

MIT WORLD PEACE UNIVERSITY

Wireless Devices and Mobile Security
Third Year B. Tech, Semester 5

INSTALLATION AND CONFIGURATION OF ANY WIFI
TRAFFIC ANALYSER TOOL.

LAB ASSIGNMENT 9

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

November 27, 2023

Contents

1 Aim	1
2 Objectives	1
3 Theory	1
3.1 Wireshark	1
3.1.1 Installation	1
3.1.2 Working	2
3.1.3 Pros	2
3.1.4 Cons	2
3.2 AirCrack	3
3.2.1 Installation	3
3.2.2 Working	3
3.2.3 Pros	4
3.2.4 Cons	4
3.3 AirSnort	4
3.3.1 Installation	5
3.3.2 Working	5
3.3.3 Pros	5
3.3.4 Cons	5
4 Platform	5
5 Working Screenshots	5
6 Conclusion	5
7 FAQ	6
References	7

1 Aim

Install, configure and demonstrate any one Wi-Fi traffic analyzer using sniffing tools such as Wireshark, AirCrack, AirSnort, etc.

2 Objectives

1. To install Wireshark on the system.
2. To capture packets using Wireshark.
3. To analyse the captured packets.

3 Theory

3.1 Wireshark

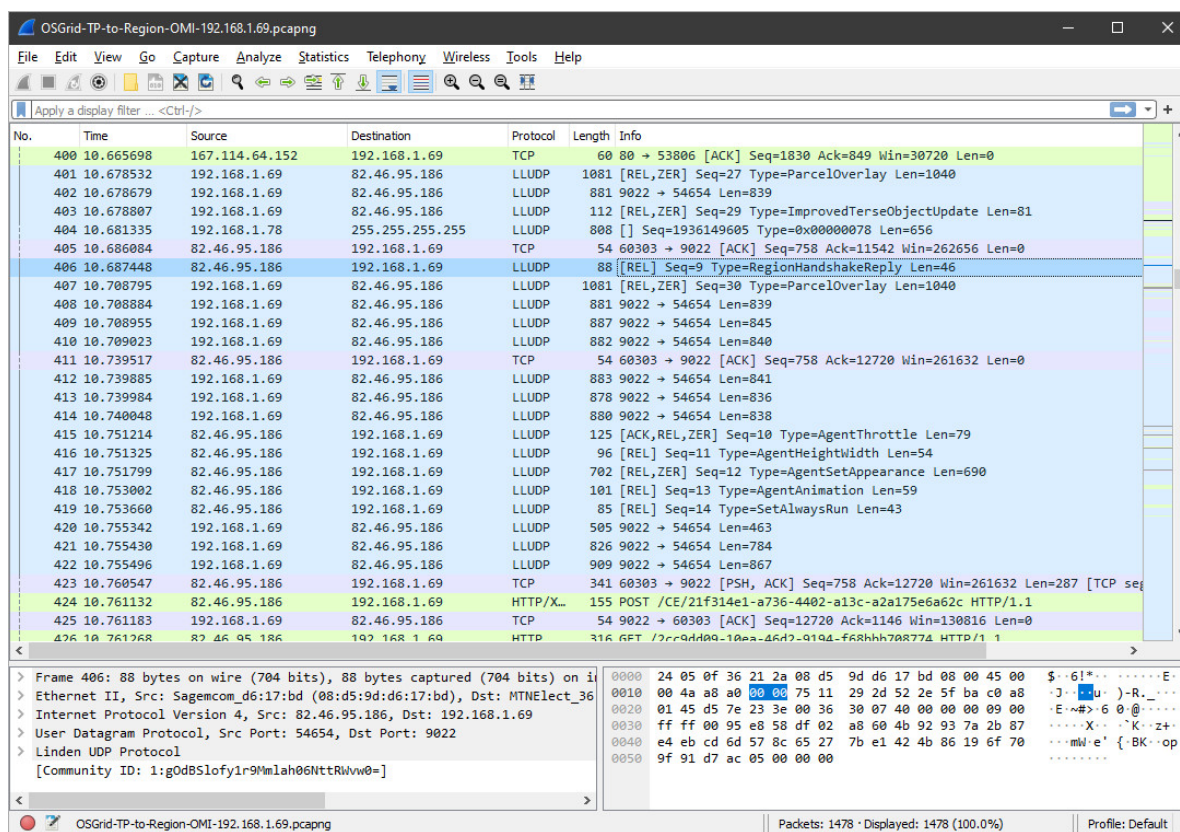


Figure 1: Wireshark GUI

3.1.1 Installation

- **Procedure:** Wireshark can be installed on various operating systems, including Windows, macOS, and Linux. Visit the official Wireshark website (<https://www.wireshark.org/>) and follow the installation instructions for your specific platform.

- **Dependencies:** Wireshark may require the installation of WinPcap (Windows), libpcap (Linux), or npcap (Windows) for packet capture.

3.1.2 Working

- Wireshark captures and analyzes packets on a network in real-time.
- Users can apply various filters to focus on specific types of traffic.
- The captured data can be displayed in different formats, facilitating detailed protocol analysis.

3.1.3 Pros

- User-friendly interface with powerful features.
- Extensive protocol support for in-depth analysis.
- Active community and regular updates.

3.1.4 Cons

- May consume significant system resources during packet capture.
- Beginners might find the wealth of features overwhelming.
- Limited to the capabilities of the network interface card (NIC).

3.2 AirCrack

```

kali@kali:~$ aircrack-ng --help

Aircrack-ng 1.6 - (C) 2006-2020 Thomas d'Otreppe
https://www.aircrack-ng.org

usage: aircrack-ng [options] <input file(s)>

Common options:

-a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
-e <essid> : target selection: network identifier
-b <bssid> : target selection: access point's MAC
-p <nbcpu> : # of CPU to use (default: all CPUs)
-q : enable quiet mode (no status output)
-C <macs> : merge the given APs to a virtual one
-l <file> : write key to file. Overwrites file.

Static WEP cracking options:

-c : search alpha-numeric characters only
-t : search binary coded decimal chr only
-h : search the numeric key for Fritz!BOX
-d <mask> : use masking of the key (A1:XX:CF:YY)
-m <maddr> : MAC address to filter usable packets
-n <nbits> : WEP key length : 64/128/152/256/512
-i <index> : WEP key index (1 to 4), default: any
-f <fudge> : bruteforce fudge factor, default: 2
-k <korek> : disable one attack method (1 to 17)
-x or -x0 : disable bruteforce for last keybytes
-x1 : last keybyte bruteforcing (default)
-x2 : enable last 2 keybytes bruteforcing
-X : disable bruteforce multithreading
-y : experimental single bruteforce mode
-K : use only old KoreK attacks (pre-PTW)
-s : show the key in ASCII while cracking
-M <num> : specify maximum number of IVs to use
-D : WEP decloak, skips broken keystreams
-P <num> : PTW debug: 1: disable Klein, 2: PTW
-1 : run only 1 try to crack key with PTW
-V : run in visual inspection mode

WEP and WPA-PSK cracking options:

-w <words> : path to wordlist(s) filename(s)
-N <file> : path to new session filename
-R <file> : path to existing session filename

WPA-PSK options:

-R <file> : path to existing session filename

WPA-PSK options:

-E <file> : create EWSA Project file v3
-I <str> : PMKID string (hashcat -m 16800)
-j <file> : create Hashcat v3.6+ file (HCCAPX)
-J <file> : create Hashcat file (HCCAP)
-S : WPA cracking speed test
-Z <sec> : WPA cracking speed test length of execution.
-r <DB> : path to airolib-ng database (Cannot be used with -w)

SIMD selection:

--simd-list : Show a list of the available SIMD architectures, for this machine.
--simd=<option> : Use specific SIMD architecture

<option> may be one of the following, depending on your platform:

generic
avx512
avx2
avx
sse2
altivec
power8
asimd
neon

Other options:

-u : Displays # of CPUs & SIMD support
--help : Displays this usage screen

kali@kali:~$

```

Figure 2: Aircrack

3.2.1 Installation

- **Procedure:** AirCrack-ng, a suite of wireless network security tools, can be installed on various platforms. Detailed installation instructions are available on the official website (<https://www.aircrack-ng.org/>).
- **Dependencies:** AirCrack-ng relies on libpcap and other libraries for packet capture and analysis.

3.2.2 Working

- AirCrack-ng is primarily used for assessing the security of Wi-Fi networks.
- It includes tools for capturing, analyzing, and cracking WEP and WPA/WPA2-PSK keys.

- Supports various attacks like packet injection and de-authentication to test network vulnerabilities.

3.2.3 Pros

- Comprehensive suite for wireless network security.
- Active development community and frequent updates.
- Capable of testing the security of WEP and WPA/WPA2-PSK.

3.2.4 Cons

- Requires a good understanding of wireless networks and security concepts.
- Use in unauthorized networks may violate ethical and legal standards.
- Effectiveness is dependent on the strength of encryption used.

3.3 AirSnort

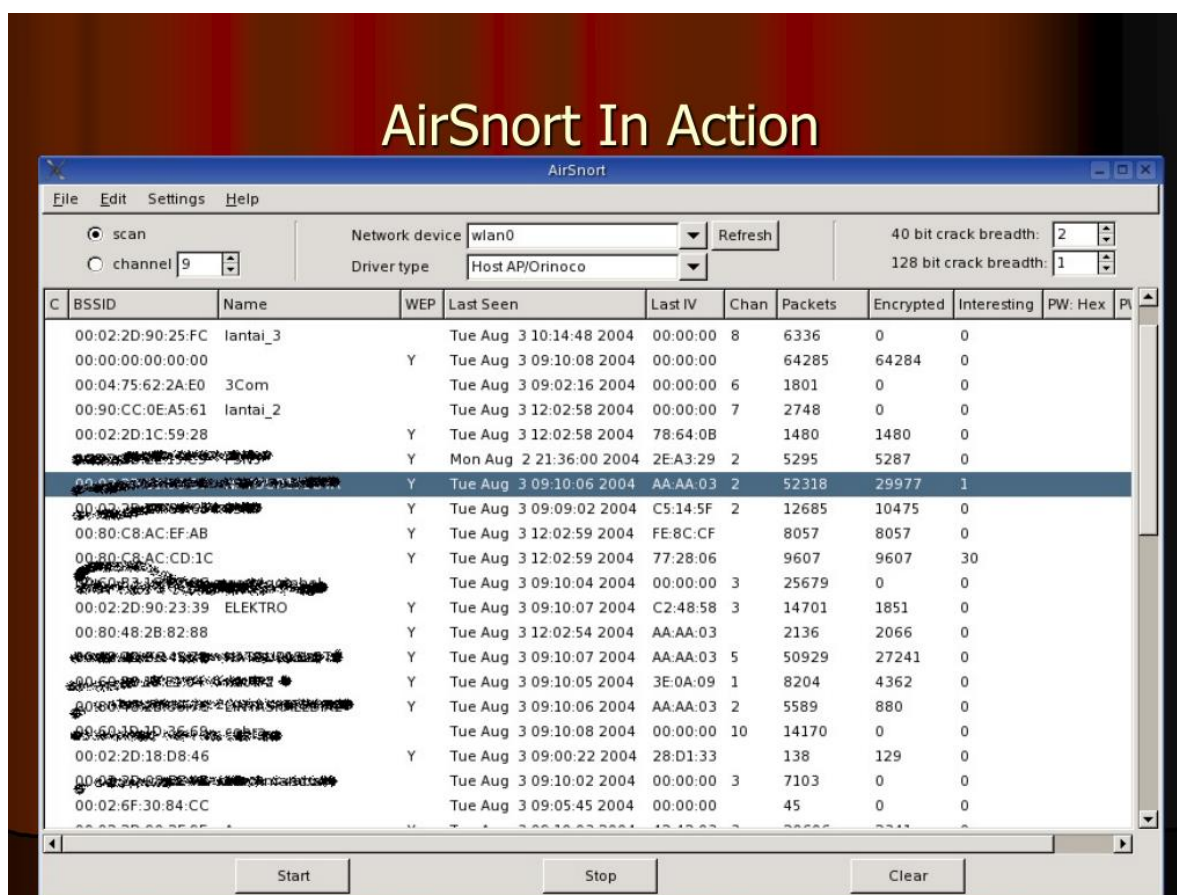


Figure 3: Airsnort

3.3.1 Installation

- **Procedure:** AirSnort, a wireless LAN (WLAN) tool, is no longer actively maintained. Installation may vary based on the available repositories or archived versions.
- **Dependencies:** Originally designed for Linux, it relies on libpcap and other libraries for packet capture.

3.3.2 Working

- AirSnort was designed to crack WEP encryption keys by capturing data packets and analyzing them.
- It focused on exploiting weaknesses in the WEP algorithm to recover network passwords.
- Due to its outdated nature, it may not be effective against modern, more secure encryption standards.

3.3.3 Pros

- Historically used for educational purposes to highlight WEP vulnerabilities.
- Provided insights into the weaknesses of early wireless encryption

3.3.4 Cons

- Outdated and no longer actively maintained.
- Limited effectiveness against modern and more secure Wi-Fi encryption.
- Not recommended for practical use in contemporary security assessments.

4 Platform

Operating System: Arch Linux x86-64

IDEs or Text Editors Used: Visual Studio Code

Compilers or Interpreters: Python 3.10.1

5 Working Screenshots

6 Conclusion

Thus, the installation and configuration of Any Wifi Traffic Analyser Tool was successfully done. We installed Wireshark, captured packets and analysed them.

7 FAQ

1. List the different open source tool to capture packet. Also, write its features.

Packet Capture Tools:

- *Wireshark*:
 - **Features:** Wireshark is a widely-used open-source packet analyzer. It allows real-time packet capture and display.
 - **Additional Features:** Protocol analysis, deep inspection of hundreds of protocols, live capture, and offline analysis.
 - **Reference:** [1]
- *Tshark*:
 - **Features:** Tshark is the command-line version of Wireshark. It offers similar features for packet capture and analysis.
 - **Additional Features:** Scriptable using Lua, supports various capture file formats.
 - **Reference:** [2]
- *Tcpdump*:
 - **Features:** Tcpdump is a command-line packet analyzer for Unix-like systems.
 - **Additional Features:** Filters for specific protocols, customizable output formats.
 - **Reference:** [3]

2. Which mode NIC uses for *Ethereal* / packet sniffing?

NIC Modes for Ethereal/Packet Sniffing: NIC primarily uses the *Promiscuous Mode* for Ethereal/packet sniffing. In this mode, the NIC captures all traffic on the network, regardless of the destination address.

3. Which wireshark filter can be used to monitor outgoing packets from a specific system on the network?

Wireshark Filter for Monitoring Outgoing Packets: To monitor outgoing packets from a specific system on the network using Wireshark, you can use the following filter:

```
ip.src == <source_IP_address>
```

Replace <source_IP_address> with the actual IP address of the system you want to monitor.

References

- [1] Wireshark.
Website: <https://www.wireshark.org/>
- [2] Tshark.
Website: <https://www.wireshark.org/docs/man-pages/tshark.html>
- [3] Tcpdump.
Website: <https://www.tcpdump.org/>
- [4] AirCrack-ng.
Website: <https://www.aircrack-ng.org/>
- [5] AirSnort.
Website: <https://sourceforge.net/projects/airsnort/>