

INFORMATION GATHERING SCRIPTED TOOL

A Project Work Synopsis

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

CSE IBM (Information Security)

Submitted by:

P.PRANAY RAO (17BCS3552)

R.SAI TEJA (17BCS3560)

J.ARAVIND (17BCS3506)

G.PAVAN KUMAR (17BCS3550)

P.SRI RAMAN (17BCS3558)

Under the Supervision of:

GURPREET SINGH PANESAR



**CHANDIGARH
UNIVERSITY**
Discover. Learn. Empower.

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

PUNJAB

MAY 2021

DECLARATION

I, **PINNINTY PRANAY RAO**, student of '**Bachelor of Engineering in Computer Science and Engineering**', session: **2020-2021__**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled '**Information Gathering Scripted Tool**' is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

P.PRANAY RAO

Candidate UID: 17BCS3552

Date:

Place:

Chandigarh University , gharuan

Annexure-4 (A typical specimen of table of contents)

Table of Contents

| | |
|---------------------------------|-----------|
| Title Page | i |
| List of Figures | ii |
| 1. INTRODUCTION* | 1 |
| Problem Definition | 1 |
| Project Overview/Specifications | 1 |
| Hardware Specification | 1 |
| Software Specification | 1 |
| 2. LITERATURE SURVEY | 2 |
| 3. PROBLEM FORMULATION | 3 |
| 4. RESERH OBJECTIVES | 5 |
| 5. METHODOLOGY | 9 |
| 6. RESULT | |
| 7. REFERENCES | 10 |

List of Figures

| <i>Table Title</i> | <i>page</i> |
|---|--------------------|
| <i>e</i> | |
| <i>3.1 Network Vulnerability</i> | <i>4</i> |
| <i>Domain and Subdomain</i> | <i>6</i> |
| <i>Common Port Numbers</i> | <i>8</i> |

1 INTRODUCTION

Information gathering scripting tool refers to the preparatory phase where a penetration tester seeks to gather as much information as possible about a target of evaluation prior to launching a penetration test. It involves three phases: footprinting, scanning, and enumeration of the network. In this research, we will be dealing with automating footprinting of an organization. Footprinting is the blueprint of the security profile of an organization, undertaken in a methodological manner. It discovers all information available about the target that is available through public domain sources. It is a time-consuming process to browse through web pages and collect information; hence in this paper, we investigate the problem of tedious web search and propose an efficient way to extract, organize and store data from search engines using a new command line tool search simplified.

Information gathering techniques can be roughly classified into the following:

- **Active:** This includes intrusive reconnaissance that sends (specifically crafted) packets to the targeted system, for example, port-scanning. Advanced networking enumeration techniques avoid direct communication with the targeted host.
- **Passive:** This includes reconnaissance that either does not communicate directly to the targeted system or that uses commonly available public information, not normally identifiable from standard log analysis.

This reconnaissance tool finds the subdomains of a domain, after finding the subdomains by vulnerability scanning we find the ports which are vulnerable. Having a unsecured subdomain can lead to a serious risk to website, there were some security incidents where the hacker used subdomain tricks to hack into the website.

Hardware Specifications:

- 4GB ram and 10GB laptop or computer
- Virtual Machine Support

Software Specifications

- Windows 10
- Kali Linux
- Python installed in linux

2 LITERATURE REVIEW

Ahana Roy; Louis Mejia Automation of cyber-reconnaissance: which is a conference paper which found the relevant information about Reconnaissance or information gathering which refers to the preparatory phase where a penetration tester seeks together as much as information as possible about a target of valuation prior to launching a penetration test. It involves three phases footprinting, scanning and enumeration of the network. In this research, we found that it is a automated footprinting tool for evaluating a target. Foot printing is the blueprint of the security profile of an organization. In this project we are going to implement where footprinting lead to finding sensitive information. And by other research paper and the process we have ensured that gathered detailed information where we can implement different methods which can lead to finding sensitive files.

Hence we have found this research paper useful and learned important things which will give results.

3 PROBLEM FORMULATION

In computer security, a vulnerability is a weakness which can be exploited by a threat actor, such as an attacker, to cross privilege boundaries (i.e. perform unauthorized actions) within a computer system. To exploit a vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this frame, vulnerabilities are also known as the attack surface.

Vulnerability management is the cyclical practice that varies in theory but contains common processes which include: discover all assets, prioritize assets, assess or perform a complete vulnerability scan, report on results, remediate vulnerabilities, verify remediation - repeat. This practice generally refers to software vulnerabilities in computing systems.

A security risk is often incorrectly classified as a vulnerability. The use of vulnerability with the same meaning of risk can lead to confusion. The risk is the potential of a significant impact resulting from the exploit of a vulnerability. Then there are vulnerabilities without risk: for example when the affected asset has no value. A vulnerability with one or more known instances of working and fully implemented attacks is classified as an exploitable vulnerability a vulnerability for which an exploit exists. The window of vulnerability is the time from when the security hole was introduced or manifested in deployed software, to when access was removed, a security fix was available/deployed, or the attacker was disabled see zero-day attack.

Security bug (security defect) is a narrower concept. There are vulnerabilities that are not related to software: hardware, site, personnel vulnerabilities are examples of vulnerabilities that are not software security bugs.

This reconnaissance tool finds the subdomains of a domain, after finding the subdomains by vulnerability scanning we find the ports which are vulnerable. Having a unsecured subdomain can lead to a serious risk to website, there were some security incidents where the hacker used subdomain tricks to hack into the website.

Network Vulnerability Scanning:

A vulnerability scan detects and classifies system weaknesses in computers, networks and communications equipment and predicts the effectiveness of countermeasures. A scan may be performed by an organization's IT department or a security service provide, possibly as a condition imposed by some authority. An Approved Scanning Vendor (ASV), for example, is a service provider that is certified and authorized by the Payment

Card Industry (PCI) to scan payment card networks. Vulnerability scans are also used by attackers looking for points of entry.

A vulnerability scanner runs from the end point of the person inspecting the attack surface in question. The software compares details about the target attack surface to a database of information about known security holes in services and ports, anomalies in packet construction, and potential paths to exploitable programs or scripts. The scanner software attempts to exploit each vulnerability that is discovered.

Running a vulnerability scan can pose its own risks as it is inherently intrusive on the target machine's running code. As a result, the scan can cause issues such as errors and reboots, reducing productivity.

There are two approaches to vulnerability scanning, authenticated and unauthenticated scans. In the unauthenticated method, the tester performs the scan as an intruder would, without trusted access to the network. Such a scan reveals vulnerabilities that can be accessed without logging into the network. In an authenticated scan, the tester logs in as a network user, revealing the vulnerabilities that are accessible to a trusted user, or an intruder that has gained access as a trusted user.



Figure3.1:NetworkVulnerability

4 RESEARCH OBJECTIVES

Network Domain:

A **network domain** is an administrative grouping of multiple private computer networks or hosts within the same infrastructure. Domains can be identified using a domain name; domains which need to be accessible from the public Internet can be assigned a globally unique name within the Domain Name System (DNS).

A domain controller is a server that automates the logins, user groups, and architecture of a domain, rather than manually coding this information on each host in the domain. It is common practice, but not required, to have the domain controller act as a DNS server. That is, it would assign names to hosts in the network based on their IP addresses.

In the Domain Name System (DNS) hierarchy, a subdomain is a domain that is a part of another (main) domain.

For example: if a domain offered an online store as part of their website `example.com`, you might use the subdomain `shop.example.com`

A Fully Qualified Domain Name consists of multiple parts. For example, the english wikipedia domain: `en.wikipedia.org`

The `en` is a subdomain. Although `wikipedia.org` is usually considered to be the domain name, `wikipedia` is actually a sub-domain of the `org` TLD (top level domain).

The Domain Name System (DNS) has a tree structure or hierarchy, which includes nodes on the tree being a domain name. A subdomain is a domain that is part of a larger domain. Each label may contain from 1 to 63 octets. The full domain name may not exceed a total length of 253 ASCII characters in its textual representation. Most domain registries only allocate a two-level domain name. Hosting services typically provide DNS Servers to resolve subdomains within that master domain.

Subdomains in this context are defined by editing the DNS zone file pertaining to the parent domain. However, there is an ongoing debate over the use of the term “subdomain” when referring to names which map to the Address record A (host) and various other types of zone records which may map to any public IP address destination and any type of server. Network Operations teams insist that it is inappropriate to use the term “subdomain” to refer to any

mapping other than that provided by zone NS (name server) records and any server-destination other than that.

SubDomain:

Subdomains are often used by internet service providers supplying web services. They allocate one (or more) subdomains to their clients who do not have their own domain name. This allows independent administration by the clients over their subdomain.

Subdomains are also used by organizations that wish to assign a unique name to a particular department, function, or service related to the organization. For example, a university might assign "cs" to the computer science department, such that a number of hosts could be used inside that subdomain, such as `www.cs.example.edu`.

There are some widely recognized subdomains including `www`, `ftp`. This allows for a structure where the domain contains administrative directories and files including the `ftp` directories and webpages. The `ftp` subdomain can contain logs and the web page directories. The `www` subdomain contains the directories for the webpages. Independent authentication for each domain provides access control over the various levels of the domain.

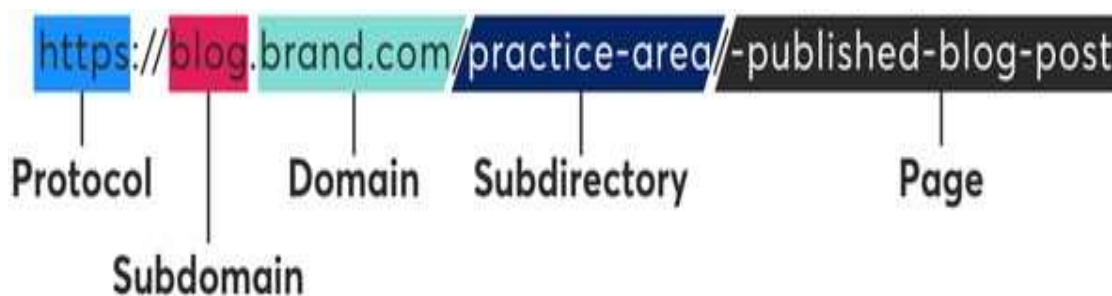


Figure 4.1: Domain and Subdomain

Port Filtering:

Port-filtering definitions. Filters. Allowing or blocking network packets into or out of a device or the network based on their application (port number)

Port filtering is when a router monitors the destination ports of the tcp/udp and/or other port based network protocol packets that pass through it with port filtering you can have the router block packets that are heading to a certain port or block some packets based on their content.

The practice of attempting to connect to a range of ports in sequence on a single computer is commonly as port scanning. This is usually associated either with malicious cracking attempts or

with network administrators looking for possible vulnerabilities to help prevent such attacks. Port connection attempts are frequently monitored and logged by computers. The technique of port knocking uses a series of port connections (knocks) from a client computer to enable a server connection.

Transmission Control Protocol:

The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating via an IP network. Major internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP, which is part of the Transport Layer of the TCP/IP suite. SSL/TLS often runs on top of TCP.

TCP is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. Three-way handshake (active open), retransmission, and error-detection adds to reliability but lengthens latency. Applications that do not require reliable data stream service may use the User Datagram Protocol (UDP), which provides a connectionless datagram service that prioritizes time over reliability. TCP employs network congestion avoidance. However, there are vulnerabilities to TCP including denial of service, connection hijacking, TCP veto, and reset attack. For network security, monitoring, and debugging, TCP traffic can be intercepted and logged with a packet sniffer.

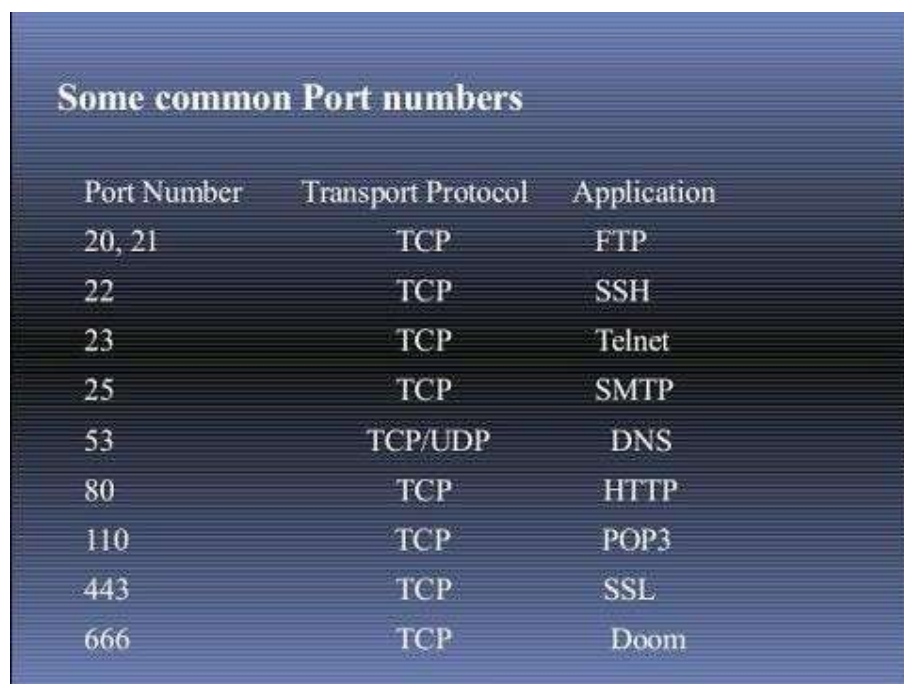
User Datagram Protocol:

In computer networking, the User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite. The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network. Prior communications are not required in order to set up communication channels or data paths.

UDP uses a simple connectionless communication model with a minimum of protocol mechanisms. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. It has

no handshaking dialogues, and thus exposes the user's program to any unreliability of the underlying network; there is no guarantee of delivery, ordering, or duplicate protection. If error-correction facilities are needed at the network interface level, an application may use Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP) which are designed for this purpose.

UDP is suitable for purposes where error checking and correction are either not necessary or are performed in the application; UDP avoids the overhead of such processing in the protocol stack. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for packets delayed due to retransmission, which may not be an option in a real-time system.



| Port Number | Transport Protocol | Application |
|-------------|--------------------|-------------|
| 20, 21 | TCP | FTP |
| 22 | TCP | SSH |
| 23 | TCP | Telnet |
| 25 | TCP | SMTP |
| 53 | TCP/UDP | DNS |
| 80 | TCP | HTTP |
| 110 | TCP | POP3 |
| 443 | TCP | SSL |
| 666 | TCP | Doom |

Figure 4.2: common port numbers

5 METHODOLOGY

In the Tool First we have to know how to script the tool with using language python.

First we have to check the hardware and software requirements which are required.

Then we have to check the ram requirements in the software and install windows 10 latest version in the laptop.

Then software should have virtual machine support where we will install virtualbox or vmware

The next step will be checking the memory requirements and installing kali linux in our virtual machine .

The Information Gathering Scripted Tool will be running by language python in kali linux terminal. First we have to gather the libraries when we are ready build the tool.

We have to figure out that features that we are adding to the tool should be included in the tool.

In the process of building team members who are good in networking and cyber security help to develop code and find the right use cases to be included in the script.

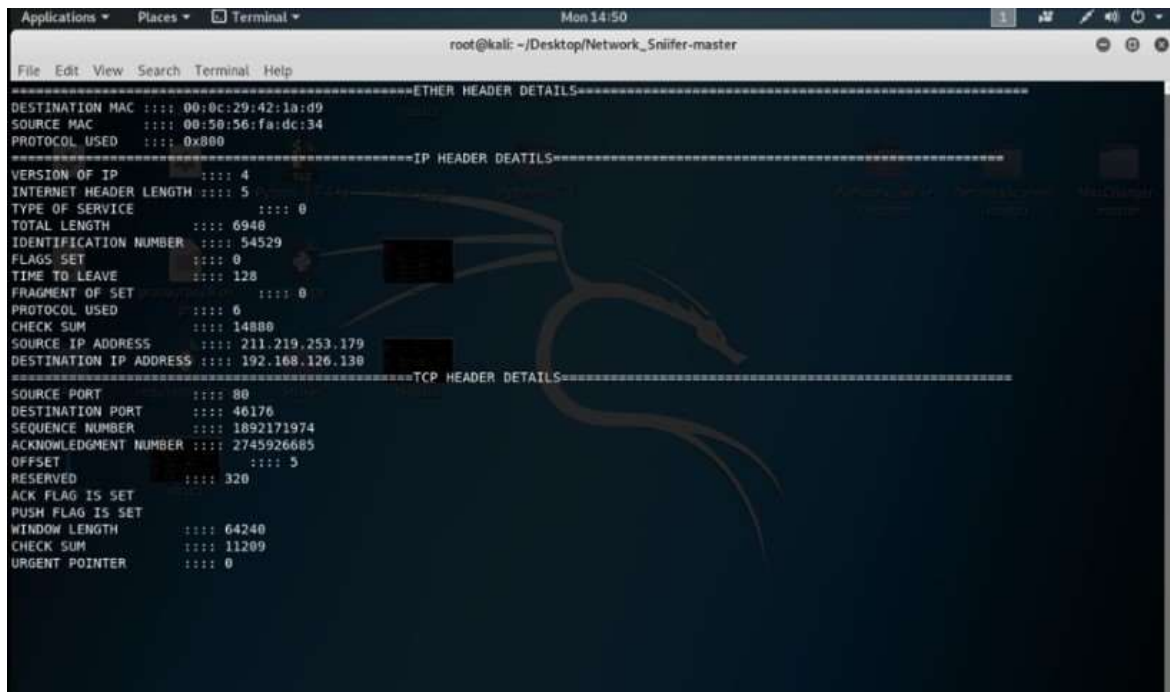
In this phase team members who are building the tool write code in python which will be tested in various stages and software development life cycle is followed here.

When the script is written perfectly it is checked for any bug fixes or errors and testing will be done .

After the tool is ready to launch in kali linux we will ensure that this python scripted which will give excellent results.

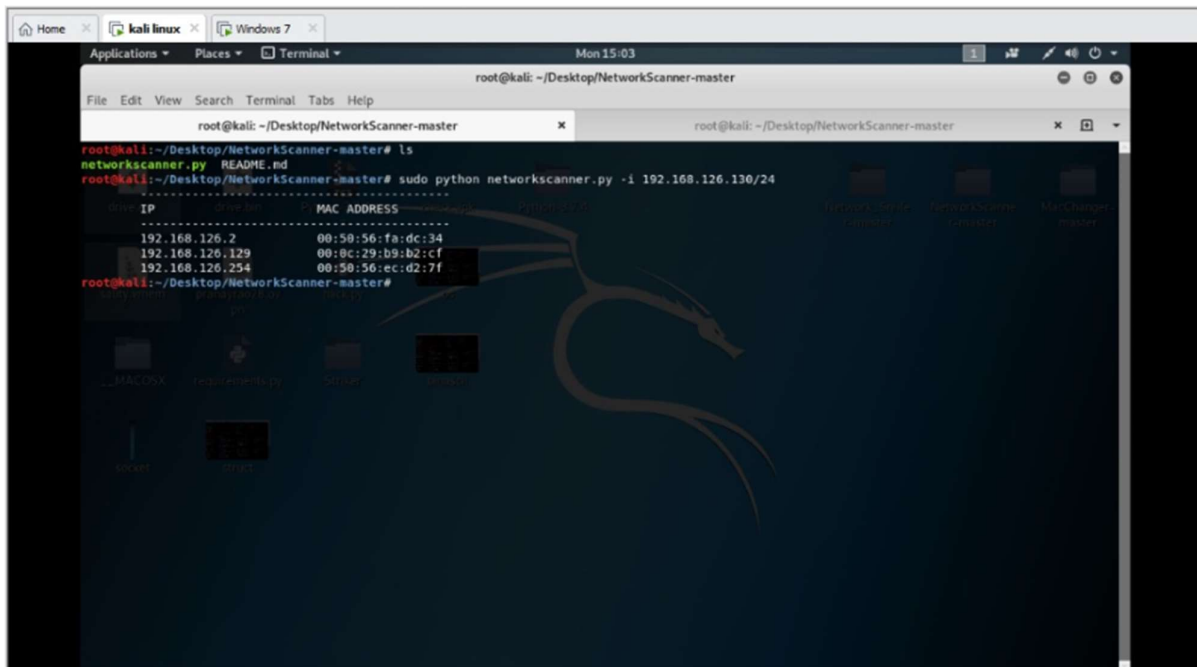
In this process of building this tool and after the completion this will provide the features that

6 RESULT AND DISCUSSION



The screenshot shows a Kali Linux terminal window titled "root@kali: ~/Desktop/Network_Sniffer-master". The terminal displays the output of a network sniffing tool, showing details for an Ethernet II, IP, and TCP header. The Ethernet II header shows a destination MAC of 00:0c:29:42:1a:d9 and a source MAC of 00:50:56:fa:dc:34. The IP header shows a source IP of 211.219.253.179 and a destination IP of 192.168.126.130. The TCP header shows a source port of 80 and a destination port of 46176. The terminal also shows a list of files and directories in the current directory, including "README.md", "networkscanner.py", "requirements.txt", "sniffer", and "sniffer.py".

```
root@kali: ~/Desktop/Network_Sniffer-master
File Edit View Search Terminal Help
=====ETHER HEADER DETAILS=====
DESTINATION MAC : 00:0c:29:42:1a:d9
SOURCE MAC      : 00:50:56:fa:dc:34
PROTOCOL USED   : 0x800
=====IP HEADER DEATILS=====
VERSION OF IP   : 4
INTERNET HEADER LENGTH : 5
TYPE OF SERVICE : 0
TOTAL LENGTH    : 6940
IDENTIFICATION NUMBER : 54529
FLAGS SET       : 0
TIME TO LEAVE   : 128
FRAGMENT OF SET : 0
PROTOCOL USED   : 6
CHECK SUM       : 14880
SOURCE IP ADDRESS : 211.219.253.179
DESTINATION IP ADDRESS : 192.168.126.130
=====TCP HEADER DETAILS=====
SOURCE PORT      : 80
DESTINATION PORT : 46176
SEQUENCE NUMBER  : 1892171974
ACKNOWLEDGMENT NUMBER : 2745926685
OFFSET           : 5
RESERVED         : 320
ACK FLAG IS SET  : 
PUSH FLAG IS SET : 
WINDOW LENGTH    : 64240
CHECK SUM        : 11209
URGENT POINTER   : 0
```



The screenshot shows a Kali Linux terminal window titled "root@kali: ~/Desktop/NetworkScanner-master". The terminal displays the output of a network scanner, showing a list of IP addresses and their corresponding MAC addresses. The scanner has scanned the network 192.168.126.130/24 and found three active hosts. The terminal also shows a list of files and directories in the current directory, including "README.md", "networkscanner.py", "requirements.txt", "sniffer", and "sniffer.py".

```
root@kali: ~/Desktop/NetworkScanner-master
File Edit View Search Terminal Tabs Help
root@kali:~/Desktop/NetworkScanner-master# ls
networkscanner.py  README.md
root@kali:~/Desktop/NetworkScanner-master# sudo python networkscanner.py -i 192.168.126.130/24
=====
IP           MAC ADDRESS
-----
192.168.126.2    00:50:56:fa:dc:34
192.168.126.129  00:0c:29:b9:b2:cf
192.168.126.254  00:50:56:ec:d2:7f
root@kali:~/Desktop/NetworkScanner-master#
```

```
Home kali linux Windows 7
Applications Places Terminal
Mon 15:05
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.126.130 netmask 255.255.255.0 broadcast 192.168.126.255
    inet6 fe80::20c:29ff:fe42:1ad9 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:42:1a:d9 txqueuelen 1000 (Ethernet)
    RX packets 11505 bytes 15396153 (14.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5843 bytes 389490 (380.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2044 bytes 88388 (86.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2044 bytes 88388 (86.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~#
```

```
Home kali linux Windows 7
Applications Places Terminal
Mon 15:07
root@kali: ~/Desktop/MacChanger-master
File Edit View Search Terminal Help
root@kali:~/Desktop/MacChanger-master# ls
mac.py README.md
root@kali:~/Desktop/MacChanger-master# sudo python mac.py -i eth0 -n 9c:56:12:85:6d:5f
-----
| Welcome To Mac Changer |
-----
[*] Current Mac:: 00:0c:29:42:1a:d9
Trying to change you mac address
[*] Mac Address has been changed succesfully from 00:0c:29:42:1a:d9 to 9c:56:12:85:6d:5f
root@kali:~/Desktop/MacChanger-master# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.126.131 netmask 255.255.255.0 broadcast 192.168.126.255
    ether 9c:56:12:85:6d:5f txqueuelen 1000 (Ethernet)
    RX packets 14292 bytes 19359941 (18.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6647 bytes 439242 (428.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2044 bytes 88388 (86.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2044 bytes 88388 (86.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/Desktop/MacChanger-master#
```

8. REFERENCES

1. Kevin J. Connolly (2003). Law of Internet Security and Privacy. Aspen Publishers. p. 131. ISBN 978-0-7355-4273-0.
2. ^ "Network Segment Definition". www.linfo.org. Retrieved January 14, 2016