



CA 1 – Firewall Research

Dean - B0009

Sean - B0007

Department of Informatics
School of Informatics and Engineering
Technological University XXXXXX
XXXXXXXXXX

Word Count Limit: 2000

Current Word Count: -180

25/01/2019

XXXXXXXXXXXXXXXXXX
Digital Forensics & Cyber Security
Network Security
Peter

Plagiarism Declaration

TECHNOLOGICAL UNIVERSITY OF xxxxxxxx
DEPARTMENT OF INFORMATICS

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
LECTURER: PETER xxxxxxxx

DECLARATION ON PLAGIARISM

I declare that the work I/We am(are) submitting for assessment by the Institute examiner(s) is entirely my(our) own work, except where the author or source has been duly referenced and attributed.

I confirm that this material has not been previously submitted for a degree or any other qualification at xxxxxx or any other institution. I further confirm that I have read and understood the Institute policy on plagiarism in assignments and examinations (3AS08.doc) and that I am not, so far as I am aware, in breach of any of these regulations.

Signed: **Dean & Sean**

Date: 15/02/2019

Table of Contents

Plagiarism Declaration	2
Table of Contents.....	3
Chosen Firewalls:	4
IP Tables:	4
PFsense:	4
Design and Build:	5
Researching the Firewalls:	6
Building the PFSense Firewall:	8
Building the IP-Tables Firewall:.....	12
Testing the Firewalls:	14
IP Tables:	14
PFsense:	15
The Comparison:	16
Bibliography:	17

Chosen Firewalls:

IP Tables:



IP Tables is an extremely common firewall that is used as it is lightweight and can be used on both core and GUI based Linux operating systems. It has no GUI and is entirely command-line based.

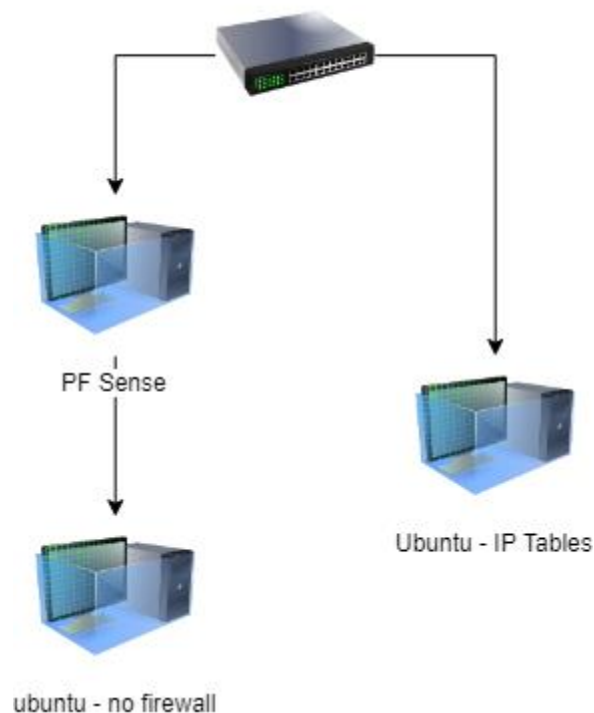
PFsense:



PFsense is the most common Firewall, I have seen while working in the security Industry mainly for it being open sourcing meaning there are hundreds of projects available on Github to increase the power of PFsense on the Network, one such tool called Kibana that uses data from PFsense to give visual information. This software is freely available at the following location
<https://www.pfsense.org/>

Design and Build:

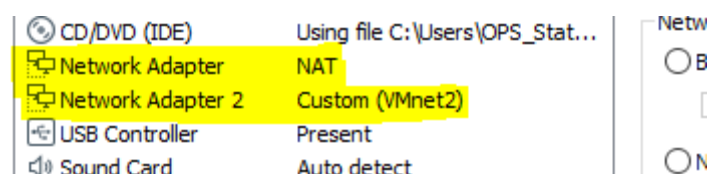
It is always important prior to building a network to design its layout that way there is a clear idea of what is being laid out.



In the above network I have a physical switch and 3 virtual machines although they are on the same machine, there are logically on different networks. In connection 1 we have a PFSense VM running as the firewall with two network adapters so the connection is in and straight out, the Ubuntu system on this network has no firewall enabled and is 100% reliant on PFSense firewall. The second connection is a straight to the Ubuntu VM which is running a IP Table Configuration. Both Ubuntu machines have IP address 192.168.1.50 and 192.168.1.60 and both firewalls will be configured to with the following rules to best test the firewalls.

The rules to set:

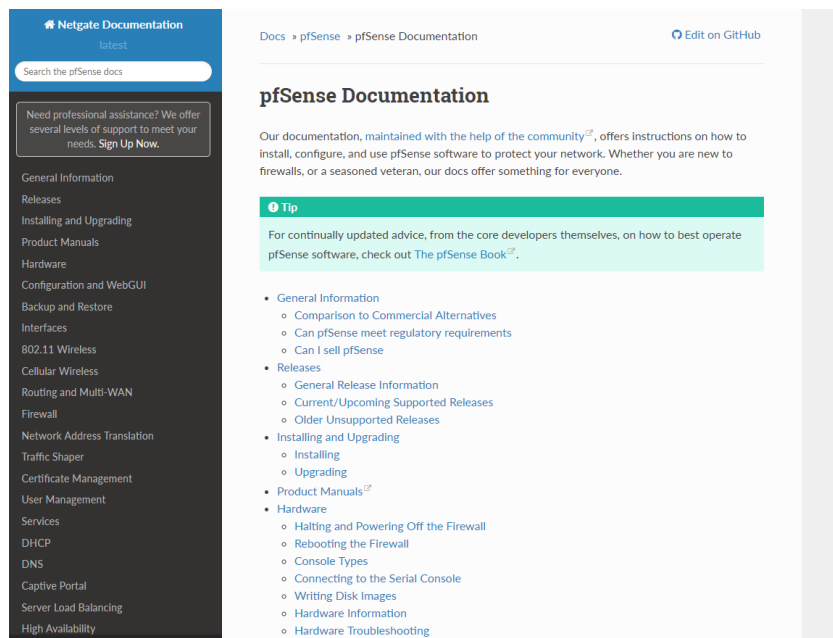
- Block all pings to the Network / Machine
- Block Facebook Webpage
- Allow Telnet and SSH to the Network / Machine



Above is a photo of the networking for the PFSense box where you can see an in connection and a out connection that connects to Ubuntu.

Researching the Firewalls:

For PFSense I knew exactly where to look as it is extremely well documented. It was easy to find documentation on Netgate that covered nearly all the areas of the software firewall. (PFSense Manual, 2019)

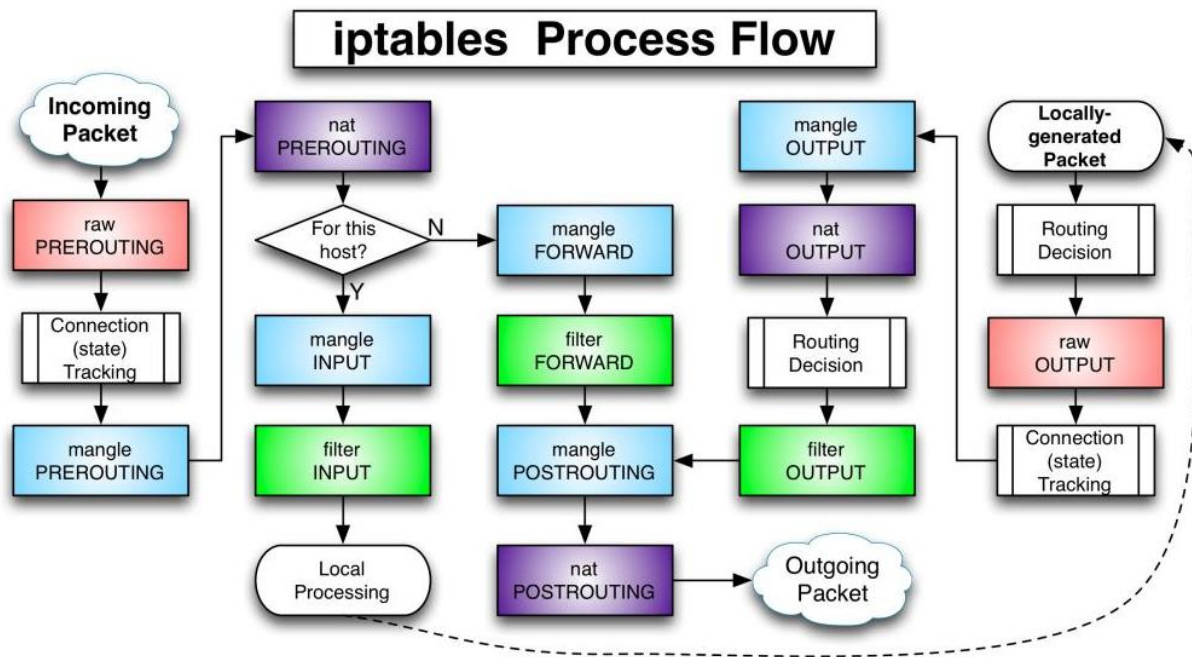


The screenshot shows the Netgate Documentation website for pfSense. On the left is a dark sidebar with a search bar and a list of navigation links including General Information, Releases, Installing and Upgrading, Product Manuals, Hardware, Configuration and WebGUI, Backup and Restore, Interfaces, 802.11 Wireless, Cellular Wireless, Routing and Multi-WAN, Firewall, Network Address Translation, Traffic Shaper, Certificate Management, User Management, Services, DHCP, DNS, Captive Portal, Server Load Balancing, and High Availability. The main content area is titled 'pfSense Documentation' and includes an introductory paragraph, a 'Tip' box about 'The pfSense Book', and a detailed table of contents with links to various sections like General Information, Releases, Installing and Upgrading, Product Manuals, and Hardware.

Since IPTables is a fundamental part of most Linux systems I felt the best was to check Linux Man Pages, and straight away found all the information I needed (Linux.die.net, 2019)



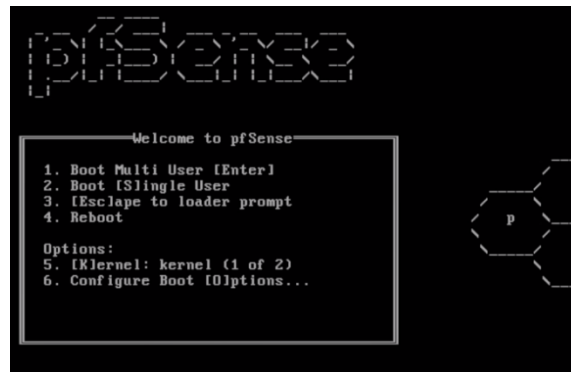
The screenshot displays the 'iptables(8) - Linux man page' from the die.net website. The page has a dark red sidebar with navigation links like 'Library', 'linux docs', 'linux man pages', and 'Toys'. The main content area is titled 'iptables(8) - Linux man page' and includes sections for Name, Synopsis, Description, and Targets. The Synopsis section lists various command-line options for iptables. The Description section explains the purpose of iptables and how chains and rules work. The Targets section describes the different actions that can be taken on a packet match. On the right side of the page, there is a green advertisement for an ebook titled 'How to Build a Security Operations Center (On a Budget)' with a 'GET YOUR FREE COPY' button.



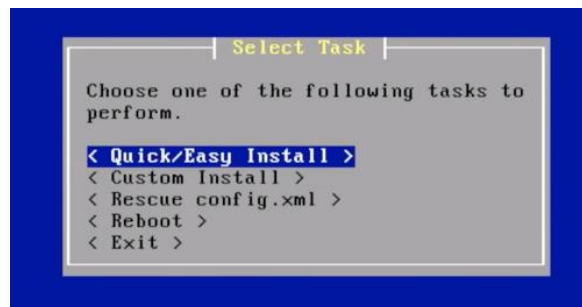
(How IPtables Work, 2019)

Building the PFsense Firewall:

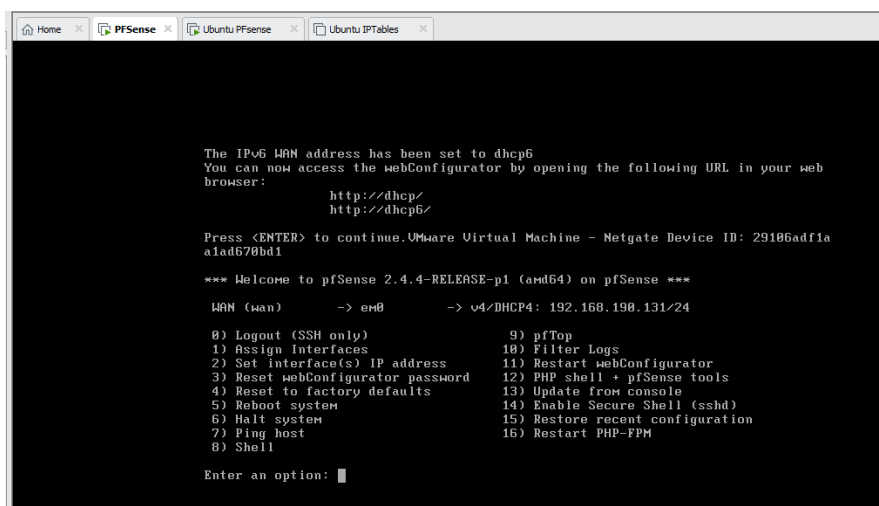
Since this whole network will be built virtually, we will build it on a proxmox virtualisation server. PFsense is available to freely download from <https://www.pfsense.org/> in ISO, that can be easily installed anywhere. - (Pfsense.org, 2019)



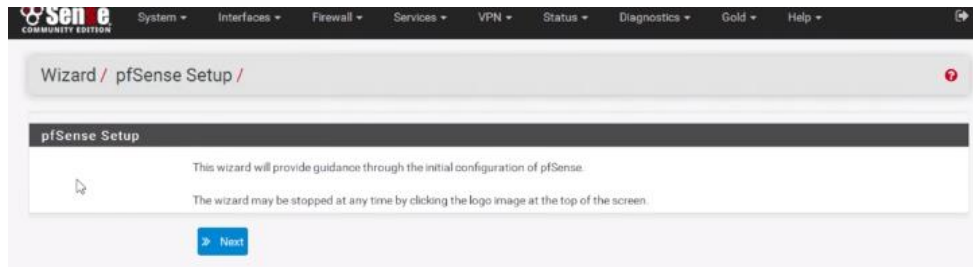
Booting PFsense for installation is straight forward; the first stage is to choose the Boot Multi User option and then just follow the on screen instructions and using the easy installer.



Upon completing the installation of PFsense you are prompted with a prompt for the IP address to connect to the web GUI and from here there are limited administration options

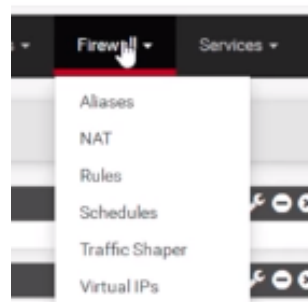


Upon accessing the Web User Interface you are promoted with the login and setup wizard,

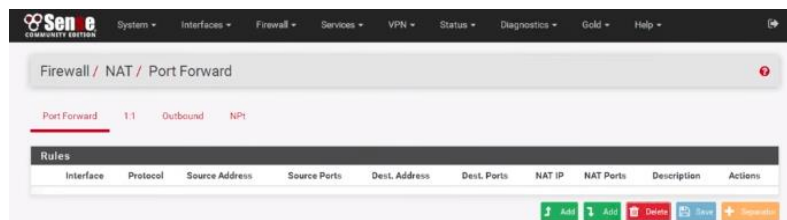


Using the wizard, it allows us to configure options such as Domain Name, DNS server, IP addressing, set administration passwords and assign local IP's to adapters. After the wizard has finished we have the option to configure other settings such as the firewall, VPN, port forwarding and more, but for this purpose I will stick with the firewall options.

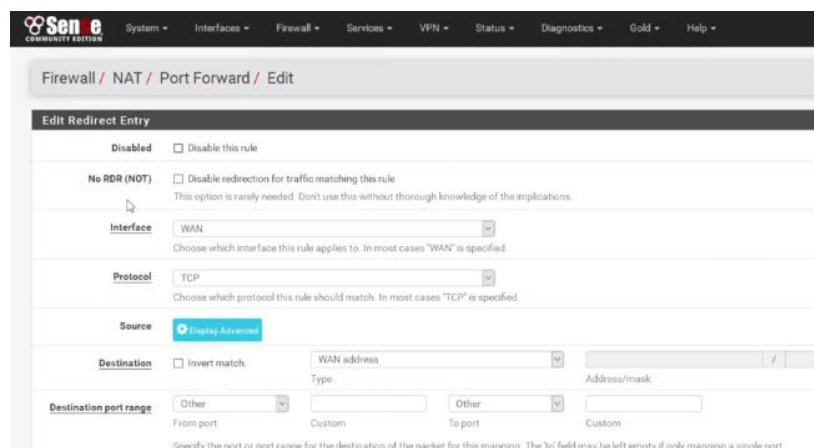
The firewall allows for some of the following controllers



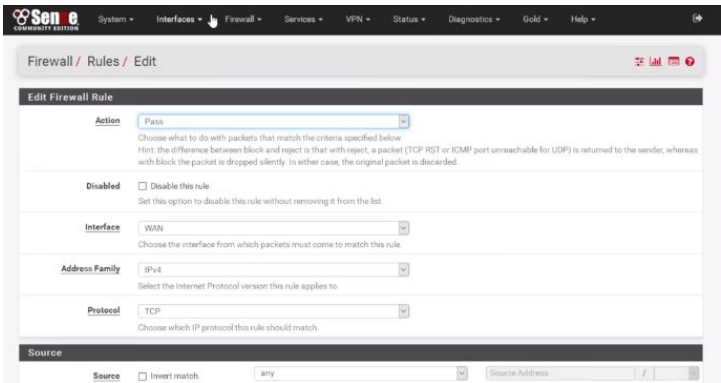
Under the NAT category it allows us to add port forwarding or redirection of traffic, I use this feature on my own home network to forward certain ports to systems in my house that contain the services, such as for security all external SSH & Telnet port attempts are redirected to a Ubuntu Honeypot I maintain on my network, the function to add these sort of rules can be seen here.



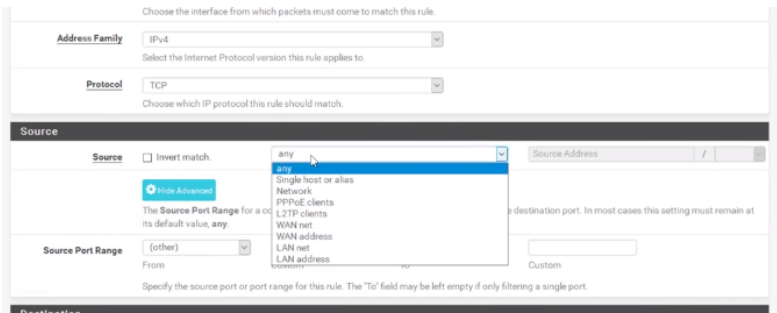
Bellow you can see some of the configuration options that are available in the NAT / Port Forward section.



When setting rules for allowing or denying traffic it is a very simplified process that allows for access control lists to be designed and laid out with easy as can be seen in the picture bellow. One of the best features of Pfsense is the community and the ability to import rules, for example if there is a new system exploit that targets under a certain port or service there is a very good chance that a member of the community has developed a rule set to mitigate this threat and has shared it online.



In the sources address information you can see the amount of configurations that are available to set rules on,



For our needs for testing purposes to block a ICMP request “Ping” we configure the following options:

Access: Block

Interface: Lan

Protocol: ICMP

Sources: Any: / 24

Saving this rule would block all network traffic from any address on the subnet 255.255.255.0, which is every device on available on the LAN network. Using the same configuration we have the ability to block or allow multiple other types such as OSPF probes and more.

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	3 / 1.55 MiB	*	*	*	WAN Address	80	*	*		Anti-Lockout Rule	
<input checked="" type="checkbox"/>	0 / 0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogus networks	
<input type="checkbox"/>	<input checked="" type="checkbox"/> 0 / 0 B	IPv4 ICMP any	*	*	*	*	*	none			

Add Add Delete Save Separator

Here you can see the rule in the table.

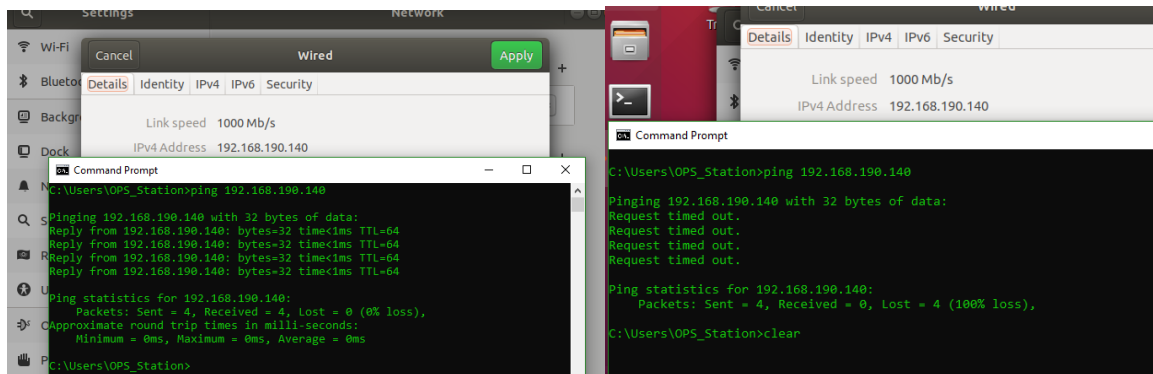
Below is a picture of allowing telnet through the PfSense firewall

The screenshot shows the configuration of a firewall rule in PfSense. The **Protocol** is set to **TCP/UDP**. Under the **Source** tab, the **Source Port Range** is configured with **From** and **To** both set to **Telnet (23)**. The **Destination** tab is also visible, with the **Destination Port Range** similarly set to **Telnet (23)**. The **Extra Options** section is currently empty.

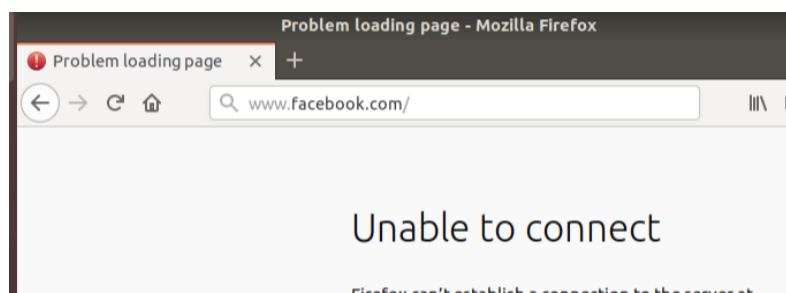
Below is a picture of blocking Facebook on the network

The screenshot shows the configuration of a firewall rule in PfSense to block Facebook. The **Address Family** is set to **IPv4+IPv6**. The **Protocol** is **TCP/UDP**. Under the **Destination** tab, the **Destination** is set to **Network** with the address **66.220.144.0** and a **mask** of **24**. The **Destination Port Range** is set to **HTTPS (443)** for both **From** and **To** ports.

Below is an image of testing a ping threw the firewall before and after the rule being set



Below is a picture demonstration the blocking of Facebook through the firewall.



Building the IP-Tables Firewall:

Ubuntu IP-Tables is a firewall that already comes pre-installed with Ubuntu OS systems. If however you do have to uninstall Iptables, it can be done by using the command **sudo apt-get install iptables**. Being that this firewall is command line base firewall, it is easy to set the rules for.

To check the version for it, we use the following command.

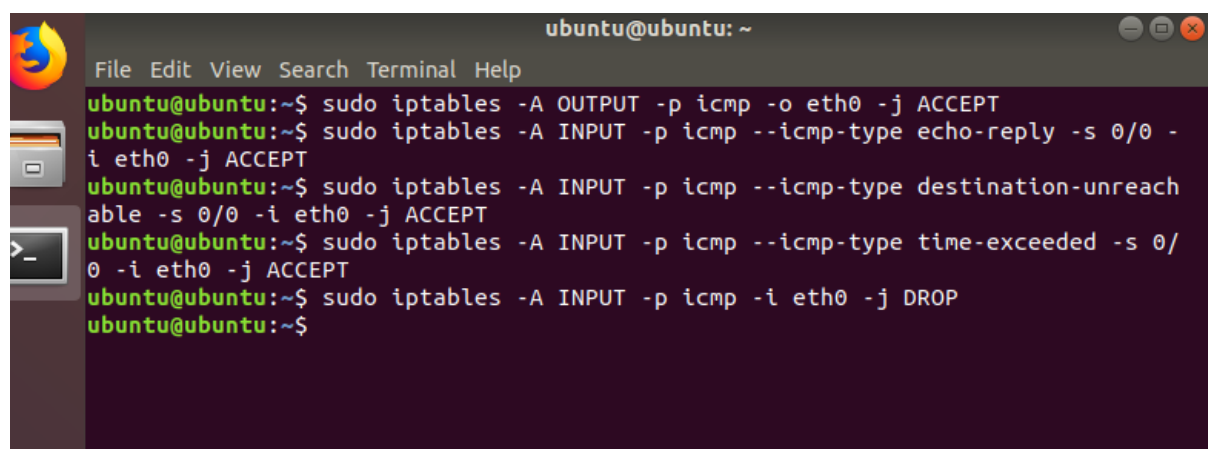
```
ubuntu@ubuntu:~$ iptables -h
iptables v1.6.1

Usage: iptables -[ACD] chain rule-specification [options]
iptables -I chain [rulenum] rule-specification [options]
iptables -R chain rulenum rule-specification [options]
iptables -D chain rulenum [options]
iptables -[LS] [chain [rulenum]] [options]
iptables -[FZ] [chain] [options]
iptables -[NX] chain
iptables -E old-chain-name new-chain-name
iptables -P chain target [options]
iptables -h (print this help information)

Commands:
Either long or short options are allowed.
--append -A chain          Append to chain
--check  -C chain          Check for the existence of a rule
--delete -D chain          Delete matching rule from chain
--delete -D chain rulenum  Delete rule rulenum (1 = first) from chain
--insert -I chain [rulenum] Insert in chain as rulenum (default 1=first)
--replace -R chain rulenum Replace rule rulenum (1 = first) in chain
--list   -L [chain [rulenum]]
```

At the current time, our Iptables currently has no rules, leaving the machine currently open.

To firstly secure our system, we will block ping on the system by implementing the following a rules.



```
ubuntu@ubuntu: ~
File Edit View Search Terminal Help
ubuntu@ubuntu:~$ sudo iptables -A OUTPUT -p icmp -o eth0 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-reply -s 0/0 -i eth0 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A INPUT -p icmp --icmp-type destination-unreachable -s 0/0 -i eth0 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A INPUT -p icmp --icmp-type time-exceeded -s 0/0 -i eth0 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A INPUT -p icmp -i eth0 -j DROP
ubuntu@ubuntu:~$
```

With the settings we have put in, we are able to receive pings through the firewall, however the reply will be dropped.

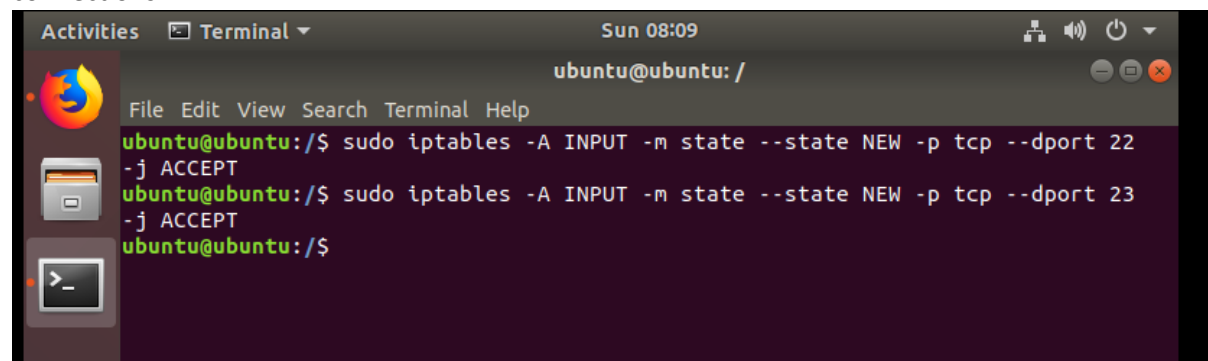
The above options in iptables will accept pings through the firewall, while dropping the reply. Blocking Facebook means knowing the IP address in which we need to block. Since Facebook only works over HTTPS we only need it disabled port 443 for the IP address.

```
ubuntu@ubuntu:~$ nslookup www.facebook.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
www.facebook.com canonical name = star-mini.c10r.facebook.com.
Name:   star-mini.c10r.facebook.com
Address: 31.13.73.35
Name:   star-mini.c10r.facebook.com
Address: 2a03:2880:f132:83:face:b00c:0:25de

ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ sudo iptables -A OUTPUT -p tcp -d 31.13.73.35 --dport 443 -j REJECT
ubuntu@ubuntu:~$
```

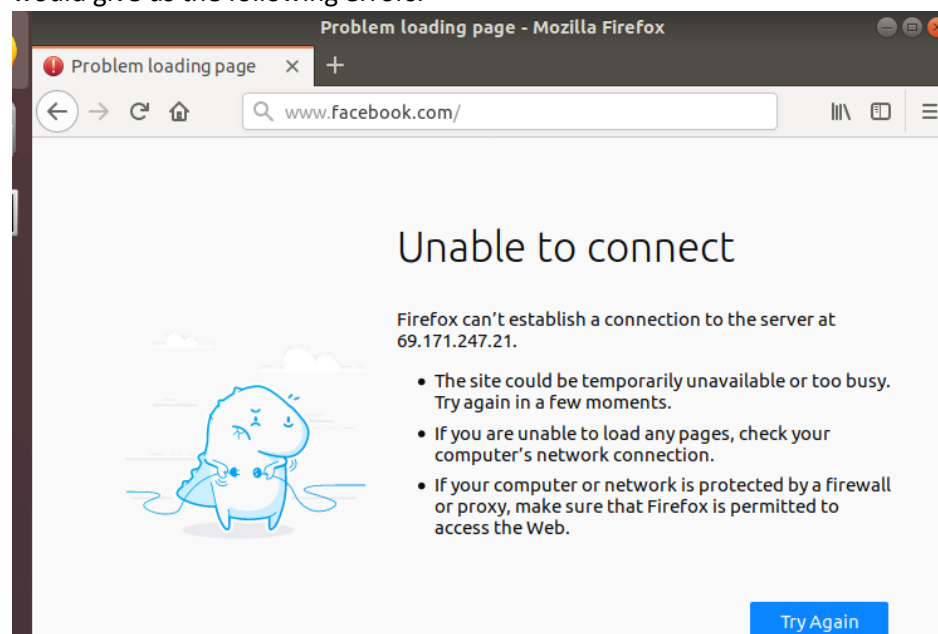
Bellow we can see the traffic being allowed through the port 22 and 23 allowing for SSH and Telnet connections



The screenshot shows a terminal window titled 'Terminal' with the date 'Sun 08:09' and the prompt 'ubuntu@ubuntu: /'. The terminal contains the following commands and output:

```
File Edit View Search Terminal Help
ubuntu@ubuntu:/$ sudo iptables -A INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
ubuntu@ubuntu:/$ sudo iptables -A INPUT -m state --state NEW -p tcp --dport 23 -j ACCEPT
ubuntu@ubuntu:/$
```

Testing these rules have been implemented we can attempt to go to Facebook's main page, which would give us the following errors.



Testing the Firewalls:

IP Tables:

To confirm the packets are being blocked, we simply need to try ping the ubuntu device, as expected the ping is not replying,

```
root@kali:~# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
```

however, the ping is still being received by the Ubuntu system.

```
root@ubuntu:/home/firewall# tcpdump -i ens38 icmp and icmp[icmptype]=icmp-echo
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens38, link-type EN10MB (Ethernet), capture size 262144 bytes
13:06:46.177000 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 552, length 64
13:06:47.198455 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 553, length 64
13:06:48.221026 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 554, length 64
13:06:49.242267 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 555, length 64
13:06:50.263462 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 556, length 64
13:06:51.285447 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 557, length 64
13:06:52.306275 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 558, length 64
13:07:06.676926 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 572, length 64
13:07:07.700137 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 573, length 64
13:07:08.724486 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 574, length 64
13:07:09.747615 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 575, length 64
13:07:10.769781 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 576, length 64
13:07:11.792562 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 577, length 64
13:07:12.814464 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 578, length 64
13:07:13.837058 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 579, length 64
13:07:14.858769 IP 192.168.1.5 > ubuntu: ICMP echo request, id 3034, seq 580, length 64
```

The above results show us that the Iptables does not block the receiving of pings but does not send a reply to the machine that is pinging it.

Secondly we'll attempt to perform an Nmap scan on the Ubuntu system, with the current rules that are in place, we are able to successfully scan the system.

```
Nmap scan report for 192.168.1.2
Host is up (0.00039s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
MAC Address: 00:0C:29:F8:A0:61 (VMware)
```

To prevent this, we'll drop all input into the machine, as to prevent a Nmap scan from taking place, this is done by entering **sudo iptables -P INPUT DROP**.

However, with this rule we absolutely block all types of data, in and out. As a result we must enter in **sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT**

This allows for established connections to take place on the device, in other words it allows connections to the internet to be made.

Secondly we must allow for loopback, as to allow us to connect our own localhost or to ping ourselves, this can be done by entering **sudo iptables -I INPUT -I lo -j ACCEPT**

```
Nmap scan report for 192.168.1.2
Host is up (0.00069s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
22/tcp    closed ssh
23/tcp    closed telnet
MAC Address: 00:0C:29:F8:A0:61 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.58 seconds
```

As a result, ports that are not already declared are shown as filtered. Due to the fact the Nmap could not pick up the open ports, it picks up the declared closed port.

PFsense:

To confirm that packets are being blocked, we simply need to ping a device behind the firewall, in this case a Ubuntu machine that has an IP of 192.168.1.60. Before implementing the rule it should be noted that I was able to ping across the LAN network to the Ubuntu machine.

```
C:\Users\OPS_Station>ping 192.168.1.60

Pinging 192.168.1.60 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.60:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

As can be seen from the results above that the firewall rule was successful and stop the packets reaching the Ubuntu VM.

The Comparison:

PFsense is a lot easier to use than IPTables, for one the firewall can be configured through a GUI interface, as a result you do not need to remember long commands to configure the rules, it also allows you to configure the rules in any order, compared to Iptables, were if you configure the rules wrong, you have to entirely reset your IPTables list by using **sudo iptables -F**.

Another Benefit of PFSense over IPTables is the feature that allows an administrator the ability to import or export rules from PFSense backup function as needed, this feature allows for administrators to be able to download rule sets from provides to do fast updates such as rules to block SMB traffic during the Petya ransomware scare last year.

Since PFSense is standalone it takes the responsibility of the network firewall configurations and security away from the user and leaves it to a configured system, whereas with individual system hosting IPTables it would be down to the user or an admin to access each system and update the rules.

Bibliography:

Pfsense.org. (2019). *Download pfSense Community Edition*. [online] Available at: <https://www.pfsense.org/download/> [Accessed 10 Feb. 2019].

PFSense Manual. (2019). <https://docs.netgate.com/pfsense/en/latest/index.html>: Netgate. [Accessed 10 Feb. 2019]

Linux.die.net. (2019). *iptables(8) - Linux man page*. [online] Available at: <https://linux.die.net/man/8/iptables> [Accessed 10 Feb. 2019].

How IPtables Work. (2019). <https://n0where.net/how-does-it-work-iptables>: n0where.