

Ethical Hacking Using Python

The background of the slide is a vibrant blue with a large, white, curved shape on the left side. Overlaid on the blue background are several digital and technological icons: a black microchip at the top left, a black globe at the top right, a series of black gears on the right side, a black brain icon inside a circle with circuit lines at the bottom right, and a black Android robot at the bottom center. A spiral of white binary code (0s and 1s) winds through the center of the slide, creating a sense of depth and movement.

CBER 721: Ethical Hacking and Defense
Final Report

Jose Lira
Student ID: 301162762

Agenda

1. Introduction
2. Protocols and Libraries
3. Ethical Hacking Tools (Layer 2)
 1. MAC Address Changer:
 2. ARP Spoofing Tool (Attack)
 3. ARP Spoofing Detector (Defense)
 4. LAN Network Scanner IP/MAC
 5. LAN Packet Sniffer (MITM: Credential Harvester)
4. Ethical Hacking Tools (Layer 3)
 1. IP Spoof
 2. SYN flooding (Denial-of-Service)
 3. Port Scanner/Banner Grabbing
5. Conclusions

Introduction

- The US Bureau of Labor Statistics projects 33% job growth for information security specialists, including pen testers, between 2020 and 2030 (Coursera).
- Penetration testers in the US make an average salary of \$102,405 (Glassdoor).
- I refuse to use Kali Linux as a magic black box.
- I want to understand how hacking applications work and how to create my own tools.
- Python language allows us to quickly program many of the same commercial hacking tools available in Kali Linux



Protocols and Libraries

- **ARP Protocol:** The Address Resolution Protocol (ARP) is a straightforward protocol that allows us to link (or translate) IP addresses to MAC addresses.
- **Socket Library:** The Socket Library in Python is the low-level networking interface that gives access to the BSD (Berkley Software Distribution) socket interface. It is included in all major operating systems: Windows, Linux distributions, and Apple macOS. In some cases, specific behaviour may depend on the platform as the libraries call directly to the OS socket APIs.
- **Scapy library:** Scapy is a practical Python module that allows the manipulation of network packets. Scapy can interpret and craft packets for many of the most utilized protocols. It can be utilized for performing different networking tasks like testing, sniffing, and scanning in a Python application.



The Tools

- **Ethical Hacking Tools (Layer 2)**

1. MAC Address Changer
2. ARP Spoofing Tool (Attack)
3. ARP Spoofing Detector (Defense)
4. LAN Network Scanner IP/MAC
5. LAN Packet Sniffer (MITM: Credential Harvester)

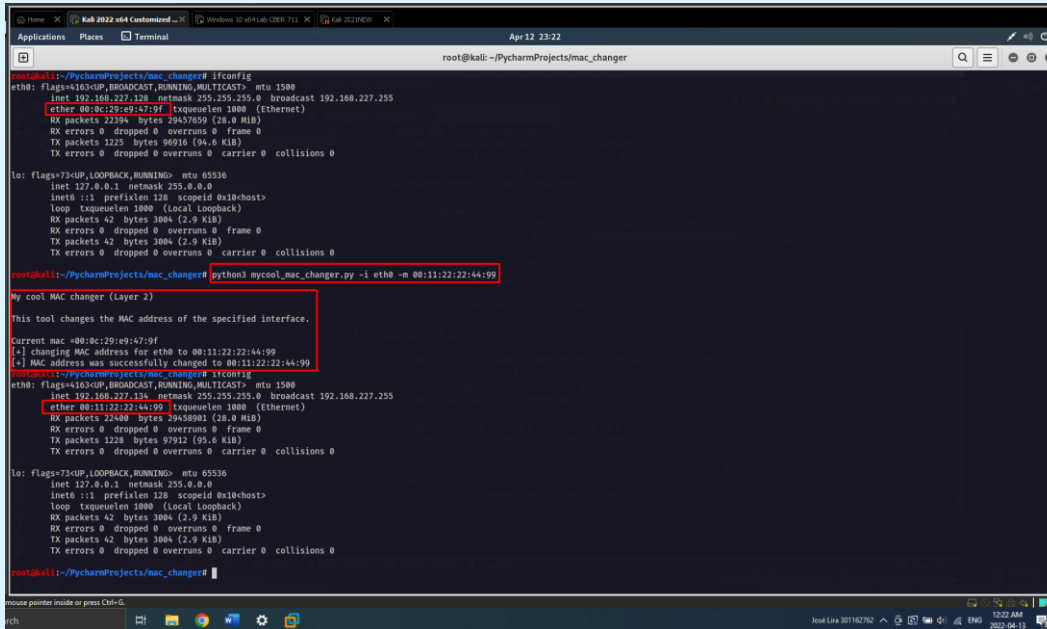
- **Ethical Hacking Tools (Layer 3)**

1. IP Spoof
2. SYN flooding (Denial-of-Service)
3. Port Scanner/Banner Grabbing



Ethical Hacking Tools (Layer 2)

1. MAC Address Changer:



```
root@kali:~/PycharmProjects/mac_changer# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.227.128 netmask 255.255.0 broadcast 192.168.227.255
    ether 00:0c:29:e9:47:9f txqueuelen 1000 (Ethernet)
    RX packets 22394 bytes 29457639 (28.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1223 bytes 96916 (94.0 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 42 bytes 3084 (2.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 42 bytes 3084 (2.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/PycharmProjects/mac_changer# python3 mycool_mac_changer.py -i eth0 -m 00:11:22:22:44:99

My cool MAC changer (Layer 2)
This tool changes the MAC address of the specified interface.
Current mac =00:0c:29:e9:47:9f
[*] changing MAC address for eth0 to 00:11:22:22:44:99
[*] MAC address was successfully changed to 00:11:22:22:44:99
root@kali:~/PycharmProjects/mac_changer# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.227.128 netmask 255.255.0 broadcast 192.168.227.255
    ether 00:11:22:22:44:99 txqueuelen 1000 (Ethernet)
    RX packets 22400 bytes 29458001 (28.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1228 bytes 97912 (95.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

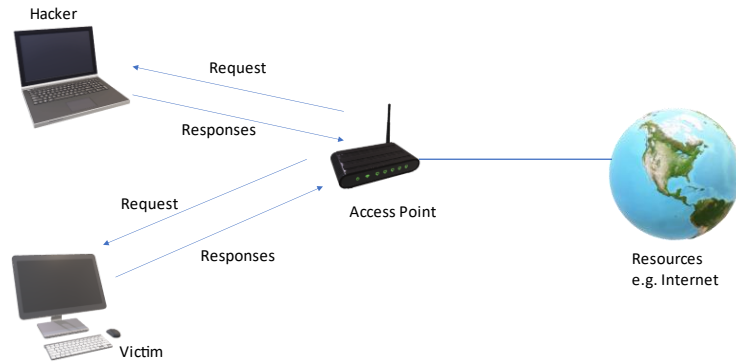
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 42 bytes 3084 (2.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 42 bytes 3084 (2.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/PycharmProjects/mac_changer#
```

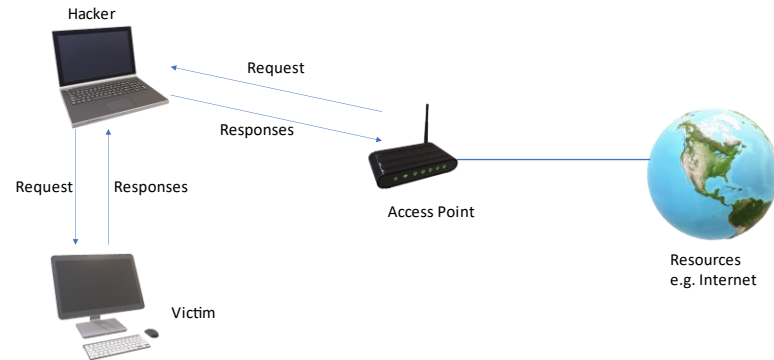
Ethical Hacking Tools (Layer 2)

2. ARP Spoofing Tool (Attack)

Normal Network



ARP Spoofing



Ethical Hacking Tools (Layer 2)

2. ARP Spoofing Tool (Attack)

The screenshot displays a Kali Linux virtual machine environment. The left terminal window shows the execution of the `arp-spoof` tool, which is configured to spoof the IP address 192.168.227.255. The tool's output shows it is running on interface `eth0` and has successfully spoofed the target IP. The right terminal window shows the output of the `ipconfig` command, displaying the network configuration for the `eth0` interface, including the IP address, subnet mask, and default gateway.

```
root@kali:~/PycharmProjects/arp_spoof# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.227.134 netmask 255.255.0.0 broadcast 192.168.227.255
    inet6 fe80::a11:22ff:fe02:4499 prefixlen 64 scopeid 0x20<link>
    ether 00:11:22:22:44:99 txqueuelen 1000 (Ethernet)
    RX packets 23795 bytes 29556415 (28.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1681 bytes 130428 (127.3 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 46 bytes 3244 (3.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 46 bytes 3244 (3.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/PycharmProjects/arp_spoof# python3 cool_arp_spoof.py -t 192.168.227.129 -g 192.168.227.2
My cool ARP Spoofing program (Layer 2)
This tool spoof the ARP protocol tricking the victim and the router.
[*] Packets sent 78°C
[-] <CTRL> + <C> detected... Resetting ARP tables... Please wait...
root@kali:~/PycharmProjects/arp_spoof#
```

```
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . . : fe80::3d81:ef68:709:c8ac%5
IPv4 Address. . . . . : 192.168.227.129
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.227.2

Ethernet adapter Bluetooth Network Connection:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

C:\Users\jllina>arp -a

Interface: 192.168.227.129 --- 0x5
Internet Address Physical Address Type
192.168.227.2 00-50-56-e5-eb-16 dynamic
192.168.227.134 00-11-22-22-44-99 dynamic
192.168.227.255 ff-ff-ff-ff-ff-ff static
224.0.0.22 01-00-5e-00-00-16 static
224.0.0.251 01-00-5e-00-00-fb static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static
255.255.255.255 ff-ff-ff-ff-ff-ff static

C:\Users\jllina>arp -a

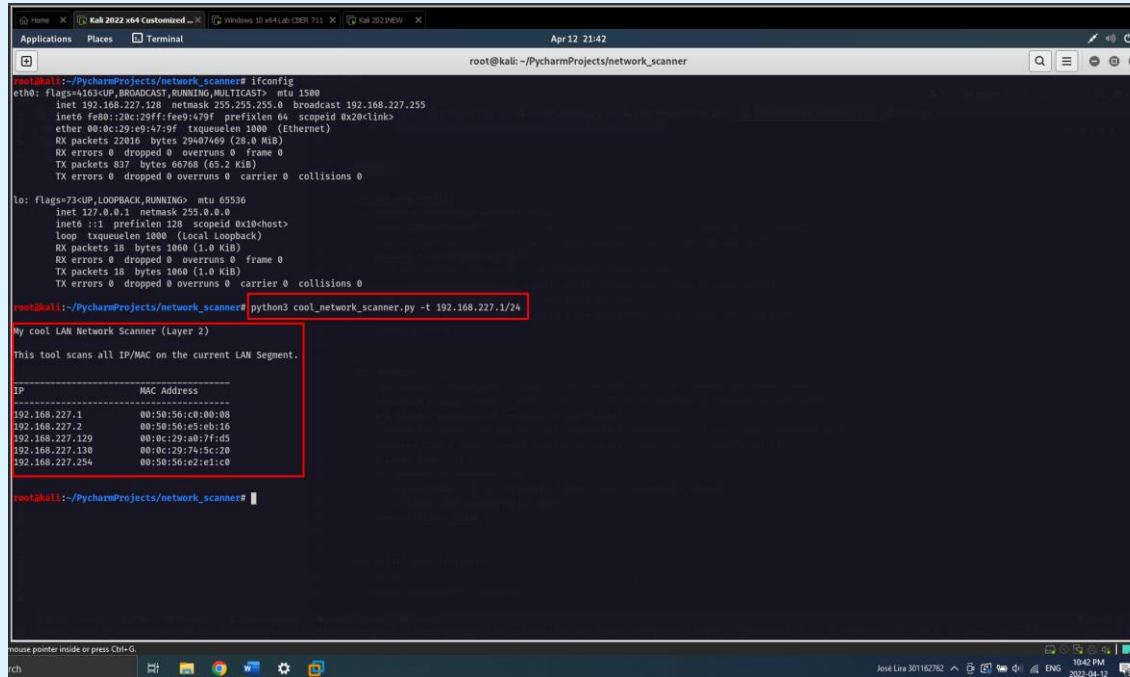
Interface: 192.168.227.129 --- 0x5
Internet Address Physical Address Type
192.168.227.2 00-50-56-e5-eb-16 dynamic
192.168.227.134 00-11-22-22-44-99 dynamic
192.168.227.255 ff-ff-ff-ff-ff-ff static
224.0.0.22 01-00-5e-00-00-16 static
224.0.0.251 01-00-5e-00-00-fb static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static
255.255.255.255 ff-ff-ff-ff-ff-ff static

C:\Users\jllina>arp -a

Interface: 192.168.227.129 --- 0x5
Internet Address Physical Address Type
192.168.227.2 00-50-56-e5-eb-16 dynamic
192.168.227.134 00-11-22-22-44-99 dynamic
192.168.227.255 ff-ff-ff-ff-ff-ff static
224.0.0.22 01-00-5e-00-00-16 static
224.0.0.251 01-00-5e-00-00-fb static
224.0.0.252 01-00-5e-00-00-fc static
239.255.255.250 01-00-5e-7f-ff-fa static
255.255.255.255 ff-ff-ff-ff-ff-ff static
```


Ethical Hacking Tools (Layer 2)

4. LAN Network Scanner IP/MAC addresses



The screenshot shows a Kali Linux terminal window with the following content:

```
root@kali: ~/PycharmProjects/network_scanner
ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.227.128 netmask 255.255.255.0 broadcast 192.168.227.255
    inet6 fe80::20c:29ff:fe09:479f prefixlen 64 scopeid 0x20<link>
    ether 08:0c:29:e9:47:9f txqueuelen 1000 (Ethernet)
    RX packets 22816 bytes 29407460 (28.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 837 bytes 66768 (65.2 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 10 bytes 1060 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 1060 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/PycharmProjects/network_scanner# python3 cool_network_scanner.py -t 192.168.227.1/24

My cool LAN Network Scanner (Layer 2)
This tool scans all IP/MAC on the current LAN Segment.

IP             MAC Address
-----
192.168.227.1   08:50:56:c8:00:08
192.168.227.2   08:50:56:e5:eb:16
192.168.227.129 08:0c:29:a0:7f:d5
192.168.227.130 08:0c:29:74:5c:20
192.168.227.254 08:50:56:e2:e1:c0

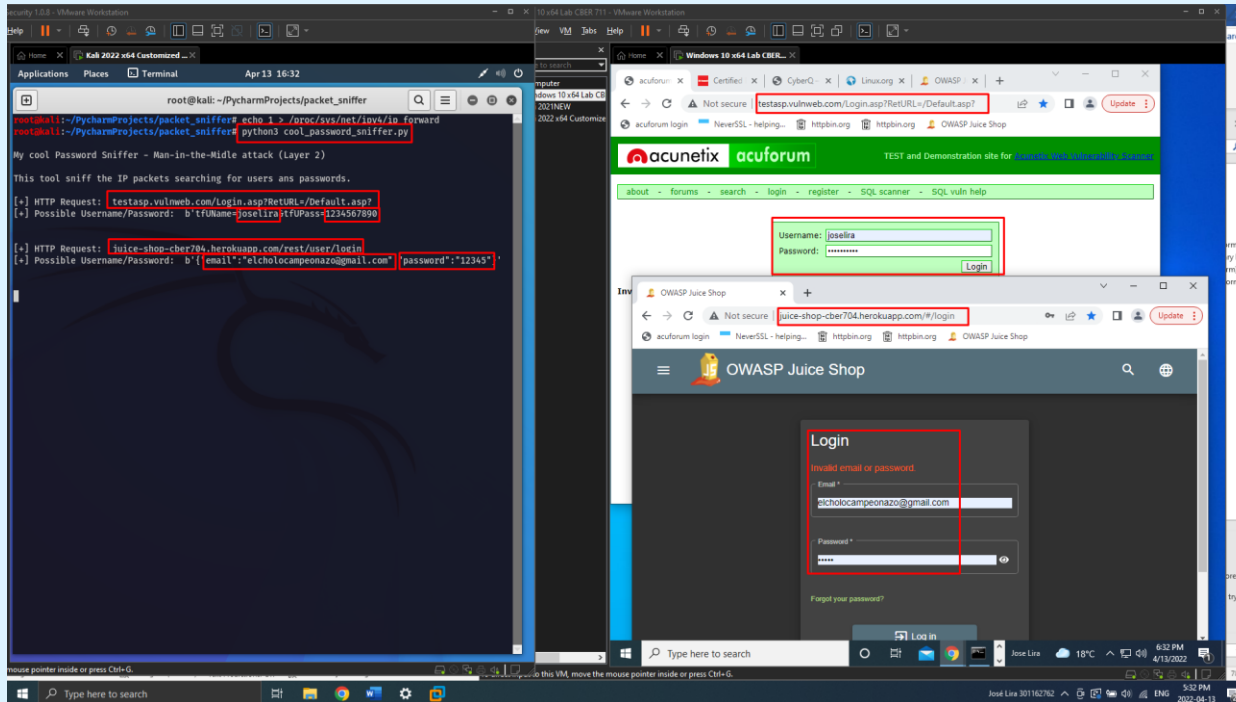
root@kali:~/PycharmProjects/network_scanner#
```

The terminal output shows the configuration of the `eth0` interface and the loopback interface `lo`. It then shows the execution of the `python3 cool_network_scanner.py -t 192.168.227.1/24` command, which displays a table of IP and MAC addresses for the scanned LAN segment.

IP	MAC Address
192.168.227.1	08:50:56:c8:00:08
192.168.227.2	08:50:56:e5:eb:16
192.168.227.129	08:0c:29:a0:7f:d5
192.168.227.130	08:0c:29:74:5c:20
192.168.227.254	08:50:56:e2:e1:c0

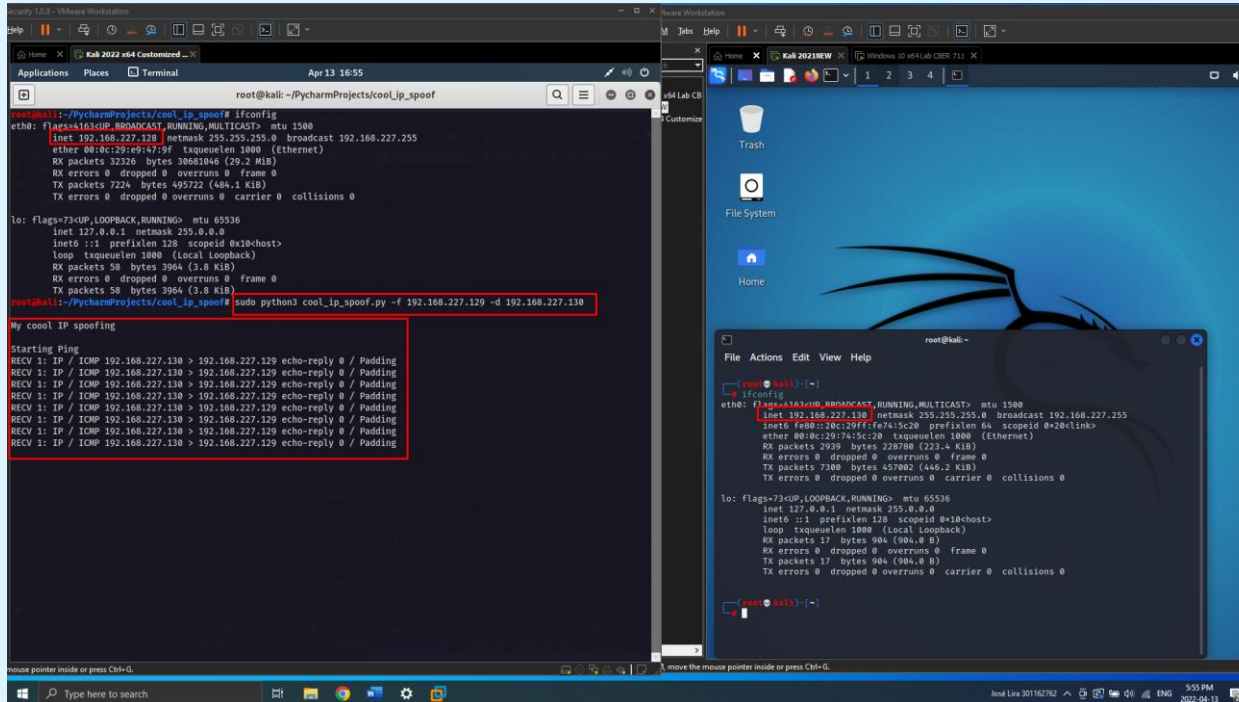
Ethical Hacking Tools (Layer 2)

5. LAN Packet Sniffer (MITM: Credential Harvester)



Ethical Hacking Tools (Layer 3)

1. IP Spoof



The screenshot displays a Kali Linux virtual machine environment. The main terminal window shows the configuration of a network interface for IP spoofing. The configuration includes setting the interface to 'eth0', enabling 'LOOPBACK', 'RUNNING', and 'MULTICAST' flags, and setting the MTU to 1500. The interface is configured with IP address 192.168.227.128, netmask 255.255.255.0, and broadcast address 192.168.227.255. The interface is also configured with a loopback queue length of 1000 and a local loopback flag. The interface is then brought up with the 'ifconfig' command.

```
root@kali:~/PycharmProjects/cool_ip_spoof# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.227.128 netmask 255.255.255.0 broadcast 192.168.227.255
    ether 08:00:27:09:47:5f txqueuelen 1000 (Ethernet)
    RX packets 32320 bytes 30681046 (29.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7224 bytes 495722 (484.1 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 0<localhost>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 58 bytes 3964 (3.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 58 bytes 3964 (3.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~/PycharmProjects/cool_ip_spoof# sudo python3 cool_ip_spoof.py -f 192.168.227.129 -d 192.168.227.130
```

The output of the script shows the results of the IP spoofing attack. It displays the starting ping and the results of the spoofing process. The spoofing process is successful, as indicated by the 'TX packets 7880' and 'TX errors 0'.

```
My cool IP spoofing

Starting Ping
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
RECV 1: IP / ICMP 192.168.227.130 > 192.168.227.129 echo-reply 0 / Padding
```

The background of the slide features a blue and white graphic with binary code (0s and 1s) and a globe, symbolizing network security and ethical hacking.

Ethical Hacking Tools (Layer 3)

2. SYN flooding (Denial-of-Service)

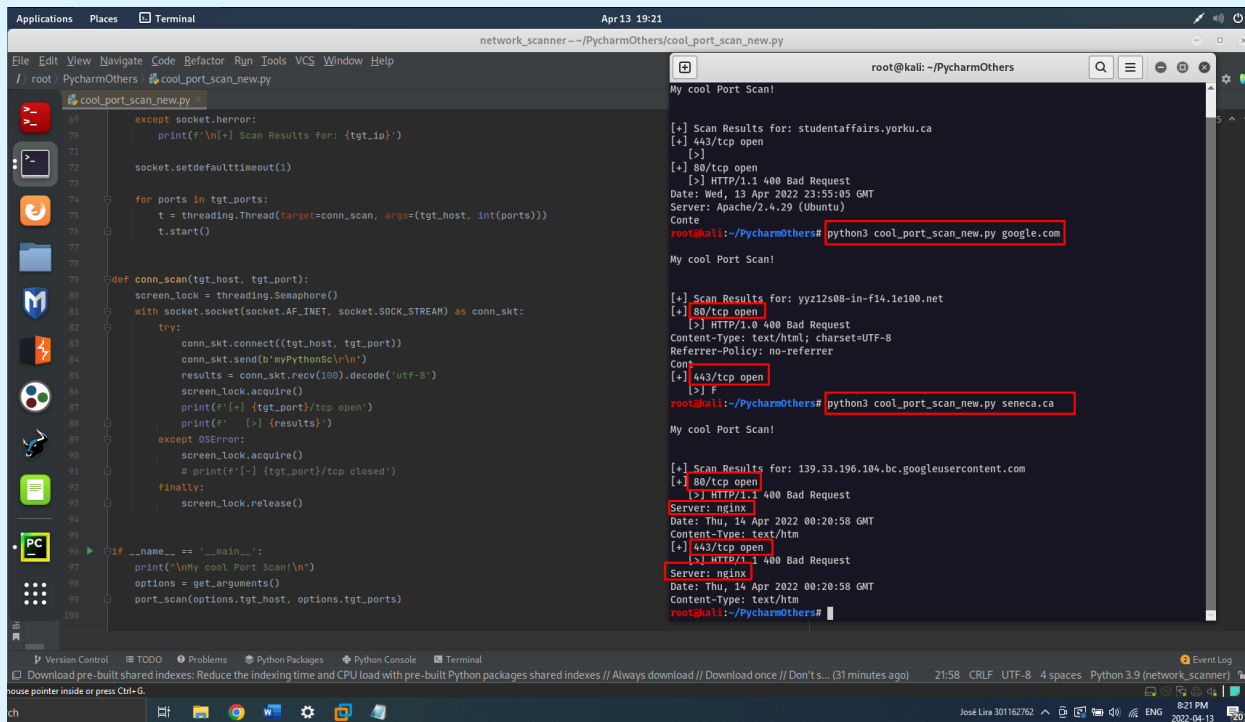
The screenshot displays a Kali Linux terminal environment during a SYN flood attack. The terminal is split into three main sections:

- Left Pane (Packet Capture):** Shows a live capture from the `eth0` interface. The filter is `display filter: <<Ctrl>`. The table lists captured packets with columns for Time, Source, Destination, Protocol, Length, and Info. Red boxes highlight several TCP SYN packets and their retransmissions.
- Middle Pane (Tool Configuration):** Shows the configuration for the `cool_syn_flood` tool. The configuration includes:
 - Interface: `eth0`
 - Flags: `flags=K3CUP,BROADCAST,RUNNING,MULTICAST`
 - Netmask: `255.255.255.0`
 - Broadcast: `192.168.227.255`
 - Prefixlen: `64`
 - Scopeid: `0x20<link>`
 - TX packets: `7504` bytes `515280` (503.2 Kib)
 - TX errors: `0` dropped `0` overruns `0` carrier `0` collisions `0`
- Right Pane (Tool Execution):** Shows the execution of the tool with the command `sudo python3 cool_syn_flood.py -d 192.168.227.129`. The output displays a list of failed connections to various ports on the target IP `192.168.227.129`.

The bottom status bar indicates that the capture is in progress, showing `Packets: 284 - Displayed: 284 (100.0%)` and `Profile: Default`.

Ethical Hacking Tools (Layer 3)

3. Port Scanner/Banner Grabbing



The screenshot displays a Kali Linux desktop environment. On the left, a code editor shows the source code for a Python port scanner named `cool_port_scan_new.py`. The code includes a `conn_scan` function that uses `socket` and `threading` to scan multiple ports simultaneously. On the right, a terminal window shows the execution of the scanner against three targets: `studentaffairs.yorku.ca`, `yyz12s08-in-f14.1e100.net`, and `139.33.196.104.bc.googleusercontent.com`. The terminal output shows the scan results, including open ports (443/tcp, 80/tcp) and HTTP status codes (400 Bad Request). The scanner also displays the server banner for the third target, which is `nginx`.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
/ root PycharmOthers cool_port_scan_new.py

# cool_port_scan_new.py
69 except socket.error:
70     print(f'\n[-] Scan Results for: {tgt_ip}')
71
72 socket.setdefaulttimeout(1)
73
74 for ports in tgt_ports:
75     t = threading.Thread(target=conn_scan, args=(tgt_host, int(ports)))
76     t.start()
77
78
79 def conn_scan(tgt_host, tgt_port):
80     screen_lock = threading.Semaphore(1)
81     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as conn_skt:
82         try:
83             conn_skt.connect((tgt_host, tgt_port))
84             conn_skt.send(b'myPythonScn\n')
85             results = conn_skt.recv(100).decode('utf-8')
86             screen_lock.acquire()
87             print(f'[-] {tgt_port}/tcp open')
88             print(f'[-] {results}')
89         except OSError:
90             screen_lock.acquire()
91             # print(f'[-] {tgt_port}/tcp closed')
92         finally:
93             screen_lock.release()
94
95
96 if __name__ == '__main__':
97     print("\nMy cool Port Scan!\n")
98     options = get_arguments()
99     port_scan(options.tgt_host, options.tgt_ports)
100
101
102 My cool Port Scan!
103
104 [+] Scan Results for: studentaffairs.yorku.ca
105 [+] 443/tcp open
106 [-]
107 [+] 80/tcp open
108 [-] HTTP/1.1 400 Bad Request
109 Date: Wed, 13 Apr 2022 23:55:05 GMT
110 Server: Apache/2.4.29 (Ubuntu)
111 Conte
112 root@kali:~/PycharmOthers# python3 cool_port_scan_new.py google.com
113
114 My cool Port Scan!
115
116 [+] Scan Results for: yyz12s08-in-f14.1e100.net
117 [+] 80/tcp open
118 [-] HTTP/1.0 400 Bad Request
119 Content-Type: text/html; charset=UTF-8
120 Referrer-Policy: no-referrer
121 Cont
122 [+] 443/tcp open
123 [-]
124 root@kali:~/PycharmOthers# python3 cool_port_scan_new.py seneca.ca
125
126 My cool Port Scan!
127
128 [+] Scan Results for: 139.33.196.104.bc.googleusercontent.com
129 [+] 80/tcp open
130 [-] HTTP/1.1 400 Bad Request
131 Server: nginx
132 Date: Thu, 14 Apr 2022 00:20:58 GMT
133 Content-Type: text/html
134 [+] 443/tcp open
135 [-] HTTP/1.1 400 Bad Request
136 Server: nginx
137 Date: Thu, 14 Apr 2022 00:20:58 GMT
138 Content-type: text/html
139 root@kali:~/PycharmOthers#
```

Conclusions

- Most of the attacks done using the professional tools in Kali Linux can be programmed using Python scapy and sockets libraries. Knowing how to implement our tools is crucial to understanding how the hacking tools work.
- There might be Penetration Testing situations in which has gained access to a server we do not have permissions or access to uses any conventional Kali Linux applications; however, most Linux distributions come with a python interpreter by default in which we can quickly run python scripts to continue the attack. Additionally, the situation in which a particular option is not available in a conventional tool will not be an issue if we know how to program the function needed in python quickly.
- The ARP spoof technique is a powerful means to perform a Man-in-the-Middle attack, so it is essential to know how to use the tool to attack as to apply the mitigation technique (ARP spoof detector).
- We only show a few Layer 2 and Layer 3 attacks in this implementation. Nevertheless, the ability to develop new attacks, and add more options to the hacking programs is an exciting field for any cybersecurity professional.





Thanks!

