Packet Sniffing and Spoofing

CN\_Ex5

Ariel University

Connection networking

Maor Saadon 318532421

Duvi Amiram 305677494

**Contents**

1.1 About the System

1.1.1 Sniffing……………………………………………………………………..4-6  
1.1.2 Spoofing……………………………………………………………………..4-6

1.2 Task A

1.2.1 How to Install and Run the Program………………………….……..……..7  
1.2.2 Code Description……………………………………………………..…………8-19  
1.2.3 Question………………………………………………………………………….22-31  
1.2.4 Output…………………………………………………………………………………..21  
1.2.5 Wireshark……………………………………………………………………….32-33

1.3 Task B

1.3.1 How to Install and Run the Program………………………….……..……..7  
1.3.2 Code Description……………………………………………………..…………8-19  
1.3.3 Question………………………………………………………………………….22-31  
1.3.4 Output…………………………………………………………………………………..21  
1.3.5 Wireshark……………………………………………………………………….32-33

1.4 Task C

1.4.1 How to Install and Run the Program………………………….……..……..7  
1.4.2 Code Description……………………………………………………..…………8-19  
1.4.3 Question………………………………………………………………………….22-31  
1.4.4 Output…………………………………………………………………………………..21  
1.4.5 Wireshark……………………………………………………………………….32-33

1.5 Task C

1.5.1 How to Install and Run the Program………………………….……..……..7  
1.5.2 Code Description……………………………………………………..…………8-19  
1.5.3 Question………………………………………………………………………….22-31  
1.5.4 Output…………………………………………………………………………………..21  
1.5.5 Wireshark……………………………………………………………………….32-33

1.6 Bibliography………………………………………………………………………..……………..48

**1.1 About the System**

1.1.1 Sniffing

What are packet sniffers?

Packet sniffers are applications or utilities that read data packets traversing the network within the Transmission Control Protocol/Internet Protocol (TCP/IP) layer. When in the hands of network administrators, these tools “sniff” internet traffic in real-time, monitoring the data, which can then be interpreted to evaluate and diagnose performance problems within servers, networks, hubs and applications.

When packet sniffing is used by hackers to conduct unauthorized monitoring of internet activity, network administrators can use one of several methods for detecting sniffers on the network. Armed with this early warning, they can take steps to protect data from illicit sniffers.

1.1.1 Spoofing

Packet spoofing or IP spoofing is the creation of Internet Protocol (IP) packets having a source IP address with the purpose of concealing the identity of the sender or impersonating another computing system. A spoofing attack occurs when a malicious party impersonates another device or user on a network in order to launch attacks against network hosts, steal data, spread malware, or bypass access controls.

The attacker creates an IP packet and sends it to the server, which is known as an SYN (synchronize) request. The attacker puts own source address as another computer’s IP address in the newly created IP packet. The server responds back with a SYN ACK response, which travels to the forged IP address. The attacker receives this SYN ACK response sent by the server and acknowledges it so as to complete a connection with the server. Once this is done the attacker can try various commands on the server computer. The most common methods include IP address spoofing attacks, ARP spoofing attacks, and DNS server spoofing attacks. Common measures that organizations can take for spoofing attack prevention include packet filtering, using spoofing detection software, and cryptographic network protocols.

**1.2 Task A**

1.2.1 How to Install and Run the Program

1.2.2 Code Description

1.2.3 Question

A sniffer program typically needs to be run with root (or administrator) privilege because it needs access to the network interface in order to capture and analyze network traffic. Without this access, the program would not be able to intercept and read the packets of data traveling over the network.

When a program is executed without root privilege, it may fail to capture network traffic because it does not have the necessary permissions to access the network interface. Additionally, the program may also be unable to access certain system-level functions or resources that it needs in order to perform its intended tasks.

Alternatively, some sniffer programs can be run with normal user privilege if they are run in promiscuous mode on a switched network, but they may only see the traffic that is directed to them by the switch.

1.2.4 Output

1.2.5 Wireshark

**1.3 Task B**

1. The IP packet length field is used to indicate the total length of an IP packet, including both the header and the data payload. The value of this field is typically set automatically by the operating system or network device when the packet is created.

It is not generally recommended to set the IP packet length field to an arbitrary value, as doing so could lead to malformed packets and result in communication errors. The actual size of the packet should match the value in the length field.

However, it's possible that some network equipment or software may allow for the modification of the IP packet length field, but it's not a common practice and it could be considered as a malicious activity, as it may cause problems for network communication.

2. When using raw socket programming to create and send IP packets, the checksum for the IP header is typically calculated automatically by the operating system or network device. This is done to ensure that the packet is not corrupted during transmission.

However, it is possible to manually calculate and set the IP header checksum when using raw socket programming, but it depends on the programming language and Operating system you are using. For example, in C/C++ on Linux, you can use the "IP\_HDRINCL" option to tell the operating system that you will be including your own IP header and not to calculate the checksum.

It's important to note that, if you choose to manually calculate the IP header checksum, it must be done correctly and the checksum value must be updated if the header is modified in any way.