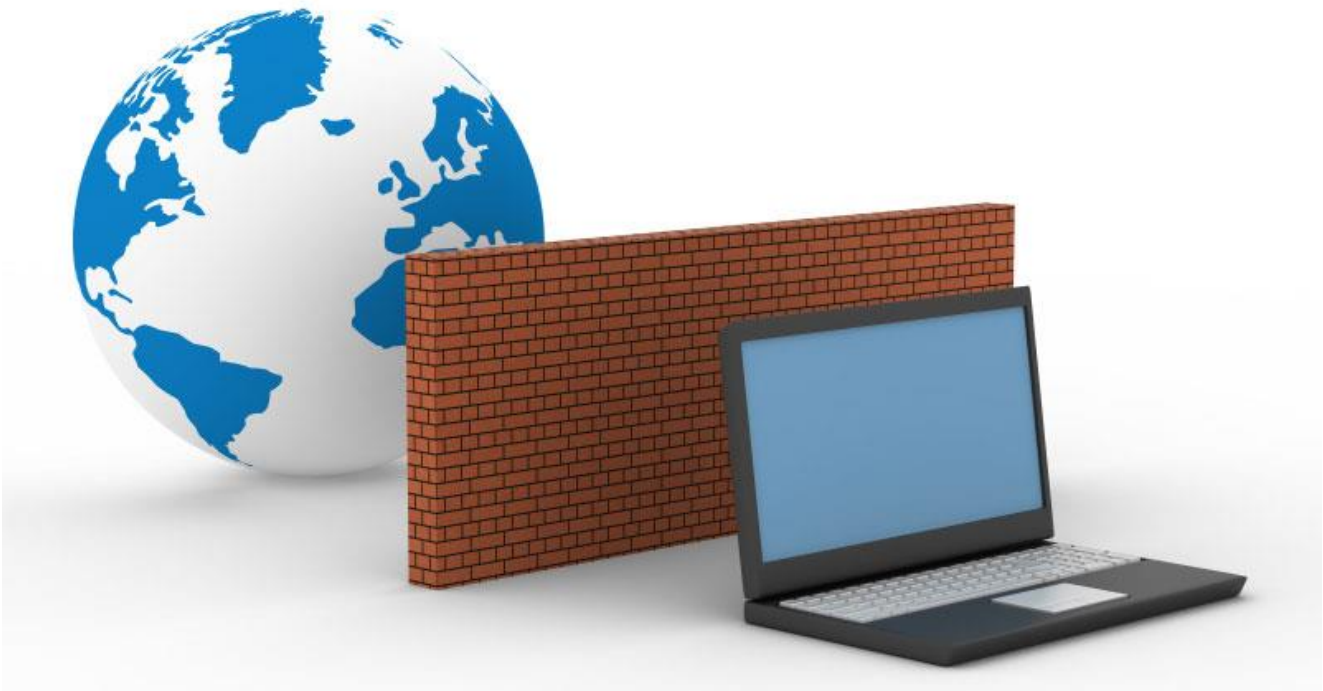
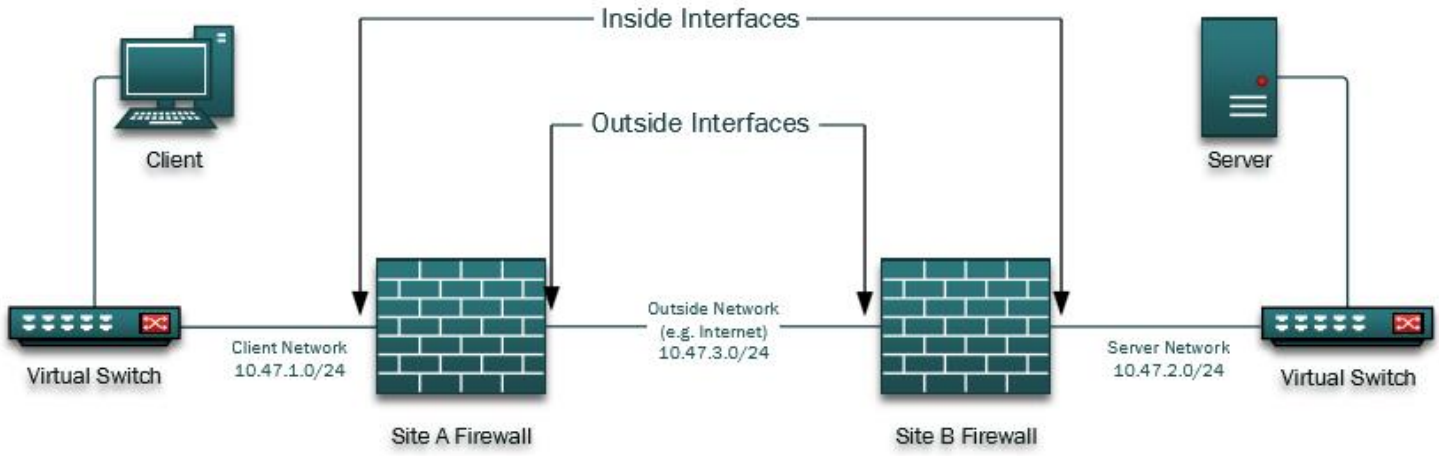
Ketaki Vitthal Kakade (kk524)
Adhithya Sivanesh (as3423)
Arman Gupta (ag986)



Network Protocol Security

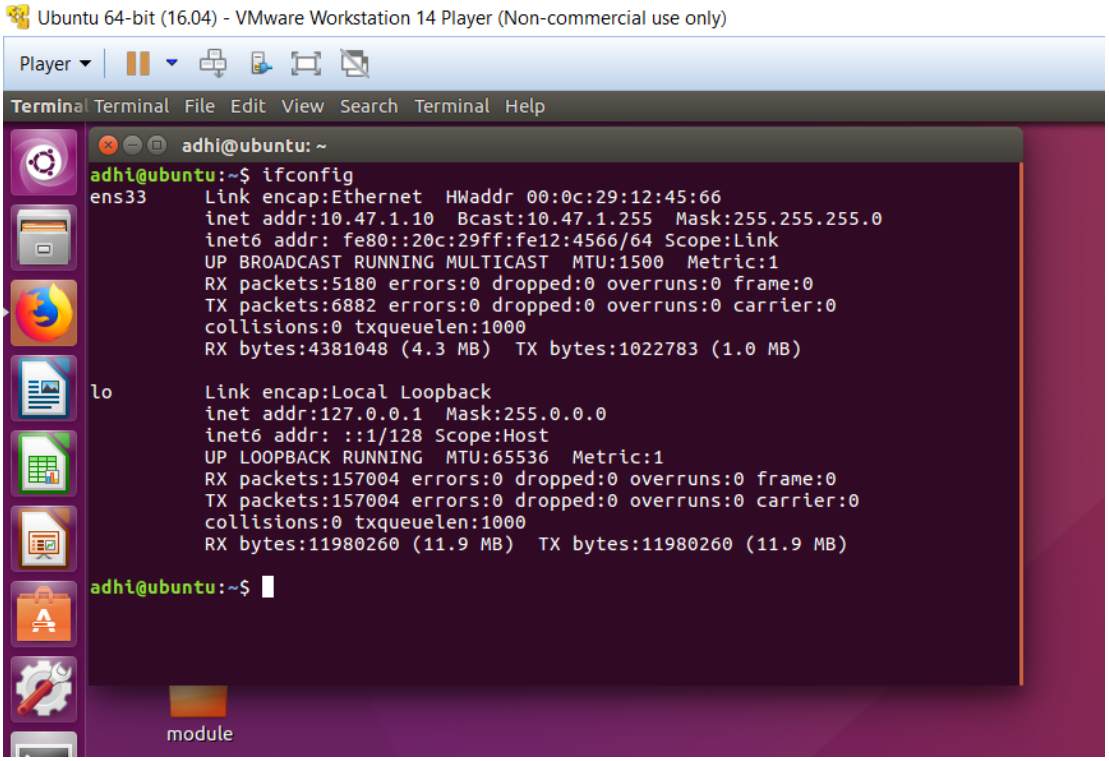
PROJECT 3



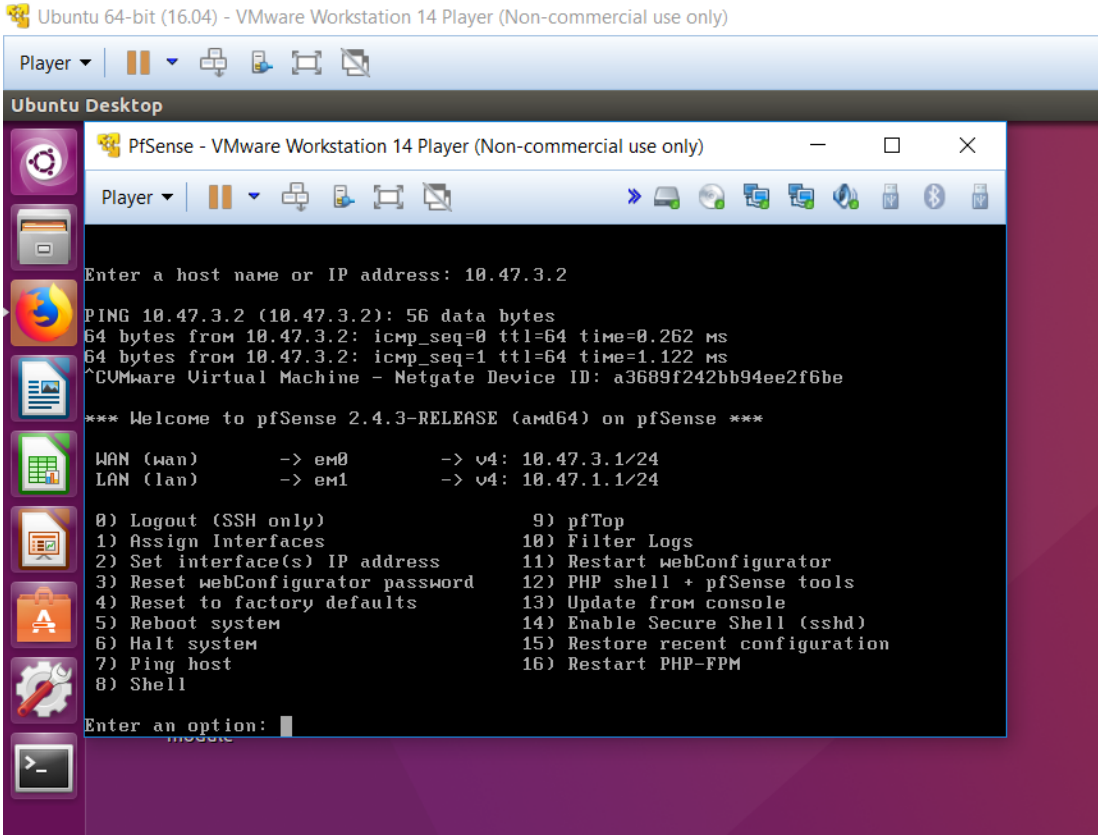
Task 1 – Basic Configuration:

To establish the basic setup according to the requirements, VMware Workstation platform was used. Two instances of Pfsense (Pfsense and Pfsense 2) were installed and two instances of Linux distribution were installed – Ubuntu (64 bit version 16.04) to work as client and Seed Ubuntu to work as server. The WAN interfaces of both the pfsense firewalls were put on the same VMnet (Vmnet0). For client side, 2 network adapters, one for Ubuntu and one network adapter for the pfsense was put into same LAN segment (internal). For the server side, one interface of the Pfsense 2 and the network adapter of the seed Ubuntu was put into the same LAN segment (internal 2). The basic configuration of the setup given in the above diagram is established by assigning the IP address provided on the corresponding interfaces (Figure 1 – site A) and (Figure 2 – site B).

Site A:
Client: 10.47.1.10/24 (on LAN segment-internal) (Fig 1.1)
Firewall LAN interface: 10.47.1.1/24 (on LAN segment-internal) (Fig 1.2)

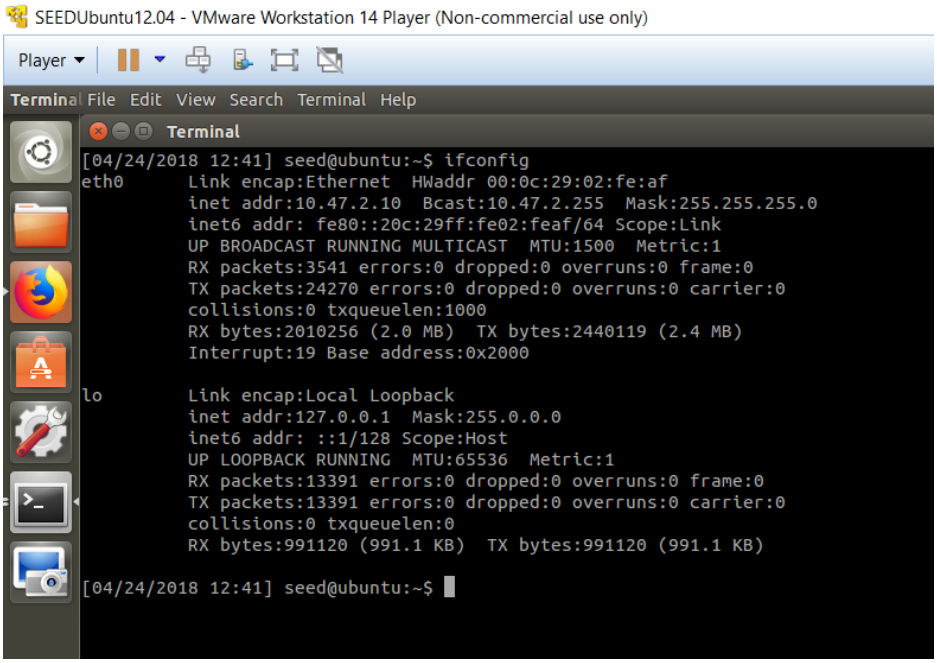


(Figure 1.1)

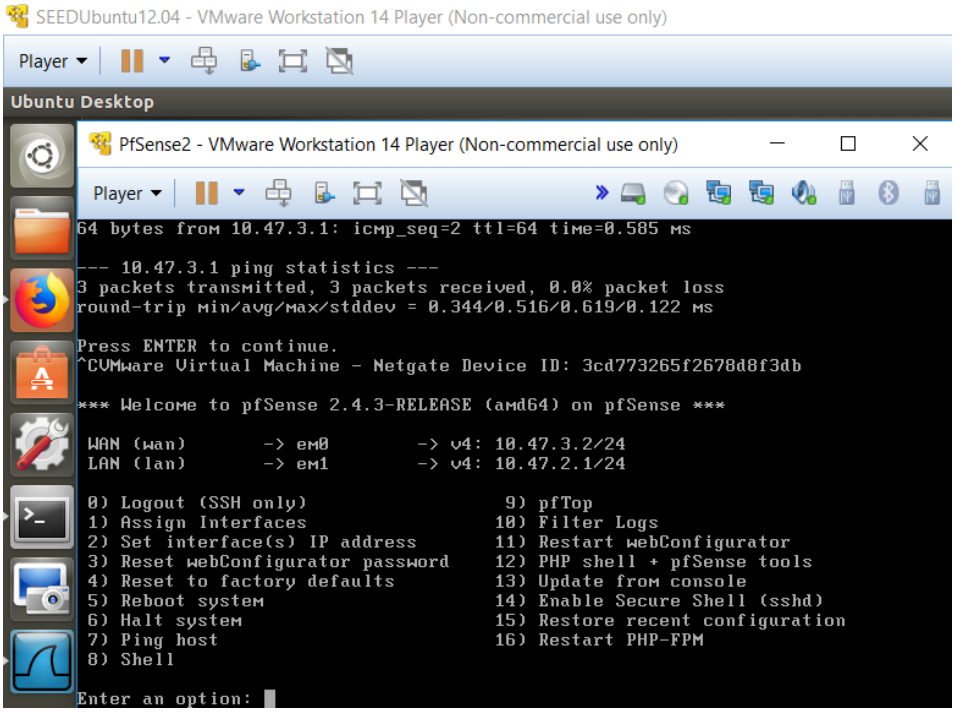


(Figure 1.2)

Site B:
Server: 10.47.2.10/24 (on LAN segment-internal 2) (Figure 2.1)
Firewall LAN interface: 10.47.2.1/24 (on LAN segment-internal 2) (Figure 2.2)



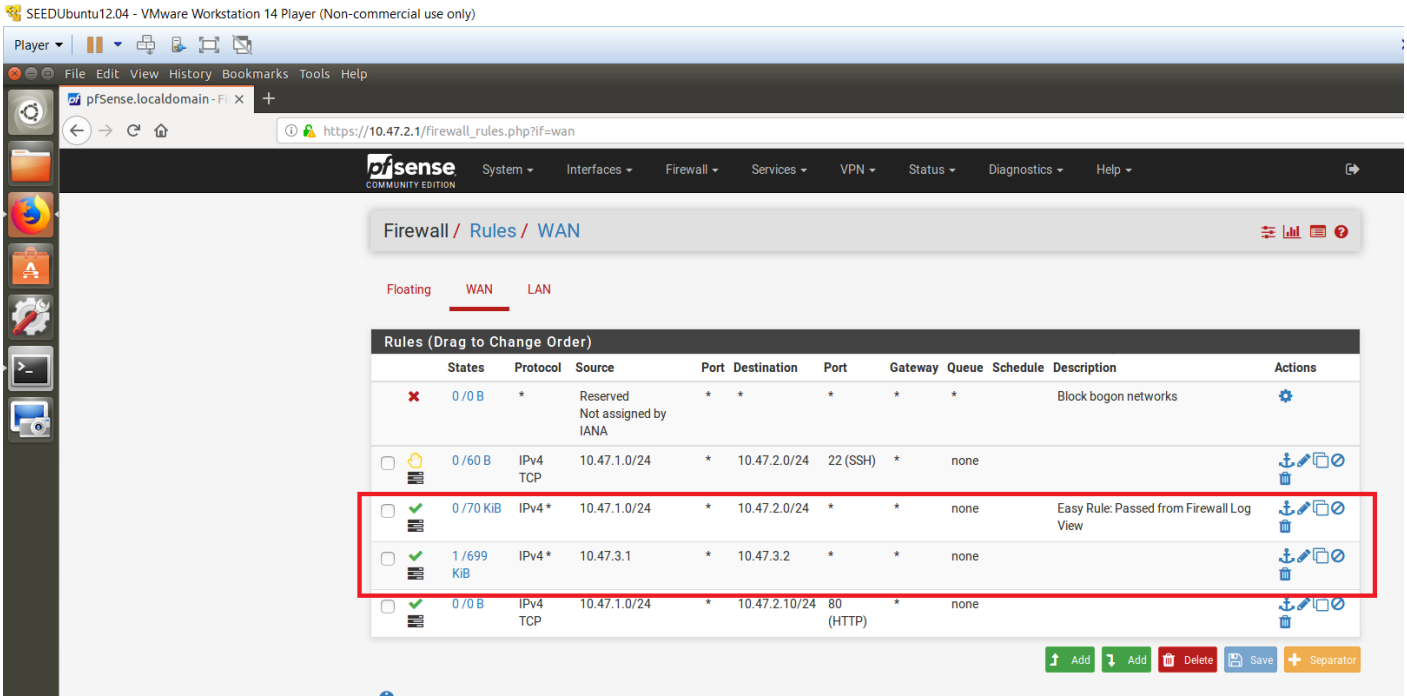
(Figure 2.1)



(Figure 2.2)

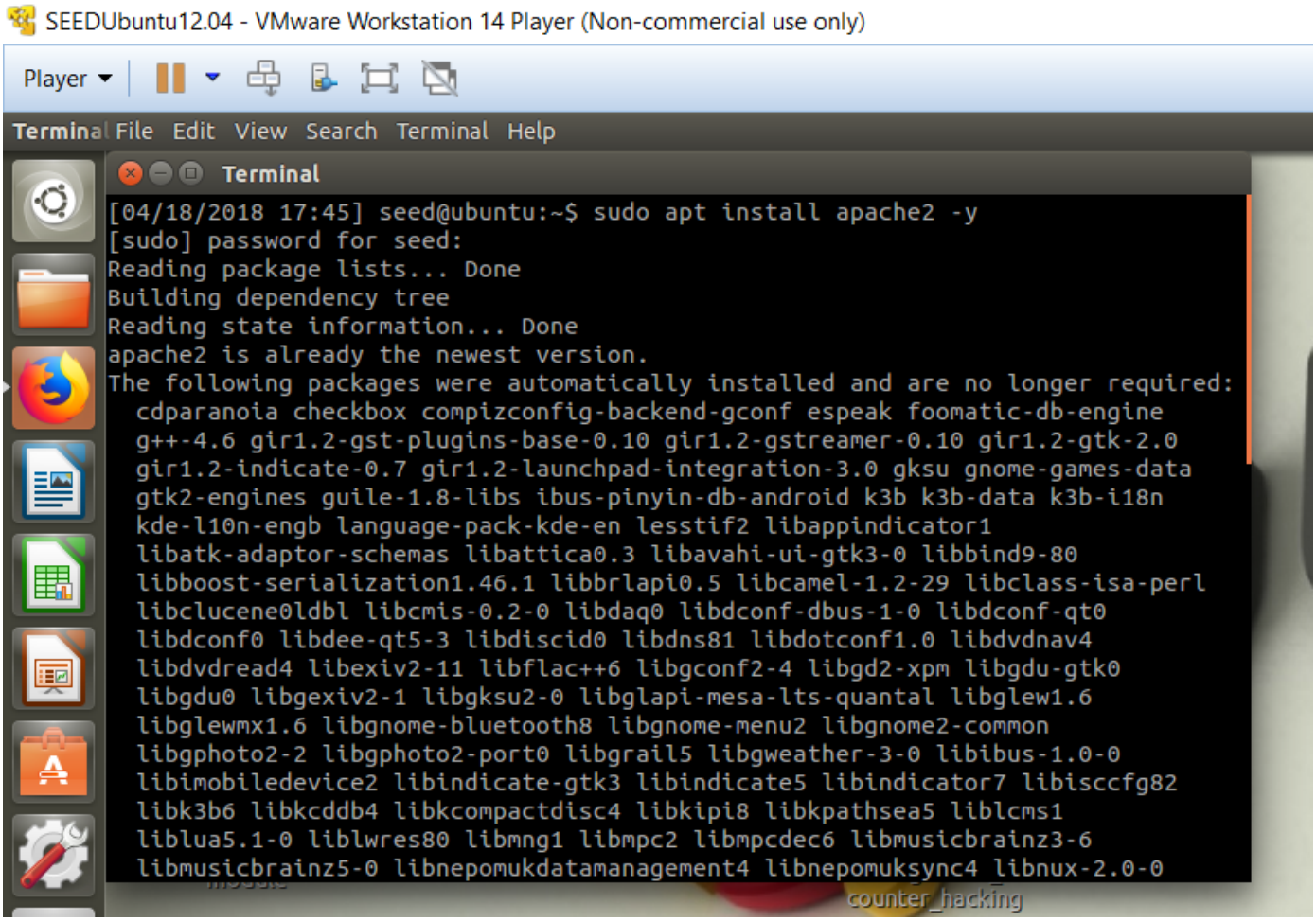
WAN connectivity:
Pfsense (client side): 10.47.3.1/29 (Vmnet0)
Pfsense 2 (server side): 10.47.3.2/29 (VMnet0)

By default, the firewall blocks all the traffic and drops the packets passing through the firewall. For the setup to work according to the requirement, we allowed the ICMP traffic flowing through both the firewalls in the access list. This allowed us to access the GUI of the Pfsense on both the end devices. The access list allows us to have end to end connectivity between the client and the server.

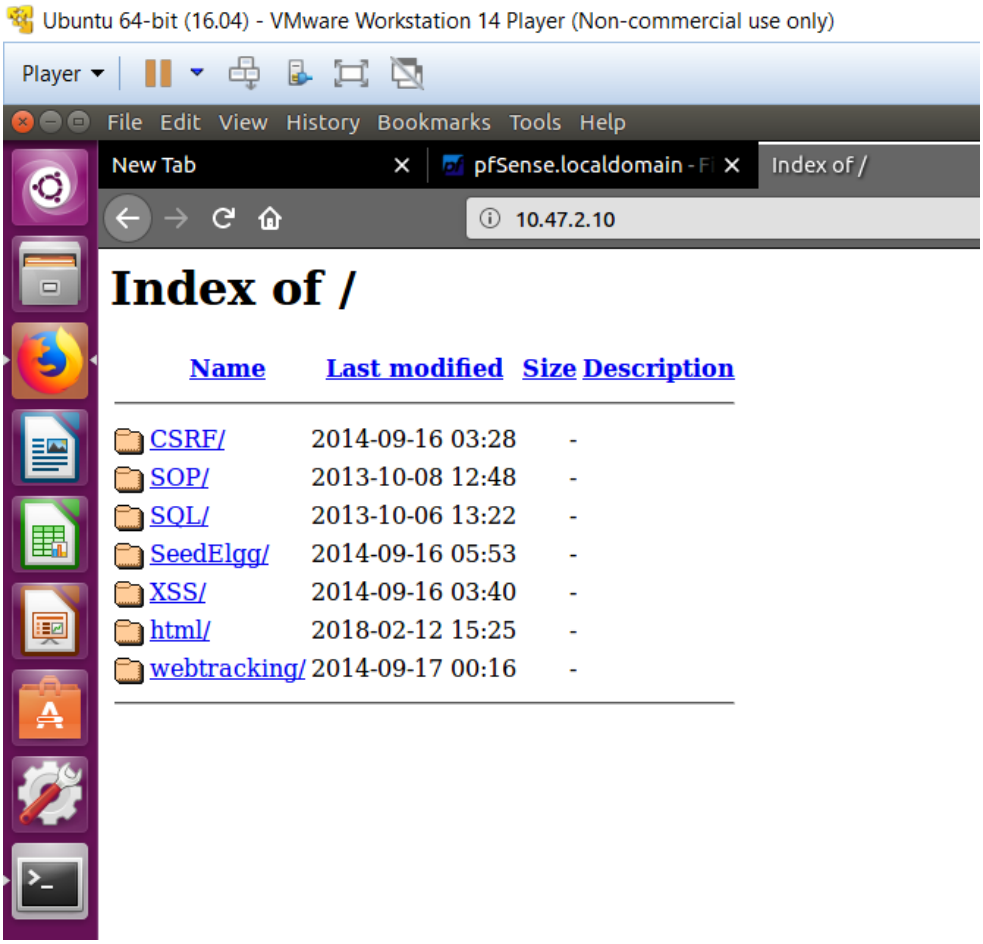


(Figure 3)

Apache web server and ssh server were installed on the server device.The operational form of the web server and the ssh server is shown in the screenshots below.

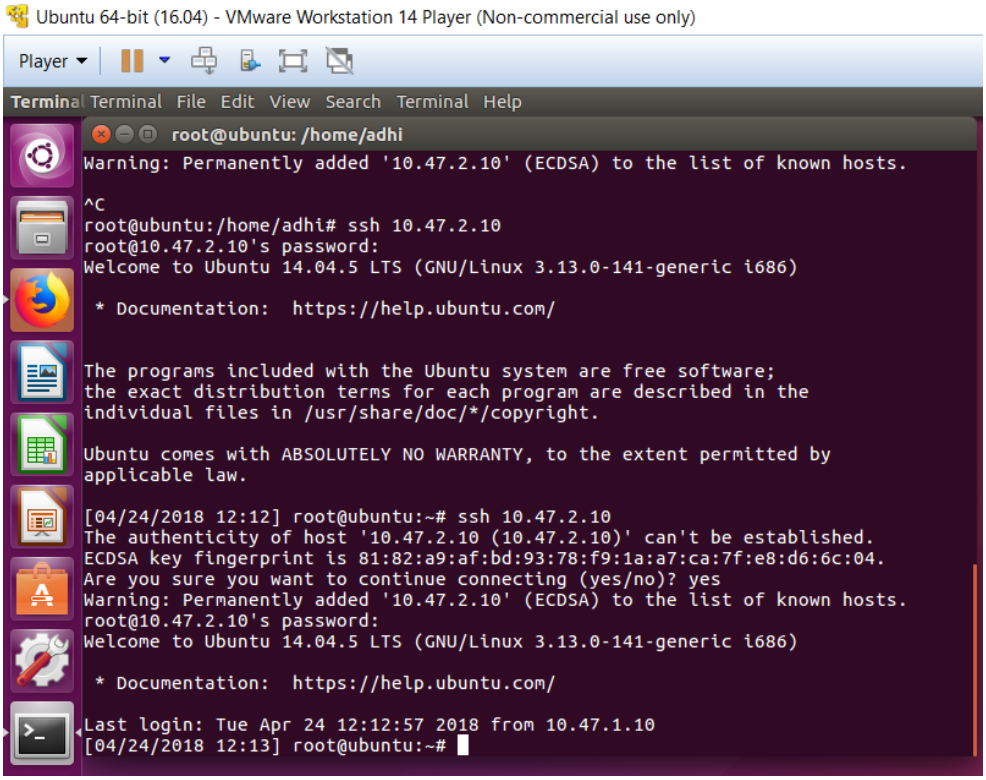

SEEDUbuntu12.04 - VMware Workstation 14 Player (Non-commercial use only)
Player
Terminal File Edit View Search Terminal Help
Terminal
[04/18/2018 17:45] seed@ubuntu:~$ sudo apt install apache2 -y
[sudo] password for seed:
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version.
The following packages were automatically installed and are no longer required:
cdparanoia checkbox compizconfig-backend-gconf espeak foomatic-db-engine
g++-4.6 gir1.2-gst-plugins-base-0.10 gir1.2-gstreamer-0.10 gir1.2-gtk-2.0
gir1.2-indicate-0.7 gir1.2-launchpad-integration-3.0 gksu gnome-games-data
gtk2-engines guile-1.8-libs ibus-pinyin-db-android k3b k3b-data k3b-i18n
kde-l10n-engb language-pack-kde-en lesstif2 libappindicator1
libatk-adaptor-schemas libattica0.3 libavahi-ui-gtk3-0 libbind9-80
libboost-serialization1.46.1 libbrlapi0.5 libcamel-1.2-29 libclass-isa-perl
libclucene0ldbl libcmis-0.2-0 libdaq0 libdconf-dbus-1-0 libdconf-qt0
libdconf0 libdee-qt5-3 libdiscid0 libdns81 libdotconf1.0 libdvdnav4
libdvdread4 libexiv2-11 libflac++6 libgconf2-4 libgd2-xpm libgdu-gtk0
libgdu0 libgexiv2-1 libgksu2-0 libglapi-mesa-lts-quantal libglew1.6
libglewmx1.6 libgnome-bluetooth8 libgnome-menu2 libgnome2-common
libgphoto2-2 libgphoto2-port0 libgrail5 libgweather-3-0 libibus-1.0-0
libimobiledevice2 libindicate-gtk3 libindicate5 libindicator7 libisccfg82
libk3b6 libkcddb4 libkcompactdisc4 libkipi8 libkpathsea5 liblcms1
liblua5.1-0 liblwres80 libmng1 libmpc2 libmpcdec6 libmusicbrainz3-6
libmusicbrainz5-0 libnepomukdatamanagement4 libnepomuksync4 libnux-2.0-0
counter_hacking

(Figure 4)

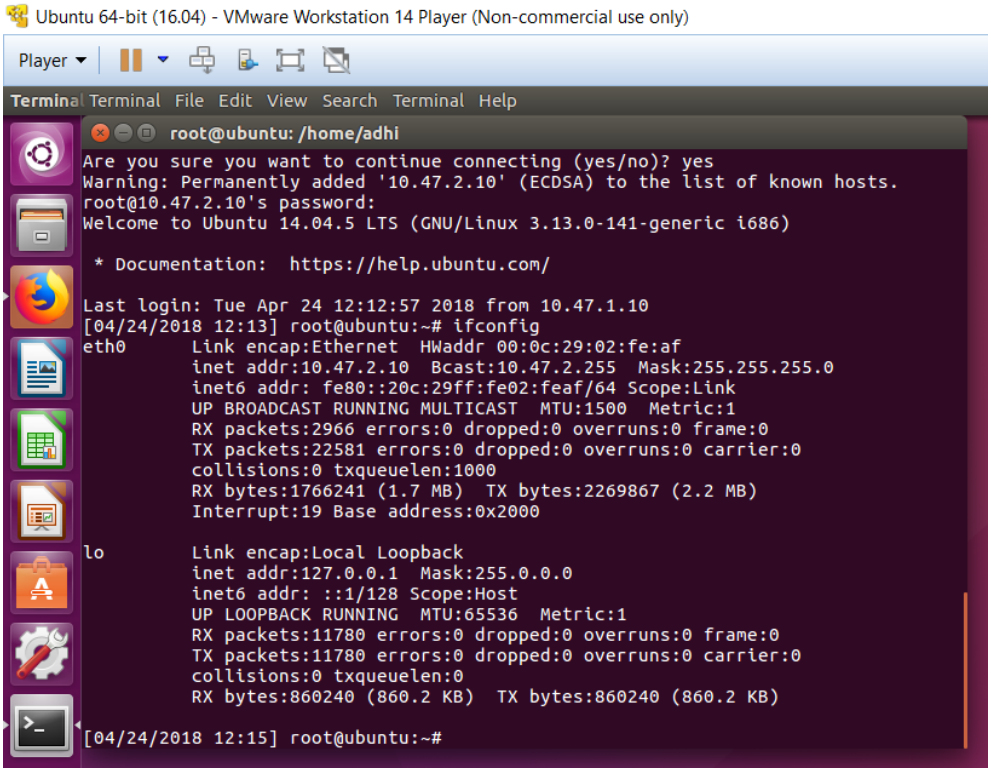


(Figure 5)

The operational form of SSH server is shown in the screenshots below.



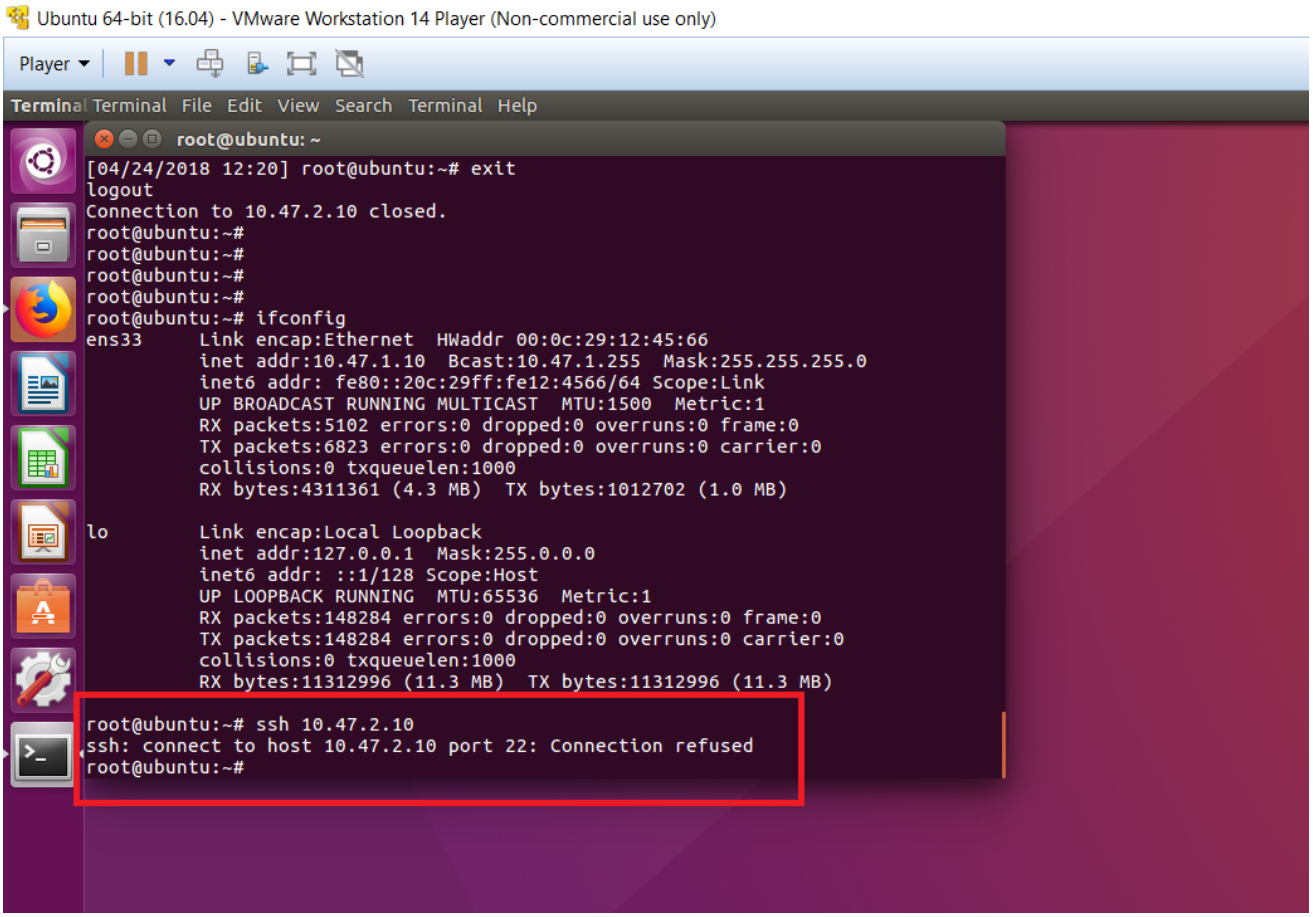
(Figure 6)



(Figure 7)

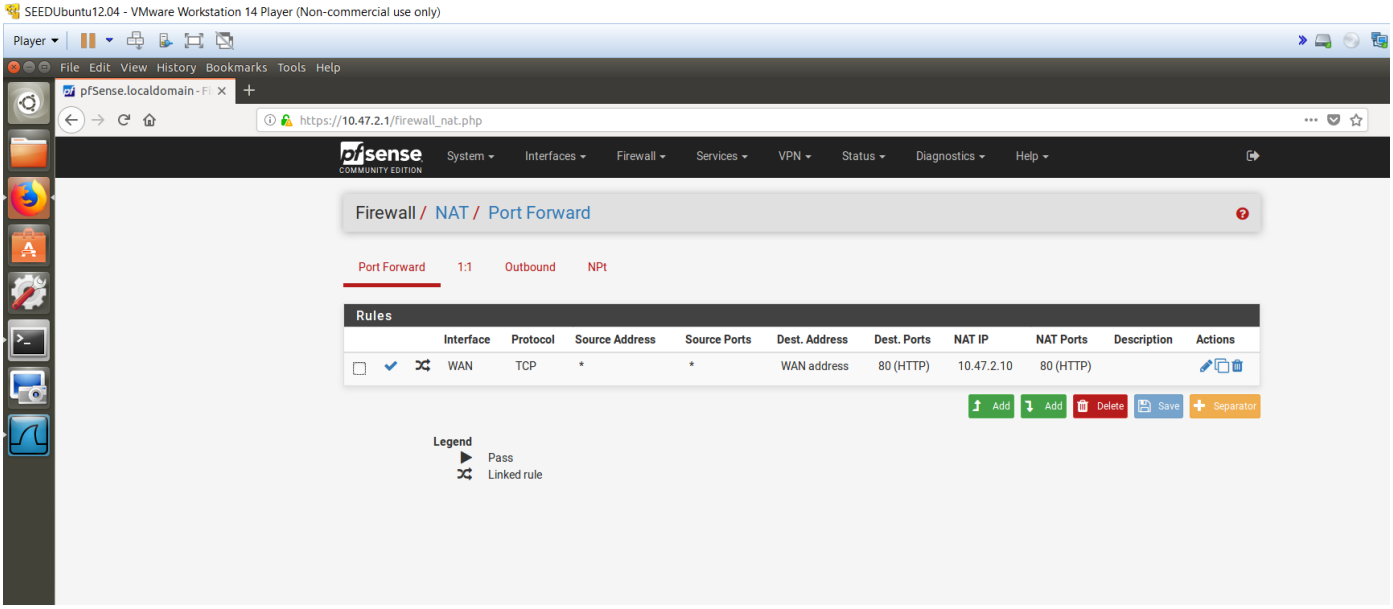
Task 2 – Basic Security Configuration

The effect of the applied access list is that the server allows the client to access the HTTP server and also rejects the SSH requests. For HTTP access, see Figure 5. The effect of SSH blocking in the access list is shown in the figure 8.



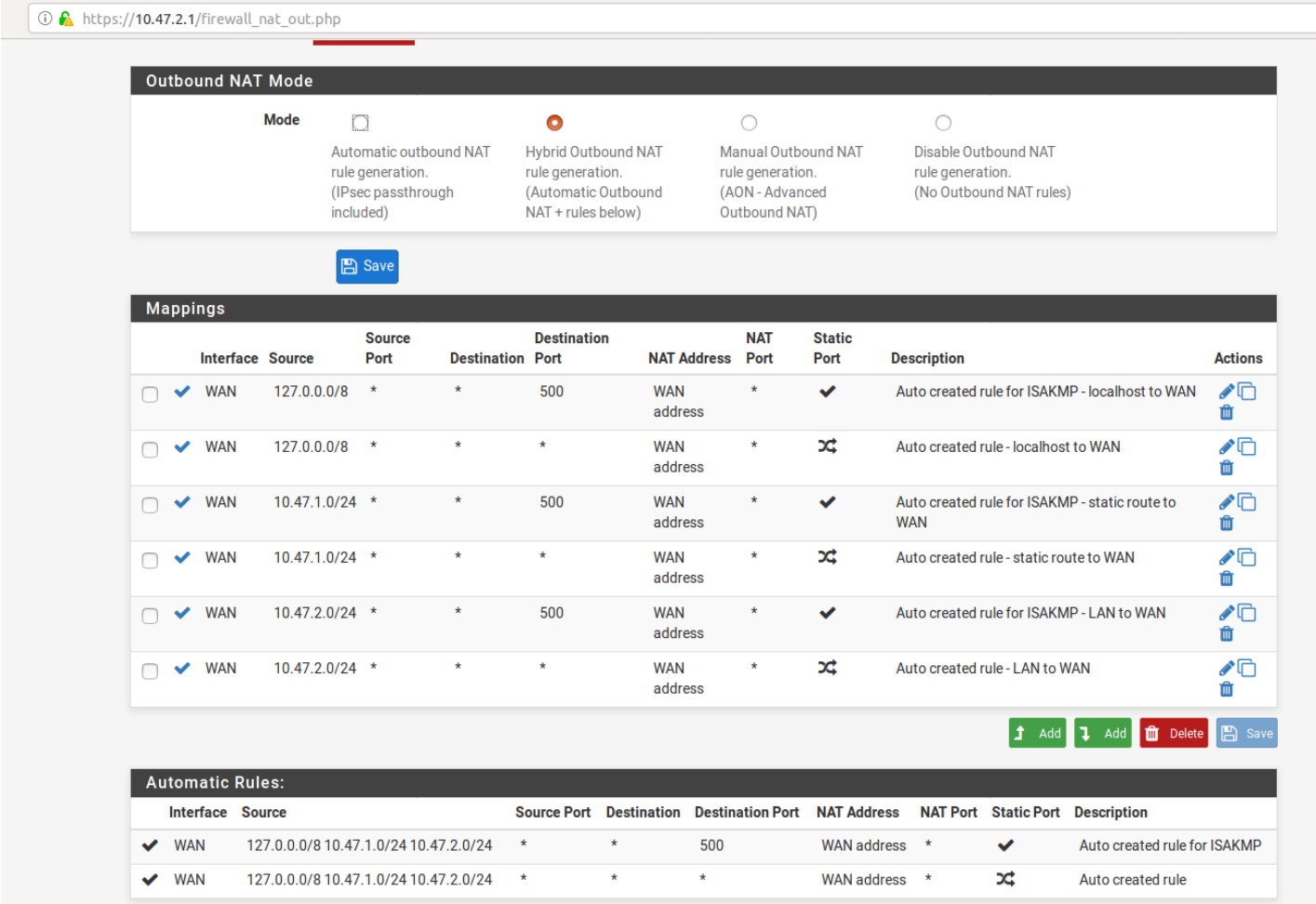
(Figure 8)

Task 3 – Basic Network Address Translation (NAT) Configuration



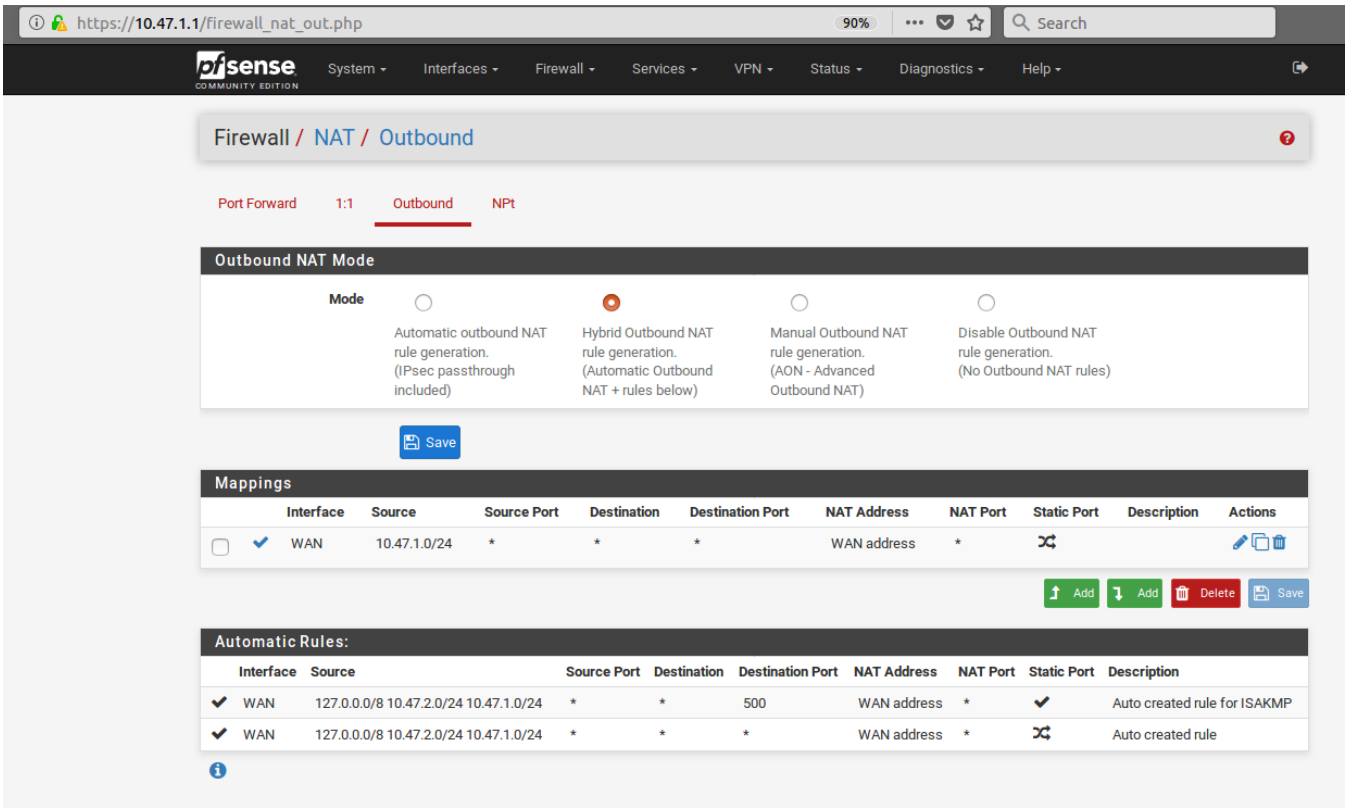
(Figure 9)

In this NAT rule, we specify that traffic coming from any source for the web server on the WAN interface of the firewall to be directed towards the web server (10.47.2.10) on port 80. We allowed to have hybrid outbound NAT'ing which allows to have implicit NAT rules along with some manual rules. So in addition to the access list, we have the following implicit rules. See Figure 10 for implicit rules applied in our network.


https://10.47.2.1/firewall_nat_out.php
Outbound NAT Mode
Mode
Automatic outbound NAT rule generation. (IPsec passthrough included)
Hybrid Outbound NAT rule generation. (Automatic Outbound NAT + rules below)
Manual Outbound NAT rule generation. (AON - Advanced Outbound NAT)
Disable Outbound NAT rule generation. (No Outbound NAT rules)
Save
Mappings
Interface Source Source Port Destination Destination Port NAT Address NAT Port Static Port Description Actions
WAN 127.0.0.0/8 * * 500 WAN address * Auto created rule for ISAKMP - localhost to WAN
WAN 127.0.0.0/8 * * * WAN address * Auto created rule - localhost to WAN
WAN 10.47.1.0/24 * * 500 WAN address * Auto created rule for ISAKMP - static route to WAN
WAN 10.47.1.0/24 * * * WAN address * Auto created rule - static route to WAN
WAN 10.47.2.0/24 * * 500 WAN address * Auto created rule for ISAKMP - LAN to WAN
WAN 10.47.2.0/24 * * * WAN address * Auto created rule - LAN to WAN
Add
Add
Delete
Save
Automatic Rules:
Interface Source Source Port Destination Destination Port NAT Address NAT Port Static Port Description
WAN 127.0.0.0/8 10.47.1.0/24 10.47.2.0/24 * * 500 WAN address * Auto created rule for ISAKMP
WAN 127.0.0.0/8 10.47.1.0/24 10.47.2.0/24 * * * WAN address * Auto created rule

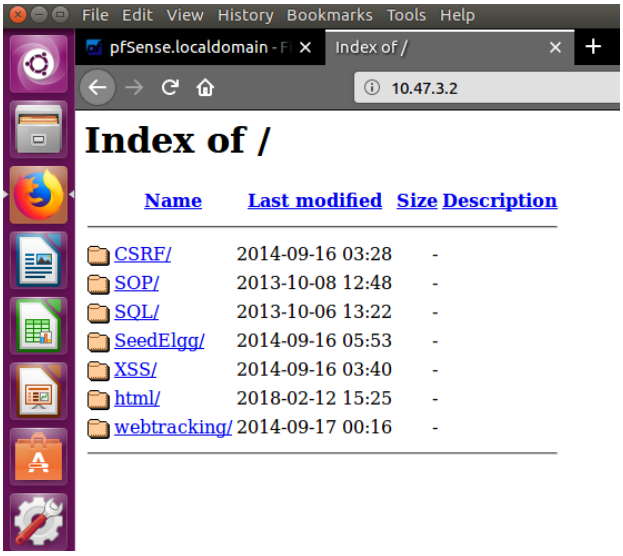
(Figure 10)

We also applied the hybrid outbound NAT'ing on the client side which is shown below.



(Figure 11)

The applied NAT'ing rules allows us to access the web server on the WAN interface (10.47.3.2) of the site B firewall. See figure 12.



(Figure 12)

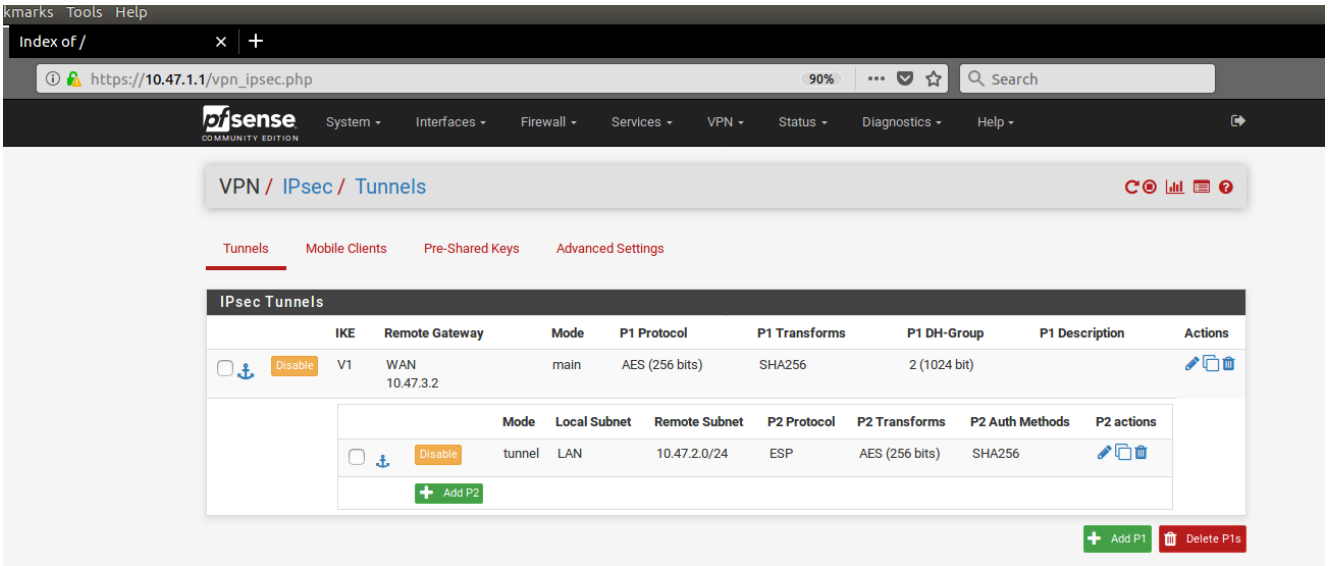
As we can access the web server on the WAN interface (10.47.3.2), we were able to see the NAT translations for this specific request. Figure 13 shows port translation when eth0(client interface) made requests for HTTP on the web server.

65	2018-04-24 09:48:29.6	10.47.2.10	193.0.14.129	DNS	113	Standard query 0x0d4f AAAA activity-stream-icons.services.mozilla.com
66	2018-04-24 09:48:30.7	10.47.2.10	193.0.14.129	DNS	80	Standard query 0xfdae A gmail.com
67	2018-04-24 09:48:30.7	10.47.2.10	193.0.14.129	DNS	80	Standard query 0x55f3 AAAA gmail.com
68	2018-04-24 09:48:40.2	10.47.3.1	10.47.2.10	TCP	74	2014→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3881224570 TSecr=0 WS=128
69	2018-04-24 09:48:40.2	10.47.2.10	10.47.3.1	TCP	74	80→2014 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3158780 TSecr=3881224570 WS=128
70	2018-04-24 09:48:40.2	10.47.3.1	10.47.2.10	TCP	66	2014→80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3881224571 TSecr=3158780
71	2018-04-24 09:48:40.2	10.47.3.1	10.47.2.10	HTTP	417	GET / HTTP/1.1
72	2018-04-24 09:48:40.2	10.47.2.10	10.47.3.1	TCP	66	80→2014 [ACK] Seq=1 Ack=352 Win=30080 Len=0 TSval=3158780 TSecr=3881224572
73	2018-04-24 09:48:40.2	10.47.2.10	10.47.3.1	HTTP	830	HTTP/1.1 200 OK (text/html)
74	2018-04-24 09:48:40.2	10.47.3.1	10.47.2.10	TCP	66	2014→80 [ACK] Seq=352 Ack=765 Win=30848 Len=0 TSval=3881224575 TSecr=3158781
75	2018-04-24 09:48:41.4	10.47.3.1	10.47.2.10	HTTP	455	GET /icons/blank.gif HTTP/1.1
76	2018-04-24 09:48:41.4	10.47.2.10	10.47.3.1	HTTP	281	HTTP/1.1 304 Not Modified
77	2018-04-24 09:48:41.4	10.47.3.1	10.47.2.10	TCP	66	2014→80 [ACK] Seq=741 Ack=980 Win=32256 Len=0 TSval=3881225768 TSecr=3159079
78	2018-04-24 09:48:41.4	10.47.3.1	10.47.2.10	HTTP	456	GET /icons/folder.gif HTTP/1.1
79	2018-04-24 09:48:41.4	10.47.2.10	10.47.3.1	HTTP	281	HTTP/1.1 304 Not Modified
80	2018-04-24 09:48:41.5	10.47.3.1	10.47.2.10	TCP	66	2014→80 [ACK] Seq=1131 Ack=1195 Win=33792 Len=0 TSval=3881225784 TSecr=3159083
81	2018-04-24 09:48:46.4	10.47.3.1	10.47.2.10	TCP	66	2014→80 [FIN, ACK] Seq=1131 Ack=1195 Win=33792 Len=0 TSval=3881230785 TSecr=3159083
82	2018-04-24 09:48:46.4	10.47.2.10	10.47.3.1	TCP	66	80→2014 [FIN, ACK] Seq=1195 Ack=1132 Win=32256 Len=0 TSval=3160334 TSecr=3881230785
83	2018-04-24 09:48:46.4	10.47.3.1	10.47.2.10	TCP	66	2014→80 [ACK] Seq=1132 Ack=1196 Win=33792 Len=0 TSval=3881230787 TSecr=3160334
84	2018-04-24 09:49:13.0	10.47.2.10	199.7.91.13	DNS	95	Standard query 0xa2fe A productsearch.ubuntu.com
85	2018-04-24 09:49:13.0	10.47.2.10	199.7.91.13	DNS	70	Standard query 0xb8dc NS <Root>
86	2018-04-24 09:49:13.0	10.47.2.10	199.7.91.13	DNS	95	Standard query 0xde4e AAAA productsearch.ubuntu.com

(Figure 13)

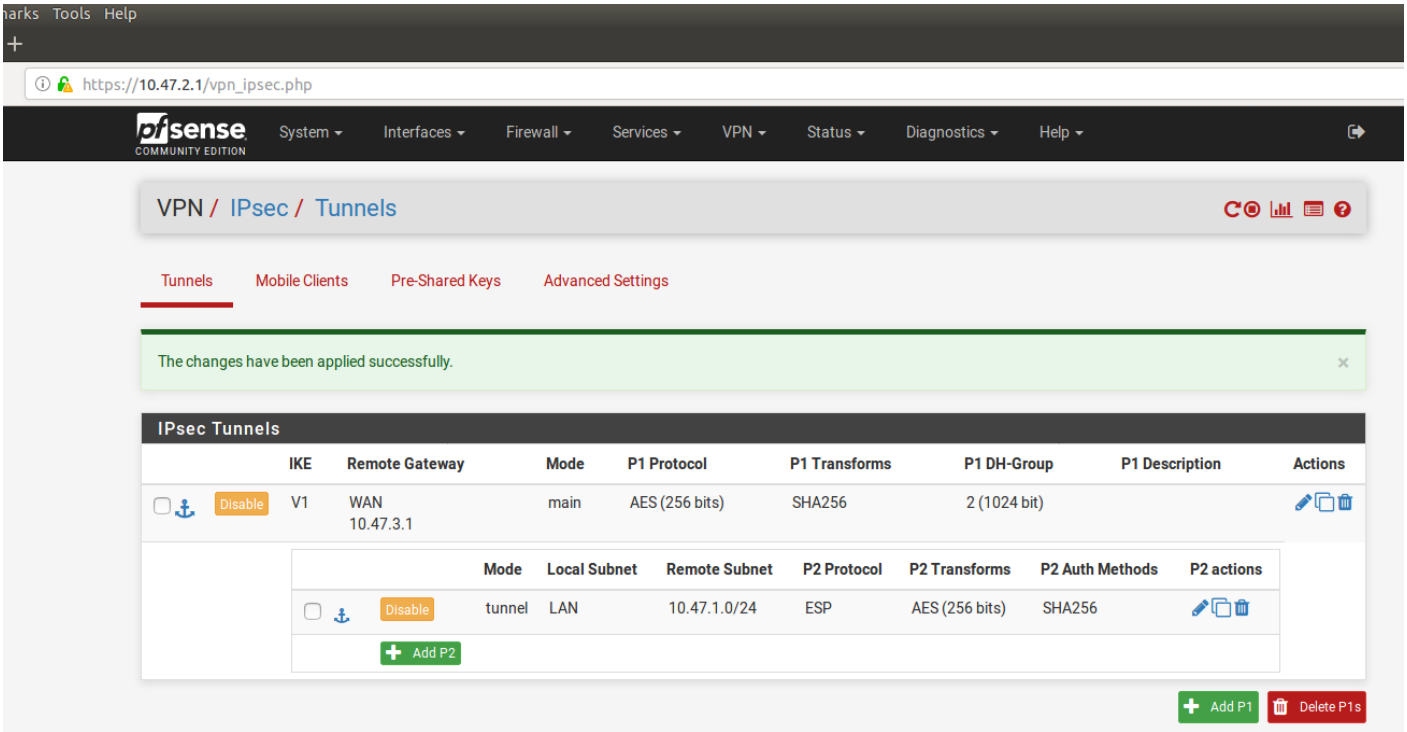
Task 4 – Basic Site-to-Site VPN Configuration

For the client side, the VPN rule is shown in Figure 14. IKEv2 was used for key exchange which supports EAP authentication, has built in NAT traversal and consumes less bandwidth. The remote gateway is mentioned as the IP address of the WAN interface of the server firewall. In Figure 14, we have configured phase 1 and phase 2.



(Figure 14)

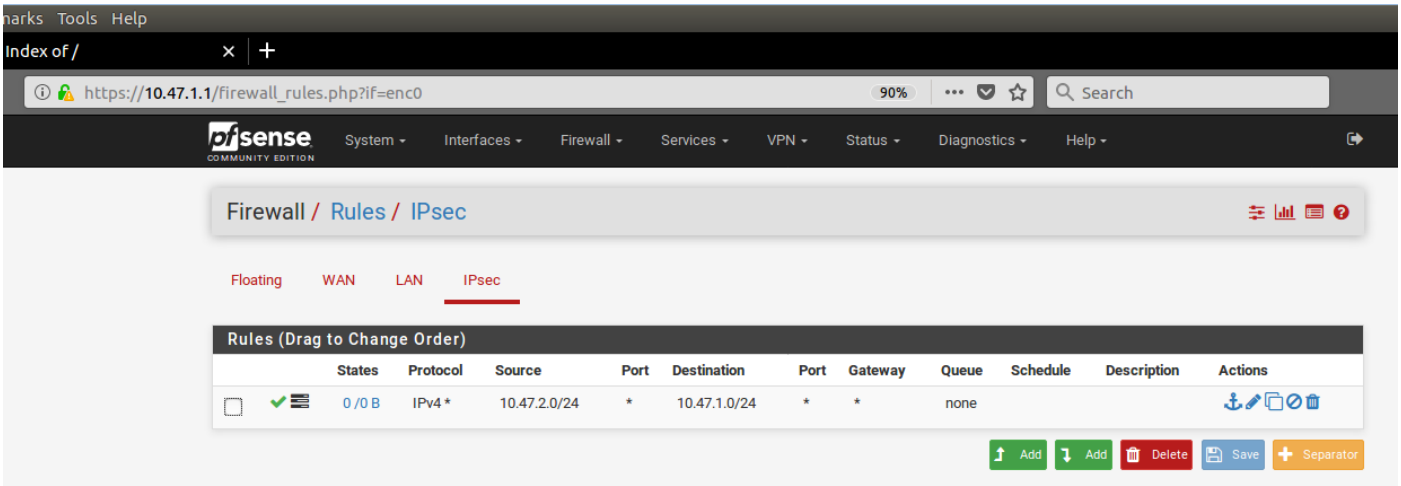
The same IPsec configurations were made on the server side which are shown in Figure 15.



(Figure 15)

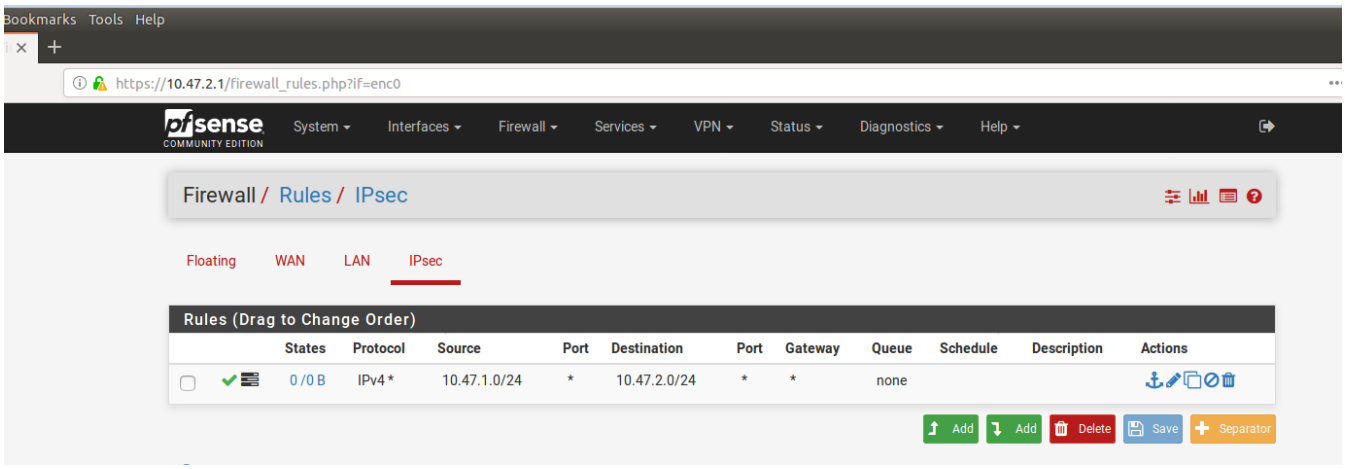
We also need to pass IPsec traffic through the firewall, so we need to add a rule for IPsec.

For client site:



(Figure 16)

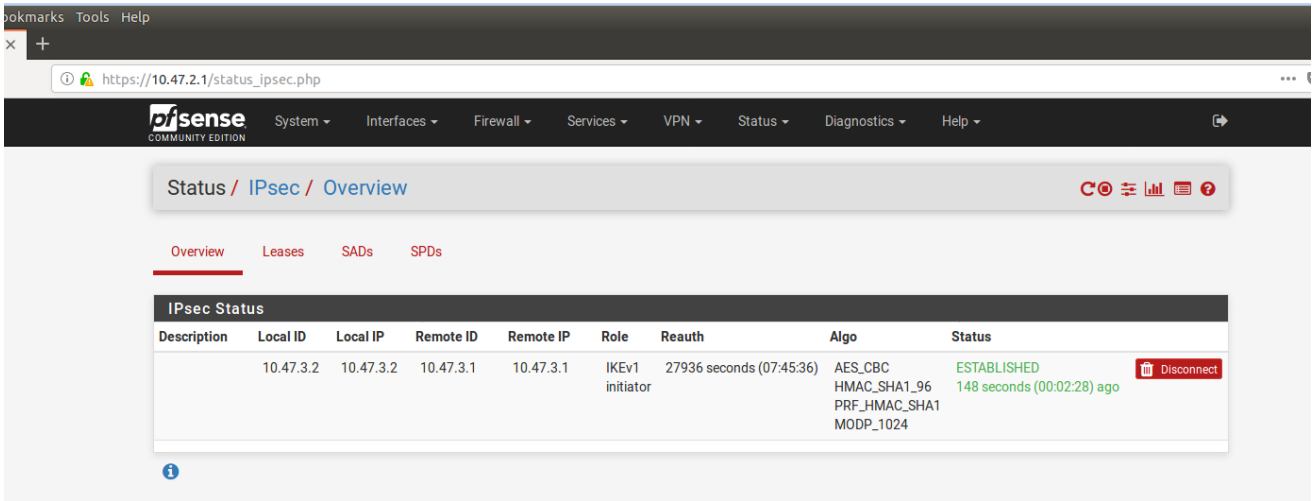
We also need to pass IPsec traffic through the firewall, so we need to add a rule for IPsec.
For server side:



(Figure 17)

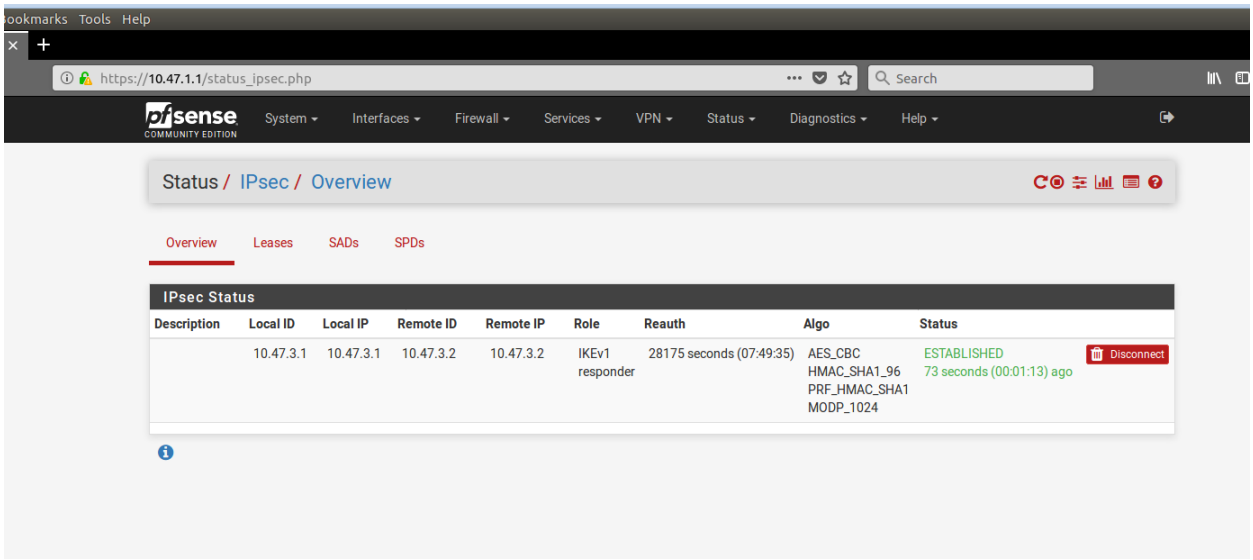
We have to connect the IPsec tunnels: status -> IPsec, click on connect. This will establish the VPN tunnel.

VPN tunnel establishment on server side:



(Figure 18)

VPN tunnel establishment on client side:



(Figure 19)