IT Security Lab 1 – Report 3

Group 4

Meszlényi Lóránt,

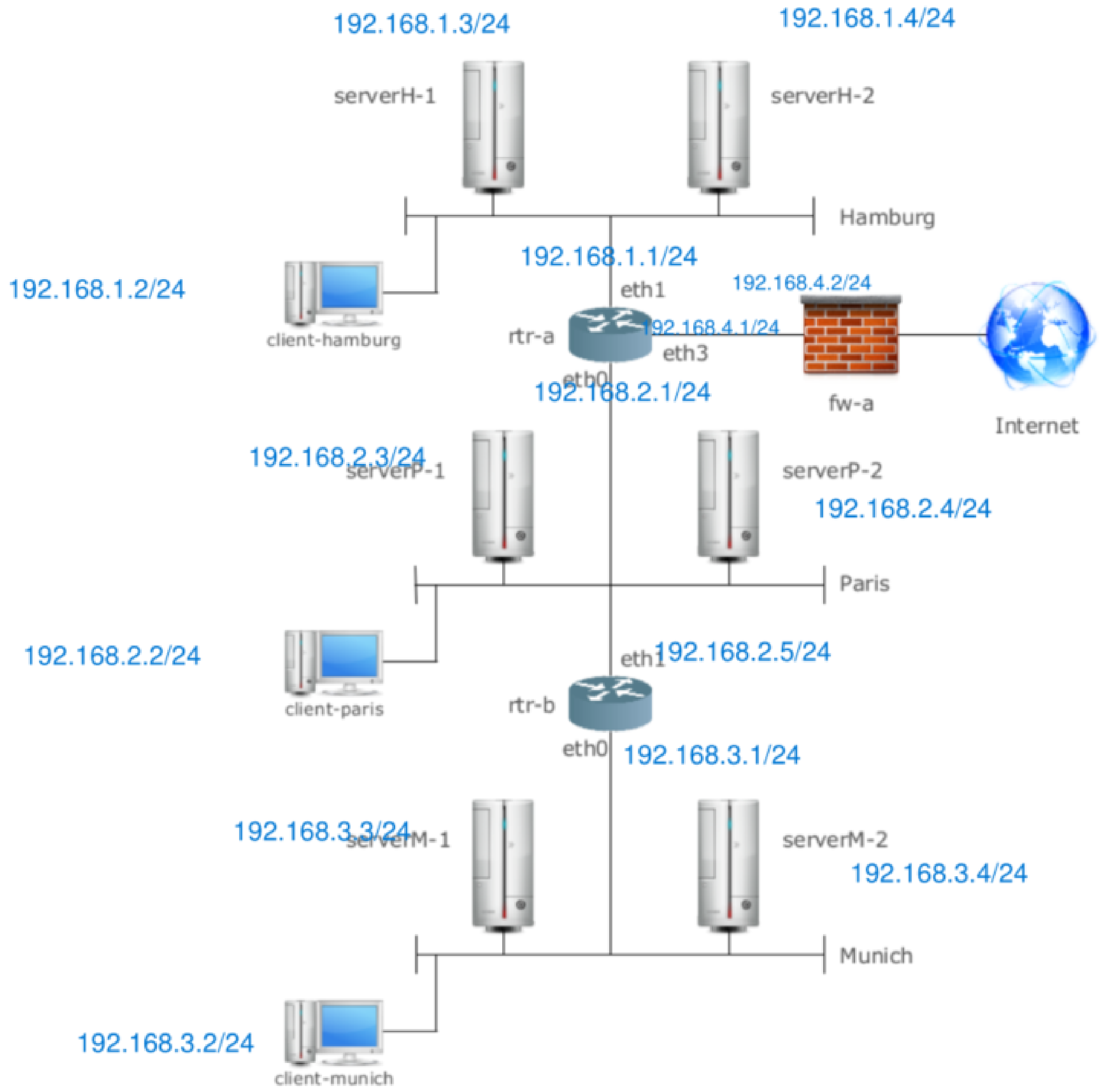
Shalkamy Omar,

Shashi Kumar Ravula,

08-12-2021

Exercise 1: Theoretical subnetting

1.1 Subnet the example network depicted in Figure 1 and give each network interface an IP address (hosts, routers,firewalls, clients, servers, etc.).



1.2 What is the gateway address that client-hamburg needs to use if it wants to connect to a server on the Internet?

Answer:

```
192.168.1.1
```

1.3 What is the broadcast address to reach only all the hosts that are connected to the Paris' network?

```
192.168.2.255
```

Exercise 2: Practical subnetting

- Configuration **Client-PA**

```
$ sudo ifconfig ens33 192.168.2.2
```

```
$ sudo ip route add 192.168.2.0/24 via 192.168.1.1
```

```
shashi@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0
ens33
192.168.2.0     192.168.1.1     255.255.255.0   UG    0      0        0
ens33
```

- Configuration **Server-HH**

```
$ sudo ifconfig ens33 192.168.1.2
```

```
$ sudo ip route add 192.168.1.0/24 via 192.168.1.1
```

```
shashi@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
192.168.1.0     192.168.2.1     255.255.255.0   UG    0      0        0
ens33
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0
ens33
```

- Configuration **Router**

```
$ sudo ifconfig eth0 192.168.1.1
```

```
$ sudo ifconfig eth0 192.168.2.1
```

```
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.1  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 00:0c:29:e4:10:53  txqueuelen 1000  (Ethernet)
        RX packets 260  bytes 81706 (79.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 3862 (3.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.1  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:0c:29:e4:10:5d  txqueuelen 1000  (Ethernet)
        RX packets 234  bytes 70516 (68.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 132  bytes 20986 (20.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Connectivity Checks

1. From `server-HH` send ping messages to the router and to `client-PA`

```
shashi@ubuntu:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.067 ms
^C
--- 192.168.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.037/0.059/0.068/0.013 ms
```

2. From the router send ping messages to server-HH and to client-PA

- `Router` to ` Client-PA`

```
shashi@ubuntu:~$ ping 192.168.2.2
┌──(kali㉿kali)-[~]
```

```
└─$ ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=0.623 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=64 time=0.743 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=64 time=0.776 ms
^C
--- 192.168.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.623/0.714/0.776/0.065 ms
```

- `Router` to `Server-HH`

```
$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.14 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=4.29 ms
^C
--- 192.168.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.139/2.712/4.286/1.573 ms
```

3. From `client-PA` send ping messages to the `router` and to `server-HH`

```
$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.24 ms
```

Exercise 3: Routing and DNS

Exercise 3: NAT

Make sure that all the machines inside the three subnets (HAMBURG, PASSAU, MUNICH) can still reach webservers (http, https) on the Internet.

- All machines are able to reach webservers

3.1 Explain

Explain the functionality enabled by the keyword 'MASQUERADE' in the context of the NAT configuration. Have you used it in your configuration?

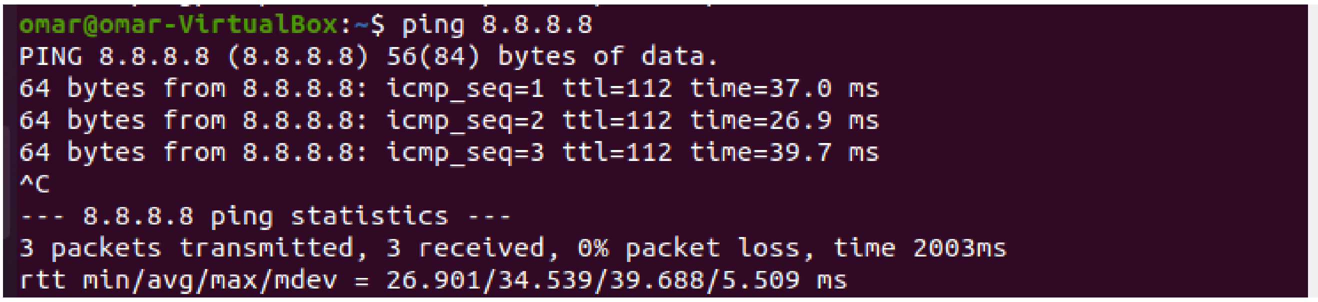
Solution Masquerade NAT allows to translate many IP addresses to one single IP address. masquerading in NAT can be used to hide one or more IP addresses on the internal network. We can make use of this to expose one single IP to public and rest inside private network.

- Yes, we used it in our configuration to hide the private ip addresses of our devices to be able to access the internet as shown below. We configured on firewall south that each packet coming from client will be postrouted into interface enp0s9 which is accessible to the internet and we make each connection coming from the subnet `ESTABLISHED` and `RELATED` for stable communication

```
iptables -t nat -A POSTROUTING -s 192.168.3.0/24 -o enp0s9 -j MASQUERADE
```

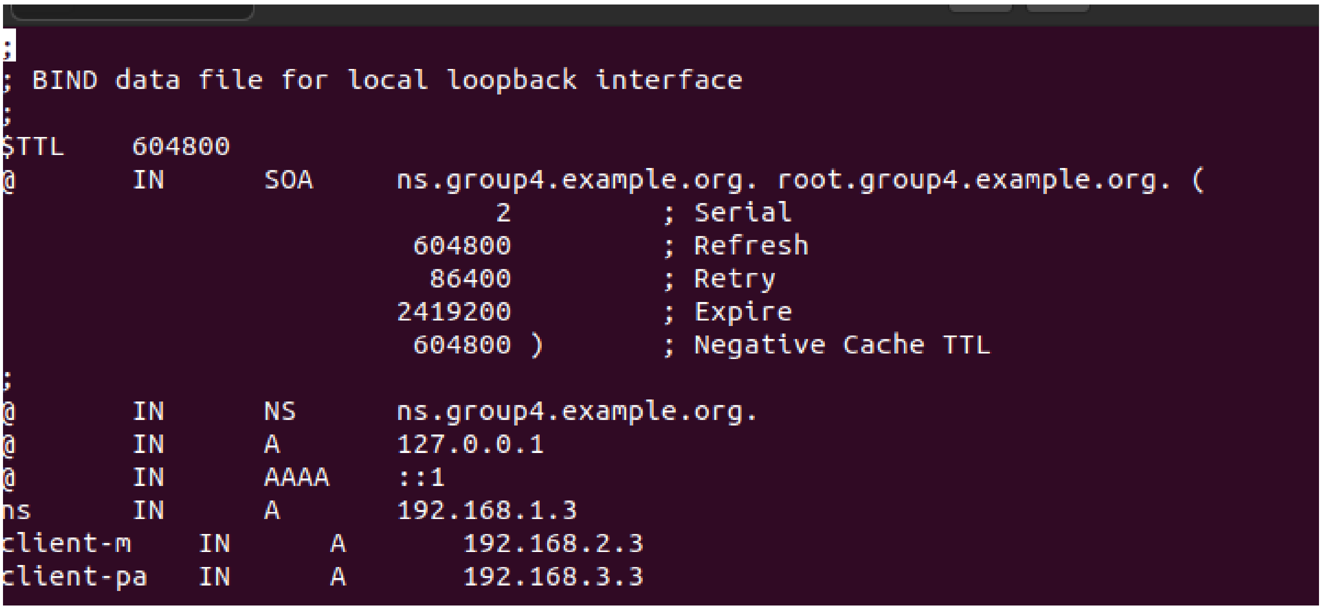
```
iptables -A FORWARD -s 192.168.3.0/24 -o enp0s9 -j ACCEPT
```

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -i enp0s9 -j ACCEPT
```



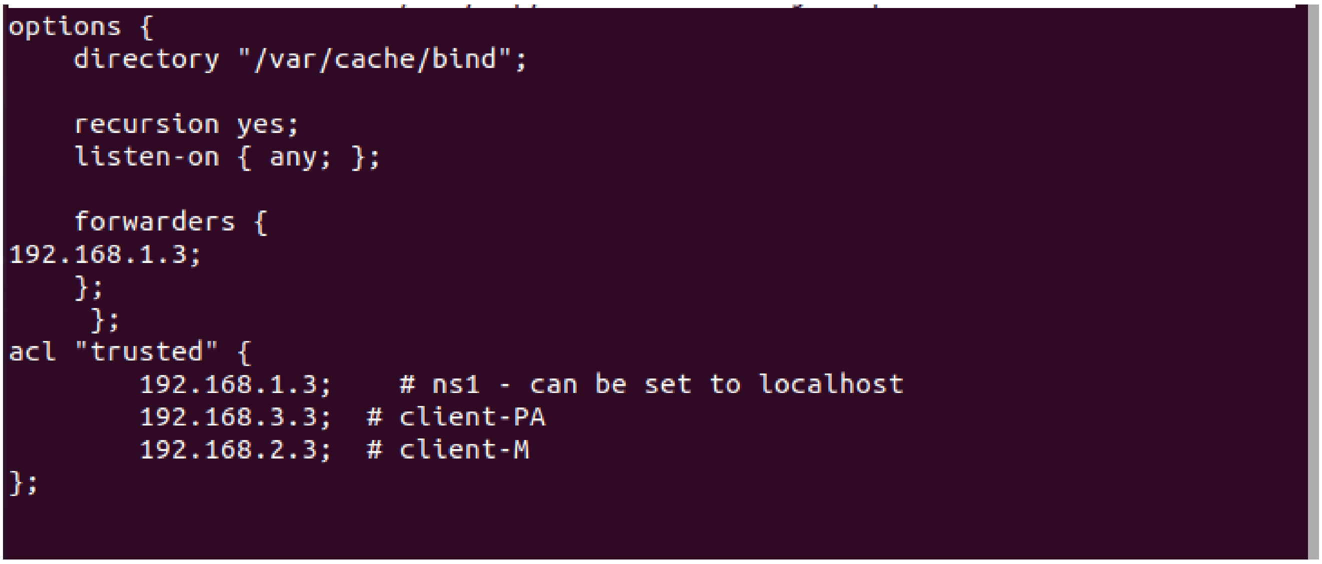
```
omar@omar-VirtualBox:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=37.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=26.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=39.7 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 26.901/34.539/39.688/5.509 ms
```

After configuring 5 VMs for the the 3 subnets and 2 firewalls we configured a DNS server on `server-HH`. We installed bind9 to confiure DNS. We created db.group4.example.org file to configure the dns and define the clients name with the relating IP address as shown below.

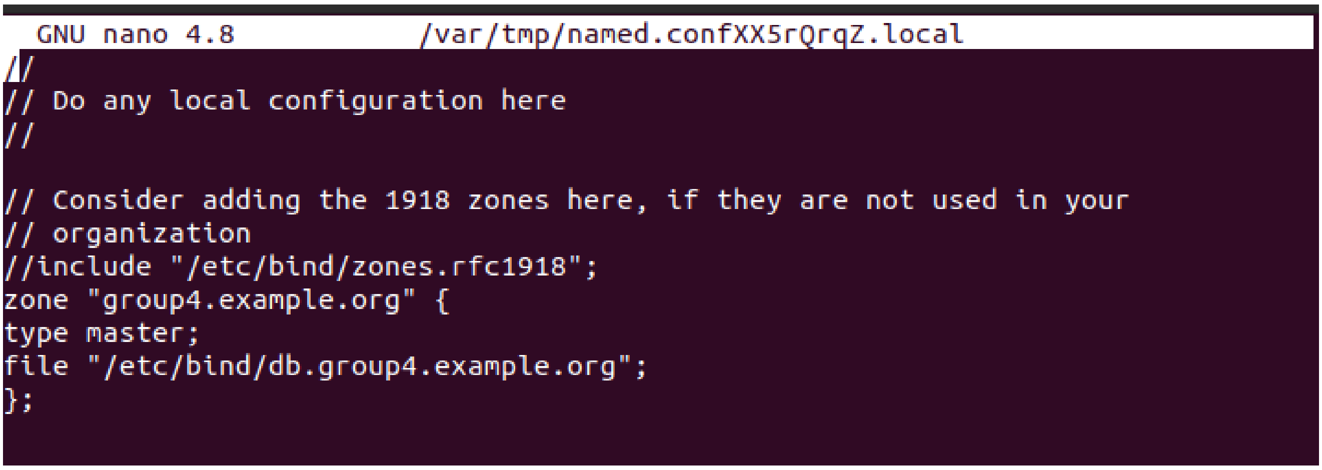


```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.group4.example.org. root.group4.example.org. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.group4.example.org.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
ns      IN      A       192.168.1.3
client-m    IN      A       192.168.2.3
client-pa   IN      A       192.168.3.3
```

After that we configured named.conf.options and named.conf.local files in order to provide the zone, forwarders and trusted access list

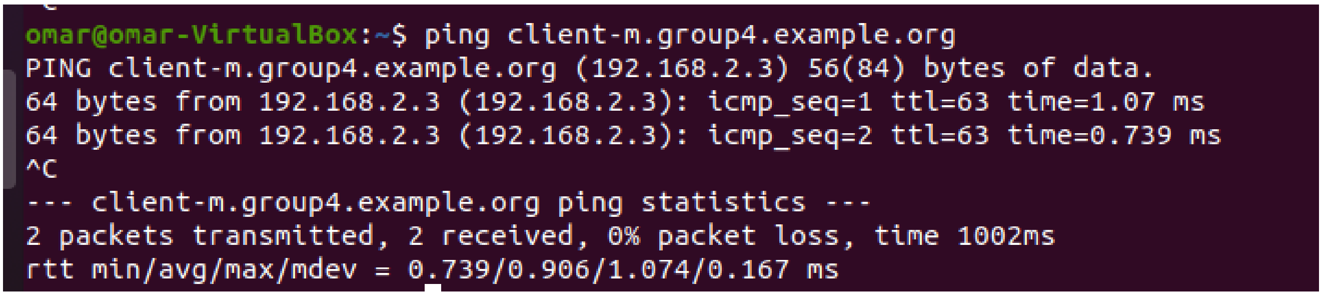


```
options {
    directory "/var/cache/bind";

    recursion yes;
    listen-on { any; };

    forwarders {
192.168.1.3;
    };
     };
acl "trusted" {
        192.168.1.3;    # ns1 - can be set to localhost
        192.168.3.3;  # client-PA
        192.168.2.3;  # client-M
};
```



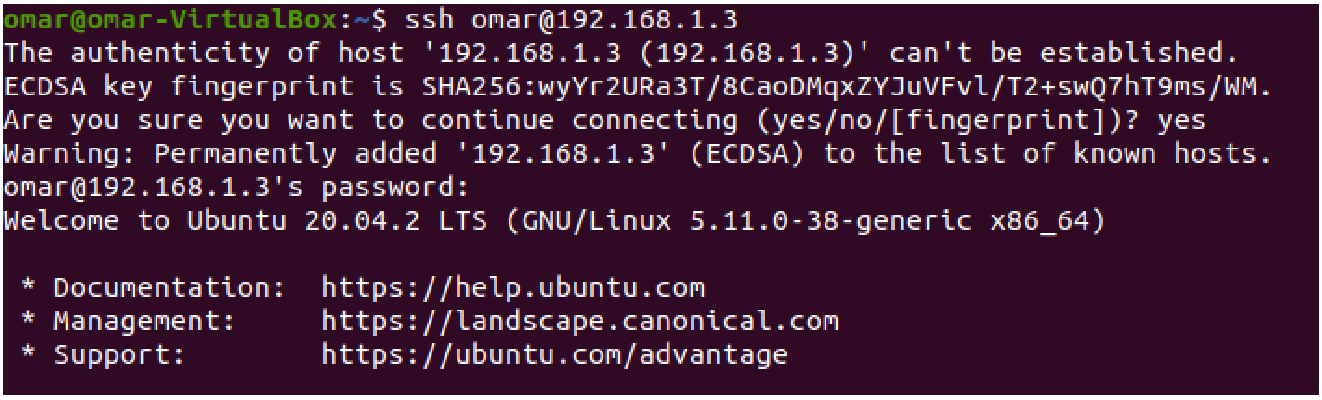
```
  GNU nano 4.8           /var/tmp/named.confXX5rQrqZ.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "group4.example.org" {
type master;
file "/etc/bind/db.group4.example.org";
};
```

The final step to configure the DNS is from the client side and this can be done by adjusting `/etc/resolv.conf` file on the client or adjusting the DNS configuration manually to `server-HH` IP address and the screenshot below shows we were able to ping using the hostnames from DNS server.



```
omar@omar-VirtualBox:~$ ping client-m.group4.example.org
PING client-m.group4.example.org (192.168.2.3) 56(84) bytes of data.
64 bytes from 192.168.2.3 (192.168.2.3): icmp_seq=1 ttl=63 time=1.07 ms
64 bytes from 192.168.2.3 (192.168.2.3): icmp_seq=2 ttl=63 time=0.739 ms
^C
--- client-m.group4.example.org ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.739/0.906/1.074/0.167 ms
```

We tried to connect through ssh from `client-PA` into `server-HH` and it worked after installing the SSH connection.



```
omar@omar-VirtualBox:~$ ssh omar@192.168.1.3
The authenticity of host '192.168.1.3 (192.168.1.3)' can't be established.
ECDSA key fingerprint is SHA256:wyYr2URa3T/8CaoDMqxZYJuVFvl/T2+swQ7hT9ms/WM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.3' (ECDSA) to the list of known hosts.
omar@192.168.1.3's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.11.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

Exercise 4: Firewalling

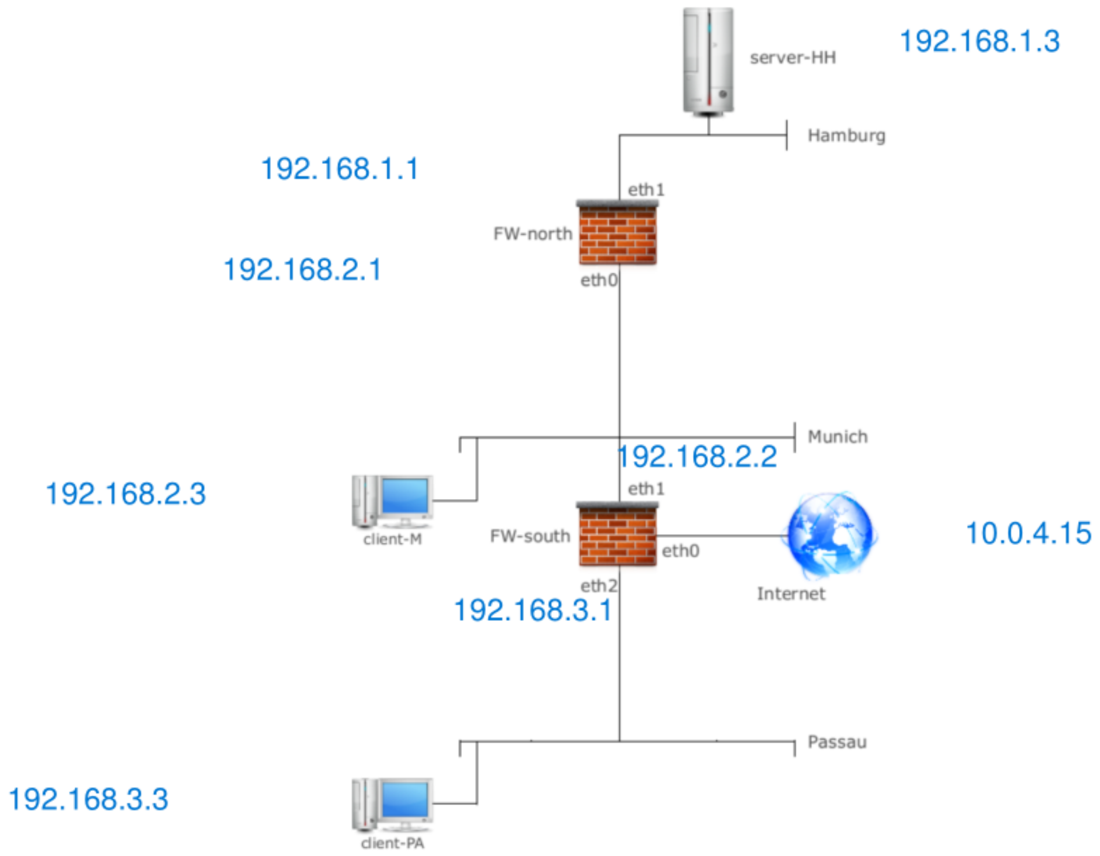


Figure 3: Firewalled networks

4.1 Set the firewalls to deny all connections by default.

Solution

- In our setup following policies are setup on both firewalls.

```
iptables -P INPUT DROP   # Drop all incoming packets
```

```
iptables -P OUTPUT DROP   # Drop all outgoing packets
iptables -P FORWARD DROP   # Disable forwarding
```

4.2 Prohibit machines on the internal network to provide services to the outside (e.g. an Internet reachable webserver on client-PA shall be prohibited), but allow all machines to use any service on any machine (internal subnets & Internet) as long as they initiate the connection

Solution Access to the internet is blocked by default

1. Explain the differences between dynamic and static packet filtering. You should have used a dynamic filtering rule in the exercise above, state which and explain how it works.
Solution:

- **Static filtering :** In static filtering, firewall rule decises which packets are allowed or denied. Firewall evaluates each packet independently and has no impact with previous packets that have passed or denied.
- **Dynamic filtering:** In dynamic filtering firewall, it reacts to an event and create or update rules to handle with that particular event. It filters traffic with particular connection states, usually filtered by IP and PORT. For eg: Opening an FTP to outside world, PORT 21 must be left open permanently open so that outside clients can attempt establishing connection.
    - Dynamic filtering allows port 21 to be opened at the start of an FTP session and then closes at the end of the session.

2. Explain in what form dynamic filtering is better than static filtering?

Solution: The advantage of dynamic filtering is stateful packet inspection. These stateful packet inspection filters the exchange of packets, effectively by opening ports in the firewall for each communications session when needed basis, and then close the port as soon as they're no longer needed. One can easily allow or block the traffic accordingly.

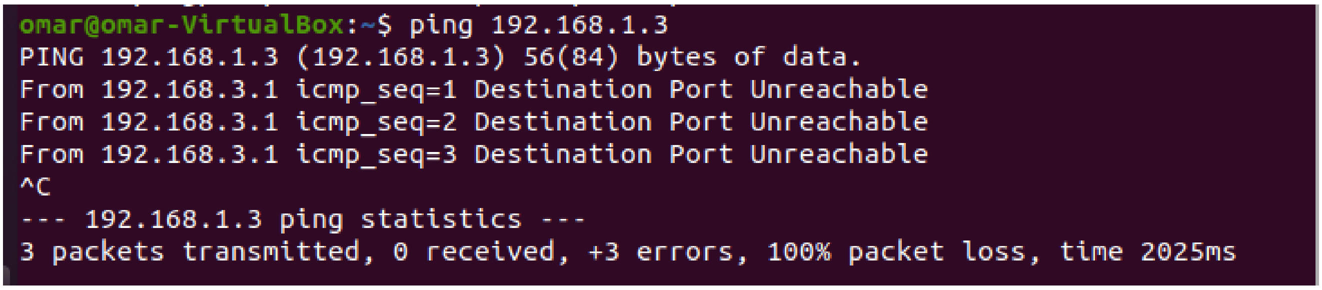
- With this option one can switch events to inspect packets, which is useful in assisting security problems.

4.3 Tell the firewall FW-south to REJECT all icmp requests from the PA Subnet and the Lab's network. Test this by trying to ping from client-PA, but also try if the reverse succeeds (e.g. it shall still be possible to ping the computer client-PA from server-HH).

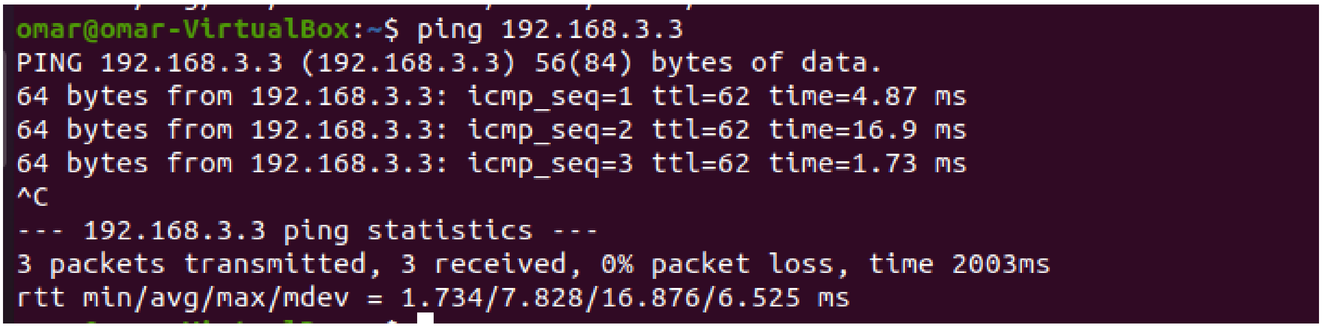
Solution

```
iptables -A FORWARD -s 192.168.3.3/24 -p ICMP --icmp-type 8 -j REJECT
```

Below you can find client-PA cannot ping server-HH after appling the rule while server-HH can ping client-PA normally



```
omar@omar-VirtualBox:~$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
From 192.168.3.1 icmp_seq=1 Destination Port Unreachable
From 192.168.3.1 icmp_seq=2 Destination Port Unreachable
From 192.168.3.1 icmp_seq=3 Destination Port Unreachable
^C
--- 192.168.1.3 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2025ms
```



```
omar@omar-VirtualBox:~$ ping 192.168.3.3
PING 192.168.3.3 (192.168.3.3) 56(84) bytes of data.
64 bytes from 192.168.3.3: icmp_seq=1 ttl=62 time=4.87 ms
64 bytes from 192.168.3.3: icmp_seq=2 ttl=62 time=16.9 ms
64 bytes from 192.168.3.3: icmp_seq=3 ttl=62 time=1.73 ms
^C
--- 192.168.3.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.734/7.828/16.876/6.525 ms
```

4.4 Remove the ability to ping the firewalls themselves. This means you must tell the firewall to REJECT all icmp requests addressed to the FW-north and FW-south.

- Firewall North & Firewall south

```
Iptables -A INPUT -p ICMP --icmp-type 0 -j DROP
Iptables -A OUTPUT -p ICMP --icmp-type 8 -j DROP
```

4.5 SSH sessions are only allowed to server-HH, all other ssh connections (despite of their destination or origin) shall be blocked.

Solution

- Tell the firewall(s) to forward all `SSH` connections to `server-HH`

North && South

```
sudo iptables -A FORWARD -p tcp -d 192.168.1.3 --dport 22 -m state --state
NEW,ESTABLISHED -j ACCEPT
```

```
sudo iptables -A FORWARD -p tcp -s 192.168.1.3 -d --sport 22 -m state --
state
ESTABLISHED -j ACCEPT
```

- Reject for all others

```
sudo iptables -A FORWARD -p tcp --sport 22 -j REJECT
```

4.6 Make sure that the FW facing the Internet (Lab's Net) prohibits that IP packets with a source address of the internal subnets arrive on the external interface (i.e. eth0 on FW-south).

Solution

- On south firewall

```
sudo iptables -A OUTPUT -o eth0 -s 192.168.3.1/24 -j REJECT
sudo iptables -A OUTPUT -o eth0 -s 192.168.2.1/24 -j REJECT
sudo iptables -A OUTPUT -o eth0 -s 192.168.1.1/24 -j REJECT
```

1. What is the reason to have FW rules that prohibit IP packets with a source address inside the internal subnet to leave to the external interface?

- The above rule will block all connection to internet with source from internal subnet addresses. Firewall on the south uses `NAT` and connected to internet, hence internal network need not be exposed to connect to internet. In such case of accessing internet, the south firewall need to route route the packets when trying to access the external network.

Exercise 5: Firewalling continued

5.1 Give the iptables commands / rules that allow users on subnet PASSAU to view web pages (http, https) on a web server running in subnet HAMBURG (start/install web-server on server-HH). The rules shall block access to this server from the MUNICH subnet.

Solution

1. Installing `Nginx` server (`server-HH` - `192.168.1.3`).

```
$: sudo apt install nginx
```

- Passau subnet (`192.168.3.1/24`)

  On North & South firewall

```
sudo iptables -A FORWARD -p tcp -s 192.168.3.1/24 -d 192.168.1.1/24 -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```

- Block from munich subnet

North firewall

```
sudo iptables -A FORWARD -p tcp -s 192.168.2.1/24 -d 192.168.1.1/24 -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j REJECT
```

- `-m conntrack`: allow the match based on connection state
- `--cstate`: parameter to define the list of states(like `new`, `established`, `closed`)

5.2 Write the iptables commands / rules to allow HTTP and HTTPS traffic from the Internet (Lab's Net) into the HAMBURG subnet. This includes access to a web server on server-HH.

```
sudo iptables -A FORWARD -i eth0 -p tcp -d 192.168.1.0/24 -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -o eth1 -p tcp -s 192.168.1.0/24 -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

5.3 Do not allow nor route any packets that try to use any host other than server-HH as their DNS nameserver. The DNS server port is 53. DNS can use both the tcp and udp protocols (udp by default).

On both firewalls:

- Allow DNS (53) from server-HH - 192.168.1.3

```
iptables -A INPUT -p udp --dport 53 -s 192.168.1.3 -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -s 192.168.1.3 -j ACCEPT
```

- Deny all other DNS requests

```
iptables -A INPUT -p udp --dport 53 -j DROP
iptables -A INPUT -p tcp --dport 53 -j DROP
```

5.4 Can you still browse the internet? Explain why / why not. You still want all machines (in all three subnets) to browse the World wide Web (http/https). There are, roughly speaking, ways to accomplish this in regard to DNS resolution.

Solution No, we are unable to browse the internet, although connection to external IP addresses is still possible. This simply means that domain names can not be resolved.

We can use:

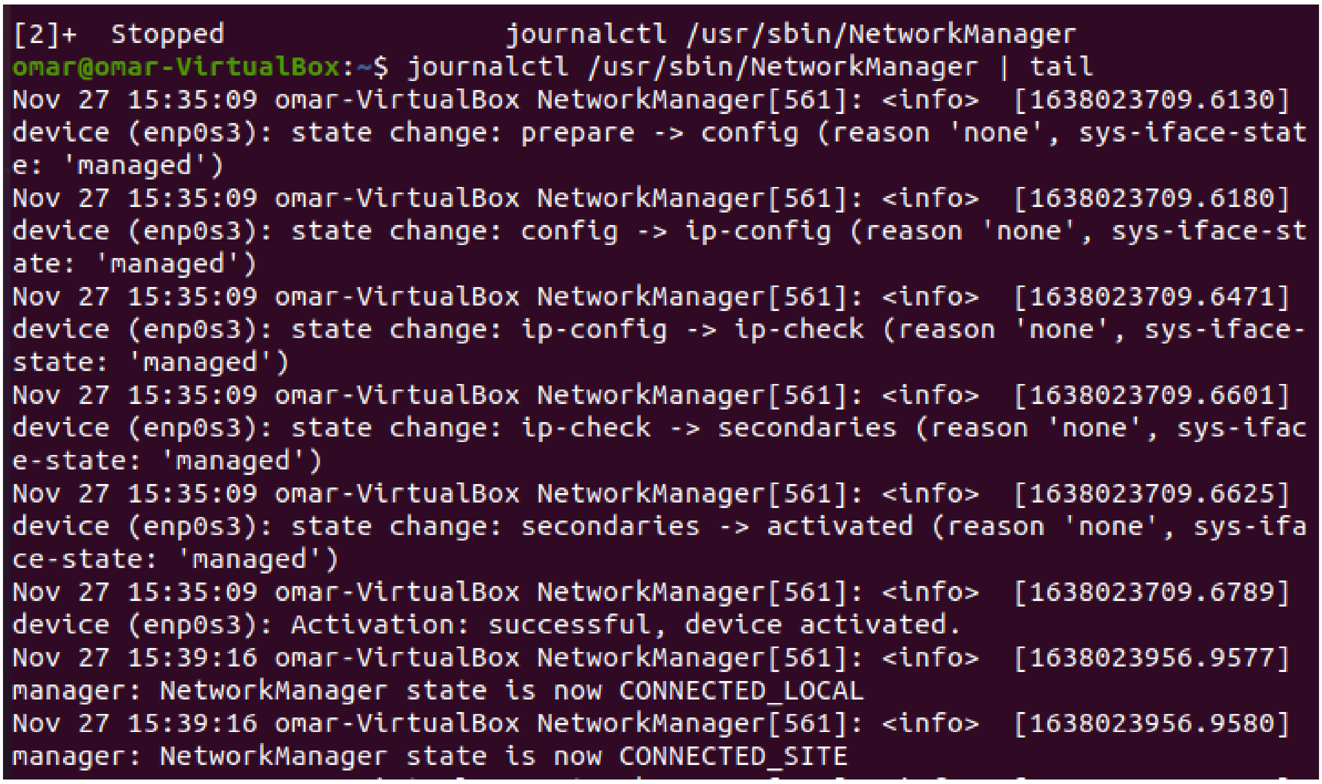
- To set in `Server-HH` one of Google's DNS Server 8.8.8.8 as additional forwarder. The server is then forwarding DNS queries to a external server

5.5 Log all traffic that attempts to connect to server-HH
Solution

```
iptables -A FORWARD -m state --state NEW -d 192.168.1.3 -j LOG --log-prefix
"New HH Connection: "
```

- log files can be viewed in

```
$: cat /var/log/syslog | tail
```



```
[2]+  Stopped                 journalctl /usr/sbin/NetworkManager
omar@omar-VirtualBox:~$ journalctl /usr/sbin/NetworkManager | tail
Nov 27 15:35:09 omar-VirtualBox NetworkManager[561]: <info>  [1638023709.6130]
device (enp0s3): state change: prepare -> config (reason 'none', sys-iface-stat
e: 'managed')
Nov 27 15:35:09 omar-VirtualBox NetworkManager[561]: <info>  [1638023709.6180]
device (enp0s3): state change: config -> ip-config (reason 'none', sys-iface-st
ate: 'managed')
Nov 27 15:35:09 omar-VirtualBox NetworkManager[561]: <info>  [1638023709.6471]
device (enp0s3): state change: ip-config -> ip-check (reason 'none', sys-iface-
state: 'managed')
Nov 27 15:35:09 omar-VirtualBox NetworkManager[561]: <info>  [1638023709.6601]
device (enp0s3): state change: ip-check -> secondaries (reason 'none', sys-ifac
e-state: 'managed')
Nov 27 15:35:09 omar-VirtualBox NetworkManager[561]: <info>  [1638023709.6625]
device (enp0s3): state change: secondaries -> activated (reason 'none', sys-ifa
ce-state: 'managed')
Nov 27 15:35:09 omar-VirtualBox NetworkManager[561]: <info>  [1638023709.6789]
device (enp0s3): Activation: successful, device activated.
Nov 27 15:39:16 omar-VirtualBox NetworkManager[561]: <info>  [1638023956.9577]
manager: NetworkManager state is now CONNECTED_LOCAL
Nov 27 15:39:16 omar-VirtualBox NetworkManager[561]: <info>  [1638023956.9580]
manager: NetworkManager state is now CONNECTED_SITE
```

5.7 In the exercise above you have enabled logging of certain events. Why is it important to keep and analyze the logs? What can you do with them?
Solution:

- System or server logs are essential in diagnosing, troubleshooting, and identifying cause of failures. They also help in understanding the various events took place in its lifetime.

- Log files contains useful information like time, device name, state, IP addresses, in case of any malicious attack these are most valuable information.
- With this information, it is easy to pinpoint various potential issues, without these it is hard to determine, what is going on in the server.

Exercise 6: -EXTRA-CREDITS- DNS in more detail

6.1 Explain the concept of DNS zones. Explain the difference between a 'managed' and a 'delegated' zone.
Solution:

1. DNS zone is portion of DNS name space, which contains DNS records.
2. Allowing zones, makes it easy to handle DNS records for adminitartive reasons and for redundancy.
3. Zones allow more fine grained granular control over the DNS records and components.
4. A DNS zone can also contains multiple subdomains and zones may also co-exist on same physical server.

Managed zones: Set or container of all DNS records that has same DNS prefix. Eg: `example.de`.

Delegated zones: Delegation allows an organization to assign control of a subdomain to another organization. The parent now has pointers to the original sources of data in the subdomain. Delegated zones are zones delegated or managed by another name server who has authority over that zone.

6.2 Explain shortly (no more than 10 sentences) how the Internet's DNS system is set up (e.g. root servers, zones, registrars ...).
Solution

1. Whenever a DNS request is sent, is usually handled by the DNS server to map IP address, which is in many most cases internet service provider(ISP).

2. This DNS server also called `Recursive DNS resolver`.

3. DNS resolver checks its cache for corresponding domain, if unavilable, it request `Root DNS server`.

4. The Root DNS resolver responds with one of TLD name server(Top lovel domain) server. For Eg: www.example.com has `.com` TLD name server (`.com NS`).

5. `.com NS` responds to follow up with responsible authoritative name server(Which is `ns1.exmaple.com`).

6. Then DNS resolver request the authoritative name server to obtain the IP address.

- Reverse DNS look up: Maps IP addresses to domain names. It allows to track the origin of the website.

6.3 Explain the concept of DNS forwarding? Are there any security gains when DNS forwarding is used? Is there any additional filtering that would be possible (give an example)?

- DNS forwarding allows DNS queries to be handled by seperate designated server rather than the server that has been contacted initially.

- DNS forwarders sinmply forward from one server to another server, rather than addressing the query.

- DNS server are configured to forward request(Most cases) for all the addresses that are not within the network to a dedicated `server/forwarder`. For Eg: forwarding a request to `8.8.8.8` or any other trusted DNS, when an internal IP is trying connect to external domain.

Security Gains:

- If requests are forwarded to known servers, and not to local Internet Service Provider (ISP). Benefits are increased performance and security from phishing, malware, botnets, and targeted online attacks. As the resolve process is trusted, clients can be sure of not being tricked with fake domains.

6.4 Explain in detail what a zone transfer is? Why is it needed? What is the difference between a zone transfer and a zone replication?

Solution

- DNS zoine transfer is the process of copying/transfering DNS records from oner server to another, usually from primary DNS to secondary DNS server.

- Zone transfer uses Transcation based TCP protocol.Takes form from client to server.

- Zone transfer is needed either for creating a backup server or for redundancy, where the redundant server acts as secondary DNS server.

Differences

Zone transfer can be a full transfer or incremental transfer but zone replication allows to specify records to copy/replicate. Zone replication allows you to be able to decide the conditions or parameters for replicating the DNS zone.

6.5 What is the concept/idea behind 'Incremental zone transfers' ? What is the gain? Are they widely used in today's DNS system?

Solution:

- **Incremental Zone Transfer:** Incremental zone transfers, the secondary DNS server retrieves only resource records that have been modified or changed within a zone, so that it remains synchronized with the primary DNS server.
- When incremental transfers are used, the databases on the primary server and the secondary server are compared against each other to check differences.
- If the zone records are identified as original (based on the serial number of the Start of Authority resource record), no zone transfer is performed. If not a transfer of the delta resource records commences (A serial number sequence is checked to see if the transfer has occured or not).
- Because of this transfer method, it requires less bandwidth and create less network traffic, making them to copy DNS records faster.
- Yes, Incremental zone transfers are widely used and they are efficient ways to copy records from primary to secondary DNS server.

6.6 Can you think of any malicious attacks? Give some details how you would carry out such attacks.
SOlution

Carrying out a basic DNS Zone Transfer Attack isn't very hard: All one need to do is , pretend that he is the secondary server and request the primary server for a copy of the zone records. And it sends without identifying the origin.

- One of the simplest ways to stop zone transfer attacks iws by restricting Zone transfers. At very minimum security level we can tell the IP address of the secondary and not transfer to anyone else.
- In more complex setup,sign in may be required for transfers.
- Enumerating DNS using `host`

```
host -t ns example.com
```

```
host -l example.com ns1.example.com
```

- Additionally tools like `dnsrecon` and `dnsenum` can be used to dump the DNS information and records.

6.7 What are the general options to secure DNS transfers? How do they roughly work?

Solution

- Some of the common ways attacks in DNS transfers happen in DNS zone transfers.

Impact of DNS Zone Transfer Vulnerability

- Zone transfer can be done withotut any form of authentication, as any client can request for copy of DNS records of the entire zone and server can send without verifying the client.
- Unless some protection mechanism is used, anyone will be able to get all the records for that particular domain, which gives the potential attacker to obtain network information and map to find various potential targets.

Preventing DNS Zone Transfer Vulnerability

- Allowing zone transfers from trusted clients
- The following is an example of how to fix this in the `BIND DNS` server.
- Navigate to `/etc/named.conf` and add these:

```
ACL trusted-servers
    {
        192.168.2.2; // ns1
        192.168.1.2; // ns2
    };
    zone example.com
    {
        type master;   file "zones/zonetransfer.me";
        allow-transfer { trusted-servers; };
    };
```

- Some of the DNS servers ask for HMAC(Keyed Message authentication codes) when requesting zone transfers. Servers can identify the legitimate clients by verifying the HMACs.