A Flexible Framework for Rogue Access Point Detection

Ricardo Gonçalves[1,2], Manuel Eduardo Correia[1,2] and Pedro Brandão[1,3]
[1]*CS Dept., Faculty of Sciences, University of Porto, Portugal*
[2]*CRACS, INESC-TEC, Porto, Portugal*
[3]*Instituto de Telecomunicações, Lisboa, Portugal*

Keywords: Wireless Networks, Rogue Access Points, Wireless Security, Detection Systems.

Abstract: The society's requirement for constant connectivity, leads to the need for an increasing number of available Wi-Fi Access Points (APs). These can be located almost everywhere: schools, coffee shops, shopping malls, airports, trains, buses. This proliferation raises problems of trustworthiness and cost-effective difficulties for verifying such security. In order to address these issues, it is necessary to detect effectively Rogue Access Points (RAPs). There are open source solutions and others developed within enterprises for commercial purposes. Relative to the latter, it has become obvious that they are not accessible to everyone due to their high costs, and the former do not address all the types of RAPs. In this paper, we research these solutions and do a thorough survey study of the most commonly used and recent Wi-Fi type of attacks. Based on this knowledge we developed a solution to detect RAPs, which covers the most commonly known attacks. This proposed solution, is a modular framework composed of *Scanners*, *Detectors* and *Actuators*, which are responsible for scanning for available APs, apply a set of heuristics to detect them and apply a countermeasure mechanism.

1 INTRODUCTION

Today, The relevance of the Internet is something unmatched with a past not so far away, where tasks were mainly related to information search conducted via a desktop computer. Currently, the tasks made using the Internet evolved to banking, shopping, messaging, video calls, gaming, social networks, geolocation, etc. In short, a lot of essential day-to-day activities now require the Internet. In addition to the evolution in terms of use and services, the Internet has also transformed itself in the eyes of its users. So much so, that with the arrival and massification of mobile devices, we have witnessed a huge migration of network services usage to wireless networks. This turned the Internet into an almost ubiquitous service, with users demanding access everywhere at any time. However, cost consideration still plays a major role in how users choose to connect to the Internet. Mobile data (cellular) contracts are still expensive and capped, but available everywhere (indoor and outdoor). Free Wi-Fi access provided by an Access Points (APs), are generally indoors or in otherwise selected public places. Conscientious users, therefore, generally prefer Wi-Fi access (faster and free) and only recur to mobile data when Wi-Fi access is not immediately available or is of very poor quality. Mobile devices are even pre-configured to prefer Wi-Fi access over mobile data.

In a generic scenario of mobile Internet usage several APs are often used. But when a user requests a wireless Internet connection, how can they be sure that they are connecting to a trusted source/device?

In order to understand and answer this question, our work addresses the problem of detecting false Wi-Fi APs that were not installed by an authorized network administrator and can be used for malicious purposes, thus becoming Rogue Access Points (RAPs). The word "rogue" clarifies the malicious intentions of this type of AP.

When an attacker sets up a RAP it will monitor the traffic that goes through it and will be able to perform several types of attacks, with Man in the Middle (Schmoyer et al., 2004) being one of the most popular.

In the first part of this paper we discuss some background knowledge needed to address this issue, specifically we focus on the types of RAPs and its more popular types of attacks. Then, we present and discuss other related work in order to put into perspective some approaches that have already been proposed in the literature. In the second part of this paper we present the framework architecture for our proposal and explain in detail its design options and where it innovates. We then finish by describing and discussing the effectiveness of a proof of concept implementation of our ideas that we deployed in the field.

Gonçalves, R., Correia, M. and Brandão, P.
A Flexible Framework for Rogue Access Point Detection.
DOI: 10.5220/0006832904660471
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 466-471
ISBN: 978-989-758-319-3

2 BACKGROUND

In this section we start by presenting the types of RAPs from the current literature. Then, we overview some of the most common types of Wi-Fi attacks that can be deployed from RAPs. Finally, we describe some proposed countermeasures that can be used to prevent or mitigate these attacks.

2.1 Types of RAPs

To classify the different types of RAPs, we divide them into four general categories: *Evil-Twin*, *Improperly Configured*, *Unauthorized*, and *Compromised*. This is the nomenclature often used in the literature.

Evil Twin. In the IEEE 802.11 standard (IEEE Wireless LAN Working Group, 2016) there are only two identifiers for users to recognize an AP: the Service Set Identifier (SSID) and the Basic Service Set Identifier (BSSID). However, these identifiers can be easily spoofed. The act of cloning a legitimate AP generates an Evil Twin AP.

Evil Twin RAPs can exist in two forms: *Coexistence* and *Replacement*. In the first, the legitimate AP and the Evil Twin coexist in the same location. The attacker increases the RAP's signal strength to force users to connect to it, as the IEEE 802.11 standard states that WLAN clients must connect to the AP that has the strongest signal. In the Replacement type, the Evil Twin replaces the legitimate AP by shutting it down, using an active attack on it. To remain undetected to its victims, the RAP needs to have an Internet connection (or connectivity to the same network as the valid AP) while in the first case it could relay the packets through the legitimate AP, as long as it could connect to it.

Improperly Configured AP. This type of RAP, as its name suggests, is an AP which was improperly configured. There is no adversary involved in the creation process. This can happen when, for instance, an administrator does not use robust authentication and encryption settings, leading to a network that can be easily intruded.The APs can also become vulnerable after software updates (Ma et al., 2008) or the lack of recent updates. This type of RAP may lead to backdoors in an organization's network infrastructure.

Unauthorized AP. This type of RAP is installed by an employee or naïve user without the network administrator's permission (Whelan et al., 2011). This RAP is connected to the wired side of the network (like a legitimate AP) and thus it is considered part of the WLAN. A RAP of this kind is installed by uninformed users for their own convenience, i.e., access to some internal network resources, but it can also be deployed with malicious intentions. Regardless, it will create an unauthorized entry point into an organization's network and compromise its security.

Compromised AP. When the shared keys used to secure the network's communications get compromised (Ma et al., 2008), the AP becomes a RAP. Thus, the attacker that obtains the keys, will be able to join the network.

2.2 Wi-Fi Attacks

RAP with Stronger Signal. In this specific attack, the RAP behaves as an Evil Twin. Here the attacker entices the victim to connect to the Evil Twin AP.

For this attack to work, the attacker sets up the wireless card into promiscuous or monitor mode in order to gather information about the nearby APs and its clients. The RAP can now be created with the same SSID, BSSID and wireless medium physical channel, and increasing the AP's signal power. Usually in this type of attacks the Evil Twin AP is created with Open authentication which eases its detection if the original AP has some authentication enabled.

These two factors are paramount, because if no authentication is used by the RAP, the signal strength will not matter. A node will only roam when the APs has the same SSID, security mode and credentials. Only if these conditions are met, can the target victim be made to automatically switch to another AP with a better signal strength.

De-authentication Attacks. To enable roaming of user, attackers often use de-authentication attacks to force their victims to connect to the RAP. This attack uses the special de-authentication frame of the 802.11 standard. The attacker sends it to the desired victim on behalf of the legitimate AP and consequently the victim will be de-authenticated from that AP. Then, in the process of re-authentication, the victim will automatically reconnect to the strongest AP.

Karma Attacks. In these attacks, an automatic process is used for cloning an AP, where the malicious node listens for other wireless devices' requests, and creates a RAPs on demand. This is possible because of a vulnerability in the method of discovering available Wi-Fi networks (Dai Zovi and Macaulay, 2005).

When a client wants to join a Wi-Fi network, a scan is made for the available networks (Probe request), and then one network is selected, matching a Preferred Network List (PNL). The attacker will listen for the probe requests that are sent with the SSIDs of the client's preferred networks. Thus, the attacker will be able to respond to the probe request by sending a probe response matching the network requested.

RAP with Radius Server. This attack targets WPA Enterprise APs. For this there are a couple of tools that can be used by an attacker: `hostapd-wpe`, and `freeradius-wpe`[1]. These tools implement the IEEE 802.1X Authenticator and the Authentication Server impersonation attacks in order to obtain client credentials and establish connectivity to the client.

This RAPs taxonomy guarantees the scope of the project and the possible types of attacks creates the base cases for our proof of concept.

2.3 Countermeasures

This sub-section summarizes some countermeasures and solutions to mitigate the described attacks.

We identify and differentiate between passive and active countermeasures, and classify if the solution can detect more than one type of RAP. There are techniques that need protocol modifications and some require the use of hardware. In passive techniques, the detector radio listens on each channel for the periodic beacons sent by an AP. While in actives, the detector transmits a probe request (to inquire about available WLANs/SSIDs) and listens for a probe response from an AP. However, a RAP usually does not reply to active probing, so passive methods are often preferred.

The majority of techniques to detect RAP have focused on Evil Twin RAP types and consequently will also detect RAP-based de-authentication/disassociation attacks. There are techniques to detect Unauthorized APs (Whelan et al., 2011), but practical techniques for the detection of Compromised APs are scarce. Moreover, there is no single technique to detect all RAP types (Alotaibi and Elleithy, 2016).

The techniques in Table 1 do not cover all the types of RAPs and some of them even lack a proper proof of concept tool for testing and benchmarking purposes. However, they are the basis for the design of some detection heuristics used in our work. During our research, we have verified that there are several freely available tools for deploying RAPs and performing Wi-Fi attacks, e.g., Airbase-ng, hostapd-wpe, PineAP, Pwnie Express, contrasting with the flagrant lack of tools that can be employed to detect them, e.g., sentrygun[2] and EvilAP_Defender[3].

Table 1: Detection techniques summary.

Protection against Evil Twin	• Active Approach (Bratus et al., 2008) • Timing-based (Han et al., 2011) • EAP-SWAT (Bauer et al., 2008) (Han et al., 2011) • CETAD (Mustafa and Xu, 2014)
Unauthorized countermeasures	• Unauthorized Approach (Yan et al., 2009) • Agent-based (Chirumamilla and Ramamurthy, 2003)
Protection against multiple RAPs	• DWSA (Branch et al., 2004) • Hybrid RAP (Ma et al., 2008) • Multi-Agent (Sriram et al., 2010)

3 PROPOSED MODEL

The developed framework combines the most effective techniques from the state of the art, and complements them with new approaches to provide more accurate results in the detection process. It is an open source command line application[4] for Linux Systems, developed in Python, and with minimal dependencies.

This framework is a RAP Detector that works both in a passive and active mode. It can be used by any type of user, i.e., it can be configured with a profile of a network administrator with knowledge about the internal network, or as a simple user with no knowledge of the network (s)he is using.

The architecture is composed by the main application, and it is connected to a set of modules: Scanners, Detectors and Actuators.

Currently our detector node (a laptop running the framework) is stationary. We are evaluating having distributed nodes that send information to a central server. The architecture of this central node would be very similar to the one being presented here, where some detectors and/or actuators modules would be used to communicate with the remote nodes.

3.1 Profiles

The application is designed to use configuration files, i.e., profiles. These profiles have information about the network(s) the user connects to. There is a built in profile for open/free APs that are usually offered by Internet Service Providers (ISPs)[5]. The user configu-

[1] https://github.com/OpenSecurityResearch/hostapd-wpe and http://www.willhackforsushi.com/?page_id=37, respectively.

[2] https://github.com/s0lst1c3/sentrygun

[3] https://github.com/moha99sa/EvilAP_Defender

[4] https://github.com/anotherik/RogueAP-Detector

[5] Currently they are tuned for Portuguese ISPs.

red profiles are optional and the application will run without them. However, as expected this will generate much less accurate results in the detection process, since some heuristics will not be used.

3.2 Modules

The main application is connected to all the modules. Scanners are passive modules with methods implemented to monitor the network in order to find nearby APs. After being started, the application will always be scanning. The information provided will *feed* the Detectors, where a set of detection heuristics will sweep the scanned APs and when a specific AP reaches some suspicion level it will interact with the Actuators to perform active techniques.

For **Scanners** to scan the Wi-Fi channels we have configured two different modules: one that uses the `iwlist` Linux command and another that uses the `scapy` packet manipulation program.

On the **Detector** modules, one type of heuristic present is the classic approach of whitelisting, where we use the information from the configuration profile regarding SSIDs and their expected BSSIDs. Since this is likely to be bypassed (BSSID spoofing), we also compare the Encryption type used by the AP.

Another heuristic is the variation of the signal strength. As already described, this tool is designed to be stationary, and normal APs are also fixed. With this in mind, we can estimate a baseline for the signal strength and use this information to improve the detection process.

In particular the algorithm uses an `auth_rssi` defined by the user for the authorized AP's RSSI, and to avoid fluctuations a delta accounts for variations. As such, the read RSSI value must fall in the allowed range of $[auth\_rssi - delta; auth\_rssi + delta]$. If this is triggered, the user is prompted to associate to the AP and continue to the active tests (Actuators).

The described heuristics assume that a configuration profile is loaded when the application runs, otherwise they will not be applied. In such case, a *no knowledge* method is always run by the application, where it conducts simple analyses on the scanned APs. A simple case is looking for APs with the same SSID and different security being used.

Another heuristic used by the application is a free Wi-Fi's authenticity validation. This takes advantage of the BSSID pattern of the studied free Wi-Fis. As an example, the free Wi-Fi provided by the ISP NOS (Portuguese media company), is generated from the router of a personal network and the BSSID generated for the free network follows a specific pattern. It is the increment by one unit of the last byte of the personal network's BSSID. We use this information to analyse the BSSIDs of the scanned free Wi-Fis.

This tool also has a *blacklist* passive heuristic for RAPs generated from known Wi-Fi attacking tools. We configured two methods to detect Pineapple AP RAPs and APs where the BSSID manufacturer is Alfa. In the case of PineAP, the default BSSID contains `13:37` (leet speech), hence this heuristic can identify some RAPs configured by inexperienced attackers. The Alfa condition is supported by the fact that these Wi-Fi cards are mainly used for Wi-Fi attacks.

The passive detectors also have a heuristic to look for de-authentication frames and alert the user. For this we use the `scapy` packet manipulation tool.

The **Actuators** are composed by a set of active detectors, a defensive mechanism and a honeypot.

The active detectors are performed when we have knowledge about the scanned AP, i.e., the scanned AP matches an AP from our list of authorized APs or for open networks that fail the authenticity validation and need further confirmation. The detectors under the Actuators type of modules perform active operations over the target AP, specifically, they associate to the AP and then run the heuristics:

- **Associate:** tries to associate to the AP and gather information on the network addresses, comparing to the values on the profile;
- **Traceroute:** compares network paths to configured destinations with the paths defined in the profile;
- **Fingerprint:** verifies if the OS fingerprint and services of the tested AP corresponds to the ones of an authorized AP.

The Actuators set of modules is also composed of an hybrid mechanism to detect Karma attacks. In passive mode, we wait for probe responses from the same AP to different probe requests. In the active mode, we send probe requests for generated fictitious SSIDs. If an AP sends responses for the fabricated SSIDs, a Karma attack is spotted. In cases like this, where it is obvious that RAPs are being created, a defensive approach can be used by sending IEEE 802.11 de-authentication packets to the clients associated with the detected RAP.

Last we have the honeypot module, which is a RAP created and controlled by the application. If an attacker targets this AP, i.e., spoofing it, we have the same confirmation as for the generated SSIDs.

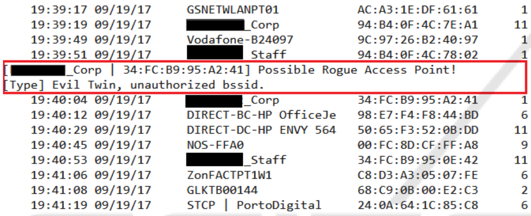
4 RESULTS

This section describes the results obtained from the RAP detections performed on two different scenarios. In the first, the RAPs were deployed by an information security specialist impersonating an attacker, where we did not know the types of RAP. On the second, the RAPs were configured and deployed by us on a contained environment, to replicate all the attacks and RAPs covered by the detection tool.

4.1 First Scenario - Unknown Setup

The tests from the first scenario were conducted in an Enterprise environment. In such environments, the Corporate APs are usually configured with WPA2 Enterprise, which is associated with directory service credentials to authenticate in the network.

Figure 1 has the first set of detections of the Rogue AP detector. In the screenshot the produced alert is selected, and from the figure it is possible to understand that an unauthorized BSSID was detected. The network advertised by the RAP follows the alert.



```
   19:39:17 09/19/17     GSNETWLANPT01           AC:A3:1E:DF:61:61      1
   19:39:19 09/19/17     [redacted]_Corp         94:B4:0F:4C:7E:A1     11
   19:39:49 09/19/17     Vodafone-B24097         9C:97:26:B2:40:97      1
   19:39:51 09/19/17     [redacted] Staff        94:B4:0F:4C:78:02      1
[[redacted]_Corp | 34:FC:B9:95:A2:41] Possible Rogue Access Point!
[Type] Evil Twin, unauthorized bssid.
   19:40:04 09/19/17     [redacted]_Corp         34:FC:B9:95:A2:41      1
   19:40:12 09/19/17     DIRECT-BC-HP OfficeJe   98:E7:F4:F8:44:BD      6
   19:40:29 09/19/17     DIRECT-DC-HP ENVY 564   50:65:F3:52:0B:DD     11
   19:40:45 09/19/17     NOS-FFA0                00:FC:8D:CF:FF:A8      9
   19:40:53 09/19/17     [redacted]_Staff        34:FC:B9:95:0E:42     11
   19:41:06 09/19/17     ZonFACTPT1W1            C8:D3:A3:05:07:FE      6
   19:41:08 09/19/17     GLKTB00144              68:C9:0B:00:E2:C3      2
   19:41:19 09/19/17     STCP | PortoDigital     24:0A:64:1C:85:C8      6
```

Figure 1: Unknown scenario, RAP with same SSID and different BSSID.

In this test case the RAP failed the BSSID parameter which caused its discovery. To bypass this detection the attacker modified the BSSID of the RAP to one of the authorized list, but in order to succeed, i.e., to have clients trying to associate with the RAP, the RSSI had to be increased. In this case, the tool detected the signal strength strange behaviour and prompted the user to associate to the AP. This association failed corroborating that it was a RAP.

In this attack type, where a RAP is impersonating a WPA2 Enterprise network, the objective is to capture the victim's credentials, and for this it only requires capturing the challenge/response authentication. This means that usually, these attacks do not actually authenticate the user with the RAP.

In the last test case of the first scenario we detect a RAP produced by a Pineapple AP with a Karma attack. The attack used the default name of Pineapple and the BSSID with *13:37*, which our tool detects.

4.2 Second Scenario - Controlled Setup

For the second scenario, we only describe the results for the heuristics not yet tested, namely: RAP with different encryption, validation of free Wi-Fis authenticity, and detection of de-authentication attacks.

Figure 2 shows the validated free Wi-Fis following the algorithm described in section 3.2, and the case of an Evil Twin attack where the RAP is configured with different encryption. This type of detection is also possible without using any profile, because of the no knowledge detectors.

Another possible heuristic is the Timing Synchronization Function (TSF) produced by RAPs. The TSF maintains the radios for all stations in the same Basic Service Set (BSS) synchronized. Each AP will have a different value that is related with the up time of the AP. In figure 2 it can be seen that the newly created RAP has a TSF different from the authorized AP. From our analysis, we also verified that some known RAPs tools have clearly different TSF values. For example, the Airbase-ng RAPs are created with extremely high values for this field, and `scapy` based RAPs use the zero value if not properly configured. These values can be used for detection purposes.

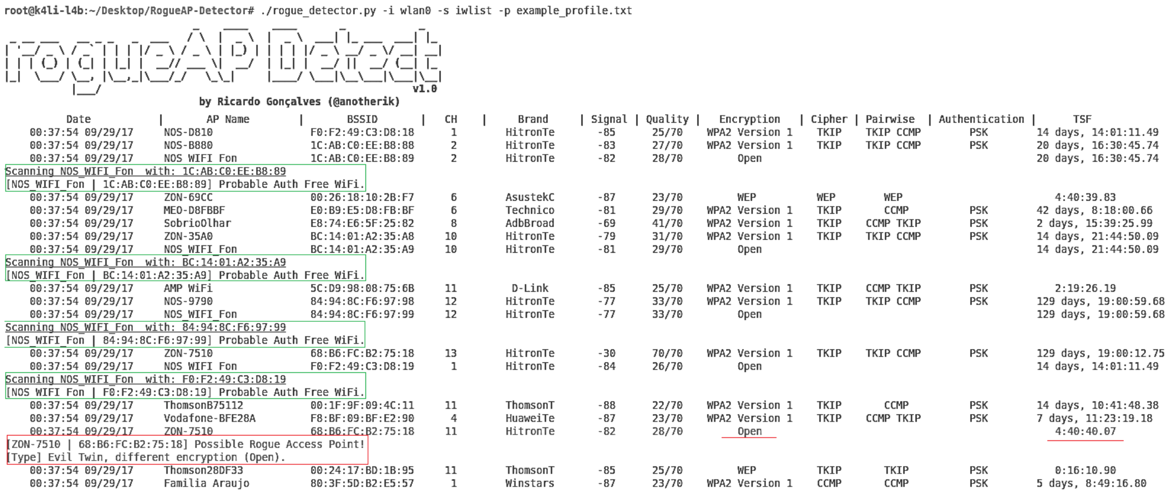
In summary, we performed a set of tests for two different scenarios, with the goal of creating a proof of concept for the developed tool.

5 CONCLUSIONS

In this work we studied the concepts behind RAPs. We described the types of RAPs and some countermeasures studied in the literature. We also discussed how these RAPs are created and explored. With this knowledge, the analysis of Wi-Fi packets, and APs specifications and parameters, we proposed a flexible framework that relies on modules and a set of heuristics to detect possible RAPs. The modular architecture eases the incorporation of new features, be it new scanning methods or other detection heuristics.

The framework addresses: both Evil Twin types of RAPs, coexistence and replacement, where it uses the Actuators' association procedure to identify them. Improperly configured APs, unauthorized and compromised APs are also identified with whitelist and blacklist heuristics. It is also possible to detect RAPs generated by attacking tools, like PineAP, Alfa cards and Airbase. A heuristic to validate authenticity of free Wi-Fis is also configured in our application.

In the future we aim to expand the framework into a distributed system where the main application would communicate with a set of nodes. This would cover a larger area, and ultimately, physically locate the RAPs. For the scanning process, we would like to upgrade `iwlist` to `iw`, the latter has more information about the Beacon frames that could be incorporated as new detection heuristics. In the detection part we plan on using the TSF parameter to discover newly created



```
root@k4li-l4b:~/Desktop/RogueAP-Detector# ./rogue_detector.py -i wlan0 -s iwlist -p example_profile.txt
                                                                                   v1.0
                                    by Ricardo Gonçalves (@anotherik)
        Date          |       AP Name        |       BSSID        |  CH   |    Brand     | Signal | Quality |   Encryption   | Cipher | Pairwise | Authentication |      TSF
     00:37:54 09/29/17    NOS-D810              F0:F2:49:C3:D8:18    1        HitronTe      -85     25/70    WPA2 Version 1    TKIP    TKIP CCMP       PSK         14 days, 14:01:11.49
     00:37:54 09/29/17    NOS-B880              1C:AB:C0:EE:B8:88    2        HitronTe      -83     27/70    WPA2 Version 1    TKIP    TKIP CCMP       PSK         20 days, 16:30:45.74
     00:37:54 09/29/17    NOS WIFI Fon          1C:AB:C0:EE:B8:89    2        HitronTe      -82     28/70         Open                                             20 days, 16:30:45.74
Scanning NOS_WIFI_Fon  with: 1C:AB:C0:EE:B8:89
[NOS_WIFI_Fon | 1C:AB:C0:EE:B8:89] Probable Auth Free WiFi.
     00:37:54 09/29/17    ZON-69CC              00:26:18:10:2B:F7    6        AsustekC      -87     23/70         WEP          WEP        WEP                          4:40:39.83
     00:37:54 09/29/17    MEO-D8FBBF            E0:B9:E5:D8:FB:BF    6        Technico      -81     29/70    WPA2 Version 1    TKIP       CCMP         PSK         42 days, 8:18:00.66
     00:37:54 09/29/17    SobrioOlhar           E8:74:E6:5F:25:82    8        AdbBroad      -69     41/70    WPA2 Version 1    TKIP    CCMP TKIP       PSK         2 days, 15:39:25.99
     00:37:54 09/29/17    ZON-35A0              BC:14:01:A2:35:A8    10       HitronTe      -79     31/70    WPA2 Version 1    TKIP    TKIP CCMP       PSK         14 days, 21:44:50.09
     00:37:54 09/29/17    NOS_WIFI_Fon          BC:14:01:A2:35:A9    10       HitronTe      -81     29/70         Open                                             14 days, 21:44:50.09
Scanning NOS_WIFI_Fon  with: BC:14:01:A2:35:A9
[NOS_WIFI_Fon | BC:14:01:A2:35:A9] Probable Auth Free WiFi.
     00:37:54 09/29/17    AMP WiFi              5C:D9:98:08:75:6B    11        D-Link       -85     25/70    WPA2 Version 1    TKIP    CCMP TKIP       PSK            2:19:26.19
     00:37:54 09/29/17    NOS-9790              84:94:8C:F6:97:98    12       HitronTe      -77     33/70    WPA2 Version 1    TKIP    TKIP CCMP       PSK         129 days, 19:00:59.68
     00:37:54 09/29/17    NOS_WIFI_Fon          84:94:8C:F6:97:99    12       HitronTe      -77     33/70         Open                                             129 days, 19:00:59.68
Scanning NOS_WIFI_Fon  with: 84:94:8C:F6:97:99
[NOS_WIFI_Fon | 84:94:8C:F6:97:99] Probable Auth Free WiFi.
     00:37:54 09/29/17    ZON-7510              68:B6:FC:B2:75:18    13       HitronTe      -30     70/70    WPA2 Version 1    TKIP    TKIP CCMP       PSK         129 days, 19:00:12.75
     00:37:54 09/29/17    NOS WIFI Fon          F0:F2:49:C3:D8:19    1        HitronTe      -84     26/70         Open                                             14 days, 14:01:11.49
Scanning NOS_WIFI_Fon  with: F0:F2:49:C3:D8:19
[NOS WIFI Fon | F0:F2:49:C3:D8:19] Probable Auth Free WiFi.
     00:37:54 09/29/17    ThomsonB75112         00:1F:9F:09:4C:11    11       ThomsonT      -88     22/70    WPA2 Version 1    TKIP       CCMP         PSK         14 days, 10:41:48.38
     00:37:54 09/29/17    Vodafone-BFE28A       F8:BF:09:BF:E2:90    4        HuaweiTe      -87     23/70    WPA2 Version 1    TKIP    CCMP TKIP       PSK         7 days, 11:23:19.18
     00:37:54 09/29/17    ZON-7510              68:B6:FC:B2:75:18    11       HitronTe      -82     28/70         Open                                                4:40:40.07
[ZON-7510 | 68:B6:FC:B2:75:18] Possible Rogue Access Point!
[Type] Evil Twin, different encryption (Open).
     00:37:54 09/29/17    Thomson28DF33         00:24:17:BD:1B:95    11       ThomsonT      -85     25/70         WEP          TKIP       TKIP         PSK            0:16:10.90
     00:37:54 09/29/17    Familia Araujo        80:3F:5D:B2:E5:57    1        Winstars      -87     23/70    WPA2 Version 1    CCMP       CCMP         PSK         5 days, 8:49:16.80
```

Figure 2: For controlled scenario – Validate free Wi-Fis authenticity & – RAP with same SSID and BSSID, different Encryption.

APs and combine it with the other heuristics in order to give a bigger accuracy in the detection rate.

ACKNOWLEDGEMENTS

This work is partially funded by the ERDF through the COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT as part of project UID/EEA/50014/2013.

REFERENCES

Alotaibi, B. and Elleithy, K. (2016). Rogue access point detection: Taxonomy, challenges, and future directions. *Wireless Personal Communications*, pages 1–30.

Bauer, K., Gonzales, H., and McCoy, D. (2008). Mitigating evil twin attacks in 802.11. In *2008 IEEE International Performance, Computing and Communications Conference*, pages 513–516. IEEE.

Branch, J. W., Petroni, N. L., Doorn, L. V., and Safford, D. (2004). Autonomic 802.11 wireless lan security auditing. *IEEE Security Privacy*, 2(3):56–65.

Bratus, S., Cornelius, C., Kotz, D., and Peebles, D. (2008). Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on Wireless network security*, pages 56–61. ACM.

Chirumamilla, M. K. and Ramamurthy, B. (2003). Agent based intrusion detection and response system for wireless lans. In *IEEE International Conference on Communications, ICC'03.*, volume 1, pages 492–496.

Dai Zovi, D. A. and Macaulay, S. A. (2005). Attacking automatic wireless network selection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 365–372. IEEE.

Han, H., Sheng, B., Tan, C. C., Li, Q., and Lu, S. (2011). A timing-based scheme for rogue ap detection. *IEEE Transactions on parallel and distributed Systems*, 22(11):1912–1925.

IEEE Wireless LAN Working Group (2016). IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2016*, pages 1–3534.

Ma, L., Teymorian, A. Y., and Cheng, X. (2008). A hybrid rogue access point protection framework for commodity wi-fi networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE.

Mustafa, H. and Xu, W. (2014). Cetad: Detecting evil twin access point attacks in wireless hotspots. In *Communications and Network Security (CNS), 2014 IEEE Conference on*, pages 238–246. IEEE.

Schmoyer, T. R., Lim, Y. X., and Owen, H. L. (2004). Wireless intrusion detection and response: a classic study using main-in-the-middle attack. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 2, pages 883–888. IEEE.

Sriram, V. S., Sahoo, G., and Agrawal, K. K. (2010). Detecting and eliminating rogue access points in ieee-802.11 wlan-a multi-agent sourcing methodology. In *Advance computing conference (IACC), 2010 IEEE 2nd international*, pages 256–260. IEEE.

Whelan, R., Van Wagenen, L., and Morris, R. (2011). System and method for detecting unauthorized wireless access points. US Patent 7,965,842.

Yan, B., Chen, G., Wang, J., and Yin, H. (2009). Robust detection of unauthorized wireless access points. *Mobile Networks and Applications*, 14(4):508–522.