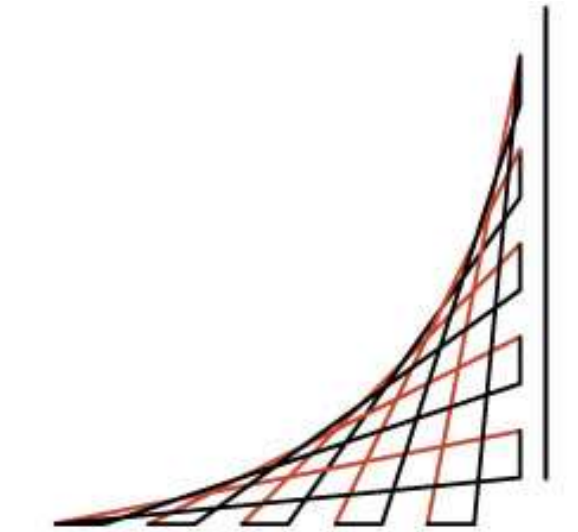


ESCUELA COLOMBIANA DE INGENIERÍA
JULIO GARAVITO

VIGILADA MINEDUCACIÓN

INGENIERIA DE SISTEMAS – SPTI 2021-1

LABORATORIO 6:

Laboratory Weapons to Exploit Network Layer

GERSON DAVID QUINTERO RODRIGUEZ

Autor:

Torres Segura Duck James Alexander

Jimmy Armando Chirivi Nivi

1. SECTION ONE

(CAPTURE NETWORK TRAFFIC WHIT SCAPY):

SNIFFING:

Sniffing is a technique that involves capturing all the information that circulates through a network. This information is stored and interpreted to discover sensitive data such as passwords, bank information, etc. This attack is one of the main ones that are made when you try to steal information.

SCAPY:

Scapy is a powerful library written and supported in Python which allows the creation, manipulation and injection of packages from a network environment. It allows to perform different tasks such as Fingerprinting, Discovering, Enumeration, among others.

1.1. EXPLAIN THE FOLLOWING TASKS SCAPY PERFORMS:

FINGERPRINTING:

Fingerprinting maps an arbitrarily large data set or piece of data to a much shorter string of bits, your fingerprint, which uniquely identifies the original data by uniquely identifying people for practical purposes, and SCAPY analyzes these data sets.

DISCOVERING:

SCAPY discovers hosts on the network during communication by using port scans.

ENUMERATION:

SCAPY discovers network information by discovering and enumerating devices or computers on a network.

VERIFY YOURSELF AS A USER:

1. Open terminal.
2. Cd /etc.
3. Nano hostname.
4. Enter your last name.
5. Save and exit.
6. Reboot the machine.

Ilustración 1: VERIFY YOURSELF AS A USER

CAPTURE NETWORK TRAFFIC WHIT SCAPY:

1. Open terminal.
2. Execute the command “scapy”.

Ilustración 2: scapy

3. Execute the command “sniff(count=20, prn=lambda x: x.summary())”.

```
>>> sniff (count=20, prn=lambda x: x.summary())
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / ARP who has 10.0.2.2 says 10.0.2.15
Ether / ARP is at 52:54:00:12:35:02 says 10.0.2.2 / Padding
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'Torres.'"
Ether / IP / UDP / DNS Qry "b'firefox.settings.services.mozilla.com.'"
Ether / IP / UDP / DNS Qry "b'firefox.settings.services.mozilla.com.'"
Ether / IP / UDP / DNS Ans "13.227.29.61"
Ether / IP / UDP / DNS Ans
Ether / IP / TCP 10.0.2.15:53602 > 13.227.29.61:https S
Ether / IP / TCP 13.227.29.61:https > 10.0.2.15:53602 SA / Padding
Ether / IP / TCP 10.0.2.15:53602 > 13.227.29.61:https A
Ether / IP / TCP 10.0.2.15:53602 > 13.227.29.61:https PA / Raw
Ether / IP / TCP 13.227.29.61:https > 10.0.2.15:53602 A / Padding
Ether / IP / TCP 13.227.29.61:https > 10.0.2.15:53602 PA / Raw
<Sniffed: TCP:6 UDP:12 ICMP:0 Other:2>
```

Ilustración 3: command

4. Explain the parameters of the command used:

count = number of packets to capture. 0 means infinity.

prn = function to apply to each packet. If something is returned, it is displayed.
prn = lambda x: x.summary().

5. Explain that important packets were captured:

The sniff listens to all the packets on the network some of these packets contain their protocols, dns, ips, input and output ports.

6. Execute the following commands:
a=sniff(count=300).
a.plot(lambda x:len(x)).
Explain the information given by the graphic.

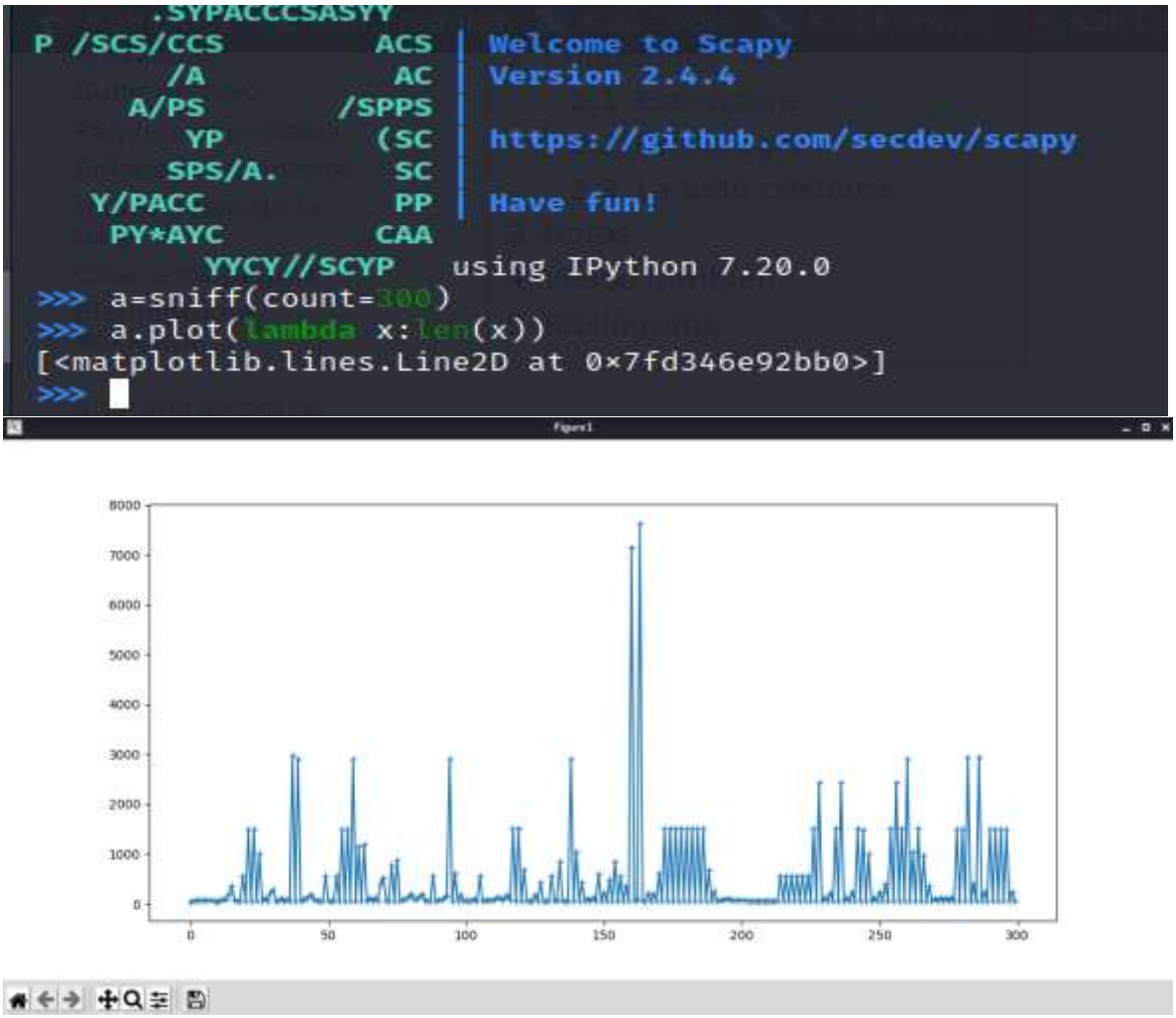


Ilustración 4: Explain the information given by the graphic

This function generates a graph with the data obtained in the sniff, where x is the amount of data and and its length.

2.2. (CAPTURE NETWORK TRAFFIC WHIT WIRESHARK):

WIRESHARK:

Multiplatform tool with graphical interface for network analysis. This allows you to see, even at a low and detailed level, everything that is happening on the network. It is open source and multiplatform. It is often used as a better option when auditing networks usually Ethernet networks and is compatible with some others.

CAPTURE NETWORK TRAFFIC WHIT WIRESHARK:

1. Start the other machine.
2. Open terminal and start the services apache and mysql with the following commands:
   a. Service apache2 start.
   b. Service mysql start.
3. Execute the command “Wireshark” in the first machine.

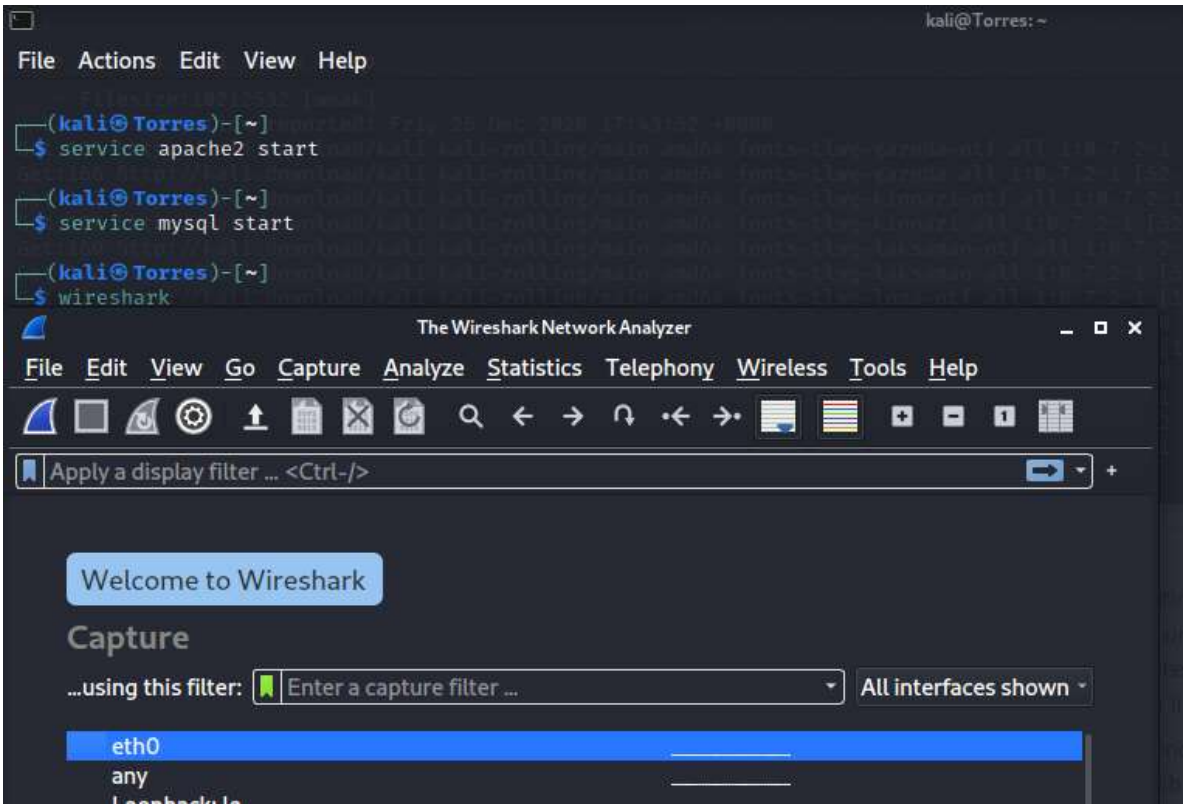


Ilustración 5: capture wireshark

4. Start traffic capture.
5. Open Firefox browser.
6. Enter to (second machine IP) / DVWA

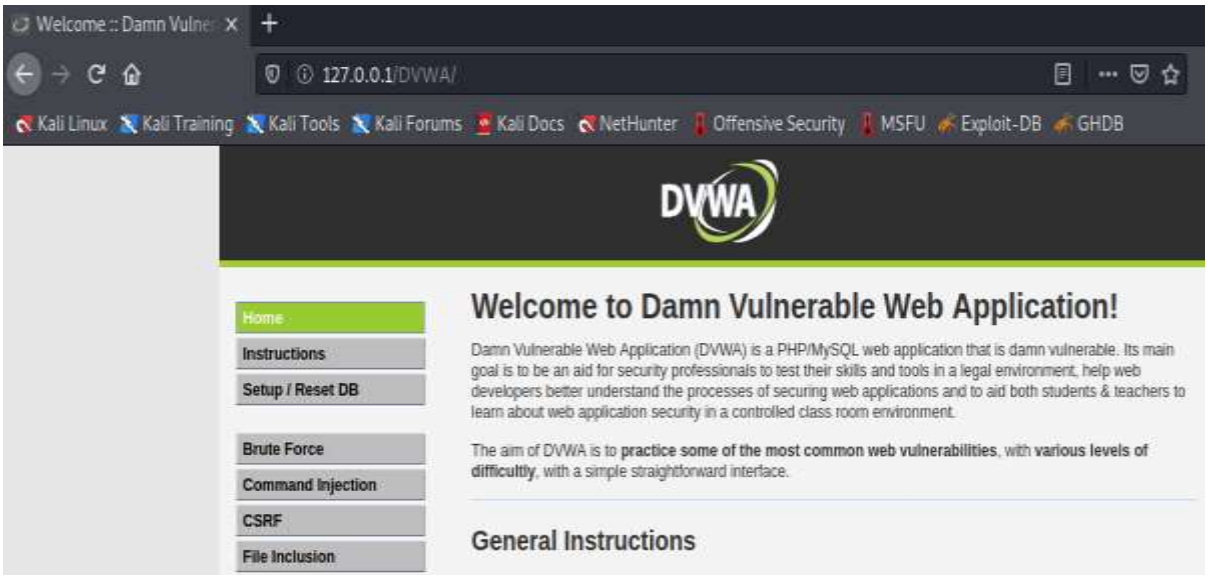


Ilustración 6: DWA

7. Enter to DVWA.
8. Enter whit User= “1337” and password=” charley”.

Ilustración 7: enter whith user

9. Stop traffic capture.
10. Filter capture by http.
11. Discover the vulnerability and explain

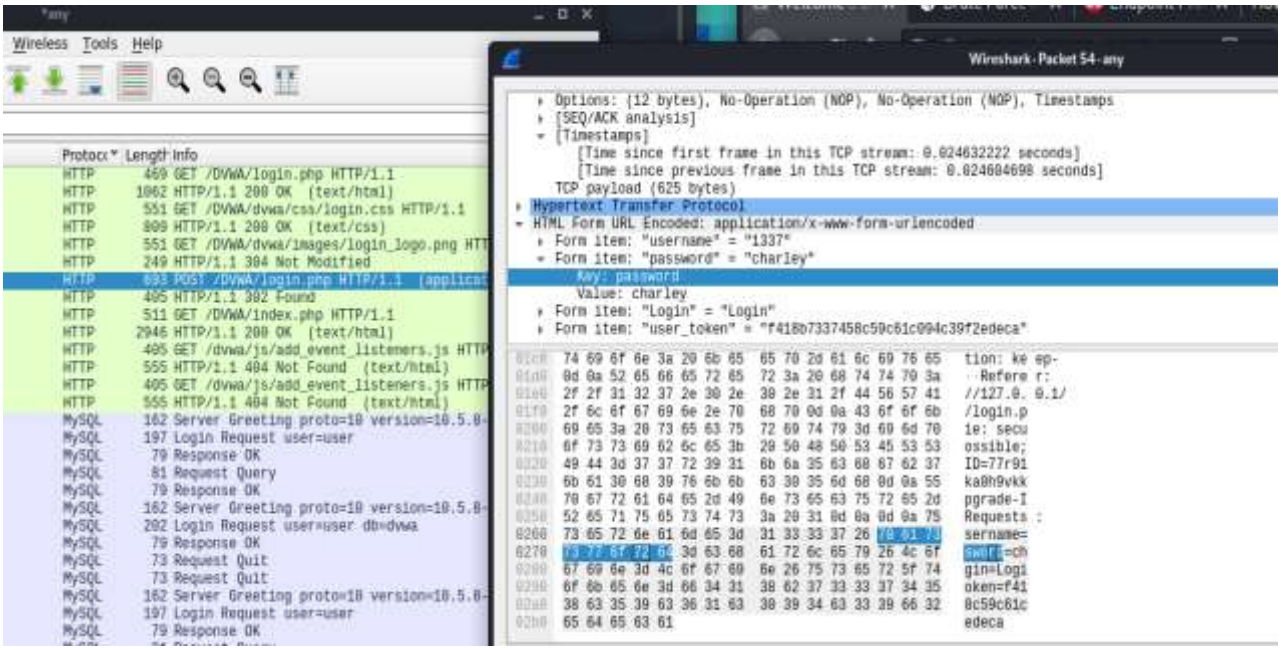
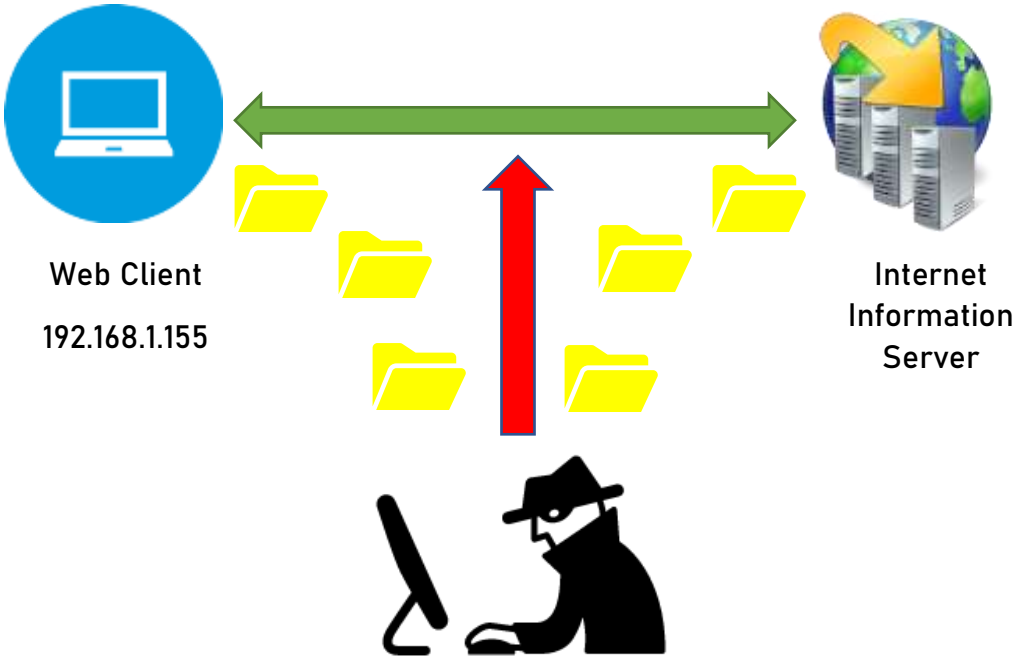


Ilustración 8: Discover the vulnerability and explain

With the WIRESHARK tool we have analyzed the packets sent and received with the http protocol where we found its vulnerability since it shows the username and password to access the DVWA application consulted in the MYSQL database.

2. SECTION TWO

In section number two we are going to experience how to perform a man attack in the middle. For this we are going to need three machines, two of them sharing information and a third one that is going to be the spy or attacker, all within the same network.



When you have the machines running, check the IP's and MAC addresses, you would have to ping between all the machines.

Web Client	192.168.0.14
Internet Information Server	192.168.0.15
Man in the middle	192.168.0.17
MAC Web Client	08:00:27:83:74:da
MAC Internet Information Server	3C:91:80:90:68:0D
MAC Man in the middle	08:00:27:a6:1f:86

Verification of the connection of all machines.

Web Client:

```
simplerisk@simplerisk:~$ ping 192.168.0.14
PING 192.168.0.14 (192.168.0.14) 56(84) bytes of data.
64 bytes from 192.168.0.14: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 192.168.0.14: icmp_seq=2 ttl=64 time=0.036 ms
^C
--- 192.168.0.14 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1023ms
rtt min/avg/max/mdev = 0.021/0.028/0.036/0.009 ms
simplerisk@simplerisk:~$ ping 192.168.0.15
PING 192.168.0.15 (192.168.0.15) 56(84) bytes of data.
64 bytes from 192.168.0.15: icmp_seq=1 ttl=128 time=0.696 ms
64 bytes from 192.168.0.15: icmp_seq=2 ttl=128 time=0.412 ms
^C
--- 192.168.0.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.412/0.554/0.696/0.142 ms
simplerisk@simplerisk:~$ ping 192.168.0.17
PING 192.168.0.17 (192.168.0.17) 56(84) bytes of data.
^C
--- 192.168.0.17 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3050ms

simplerisk@simplerisk:~$ ping 192.168.0.17
PING 192.168.0.17 (192.168.0.17) 56(84) bytes of data.
64 bytes from 192.168.0.17: icmp_seq=1 ttl=64 time=0.871 ms
64 bytes from 192.168.0.17: icmp_seq=2 ttl=64 time=1.16 ms
^C
--- 192.168.0.17 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.871/1.017/1.164/0.149 ms
simplerisk@simplerisk:~$ _
```

Ilustración 9:Verification of the connection

Internet Information Server:

```
C:\WINDOWS\system32>ping 192.168.0.14

Haciendo ping a 192.168.0.14 con 32 bytes de datos:
Respuesta desde 192.168.0.14: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.0.14: bytes=32 tiempo=281ms TTL=64

Estadísticas de ping para 192.168.0.14:
    Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 281ms, Media = 140ms
Control-C
^C
C:\WINDOWS\system32>ping 192.168.0.15

Haciendo ping a 192.168.0.15 con 32 bytes de datos:
Respuesta desde 192.168.0.15: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.15: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.0.15:
    Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
Control-C
^C
C:\WINDOWS\system32>ping 192.168.0.17

Haciendo ping a 192.168.0.17 con 32 bytes de datos:
Respuesta desde 192.168.0.17: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.0.17: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.0.17:
    Paquetes: enviados = 2, recibidos = 2, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms
Control-C
^C
C:\WINDOWS\system32>
```

Ilustración 10:Verification of the connection

Man in the middle:

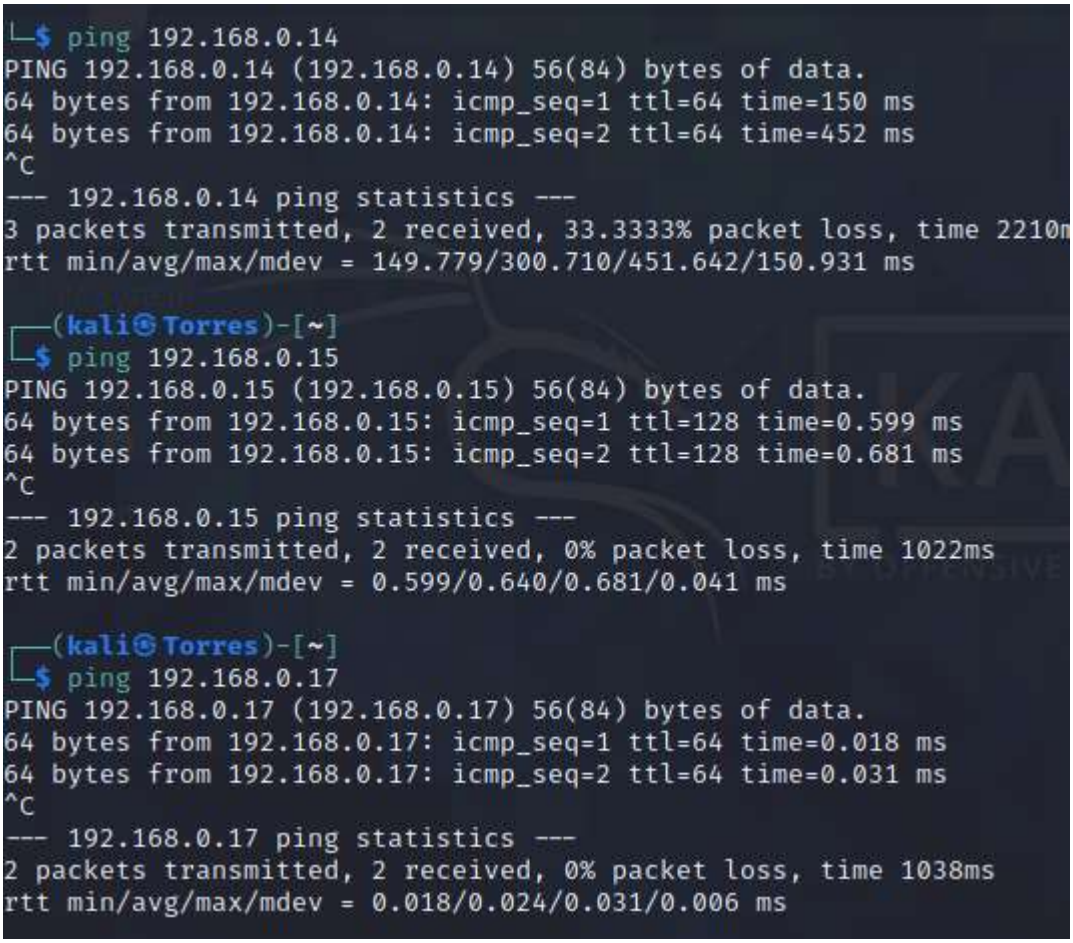


Ilustración 11: Verification of the connection

Note that MAC addresses must be different:

1. Run the following command "ifconfig eth0 down".
2. Run the following command "macchanger -A eth0" and explain what -A is for in the command.

-A: set random vendor MAC of any Kind.

3. Run the following command "ifconfig eth0 up".

```
┌──(root💀Torres)-[/home/kali]
└─# ifconfig eth0 down

┌──(root💀Torres)-[/home/kali]
└─# macchanger -A eth0
Current MAC:   08:00:27:a6:1f:86 (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:a6:1f:86 (CADMUS COMPUTER SYSTEMS)
New MAC:       00:16:ef:2d:0f:1d (Koko Fitness, Inc.)

┌──(root💀Torres)-[/home/kali]
└─# ifconfig eth0 down

┌──(root💀Torres)-[/home/kali]
└─# ifconfig eth0 up
```

Ilustración 12: run command

4. Verify with the command "ifconfig" that the MAC address was changed.

```
┌──(root💀Torres)-[/home/kali]
└─# ifconfig | grep ether
        ether 00:16:ef:2d:0f:1d  txqueuelen 1000  (Ethernet)

┌──(root💀Torres)-[/home/kali]
└─#
```

Ilustración 13: run command

Ping

5. Finally, check the ARP tables of each machine with the command "arp" and "arp -a" in Windows.

```
192.168.0.15             ether   3c:91:80:90:68:0d   C
eth0
192.168.0.14             ether   08:00:27:83:74:da   C
eth0
192.168.0.1              ether   64:55:b1:39:85:fe   C
eth0

┌──(kali㉿Torres)-[~]
└─$
```

Ilustración 14: run command arp

```
Interfaz: 192.168.0.15 --- 0x17
  Dirección de Internet          Dirección física      Tipo
  192.168.0.1           64-55-b1-39-85-fe     dinámico
  192.168.0.14          08-00-27-83-74-da     dinámico
  192.168.0.17          08-00-27-a6-1f-86     dinámico
  192.168.0.21          08-00-27-a6-1f-86     dinámico
  192.168.0.255         ff-ff-ff-ff-ff-ff     estático
  224.0.0.2             01-00-5e-00-00-02     estático
  224.0.0.22            01-00-5e-00-00-16     estático
  224.0.0.251           01-00-5e-00-00-fb     estático
  224.0.0.252           01-00-5e-00-00-fc     estático
  239.255.255.250       01-00-5e-7f-ff-fa     estático
  255.255.255.255       ff-ff-ff-ff-ff-ff     estático
```

Ilustración 15: run command arp

```
simplerisk@simplerisk:~$ arp -a
? (192.168.0.15) at 3c:91:80:90:68:0d [ether] on eth0
? (192.168.0.21) at 08:00:27:a6:1f:86 [ether] on eth0
? (192.168.0.16) at <incomplete> on eth0
? (192.168.0.17) at 08:00:27:a6:1f:86 [ether] on eth0
? (192.168.0.1) at 64:55:b1:39:85:fe [ether] on eth0
```

Ilustración 16: run command arp

3. SECTION THREE

Config the internet information server with the following steps:

1. Go to control panel.
2. Go to programs and features.
3. Go to turn windows features on or off.
4. Search the item “Internet information services” and active.
5. Save the changes.
6. Change the folder “wwwroot” by the given. The folder is locater a C:/inetpub/wwwroot
7. Verified that the page http://192.168.56.1/ /Loggin.html works.

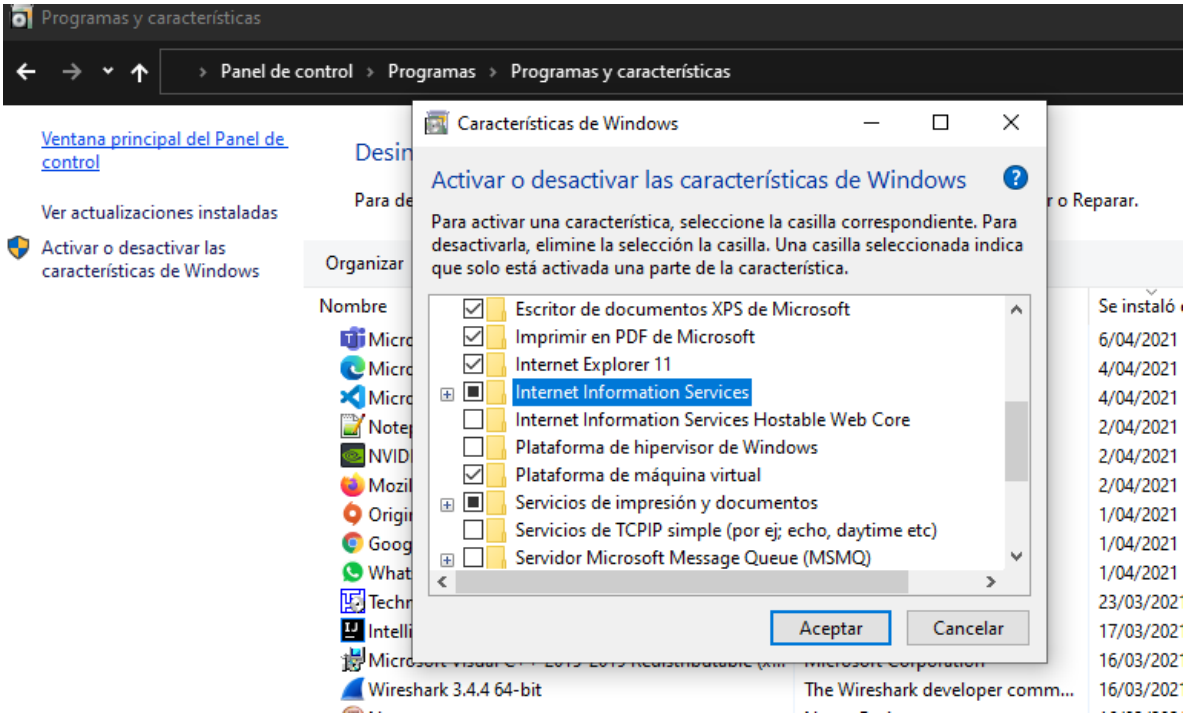


Ilustración 17: Internet information services

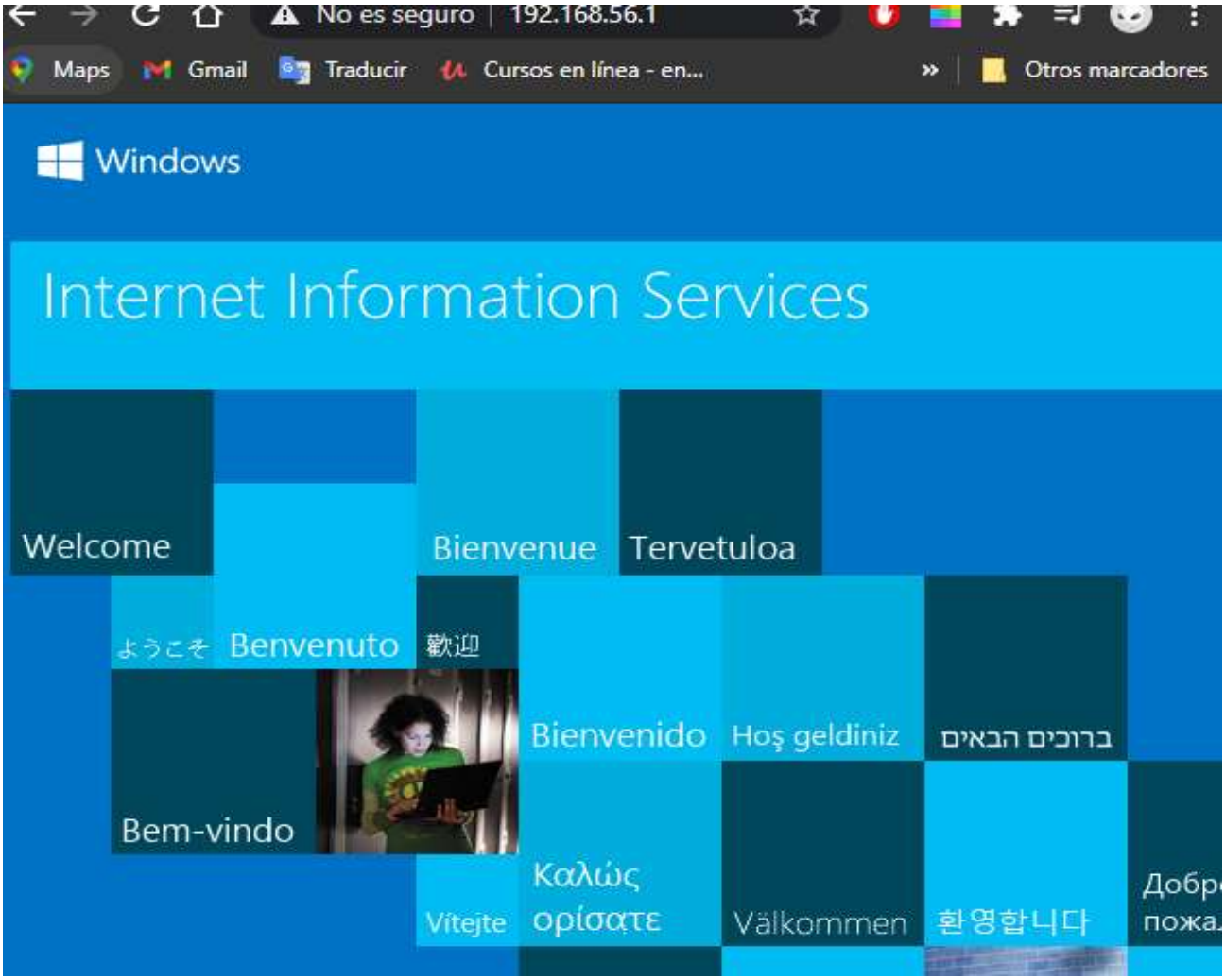


Ilustración 18: run Internet information services