 SSID Flooding with MDK3

One neat trick that MDK3 can do is SSID flooding, or beacon flooding. What this means is that MDK3 can broadcast hundreds or even thousands of fake access points. Others that are in the area will see all of these fake access points when they go to search for WiFi access points to connect to. As you can probably see, SSID flooding is not denial of service. However, this is still a pretty cool trick. Potentially, you could set up a dedicated computer with a wireless access point and have MDK3 running in SSID flooding mode at all times. You could, in effect, hide your legitimate wireless access point in a sea of fake access points. A sort of security through obscurity to prevent WiFi hacking attacks.

Here is the syntax to enable simple SSID flooding (MDK3 will generate random fake access point names:

***mdk3 <interface> b -c 1***

Just replace <interface> with the name of your wireless interface. Remember, usually it’s monO.

The b option tells MDK3 to use beacon/SSID flooding mode.

-c1  tells MDK3 to broadcast all the fake access points on channel 1. (To better hide the fact these are all fake access points, you can try running multiple instances of MDK3 and specify a different channel each time.

You can also specify a list of specific SSID names from a file by appending the command above with:

-f <file name>

Let’s say your business’s wireless AP broadcasts as “ACME Business”. You want to use MDK3’s SSID flooding mode to hide your access point amongst a bunch of similarly named but decoy access points. You could create a text file named “SSIDs” and fill with fake access point names, perhaps names like “ACME WiFi” “ACME Network” “WiFi ACME”. Then, to bring this all together, you can simply run:

*mdk3 <interface> b -c 1 -f SSIDs*

There are tons of possible options for the SSID flooding mode:

b – Beacon Flood Mode

This spoofs tons of SSIDs. Remember, security through obscurity

OPTIONS:

-n <ssid>

Use a specific SSID <ssid> instead of randomly generated ones

-f <filename>

Read SSIDs from a file

-v <filename>

Read MAC addresses and SSIDs from a file.

-d

Display Ad-Hoc APs

-w

Set WEP bit (Generates encrypted networks)

-g

Display APs as 54 Mbit

-t

Display APs using WPA TKIP encryption

-a

Display APs using WPA AES encryption

-m

Use valid accesspoint MAC from OUI database

-h

Hop to the channel where the AP is spoofed

-c <chan>

Fake an AP on a channel <chan>.

-s <pps>

Set the DoS speed in packets per second (the default: 50)

Authentication Flooding with MDK3

Moving on to MDK3’s actual DOS options, you will first look at authentication flooding, then conclude with deauthentication flooding. The idea behind authentication flooding is simple. Too many authentication requests at one time may cause the wireless access point to freeze up and perhaps stop working entirely (until someone reboots the thing, that is).

I will warn you that in my experience, authentication flooding doesn’t always work. Most wireless access points are robust enough to handle an authentication flood from one instance of MDK3. (However, if you had multiple laptops running authentication floods this may work.)

Deauthentication flooding works MUCH better (that’s why I am saving it for last) and it doesn’t require the resources that authentication flooding does.  So let’s look at authentication flooding. A simple command to do authentication flooding is:

*mdk3 <interface> a -a <ap\_mac address>*

All you need is the AP’s MAC address as you can see above.

The list of all possible options are below:

*a – Authentication DoS mode*

*Send authentication frames to all APs found in range. By flooding the target AP with authentication requests, we can try to knock it offline.*

*OPTIONS:*

*-a <ap\_mac address>*

*You need the access point’s MAC address, which can be obtained with airodump*

*-m*

*Use a valid client MAC from OUI database*

*-c*

*Don’t check that the test was successful, just run the attack*

*-i <ap\_mac>*

*Performs an intelligent test on the access point (-a and -c will be ignored). This test connects clients to the AP and reinjects sniffed data to keep them alive*

*-s <pps>*

*Sets the speed in packets per second (Default: unlimited)*

Deauthentication Flooding with MDK3

The DoS WiFi hacking technique that works best uses deauthenticate requests rather than faking authentication requests.

*mdk3 <interface> d -b blacklist\_file*

Again, the only thing you need is the target access point’s MAC address. Save that MAC address in a text file and specify it after the -b option. This will sent deauth packets to any and all clients connected to the access point specified in the file. (You can add more MAC addresses to deauth if you are evaluating multiple APs in range.

*d – Deauthentication / Disassociation Amok Mode*

*Kicks everybody found from AP*

*OPTIONS:*

*-w <filename>*

*Read file containing MAC addresses to ignore (Whitelist mode)*

*-b <filename>*

*Read from a file containing MAC addresses to attack (Blacklist Mode)*

*-s <pps>*

*Set the speed in packets per second (Default: unlimited)*

*-c [chan,chan,chan,...]*

*Enables channel hopping. Without providing any channels, mdk3 will hop all channels until it finds the target you specified*