LAB 5

**Task 1.A: Implement a Simple Kernel Module.**

A simple loadable kernel module is shown below. When the module is loaded, it prints "Hello World!"; when the module is deleted from the kernel, it prints "Bye-bye World!"

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

**Task 1.B Implement a Simple Firewall Using Netfilter.**

1. To test the firewall, the command $ dig @8.8.8 www.example.com was used to produce UDP packets. The request was denied, indicating that the firewall was functioning correctly.

Text

Description automatically generated

Text

Description automatically generated

2) The code has been attached.Table

Description automatically generated with medium confidence

3) The code has been attached.

Text, letter

Description automatically generated

**Task 2: Experimenting with Stateless Firewall Rules**

Through the above rules, only icmp packets will be accepted and all other packets shall be dropped. Hence, we can only the router and not telnet.

Text

Description automatically generated

**Task 2.B: Protecting the Internal Network**

* 10.9.0.5 is our outside host & 192.168.60.5/.6/.7 are our inside hosts.
* Outside hosts are unable to ping inside hosts.

Text

Description automatically generated

* The router can be pinged from the outside.

Text

Description automatically generated

* Outside hosts can ping internal hosts.

Text

Description automatically generated

* Between both the internal and external networks, all other packets should be prohibited.

Text

Description automatically generated

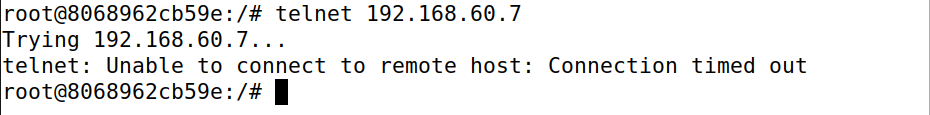
**Task 2.C: Protecting Internal Servers**

Table

Description automatically generated

* A telnet server is often used by internal hosts (listening to port 23). Only the telnet server on 192.168.60.5 is accessible from the outside, not the other internal hosts.
* Telnet could only connected to 192.168.60.5, not 192.168.60.6 or 192.168.60.7.

Text

Description automatically generated

* External hosts are unable to access internal servers.

Graphical user interface, text, application

Description automatically generated

* All internal servers are accessible to internal hosts.

Text, letter

Description automatically generated

* External servers are inaccessible to internal hosts.

Text

Description automatically generated with medium confidence

**Task 3: Connection Tracking and Stateful Firewall.**

**Task 3.A: Experiment with the Connection Tracking.**

* ICMP

A screenshot of a computer

Description automatically generated with medium confidence

The command > conntrack –L resulted in 0 flow entries on the router at first. To track the ICMP connection, use > ping 192.168.60.5 to send ICMP packets.

On the tracking connection, 1 flow entry was found after the execution. In about 7-10 seconds, the connection state was retained.

* UDP

Text

Description automatically generated





* TCP

Text

Description automatically generated with low confidence

**Task 3.B: Setting Up a Stateful Firewall**

* The outside host can telnet only 192.168.60.5.
* Internal hosts can telnet outside hosts.

**Differences between the two approaches.**

* Stateful firewalls may track and protect depending on traffic patterns and flows by monitoring and identifying all traffic states on a network. On the other hand, stateless firewalls focus solely on individual packets and filter traffic using pre-defined rules.
* One of the most significant drawbacks of stateful firewalls is "connection" dependent. To put it another way, stateful firewalls collect a lot of security data based on the connection and its state (i.e., the logical port assigned to the service being used). The issue here is that many modern programs can (and frequently do) use several ports depending on the services they provide. They may also use non-standard ports or switch ports in the middle of a session.
* Next-generation firewalls go beyond connection-based traffic inspection to allow you to focus on inspecting apps themselves. When reviewing traffic by application, they also allow you to combine several security services such as web filtering and intrusion prevention.

**Task 4: Limiting Network Traffic**

ping 192.168.60.5 from 10.9.0.5

Table

Description automatically generated with medium confidence

A picture containing table

Description automatically generated

**Task 5: Load Balancing**

* **The nth mode is used (round-robin)**

iptables -t nat -A PREROUTING -p udp —dport 8080 -m statistic —mode nth —every 3 —packet 0 -j DNAT —to-destination 192.168.60.5:8080

> iptables -t nat -A PREROUTING -p udp —dport 8080

The first packet of every three is routed to 192.168.60.5:8080.

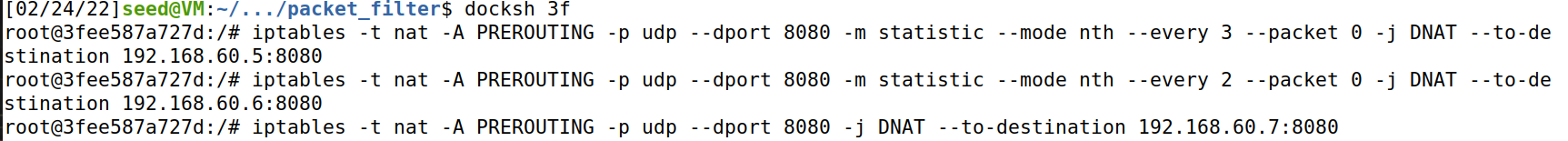
> iptables -t nat -A PREROUTING -p udp —dport 8080 -m statistic —mode nth —every 2 —packet iptables -t nat -A PREROUTING -p udp —dport 8080 -m statistic —mode nth —every 2 —packet iptables -t 0 -j DNAT —to-destination 0 -j DNAT —to-destination 0 -j DNAT 192.168.60.6:8080

The first packet of every two is routed to 192.168.60.6:8080.

> iptables -t nat -A PREROUTING -p udp —dport 8080 -j DNAT —to-destination iptables -t nat -A PREROUTING -p udp —dport 8080 -j DNAT —to-destination

192.168.60.7:8080

One out of every three packets will be sent to 192.168.60.7:8080.



Table

Description automatically generated

A picture containing chart

Description automatically generated

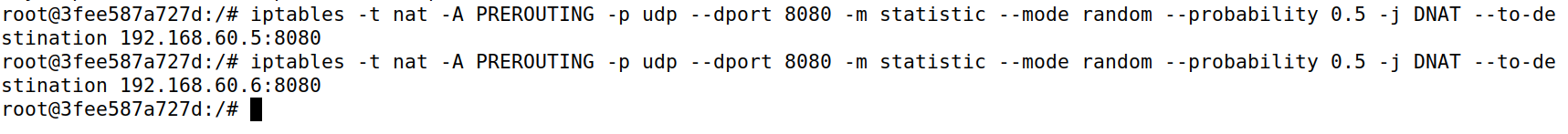
Text

Description automatically generated with medium confidence

A picture containing chart

Description automatically generated

**Using the random mode.**



Text, letter

Description automatically generated

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated