An Automated Approach to Detect Deauthentication and Disassociation Dos Attacks on Wireless 802.11 Networks

Haitham Ameen Noman[1], Shahidan M. Abdullah[2] and Haydar Imad Mohammed [3]

[1] Department of Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

[2] Department of Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

**[3] Department of Computer and Communication Systems Engineering, University Putra Malaysia
Kuala Lumpur, Serdang, Malaysia**

Abstract

Wireless networks are unlike wired when it comes to security factor, they are considered fundamentally insecure due to its nature of transmitting the data via radio waves and also the security design of WLAN structure exposed the medium to versatile attacks. This paper sheds the light particularly on the availability factor, in which the attacker tends to exploit certain design flaws in wireless layer two (MAC Layer) to disrupt the connection on the authenticated clients. This can be performed by sending forged deauthentication or disassociation packets using "IJAM" a customizable tool written in Python. On the other hand this paper discusses the possible ways to detect these types of attacks and how important is it to implement an automated method to detect these attacks.

Keywords: *Deauthentication, Disassociation, Dos, WLAN*

1. Introduction

Wireless 802.11 standard [1] was founded to help people connecting more easily to networks and internet. However this medium suffers from several security concerns in terms of confidentiality, integrity and availability. In order to overcome those concerns encryption algorithms like WEP and WPA come across to mitigate both confidentiality and integrity by adding additional security layer to wireless medium. These two encryption algorithms would encrypt network traffic when implemented successfully [1] [2]. An interceptor will be incapable to connect to an encrypted network nor to read or change the exchanged data. Unfortunately WLAN developers have neglected the availability factor and left it exposed to different types of denial of service attacks. WLAN 802.11 frames consist of three major frames management, control and data frames [3] Data frame is whereas encryption applied. On the other side, both frames (Management and control) are responsible for power saving, association, deauthentication, disassociation, and authentication between the access point and the clients [4]. The absence of encryption implementation at both of management and control frames exposes the medium to persistent diverse types of DOS attacks at Data Link OSI Layer [3][4].The attacker might simply spoof the unencrypted Deauthentication/Disassociation message with the MAC address of particular access point and keep retransmitting it to all clients continuously causing a disconnection state in WLAN networks. Wireless Availability attacks can be classified according to each OSI layer with its risk level as depicted underneath in figure 1.This paper only focuses on MAC layer attack whereas the attacker is not conditionally connected to the network in order to launch an attack. The reason behind focusing on this layer is the shortage of concurrent solutions that mitigate Dos attacks on this particular level [5]. Different hard work and researches were performed to mitigate the attacks on application, transport and network layers yet unfortunately both layer two and one were neglected and left to be exploited by malicious hackers.[3]-[4]-[5] Deauthentication/Disassociation attacks are both parts from layer two, [5] however detecting these types of attacks requires a skillful network administrator therefore, the need for an automated monitoring tool raised to provide an easy way to alert the regular user if there is an aired Deauthentication/Disassociation attacks. The attacker tends to launch continuous flood of either deauthentication or disassociation for the sake of divulging hidden SSID of the targeted access point also to obtain a handshake to be used later on in cracking WPA2 encryption. [3] Another reasonable justification for the attack is to prevent the user to connect to the legitimate access point and to trap the clients to connect to rogue access point in order to steal, redirect or tamper client’s data while en route.

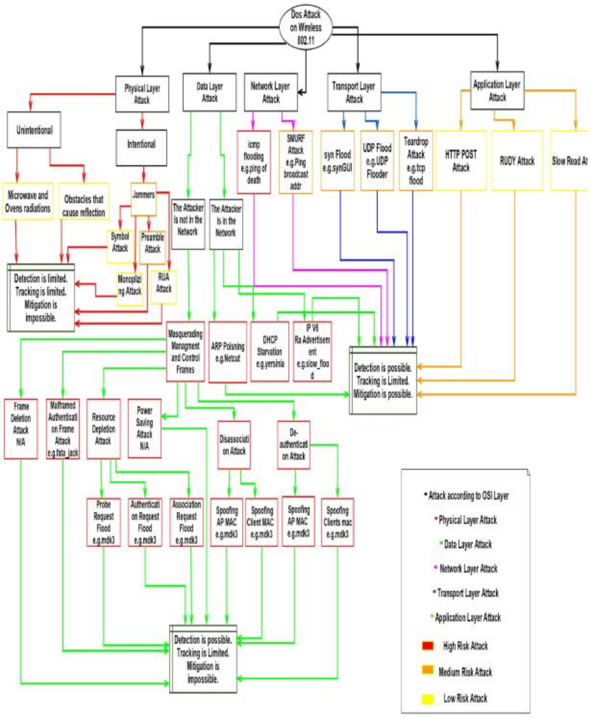

Dos Attack on Wireless 802.11
Physical Layer Attack
Data Layer Attack
Network Layer Attack
Transport Layer Attack
Application Layer Attack
Unintentional
Intentional
icmp flooding e.g.ping of death
SMURF Attack e.g.Ping broadcast addr
syn Flood e.g.synGUI
UDP Flood e.g.UDP Flooder
Teardrop Attack e.g.tcp flood
HTTP POST Attack
RUDY Attack
Slow Read At
Microwave and Ovens radiations
Obstacles that cause reflection
Jammers
The Attacker is not in the Network
The Attacker is in the Network
Symbol Attack
Preamble Attack
Monoplizing Attack
RUA Attack
Detection is limited. Tracking is limited. Mitigation is impossible.
Masquerading Managment and Control Frames
ARP Poisning e.g.Netcut
DHCP Starvation e.g.yersinia
IP V6 Ra Advertisement e.g.slow_flood
Detection is possible. Tracking is Limited. Mitigation is possible.
Frame Deletion Attack N/A
Malframed Authentication on Frame Attack e.g.fata_jack
Resource Depletion Attack
Power Saving Attack N/A
Disassociation Attack
De-authentication Attack
Probe Request Flood e.g.mdk3
Authentication Request Flood e.g.mdk3
Association Request Flood e.g.mdk3
Spoofing AP MAC e.g.mdk3
Spoofing Client MAC e.g.mdk3
Spoofing AP MAC e.g.mdk3
Spoofing Clients mac e.g.mdk3
Detection is possible. Tracking is Limited. Mitigation is impossible.
• Attack according to OSI Layer
• Physical Layer Attack
• Data Layer Attack
• Network Layer Attack
• Transport Layer Attack
• Application Layer Attack
🟥 High Risk Attack
🟧 Medium Risk Attack
🟨 Low Risk Attack

Fig. 1Wireless DOS Attack Types

2. Wireless MAC Layer Attack Types

Association/Authentication flood attack: In this state, the attacker spoofs source MAC addresses in an attempt to authenticate and associate to a particular access point. The attacker continually sends floods of either association or authentication requests, to fully consume the memory and processing capacity of targeted access point, leaving connected clients with either limited or no connectivity connection status [3]-[5].

Deauthentication/Disassociation flood attack: Wireless Network is susceptible to Denial of Service Attack “DOS Attack” by means the attacker can use a spoofed deauthentication command to force the access point to re authenticate the connected clients unfortunately this kind of attacks is considered unstoppable till this day in (a, b, g, n) standards. However, the new standards (ac) offers a partly protection against this attack only when encryption is implemented. [10] The following figures illustrate the Deauthentication attack mechanism.

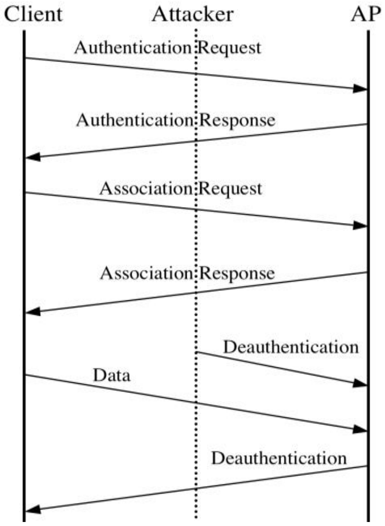
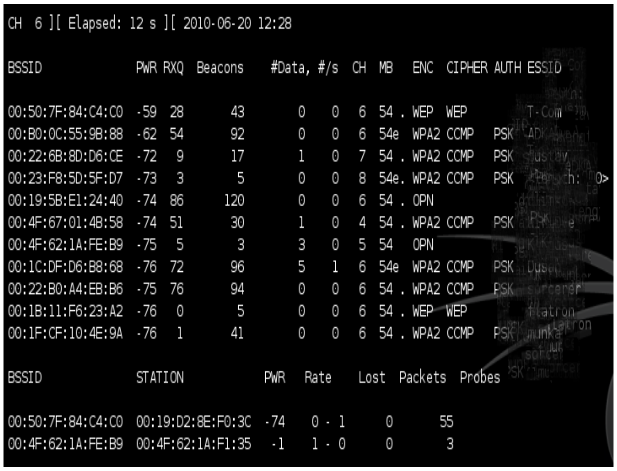


Fig. 2 Deauthentication Attack

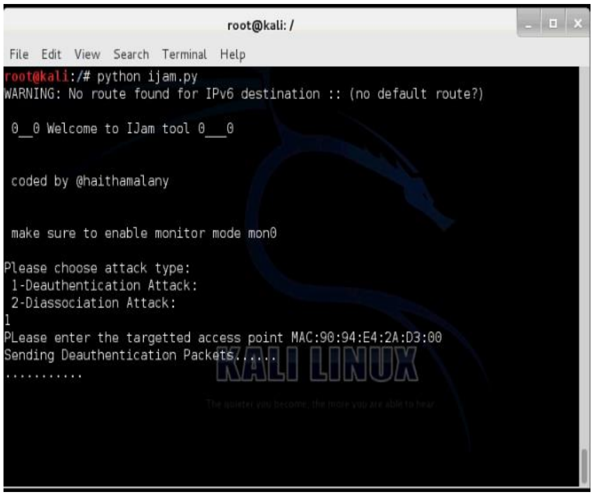
Aircrack-ng utility on Linux is used for scanning and cracking wireless networks encryption [11]. This utility contains a remarkable tool called aireplay-ng, it provides an option to spoof and send deauthentication packets to one or more associated clients with an explicit access point [5]. In order to find and scan for associated clients with a particular access point, Airodump-ng tool was found to achieve such purpose efficiently [6]-[11] as it can be depicted in the following figure:



```
CH  6 ][ Elapsed: 12 s ][ 2010-06-20 12:28

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:50:7F:84:C4:C0  -59  28       43        0    0   6  54 . WEP  WEP         T-Com
00:B0:0C:55:9B:88  -62  54       92        0    0   6  54e  WPA2 CCMP   PSK  ADK
00:22:6B:8D:D6:CE  -72   9       17        1    0   7  54 . WPA2 CCMP   PSK  Gustav
00:23:F8:5D:5F:D7  -73   3        5        0    0   8  54e. WPA2 CCMP   PSK  <length:  0>
00:19:5B:E1:24:40  -74  86      120        0    0   6  54 . OPN
00:4F:67:01:4B:58  -74  51       30        1    0   4  54 . WPA2 CCMP   PSK
00:4F:62:1A:FE:B9  -75   5        3        3    0   5  54   OPN
00:1C:DF:D6:B8:68  -76  72       96        5    1   6  54e  WPA2 CCMP   PSK  Dusan
00:22:B0:A4:EB:B6  -75  76       94        0    0   6  54 . WPA2 CCMP   PSK  sorcerer
00:1B:11:F6:23:A2  -76   0        5        0    0   6  54 . WEP  WEP         flatron
00:1F:CF:10:4E:9A  -76   1       41        0    0   6  54 . WPA2 CCMP   PSK  munka

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

00:50:7F:84:C4:C0  00:19:D2:8E:F0:3C  -74    0 - 1      0       55
00:4F:62:1A:FE:B9  00:4F:62:1A:F1:35   -1    1 - 0      0        3
```

Fig. 3 Airodump-ng Scanning Results

The following screenshots demonstrates a practical example for continuous deauthentication packets that is being sent to a particular wireless access point using Python written tool IJAM. [9]



```
root@kali:/# python ijam.py
WARNING: No route found for IPv6 destination :: (no default route?)

 0__0 Welcome to IJam tool 0___0


 coded by @haithamalany


 make sure to enable monitor mode mon0

Please choose attack type:
 1-Deauthentication Attack:
 2-Diassociation Attack:
1
PLease enter the targetted access point MAC:90:94:E4:2A:D3:00
Sending Deauthentication Packets......
...........
```

Fig. 4 Deauth packets are being sent in IJAM

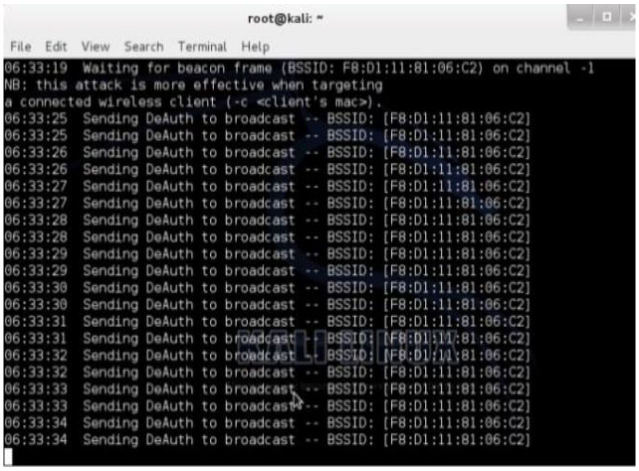
The reason behind building IJAM is to provide a Swiss-knife tool that brings the two attacks in one place since almost all available tools support solely Deauthentication option; however Disassociation attack is basically identical to the Deauthentication in terms of mechanism, although it is slightly less efficient [8]. Advanced users are likely tend to configure their access points to hide SSID to make sure that only authenticated and legitimate clients connect to the network, however carrying out deauthentication can impose the access point to divulge its SSID so that it becomes known to the attacker in order to connect to. A client may possibly be authenticated with various access points at the same time. Association permits a particular client to determine which access point will be used for communicating with the network. In order to terminate this relationship Deauthentication/Disassociation messages are used. Deauthentication/Disassociation values are both fixed and not encrypted hence they can be forged without difficulty.

3. Attack Analysis

Wireless Alfa adapter was used under Kali Linux in the following two methodologies as it provides packet injection feature. [12]

A- Wireshark Real-Time monitoring

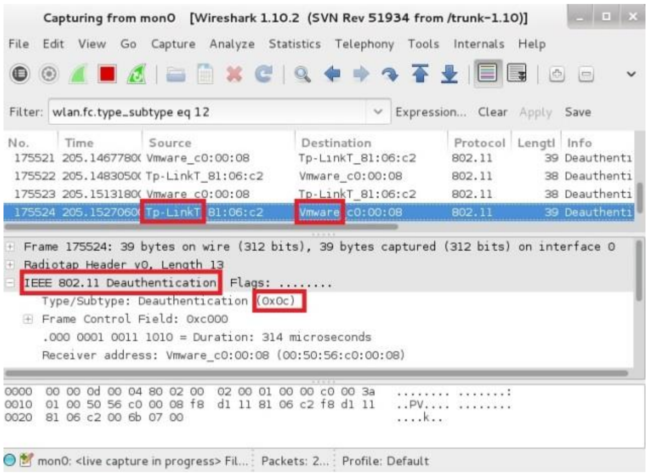
The attacker tends to send either Deauthentication or Disassociation packet to disrupt the connection on the clients by masquerade the access point MAC address using IJAM tool as it can be concluded in the following figure:



```
06:33:19  Waiting for beacon frame (BSSID: F8:D1:11:81:06:C2) on channel -1
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
06:33:25  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:25  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:26  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:26  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:27  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:27  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:28  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:28  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:29  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:29  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:30  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:30  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:31  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:31  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:32  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:32  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:33  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:33  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:34  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
06:33:34  Sending DeAuth to broadcast -- BSSID: [F8:D1:11:81:06:C2]
```

Fig. 5 Deauthentication Packet

What can be depicted from figure 5 is the attacker sends numerous periodic deauthentication packets to all connected clients to a particular access point with the shown MAC address.On the meantime the clients will instantly got deauthenticated which gives the attacker the ability to create rogue access point in order to trap the clients to connect to it. While performing the attack, Wireshark network monitoring tool was used to analyze the real time aired packets, as it illustrated in Figure 6


Capturing from mon0 [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
mon0: <live capture in progress> Fil... Packets: 2... Profile: Default

Filter: wlan.fc.type_subtype eq 12

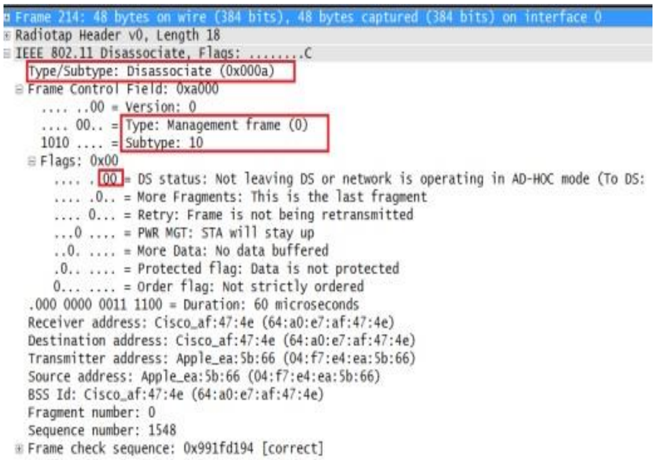
```
No.    Time           Source              Destination         Protocol Lengtl Info
175521 205.1467780( Vmware_c0:00:08     Tp-LinkT_81:06:c2   802.11       39 Deauthenti
175522 205.1483050( Tp-LinkT_81:06:c2   Vmware_c0:00:08     802.11       38 Deauthenti
175523 205.1513180( Vmware_c0:00:08     Tp-LinkT_81:06:c2   802.11       38 Deauthenti
175524 205.1527060( Tp-LinkT_81:06:c2   Vmware_c0:00:08     802.11       39 Deauthenti

⊞ Frame 175524: 39 bytes on wire (312 bits), 39 bytes captured (312 bits) on interface 0
⊞ Radiotap Header v0, Length 13
⊟ IEEE 802.11 Deauthentication  Flags: ........
    Type/Subtype: Deauthentication (0x0c)
  ⊞ Frame Control Field: 0xc000
    .000 0001 0011 1010 = Duration: 314 microseconds
    Receiver address: Vmware_c0:00:08 (00:50:56:c0:00:08)

0000  00 00 0d 00 04 80 02 00  02 00 01 00 00 c0 00 3a   ........ .......:
0010  01 00 50 56 c0 00 08 f8  d1 11 81 06 c2 f8 d1 11   ..PV.... ........
0020  81 06 c2 00 6b 07 00                               ....k..
```

Fig. 6 Analyzing Deauthentication Attack in Wireshark

The resulting info can be concluded from the attack:

- The management frame type that was forged and sent by the attacker is Deauthentication represented by two bits (00).
- Deauthentication has a fixed subtype value represented by four bits (1100)
- The attacker is using Vmware (Virtual Machine runs under Kali Linux)
- The targeted access point type is "TP-Link".
- The attacker is sending Broadcast message to deauthenticate all clients.
- The physical address of the attacker (MAC Address) which is identical to the access point.

Note: The attacker might reverse the attacking scenario by spoofing connect client's MAC address to send a continuous floods of deauthentication requests as an alternative way of masquerading the MAC address of the access point [9]. The following figure depicts the analysis of disassociation Dos attack scenario.



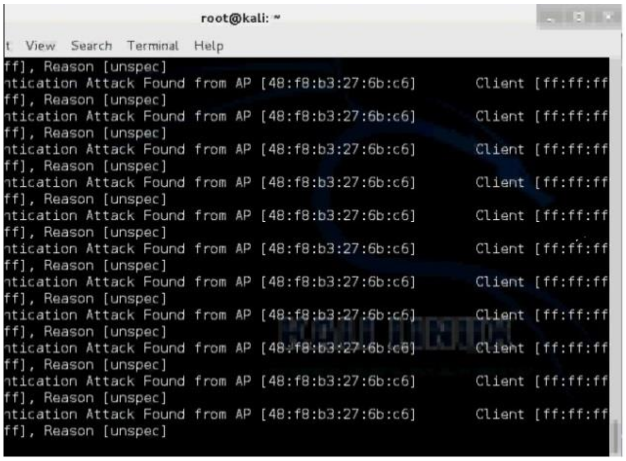
```
⊞ Frame 214: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface 0
⊞ Radiotap Header v0, Length 18
⊟ IEEE 802.11 Disassociate, Flags: .......C
    Type/Subtype: Disassociate (0x000a)
  ⊟ Frame Control Field: 0xa000
      .... ..00 = Version: 0
      .... 00.. = Type: Management frame (0)
      1010 .... = Subtype: 10
    ⊟ Flags: 0x00
        .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS:
        .... .0.. = More Fragments: This is the last fragment
        .... 0... = Retry: Frame is not being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = Protected flag: Data is not protected
        0... .... = Order flag: Not strictly ordered
    .000 0000 0011 1100 = Duration: 60 microseconds
    Receiver address: Cisco_af:47:4e (64:a0:e7:af:47:4e)
    Destination address: Cisco_af:47:4e (64:a0:e7:af:47:4e)
    Transmitter address: Apple_ea:5b:66 (04:f7:e4:ea:5b:66)
    Source address: Apple_ea:5b:66 (04:f7:e4:ea:5b:66)
    BSS Id: Cisco_af:47:4e (64:a0:e7:af:47:4e)
    Fragment number: 0
    Sequence number: 1548
  ⊞ Frame check sequence: 0x991fd194 [correct]
```

Fig. 7 Wireshark Monitoring Disassociation Attack

What can be noticed from above scenarios, the values of deauthentication and disassociation values are both fixed and sent in plain form (unencrypted form) so that they can be intercepted and replayed effortlessly.

B- Automated Attack Detection Method

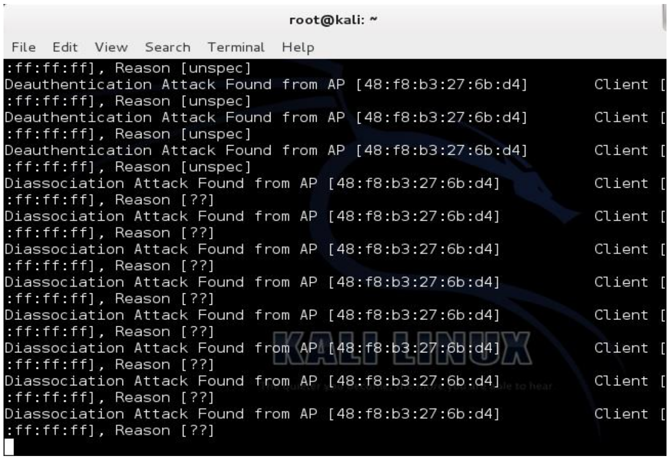
As Wireshark and most network monitoring tools require sort of deep knowledge and experience to analyze packets, the need for automated detection method raised.
A usable portable based tool can fulfill that need also written in Python and relying on the privileges that being offered by Linux operating system. This tool sets Wireless Interface (Wlan0) into Monitor Mode (Mon0) to monitor both Deauthentication (Deauth) and Disassociation (Diass) attacks on any ranged targeted access point alongside both victim and attacker MAC addresses. It requires no skill to function seamlessly.



```
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
ntication Attack Found from AP [48:f8:b3:27:6b:c6]       Client [ff:ff:ff
ff], Reason [unspec]
```

Fig. 8 Deauthentication attack is being auto detected.

What can be concluded from detection scenario is there is a real time deauthentication attack that is being carried out against an access point with MAC address “48:f8:b3:27:6b:c6” to every connected client. The reason behind this attack is as shown above in figure 8 “Unspecific” which means by far it is intentioned. The tool can also detect mixed attacks (Both Deauthentication and Disassociation Dos attacks) when they took place together as it can be noticed in the following figure:



```
:ff:ff:ff], Reason [unspec]
Deauthentication Attack Found from AP [48:f8:b3:27:6b:d4]       Client [
:ff:ff:ff], Reason [unspec]
Deauthentication Attack Found from AP [48:f8:b3:27:6b:d4]       Client [
:ff:ff:ff], Reason [unspec]
Deauthentication Attack Found from AP [48:f8:b3:27:6b:d4]       Client [
:ff:ff:ff], Reason [unspec]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
Diassociation Attack Found from AP [48:f8:b3:27:6b:d4]          Client [
:ff:ff:ff], Reason [??]
```

Fig. 9 Mixed Attacks Detection

The following model illustrates the mechanism of the tool

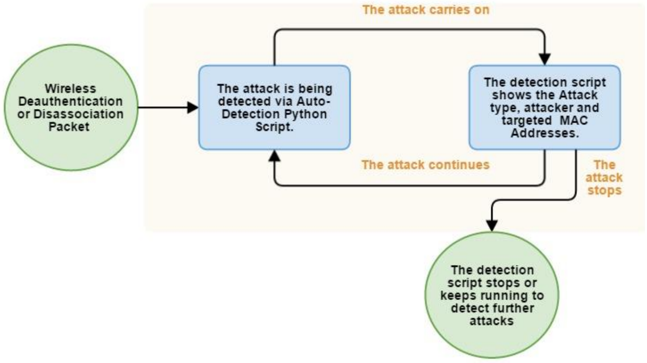

The attack carries on
Wireless Deauthentication or Disassociation Packet
The attack is being detected via Auto-Detection Python Script.
The detection script shows the Attack type, attacker and targeted MAC Addresses.
The attack continues
The attack stops
The detection script stops or keeps running to detect further attacks

Fig. 10 Automated Detection Model

There are certain circumstances where both Deauthentication and Disassociation attacks do not work efficiently for one of the underneath reasons:

1- The attacker needs enough transmit power for the packets to reach and be heard by the clients so that the targeted access point must be not far.

2- Some clients tend to ignore broadcast deauthentication in such case, the attacker needs to send a deauthentication directed at the selected client.

The targeted access point is implementing encryption and running on AC standards.

4. Conclusions

Based on the practical experiments that were held in small homed lab, the need for automated detection method raised alongside a patch to mitigate both deauthentication and disassociation attacks.

The attack might cause serious damage if performed against ad-hoc sensitive solely Wi-Fi based devices like health care lab machines or even Apple Mac Air notebook. Current standards needs urgent patches as the new standards (AC) will need long time to be deployed not to mention its incapability to provide protection to open Wi-Fi networks. Automated detection mechanism was implemented successfully and proved its efficiency on Linux operating system. At meantime the detector tool doesn’t run on Windows due to the limitation of windows operating system in enabling monitor mode.

Acknowledgments

This work was supported by Universiti Teknologi Malaysia (UTM)

References

[1] IEEE 802.11 Wireless Local Area Network Task Group I, TGI - MAC Enhancements for Enhanced Security

[2] Jie Yang,Yingying (Jennifer) Chen and Wade Trappe (2013) 'Detection and Localization of Multiple Spoofing Attackers in Wireless Networks', IEEE Transection on Parallel and Distributed System, 24(1), pp.

[3] David Cossa (n.d.) 'The Dangers of Deauthentication Attacks in an Increasingly Wireless World', Iowa State University, 537(), pp. [Online]. Available at:http://home.eng.iastate.edu/~gamari/CprE537_S13/project%20reports/deauthentication.pdf(Accessed: 13 August 2014).

[4] Stuart Compton and Charles Hornat (2010)'802.11 Denial of Service Attacks and Mitigation', SANS Institute.

[5] Motorola White Paper (2011), Can Wireless LAN Denial of Service Attacks Be Prevented? Understanding WLAN DoS Vulnerabilities & Practical Countermeasurs.

[6] T.Moore, Validating 802.11 Diassociation and Deauthentication Message, (2002), IEEE TGI.

[7] T. Moore, (2002), Validating 802.11 disassociation and deauthentication messages, IEEE TGi, Pp 802.11

[8] J. Bellardo and S. Savage, (2003), 802.11 Denial of service attacks real vulnerability and practical solutions, proceeding of the 12th USENIX Security Symposium, pp. 15-28.

[9] B. Aslam M. Islam and S.Khan (2006), 802.11 Disassociation Dos Attack and Its Solutions. A Survey in proceeding of the First Mobile Computing and Wireless Communication international conference ,pp. 221-226

[10] Next-Gen 802.11 ac Wi-Fi for Dummies: http://www.intel.com/content/dam/www/public/us/en/documents/pdf/next-gen-80211ac-wifi-for-dummies.pdf

[11] Aircrack-ng official website http://www.aircrack-ng.org/

Haitham Ameen Noman Holding bachelor degree from Al-Ahliyya Amman University in 2009 , earned Masters degree in Network and computer security from New York Institute of technology NYIT in 2012, currently studying PhD in computer security. Worked for two years as a software developer at Jordan in Optimiza Company. Published one paper about Yahoo Messenger Vulnerability in International Journal of Scientific and Engineering Research.

Shahidan M. Abdullah Associate Prof, doctor at university of teknologi of Malaysia.

Haydar Imad Mohammed studying master’s degree in communication engineering