VICTORIA UNIVERSITY OF WELLINGTON TE HERENGA WAKA

CYBR371: System and Network Security

2024–T1: Arman Khouzani, Mohammad Nekooei

Assignment 2 – V3 (15%): Network Security

Submission Deadline: ***23:59:00 (NZST) on Sunday, 02 June 2024***

- There are two parts for a total of 150 points. Since this assignment has 15% contribution, think of each 10 points contributing 1% to your overall grade.
- you should upload **a single PDF file** (12 pages max in total, excluding references) containing your answers to questions in the order they appear on this document.
- The answer to each question must start on a new page, properly marked showing which question. If you decide to skip a question, you must still label it on your document and leave the answer blank. **Do not include the question statement in your report.**
- For screenshots, do not use a phone to snap a shot, use **`PrintScreen`**!
- Since **`NetLab`** does not allow saving, and is time-limited, it is recommended that you run a Linux VM on your own machine. Just beware: if you are using the lab machines, move your image file to **`/local/scratch/[yourname]`** directory, which saves it to that local machine. This is because you have a limited disk quota on your home directory, and also your VM would run faster.
- Recall that you can ask for help from tutors during lab times. However, never, under any circumstances, share your answers. **There is zero tolerance toward plagiarism.** If you are in doubt what constitutes plagiarism, ask me!

Question:	1	2	3	4	5	6	7	Total
Points:	10	15	15	15	30	15	50	150
Score:								

Part A: Network Attacks and Vulnerabilities

Q.1 [10 points] Write a Scapy script which monitors ICMP requests and replies back with packets simulating a Windows host.

Q.2 [15 points] Write a Scapy script which illustrates the ICMP Teardrop attack. The attack must use 8 packets with overlapping offsets. Include the script in your submission document, explain the code and the fragmented and overlapping bits. You must also illustrate the attack on the receiving end (i.e. target host) using any packet capturing, analysis or detection tools.

Q.3 [15 points] Explain the Xmas Tree attack and write a Scapy script to deploy this attack on a target host.

Q.4 [15 points] (a) Explain the term "Backscatter traffic" and why it is generated by some but not all types of Distributed Denial of Service (DDoS) attacks.

(b) Explain how backscatter traffic can be used to secure a network.

Q.5 [30 points] Demonstrate the DoS amplification attack through two different methods. In partcular, for each method:

(a) Provide a Scapy code that can launch the attack.

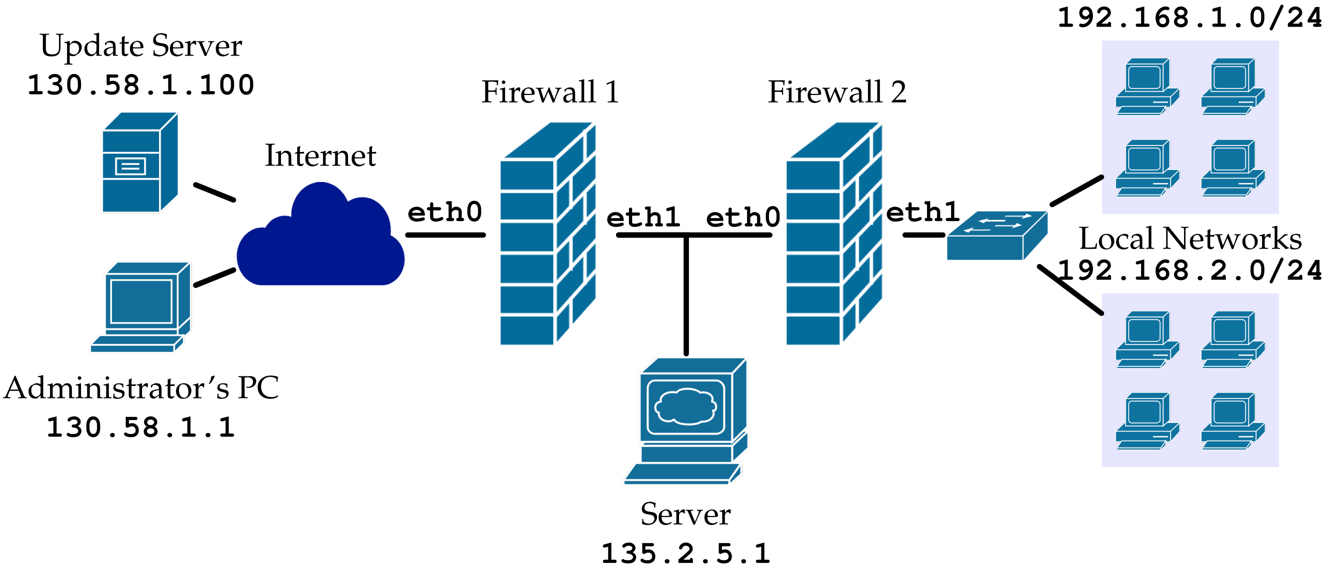
(b) Demonstrate its working by showing what is sent by the attacker and what is received at the target.

(c) Compute the amplification ratio in your practical example.

Part B: Firewalls and Intrusion Detection Systems

Q.6 [15 points] Explain the capability and the process (i.e. procedure/steps) by which some firewalls, intrusion prevention systems and honeypots can use the TCP protocol properties such as Window Size and Maximum Segment Size to slow down (NOT stop) the propagation of worms across the networks.

Q.7 As a system/network engineer you have been asked to create a firewall ruleset for a DMZ. The DMZ topology is depicted below:



The server offers the following services and characteristics:

- Operating system: Ubuntu 22.04 LTS
- Server's IP address: `135.2.5.1`
- Services: **ICMP**, **SSH**, and **Apache**.

Our two network firewalls (firewall 1 and firewall 2) are each just a Linux machine. Each have two network interface cards (`eth0` and `eth1`):

- Firewall 1: its interface `eth0` is facing the internet (WAN), and its interface `eth1` is facing the DMZ.
- Firewall 2: its interface `eth0` faces the DMZ and its interface `eth1` is facing the LANs.

Other Information:

- Clients' networks: `192.168.1.0/24`, `192.168.2.0/24`
- Update server: `130.58.1.100`, Port `4119`

For simplicity, you can ignore any NAT taking place between private and public IP addresses – bonus mark if you take those also into account!

You can also assume that proper routing tables are set in each of the firewalls, so based on the destination IP address of packets, they get routed to the correct interface.

Requirements:

A. Provide service for HTTP and HTTPS requests for all clients within the internal and external networks. Drop inbound traffic to port 80 (http) from source ports less than 1024.

B. Protect the server against ICMP ping flooding from external network.

C. Protect the server against UDP Fraggle attacks from anywhere.

D. Provide remote SSH service for administrator from the remote system with an IP address of `130.58.1.1`.

E. Protect the server against SSH dictionary attack from anywhere.

F. Protect the server against Xmas Tree attack from external network.

G. Drop all incoming (i.e. inbound) packets from reserved ports `135` and `139` from anywhere as well as all outbound traffic to these ports.

H. Redirect all the DNS requests from your internal network to Google's `8.8.4.4` IP address and associated port.

I. The server is not allowed to create any new outgoing connections to any networks, except to download security updates from the Update Server.

J. There is a new worm outbreak! The worm targets the TCP 8080 or UDP port 4040 and contains the signature "AC 1D 1C C0 FF EE" (hex) follow by "PASS : CYBR371" (ascii) within the first 40 bytes. The worm is coming from external network targeting the server.

K. Any machine on the LAN should be able to access the Internet (all the Internet has to offer, not just the www!).

L. ICMP services should be available for normal usage of our server from anywhere. For instance, anyone should be able to ping our server.

(a) [20 points] Create firewall policy tables for network "Firewall 1" and network "Firewall 2" with the given information. Use the template below (example only). The policies must be complete, specific and, take into account the bidirectional nature of the connections. The rules must filter the traffic accurately, must not cause denial of service to legitimate hosts and must be immune to evasion by attackers. Incomplete policies will not be assigned any marks.

In the first column of the table, you can specify the letter(s) indicating which requirement(s) it is specifically related to (A through K).

Req	Proto.	Direction	Src. IP/Net	Dest. IP/Net	Src. Port	Dest. Port	Action
?	TCP	LAN→DMZ	192.168.1.0/24 192.168.2.0/24	135.2.5.1	any	23 (Telnet)	Allow New, Established
	TCP	DMZ→LAN	135.2.5.1	192.168.1.0/24 192.168.2.0/24	23 (Telnet)	any	Allow Established
?							

Hint: *For your convenience, I have provided the two tables (one for each of the firewalls) for requirements K and L below (assuming there were no other requirements):*

Table 1: Firewall 1 (between Internet (WAN) and DMZ)

Req	Proto.	Direction	Src. IP/Net	Dest. IP/Net	Src. Port	Dest. Port	Action
K	any	DMZ→WAN	192.168.1.0/24 192.168.2.0/24	any	any	any	Allow New, Established, Related
	any	WAN→DMZ	any	192.168.1.0/24 192.168.2.0/24	any	any	Allow Established, Related
L	ICMP	WAN→DMZ	any	135.2.5.1	any	any	Allow echo-request
	ICMP	DMZ→WAN	135.2.5.1	any	any	any	Allow echo-reply
ALL	any	any	any	any	any	any	Drop

Table 2: Firewall 2 (between DMZ and LAN)

Req	Proto.	Direction	Src. IP/Net	Dest. IP/Net	Src. Port	Dest. Port	Action
K	any	LAN→DMZ	192.168.1.0/24 192.168.2.0/24	any	any	any	Allow New, Established, Related
	any	DMZ→LAN	any	192.168.1.0/24 192.168.2.0/24	any	any	Allow Established, Related
L	ICMP	LAN→DMZ	192.168.1.0/24 192.168.2.0/24	135.2.5.1	any	any	Allow echo-request
	ICMP	DMZ→LAN	135.2.5.1	192.168.1.0/24 192.168.2.0/24	any	any	Allow echo-reply
ALL	any	any	any	any	any	any	Drop

(b) [15 points] Write the appropriate set of iptables (netfilter) rules to fulfil the requirements for each firewall. The iptables rules must be complete, specific and, take into account the bidirectional nature of the connections. The rules must filter the traffic accurately, must not cause denial of service to legitimate hosts and must be immune to evasion by attackers. The iptables rules must match the order of the policy table rules. Incomplete rules will not be assigned any marks.

Hint 1: *Note that there are no services on the firewall machines. So there should be no requests targeting the firewalls. Their machines are also not meant to be used by any users on them. So you can safely assume no traffic originates from or is targeted at the firewall. With the simplifying assumption of no NAT taking place between private and public IP addresses, you can assume no rule needs to be written for the* ***`INPUT`*** *and* ***`OUTPUT`*** *chains. So almost all of your rules will be on the* ***`FORWARD`*** *chain and on the (default)* ***`filter`*** *table. There is one obvious exception for one of the requirements!*

Hint 2: *Some of you had (mistakenly) assumed that the* ***`INPUT`*** *and* ***`OUTPUT`*** *chains are for ingress and egress traffic to and out of our LAN. That is not the case: the* ***`INPUT`*** *chain is for the traffic that terminates in the machine, because its IP address matches that of the machines, and the* ***`OUTPUT`*** *chain is for traffic that originates from the machine. This is nothing new, as per Hint 1. This hint is addressing this question: how do we then specify the "direction of traffic" in the* ***`iptables`*** *rules? Simple: by specifying the "interface" that a packet has appeared on. For instance, for firewall 2, a packet that has appeared on interface* ***`eth1`*** *is part of egress (outgoing) traffic, and a packet that has appeared on* ***`eth0`*** *is part of ingress (incoming) traffic.*

Hint 3: *For your convenience, I have provided the* **`iptables`** *commands, assuming if the only requirements were K and L. Note the following:*

- *For the hosts on the LAN to be able to connect to the internet, their requests (and their corresponding responses) must be allowed to pass both firewall 1 and firewall 2, in our configuration.*
- *Also, keep in mind that we are told we can ignore any address translation taking place between private and public IP addresses of the LAN, so we can have our rules based on their IP addresses as is.*
- *We can specify the direction of traffic by specifying the input interface of each rule. Although it is sufficient to just specify the input interface, I have chosen to also specify the output interface, because so long as security is concerned, it doesn't hurt to tighten the rules as much as possible (and performance is not our concern here!).*
- *The state matching option is typically more relevant for connection-oriented protocols like TCP. For ICMP traffic, especially simple echo requests and replies, the state tracking is less critical. We can simplify the ICMP rules by not specifying the state.*

Firewall 1 (between Internet and DMZ)

```
# Flush all rules from all chains
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -t raw -F

# Delete all user-defined chains in all tables
iptables -X
iptables -t nat -X
iptables -t mangle -X
iptables -t raw -X

# Set default policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP


# Allow LAN subnets to access the internet (req. K)
iptables -A FORWARD -i eth1 -s 192.168.1.0/24 -o eth0 -j ACCEPT
iptables -A FORWARD -i eth1 -s 192.168.2.0/24 -o eth0 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -d 192.168.1.0/24 -m state --state ESTABLISHED,RELATED
    -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -d 192.168.2.0/24 -m state --state ESTABLISHED,RELATED
    -j ACCEPT


# Allow ICMP (ping) traffic to the webserver and their responses (req. L)
iptables -A FORWARD -i eth0 -o eth1 -p icmp --icmp-type echo-request -d 135.2.5.1 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -p icmp --icmp-type echo-reply -s 135.2.5.1 -j ACCEPT
```

Firewall 2 (between DMZ and LAN)

```
# Flush all rules from all chains
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -t raw -F

# Delete all user-defined chains in all tables
iptables -X
iptables -t nat -X
iptables -t mangle -X
iptables -t raw -X

# Set default policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# Allow LAN subnets to access the internet (req. K)
iptables -A FORWARD -i eth1 -s 192.168.1.0/24 -o eth0 -j ACCEPT
iptables -A FORWARD -i eth1 -s 192.168.2.0/24 -o eth0 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -d 192.168.1.0/24 -m state --state ESTABLISHED,RELATED
    -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -d 192.168.2.0/24 -m state --state ESTABLISHED,RELATED
    -j ACCEPT

# Allow ICMP (ping) from LAN to DMZ (req. L)
iptables -A FORWARD -i eth1 -o eth0 -p icmp --icmp-type echo-request -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -p icmp --icmp-type echo-reply -j ACCEPT
```

Some final points:

- *It will also be correct (and maybe even preferable) if you choose* ***`-m conntrack --cstate`*** *instead of* ***`-m state --state`*** *in the rules that specify the state.*
- *Also, while it is critical to specify the state in the traffic going to the LAN (they should only be part of* ***`ESTABLISHED`*** *or* ***`RELATED`*** *sessions), I have chosen not to specify the state of the outgoing traffic from the LAN. It will of course be also correct (and perhaps even preferable) if you specify the state for the outgoing traffic as well. So for instance, the rule:*

  ```
  iptables -A FORWARD -i eth1 -s 192.168.1.0/24 -o eth0 -j ACCEPT
  ```

  would be:

  ```
  iptables -A FORWARD -i eth1 -s 192.168.1.0/24 -o eth0 -m state --state NEW,ESTABLISHED
      ,RELATED -j ACCEPT
  ```

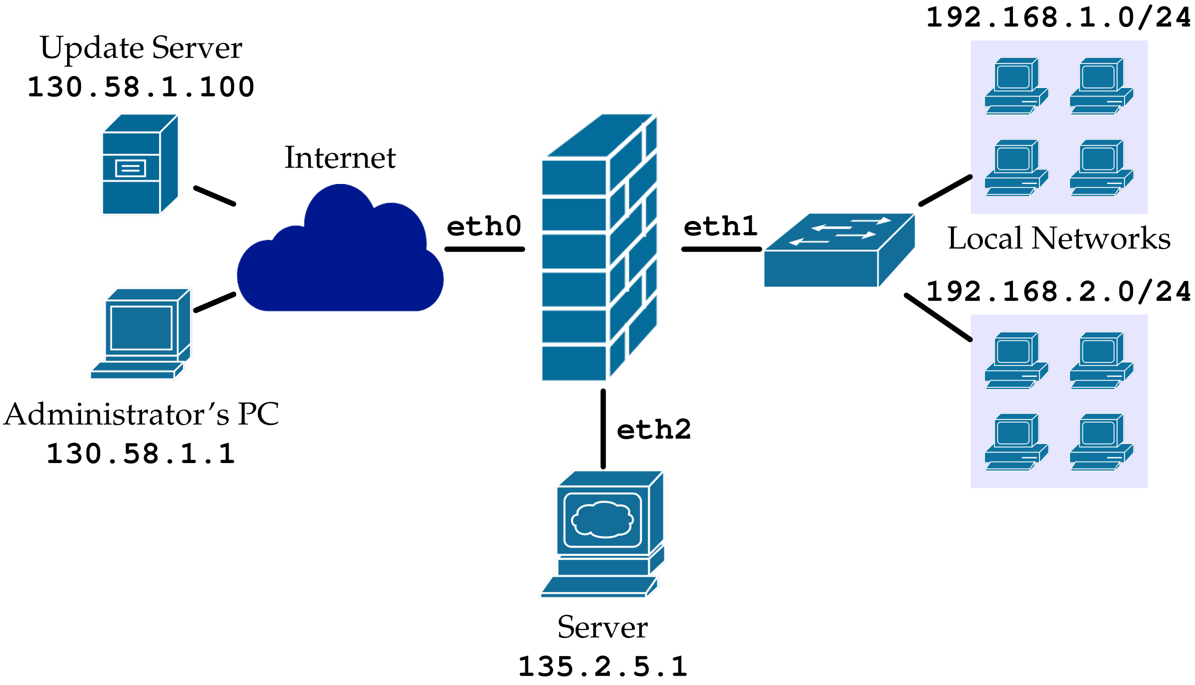
- *We could further tighten the rules for allowing hosts on the LAN to access the internet by ensuring only client-side kind of applications initiate the connections from there. Client programs typically use 'ephemeral' ports (ports 1024 and above) for outgoing connections. So, for instance, we could have added* *`--sport 1024:65535`* *to the rule:*

  ```
  iptables -A FORWARD -i eth1 -s 192.168.1.0/24 -o eth0 -j ACCEPT
  ```

  and so on. However, doing so may have some unintended consequences! For instance, if our staff (in our LAN) want to use a tunnelling application that uses the GRE (Generic Routing Encapsulation) protocol, or they want to use a VPN application that uses IPSEC, then they will have difficulty, because those protocols *don't even use port numbers, unlike TCP, UDP, SCTP (Stream Control Transmission Protocol), etc. Even ICMP does not use port numbers (it uses message types in its header, if you recall!). So it is better to avoid it.*

- *Finally, for firewall 2, the rule that allowed access to internet "technically" covers pinging to the server as well (because we did not specify the protocol in the rule, so it applies to all protocols). However, it is good practice to specify protocols explicitly for clarity and to avoid unintended consequences. Adding specific rules for ICMP traffic can help ensure the network behaves as expected, particularly if other rules are added later that might affect different protocols differently.*

(c) [15 points] Suppose instead of two separate firewalls, we use the following (3-legged) DMZ topology instead:



Write the appropriate set of iptables rules that fulfil the same requirements as before. You can make the same assumptions as in the previous part, that is:

- For simplicity, you can ignore any NAT taking place between private and public IP addresses – bonus mark if you take those also into account!
- You can also assume that proper routing tables are set in each of the firewalls, so based on the destination IP address of packets, they get routed to the correct interface.