

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering –
Data Science

By

ESWAR.M

(40733010)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMIL NADU**

NOV 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ESWAR.M (40733010)** who carried out the project entitled “**Network sniffer and Spoofer**” under my supervision from Feb 2023 – April 2023

Internal Guide Name

Dr.U.ANUBHARATHI

Head of the Department

Dr.L.LAKSHMANAN M.E., Ph.D.,

Submitted for Viva voce Examination held on **17-04-2023**

Internal Examiner

External Examiner

DECLARATION

I **ESWAR.M** hereby declare that the Project Report Entitled **NETWORK SNIFFER AND SPOOFER** done by me under the guidance of **Dr.U.ANUBHARATHI** and **MR SRIRAM KESAVAN** at **TGCYBERLABS** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering-Data Science.

DATE: 17-04-2023



PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board **of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.U.ANUBHARATHI** for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

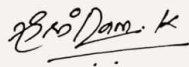
TRAINING CERTIFICATE

CERTIFICATE OF COMPLETION

This certificate is awarded to:

Eswar

has completed the Professional Training Program conducted by
Techgreech Cyberlabs Private Limited from 14/02/2023 - 28/04/2023



Mr. Sriram
Director, TG Cyberlabs

TG Cyberlabs.
Security our Priority...

Cert-ID: TGCYBLS-PT2-0003

ABSTRACT

Network sniffer is a program and/or device that monitor data traveling over a network. Network sniffers can be used both for legitimate network management functions and for stealing information off a network. Network operations and maintenance personnel may use network sniffers to monitor network traffic, analyze packets, watch network resource utilization, conduct forensic analysis of network security breaches and troubleshoot network problems. Unauthorized network sniffers can be extremely dangerous to a network's security because they are virtually impossible to detect and can be inserted almost anywhere. This makes them a favorite weapon in the hacker's arsenal. Typical use of network sniffer is to analyze network traffic, so that underlying troubles in network can be found out. However, two directional usages of sniffer have coexisted since it was produced

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	INTRODUCTION	7
	1.1 What is network	7
	1.2 What network sniffing	7
	1.3 Types of network sniffers	8
	1.4 Uses of network sniffers	8
1	1.5 Difference between sniffing and spoofing	9
	1.6 Protection against Sniffing and Spoofing	9
	SYSTEM ANALYSIS	10
2	2.1 Aim & Scope	10
	2.2 How does it work	10
	SYSTEM DESIGN AND IMPLEMENTATION	12
3	3.1 Algorithm	12
	3.2 Modules used	12
	RESULT & DISCUSSION	13
	4.1 Result	13
4	4.2 Discussion	14
	4.3 Source Code	15
5	CONCLUSION	16

CHAPTER 1

INTRODUCTION

1.1 What is network

A computer network is a system that connects numerous independent computers in order to share information (data) and resources. The integration of computers and other different devices allows users to communicate more easily.

A computer network is a collection of two or more computer systems that are linked together. A network connection can be established using either cable or wireless media. Hardware and software are used to connect computers and tools in any network.

A computer network consists of various kinds of nodes. Servers, networking hardware, personal computers, and other specialized or general-purpose hosts can all be nodes in a computer network. Hostnames and network addresses are used to identify them.

1.2 What is Network Sniffing

Sniffing is the technique used to monitor and record all data packets continuously that go through a network. Network/system administrators employ sniffers to monitor and troubleshoot network traffic. Attackers use sniffers to capture data packets carrying sensitive passwords and account information. Sniffers are implemented as hardware or software in the system. A hostile intruder can gather and analyze all network traffic by using a packet sniffer in promiscuous mode on a

network.

A packet sniffer is another term for a network sniffer. Because every packet of data is sniffed through the network to avoid network-related issues, it's called a packet sniffer. The packet sniffer tool is implemented to investigate cybercrime, hackers, and data theft. It can be employed for both ethical and unethical reasons.



1.3 Types of Network Sniffers

Following are the different types of Network Sniffers –

Mac sniffers – Sniffers are used to sniff data relevant to the MAC address filter.

Protocol sniffer – It sniffs the data on the network for network protocols.

LAN sniffer – This type of device is primarily employed in internal systems or networks, and it can inspect an entire range of IP addresses.

IP sniffers – Sniff all data relevant to a specific IP filter. It records the data packets for analysis and diagnosis. IP sniffers capture network traffic and log the

information, generally delivered in a human-readable format for analysis. They may be used by network administrators and hackers of all stripes to assess the current condition of a network, identify network vulnerabilities, and evaluate network performance.

ARP sniffers – Rather than sending packets to the host only and passed to the network administrator, packets are sent to the ARP caches of both network hosts in this sniffing. It also allows attackers to map IP addresses to MAC addresses, carrying out packet spoofing and other vulnerabilities or poisoning attacks.

Password sniffers – It is a technique for extracting information from network traffic to harvest passwords. Hackers used to target sessions to steal credentials and other information. Websites that don't have an SSL protocol encryption to protect themselves are vulnerable to attack and exploitation.

1.4 Use of Network Sniffers

Hackers primarily employ network sniffers to gather information on passwords and other sensitive information. The sniffer decodes data in packets traveling from source to destination, between client and server, or between organizations. They functioned as middlemen and employed a packet injection attack to grab the data. For example, a network sniffer can track down someone using too much bandwidth at a university or company by monitoring network traffic. They are also used to detect security vulnerabilities in our system.

Today, however, black hat hacking is a widespread application for them. In the wrong hands, network sniffing tools can allow anyone with little to no hacking expertise to monitor network traffic across unsecured WiFi networks to steal passwords and other sensitive data. This reason can give network sniffing tools a bad name, yet network sniffers have many valid purposes.

1.5 Difference Between Sniffing and Spoofing

Sniffing takes place when an attacker collects data packets that pass over a network by utilizing packet sniffers and data traffic in the network.

In contrast to Sniffing, Spoofing happens when an attacker steals a user's rights and uses them to acquire legitimate user access to a system to execute attacks against network hosts, steal data, distribute malware, or evade access controls.

Spoofing is when an attacker creates TCP/IP using another person's IP address. A sniffer software is placed between two interactive endpoints in packet Sniffing, where the attacker pretends to be one end of the connection to the target and snoops on data sent between the two points.

1.6 Protection against Sniffing and Spoofing

Sniffing	Spoofing
• Set up a strong antivirus on your device	• Implement packet filtering with deep packet inspection
• Secure your data with a VPN	• Verify the authenticity of users and systems
• Avoid visiting unencrypted websites	• Use Spoofing detection software

CHAPTER 2

SYSTEM ANALYSIS

2.1 Aim & Scope

Aim of this project is to develop an network sniffer tool which can be used in the real world network security so that the vulnerability caused by the unchecked network can be fixed by the organization/company before black hat hacker finds the issue and exploit it

Scope of this project is to scan the Ip network of a particular device and checking how many device are connected to a particular device along with its IP and mac address

2.2 How does it work

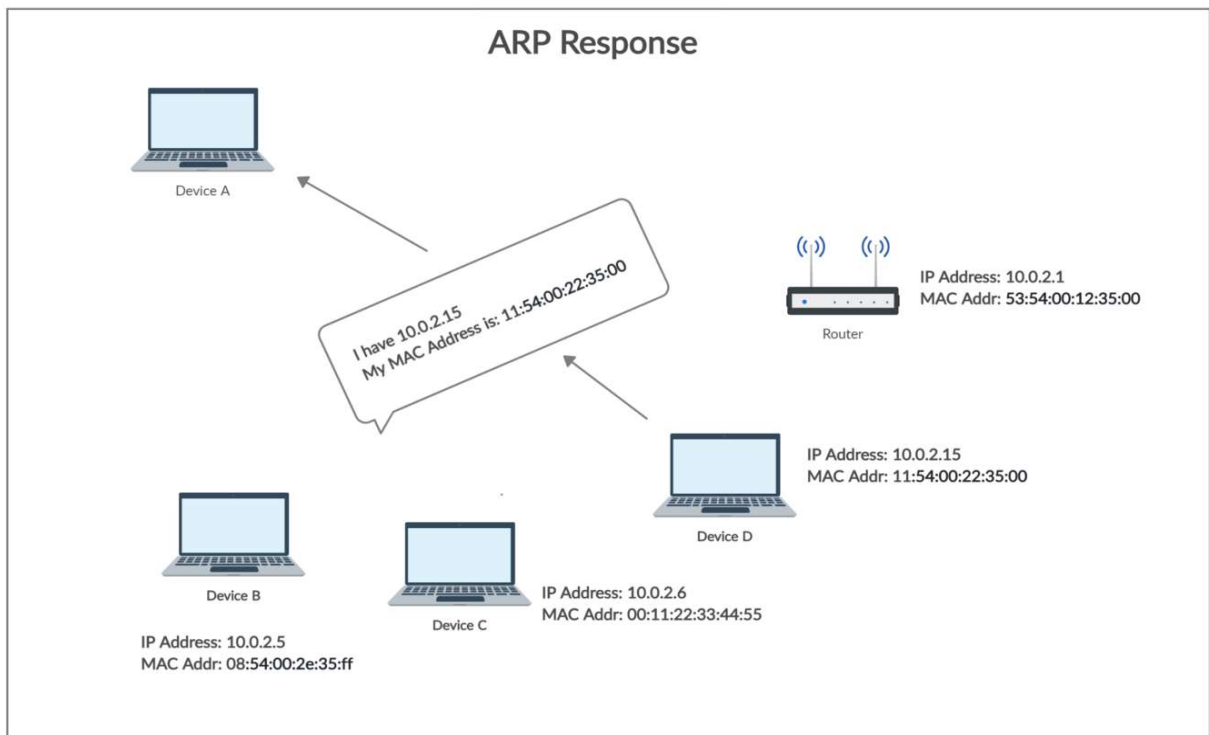
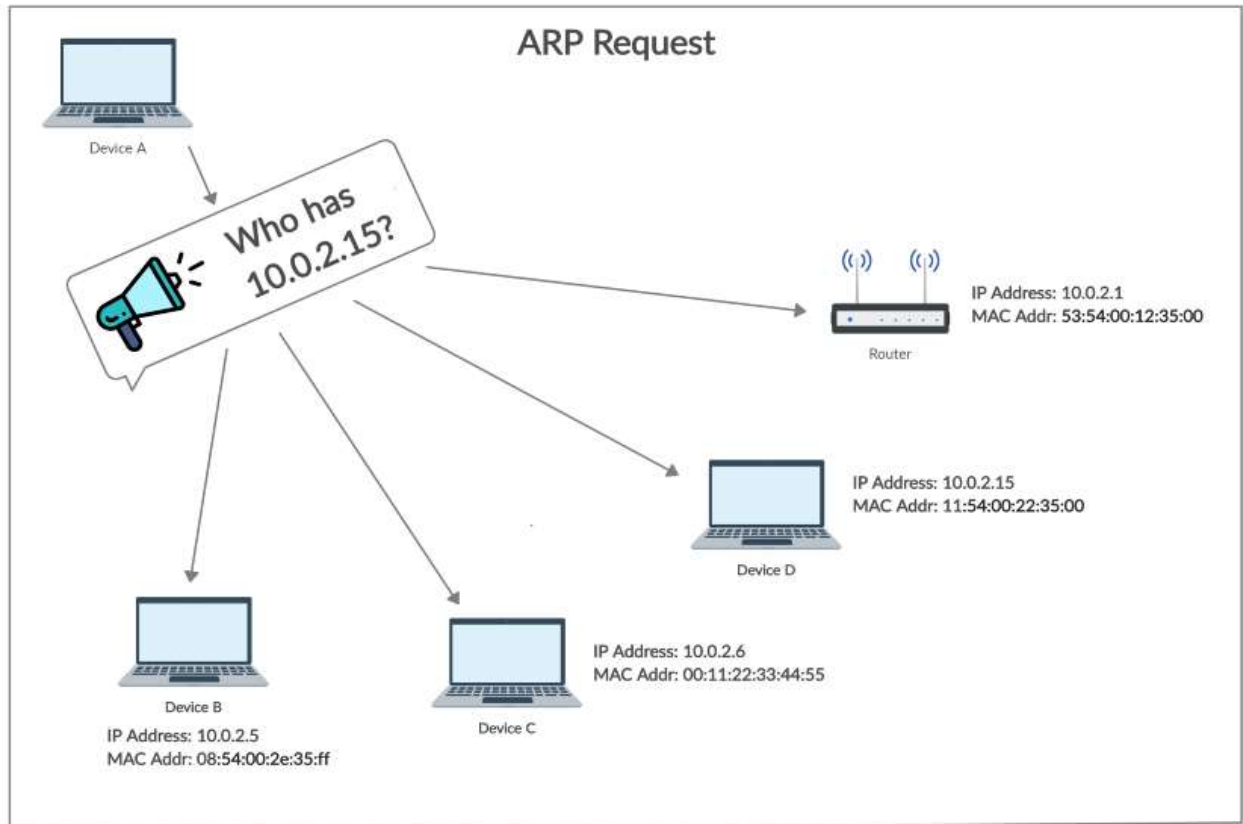
To understand how the Network Scanner scans the entire network we need to first understand what is ARP (Address Resolution Protocol).

In a network, most of the computers use the IP Address to communicate with other

devices, however, in reality, the communication happens over the MAC Address. ARP is used to find out the MAC Address of a particular device whose IP address is known. For instance, a device wants to communicate with the other device on the network, then the sending device uses ARP to find the MAC Address of the device that it wants to communicate with. ARP involves two steps to find the MAC address:

1. The sending device sends an ARP Request containing the IP Address of the device it wants to communicate with. This request is broadcasted meaning every device in the network will receive this but only the device with the intended IP address will respond.
2. After receiving the broadcast message, the device with the IP address equal to the IP address in the message will send an ARP Response containing its MAC Address to the sender.

Network Scanner uses ARP Request and Response to scan the entire network to find active devices on the network and also to find their MAC Addresses.



CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 Algorithm

Step 1: Import the modules.

Step 2: Implementing the functionality to allow the users to pass command line arguments.

Step 3: Writing function that scans the network

1. Create an ARP Request.
2. Create an Ethernet Frame.
3. Place the ARP Request inside the Ethernet Frame.
4. Send the combined frame and receive responses.
5. Parse the responses and print the results.

Step 4: Writing function to print the results in a certain format.

3.2 Modules used

1. **argparse**: To understand what this does read my first article hereThe argparse module makes it convenient to write user-friendly command-line interfaces. The program defines the arguments it requires and argparse will figure out how to parse these arguments. It also generates help and usage messages and issues errors when the user enters any invalid arguments..
2. **Scapy**: Enables the user to send, sniff and dissect and forge network packets.

This capability allows the development of tools that can probe, scan, or attack networks. It can forge or decode packets of a wide number of protocols, send them on the wire, capture them, match requests and replies, and much more. It easily handles most tasks like scanning, tracerouting, probing, unit tests, attacks, or network discovery.

CHAPTER -4

RESULTS & DISCUSSION

4.1 Results

This tool was successfully developed and tested on the real world target and managed to work perfectly without any false positive till now from the date of development of the tool this working is really smooth and fast in speed which can be

used for real world Network security

```
kali-linux-2022.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali:~/tcyberlabs
ls
kali@kali:~/tcyberlabs
$ cat > network_scanner.py
^C
kali@kali:~/tcyberlabs
$ python3 network_scanner.py
mousepad network_scanner.py
kali@kali:~/tcyberlabs
$ python3 network_scanner.py -t 192.168.1.3/24
kali@kali:~/tcyberlabs
$ mousepad network_scanner.py
^C
kali@kali:~/tcyberlabs
$ sudo python3 network_scanner.py -t 192.168.1.3/24
[sudo] password for kali:
IP Address      MAC Address
192.168.1.3     e8d4e8365118c
192.168.1.1     789a2e2a1418
192.168.1.2     74d21d5f560ea
192.168.1.4     bece0d43c336a
kali@kali:~/tcyberlabs
$ python3 network_scanner.py -t 192.168.1.3/24
Traceback (most recent call last):
  File "/home/kali/tcyberlabs/network_scanner.py", line 39, in <module>
    scanned_output = scan(options.target)
  File "/home/kali/tcyberlabs/network_scanner.py", line 24, in scan
    answered_list = scapy.srp(broadcast_ether_arf_req_frame, timeout = 1, verbose = False)[0]
  File "/usr/lib/python3/dist-packages/scapy/sendrecv.py", line 352, in srp
    s = conf.L2socket(promisc=promisc, iface=iface)
  File "/usr/lib/python3/dist-packages/scapy/arch/linux.py", line 398, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.ntohs(type)) # noqa: E501
  File "/usr/lib/python3.10/socket.py", line 232, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
kali@kali:~/tcyberlabs
$ sudo python3 network_scanner.py -t 192.168.1.3/24
IP Address      MAC Address
192.168.1.3     e8d4e8365118c
192.168.1.1     789a2e2a1418
192.168.1.2     74d21d5f560ea
192.168.1.4     bece0d43c336a
kali@kali:~/tcyberlabs
$
```

```
Command Prompt
Connection-specific DNS Suffix . :
Ethernet adapter VirtualBox Host-Only Network:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::59ac:e3db:7d50:3072X16
IPv4 Address. . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wifi:

Connection-specific DNS Suffix . :
IPv6 Address. . . . . : 2401:4900:1cd1:29cc:4d43:e5d5:b305:a1ce
Temporary IPv6 Address. . . . . : 2401:4900:1cd1:29cc:81cd:2606:5b91:3d5c
Link-local IPv6 Address . . . . . : fe80::4d43:e5d5:b305:a1ceX7
IPv4 Address. . . . . : 192.168.1.3
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::1X7
192.168.1.1

C:\Users\vivek>
```

4.2 Discussion

From the above image, we can note two things about the Windows 10 :

IP Address = 10.0.2.15

MAC Address = 08-00-27-e6-e5-59

We can see that the network scanner script that we created scans the entire network including the Windows 10 which is in the same network.

4.3 Source Code

```
import scapy.all as scapy
import argparse
```

```
def get_args():
```

```
    parser = argparse.ArgumentParser()
```

```
    parser.add_argument('-t', '--target', dest='target', help='Target IP  
Address/Addresses')
```

```
    options = parser.parse_args()
```

```
    if not options.target:
```

```
        parser.error("[-] Please specify an IP Address or Addresses, use --help for more  
info.")
```

```
    return options
```

```
def scan(ip):
```

```
    arp_req_frame = scapy.ARP(pdst = ip)
```

```
    broadcast_ether_frame = scapy.Ether(dst = "ff:ff:ff:ff:ff:ff")
```

```
    broadcast_ether_arp_req_frame = broadcast_ether_frame / arp_req_frame
```

```

    answered_list = scapy.srp(broadcast_ether_arp_req_frame, timeout = 1, verbose =
False)[0]
    result = []
    for i in range(0,len(answered_list)):
        client_dict = {"ip" : answered_list[i][1].psrc, "mac" : answered_list[i][1].hwsrc}
        result.append(client_dict)

    return result

def display_result(result):
    print("-----\nIP Address\tMAC Address\n-----")
    for i in result:
        print("{}\t{}".format(i["ip"], i["mac"]))

options = get_args()
scanned_output = scan(options.target)
display_result(scanned_output)

```

CHAPTER -5

CONCLUSION

Sniffers are tools -- sometimes referred to as network analyzers -- commonly used for monitoring network traffic. The term *packet sniffing* refers to the technique of copying individual packets as they traverse a network. When used by systems administrators, network sniffers can be invaluable tools for diagnosing or troubleshooting network problems. When used by malevolent individuals, however, sniffers can also represent

a significant threat to your network. Unfortunately, they are sometimes hard to detect.

Years ago, sniffers were hardware devices that were physically connected to the network. More recently, advances in technology have allowed for the development of software sniffers. This brings the art of network sniffing to anyone who wants to perform this task.

One important, and disturbing, aspect of packet sniffers is their ability to place their host machine's network adapter into "promiscuous mode." When network adapters are in promiscuous mode, they receive not only the data directed to the machine hosting the sniffing software, but also all other data traffic on the physically connected local network. Because of this capability, packet sniffers have the potential to be used as powerful spying tools on company networks.

REFERENCE

1. <https://levelup.gitconnected.com/writing-a-network-scanner-using-python-a41273baf1e2>
2. <https://levelup.gitconnected.com/changing-mac-address-using-python-8a16fc4a3563#f878>
3. https://www.academia.edu/35960159/Project_Report_of_a_Packet_Sniffer_Program
4. <https://www.youtube.com/watch?v=-rSqbgI7oZM>
5. <https://www.youtube.com/watch?v=XPZH7il68cc&t=245s>
6. <https://www.computerworld.com/article/2569788/sniffing-out-network-sniffers.html>