Man-In-the-Middle, De-authentication and Rogue AP Attacks in 802.11 networks

Man-In-the-Middle, De-authentication and Rogue AP Attacks in 802.11 Networks

Purpose

We would like to share the log files (pcap file, collected with airodump-ng) of experiments conducted within the Networks group in Loughborough University. The experiments involve a wireless client associated to an AP and an attacker launching injection attacks (see more details below). The purpose of the experiments is to test our multi-layer fusion ideas for detection of injection type of attacks. We decided to share the data with the research community in an effort to help others test whether their ideas/algorithms can detect the malicious frames and as a platform for comparing results. Three main types of attacks have been launched using publicly available tools:

1) Man-In-The-Middle at the Physical Layer attack - using Airpwn tool
2) De-authentication Attack Using Aircrack-ng
3) Rogue Access Point (AP) attack

Description of Attacks

1. MitM @ PHY Attack

To implement the MitM at PHY attack, we used the Airpwn tool. Airpwn takes advantage of the duration of time that a server requires to respond to web-page requests. In that lag time, it can inject its own content onto the wireless channel of an access point. For example, a client may request a page from wikipedia.org that takes, approximately 13 ms round-trip time. If an attacker near the victim is running the Airpwn tool, it will see the legal client's request and immediately respond with its own HTML code.

Since there are no hops between the attacker and the victim, it takes the attacker much less time to respond. When the client receives the data, it will assume the original request was answered and process the injected code. Even though the attack is launched at the application layer by injecting an HTTP packet, the actual attack is practical only because there are no mechanisms in WiFi 802.11 to prevent a misbehaving node from injecting their own malicious code in the form of valid 802.11 frames.

The experimental testbed is composed by an authorised AP that provides internet connection to a client and an attacker running the Airpwn attacking tool. A third party monitoring system (mon) is utilised to capture all relevant information (not shown in Fig. 1). However, the same monitoring activity can also take place in the client machine.

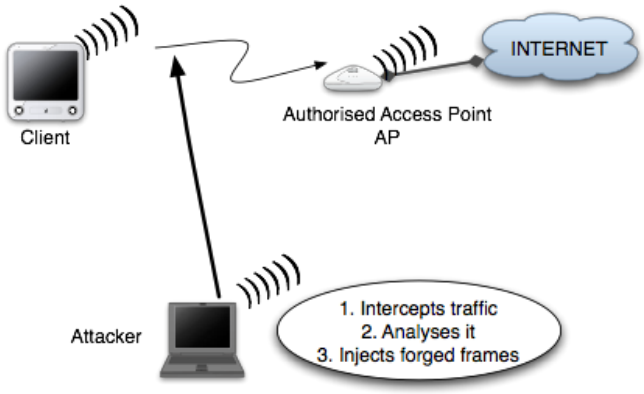

Figure 1

In our experiments (airpwn_pcap_files.zip), two types of attacks were launched against the client. Both attack codes were default options in the Airpwn suite. We refer to these attacks as Attack 01 and Attack 02. A mixture of both attacks is found in the “inthemix” pcap file. The normal pcap file does not include any malicious activity.

In the first type of attack, the attacker eavesdrops the HTTP request frame from a client destined to a web server and then proceeds by injecting a forged frame. In this type of attack the forged frame contains HTML code that replaces the title of the authentic web page to a custom one as seen in the figure left (notice the “Hello Defcon ...” message in Fig. 2).

Figure 2: Attack 01

In the second type of attack, the attacker listens for requests for images hosted on the web site and injects its own images (see the AIRPWNED image in Fig. 3). In addition, the attacker injects TCP reset frames so the client proceeds requesting the remaining objects of the web site.

Figure 3: Attack 02

Note: The attacker is spoofing the MAC address to make it look like the injected frames come from the actual AP. The malicious frames are the TCP frames with TTL value 255. This is the default value that Airpwn is using and was not changed in order to help us identify which frame is actually malicious. Obviously, while evaluating our algorithm, this value was changed to a more realistic value, and more specifically equal to 50. The TTL value of 50 was chosen because it is an approximate of the difference between the default TTL value for Linux web servers (64) and the number of hops from the victims' location to some of the most popular destinations (google, bbc), which from our location is around 14.

The following display filter has been used in Wireshark to isolate the traffic flow from the AP to the victim client (and broadcasts) during the MitM attack experiments:

(wlan.sa == 00:25:9c:cf:8a:73 || wlan.sa == 00:25:9c:cf:8a:71)&& (wlan.da == 40:c3:36:07:d4:bf || wlan.da == ff:ff:ff:ff:ff:ff)&& !(wlan.fc.type==0)&&tcp

2. De-authentication Attack

Similarly, to the above testbed and by using the Aircrack tool, we launched a de-authentication attack of the wireless client in a WPA2 encrypted network (see deauth pcap file). This attack is commonly utilised in DoS attacks.

Note: In this scenario, the attacker is again spoofing its MAC address to fool the victim that the frames are coming from the authentic AP. However, the metric NAV helped us identify which frames are maliciously injected by the attacker. The malicious frames are the ones of de-authentication type and having a NAV value of 314. We had to use a different NAV value for evaluating the Detection Rate of our algorithm. In the published papers we argue that just by using this metric to distinguish is not enough as the attacker could easily find hardware that injects frames with the same NAV value and therefore, using further metrics in a collaborative manner is required.

The following display filter has been used in Wireshark to isolate the traffic flow from the AP to the victim client (and broadcasts) during the De-authentication attack experiments:

(wlan.sa == 00:25:9c:cf:8a:73 || wlan.sa == 00:25:9c:cf:8a:71)&&
(wlan.da == 40:c3:36:07:d4:bf || wlan.da == ff:ff:ff:ff:ff:ff)&&
wlan.fc.type==0

3. Rogue AP Attack

In this scenario, we are using the HostAPd tool (http://hostap.epitest.fi/hostapd/) to launch a Rogue AP (RAP) attack. The Rogue AP advertises itself as a real AP and tries to entice the wireless clients to associate with it. The attack starts at frame 9028 of the pcap file, when attacker sends a Beacon frame with SSID=linkstys2.

The legal AP SSID is linksys

The legal AP MAC address is: 00:25:9c:cf:8a:73

The Rogue AP SSID is linksys2

The Rogue AP MAC address is: 00:1b:b1:05:44:f2

The client/victim MAC address is: 40:c3:36:07:d4:bf

The following display filter has been used in Wireshark to isolate the traffic flow from the AP to the victim client：

```
wlan.da==40:c3:36:07:d4:bf    ||    wlan.da==ff:ff:ff:ff:ff:ff)&&
(!wlan.fc.type==0)&&tcp
```

Acknowledgments

The measurement data (pcap files) were collected using EPSRC support. Anyone who uses the data agrees to:

1. Acknowledge HSN in Wolfson School of Mechanical, Electrical and Manufacturing Engineering at Loughborough University and EPSRC in any work they do.
2. Agree that the data will not be used for any illegal purpose, or passed on to others.

Related Publications

- F. J. Aparicio-Navarro, K. G. Kyriakopoulos, D. J. Parish, "A Multi-Layer Data Fusion System for Wi-Fi Attack Detection Using Automatic Belief Assignment," in *Proc. of the World Congress on Internet Security (WorldCIS)*, 2012, pp. 45-50.

- F. J. Aparicio-Navarro, K. G. Kyriakopoulos, D. J. Parish, "An Automatic and Self-Adaptive Multi-Layer Data Fusion System for WiFi Attack Detection," in *International Journal of Internet Technology and Secured Transactions*, vol. 5, no. 1, 2013, pp. 42-62.

- K. G. Kyriakopoulos, F. J. Aparicio-Navarro, D. J. Parish, "Manual and Automatic Assigned Thresholds in Multi-Layer Data Fusion Intrusion Detection System for 802.11 Attacks," in *IET Information Security*, vol. 8, no. 1, 2014, pp. 42-50.

- F. J. Aparicio-Navarro, K. G. Kyriakopoulos, D. J. Parish, "Automatic Dataset Labelling and Feature Selection for Intrusion Detection Systems," in *Proc. of the IEEE Military Communications Conference (MILCOM)*, 2014, pp. 46-51.

- F. J. Aparicio-Navarro, K. G. Kyriakopoulos, D. J. Parish, "Empirical Study of Automatic Dataset Labelling," in *Proc. of the 9th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2014, pp. 372-378.