HACETTEPE UNIVERSITY
ENGINEERING DEPARTMENT
COMPUTER ENGINEERING

LESSON
BBM 465 INFORMATION SECURITY LAB.

ADVISOR
Yasin Şahin, Işıl Karabey

EXPERIMENT 5
host-to-host VPN using IPSec Protocol

GROUP NO : 30

GROUP MEMBERS

Hasan Hüseyin TOPÇU	21228764
Taha BAŞKAK	21228104

Contents

Technical Information

VPN, host-to-host VPN

The VPN, which is a virtual private network, provides the connection between two hosts via a crypto VPN tunnel. A virtual network is created between the two hosts and the entire data stream is carried over this network. Thus, applications running on this VPN benefit from the functionality, security, and management of VPN. VPNs usually provide authenticated connections using tunneling protocols and encryption techniques. VPN is a known application among today's computer users because it is an entrance ban applied to some internet sites. Users can bypass these prohibitions using VPN. The VPN service handles by the ISP(Internet Serivce Provider) it receives. In other words, the VPN serves according to the region where the internet service provider is located. If the ISP in which the VPN used is in service is in the US, the user is also served like the US.

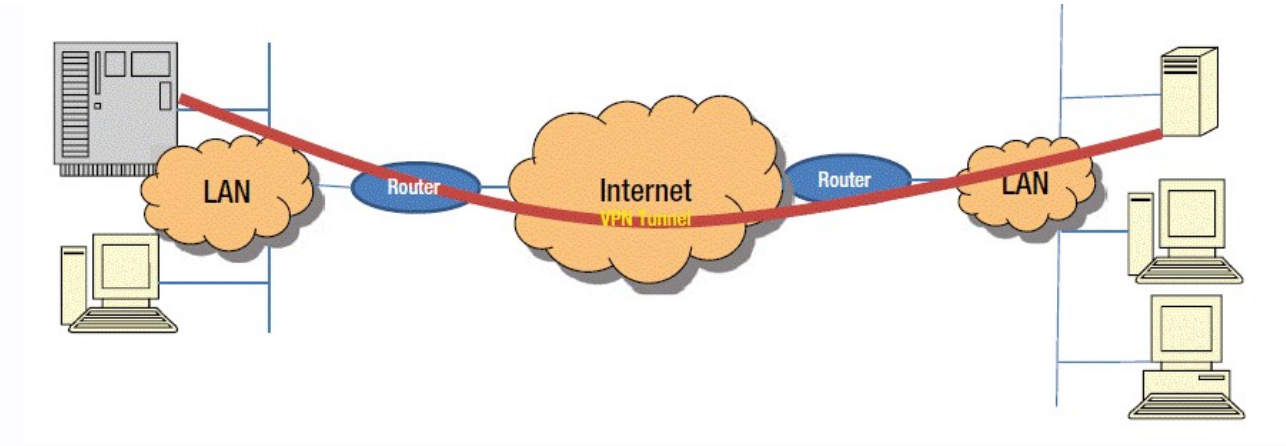


Figure 12-4. Host-to-Host VPN

Figure Reference : http://academlib.com/26730/computer_science/remote_access_host-to-site

IPSec

The Internet Protocol Security (IPSec) protocol secures network communications by verifying and encrypting incoming and outgoing ip packets. IPSec can provide security between host-to-host, newtwork-to-host, and network-to-network connections. IPSec basically provides the following security situations: Data Confidentiality, Data Integrity, Message Origin Authentication.

IPSec provides security for data security using various encryption algorithms. Thus, even if the malicious users access the data directly, they will still be encrypted. IPSec uses encryption algorithms such as DES, DES3, and AES for data security.

IPSec also provides data integrity. Data integrity is a guarantee that the sender does not change the addition or subtraction until it reaches the receiver. IPSec uses hash algorithms such as MD5 and SHA-1 for data integrity.

Another function that IPSec provides is message origin authentication. That is, they can guarantee that the donation comes from whom and the sender is the sender. This function ensures that the data security is done with the correct source.

IPSec actually consists of several IP protocols. These are IKE(Internet Key Exchange), ESP(Encapsulating Security Payload) and AH(Authentication Header). IPSec uses these protocols to create a secure tunnel between two hosts.

Internet Key Exchange(IKE) : He is responsible for the pyrogenation of the keys. The sender and receiver exchange keys. Because IPSec uses symmetric encryption algorithm for data encryption, the sharing of keys must be secure, so IKE is used. IKE protocols enable key sharing in a secure way.

Authentication Header(AH) : AH provides data integrity and data source authentication. The AH is buried in the desired data to be protected. Since the ESP protocol, the AH protocol has lost its significance.

Encapsulating Security Payload(ESP) : IPSec provides data confidentiality with this ESP. ESP provides confidentiality of the data, that is, secure data flow traffic. The pinging process will be secure communication so that the ICMP will not be with ESP but with the ESP. ESP can also see the AH in action, ie authentication. The ESP denumed the AH recommendation.

Libreswan

Libreswan is a free software for the supported and standardized VPN protocol using IPSec and IKE protocols. NSS uses the library. We used this application for this experiment.

NAT-TRAVERSAL

The address information of the sender needs to be changed in each IP packet that moves between interconnected hosts. This process of changing IP is called NAT (network address translation). With NAT, millions of computers connected to the Internet need not be given different IP addresses. You can not ping any items outside the virtual NAT network because of the way VirtualBox applies NAT. For this reason, we used bridge adapter for this neney not NAT. Bridged network is for more sophisticated networking requirements such as network simulations and servers running at a guest. When enabled, VirtualBox connects to one of your installed network cards and directly switches network packets by bypassing your host operating system's network stack.

Wireshark

WireShark is a program that allows you to analyze all internet network traffic. The selected internet interface analyzes incoming and outgoing packets by listening. It provides the source and destination address, the protocol used, the protocol in, and the data content such as the contents of the data. More than 750 protocols can be analyzed. It is a widely and widely used packet analysis program suitable for many operating systems. In this experiment, we used wireshark to observe which protocol was used for pinging.

Experiment Phases

In this experiment, we created a host-to-host VPN tunnel application using the IPSec protocol. In this way, we have encrypted secure network connection between two hosts. The ipsec protocol was used for this secure network traffic. The IPSec protocol uses two IP protocols: AH and ESP. AH for authentication, ESP is done in two tasks: Identity Principle Verification and Data Encryption. We used the **libreswan** program, which uses the ipsec protocol for this experiment. We use RSA key encryption algorithm for the securily VPN tunnel. There is another method its name is PSK(Pre Shraed Key). PSK is easer but not safe therefore we used RSA Key method in this experiment. There are two side, the left host and the right host, who have performed the following operations on these hosts. Finally, we ping it and observe that this ping is done with esp protocol instead of ICMP.

Note : We used both ip address and hostname because the hostname is more understandable than ip address. The steps in left host and right host are done in order.

Left Host
ip address : 192.168.2.216
hostname : left.hasan.fedora
operating system : fedora 24

Step 1 : We installed fedora 24 on virtualbox and learned internet ip address via **ifconfig** command as seen as figure 2. After that, ping to right host because we try the connection before begining and we observed not succesfull this ping. Result of our research this error is relevant NAT. We explain this condition in the technical information part and solution is using Bridged Mode instead of NAT for the VM in the host VirtualBox GUI as seen as figure 1.

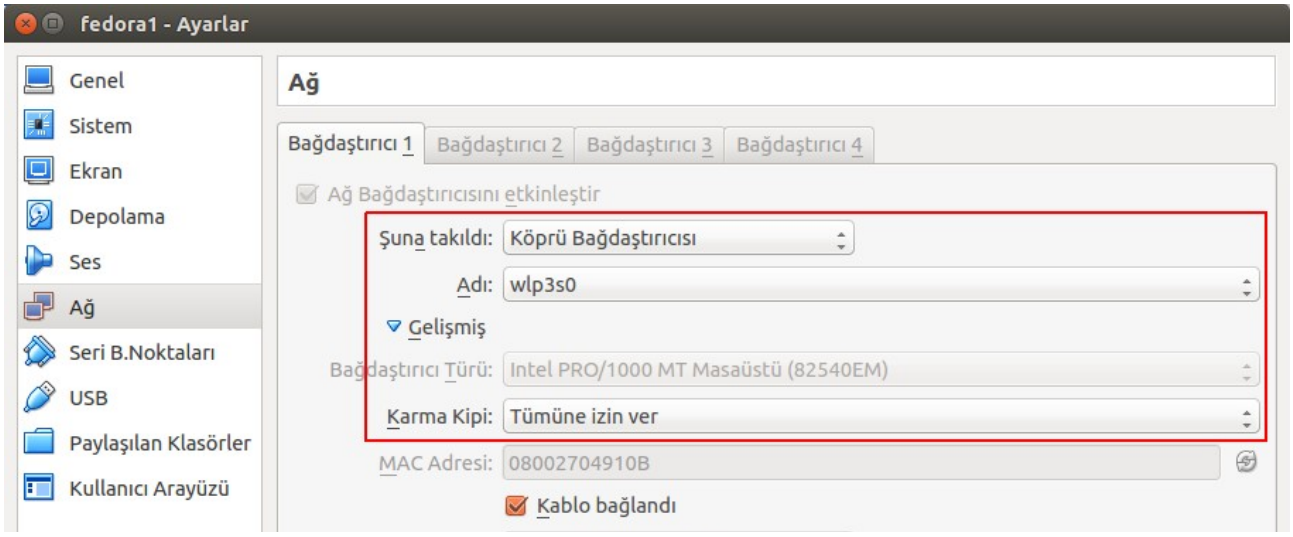
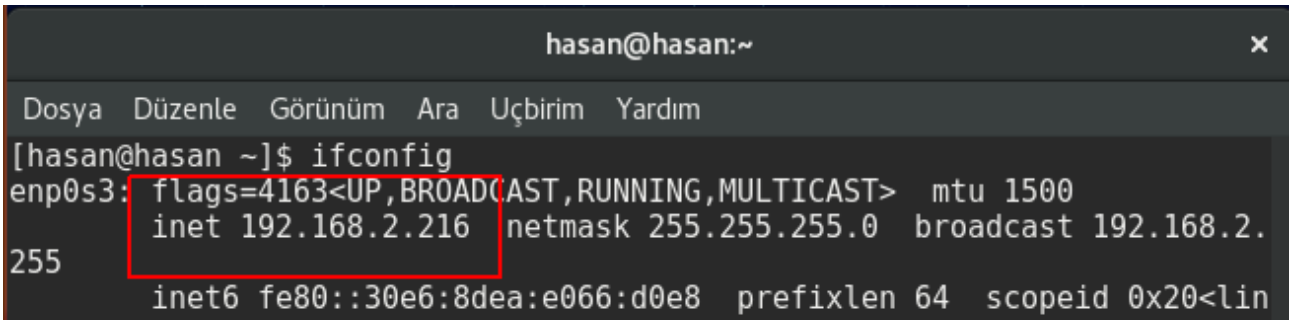


Figure 1


hasan@hasan:~
Dosya Düzenle Görünüm Ara Uçbirim Yardım
[hasan@hasan ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.2.216 netmask 255.255.255.0 broadcast 192.168.2.
255
inet6 fe80::30e6:8dea:e066:d0e8 prefixlen 64 scopeid 0x20<lin
Figure 2

Step 2 : After learning the IP address of the right host, the "192.168.2.127 right.taha.fedora" line was added to the "/etc/hosts" file and the pinged to right host for connection test and the protocol used for ping was observed as icmp on wireshark as seen as figure 3 and figure 4.

```
[hasan@hasan ~]$ ping right.taha.fedora
PING right.taha.fedora (192.168.2.127) 56(84) bytes of data.
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=1 ttl=64 time
=0.690 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=2 ttl=64 time
=0.756 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=3 ttl=64 time
=0.690 ms
^C
--- right.taha.fedora ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.690/0.712/0.756/0.031 ms
[hasan@hasan ~]$ ping 192.168.2.127
PING 192.168.2.127 (192.168.2.127) 56(84) bytes of data.
64 bytes from 192.168.2.127: icmp_seq=1 ttl=64 time=0.367 ms
64 bytes from 192.168.2.127: icmp_seq=2 ttl=64 time=0.917 ms
64 bytes from 192.168.2.127: icmp_seq=3 ttl=64 time=0.727 ms
64 bytes from 192.168.2.127: icmp_seq=4 ttl=64 time=0.680 ms
^C
--- 192.168.2.127 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.367/0.672/0.917/0.200 ms
[hasan@hasan ~]$
```

Figure 3

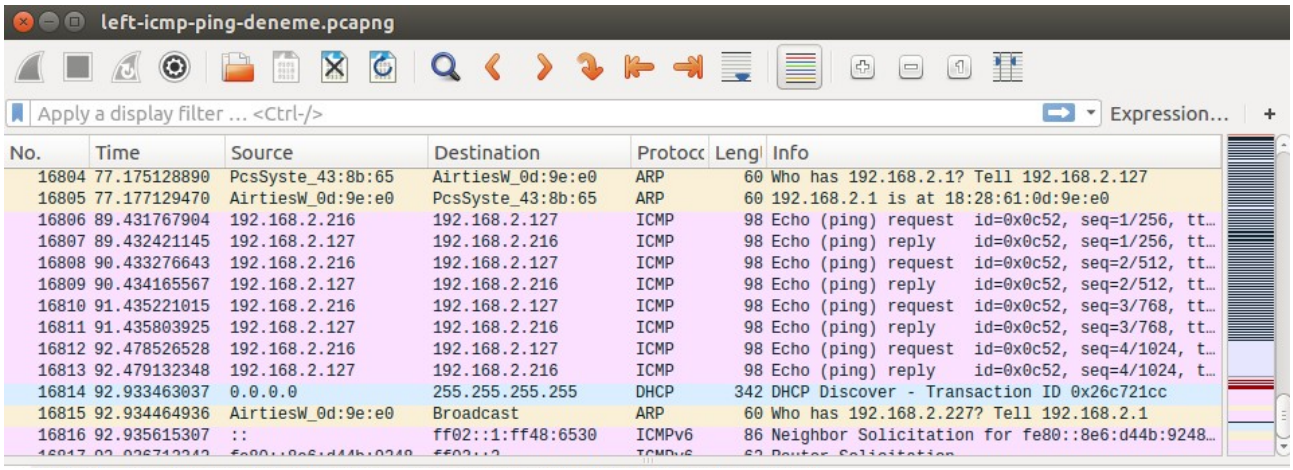


Figure 4

Step 3 : sudo dnf insall libreswan

Installed libreswan program as seen as figure 5.

```
[hasan@hasan ~]$ sudo dnf install wireshark
[sudo] password for hasan:
Last metadata expiration check: 0:17:04 ago on Sat Dec 31 15:18:10 2016
.
```

Figure 5

Step 4 : ipsec newhostkey --output /etc/ipsec.secrets
This command generated RSA key for host to using securily network traffic as seen as figure 6.
”--output /etc/ipsec.secrets” is used to determined name and path.
When we executed firt time this command, we got an error for about initializing NSS database.
sudo ipsec initnss –configdir /etc/ipsec.d
We must run this command once time so initializing nss database. NSS is a userspace library utilized by the libreswan IKE daemon 'pluto' for cryptographic operations.
After that, we executed “**ipsec newhostkey --output /etc/ipsec.secrets**” command and generated RSA key pair with CKAID as seen as figure 6.

```
[hasan@hasan ~]$ sudo ipsec newhostkey --output /etc/ipsec.secrets
[sudo] password for hasan:
/usr/libexec/ipsec/newhostkey: WARNING: file "/etc/ipsec.secrets" exists, append
ing to it
NSS database in /etc/ipsec.d not initialized.
    Please run 'ipsec initnss --configdir /etc/ipsec.d'
[hasan@hasan ~]$ sudo ipsec initnss --configdir /etc/ipsec.d
Initializing NSS database

[hasan@hasan ~]$ sudo ipsec newhostkey --output /etc/ipsec.secrets
/usr/libexec/ipsec/newhostkey: WARNING: file "/etc/ipsec.secrets" exists, append
ing to it
Generated RSA key pair with CKAID b8a1d6e07ba54e5c07bc6b413e40f4ee054ab289 was s
tored in the NSS database
[hasan@hasan ~]$
```

Figure 6

Step 5 : ipsec showhostkey --left –ckaid b8a1d6e07ba54e5c07bc6b413e40f4ee054ab289
We saw rsa key wiht this command as seen as figure 7.
ckaid parameter is determined in the step 4.
We run this command because the leftrsasigkey value is used to configuretion of ipsec.conf file.

```
[hasan@hasan ~]$ sudo ipsec showhostkey --left --ckaid b8a1d6e07ba54e5c07bc6b413
e40f4ee054ab289
        # rsakey AQPUnhapP
        leftrsasigkey=0sAQPUnhapP+VnUD64GkpGUHqCoThDE6/gYAjhA/eVxouoSwc4pLn6+wfr
QFBA7kh7Y2wPIO1h2tTqBpog0AaZbLmctQuzaWIYn1QkXc/JpHHu9TKbxeDO7tC4IdqPMViOcbpAbm3T
cif4mDbpXjWf16Z2PzdEJgIukaWoNph2Y/gNLvqo0a33B7upblSjw2oB/huCkpWYAzRBoFy0tQYZX4RW
0AblwuSESt/TgSQecjTd1fe9/BLm4ZSH/BCwup+2FpGaXAViaOTPzah9/S3Ut4h1FwFpSdxMPowTVX/x
NapBC9usYWqA+/zQjUJRE2/op0RJttDT6kyzX0h6L6x2B72xyxtYfOma/IK7dMAwvgSeHjMcAXFOf+X+
KnHOiFJPUjiF3RtpfxnsOiLgu10SJm8AIKYmluJskRnBpyI5G8XKUd1O8hK7vDbTuwn43kdl+HlUKtWo
591idrcQCRmDkPLVHT9BiHGUQ0+g+vvoFsfiyWjDpPjhUEbDScv95FJNp23PodhIczg5iQ4n1e+0tyRm
klA7YAvOLeOjAYL1WJfLKkOU9z1tqC/8xNvHSUXR8diIKl+RlzTc8rvKHmTW88dujryRHPr4s9Lz/PhB
TkFMAc539S0IkGXs0Rc=
[hasan@hasan ~]$
```

Figure 7

Step 6 : We configureted ipsec.conf file under "/etc" path. This file must same for right and left hosts. ipsec.conf file :

#config setup part is general so we did not configurate this part.
config setup

protostack=netkey
dumpdir=/var/run/pluto/

virtual_private=
%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4:172.16.0.0/12,%v4:25.0.0.0/8,%v4:100.64.0.0/10,%v6:fd00::/8,%v6:fe80::/10

 # hasanTahaTunnel is our tunnel's name
conn hasanTahaTunnel

 # leftid is specify id for left host
 # left parameter is ip address of left hosts
 # leftrsasigkey is RSA key.

leftid=@left
left=192.168.2.216
leftrsasigkey=0sAQPUnhapP+VnUD64GkpGUHqCoThDE6/gYAjhA/eVxouoSwc4pLn6+wfrQFBA7kh7Y2wPIO1h2tTqBpog0AaZbLmctQuzaWIYn1QkXc/JpHHu9TKbxeDO7tC4IdqPMViOcbpAbm3Tcif4mDbpXjWf16Z2PzdEJgIukaWoNph2Y/gNLvqo0a33B7upblSjw2oB/huCkpWYAzRBoFy0tQYZX4RW0AblwuSESt/TgSQecjTd1fe9/BLm4ZSH/BCwup+2FpGaXAViaOTPzah9/S3Ut4h1FwFpSdxMPowTVX/xNapBC9usYWqA+/zQjUJRE2/op0RJttDT6kyzX0h6L6x2B72xyxtYfOma/IK7dMAwvgSeHjMcAXFOf+X+KnHOiFJPUjiF3RtpfxnsOiLgu10SJm8AIKYmluJskRnBpyI5G8XKUd1O8hK7vDbTuwn43kdl+HlUKtWo591idrcQCRmDkPLVHT9BiHGUQ0+g+vvoFsfiyWjDpPjhUEbDScv95FJNp23PodhIczg5iQ4n1e+0tyRmklA7YAvOLeOjAYL1WJfLKkOU9z1tqC/8xNvHSUXR8diIKl+RlzTc8rvKHmTW88dujryRHPr4s9Lz/PhBTkFMAc539S0IkGXs0Rc=

 # rightid is specify id for left host
 # right parameter is ip address of right hosts
 # rightrsasigkey is RSA key.

rightid=@right
right=192.168.2.127
rightrsasigkey=0sAQPEv9LZ1CxXZE2uAEAHIoRKaJdXwgeT/Pf9d/Ii+P2WBGqX1xcjKMHAk8Wwo4od9KgKlzMdNxtp0oot24AW7//2Fw/QMLMWAs+aMdSefARhaDsw8dv6YjLPva2NztzEonggGeDCIt2SCio8WN/pBQynkwemGZqnbxN1AOhZSLTQYrWc5to9DnulCLoVsggGck3pXFRwFCMc1aloluwcdiTH55xTDO6uwBNLXINcDWFCFmSflXUIRWdsSQAKjPUio2TtUMro6aC0JPGNnLD+GkfPgxevAh/j1ffhnCu4AkNcEnYnUQGKITuClbA/ykYo9hzV3Xri2AT/knnDUQGG1Ldn52lJJDcXTD9hx6FUjBAceNyG3TqddMUNhIbI1Q4/OGF9LRBJ/z9kOZQyrkF12tQZWDATmM9eZVDNxCQ9C5WOwsUCBbbipqV5zNz6Fix7eJ4gFTPzuz7pEY/8Sz0e3dXzi5TV5E+zZy8pA0tm3b92eK5oHXg4QX20bCaxMlgNmVStcunVpAb88xEe/I/MqQvfpw0ZleYoE7x70hq3mJVTADqjDOMOI71KqNe//dr90itRQKbv

 # Authentication by rsasig
authby=rsasig
 # use auto=start when done testing the tunnel
auto=start

Step 7 : systemctl start ipsec.service # start ipsec service as seen as figure 8
sudo ipsec auto –add hasanTahaTunnel # add new tunnel that name is hasanTahaTunnel
sudo ipsec suto –up hasanTahaTunnel # hasanTahaTunnel is activated as seen figure 8

We observed “IPSec SA established tunnel mode {ESP=....}” the last line as seen as figure 8.

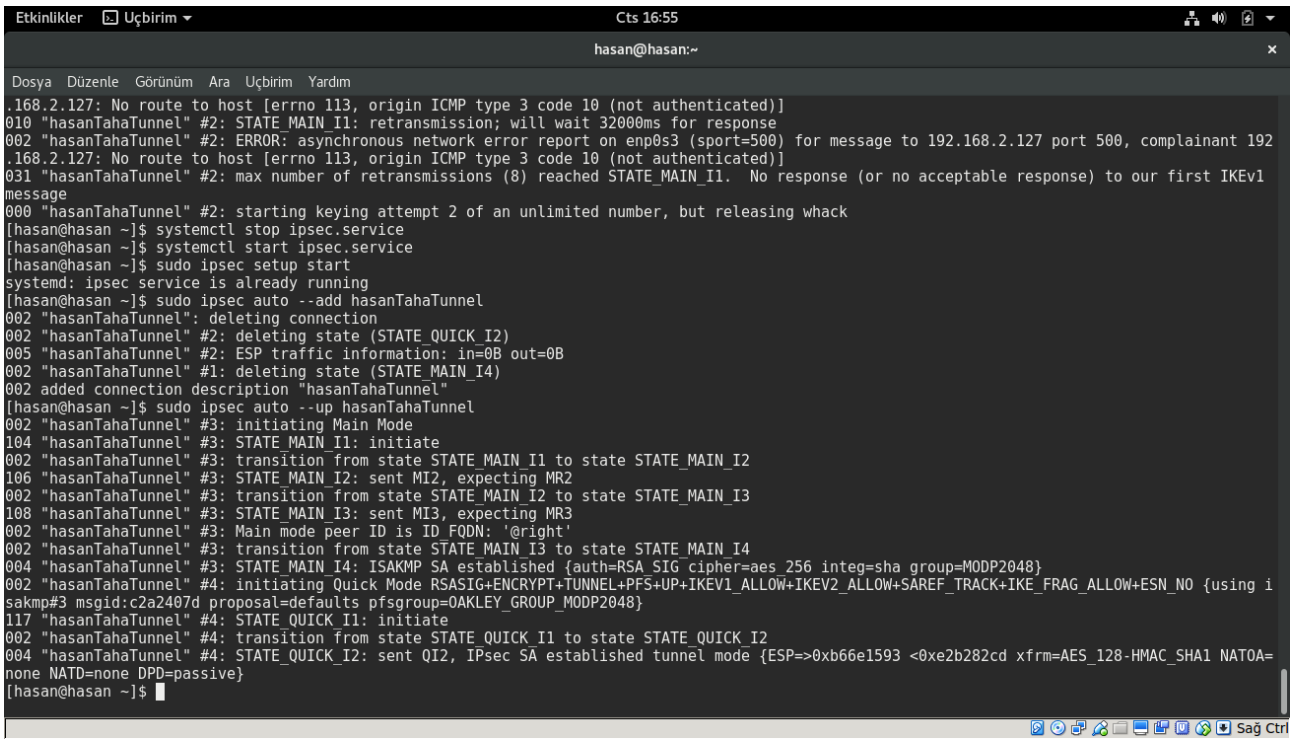


Figure 8

Step 8 : ping 192.168.2.127 # ping to right host using ip address as seen as figure 9.
ping right.taha.fedora # ping to right host using hostname as seen as figure 10.

We tested securily network traffic by ping to right host from left host and we observed that provide conneciton with ESP as seen as figure 11.

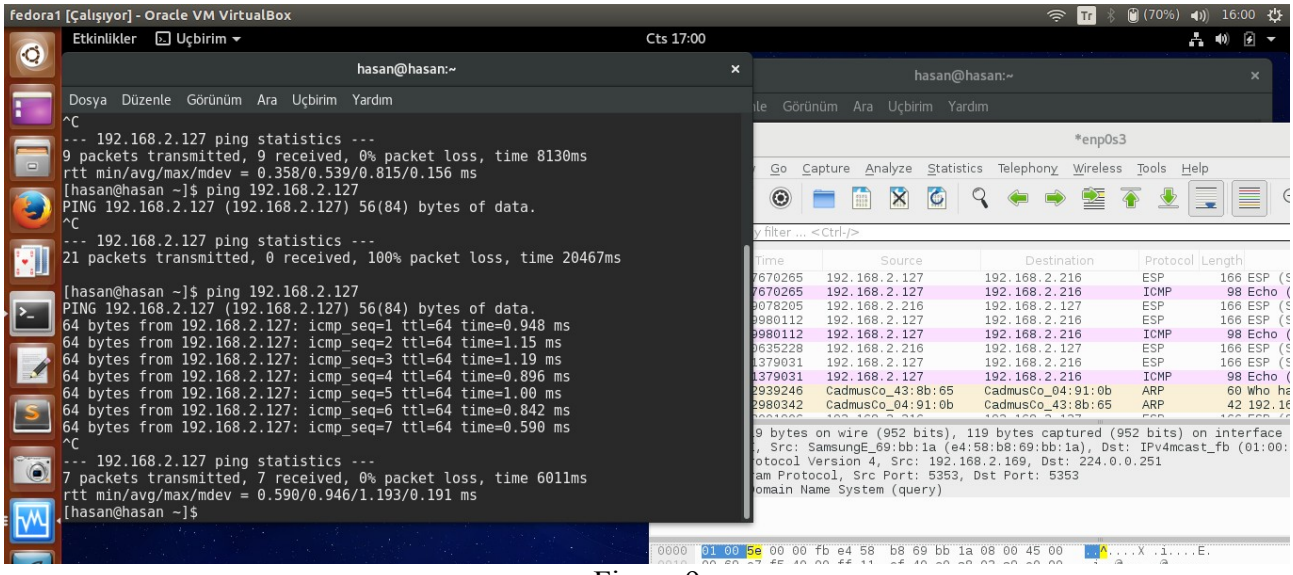


Figure 9

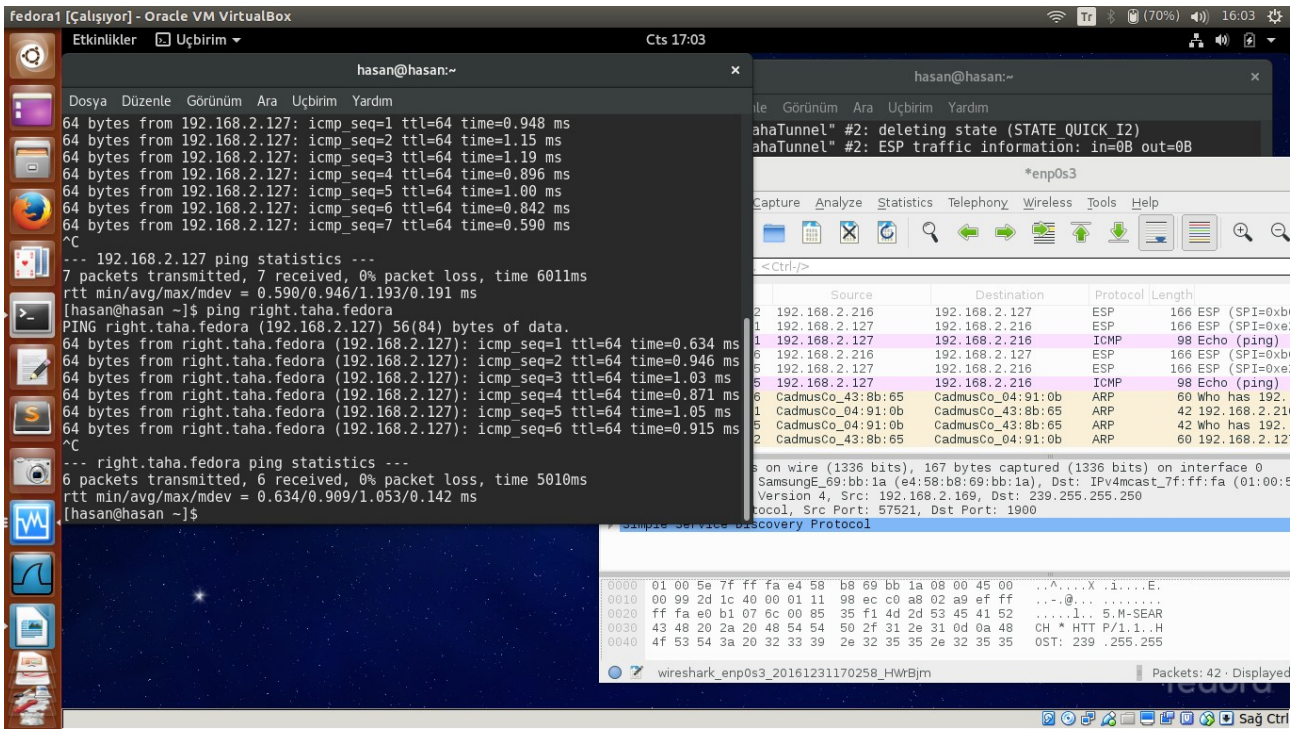

fedora1 [Çalışıyor] - Oracle VM VirtualBox
Etkinlikler
Uçbirim
Cts 17:03
hasan@hasan:~
Dosya Düzenle Görünüm Ara Uçbirim Yardım
64 bytes from 192.168.2.127: icmp_seq=1 ttl=64 time=0.948 ms
64 bytes from 192.168.2.127: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from 192.168.2.127: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 192.168.2.127: icmp_seq=4 ttl=64 time=0.896 ms
64 bytes from 192.168.2.127: icmp_seq=5 ttl=64 time=1.00 ms
64 bytes from 192.168.2.127: icmp_seq=6 ttl=64 time=0.842 ms
64 bytes from 192.168.2.127: icmp_seq=7 ttl=64 time=0.590 ms
^C
--- 192.168.2.127 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6011ms
rtt min/avg/max/mdev = 0.590/0.946/1.193/0.191 ms
[hasan@hasan ~]$ ping right.taha.fedora
PING right.taha.fedora (192.168.2.127) 56(84) bytes of data.
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=1 ttl=64 time=0.634 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=2 ttl=64 time=0.946 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=4 ttl=64 time=0.871 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=5 ttl=64 time=1.05 ms
64 bytes from right.taha.fedora (192.168.2.127): icmp_seq=6 ttl=64 time=0.915 ms
^C
--- right.taha.fedora ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 0.634/0.909/1.053/0.142 ms
[hasan@hasan ~]$
hasan@hasan:~
*enp0s3
Capture Analyze Statistics Telephony Wireless Tools Help
Source Destination Protocol Length
192.168.2.216 192.168.2.127 ESP 166 ESP (SPI=0x
192.168.2.127 192.168.2.216 ESP 166 ESP (SPI=0x
192.168.2.127 192.168.2.216 ICMP 98 Echo (ping)
192.168.2.216 192.168.2.127 ESP 166 ESP (SPI=0x
192.168.2.127 192.168.2.216 ESP 166 ESP (SPI=0x
192.168.2.127 192.168.2.216 ICMP 98 Echo (ping)
CadmusCo_43:8b:65 CadmusCo_04:91:0b ARP 60 Who has 192.
CadmusCo_04:91:0b CadmusCo_43:8b:65 ARP 42 192.168.2.21
CadmusCo_04:91:0b CadmusCo_43:8b:65 ARP 42 Who has 192.
CadmusCo_43:8b:65 CadmusCo_04:91:0b ARP 60 192.168.2.12
wireshark_enp0s3_20161231170258_HWrBjm
Packets: 42 · Displayed
Sağ Ctrl

Figure 10

No.	Time	Source	Destination	Protocc	Lengl	Info
2	0.610761882	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
3	0.611615666	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
5	1.612987420	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
6	1.614024781	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
8	2.614882927	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
9	2.615961412	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
11	3.616870475	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
12	3.617670265	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
14	4.619078205	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
15	4.619980112	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
17	5.620635228	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
18	5.621379031	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
22	6.622001906	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
23	6.622508255	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)

```
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 152
    Identification: 0x241c (9244)
  ▶ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: Encap Security Payload (50)
    Header checksum: 0x8f70 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.2.216
    Destination: 192.168.2.127
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▶ Encapsulating Security Payload
```

Figure 11

Right Host
ip : 192.168.2.127
hostname : right.taha.fedora
operating system : fedora 24

Step 1 : We installed fedora 24 on virtualbox and learned internet ip address via **ifconfig** command as seen as figure 2. After that, ping to left host because we try the connection before begining and we observed not succesfull this ping. Problem and solution is same with left host so using Bridged Mode instead of NAT for the VM in the host VirtualBox GUI as seen as figure 1.

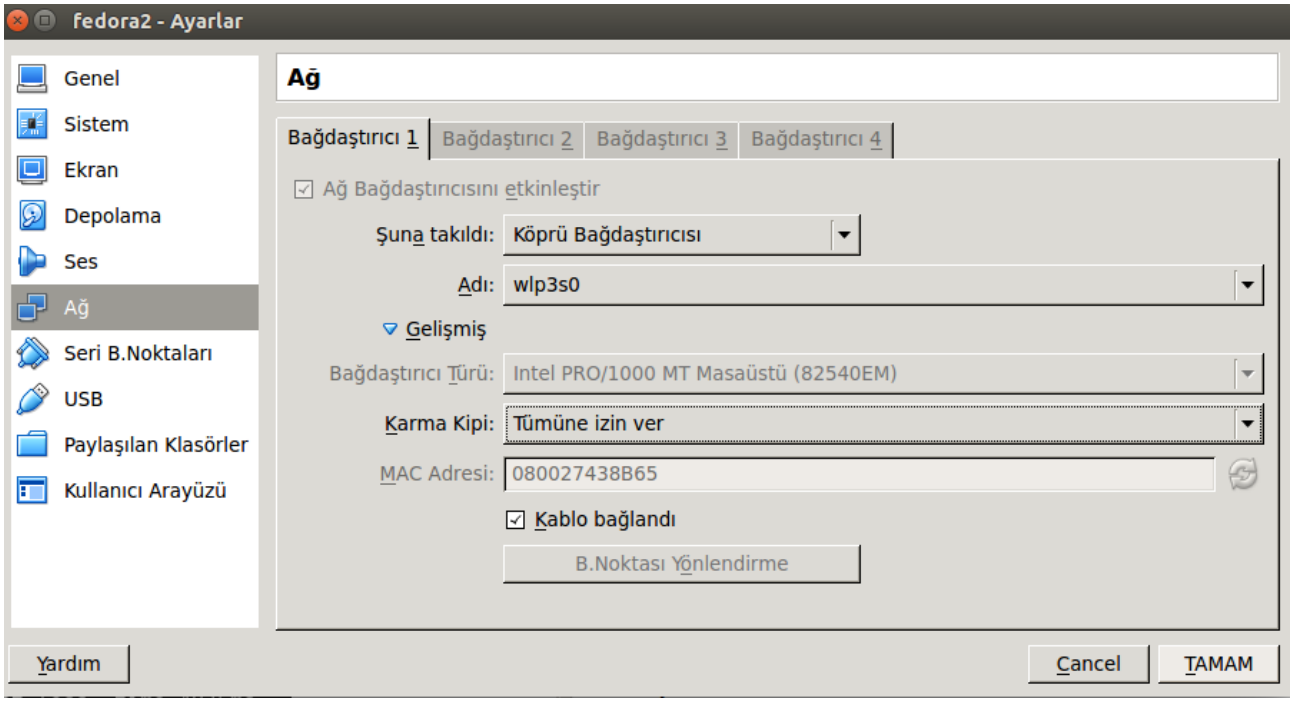


Figure 1

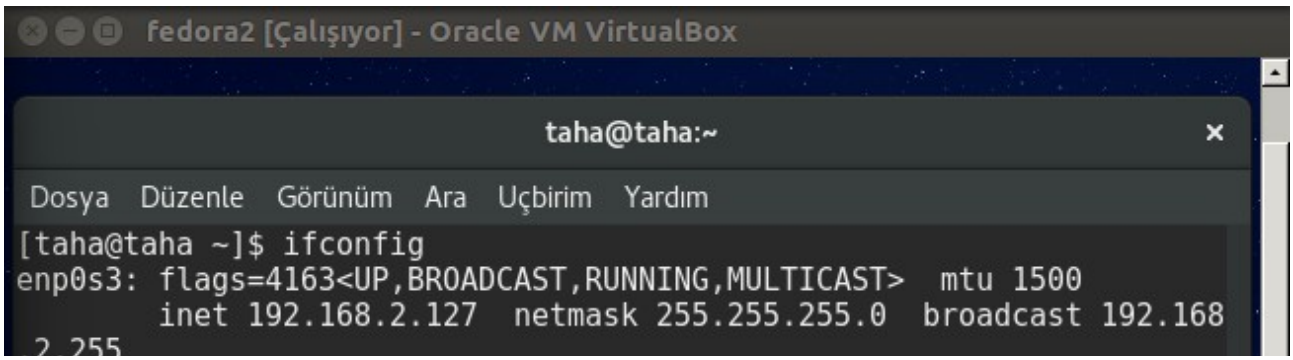


Figure 2

Step 2 : After learning the IP address of the left host, the "192.168.2.216 left.hasan.fedora" line was added to the “/etc/hosts” file and the pinged to left host for connection test and the protocol used for ping was observed as icmp on wireshark as seen as figure 3 and figure 4.

```
[taha@taha ~]$ ping left.hasan.fedora
PING left.hasan.fedora (192.168.2.216) 56(84) bytes of data.
64 bytes from left.hasan.fedora (192.168.2.216): icmp_seq=1 ttl=64 t
ime=0.653 ms
64 bytes from left.hasan.fedora (192.168.2.216): icmp_seq=2 ttl=64 t
ime=0.718 ms
64 bytes from left.hasan.fedora (192.168.2.216): icmp_seq=3 ttl=64 t
ime=0.617 ms
64 bytes from left.hasan.fedora (192.168.2.216): icmp_seq=4 ttl=64 t
ime=0.602 ms
^C
--- left.hasan.fedora ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3099ms
rtt min/avg/max/mdev = 0.602/0.647/0.718/0.051 ms
[taha@taha ~]$ ping 192.168.2.216
PING 192.168.2.216 (192.168.2.216) 56(84) bytes of data.
64 bytes from 192.168.2.216: icmp_seq=1 ttl=64 time=0.507 ms
64 bytes from 192.168.2.216: icmp_seq=2 ttl=64 time=0.896 ms
64 bytes from 192.168.2.216: icmp_seq=3 ttl=64 time=0.637 ms
^C
--- 192.168.2.216 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.507/0.680/0.896/0.161 ms
[taha@taha ~]$
```

Figure 3

No.	Time	Source	Destination	Protocol	Length	Info
18227	48.911642732	192.168.2.127	192.168.2.216	ICMP	98	Echo (ping) request id=0x0c4e, seq=1/256, tt…
18228	48.912077662	192.168.2.216	192.168.2.127	ICMP	98	Echo (ping) reply id=0x0c4e, seq=1/256, tt…
19179	49.916053881	192.168.2.127	192.168.2.216	ICMP	98	Echo (ping) request id=0x0c4e, seq=2/512, tt…
19180	49.916640237	192.168.2.216	192.168.2.127	ICMP	98	Echo (ping) reply id=0x0c4e, seq=2/512, tt…
20200	50.939895434	192.168.2.127	192.168.2.216	ICMP	98	Echo (ping) request id=0x0c4e, seq=3/768, tt…
20201	50.940606457	192.168.2.216	192.168.2.127	ICMP	98	Echo (ping) reply id=0x0c4e, seq=3/768, tt…
21074	51.963911781	192.168.2.127	192.168.2.216	ICMP	98	Echo (ping) request id=0x0c4e, seq=4/1024, t…
21075	51.964289970	192.168.2.216	192.168.2.127	ICMP	98	Echo (ping) reply id=0x0c4e, seq=4/1024, t…
21328	52.992795687	192.168.2.127	192.168.2.216	ICMP	98	Echo (ping) request id=0x0c4e, seq=5/1280, t…
21329	52.995635184	192.168.2.216	192.168.2.127	ICMP	98	Echo (ping) reply id=0x0c4e, seq=5/1280, t…

```
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 84
Identification: 0xbc2c (48172)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 64
Protocol: ICMP (1)
Header checksum: 0xf7d4 [validation disabled]
Source: 192.168.2.127
Destination: 192.168.2.216
[Source GeoIP: Unknown]
```

Figure 4

Step 3 : sudo dnf insall libreswan

Installed libreswan program as seen as figure 5.

```
[taha@taha ~]$ sudo dnf install libreswan
Last metadata expiration check: 1:24:32 ago on Sat Dec 31 14:25:19 2
016.
```

Figure 5

Step 4 : ipsec newhostkey --output /etc/ipsec.secrets
This command generated RSA key for host to using securily network traffic as seen as figure 6.
"--output /etc/ipsec.secrets" is used to determined name and path.
When we executed firt time this command, we got an error for about initializing NSS database as left host.
sudo ipsec initnss –configdir /etc/ipsec.d
We must run this command once time so initializing nss database. NSS is a userspace library utilized by the libreswan IKE daemon 'pluto' for cryptographic operations.
After that, we executed "**ipsec newhostkey --output /etc/ipsec.secrets**" command and generated RSA key pair with CKAID as seen as figure 6.

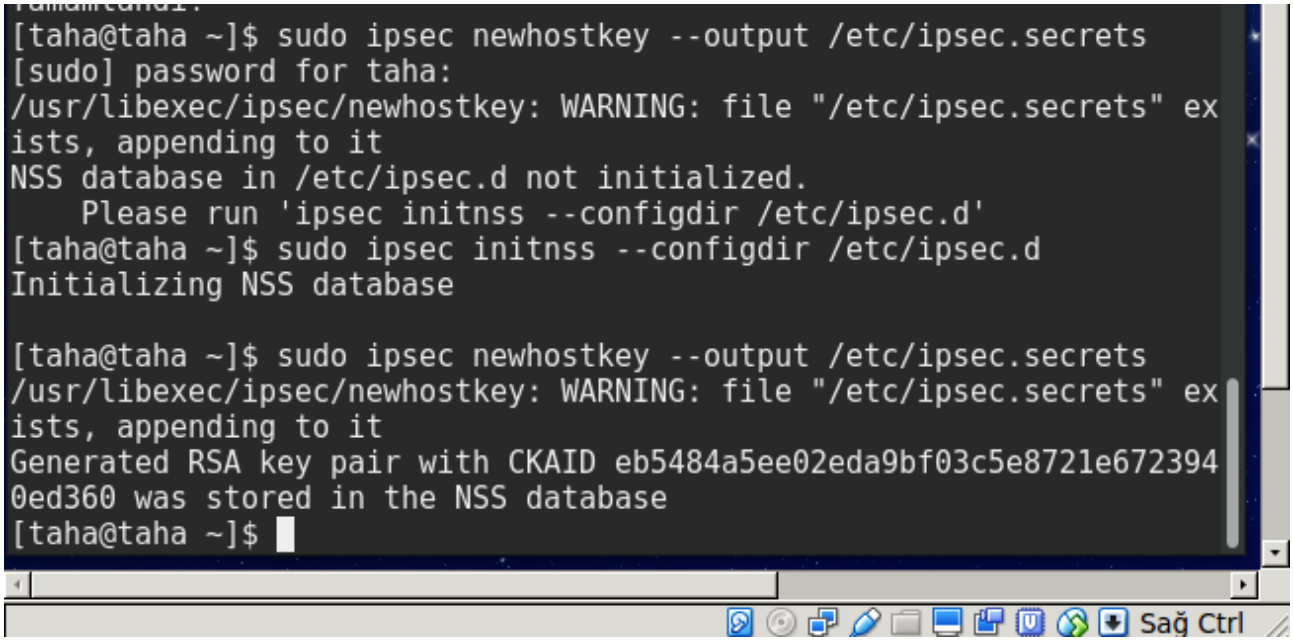


Figure 6

Step 5 : ipsec showhostkey --right –ckaid eb5484a5ee02eda9bf03c5e8721e6723940ed360
We saw rsa key wiht this command as seen as figure 7.
ckaid parameter is determined in the step 4.
We run this command because the rightrsasigkey value is used to configuretion of ipsec.conf file.

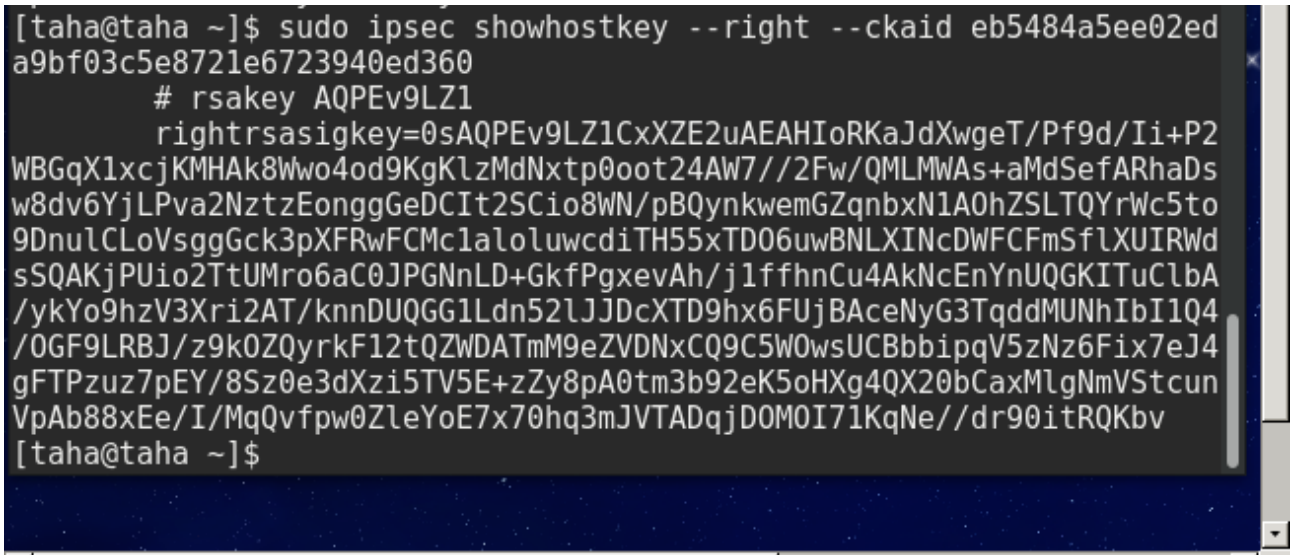


Figure 7

Step 6 : ipsec.conf file is configurated as step 6 in left hosts part and copy under /etc folder. We explain again ipsec.conf file because we done in left host part and its same.

Step 7 : systemctl start ipsec.service # start ipsec service as seen as figure 8
sudo ipsec auto –add hasanTahaTunnel # add new tunnel that name is hasanTahaTunnel
sudo ipsec suto –up hasanTahaTunnel # hasanTahaTunnel is activated as seen figure 8

We observed “IPSec SA established tunnel mode {ESP=....}” the last line as seen as figure 8.

```
[taha@taha ~]$ sudo ipsec setup start
Redirecting to: systemctl start ipsec.service
[taha@taha ~]$ systemctl start ipsec.service
[taha@taha ~]$ sudo ipsec setup start
systemd: ipsec service is already running
[taha@taha ~]$ sudo ipsec auto --add hasanTahaTunnel
002 "hasanTahaTunnel": deleting connection
002 "hasanTahaTunnel" #4: deleting state (STATE_QUICK_R2)
005 "hasanTahaTunnel" #4: ESP traffic information: in=0B out=0B
002 "hasanTahaTunnel" #2: deleting state (STATE_QUICK_I2)
005 "hasanTahaTunnel" #2: ESP traffic information: in=0B out=0B
002 "hasanTahaTunnel" #3: deleting state (STATE_MAIN_R3)
002 "hasanTahaTunnel" #1: deleting state (STATE_MAIN_I4)
002 added connection description "hasanTahaTunnel"
[taha@taha ~]$ sudo ipsec auto --up hasanTahaTunnel
002 "hasanTahaTunnel" #5: initiating Main Mode
104 "hasanTahaTunnel" #5: STATE_MAIN_I1: initiate
002 "hasanTahaTunnel" #5: transition from state STATE_MAIN_I1 to state STATE_MAIN_I2
106 "hasanTahaTunnel" #5: STATE_MAIN_I2: sent MI2, expecting MR2
002 "hasanTahaTunnel" #5: transition from state STATE_MAIN_I2 to state STATE_MAIN_I3
108 "hasanTahaTunnel" #5: STATE_MAIN_I3: sent MI3, expecting MR3
002 "hasanTahaTunnel" #5: Main mode peer ID is ID_FQDN: '@left'
002 "hasanTahaTunnel" #5: transition from state STATE_MAIN_I3 to state STATE_MAIN_I4
004 "hasanTahaTunnel" #5: STATE_MAIN_I4: ISAKMP SA established {auth=RSA_SIG cipher=a
es_256 integ=sha group=MODP2048}
002 "hasanTahaTunnel" #6: initiating Quick Mode RSASIG+ENCRYPT+TUNNEL+PFS+UP+IKEV1_AL
LOW+IKEV2_ALLOW+SAREF_TRACK+IKE_FRAG_ALLOW+ESN_NO {using isakmp#5 msgid:1f7953ac prop
osal=defaults pfsgroup=OAKLEY_GROUP_MODP2048}
117 "hasanTahaTunnel" #6: STATE_QUICK_I1: initiate
002 "hasanTahaTunnel" #6: transition from state STATE_QUICK_I1 to state STATE_QUICK_I
2
004 "hasanTahaTunnel" #6: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel mode
{ESP=>0xcd26d0e6 <0x81e2f59b xfrm=AES_128-HMAC_SHA1 NATOA=none NATD=none DPD=passive}
[taha@taha ~]$
```

Figure 8

Step 8 : ping 192.168.2.216 # ping to left host using ip address as seen as figure 9.
ping left.hasan.fedora # ping to left host using hostname as seen as figure 10.

We tested securily network traffic by ping to left host from right host and we observed that provide conneciton with ESP as seen as figure 11.

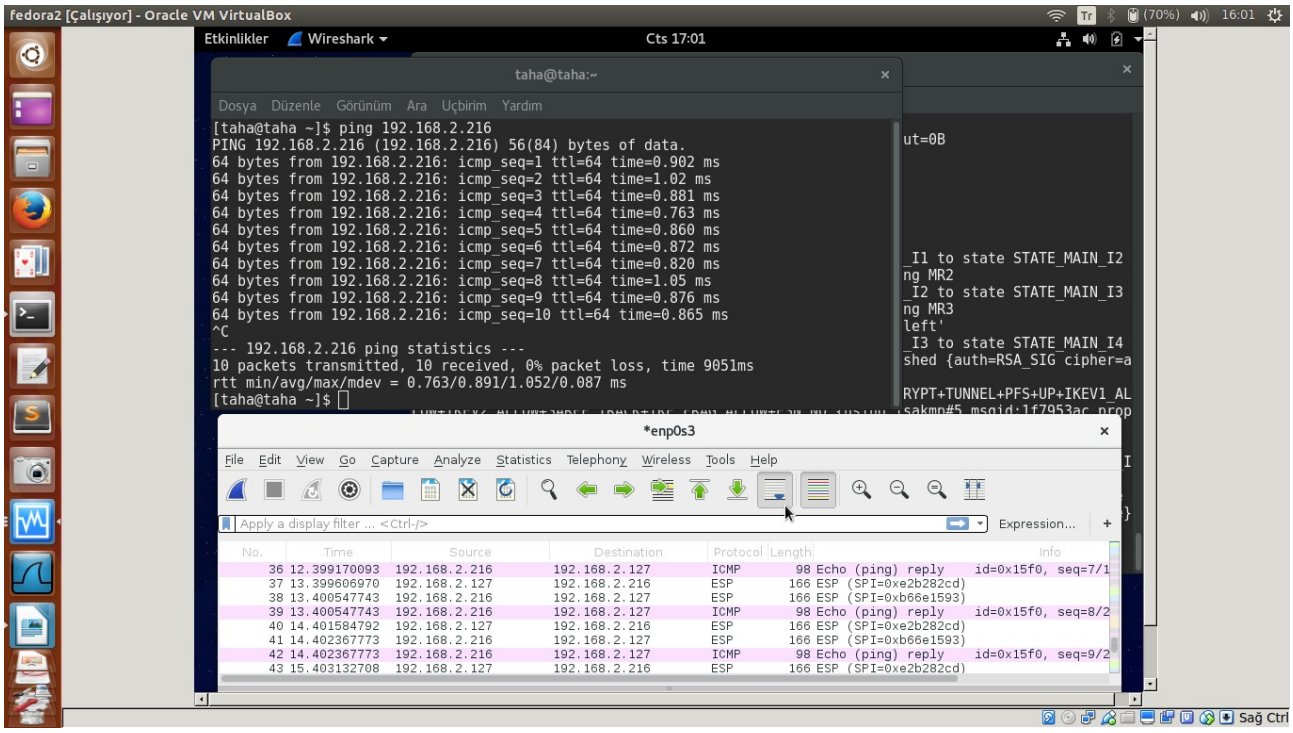


Figure 9

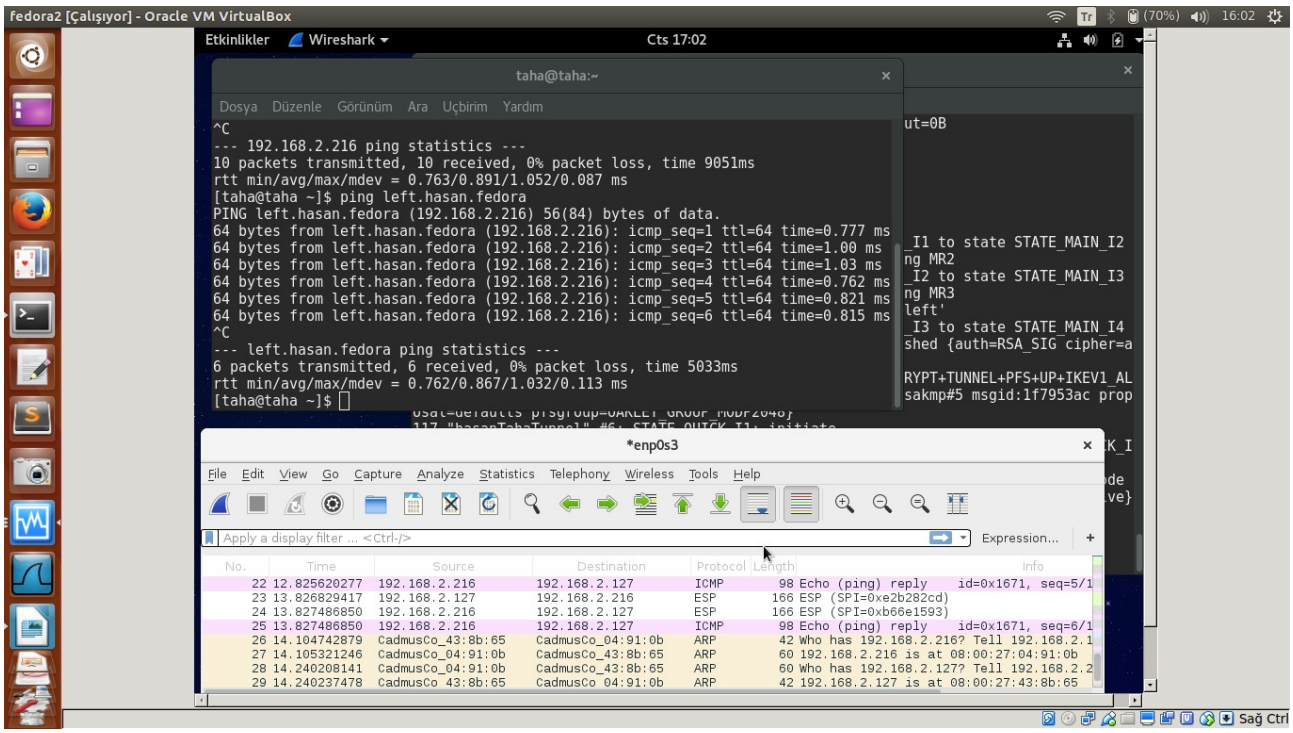


Figure 10

No.	Time	Source	Destination	Protocc	Lengl	Info
4	8.793622797	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
5	8.794318571	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
7	9.796550795	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
8	9.797442563	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
13	10.798663824	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
14	10.799575581	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
16	11.804929292	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
17	11.805592436	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
20	12.824891911	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
21	12.825620277	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)
23	13.826829417	192.168.2.127	192.168.2.216	ESP	166	ESP (SPI=0xe2b282cd)
24	13.827486850	192.168.2.216	192.168.2.127	ESP	166	ESP (SPI=0xb66e1593)

```
▶ Frame 4: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0
▶ Ethernet II, Src: PcsSyste_43:8b:65 (08:00:27:43:8b:65), Dst: PcsSyste_04:91:0b (08:00:27:04:91:0b)
▼ Internet Protocol Version 4, Src: 192.168.2.127, Dst: 192.168.2.216
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 152
    Identification: 0x3a99 (15001)
  ▶ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: Encap Security Payload (50)
    Header checksum: 0x78f3 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.2.127
    Destination: 192.168.2.216
    [Source GeoIP: Unknown]
```

Figure 11

References

http://askubuntu.com/questions/309443/how-to-allow-vms-and-host-ping-each-other
https://libreswan.org/man/
http://www.slashroot.in/linux-ipsec-site-site-vpnvirtual-private-network-configuration-using-openswan
http://linuxpitstop.com/install-openswan-and-create-site-to-site-vpn-on-centos-linux/
https://libreswan.org/wiki/Host_to_host_VPN
http://www.putorius.net/2012/08/creating-end-to-end-ipsec-tunnel.html
http://academlib.com/26730/computer_science/remote_access_host-to-site
http://www.hostbul.net/blog/209/vpn-nedir-nasil-kullanilir#.WGf_r-2W8x8
https://en.wikipedia.org/wiki/Virtual_private_network
http://vpn.sockslist.net/cgi-bin/vpn/virtual_private_network_vpn.html
https://en.wikipedia.org/wiki/IPsec
http://labrisnetworks.com/tr/blog-aginizdaki-guvenlik-aciklarini-kapatmak-icin-kucuk-tavsiyeler/
http://cenkercetin.com/ipsec-nedir-ne-is-yapar/
http://bidb.itu.edu.tr/seyirdefteri/blog/2013/09/07/ipsec-vpn-(internet-protocol-security-internet-protokol%C3%BC-g%C3%BCvenli%C4%9Fi)
https://libreswan.org/
https://www.virtualbox.org/manual/ch06.html
https://en.wikipedia.org/wiki/NAT_traversal