

Forum:[Hack Forums](#)

Thread: [Advanced attacks on WiFi infrastructure and its clients - Tutorial - part 2](#)

2xTheTap115 - April 28, 2020, 7:28 am

- AUTHENTICATION DoS MODE

This attack involves connecting as many clients as possible to the attacked AP by sending authentication frames, which results in the suspension of the given access point. From an AP perspective, the attack causes the access point to maintain a connection with the constantly growing number of clients, which after a while results in its freezing. This is one of many attacks that can effectively disable or block the proper functioning of an access point. To perform such an attack will be used on the command:

Code `mdk3 INTERFACE a -c -a NETWORK BSSID`

In the above command, the "c" option is used to not check whether a given attack succeeded and the "a" option is used to select the type of attack (in our case, authentication dos mode). The "-a" option is used to specify the target of an attack, if the target is not specified then all APs in range will be attacked.

- DEAUTHENTICATION MODE

This is the most popular of attacks not only because of its effectiveness but also because there is no defense technique against this attack. The attack itself gives you full control over the transmission directed to a given AP and allows the attacker to "squeeze" from the network of any device. This attack involves sending deauthentication frames to the network under attack. The attack will find its application not only during the handshake capture process but also when we will block access to the network for specific users. The attack itself can be carried out using aireplay-ng but the mdk3 program will also do great. We will use the command:

Code `mdk3 INTERFACE d -b /FREIGHT/TO/FILE.TXT -c CHANNEL`

Similarly to the previous examples, the "d" parameter is used to select the type of attack, the "-b" option is used to indicate the black list of MAC addresses to be attacked and the "c" option will be used to specify the channel on which the program will "broadcast".

It is true that these are not all techniques used to jam the wifi network. I intentionally skipped repetitive attacks like 802.1X and WIDS/WIPS confusion using mdk3. I would like the reader to make some effort and look for information on the subject on the internet. I'm sure you will find there many interesting things on this subject.

2. EVIL TWIN TYPE ATTACK

One of the most effective and dangerous attacks on a WLAN infrastructure is a malicious twin attack. The whole idea of this attack is to install an attacker-controlled AP within the range of the attacked WLAN network, which will broadcast a network with the same ESSID and BSSID. In such a situation, many wireless clients can connect in good faith to such a "malicious twin" with the full conviction that he is the "original access point". In addition, if we combine such an attack with DoS on the original AP, we can further increase efficiency. After establishing the connection with the client of the attacked network, the attacker can launch a Man-in-the-Middle attack and in a completely transparent way for the client to transfer traffic back to the attacked network while

reading all communications. Of course, MiTM attacks are just one of many possible attack vectors that an attacker can implement. In a real scenario, such an attack is very difficult to detect and neutralize and well implemented allows the attacker to capture an incredible amount of data from the clients of the original AP and prepare the ground for other attacks. It is worth adding here that the effectiveness of such an attack depends on the smaller distance between the attacker's AP and the network client and the original access point. If the above conditions are met (or the signal of a bad twin is stronger) the attack will succeed and the client will connect to the crafted network. To perform this task we will need information about the network we will be impersonating (BSSID and ESSID of the attacked AP) and the aircrack-ng package. So let's do the command:

Code airodump-ng INTERFACE

At this stage, we wait for the client to connect to the original AP. Now we will create our evil twin, we will use the command:

Code airbase-ng -a BSSID -e "ESSID" -c CHANNEL INTERFACE

It seems to me that the syntax of the command is obvious and does not need to be explained. To check if our fake access point has appeared on the list of available networks, issue the command:

Code airodump-ng -c INTERFACE CHANNEL

We make sure that the client is still present in the network and proceed to the next phase of the attack which is DoS. In fact, we can simply throw the client out of the network forcing him to reconnect. However, our goal is to block the connection to the original AP. For this reason, we will send a continuous transmission of deauthentication packets in case we are not close enough to force "automatic connection" (at the beginning I send about 5 packets and then wait a few seconds to check if the client connects automatically only after that I send more packets) . This should effectively prevent connection. Of course, we can apply any other attack here, everything depends on you, but I will use the deauthentication technique in this example. We give the command:

Code aireplay-ng --deauth 0 -a BSSID_ORIGINAL_AP INTERFACE

3. UNAUTHORIZED ACCESS POINT

This type of attack is based on installing an access point controlled by the attacker directly in the authorized network. Most often it is used as a back entrance and helps to avoid security such as IDS systems or other mechanisms controlling network behavior on its external perimeter. In addition, it is worth noting that for successful attack it is necessary to access the network, and thus earlier security breach of the network. The attack itself can be carried out in two ways. The first is the physical installation of an access point in an authorized network. The implementation of such an attack will require physical access to the network infrastructure and in many situations such an attack vector will not be chosen by the attacker. The second most commonly used is to create a software access point connected by a "bridge" (bridged connection) to the authorized network. This solution allows you to turn any laptop (and even a rooted android phone equipped with a special adapter) into an unauthorized access point connecting it to the local network. Of course, there are solutions such as Wi-Fi Pineapple or even LAN Turtle (although this is based on GSM transmission for data transmission) thanks to which we will carry out an attack with unauthorized access but in this guide we will use the software version. We'll start by creating an AP using airbase-ng:

Code airbase-ng -e ESSID -c CHANNEL INTERFACE

The next step will be to create a bridged connection to the authorized network. Let's create a network interface that will act as a bridge using the brctl tool. So let's give the command:

Code brctl addbr INTERFACE NAME

In this step, we will add to the bridge the interface of the authorized network as well as the interface created by the command airbase-ng (the interface on which our unauthorized access point works).

We issue the following commands:

```
Code brctl addif NAME AUTHORIZED_ INTERFACE_ AUTHORIZED INTERFACE
Code brctl addif AUTHORIZED_ INTERFACE NAME AUTHORIZED_ INTERFACE_AP
We will then activate the individual interfaces using ifconfig:
```

```
Code ifconfig INTERFACE 0.0.0.0 upCode ifconfig INTERFACE 0.0.0.0 up
The final step will be to enable packet forwarding in our system to allow packet communication for free:
```

```
Code echo 1 > /proc/sys/net/ipv4/ip_forward
From now on, every user who connects to our access point will have half access to the network resources of the authorized network via the bridge we have created. And for us it means that we have full access to data sent in this network.
```

Panagiota - May 12, 2020, 10:07 am

Hi man, thanks for this thread. I was not aware of the "authentication DoS mode", thanks for letting me know. Usually I use the de-authentication mode but I had some inconsistencies in my private tests by using mdk3, clients were not always getting disconnected. However, if I use aireplay-ng and specify their mac address they always get disconnected.

2xTheTap115 - May 12, 2020, 10:49 am

(May 12, 2020 10:07 AM)Wolfsage Wrote: [\[CITE\]](#) Hi man, thanks for this thread. I was not aware of the "authentication DoS mode", thanks for letting me know. Usually I use the de-authentication mode but I had some inconsistencies in my private tests by using mdk3, clients were not always getting disconnected. However, if I use aireplay-ng and specify their mac address they always get disconnected.

You're welcome buddy?? I'll tell you that I also encountered this problem just when I did mdk3, I'm not quite sure what it is caused by but it seems to me that it may be a matter of the distance between the network card and AP? But as you say, aircrack-ng is always reliable??

P.S. And out of curiosity, what card are you using?

Panagiota - May 16, 2020, 10:16 am

(May 12, 2020 10:49 AM)2xTheTap115 Wrote: [\[CITE\]](#) P.S. And out of curiosity, what card are you using?

I use several, ALFA AWUS036H, AWUS036NHA and a few others...

2xTheTap115 - May 17, 2020, 12:14 am

(May 16, 2020 10:16 AM)Wolfsage Wrote: [\[CITE\]](#) I use several, ALFA AWUS036H, AWUS036NHA and a few others...

I also use AWUS036H. In my opinion, these are the most reliable ones on the market?