6F 6E 73 3A 20 53 41 4D 45 4F 52 49 47 49 4E  ions: SAMEORIGIN
0D 0A 43 6F 6E 74 65 6E 74 2D 45 6E 63 6F 64 69  ..Content-Encodi
6E 67 3A 20 67 7A 69 70 0D 0A 0D 0A 31 34 0D 0A  ng: gzip....14..
1F 8B 08 00 00 00 00 00 00 03 03 00 00 00 00 00  ................
00 00 00 00 0D 0A 30 0D 0A 0D 0A                 ......0....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/21-07:44:01.426027 0:50:56:95:C9:37 -> 0:50:56:95:26:FA type:0x800 len:0x42
192.168.200.7:40129 -> 192.168.200.254:80 TCP TTL:64 TOS:0x0 ID:20585 IpLen:20 D
gmLen:52 DF
***A**** Seq: 0x43BD3701  Ack: 0x727205F9  Win: 0x3EA  TcpLen: 32
TCP Options (3) => NOP NOP TS: 42467189 2822623434

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

```
botnet         Documents         hydra-5.4-src  snort       VMwareTools-8.6.0-515842.tar.gz
Desktop        Downloads         Music          snort_logs  vmware-tools-distrib
detect_rules1  en_GB.UTF-8       Pictures       Templates   vyatta_configs
detect_sigs    examples.desktop  Public         Videos
napier@ubuntu:~$ pwd
/home/napier
napier@ubuntu:~$ vi detect_sigs
napier@ubuntu:~$ vi detect_sigs
napier@ubuntu:~$ vi detect_sigs
napier@ubuntu:~$ 
```

Create Synthetic Test Traffic

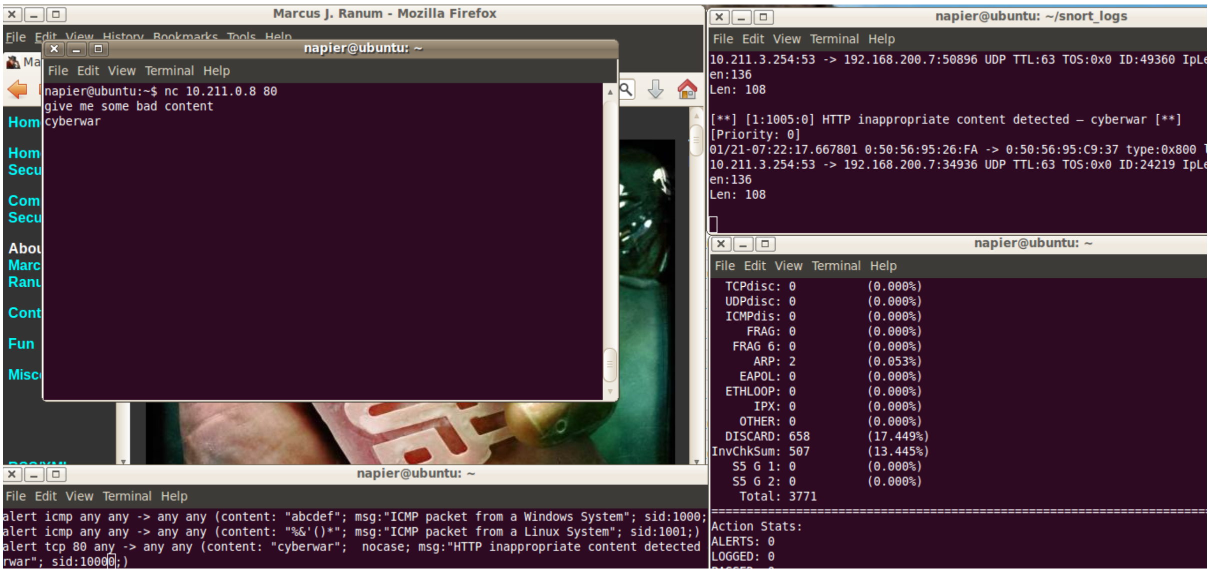
If you need a quick way of testing rules, telnet, netcat and hping are good tools for quickly creating appropriate client/server test traffic.
On Kali mimic a web server:



```
root@kali:~# nc -lp 80
give me some bad content
cyberwar
```

On Linux box, mimic the web browser requesting content,

```
nc KALI_IP_ADDR 80
```



```
napier@ubuntu:~$ nc 10.211.0.8 80
give me some bad content
cyberwar
```

```
10.211.3.254:53 -> 192.168.200.7:50896 UDP TTL:63 TOS:0x0 ID:49360 IpLe
en:136
Len: 108

[**] [1:1005:0] HTTP inappropriate content detected – cyberwar [**]
[Priority: 0]
01/21-07:22:17.667801 0:50:56:95:26:FA -> 0:50:56:95:C9:37 type:0x800 
10.211.3.254:53 -> 192.168.200.7:34936 UDP TTL:63 TOS:0x0 ID:24219 IpLe
en:136
Len: 108
```

```
  TCPdisc: 0          (0.000%)
  UDPdisc: 0          (0.000%)
  ICMPdis: 0          (0.000%)
     FRAG: 0          (0.000%)
   FRAG 6: 0          (0.000%)
      ARP: 2          (0.053%)
    EAPOL: 0          (0.000%)
  ETHLOOP: 0          (0.000%)
      IPX: 0          (0.000%)
    OTHER: 0          (0.000%)
  DISCARD: 658        (17.449%)
InvChkSum: 507        (13.445%)
   S5 G 1: 0          (0.000%)
   S5 G 2: 0          (0.000%)
    Total: 3771
===================================================================
Action Stats:
ALERTS: 0
LOGGED: 0
```

```
alert icmp any any -> any any (content: "abcdef"; msg:"ICMP packet from a Windows System"; sid:1000;
alert icmp any any -> any any (content: "%&'()*"; msg:"ICMP packet from a Linux System"; sid:1001;)
alert tcp 80 any -> any any (content: "cyberwar";  nocase; msg:"HTTP inappropriate content detected
rwar"; sid:10000;)
```

Marcus J. Ranum - Mozilla Firefox
napier@ubuntu: ~
napier@ubuntu: ~/snort_logs

Q. What is the working Snort rule?

Try Creating the Following Snort IDS Detection Rules

As well as developing your own detection rules, peer reviewed Snort detection rules can be downloaded and used from the Internet, such as from the Snort web site. These can then be tailored quickly to specific needs.

Note: Each snort rule must have a unique **SID.**

Telnet Detection Rule

Create a simple rule to generate alerts for attempted connections out to the Telenet from anywhere, and a rule for connections to the SSH service from all machines except the Ubuntu machine. Use wireshark to analyse the traffic from Ubuntu to the SSH and Telnet servers on Windows VM. Protocol and Port should be enough to create rules. The alert should be "Intrusion – Telnet Traffic Detected " "Intrusion – SSH Traffic Detected ".

Q. What is the working Snort rule?

Outbound Policy Breach - Job Search Detection Rule

Create another Snort detection rule to detect employees doing internet searches for the word “jobs” or similar. The alert should be “Policy Breach - Job searching found in Web traffic”.

Test Detection Rules

Test the rule, by using a search engine to search for the string “Security jobs”, or again create synthetic test traffic using a tool such as netcat.

Q. What is the working Snort rule?

FTP CWD to ~root Dir Detection Rule

Create a rule to detect the FTP command “CWD ~root”, which is could be an indication of an intrusion:

```
alert tcp any any -> any 21 (msg:”FTP cwd ~root attempt”; content:”CWD”;
nocase; content:”~root”; sid:10005;)
```

Run Snort with your new rule, and monitor alert output.

From Kali, connect to the Windows FTP server with netcat:

```
nc 192.168.x.7 21
220 Microsoft FTP Service
USER Administrator
331 Password required for Administrator
PASS napier
230 User Administrator logged in.
CWD ~root
```

Q. What is the working Snort rule?

Reverse Shell Detection Rule

Create a simple rule to generate alerts for connections out to port 4444 (often used by attackers for reverse shells).

From Kali, use netcat as a server, to start a listener service on port 4444:

```
nc -lp 4444
```

Check the listener is running using netcat

From Windows/Ubuntu, after running Snort with your new rule, connect to Kali using netcat as a client:

```
nc KALI_IP_ADDRESS 4444
```

Q. What is the working Snort rule?

Identifying Running Services Locally

Within a network infrastructure we have services which run on hosts. These services provide a given functionality, such as for sending/receiving email, file storage, and so on.

From → To	Command	Observation
DMZ	On your Windows host, run the command: `netstat -a` and outline some of the services which are running on your host (define the port number and the name of the service and only pick off the LISTENING status on the port).	Outline some of the services which are running on your host (define the port number and the name of the service):
LAN	For the Ubuntu Virtual Machine, and run the command: `netstat -l.`	Outline some of the services which are running on your host (define the port number and the name of the service):
DMZ	Next we will determine if these services are working. There should be a Web server working on each of the virtual machines (Ubuntu and Windows 2003), so from the Windows host and using a Web browser, access the home page: `http://192.168.x.7`	Is the service working: [Yes] [No]
LAN	From Ubuntu, access the Web server at:	Is the service working: [Yes] [No]

	`http://192.168.y.7`	
LAN	Next we will determine if these services are working using a command line. From your UBUNTU host, undertake the following: `telnet 192.168.y.7 80` then enter: `GET /`	Outline the message that is returned:
DMZ	Repeat the previous example from the WINDOWS host: `telnet 192.168.x.7 80`	

DMZ	There should be an FTP server working on Ubuntu and Windows 2003. From WINDOWS, access the FTP server on the UBUNTU server:
	`telnet 192.168.x.7 21`
	then enter:
	`USER napier`
	`PASS napier123`
	`QUIT`
		Outline the messages that you received:
		What happens to each of these when you try with an incorrect username and password:

LAN	From UBUNTU access the WINDOWS host with
	`telnet 192.168.x.7 21`
	then enter:
	`USER Administrator`
	`PASS napier`
	`QUIT`
		Outline the messages that you received:
		What happens to each of these when you try with an incorrect username and password:

DMZ	On the UBUNTU instance you will see that the **VNC** service is running, which is the remote access service. From your WINDOWS host, access the VNC service using a VNC client, and see what happens.	What does this service do:
DMZ	Next we will assess the SMTP service running on the WINDOWS virtual machine. From your UBUNTU machine console run a service to access SMTP: `telnet 192.168.y.7 25` Table 1 outlines the commands to use.	On the WINDOWS virtual machine, go into the C:\inetpub\mailroot\queue folder, and view the queued email message. Was the mail successfully queued? If not, which mail folder has the file in? Outline the format of the EML file?

Table 1: SMTP commands

```
220 napier Microsoft ESMTP MAIL Service, Version: 6.0.3790.3959 ready at  Sun, 2 Dec 2009 21:56:01 +0000
help
214-This server supports the following commands:
214 HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HELP AUTH TURN ETRN BDAT VRFY
helo me
250 napier Hello [192.168.75.1]
mail from: email@domain.com
250 2.1.0 email@domain.com....Sender OK
rcpt to: fred@mydomain.com
250 2.1.5 fred@mydomain.com
Data
354 Start mail input; end with <CRLF>.<CRLF>
From: Bob <bob@test.org>
To: Alice <alice@test.org >
Date: Sun, 20 Dec 2013
Subject: Test message
Hello Alice.
This is an email to say hello
.
```

```
250 2.6.0 <NAPIERMp7lzvxrMVHFb00000001@napier> Queued mail for delivery
```

Enumeration – Host scan

Use **nmap** to perform various recon network scans:

From → To	Command	Observation
KALI to WAN	`sudo nmap -sn -n 10.200.0.0/24`	Which hosts are on-line:
KALI to DMZ	`sudo nmap -sn -n 192.168.y.0/24`	Which hosts are on-line:
DMZ to LAN	`nmap -sP -n 192.168.x.0/24`	Which hosts are on-line:
LAN to DMZ	Run Wireshark on host in LAN, and run: `sudo nmap -sP -n 192.168.y.0/24`	Which transport layer protocol does NMAP use to discover the host: [ICMP] or [ARP]
LAN to LAN	Run Wireshark on host in LAN, and run: `sudo nmap -sP -n 192.168.x.0/24`	Which transport layer protocol does NMAP use to discover the host: [ICMP] or [ARP]

Recon Network Scanning Detection Rule

Research the nmap processor which can be used to detect network scanning, and add to the beginning of your snort rule file.
Test using nmap.

Q. What is the working Snort preprocessor?

Part 2: NAT and 1:1 mappings

No other group can access any of your hosts, as you are behind NAT. Now we need to setup a 1:1 mapping and a virtual IP address (with Proxy ARP) to map an internal address to an external one. First we need to find an IP address from the 10.200.0.0/24 network **which is not being used**, and then we will use this to allow other group's access to the hosts in the DMZ (Figure 3).

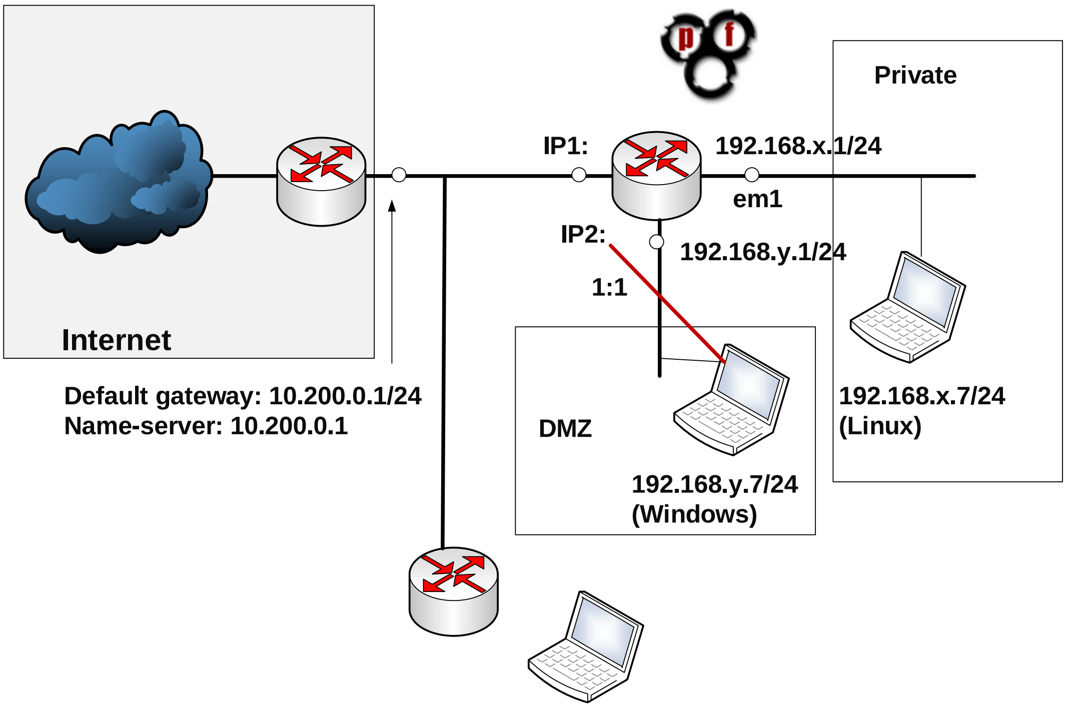


Figure 3: Setup 1:1 NAT for mapping of servers

Find an Public Network IP Address we can map our DMZ server to - Run NMAP from the Private network with:

Run NMAP from the Private network with: **nmap –sP 10.200.0.0/24**

Which hosts are on-line?

Pick and address which is not being used, and tell others in the lab which address you are picking:

Now, on the firewall, setup a 1:1 mapping of the External IP address that you have selected and the Internal IP address on the DMZ (Figure 4).

Next, setup a Virtual IP address (with Proxy ARP) for the external address you have selected, which will advertise the IP address (Figure 5).

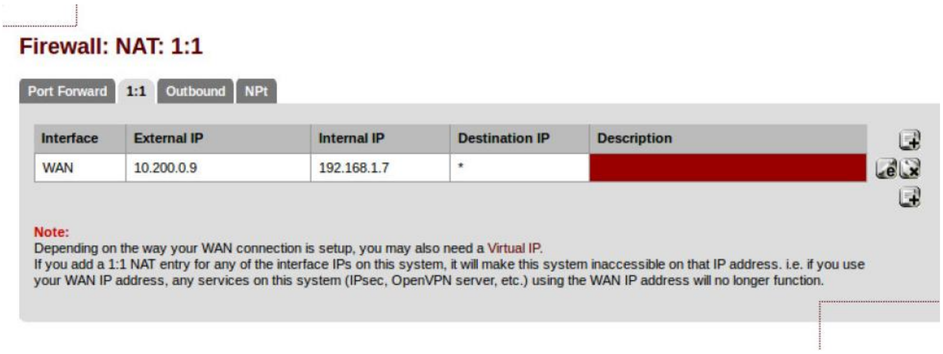
Test 1:1 NAT mapping

Now from the WAN interface, ping the host in the DMZ. Can you ping it?

Finally ask, someone in another group to ping your host in the DMZ. Can they ping it?

Now get them to access the Web server on your host.

Finally get them to NMAP your host? What can you observe from the NMAP?

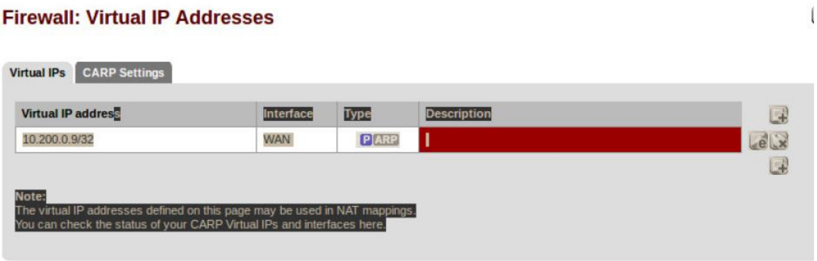


Firewall: NAT: 1:1

Interface	External IP	Internal IP	Destination IP	Description
WAN	10.200.0.9	192.168.1.7	*	[redacted]

Note:
Depending on the way your WAN connection is setup, you may also need a Virtual IP.
If you add a 1:1 NAT entry for any of the interface IPs on this system, it will make this system inaccessible on that IP address. i.e. if you use your WAN IP address, any services on this system (IPsec, OpenVPN server, etc.) using the WAN IP address will no longer function.

Figure 4: 1:1 NAT settings



Firewall: Virtual IP Addresses

Virtual IP address	Interface	Type	Description
10.200.0.9/32	WAN	P ARP	[redacted]

Note:
The virtual IP addresses defined on this page may be used in NAT mappings.
You can check the status of your CARP Virtual IPs and interfaces here.

Figure 5: Virtual IP addresses

Note: this creates an implicit Firewall Rule

Connecting to another network

Now, wait for other teams to finish (or use the Test setup). You should have ready:

- A forward facing Web and FTP site ready to connect from outside your network.

NMAP their server, and then make sure you can connect to the service. Now get them to block your specific source (just one address), and recheck that you cannot connect. Finally change your IP address, and re-do the NMAP, and make sure you can connect.

Please note some of the information related to their server. What information can you determine? Can you determine the MAC address of their server?