Otto-Friedrich-University Bamberg

Professorship for Computer Science,
Communication Services, Telecommunication,
Systems and Computer Networks

# Foundation of Internet Communication

## Assignment-03: NATting, Firewalls, and Virtual Private Networks

Submitted by:
**Group J**

Reem Eslam Mohamed Mekky Khalil
Shivasharan Reddy
Azar Ghadami
Reema Miranda
Manjunath B Marigoudar

Supervisor: Prof. Dr. Udo Krieger

Bamberg, June 21, 2020
Summer Term 2020

# Contents

# List of Figures

# Chapter 1

# Network Address Translation (NAT) and Firewall

## 1.1 Network Address Translation (NAT)

- There are several situations where we need address translation such as, a network which do not have sufficient public IP addresses want to connect with the Internet, two networks which have same IP addresses want to merge or due to security reason a network want to hide its internal IP structure from the external world. NAT (Network Address Translation) is the process which translates IP address. NAT can be performed at firewall, server and router.

- Network Address Translation allows a single device, such as a router, to act as an agent between the Internet (or "public network") and a local (or "private") network. This means that only a single, unique IP address is required to represent an entire group of computers.

- In this session we are configuring Natting and also firewall

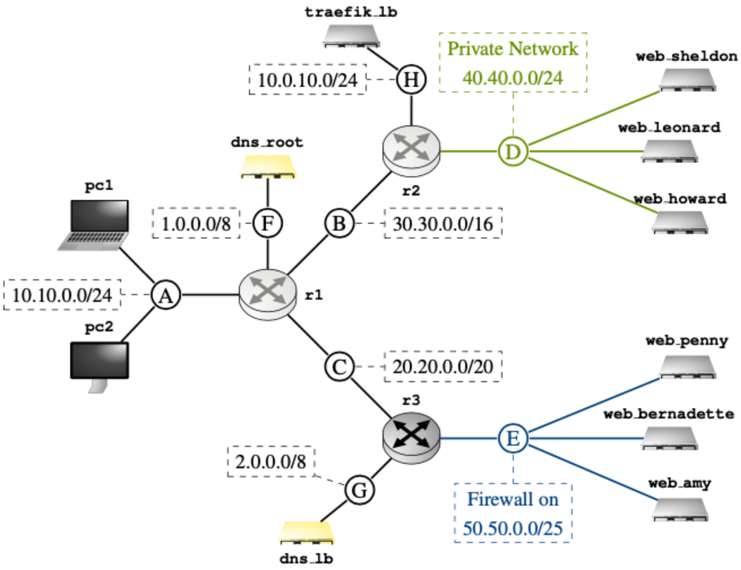
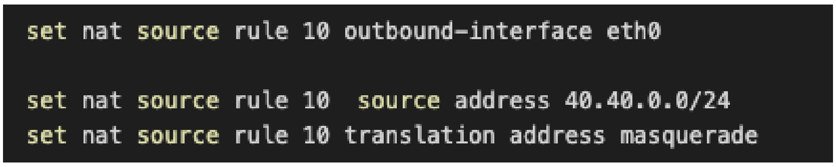


Figure 1.1: Nat and firewall experiment

### 1.1.1 Network Address Translation (NAT)

We consider our network D to be a private network and thus achieve that balancer serves as an entry-point to gik.de. Therefore, we change the configuration in the following way. We configure natting commands on router-2.

- To begin the natting we add a outbound interface command with Ethernet 0 ”set Nat source rule 10 outbound-interface eth0”
- for a source NAT with the gateway 40.40.0.2/24 in network D, we use the following command ”set Nat source rule 10 source address 40.40.0.0/24”
- After adding source NAT with the gateway 40.40.0.2/24 in network D, we have to masquerade domain D to reach domains B, A and F so we use ”set Nat source rule 10 translation address masquerade”

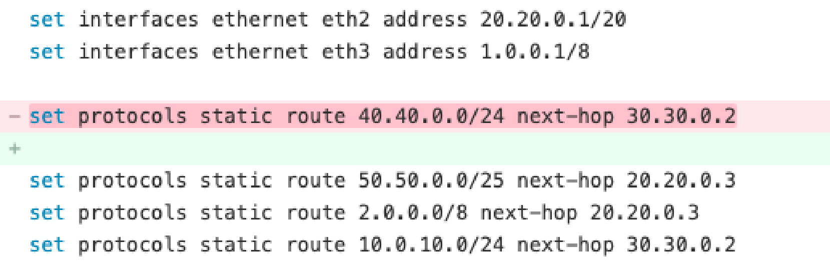


```
set nat source rule 10 outbound-interface eth0

set nat source rule 10  source address 40.40.0.0/24
set nat source rule 10 translation address masquerade
```

Figure 1.2: Natting commands on R2

### 1.1.2 On r1 and r3 remove the route to reach collision domain (CD) D

In this session we remove routing configuration from router-1 and router-3 so that we cannot reach collision domain D



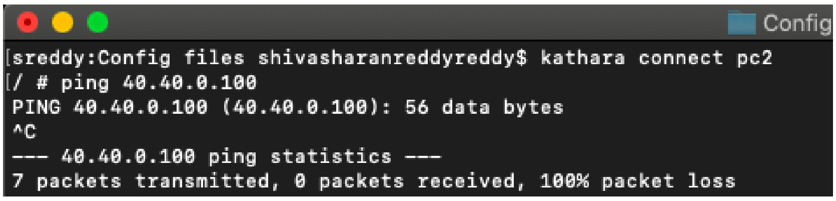
```
  set interfaces ethernet eth2 address 20.20.0.1/20
  set interfaces ethernet eth3 address 1.0.0.1/8

- set protocols static route 40.40.0.0/24 next-hop 30.30.0.2
+

  set protocols static route 50.50.0.0/25 next-hop 20.20.0.3
  set protocols static route 2.0.0.0/8 next-hop 20.20.0.3
  set protocols static route 10.0.10.0/24 next-hop 30.30.0.2
```

Figure 1.3: Removing route

### 1.1.3 From PC2 ensure that, e.g., web Sheldon is not pingable, but curl gik.de still delivers a result.

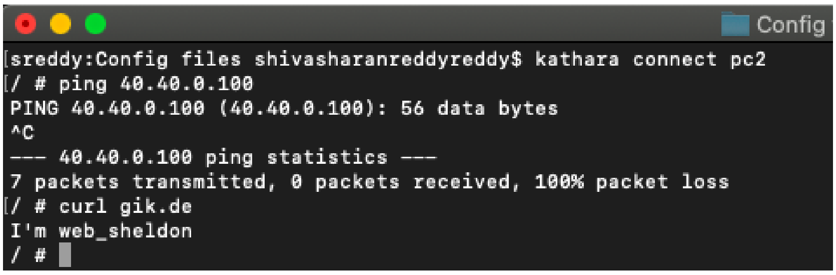
- In this session our aim is to ensure that, web Sheldon is not pingable, but curl gik.de still delivers a result

- If we try to ping Web Sheldon from Pc2 it is not reachable after removing the routes in R1 but curl gik.de still delivers a results .



```
[sreddy:Config files shivasharanreddyreddy$ kathara connect pc2
[/ # ping 40.40.0.100
 PING 40.40.0.100 (40.40.0.100): 56 data bytes
 ^C
 --- 40.40.0.100 ping statistics ---
 7 packets transmitted, 0 packets received, 100% packet loss
```

Figure 1.4: web Sheldon is not pingable



```
[sreddy:Config files shivasharanreddyreddy$ kathara connect pc2
[/ # ping 40.40.0.100
PING 40.40.0.100 (40.40.0.100): 56 data bytes
^C
--- 40.40.0.100 ping statistics ---
7 packets transmitted, 0 packets received, 100% packet loss
[/ # curl gik.de
I'm web_sheldon
/ #
```

Figure 1.5: web Sheldon is not pingable but curl gik.de still delivers a result

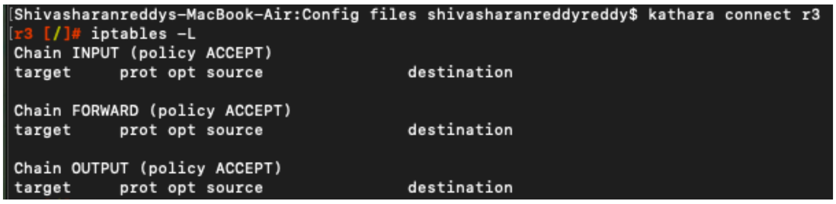
### 1.1.4 On CD B and CD D capture the call curl gik.de with wire shark and explain the behaviour.

- In this session we try to capture the packets on wire-shark by using curl gik.de see weather the natting we configured is working properly.
- we have added natting on the collion D , hence we can observe the the router acts as a agent between the three web server's and other collision domain so This means that only a single, unique IP address is required to represent an entire group of computers.
- Hence, all the three web servers, web-Sheldon, web-Howard and web-Leonard ip's will be mapped to the IP 40.40.0.2

## 1.2 Firewall

In this session we configure firewall On network E, we want to restrict the access to our servers and drop all packets except the ones, which establish tcp connections on port 80. Therefore, we need to enable a firewall with iptables on r3 Before we move further we run iptables on the router 3 and look for the output We use below command to check the firewall rules.

- iptables -L



```
[Shivasharanreddys-MacBook-Air:Config files shivasharanreddyreddy$ kathara connect r3
[r3 [/]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

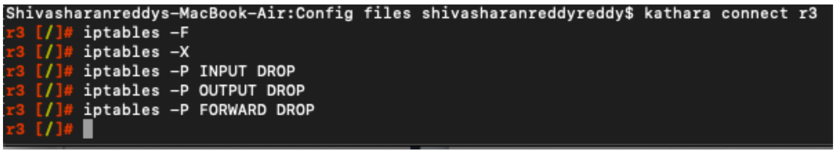
Figure 1.6: R3 before applying firewall rules

### 1.2.1 we need to enable a firewall with iptables on r3.

#### (A) Create a default filter policy to drop all packets

To drop all the packets entering network we use set of firewall iptables commands listed below on r3.

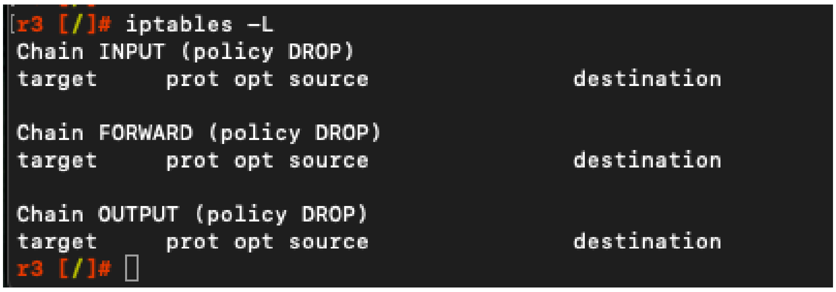
- iptables -F
- iptables -X
- iptables -P INPUT DROP
- iptables -P OUTPUT DROP
- iptables -P FORWARD DROP



```
Shivasharanreddys-MacBook-Air:Config files shivasharanreddyreddy$ kathara connect r3
r3 [/]# iptables -F
r3 [/]# iptables -X
r3 [/]# iptables -P INPUT DROP
r3 [/]# iptables -P OUTPUT DROP
r3 [/]# iptables -P FORWARD DROP
r3 [/]#
```

Figure 1.7: default filter policy to drop all packets

So, after we configure these commands on R3 we have to use iptables -L on R3 and check if all the packets are been dropped. We can see in the below figure that all the packets are dropping.



```
r3 [/]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
r3 [/]#
```

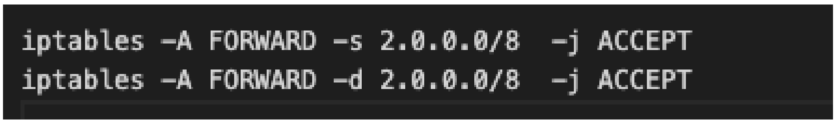
Figure 1.8: Packets are dropped

#### (B) Allow unlimited traffic on the loop-back interface lo again.

To allow unlimited traffic on the loop-back interface lo we use following set of commands. We have two commands for loop back interface, one for input and another for output.

- iptables -A INPUT -i lo -j ACCEPT

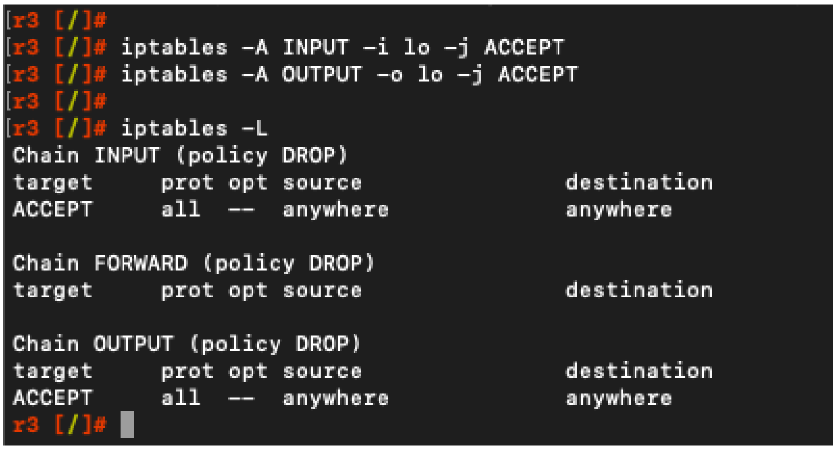
- iptables -A OUTPUT -o lo -j ACCEPT



```
iptables -A FORWARD -s 2.0.0.0/8  -j ACCEPT
iptables -A FORWARD -d 2.0.0.0/8  -j ACCEPT
```

Figure 1.9: Loop back interface firewall commands

After we configure these commands on r3 we use iptables -L to check if the firewall is allowing unlimited traffic on the loop back interface lo.



```
[r3 [/]#
[r3 [/]# iptables -A INPUT -i lo -j ACCEPT
[r3 [/]# iptables -A OUTPUT -o lo -j ACCEPT
[r3 [/]#
[r3 [/]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
r3 [/]#
```

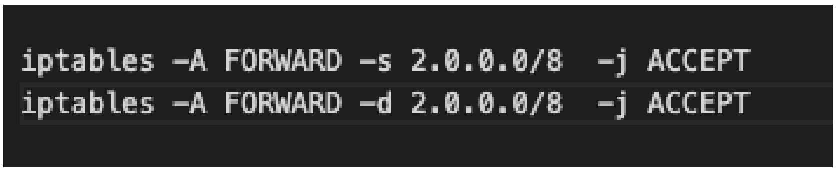
Figure 1.10: Allowing traffic on lo interface

### (C) Forward packets of the DNS network 2.0.0.0/8 without any restrictions.

To forward packets of the DNS network 2.0.0.0/8 without any restrictions we configure two commands on r3, one for source address and another for Destination.

- iptables -A FORWARD -s 2.0.0.0/8 -j ACCEPT

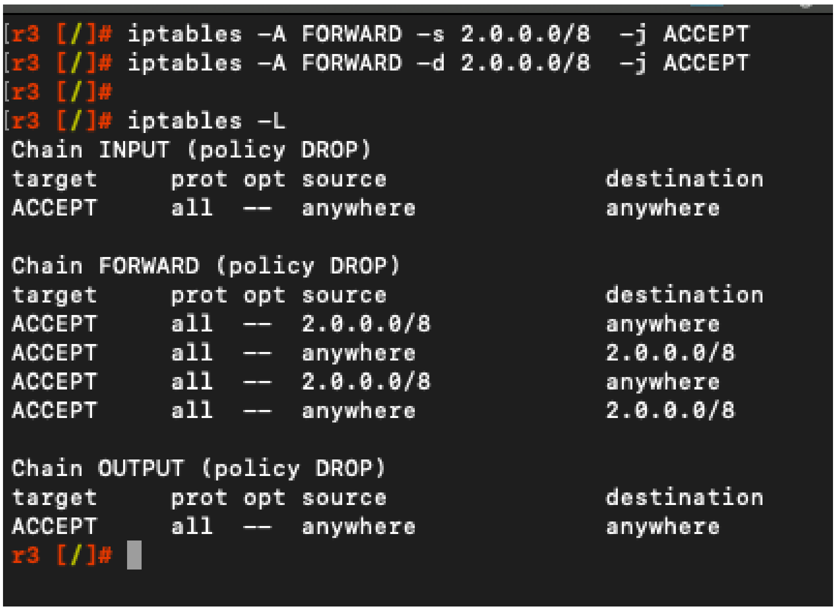
- iptables -A FORWARD -d 2.0.0.0/8 -j ACCEPT



```
iptables -A FORWARD -s 2.0.0.0/8  -j ACCEPT
iptables -A FORWARD -d 2.0.0.0/8  -j ACCEPT
```

Figure 1.11: Firewall commands -03

After we configure these commands on r3 we use iptables -L to check if the firewall is Forwarding packets of the DNS network 2.0.0.0/8 without any restrictions.



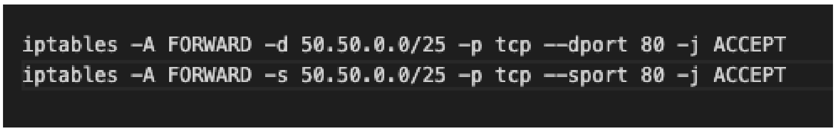
```
[r3 [/]# iptables -A FORWARD -s 2.0.0.0/8  -j ACCEPT
[r3 [/]# iptables -A FORWARD -d 2.0.0.0/8  -j ACCEPT
[r3 [/]#
[r3 [/]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  2.0.0.0/8            anywhere
ACCEPT     all  --  anywhere             2.0.0.0/8
ACCEPT     all  --  2.0.0.0/8            anywhere
ACCEPT     all  --  anywhere             2.0.0.0/8

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
r3 [/]#
```

Figure 1.12: Forward packets of the DNS network

### (D) Allow all web servers in CD E to accept connections on tcp port 80.

To Allow all web servers in CD E to accept connections on tcp port 80, we configure two commands on r3, one for source address and another for Destination.

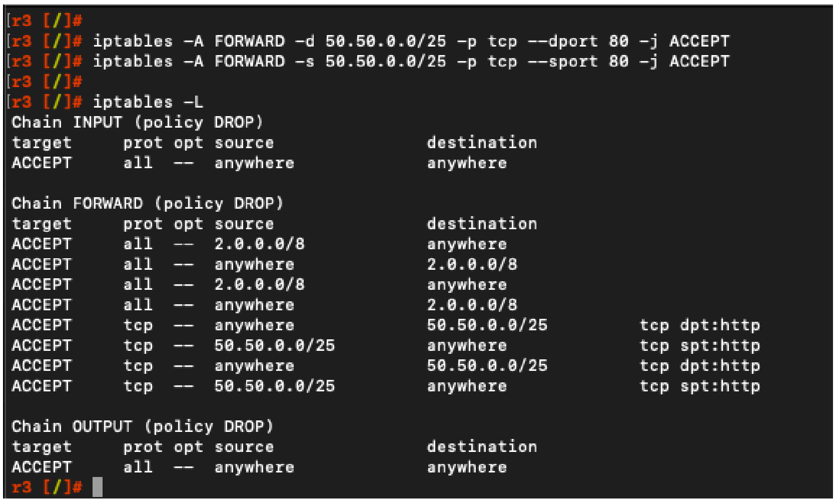
- iptables -A FORWARD -d 50.50.0.0/25 -p tcp –dport 80 -j ACCEPT
- iptables -A FORWARD -s 50.50.0.0/25 -p tcp –sport 80 -j ACCEPT



```
iptables -A FORWARD -d 50.50.0.0/25 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -s 50.50.0.0/25 -p tcp --sport 80 -j ACCEPT
```

Figure 1.13: Firewall commands-04

After we configure these commands on r3 we use iptables -L to check if the firewall is Allowing all web servers in CD E to accept connections on tcp port 80.

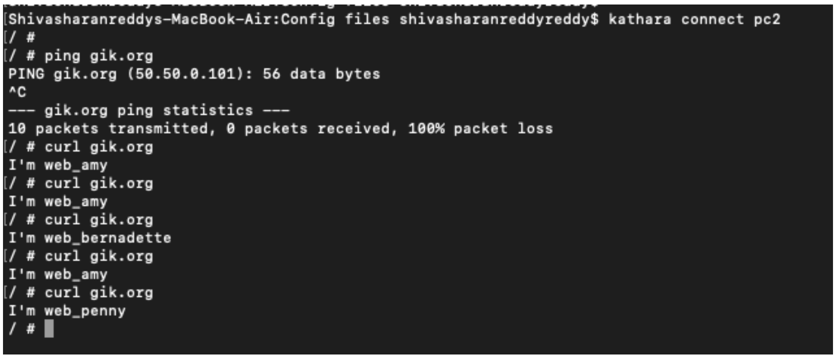


```
[r3 [/]#
[r3 [/]# iptables -A FORWARD -d 50.50.0.0/25 -p tcp --dport 80 -j ACCEPT
[r3 [/]# iptables -A FORWARD -s 50.50.0.0/25 -p tcp --sport 80 -j ACCEPT
[r3 [/]#
[r3 [/]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  2.0.0.0/8            anywhere
ACCEPT     all  --  anywhere             2.0.0.0/8
ACCEPT     all  --  2.0.0.0/8            anywhere
ACCEPT     all  --  anywhere             2.0.0.0/8
ACCEPT     tcp  --  anywhere             50.50.0.0/25         tcp dpt:http
ACCEPT     tcp  --  50.50.0.0/25         anywhere             tcp spt:http
ACCEPT     tcp  --  anywhere             50.50.0.0/25         tcp dpt:http
ACCEPT     tcp  --  50.50.0.0/25         anywhere             tcp spt:http

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
r3 [/]#
```

Figure 1.14: Allowing all web servers in CD E to accept connections on tcp port 80

### 1.2.2 On PC2 ensure that ping gik.org fails, but curl gik.org displays the website content.

After all the configuration on r3 we need to connect PC1 and check if it full-fills all requirements, ping gik.org should fail but curl gik.org displays the website content. As we can see ping gik.org is failing on PC1 as we configured in firewall rules but curl gik.org should should display website content as shown below.



```
[Shivasharanreddys-MacBook-Air:Config files shivasharanreddyreddy$ kathara connect pc2
[/ #
[/ # ping gik.org
 PING gik.org (50.50.0.101): 56 data bytes
 ^C
 --- gik.org ping statistics ---
 10 packets transmitted, 0 packets received, 100% packet loss
[/ # curl gik.org
 I'm web_amy
[/ # curl gik.org
 I'm web_amy
[/ # curl gik.org
 I'm web_bernadette
[/ # curl gik.org
 I'm web_amy
[/ # curl gik.org
 I'm web_penny
 / #
```

Figure 1.15: ping gik.org is failing but curl gik.de is displaying website content on pc2

# Chapter 2

# Virtual Private Network (VPN)

- We use Soft-Ether Virtual Private network in order to provide secure connection to other network over internet

Unfortunately, our administrator, who currently works at home, needs unrestricted remote access to the servers. Since we do not want to get rid of our security improvements, we setup a VPN to achieve that goal.

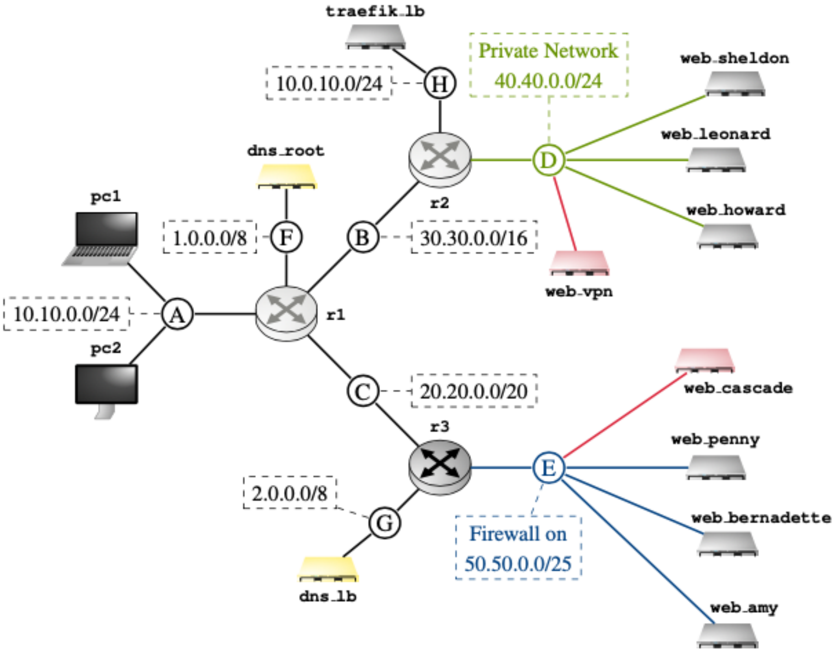


Figure 2.1: VPN configuration

## 2.1 Softether Server Client

First, we want to create a remote access VPN to give our administrator, who uses pc1, unrestricted access to the servers in our private network D. If she is connected to the VPN server, her network topology looks like the one of Figure 3 and her machine seems to be in the same local area network (LAN).

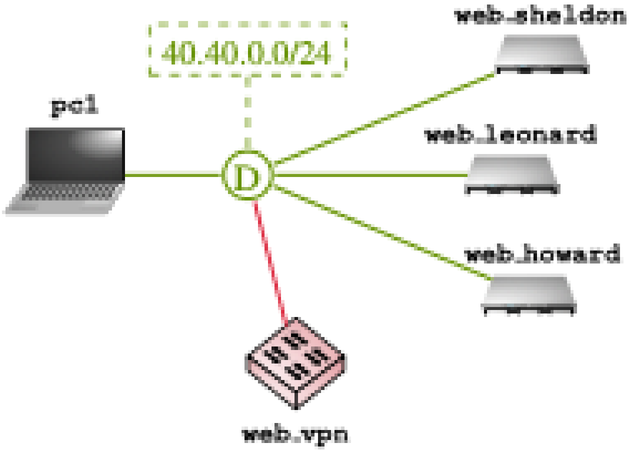


Figure 2.2: Topology appearance after a connection to the VPN is established

### 2.1.1 To achieve the described behaviour, we create a Softether server, called web vpn in the topology of Figure

We use following commands to create a web-vpn

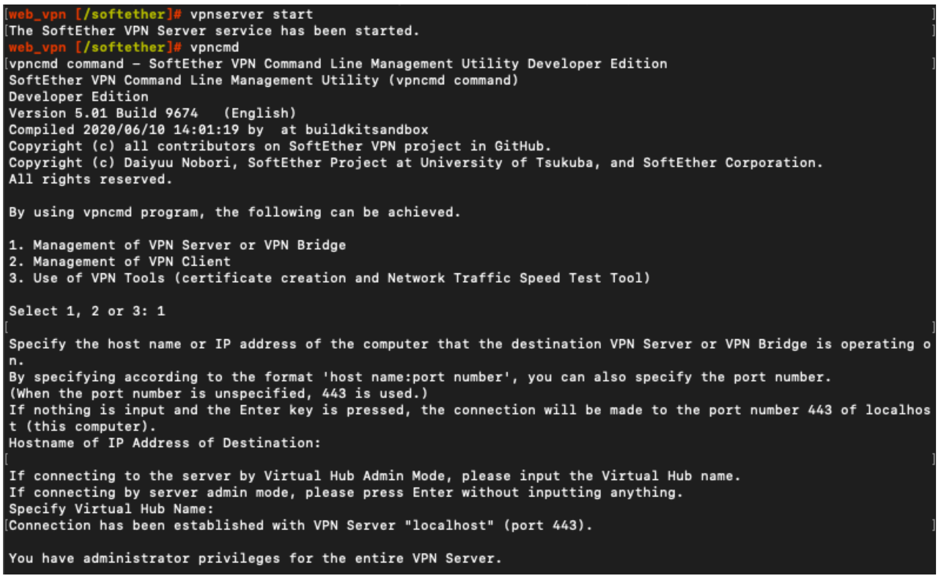
- ip addr add 40.40.0.99/24 brd + dev eth0
- ip route add default via 40.40.0.2 dev eth0

After we configure the ip address and the routing commands we move on to the next part.

### 2.1.2 (1) Our VPN server is started with the command vpnserver start and can be configured by vpncmd.

We use the following commands to start the vpn server, and configure the vpn.

- vpnserver start - This command helps us to start the vpn server
- vpncmd - This command helps us to configure on Vpn



```
web_vpn [/softether]# vpnserver start
The SoftEther VPN Server service has been started.
web_vpn [/softether]# vpncmd
vpncmd command - SoftEther VPN Command Line Management Utility Developer Edition
SoftEther VPN Command Line Management Utility (vpncmd command)
Developer Edition
Version 5.01 Build 9674   (English)
Compiled 2020/06/10 14:01:19 by  at buildkitsandbox
Copyright (c) all contributors on SoftEther VPN project in GitHub.
Copyright (c) Daiyuu Nobori, SoftEther Project at University of Tsukuba, and SoftEther Corporation.
All rights reserved.

By using vpncmd program, the following can be achieved.

1. Management of VPN Server or VPN Bridge
2. Management of VPN Client
3. Use of VPN Tools (certificate creation and Network Traffic Speed Test Tool)

Select 1, 2 or 3: 1

Specify the host name or IP address of the computer that the destination VPN Server or VPN Bridge is operating o
n.
By specifying according to the format 'host name:port number', you can also specify the port number.
(When the port number is unspecified, 443 is used.)
If nothing is input and the Enter key is pressed, the connection will be made to the port number 443 of localhos
t (this computer).
Hostname of IP Address of Destination:

If connecting to the server by Virtual Hub Admin Mode, please input the Virtual Hub name.
If connecting by server admin mode, please press Enter without inputting anything.
Specify Virtual Hub Name:
Connection has been established with VPN Server "localhost" (port 443).

You have administrator privileges for the entire VPN Server.
```

Figure 2.3: Vpn start

- When we use vpnserver start, as we can see in the above screen shot The SoftEther VPN Server service has been started.
- After starting the server we have to make configurations on the vpn hence By using vpncmd program the following can be achieved.
- (a) Management of VPN Server or VPN Bridge
- (b) Management of VPN Client
- (c) Use of VPN Tools (certificate creation and Network Traffic Speed Test Tool)

### 2.1.3 First we create a hub and append a tap device called intern to it

Virtual Hub is a virtual network that helps us to connect to other resources. Ex:To Connect vpnclient(PC1).

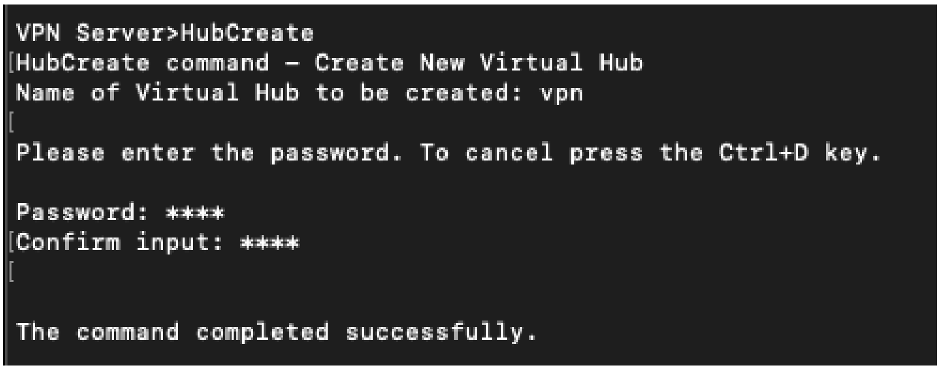
For creating a hub we need to enter Management of VPN server or VPN bridge hence we select first option from list of vpncmd options. we use the following commands to create a hub

- HubCreate

once we enter Hubcreate it will ask for a Hub name hence we specify hub name as ”vpn”

- Hub name:vpn

Once we specify the hub name the server asks for a password, we then confirm the password, Then the Hub is successfully created as shown in the below screenshot.



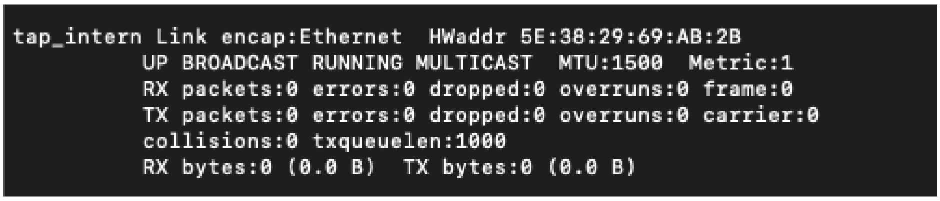
```
VPN Server>HubCreate
[HubCreate command - Create New Virtual Hub
Name of Virtual Hub to be created: vpn
[
Please enter the password. To cancel press the Ctrl+D key.

Password: ****
[Confirm input: ****
[

The command completed successfully.
```

Figure 2.4: Hub creation

Once the hub is created now we have to append a tap device called intern to it. we can do this by creating a bridge, we use the following command to append a tap device called intern

- ”BridgeCreate VPN /DEVICE:intern /TAP:yes”

and we can check it with ifconfig if the device is appended to it as shown below,

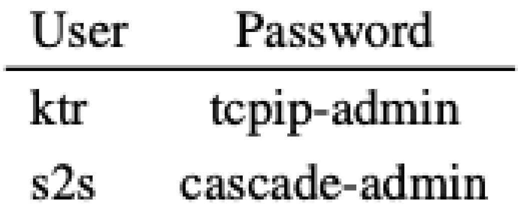


```
tap_intern Link encap:Ethernet  HWaddr 5E:38:29:69:AB:2B
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figure 2.5: append a tap device called intern

### 2.1.4 On the hub, we add the users of Table

Once the hub is created we have to add users to it as mention in the below table.



| User | Password |
|---|---|
| ktr | tcpip-admin |
| s2s | cascade-admin |

Figure 2.6: Users table

For creating users we have few set of command.

- UserCreate

Once you give this command server asks for few details regarding user such as,

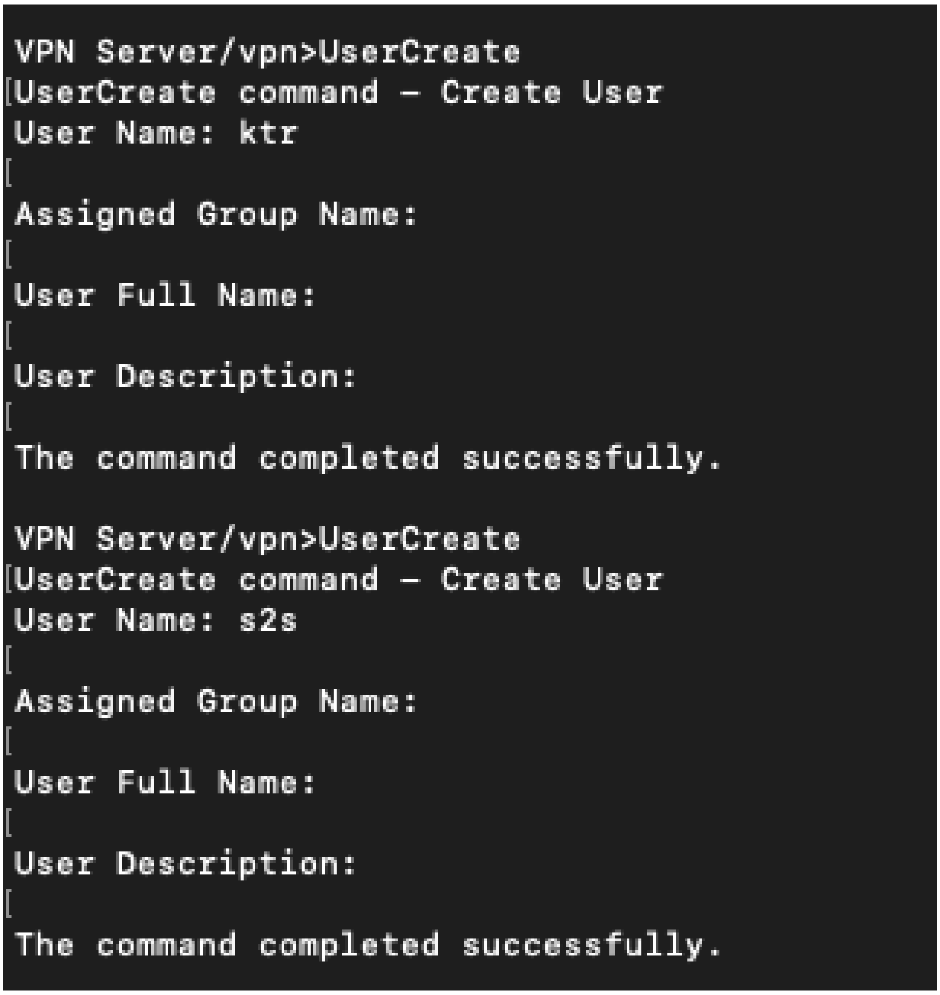
- User Name:

- Assigned Group Name:

- User Full Name:

- User Description:

After we specify all the given details we create a user names ”ktr” successfully as shown below.



```
VPN Server/vpn>UserCreate
[UserCreate command - Create User
User Name: ktr
[
Assigned Group Name:
[
User Full Name:
[
User Description:
[
The command completed successfully.

VPN Server/vpn>UserCreate
[UserCreate command - Create User
User Name: s2s
[
Assigned Group Name:
[
User Full Name:
[
User Description:
[
The command completed successfully.
```

Figure 2.7: Users creation

### 2.1.5 A new network interface card (NIC) tap intern was created, which needs to be connected to eth0 by a bridge.

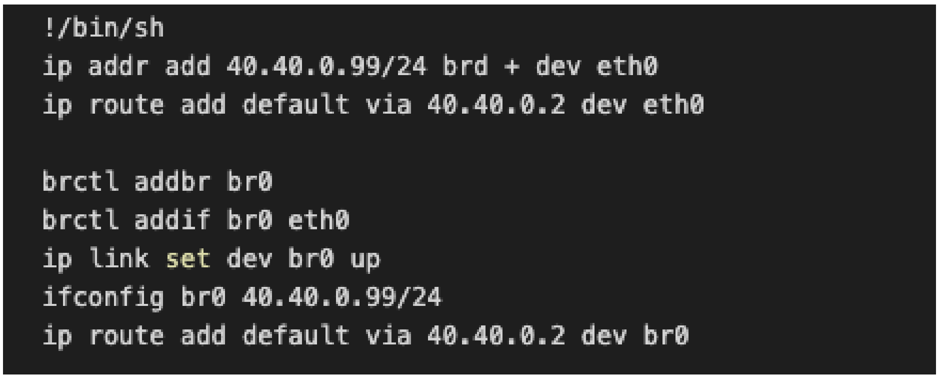
Now in this section we create a bridge hence we need a set of configurations on web vpn,

- brctl addbr br0
- brctl addif br0 eth0
- ip link set dev br0 up

- ifconfig br0 40.40.0.99/24

- ip route add default via 40.40.0.2 dev br0

Whit above mention configurations on web vpn we can create a bridge.



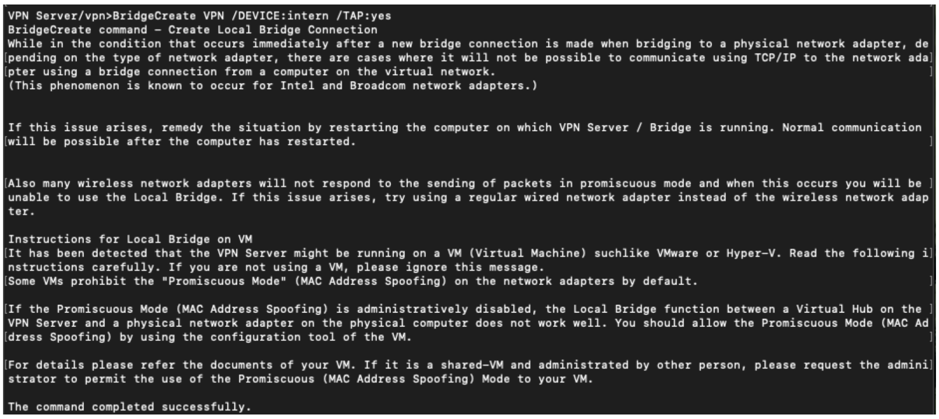
```
!/bin/sh
ip addr add 40.40.0.99/24 brd + dev eth0
ip route add default via 40.40.0.2 dev eth0

brctl addbr br0
brctl addif br0 eth0
ip link set dev br0 up
ifconfig br0 40.40.0.99/24
ip route add default via 40.40.0.2 dev br0
```

Figure 2.8: Bridge configurations on web-vpn

And now after we add configurations on web-vpn we can test if we can create a bridge buy the following commands mentioned below,

- BridgeCreate VPN /DEVICE:intern /TAP:yes

Once we run the above command we can successfully create a bridge as shown below.



```
VPN Server/vpn>BridgeCreate VPN /DEVICE:intern /TAP:yes
BridgeCreate command - Create Local Bridge Connection
While in the condition that occurs immediately after a new bridge connection is made when bridging to a physical network adapter, de
[pending on the type of network adapter, there are cases where it will not be possible to communicate using TCP/IP to the network ada]
[pter using a bridge connection from a computer on the virtual network.                                                              ]
(This phenomenon is known to occur for Intel and Broadcom network adapters.)


If this issue arises, remedy the situation by restarting the computer on which VPN Server / Bridge is running. Normal communication
[will be possible after the computer has restarted.                                                                                  ]


[Also many wireless network adapters will not respond to the sending of packets in promiscuous mode and when this occurs you will be ]
unable to use the Local Bridge. If this issue arises, try using a regular wired network adapter instead of the wireless network adap
ter.

Instructions for Local Bridge on VM
[It has been detected that the VPN Server might be running on a VM (Virtual Machine) suchlike VMware or Hyper-V. Read the following i]
nstructions carefully. If you are not using a VM, please ignore this message.
[Some VMs prohibit the "Promiscuous Mode" (MAC Address Spoofing) on the network adapters by default.                                 ]

[If the Promiscuous Mode (MAC Address Spoofing) is administratively disabled, the Local Bridge function between a Virtual Hub on the ]
VPN Server and a physical network adapter on the physical computer does not work well. You should allow the Promiscuous Mode (MAC Ad
[dress Spoofing) by using the configuration tool of the VM.                                                                          ]

[For details please refer the documents of your VM. If it is a shared-VM and administrated by other person, please request the admini]
strator to permit the use of the Promiscuous (MAC Address Spoofing) Mode to your VM.

The command completed successfully.
```

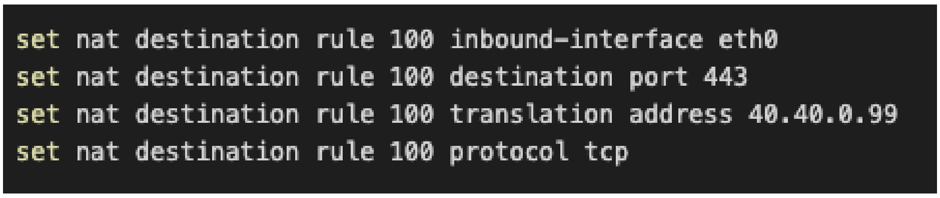
Figure 2.9: Bridge creation

### 2.1.6 Unfortunately, web vpn is inside a NAT and not reachable for our administrator on pc1. To facilitate this, we need to open tcp port 443 on r2 to forward traffic from it's eth0 interface to web vpn.

When web VPN is inside a nat and not rechable for our administrator on PC1, we need to open a TCP port 443 on r2 to forward traffic, this can be achieved by adding configurations on router-2 as shown bellow,

- set nat destination rule 100 inbound-interface eth0
- set nat destination rule 100 destination port 443
- set nat destination rule 100 translation address 40.40.0.99
- set nat destination rule 100 protocol tcp

With the above mentioned configuration's we will be able to forword traffic from eth0 of router to web vpn.



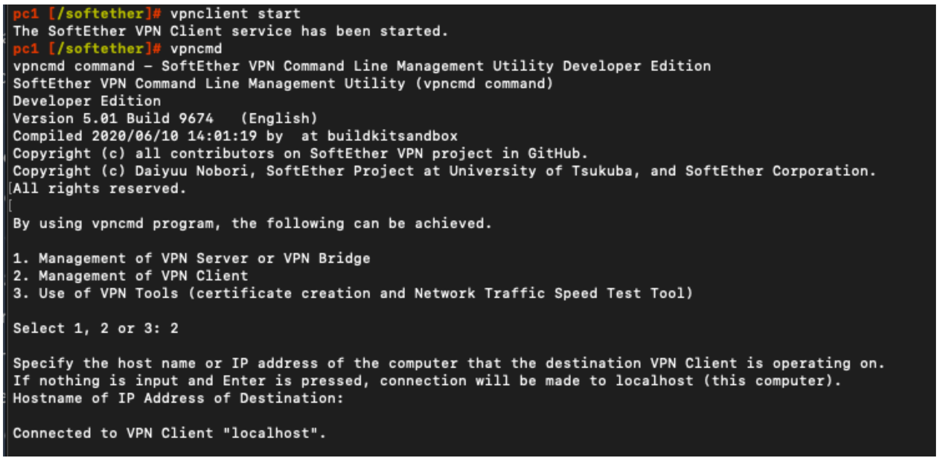
```
set nat destination rule 100 inbound-interface eth0
set nat destination rule 100 destination port 443
set nat destination rule 100 translation address 40.40.0.99
set nat destination rule 100 protocol tcp
```

Figure 2.10: R2 config

### 2.1.7 Now let us try, if our administrator can connect from pc1 by, starting a VPN client with vpnclient start on it.

In this session we connect tp PC1 and try to start vpn client on it by using the command

- vpnclient start
- vpncmd



```
pc1 [/softether]# vpnclient start
The SoftEther VPN Client service has been started.
pc1 [/softether]# vpncmd
vpncmd command - SoftEther VPN Command Line Management Utility Developer Edition
SoftEther VPN Command Line Management Utility (vpncmd command)
Developer Edition
Version 5.01 Build 9674   (English)
Compiled 2020/06/10 14:01:19 by  at buildkitsandbox
Copyright (c) all contributors on SoftEther VPN project in GitHub.
Copyright (c) Daiyuu Nobori, SoftEther Project at University of Tsukuba, and SoftEther Corporation.
All rights reserved.

By using vpncmd program, the following can be achieved.

1. Management of VPN Server or VPN Bridge
2. Management of VPN Client
3. Use of VPN Tools (certificate creation and Network Traffic Speed Test Tool)

Select 1, 2 or 3: 2

Specify the host name or IP address of the computer that the destination VPN Client is operating on.
If nothing is input and Enter is pressed, connection will be made to localhost (this computer).
Hostname of IP Address of Destination:

Connected to VPN Client "localhost".
```

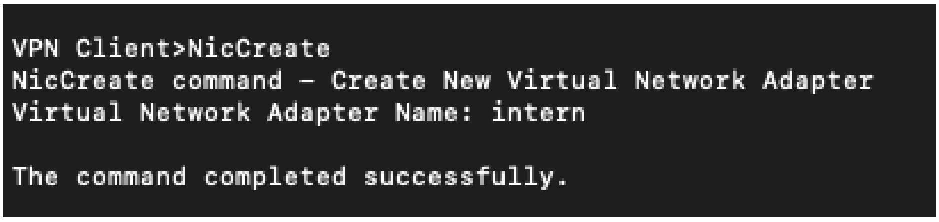
Figure 2.11: Starting vpn client on PC1

After connecting vpn client on pc1 we choose Management of VPN Client

**(a and b) We add a virtual interface called intern and create an account for the user ktr, which connects to web vpn via r2.**

In this session after we connect to VPN client on pc1 and we enter into vpncmd, now we have to add a virtual interface called intern, we can do this by following commands.

- NicCreate
- Virtual Network Adapter Name: intern



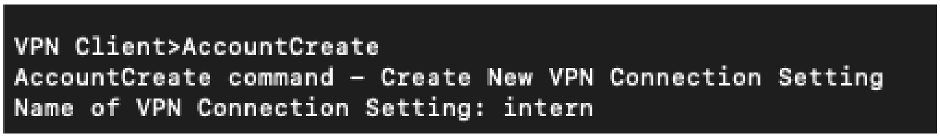
```
VPN Client>NicCreate
NicCreate command - Create New Virtual Network Adapter
Virtual Network Adapter Name: intern

The command completed successfully.
```

Figure 2.12: Adding a virtual interface called intern

### 2.1.8 create an account for the user ktr, which connects to web vpn via r2

In this session we create a account for the user ktr which connects to web vpn via r2, for thatwe use the following commands.

- VPN Client¿AccountCreate
- Name of VPN Connection Setting: intern



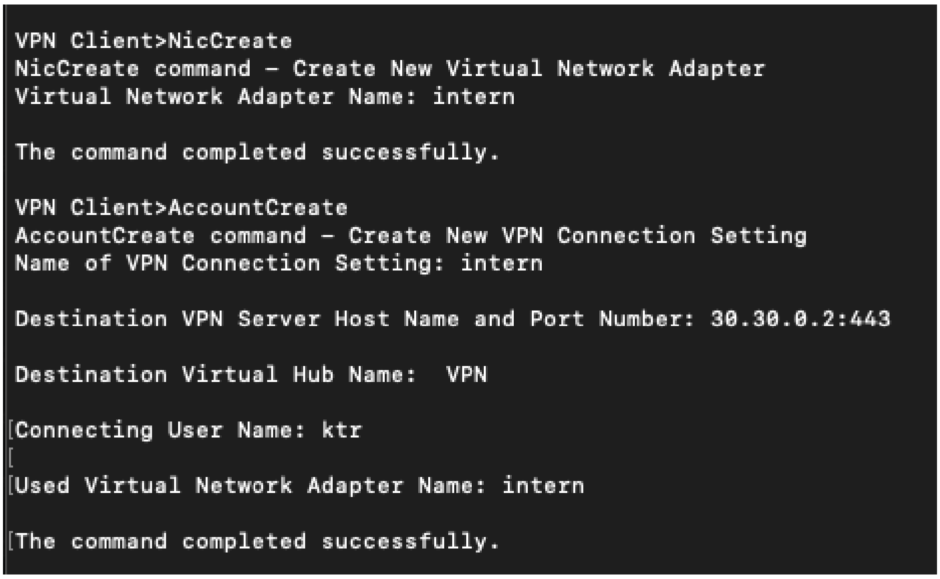
```
VPN Client>AccountCreate
AccountCreate command - Create New VPN Connection Setting
Name of VPN Connection Setting: intern
```

Figure 2.13: Creating an account for ktr

### 2.1.9 If the connection succeeds, we can add a private IP address to the newly created NIC vpn intern, which connects to the network

In this session we add a private IP address to the newly created NIC vpn intern which connects to the network

- Destination VPN Server Host Name and Port Number: 30.30.0.2:443
- Destination Virtual Hub Name:VPN
- Connecting User Name: ktr
- Used Virtual Network Adapter Name: intern



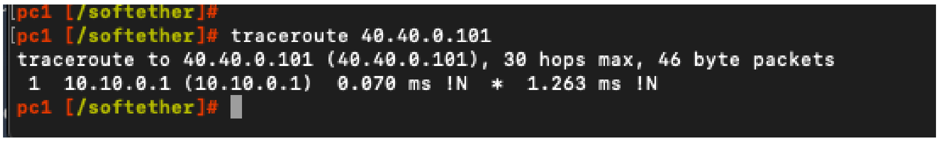
```
VPN Client>NicCreate
NicCreate command - Create New Virtual Network Adapter
Virtual Network Adapter Name: intern

The command completed successfully.

VPN Client>AccountCreate
AccountCreate command - Create New VPN Connection Setting
Name of VPN Connection Setting: intern

Destination VPN Server Host Name and Port Number: 30.30.0.2:443

Destination Virtual Hub Name:  VPN

[Connecting User Name: ktr
[
[Used Virtual Network Adapter Name: intern

[The command completed successfully.
```

Figure 2.14: add a private IP address to the newly created NIC vpn intern

### 2.1.10 On PC1 we start a trace-route to any web server of CD D and confirm that our target is only one hop away.

Now we try trace route to any of the web server of CD d and confirm that our target is only one hop away. we use the following command to do that

- trace-route 40.40.0.100



```
[pc1 [/softether]#
[pc1 [/softether]# traceroute 40.40.0.101
traceroute to 40.40.0.101 (40.40.0.101), 30 hops max, 46 byte packets
 1  10.10.0.1 (10.10.0.1)  0.070 ms !N  *  1.263 ms !N
pc1 [/softether]#
```

Figure 2.15: Trace-route to web sheldon

## 2.2 SoftEther Cascade Connections

VPN cascading is multi VPN connection.A further advantage of the cascade is that we can enable every web server to communicate with each other. So if we would replace our simple web servers with enhanced applications, which may consist of several microservices, they could securely communicate and use the resources of both sites. within VPN connection.

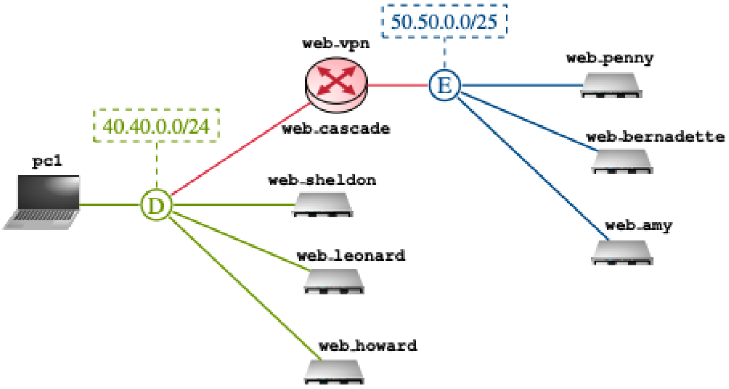
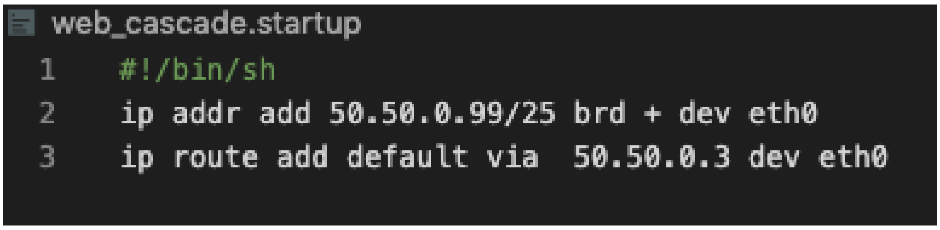


Figure 2.16: Topology appearance after a connection to the site-to-site VPN is established

- Finally, we connect both server farms over a site-to-site (s2s) VPN with a cascade connection. Our administrator can connect to both sites over a single VPN connection and reach all nodes without any restrictions.

### 2.2.1 create a Softether server, called web cascade

- web cascade is created as machine in the network.

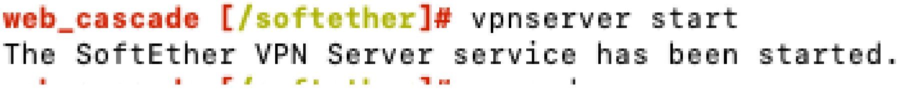


```
#!/bin/sh
ip addr add 50.50.0.99/25 brd + dev eth0
ip route add default via  50.50.0.3 dev eth0
```

Figure 2.17: Web Cascade startup file

1. **Start the web cascade**

- vpnserver start
- vpncmd

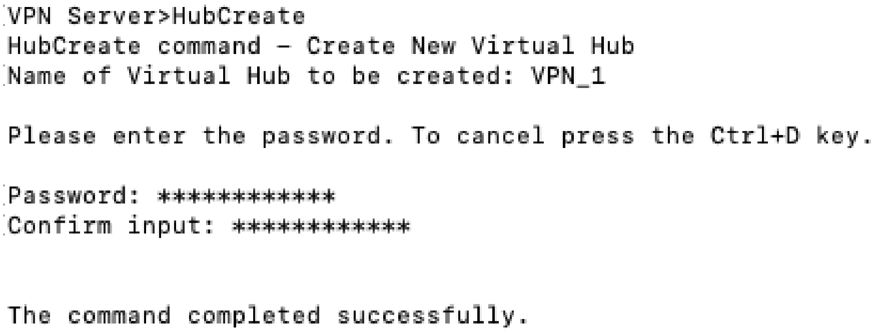


```
web_cascade [/softether]# vpnserver start
The SoftEther VPN Server service has been started.
```

Figure 2.18: Start the Web cascade

**2.Create a Hub**

- HubCreate

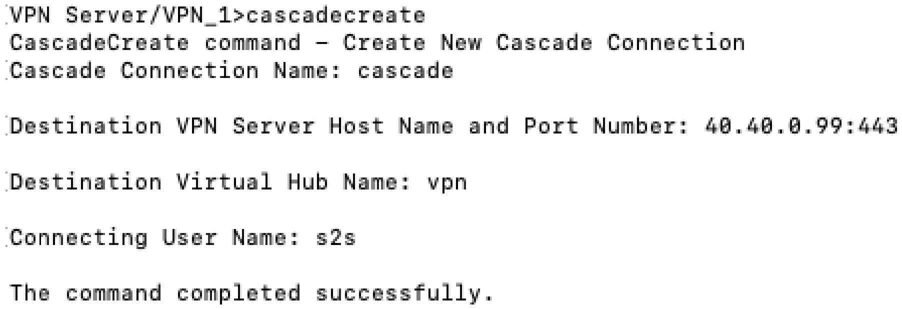


```
VPN Server>HubCreate
HubCreate command - Create New Virtual Hub
Name of Virtual Hub to be created: VPN_1

Please enter the password. To cancel press the Ctrl+D key.

Password: ************
Confirm input: ************


The command completed successfully.
```

Figure 2.19: Start the Web_cascade

**3.On the created hub we add a cascade, which connects to the remote hub on web_vpn with the credentials of s2s.**

- CascadeCreate



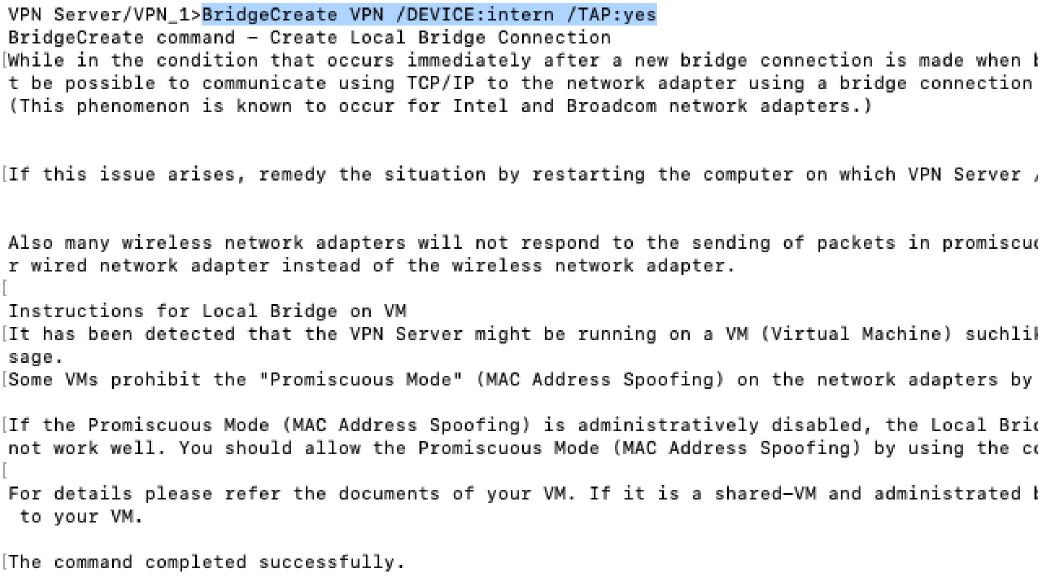
```
VPN Server/VPN_1>cascadecreate
CascadeCreate command - Create New Cascade Connection
Cascade Connection Name: cascade

Destination VPN Server Host Name and Port Number: 40.40.0.99:443

Destination Virtual Hub Name: vpn

Connecting User Name: s2s

The command completed successfully.
```

Figure 2.20: Create Cascade Connection

### 2.2.2 After the configuration, a NIC tap intern was created, which needs to be connected to eth0 by a bridge

**Create a Bridge**

- BridgeCreate VPN /DEVICE:intern /TAP:yes



```
VPN Server/VPN_1>BridgeCreate VPN /DEVICE:intern /TAP:yes
BridgeCreate command - Create Local Bridge Connection
[While in the condition that occurs immediately after a new bridge connection is made when
t be possible to communicate using TCP/IP to the network adapter using a bridge connection
(This phenomenon is known to occur for Intel and Broadcom network adapters.)


[If this issue arises, remedy the situation by restarting the computer on which VPN Server


Also many wireless network adapters will not respond to the sending of packets in promiscu
r wired network adapter instead of the wireless network adapter.
[
Instructions for Local Bridge on VM
[It has been detected that the VPN Server might be running on a VM (Virtual Machine) suchli
sage.
[Some VMs prohibit the "Promiscuous Mode" (MAC Address Spoofing) on the network adapters by

[If the Promiscuous Mode (MAC Address Spoofing) is administratively disabled, the Local Bri
not work well. You should allow the Promiscuous Mode (MAC Address Spoofing) by using the c
[
For details please refer the documents of your VM. If it is a shared-VM and administrated
 to your VM.

[The command completed successfully.
```

Figure 2.21: Create Bridge

### 2.2.3 Additonally add a route to 40.40.0.0/24 in r2

- ip route add 40.40.0.0/24 via 30.30.0.1

### 2.2.4 On r3 we need to modify the firewall to allow unrestricted access to web cascade

- iptables -A FORWARD -s 50.50.0.99/25 -j ACCEPT

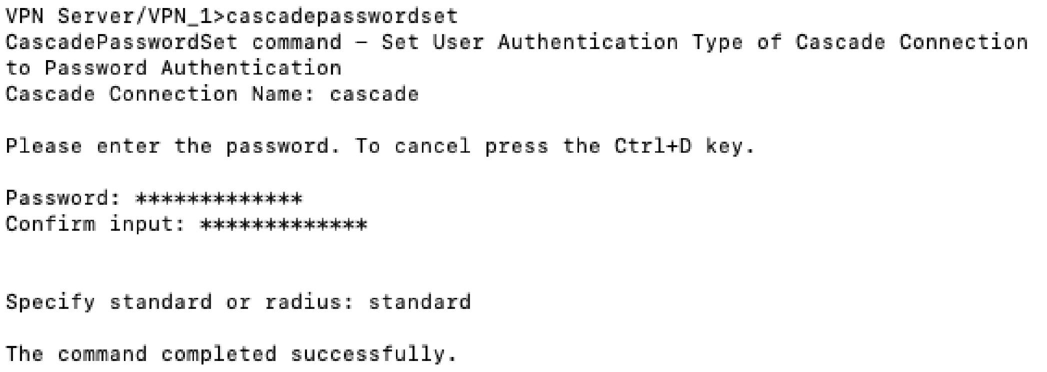
- iptables -A FORWARD -d 50.50.0.99/25 -j ACCEPT

### 2.2.5 If the cascade connection is established, we can add a route on web vpn to CD E

- ip route add default via 50.50.0.99 dev eth0

### 2.2.6 Set Password for Cascade Connection

- cascadepasswordset

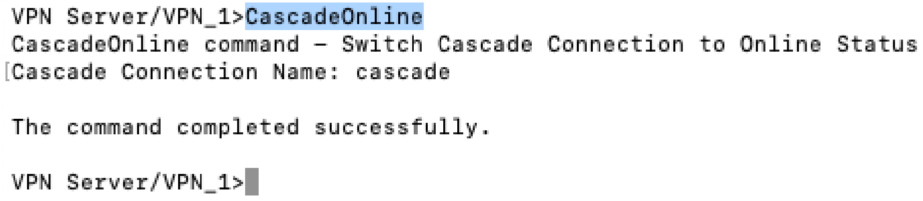


```
VPN Server/VPN_1>cascadepasswordset
CascadePasswordSet command - Set User Authentication Type of Cascade Connection
to Password Authentication
Cascade Connection Name: cascade

Please enter the password. To cancel press the Ctrl+D key.

Password: *************
Confirm input: *************


Specify standard or radius: standard

The command completed successfully.
```

Figure 2.22: Connects to the remote hub on web_vpn with the credentials of s2s.

### 2.2.7 Set the Cascade connection to online Status

- CascadeOnline



```
 VPN Server/VPN_1>CascadeOnline
 CascadeOnline command - Switch Cascade Connection to Online Status
[Cascade Connection Name: cascade

 The command completed successfully.

 VPN Server/VPN_1>
```

Figure 2.23: Setting the status of the Cascade to online

### 2.2.8 Within the networks being routable over the gateways web vpn and web cascade,we can setup all web servers to reach each other

**1.On web_sheldon, web_leonard, and web_howard we add a route to CD E over web vpn**

- web_sheldon : ip route add default via 40.40.0.99 dev eth0

### 2.2.9 On web penny, web bernadette, and web amy we add a route to CD D over web cascade.

- web_penny : ip route add default via 50.50.0.99 dev eth0

- web_bernadette : ip route add default via 50.50.0.99 dev eth0

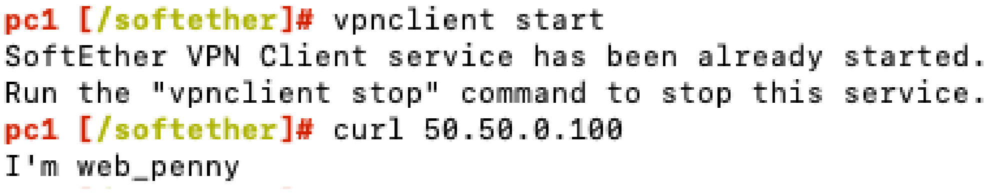
- web_amy : ip route add default via 50.50.0.99 dev eth0

### 2.2.10 Finally for the administrator on pc1, we add a route to 50.50.0.0/25 over web_vpn, which acts as a gateway for both restricted networks.

- Web_vpn : ip route add default via 50.50.0.99 dev eth0

### 2.2.11 start a Curl to any web server of CD E from pc1

- We will curl to web_penny from Pc1(50.50.0.100)

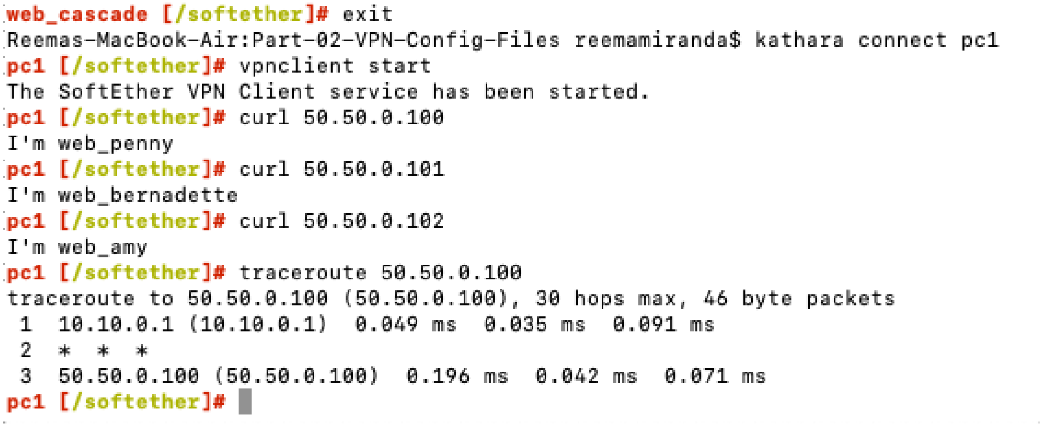


```
pc1 [/softether]# vpnclient start
SoftEther VPN Client service has been already started.
Run the "vpnclient stop" command to stop this service.
pc1 [/softether]# curl 50.50.0.100
I'm web_penny
```

Figure 2.24: Curl 50.50.0.100

### 2.2.12 start a traceroute to any web server of CD E from pc1 and confirm that your target is only two hops away and gets routed over web_vpn.

- traceroute 50.50.0.100



```
web_cascade [/softether]# exit
Reemas-MacBook-Air:Part-02-VPN-Config-Files reemamiranda$ kathara connect pc1
pc1 [/softether]# vpnclient start
The SoftEther VPN Client service has been started.
pc1 [/softether]# curl 50.50.0.100
I'm web_penny
pc1 [/softether]# curl 50.50.0.101
I'm web_bernadette
pc1 [/softether]# curl 50.50.0.102
I'm web_amy
pc1 [/softether]# traceroute 50.50.0.100
traceroute to 50.50.0.100 (50.50.0.100), 30 hops max, 46 byte packets
 1  10.10.0.1 (10.10.0.1)  0.049 ms  0.035 ms  0.091 ms
 2  *  *  *
 3  50.50.0.100 (50.50.0.100)  0.196 ms  0.042 ms  0.071 ms
pc1 [/softether]#
```

Figure 2.25: Traceroute 50.50.0.100

- As we can see in the above diagram we are able to reach the webserver web_penny (on CD E) in 2 hops.