Co-funded by the
Erasmus+ Programme
of the European Union

ISSES

ISSES – Information Security Services
Education in Serbia

Supported by the Erasmus+ Capacity Building in the field of Higher Education (CBHE) grant N° 586474-EPP-1-2017-1-RS-EPPKA2-CBHE-JP

Network security (IP packet filters)

- Laboratory exercise -

Use the *Putty* to connect to all three virtual machines using the *SSH* protocol.

1. On VM2, check if routing is enabled with the command:

- `cat /proc/sys/net/ipv4/ip_forward`

If 0 is obtained, it means that routing is not enabled and should be enabled with the command:

- `sudo sysctl –w net.ipv4.ip_forward=1`

Check again that routing is enabled.

2. Raise interfaces and establish routing between VM1 and VM3 through VM2:

- VM1
  - `sudo ip link set eth1 up`
  - `sudo ip addr add 10.12.x.1/24 dev eth1`
  - `sudo ip route add 10.23.x.0/24 via 10.12.x.2`
- VM2
  - `sudo ip link set eth1 up`
  - `sudo ip addr add 10.12.x.2/24 dev eth1`
  - `sudo ip link set eth2 up`
  - `sudo ip addr add 10.23.x.2/24 dev eth2`
- VM3
  - `sudo ip link set eth2 up`
  - `sudo ip addr add 10.23.x.3/24 dev eth2`
  - `sudo ip route add 10.12.x.0/24 via 10.23.x.2`

3. Check that packets pass between VM1 and VM3. Execute the command on VM3:

- `ping 10.12.x.1`
- `traceroute 10.12.x.1`
- Expected output:

```
traceroute to 10.12.x.1 (10.12.x.1), 30 hops max, 60 byte packets
1 10.23.x.2 (10.23 .x.2) 0.274 ms 0.207 ms 0.174 ms
2 10.12.x.2 (10.12.x.2) 0.415 ms 0.401 ms 0.371 ms
```

4. Running services on VM3:

- Install *FTP* server using the command:
  - `sudo apt install vsftpd`
  - `sudo systemctl restart vsftpd.service`

- `sudo python3 -m /home/student/Documents/web/http.server 80`
  (or `cd /home/student/Documents/web` and then
  `sudo python3 – m http.server 80`)
  (web server from the *DNS* lab exercise is assumed to be there)

- Check on VM1 if *ftp*, *ssh* and web services are available:
  - `ftp 10.23.x.3` (exit using `quit`)
  - `ssh 10.23.x.3`
  - `curl 10.23.x.3`

Establishing a packet filter.

Example 1.

On VM2, check which rules are set with the command:

- `sudo iptables -L`

For both inbound and outbound packets and for packets that are routed *policy accept* is set.

Setting a *DROP* routing policy (*FORWARD*) that filters all packets:

- `sudo iptables --policy FORWARD DROP`

Check the contents of the filter table again:

- `sudo iptables -L`

Try again all services on VM1 in the same way as in the second bullet of step 4.

Return the *FORWARD* policy to *ACCEPT*:

- `sudo iptables --policy FORWARD ACCEPT`

Set *FORWARD* filter for *ssh*

- `sudo iptables -A FORWARD -i eth1 -p tcp --dport 22 -d 10.23.x.3 -j DROP`

Check the contents of the filter table again:

- `sudo iptables -L`

Try all services again.

Set a *FORWARD* filter for web

- `sudo iptables -A FORWARD -i eth1 -p tcp --dport 80 -d 10.23.x.3 -j DROP`

Check the contents of the filter table again:

- `sudo iptables -L`

Try all services again.

Set the *FORWARD* filter for *ftp*

- `sudo iptables -A FORWARD -i eth1 -p tcp --dport 21 -d 10.23.x.3 -j DROP`

Check the contents of the filter table again:

- `sudo iptables -L`

Try all services again.

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Example 2.

Set the following filter:

- `sudo iptables -A FORWARD -i eth1 -p tcp --dport 22 -d 10.23.x.3 -j ACCEPT`
- `sudo iptables -A FORWARD -i eth1 -p tcp -d 10.23.x.3 -j DROP`

Try all services again.

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Set the following filter:

- `sudo iptables -A FORWARD -i eth1 -p tcp -d 10.23.x.3 -j DROP`
- `sudo iptables -A FORWARD -i eth1 -p tcp --dport 22 -d 10.23.x.3 -j ACCEPT`

Try all services again.

What is the difference between cases 1 and 2? Why is there this difference?

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Example 3.

Set the following filters on VM2:

- `sudo iptables -A FORWARD -i eth1 -p tcp --dport 22 -d 10.23.x.3 -j DROP`
- `sudo iptables -A FORWARD -i eth1 -p tcp -d 10.23.x.3 -j ACCEPT`

Set the following filter on VM3:

- `sudo iptables -A INPUT -i eth2 -p tcp --dport 80 -d 10.23.x.3 -j DROP`

Try all services again.

Which device blocked the web and which *ssh* packets?

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Example 4.

Set the following filters on VM2:

- `sudo iptables -A FORWARD -i eth2 -p tcp --sport 22 -s 10.23.x.3 -j DROP`
- `sudo iptables -A FORWARD -i eth2 -p tcp -s 10.23.x.3 -j ACCEPT`

On VM3, open another window and run:

- `sudo tcpdump -i eth2`
  which records all packets passing through the *eth2* interface

Try all services again. Then stop *tcpdump*.

Are packages for VM1 registered on VM3?

Why doesn't the *ssh* session work?

Which packets are filtered and where?

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Example 5.

Set the following filters on VM2:

- `sudo iptables -A FORWARD -i eth1 -p ip -d 10.23.x.3 -j DROP`
- `sudo iptables -A FORWARD -i eth1 -p tcp -d 10.23.x.3 -j ACCEPT`

Try again all services and ping to 10.23.x.3.

Why packets are not passed?

What filters the first and what filters the second row of the filter table?

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Example 6.

Set the following filters on VM2:

- `sudo iptables -A FORWARD -i eth1 -p ip -d 10.23.x.0/24 -j DROP`
- `sudo iptables -A FORWARD -i eth1 -p ip -d 10.23.x.3 -j ACCEPT`

Try again all services and ping to 10.23.x.3

Reset filter tables to initial status:

- `sudo iptables -F`
- `sudo iptables -X`

Now set filters:

- `sudo iptables -A FORWARD -i eth1 -p ip -d 10.23.x.3 -j ACCEPT`
- `sudo iptables -A FORWARD -i eth1 -p ip -d 10.23.x.0/24 -j DROP`

Try again all services and ping to 10.23.x.3

What is the difference between the first and the second case?

NOTICE FOR STUDENTS

Topics of this laboratory exercise may involve the study of various mechanisms that violate information security and make intrusions into computer systems and networks.

The application of these mechanisms when executed towards the systems of individuals and legal entities, which are not familiar with them and are not consentient with the activities on checking vulnerability and testing intrusions into their systems, is punishable under the Criminal Law of the Republic of Serbia (Articles 298 to 304a).

Students performing this laboratory exercise may use these methods for study purposes only within the closed laboratory environment provided for teaching the Advanced Network and System Security course and described in Network Security Laboratory (NS Lab) Design and Hybrid NS and Crypto Lab Implementation documents available at: https://isses.etf.bg.ac.rs/labs/

Students may not imply that they are in any way encouraged or that they are recommended to use these methods toward systems of any third party entity or individual.

Any eventual activity that a student would undertake using these methods and mechanisms according to systems that are not within the laboratory on the course is the sole responsibility of the student.