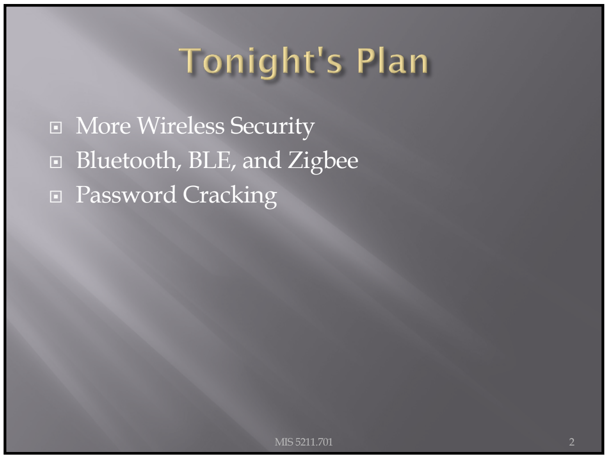
INTRO TO ETHICAL HACKING

MIS 5211.701

Week 12

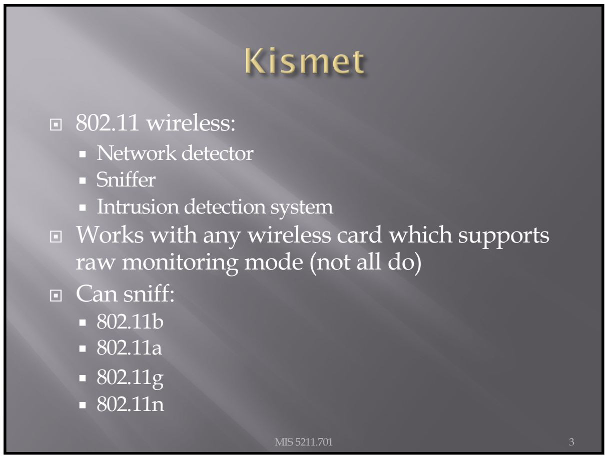
1



Tonight's Plan

- More Wireless Security
- Bluetooth, BLE, and Zigbee
- Password Cracking

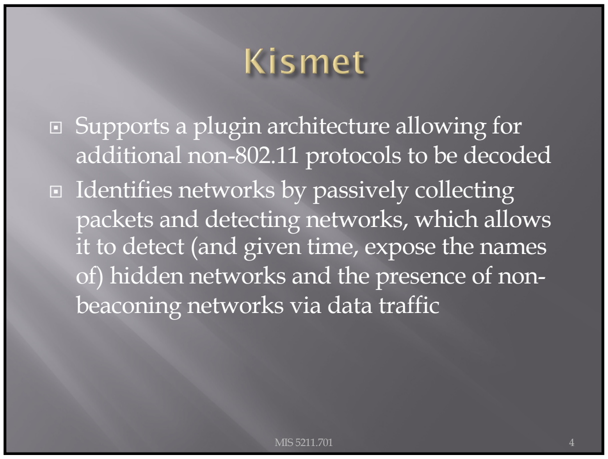
2



Kismet

- 802.11 wireless:
  - Network detector
  - Sniffer
  - Intrusion detection system
- Works with any wireless card which supports raw monitoring mode (not all do)
- Can sniff:
  - 802.11b
  - 802.11a
  - 802.11g
  - 802.11n

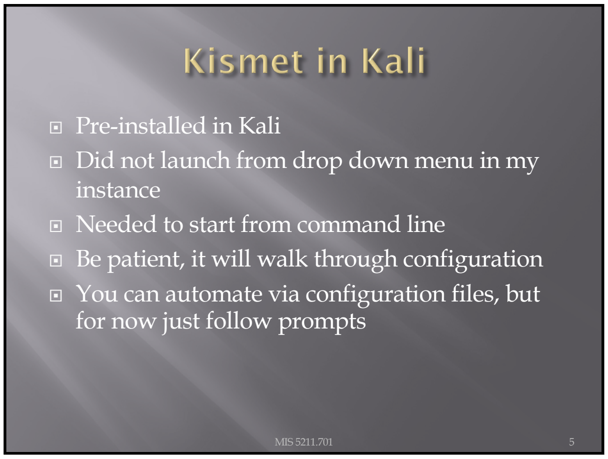
3



Kismet

- Supports a plugin architecture allowing for additional non-802.11 protocols to be decoded
- Identifies networks by passively collecting packets and detecting networks, which allows it to detect (and given time, expose the names of) hidden networks and the presence of non-beaconing networks via data traffic

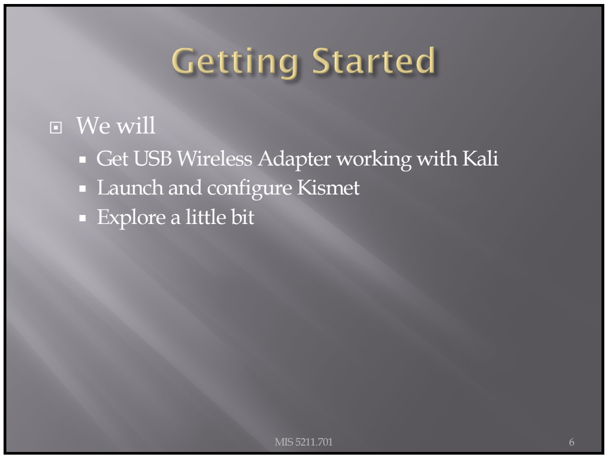
4



Kismet in Kali

- Pre-installed in Kali
- Did not launch from drop down menu in my instance
- Needed to start from command line
- Be patient, it will walk through configuration
- You can automate via configuration files, but for now just follow prompts

5

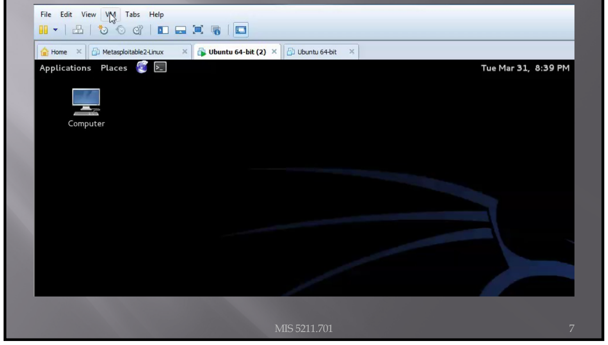


Getting Started

- We will
  - Get USB Wireless Adapter working with Kali
  - Launch and configure Kismet
  - Explore a little bit

6

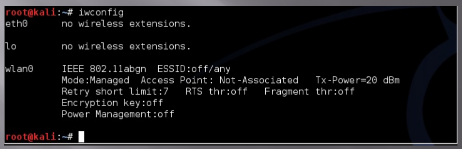
Connecting Wireless Card



7

Checking Card

- Use the command: iwconfig
- This should give something like the following:



```
root@kali:~# iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

root@kali:~# 
```

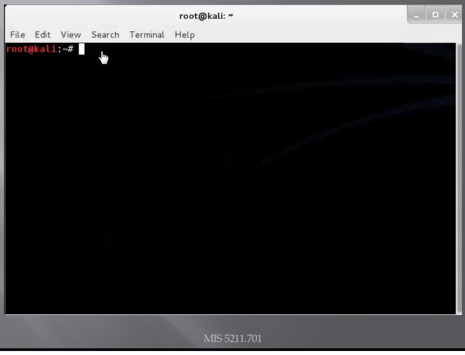
8

Kismet

- Kismet is a wireless network detector, sniffer, and intrusion detection system. Kismet works predominately with Wi-Fi (IEEE 802.11) networks, but can be expanded via plug-ins to handle other network types.
- Features
  - 802.11 sniffing
  - Standard PCAP logging (compatible with Wireshark, TCPDump, etc)
  - Client/Server modular architecture
  - Plug-in architecture to expand core features
  - Multiple capture source support
  - Live export of packets to other tools via tun/tap virtual interfaces
  - Distributed remote sniffing via light-weight remote capture
  - XML output for integration with other tools
- http://kismetwireless.net/

9

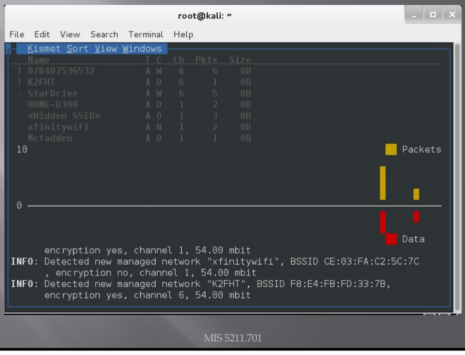
Starting Kismet



```
root@kali:~#
```

10

Kismet Example



```
  Name                 T C  Ch  Pkts  Size
! 07B407536532         A W   6     6    0B
! K2FHT                A O   6     1    0B
. StarDrive            A W   6     5    0B
  HOME-D390            A O   1     2    0B
  <Hidden SSID>        A O   1     3    0B
  xfinitywifi          A N   1     2    0B
  Mcfadden             A O   1     1    0B

      encryption yes, channel 1, 54.00 mbit
INFO: Detected new managed network "xfinitywifi", BSSID CE:03:FA:C2:5C:7C
      , encryption no, channel 1, 54.00 mbit
INFO: Detected new managed network "K2FHT", BSSID F8:E4:FB:FD:33:7B,
      encryption yes, channel 6, 54.00 mbit
```

11

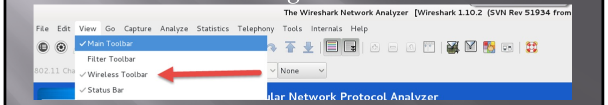
Wireshark

- Pre-installed in Kali

12

Startup of Wireshark

- Will throw an error due to running as root in Kali, just click OK and move on
- Will need to turn wireless menu on by going to View tab and clicking on "Wireless Toolbar"

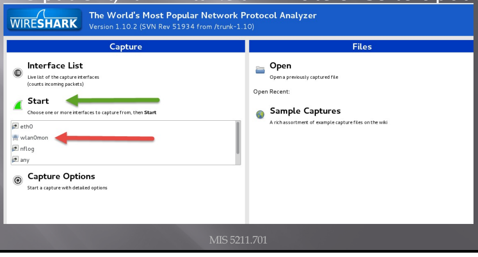

The Wireshark Network Analyzer [Wireshark 1.10.2 (SVN Rev 51934 from
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
✓ Main Toolbar
Filter Toolbar
✓ Wireless Toolbar
✓ Status Bar
802.11 Cha
None
ular Network Protocol Analyzer

MIS 5211.701 13

13

Configuring Interface

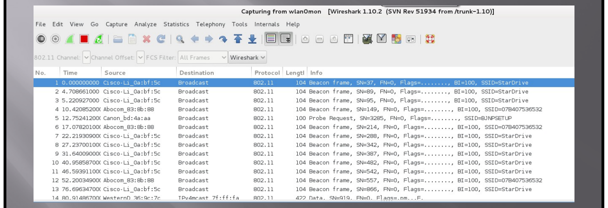
- Select "wlan0mon"
- Click on "Start"
- Be patient, it will take a minute or so to update


WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.10.2 (SVN Rev 51934 from /trunk-1.10)
Capture
Interface List
Live list of the capture interfaces
(counts incoming packets)
Start
Choose one or more interfaces to capture from, then Start
eth0
wlan0mon
nflog
any
Capture Options
Start a capture with detailed options
Files
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki

MIS 5211.701 14

14

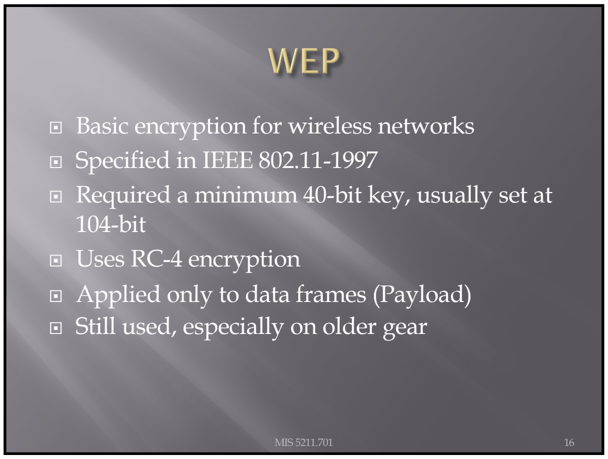
More Wireshark


Capturing from wlan0mon [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
802.11 Channel:
Channel Offset:
FCS Filter:
All Frames
Wireshark

```
No.  Time         Source               Destination          Protocol Lengtl Info
   1 0.000000000  Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=37, FN=0, Flags=........, BI=100, SSID=StarDrive
   2 4.708661000  Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=89, FN=0, Flags=........, BI=100, SSID=StarDrive
   3 5.220927000  Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=95, FN=0, Flags=........, BI=100, SSID=StarDrive
   4 10.42085200( Abocom_83:8b:88      Broadcast            802.11      104 Beacon frame, SN=149, FN=0, Flags=........, BI=100, SSID=07B407536532
   5 12.75241200( Canon_bd:4a:aa       Broadcast            802.11      100 Probe Request, SN=3285, FN=0, Flags=........, SSID=BJNPSETUP
   6 17.07820100( Abocom_83:8b:88      Broadcast            802.11      104 Beacon frame, SN=214, FN=0, Flags=........, BI=100, SSID=07B407536532
   7 22.21930900( Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=288, FN=0, Flags=........, BI=100, SSID=StarDrive
   8 27.23700100( Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=342, FN=0, Flags=........, BI=100, SSID=StarDrive
   9 31.64009000( Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=387, FN=0, Flags=........, BI=100, SSID=StarDrive
  10 40.95858700( Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=482, FN=0, Flags=........, BI=100, SSID=StarDrive
  11 46.59391100( Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=542, FN=0, Flags=........, BI=100, SSID=StarDrive
  12 52.20034900( Abocom_83:8b:88      Broadcast            802.11      104 Beacon frame, SN=557, FN=0, Flags=........, BI=100, SSID=07B407536532
  13 76.69634700( Cisco-Li_0a:bf:5c    Broadcast            802.11      104 Beacon frame, SN=866, FN=0, Flags=........, BI=100, SSID=StarDrive
  14 80.91486700( WesternD_36:9c:7c    IPv4mcast_7f:ff:fa   802.11      422 Data, SN=919, FN=0, Flags=.pm...F.
```

MIS 5211.701 15

15

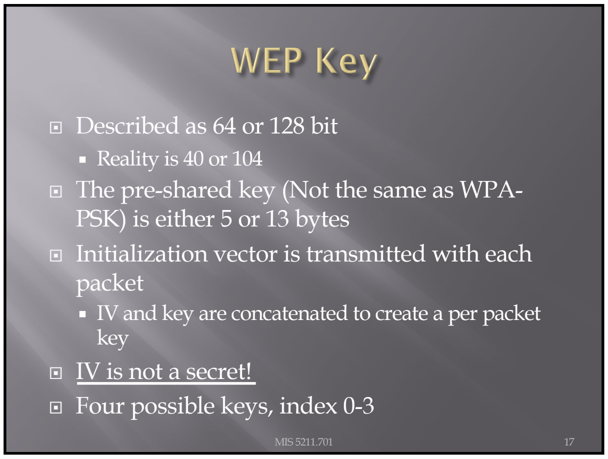


WEP

- Basic encryption for wireless networks
- Specified in IEEE 802.11-1997
- Required a minimum 40-bit key, usually set at 104-bit
- Uses RC-4 encryption
- Applied only to data frames (Payload)
- Still used, especially on older gear

MIS 5211.701 16

16

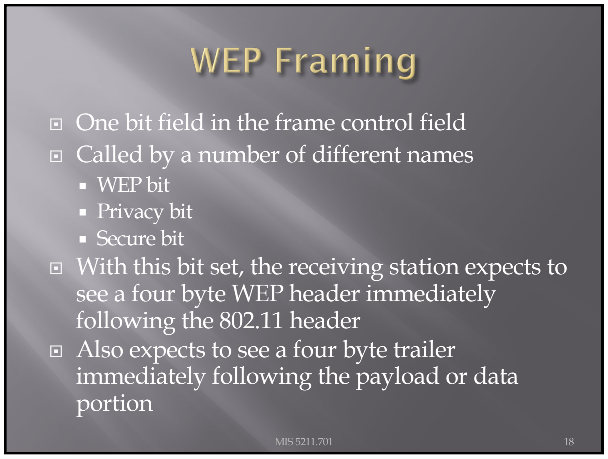


WEP Key

- Described as 64 or 128 bit
  - Reality is 40 or 104
- The pre-shared key (Not the same as WPA-PSK) is either 5 or 13 bytes
- Initialization vector is transmitted with each packet
  - IV and key are concatenated to create a per packet key
- IV is not a secret!
- Four possible keys, index 0-3

MIS 5211.701 17

17

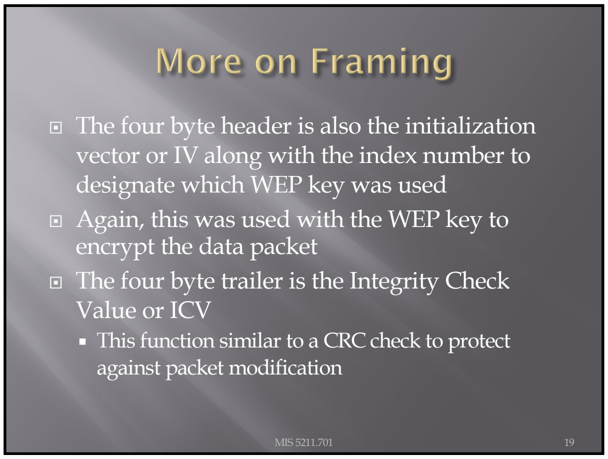


WEP Framing

- One bit field in the frame control field
- Called by a number of different names
  - WEP bit
  - Privacy bit
  - Secure bit
- With this bit set, the receiving station expects to see a four byte WEP header immediately following the 802.11 header
- Also expects to see a four byte trailer immediately following the payload or data portion

MIS 5211.701 18

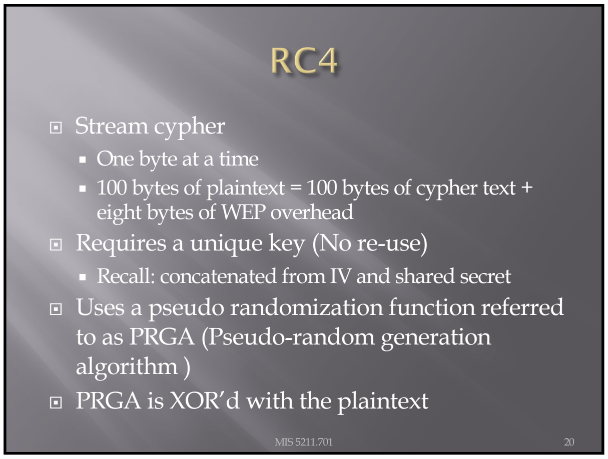
18



More on Framing

- The four byte header is also the initialization vector or IV along with the index number to designate which WEP key was used
- Again, this was used with the WEP key to encrypt the data packet
- The four byte trailer is the Integrity Check Value or ICV
  - This function similar to a CRC check to protect against packet modification

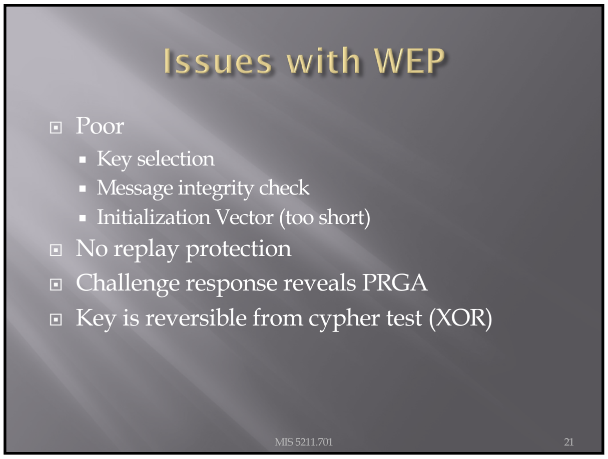
19



RC4

- Stream cypher
  - One byte at a time
  - 100 bytes of plaintext = 100 bytes of cypher text + eight bytes of WEP overhead
- Requires a unique key (No re-use)
  - Recall: concatenated from IV and shared secret
- Uses a pseudo randomization function referred to as PRGA (Pseudo-random generation algorithm)
- PRGA is XOR'd with the plaintext

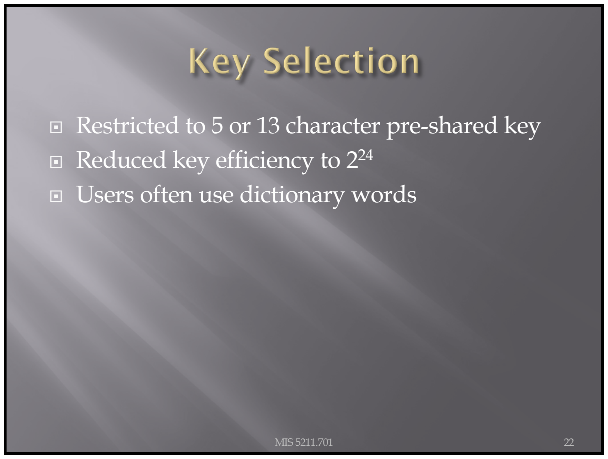
20



Issues with WEP

- Poor
  - Key selection
  - Message integrity check
  - Initialization Vector (too short)
- No replay protection
- Challenge response reveals PRGA
- Key is reversible from cypher test (XOR)

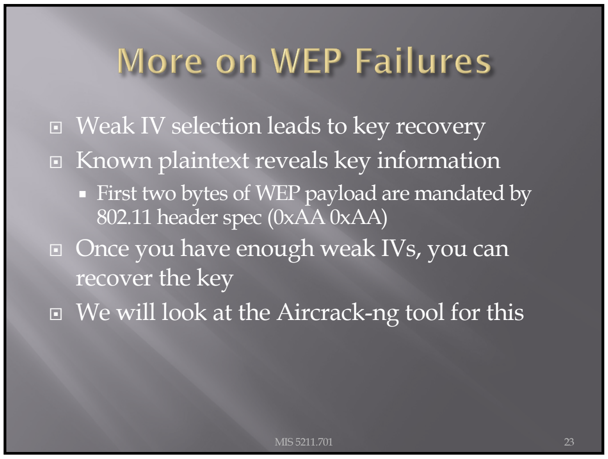
21



Key Selection

- Restricted to 5 or 13 character pre-shared key
- Reduced key efficiency to 2^{24}
- Users often use dictionary words

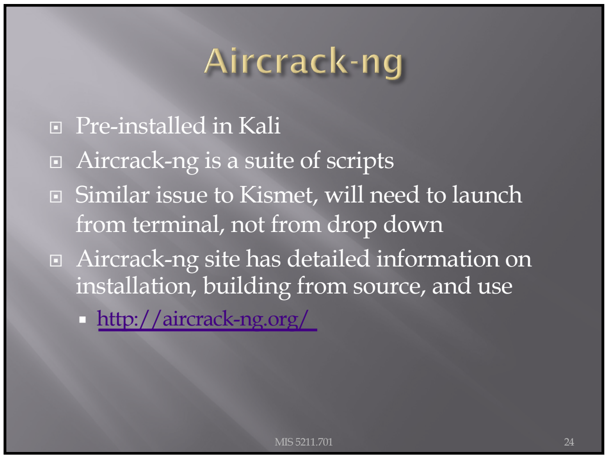
22



More on WEP Failures

- Weak IV selection leads to key recovery
- Known plaintext reveals key information
  - First two bytes of WEP payload are mandated by 802.11 header spec (0xAA 0xAA)
- Once you have enough weak IVs, you can recover the key
- We will look at the Aircrack-ng tool for this

23



Aircrack-ng

- Pre-installed in Kali
- Aircrack-ng is a suite of scripts
- Similar issue to Kismet, will need to launch from terminal, not from drop down
- Aircrack-ng site has detailed information on installation, building from source, and use
  - http://aircrack-ng.org/

24

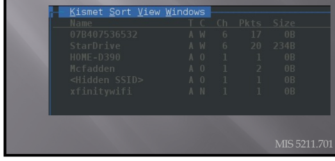
Extra Help w/ Aricrack

- Lots of extras at:
- http://aircrack-ng.org/doku.php?id=tutorial

25

Recall Kismet

- Need to connect wireless card to Kali
- Need to verify using iwconfig command
- Then launch Kismet for a little recon
  - This will also force the wireless card in to monitor mode
- Since StarDrive is my AP we'll focus on it

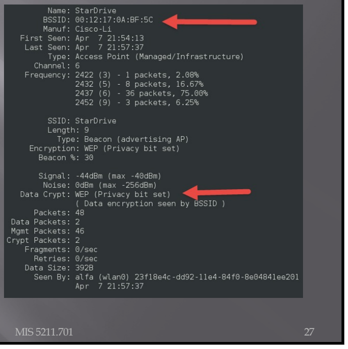


```
Kismet Sort View Windows
Name                T C  Ch  Pkts  Size
07B407536532        A W   6    17    0B
StarDrive           A W   6    20  234B
HOME-D390           A O   1     1    0B
Mcfadden            A O   1     2    0B
<Hidden SSID>       A O   1     1    0B
xfinitywifi         A N   1     1    0B
```

26

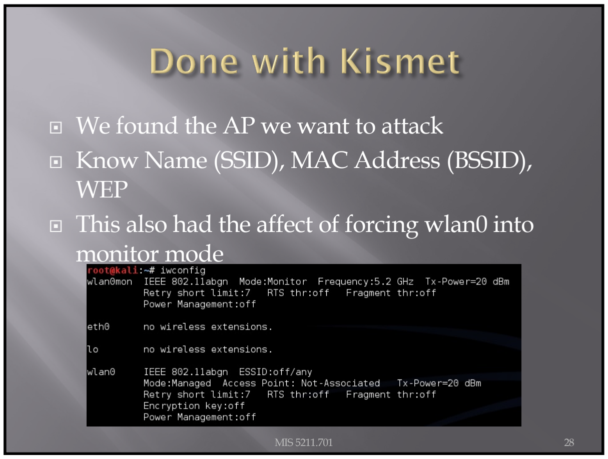
StarDrive

- Double clicking on name gives me detail screen
- Note
  - MAC Address
  - WEP bit
- "Network" menu has option to close window and return to summary



```
         Name: StarDrive
        BSSID: 00:12:17:0A:BF:5C
        Manuf: Cisco-Li
   First Seen: Apr  7 21:54:13
    Last Seen: Apr  7 21:57:37
         Type: Access Point (Managed/Infrastructure)
      Channel: 6
    Frequency: 2422 (3) - 1 packets, 2.08%
               2432 (5) - 8 packets, 16.67%
               2437 (6) - 36 packets, 75.00%
               2452 (9) - 3 packets, 6.25%

         SSID: StarDrive
       Length: 9
         Type: Beacon (advertising AP)
   Encryption: WEP (Privacy bit set)
     Beacon %: 30

       Signal: -44dBm (max -40dBm)
        Noise: 0dBm (max -256dBm)
   Data Crypt: WEP (Privacy bit set)
               ( Data encryption seen by BSSID )
      Packets: 48
 Data Packets: 2
 Mgmt Packets: 46
Crypt Packets: 2
    Fragments: 0/sec
      Retries: 0/sec
    Data Size: 392B
      Seen By: alfa (wlan0) 23f18e4c-dd92-11e4-84f0-8e04841ee201
               Apr  7 21:57:37
```

27

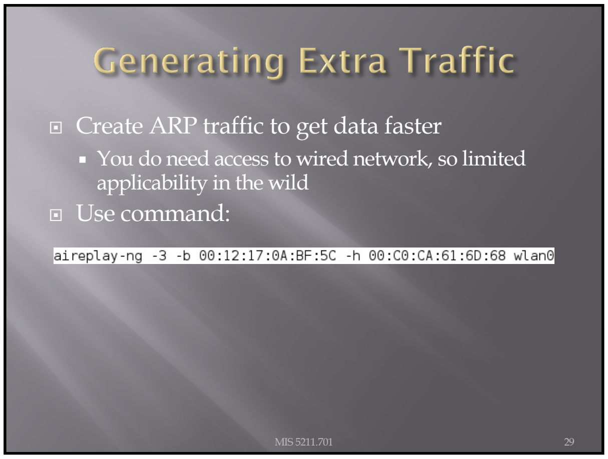


Done with Kismet

- We found the AP we want to attack
- Know Name (SSID), MAC Address (BSSID), WEP
- This also had the affect of forcing wlan0 into monitor mode

```
root@kali:~# iwconfig
wlan0mon  IEEE 802.11abgn  Mode:Monitor  Frequency:5.2 GHz  Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off

eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11abgn  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
```

28

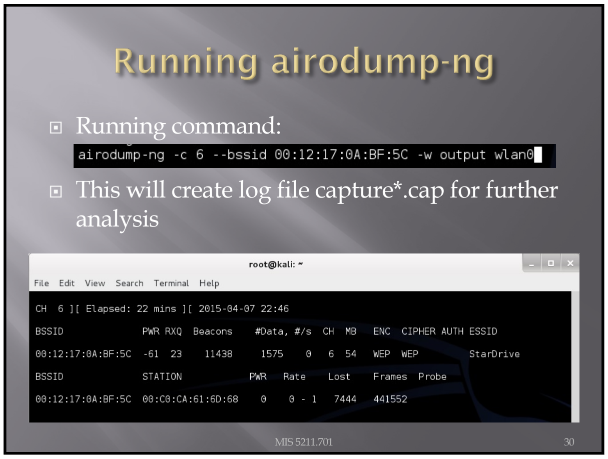


Generating Extra Traffic

- Create ARP traffic to get data faster
  - You do need access to wired network, so limited applicability in the wild
- Use command:

```
aireplay-ng -3 -b 00:12:17:0A:BF:5C -h 00:C0:CA:61:6D:68 wlan0
```

29



Running airodump-ng

- Running command:

```
airodump-ng -c 6 --bssid 00:12:17:0A:BF:5C -w output wlan0
```

- This will create log file capture*.cap for further analysis

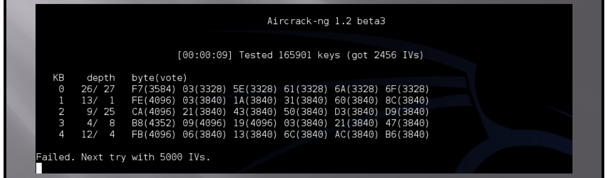
root@kali: ~

```
CH  6 ][ Elapsed: 22 mins ][ 2015-04-07 22:46

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

00:12:17:0A:BF:5C  -61  23    11438     1575    0   6  54   WEP  WEP         StarDrive

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

00:12:17:0A:BF:5C  00:C0:CA:61:6D:68    0    0 - 1   7444   441552
```

30

Finally, aircrack-ng

- Once enough data has been collected, run
  - aircrack-ng output*.cap
- If you don’t have enough data you will see

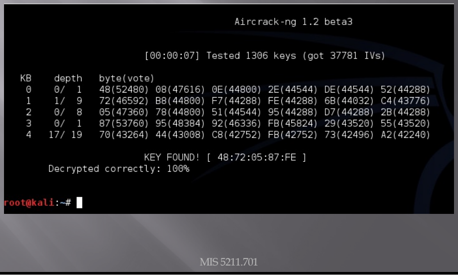


```
                                 Aircrack-ng 1.2 beta3


                     [00:00:09] Tested 165901 keys (got 2456 IVs)

   KB    depth   byte(vote)
    0   26/ 27   F7(3584) 03(3328) 5E(3328) 61(3328) 6A(3328) 6F(3328)
    1   13/  1   FE(4096) 03(3840) 1A(3840) 31(3840) 60(3840) 8C(3840)
    2    9/ 25   CA(4096) 21(3840) 43(3840) 50(3840) D3(3840) D9(3840)
    3    4/  8   B8(4352) 09(4096) 19(4096) 03(3840) 21(3840) 47(3840)
    4   12/  4   FB(4096) 06(3840) 13(3840) 6C(3840) AC(3840) B6(3840)

Failed. Next try with 5000 IVs.
```

31

With Enough Data

- Eventually, with enough IVs you can get to this:



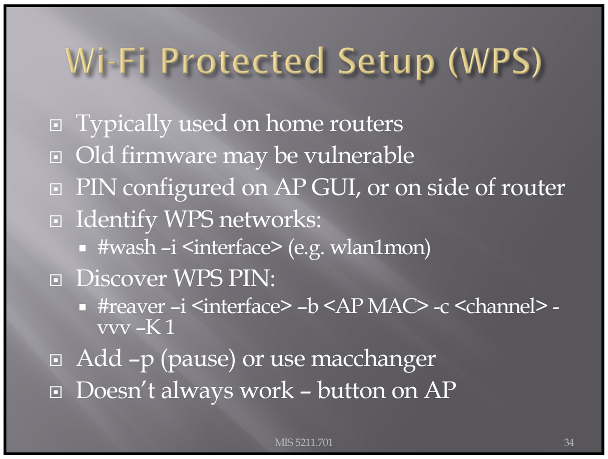
```
                                 Aircrack-ng 1.2 beta3


                     [00:00:07] Tested 1306 keys (got 37781 IVs)

   KB    depth   byte(vote)
    0    0/  1   48(52480) 08(47616) 0E(44800) 2E(44544) DE(44544) 52(44288)
    1    1/  9   72(46592) B8(44800) F7(44288) FE(44288) 6B(44032) C4(43776)
    2    0/  8   05(47360) 78(44800) 51(44544) 95(44288) D7(44288) 2B(44288)
    3    0/  1   87(53760) 95(48384) 92(46336) FB(45824) 29(43520) 55(43520)
    4   17/ 19   70(43264) 44(43008) C8(42752) FB(42752) 73(42496) A2(42240)

                     KEY FOUND! [ 48:72:05:87:FE ]
      Decrypted correctly: 100%


root@kali:~#
```

32

WPA-PSK

- Recall, WPA introduced TKIP
- WPA2 introduced CCMP and kept TKIP
- Both work with both personal and enterprise
  - Personal – PSK, Enterprise 802.1x
- WPA and WPA2 very similar for PSK

33

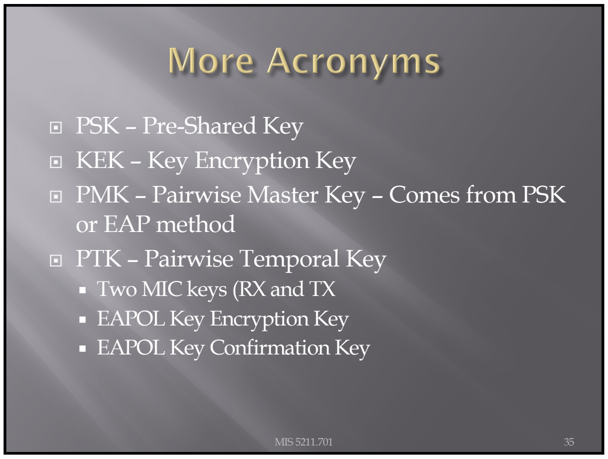


Wi-Fi Protected Setup (WPS)

- Typically used on home routers
- Old firmware may be vulnerable
- PIN configured on AP GUI, or on side of router
- Identify WPS networks:
  - #wash –i <interface> (e.g. wlan1mon)
- Discover WPS PIN:
  - #reaver –i <interface> –b <AP MAC> -c <channel> -vvv –K 1
- Add –p (pause) or use macchanger
- Doesn't always work – button on AP

MIS 5211.701 34

34

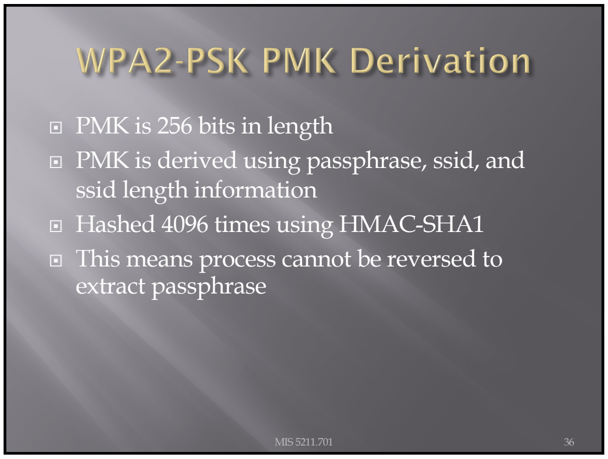


More Acronyms

- PSK – Pre-Shared Key
- KEK – Key Encryption Key
- PMK – Pairwise Master Key – Comes from PSK or EAP method
- PTK – Pairwise Temporal Key
  - Two MIC keys (RX and TX
  - EAPOL Key Encryption Key
  - EAPOL Key Confirmation Key

MIS 5211.701 35

35

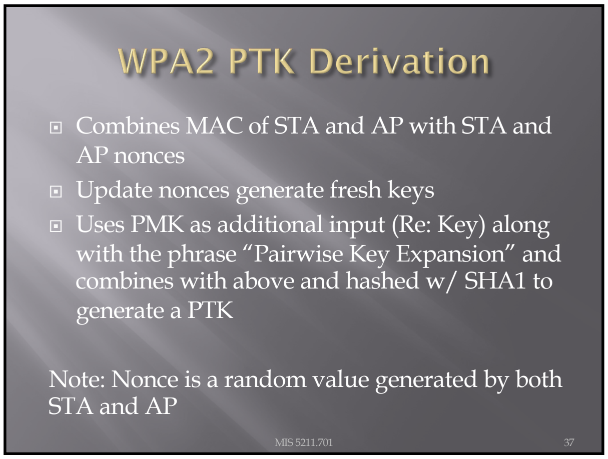


WPA2-PSK PMK Derivation

- PMK is 256 bits in length
- PMK is derived using passphrase, ssid, and ssid length information
- Hashed 4096 times using HMAC-SHA1
- This means process cannot be reversed to extract passphrase

MIS 5211.701 36

36

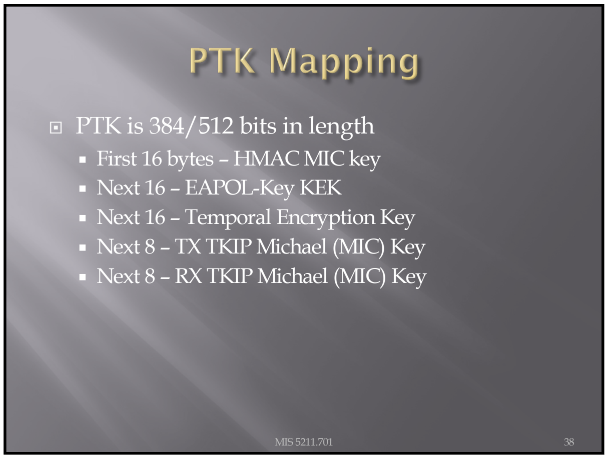


WPA2 PTK Derivation

- Combines MAC of STA and AP with STA and AP nonces
- Update nonces generate fresh keys
- Uses PMK as additional input (Re: Key) along with the phrase "Pairwise Key Expansion" and combines with above and hashed w/ SHA1 to generate a PTK

Note: Nonce is a random value generated by both STA and AP

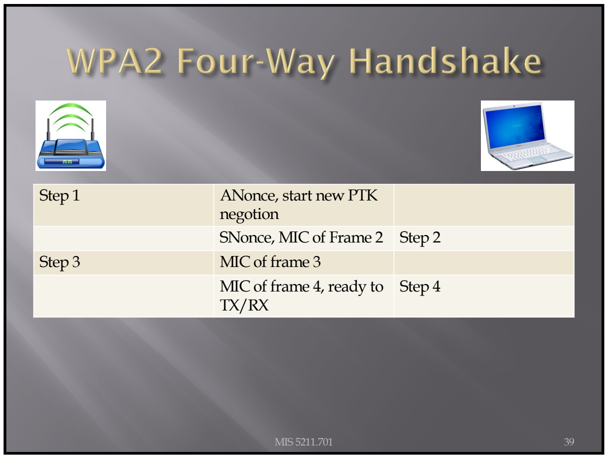
37



PTK Mapping

- PTK is 384/512 bits in length
  - First 16 bytes – HMAC MIC key
  - Next 16 – EAPOL-Key KEK
  - Next 16 – Temporal Encryption Key
  - Next 8 – TX TKIP Michael (MIC) Key
  - Next 8 – RX TKIP Michael (MIC) Key

38



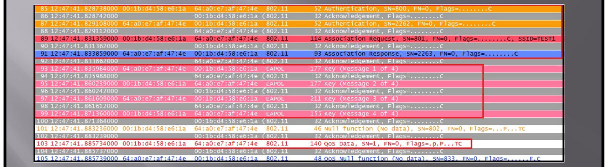
WPA2 Four-Way Handshake

Step 1	ANonce, start new PTK negotion	
	SNonce, MIC of Frame 2	Step 2
Step 3	MIC of frame 3	
	MIC of frame 4, ready to TX/RX	Step 4

39

WPA2 Four-Way Capture

- Example
  - First four lines are 4-Way Handshake



```
 85 12:47:41.828738000 00:1b:d4:58:e6:1a  64:a0:e7:af:47:4e  802.11       52 Authentication, SN=800, FN=0, Flags=........C
 86 12:47:41.828742000                    00:1b:d4:58:e6:1a (802.11       32 Acknowledgement, Flags=........C
 87 12:47:41.829108000 64:a0:e7:af:47:4e  00:1b:d4:58:e6:1a  802.11       52 Authentication, SN=2262, FN=0, Flags=........C
 88 12:47:41.829112000                    64:a0:e7:af:47:4e (802.11       32 Acknowledgement, Flags=........C
 89 12:47:41.831359000 00:1b:d4:58:e6:1a  64:a0:e7:af:47:4e  802.11      114 Association Request, SN=801, FN=0, Flags=........C, SSID=TEST1
 90 12:47:41.831362000                    00:1b:d4:58:e6:1a (802.11       32 Acknowledgement, Flags=........C
 91 12:47:41.833859000 64:a0:e7:af:47:4e  00:1b:d4:58:e6:1a  802.11       93 Association Response, SN=2263, FN=0, Flags=........C
 92 12:47:41.833862000                    64:a0:e7:af:47:4e (802.11       32 Acknowledgement, Flags=........C
 93 12:47:41.835984000 64:a0:e7:af:47:4e  00:1b:d4:58:e6:1a  EAPOL      177 Key (Message 1 of 4)
 94 12:47:41.835988000                    64:a0:e7:af:47:4e (802.11       32 Acknowledgement, Flags=........C
 95 12:47:41.860239000 00:1b:d4:58:e6:1a  64:a0:e7:af:47:4e  EAPOL      177 Key (Message 2 of 4)
 96 12:47:41.860242000                    00:1b:d4:58:e6:1a (802.11       32 Acknowledgement, Flags=........C
 97 12:47:41.861609000 64:a0:e7:af:47:4e  00:1b:d4:58:e6:1a  EAPOL      211 Key (Message 3 of 4)
 98 12:47:41.861612000                    64:a0:e7:af:47:4e (802.11       32 Acknowledgement, Flags=........C
 99 12:47:41.871360000 00:1b:d4:58:e6:1a  64:a0:e7:af:47:4e  EAPOL      155 Key (Message 4 of 4)
100 12:47:41.871364000                    00:1b:d4:58:e6:1a (802.11       32 Acknowledgement, Flags=........C
101 12:47:41.883236000 00:1b:d4:58:e6:1a  64:a0:e7:af:47:4e  802.11       46 Null function (No data), SN=802, FN=0, Flags=...P...TC
102 12:47:41.883239000                    00:1b:d4:58:e6:1a (802.11       32 Acknowledgement, Flags=........C
103 12:47:41.885734000 00:1b:d4:58:e6:1a  64:a0:e7:af:47:4e  802.11      140 QoS Data, SN=1, FN=0, Flags=.p.P...TC
104 12:47:41.885737000                    00:1b:d4:58:e6:1a (802.11       32 Acknowledgement, Flags=........C
105 12:47:41.885739000 64:a0:e7:af:47:4e  00:1b:d4:58:e6:1a  802.11       48 QoS Null function (No data), SN=833, FN=0, Flags=......F.C
```

- Source has capture file if you want to look for yourself

Source: http://mrncciew.com/2014/08/16/decrypt-wpa2-psk-using-wireshark/

40

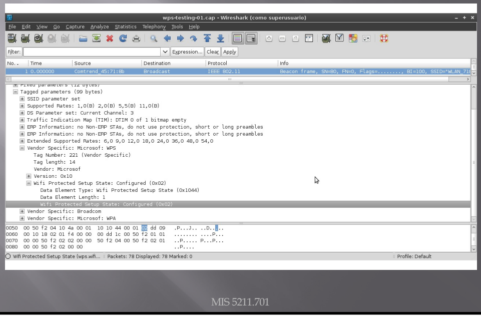
Identifying WPA2-PSK

- AP beacon frames identify capability information
  - Cypher suite support
  - Auth key management
- Wireshark can filter traffic, then manual inspection can identify

41

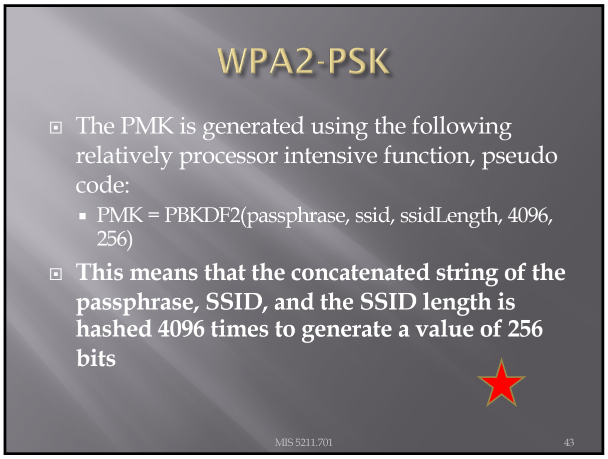
Identifying WPA2-PSK

- Example of beacon frame in wireshark


wps-testing-01.cap - Wireshark (como superusuario)
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
No.   Time       Source              Destination  Protocol      Info
  1 0.000000     Comtrend_45:71:8b   Broadcast    IEEE 802.11   Beacon frame, SN=80, FN=0, Flags=........, BI=100, SSID="WLAN_71

  ⊞ Fixed parameters (12 bytes)
  ⊟ Tagged parameters (99 bytes)
    ⊞ SSID parameter set
    ⊞ Supported Rates: 1,0(B) 2,0(B) 5,5(B) 11,0(B)
    ⊞ DS Parameter set: Current Channel: 3
    ⊞ Traffic Indication Map (TIM): DTIM 0 of 1 bitmap empty
    ⊞ ERP Information: no Non-ERP STAs, do not use protection, short or long preambles
    ⊞ ERP Information: no Non-ERP STAs, do not use protection, short or long preambles
    ⊞ Extended Supported Rates: 6,0 9,0 12,0 18,0 24,0 36,0 48,0 54,0
    ⊟ Vendor Specific: Microsof: WPS
        Tag Number: 221 (Vendor Specific)
        Tag length: 14
        Vendor: Microsof
      ⊞ Version: 0x10
      ⊟ Wifi Protected Setup State: Configured (0x02)
          Data Element Type: Wifi Protected Setup State (0x1044)
          Data Element Length: 1
          Wifi Protected Setup State: Configured (0x02)
    ⊞ Vendor Specific: Broadcom
    ⊞ Vendor Specific: Microsof: WPA

0050  00 50 f2 04 10 4a 00 01  10 10 44 00 01 02 dd 09   .P...J.. ..D.....
0060  00 10 18 02 01 f4 00 00  00 dd 1c 00 50 f2 01 01   ........ ....P...
0070  00 00 50 f2 02 02 00 00  50 f2 04 00 50 f2 02 01   ..P..... P...P...
0080  00 00 50 f2 02 00 00                               ..P....

○ Wifi Protected Setup State (wps.wifi...  Packets: 78 Displayed: 78 Marked: 0        Profile: Default
```

42

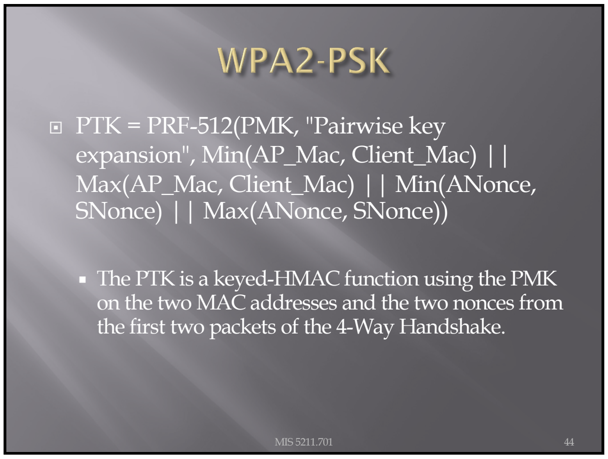


WPA2-PSK

- The PMK is generated using the following relatively processor intensive function, pseudo code:
  - PMK = PBKDF2(passphrase, ssid, ssidLength, 4096, 256)
- **This means that the concatenated string of the passphrase, SSID, and the SSID length is hashed 4096 times to generate a value of 256 bits**

MIS 5211.701 43

43

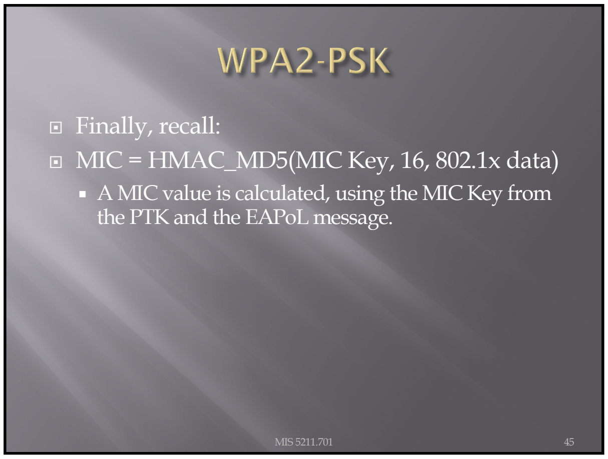


WPA2-PSK

- PTK = PRF-512(PMK, "Pairwise key expansion", Min(AP_Mac, Client_Mac) | | Max(AP_Mac, Client_Mac) | | Min(ANonce, SNonce) | | Max(ANonce, SNonce))
  - The PTK is a keyed-HMAC function using the PMK on the two MAC addresses and the two nonces from the first two packets of the 4-Way Handshake.

MIS 5211.701 44

44



WPA2-PSK

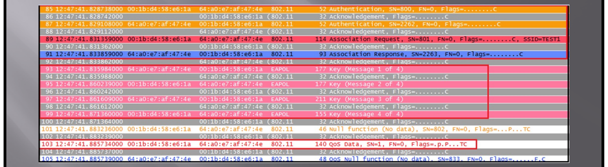
- Finally, recall:
- MIC = HMAC_MD5(MIC Key, 16, 802.1x data)
  - A MIC value is calculated, using the MIC Key from the PTK and the EAPoL message.

MIS 5211.701 45

45

WPA2-PSK

- So, we captured the Mac Addresses and the ANonce and SNonce from the four way handshake



```
 85 12:47:41.828738000 00:1b:d4:58:e6:1a 64:a0:e7:af:47:4e  802.11     52 Authentication, SN=800, FN=0, Flags=........C
 86 12:47:41.828742000                   00:1b:d4:58:e6:1a (802.11     32 Acknowledgement, Flags=........C
 87 12:47:41.829108000 64:a0:e7:af:47:4e 00:1b:d4:58:e6:1a  802.11     52 Authentication, SN=2262, FN=0, Flags=........C
 88 12:47:41.829112000                   64:a0:e7:af:47:4e (802.11     32 Acknowledgement, Flags=........C
 89 12:47:41.831359000 00:1b:d4:58:e6:1a 64:a0:e7:af:47:4e  802.11    114 Association Request, SN=801, FN=0, Flags=........C, SSID=TEST1
 90 12:47:41.831362000                   00:1b:d4:58:e6:1a (802.11     32 Acknowledgement, Flags=........C
 91 12:47:41.833859000 64:a0:e7:af:47:4e 00:1b:d4:58:e6:1a  802.11     93 Association Response, SN=2263, FN=0, Flags=........C
 92 12:47:41.833862000                   64:a0:e7:af:47:4e (802.11     32 Acknowledgement, Flags=........C
 93 12:47:41.835984000 64:a0:e7:af:47:4e 00:1b:d4:58:e6:1a  EAPOL     177 Key (Message 1 of 4)
 94 12:47:41.835988000                   64:a0:e7:af:47:4e (802.11     32 Acknowledgement, Flags=........C
 95 12:47:41.860239000 00:1b:d4:58:e6:1a 64:a0:e7:af:47:4e  EAPOL     177 Key (Message 2 of 4)
 96 12:47:41.860242000                   00:1b:d4:58:e6:1a (802.11     32 Acknowledgement, Flags=........C
 97 12:47:41.861609000 64:a0:e7:af:47:4e 00:1b:d4:58:e6:1a  EAPOL     211 Key (Message 3 of 4)
 98 12:47:41.861612000                   64:a0:e7:af:47:4e (802.11     32 Acknowledgement, Flags=........C
 99 12:47:41.871360000 00:1b:d4:58:e6:1a 64:a0:e7:af:47:4e  EAPOL     155 Key (Message 4 of 4)
100 12:47:41.871364000                   00:1b:d4:58:e6:1a (802.11     32 Acknowledgement, Flags=........C
101 12:47:41.883236000 00:1b:d4:58:e6:1a 64:a0:e7:af:47:4e  802.11     46 Null function (No data), SN=802, FN=0, Flags=...P...TC
102 12:47:41.883239000                   00:1b:d4:58:e6:1a (802.11     32 Acknowledgement, Flags=........C
103 12:47:41.885734000 00:1b:d4:58:e6:1a 64:a0:e7:af:47:4e  802.11    140 QoS Data, SN=1, FN=0, Flags=.p.P...TC
104 12:47:41.885737000                   00:1b:d4:58:e6:1a (802.11     32 Acknowledgement, Flags=........C
105 12:47:41.885739000 64:a0:e7:af:47:4e 00:1b:d4:58:e6:1a  802.11     48 QoS Null function (No data), SN=833, FN=0, Flags=......F.C
```

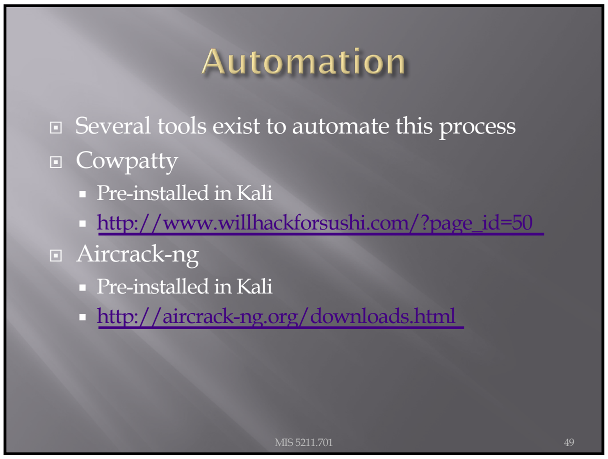
Source: http://mrncciew.com/2014/08/16/decrypt-wpa2-psk-using-wireshark/

WPA2-PSK

- Now, if we had the right passphrase, SSID, and SSID length; we have everything we need to generate our own key.
- But we don't have this information!
- At least not directly

Process

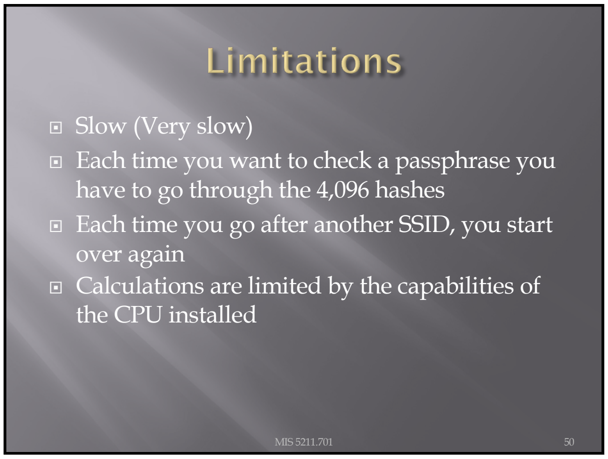
- Collect data from four way handshake
  - Mac Addresses
  - ANonce and SNonce
  - MIC and EAP
- Read in value from a dictionary list
- Calculate PMK using dictionary word and SSID
- Calculate PTK using above information
- Calculate MIC of frame using PTK
- Compare calculated MIC to observed MIC
- If equal, done! If not equal read in next dictionary word and start over



Automation

- Several tools exist to automate this process
- Cowpatty
  - Pre-installed in Kali
  - http://www.willhackforsushi.com/?page_id=50
- Aircrack-ng
  - Pre-installed in Kali
  - http://aircrack-ng.org/downloads.html

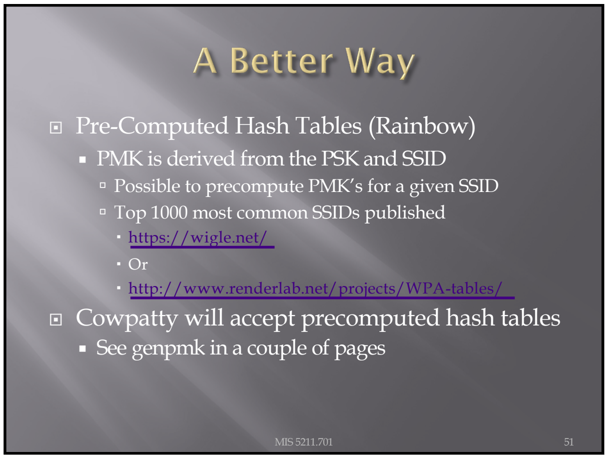
49



Limitations

- Slow (Very slow)
- Each time you want to check a passphrase you have to go through the 4,096 hashes
- Each time you go after another SSID, you start over again
- Calculations are limited by the capabilities of the CPU installed

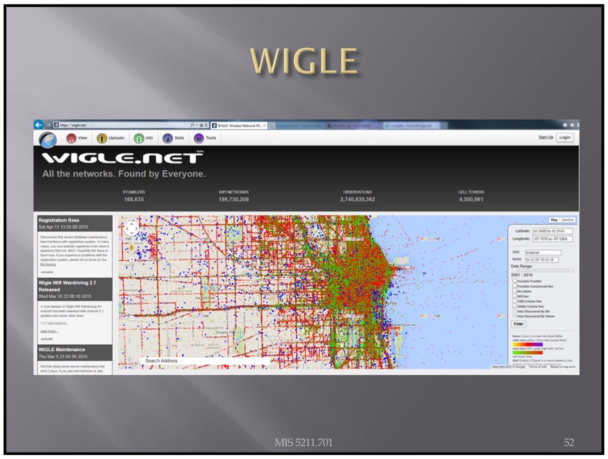
50



A Better Way

- Pre-Computed Hash Tables (Rainbow)
  - PMK is derived from the PSK and SSID
    - Possible to precompute PMK's for a given SSID
    - Top 1000 most common SSIDs published
      - https://wigle.net/
      - Or
      - http://www.renderlab.net/projects/WPA-tables/
- Cowpatty will accept precomputed hash tables
  - See genpmk in a couple of pages

51



WIGLE

STUMBLERS	WIFI NETWORKS	OBSERVATIONS	CELL TOWERS
168,635	186,750,208	2,740,830,362	4,500,961

Registration fixes
Sat Apr 11 13:55:00 2015

Discovered that recent database maintenance had interfered with registration system. In many cases, you successfully registered even when it appeared that you didn't. Hopefully the issue is fixed now; if you experience problems with the registration system, please let us know on the the forums

-arkasha

Wigle Wifi Wardriving 2.7 Released
Wed Mar 18 22:06:10 2015

A new release of Wigle Wifi Wardriving for Android has been released with Android 5.1 updates and some other fixes.

* 2.7 (03/18/2015)

read more...

-bobzilla

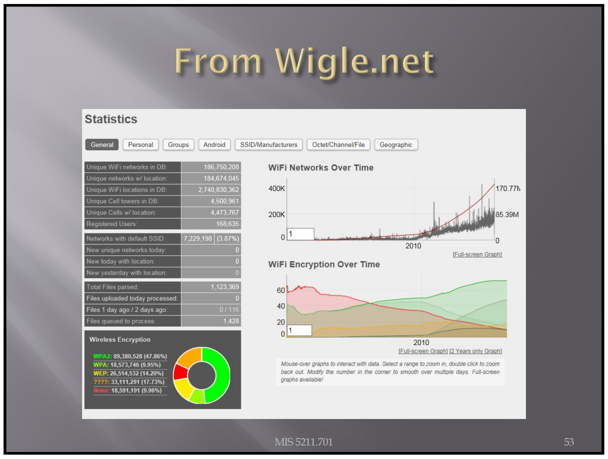
WiGLE Maintenance
Thu Mar 5 21:04:56 2015

We'll be doing some server maintenance the next 2 days, if you see odd behavior or see

☐ Possible FreeNet
☐ Possible Commercial Net
☐ No Labels
☐ Wifi Net
☐ GSM Cellular Net
☐ CDMA Cellular Net
☐ Only Discovered By Me
☐ Only Discovered By Others

Notes: Zoom in to see individual SSIDs.
wide view: yellow: more nets, purple fewer
near view: WiFi: green high QoS, red low
cell tower: blue
QoS: Quality of Signal is a metric based on the

52



From Wigle.net

Statistics

Unique WiFi networks in DB:	186,750,208	
Unique networks w/ location:	184,674,045	
Unique WiFi locations in DB:	2,740,830,362	
Unique Cell towers in DB:	4,500,961	
Unique Cells w/ location:	4,473,767	
Registered Users:	168,635	
Networks with default SSID:	7,229,198	(3.87%)
New unique networks today:	0	
New today with location:	0	
New yesterday with location:	0	
Total Files parsed:	1,123,369	
Files uploaded today processed:	0	
Files 1 day ago / 2 days ago:	0 / 116	
Files queued to process:	1,428	

Wireless Encryption

WPA2:	89,380,528 (47.86%)
WPA:	18,573,746 (9.95%)
WEP:	26,514,532 (14.20%)
????:	33,111,291 (17.73%)
None:	18,591,191 (9.96%)

Mouse-over graphs to interact with data. Select a range to zoom in, double click to zoom back out. Modify the number in the corner to smooth over multiple days. Full-screen graphs available!

53

More Tools (genpmk)

- Basic tool to precompute hashes
- Can speed up attacks by a factor of 1300
- “genpmk” written by Josh Wright
  - Pre-installed in Kali
  - Packaged with Cowpatty

54

But I Want To Do This Myself

- CUDA Acceleration
  - Parallel computing architecture developed by nVIDIA
- Pyrite – CUDA acceleration of Cowpatty PMK tables
  - Included in Kali
- Pyrite also supports AMD/ATI 43XX cards (they typically cost less)
- Could also go to the cloud

55

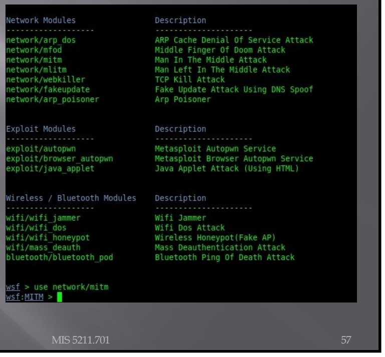
Wireless Network Encryption (Summary)

Method	Algorithm	IV Size	Key Length	Key Manageme nt	Integrity Check
WEP	RC4	24	40/104	None	CRC-32
WPA	RC4, TKIP	48	128	4-way	Michael Algorithm and CRC-32
WPA2	AES-CCMP	48	128	4-way	CBC-MAC
WPA3	AES-GCMP 256	Arbitrary 1-2^64	192	ECDH and ECDSA	BIP-GMAC-256

56

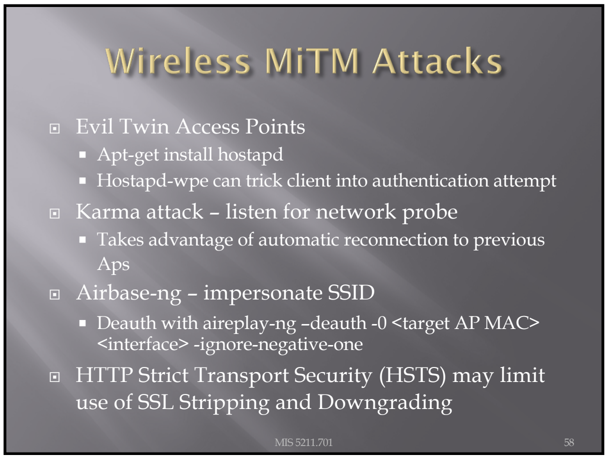
Wireless Attacks - Websploit

- NOTE – Intentional wireless jamming can be illegal
- Obtain via apt-get install websploit, then type websploit at command prompt



```
Network Modules                Description
-------------------            ---------------------
network/arp_dos                ARP Cache Denial Of Service Attack
network/mfod                   Middle Finger Of Doom Attack
network/mitm                   Man In The Middle Attack
network/mlitm                  Man Left In The Middle Attack
network/webkiller              TCP Kill Attack
network/fakeupdate             Fake Update Attack Using DNS Spoof
network/arp_poisoner           Arp Poisoner


Exploit Modules                Description
-------------------            ---------------------
exploit/autopwn                Metasploit Autopwn Service
exploit/browser_autopwn        Metasploit Browser Autopwn Service
exploit/java_applet            Java Applet Attack (Using HTML)


Wireless / Bluetooth Modules   Description
-------------------            ---------------------
wifi/wifi_jammer               Wifi Jammer
wifi/wifi_dos                  Wifi Dos Attack
wifi/wifi_honeypot             Wireless Honeypot(Fake AP)
wifi/mass_deauth               Mass Deauthentication Attack
bluetooth/bluetooth_pod        Bluetooth Ping Of Death Attack


wsf > use network/mitm
wsf:MITM >
```

57

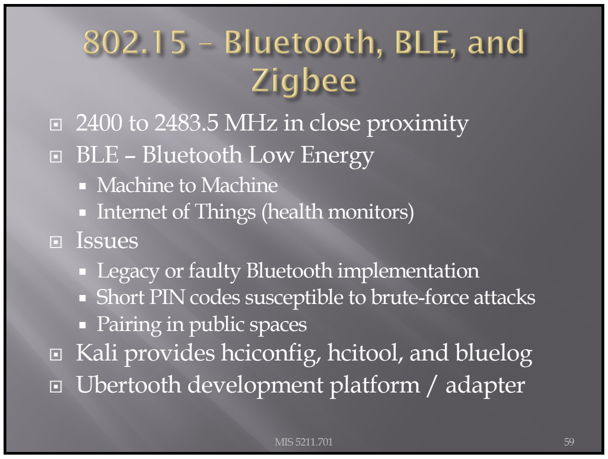


Wireless MiTM Attacks

- Evil Twin Access Points
  - Apt-get install hostapd
  - Hostapd-wpe can trick client into authentication attempt
- Karma attack – listen for network probe
  - Takes advantage of automatic reconnection to previous Aps
- Airbase-ng – impersonate SSID
  - Deauth with aireplay-ng –deauth -0 <target AP MAC> <interface> -ignore-negative-one
- HTTP Strict Transport Security (HSTS) may limit use of SSL Stripping and Downgrading

MIS 5211.701 58

58

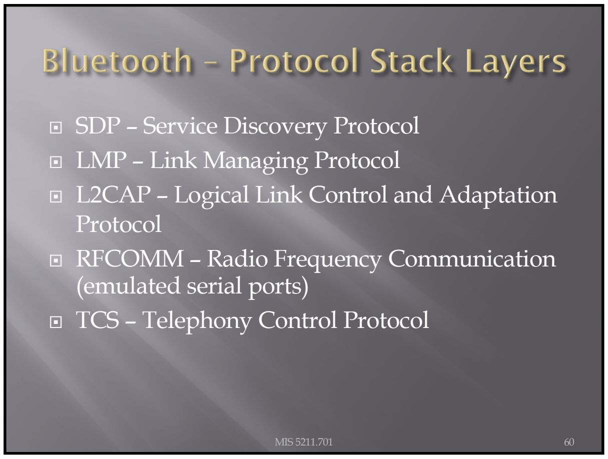


802.15 – Bluetooth, BLE, and Zigbee

- 2400 to 2483.5 MHz in close proximity
- BLE – Bluetooth Low Energy
  - Machine to Machine
  - Internet of Things (health monitors)
- Issues
  - Legacy or faulty Bluetooth implementation
  - Short PIN codes susceptible to brute-force attacks
  - Pairing in public spaces
- Kali provides hciconfig, hcitool, and bluelog
- Ubertooth development platform / adapter

MIS 5211.701 59

59

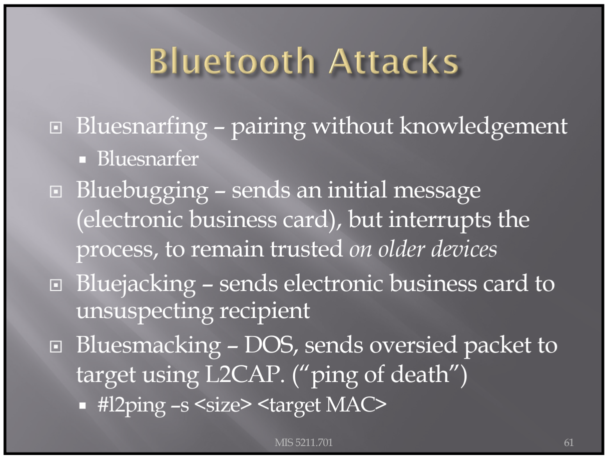


Bluetooth – Protocol Stack Layers

- SDP – Service Discovery Protocol
- LMP – Link Managing Protocol
- L2CAP – Logical Link Control and Adaptation Protocol
- RFCOMM – Radio Frequency Communication (emulated serial ports)
- TCS – Telephony Control Protocol

MIS 5211.701 60

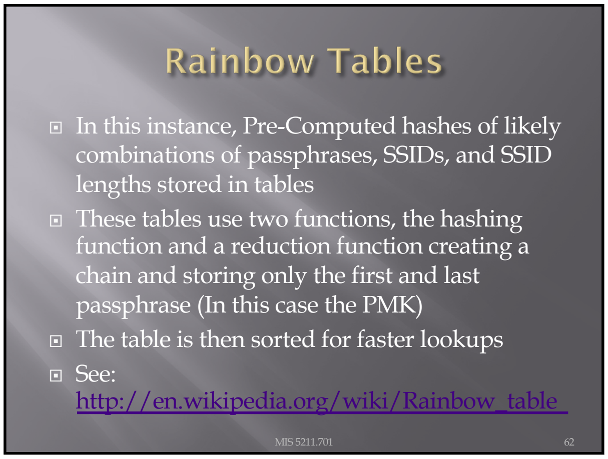
60



Bluetooth Attacks

- Bluesnarfing – pairing without knowledgement
  - Bluesnarfer
- Bluebugging – sends an initial message (electronic business card), but interrupts the process, to remain trusted *on older devices*
- Bluejacking – sends electronic business card to unsuspecting recipient
- Bluesmacking – DOS, sends oversied packet to target using L2CAP. ("ping of death")
  - #l2ping –s <size> <target MAC>

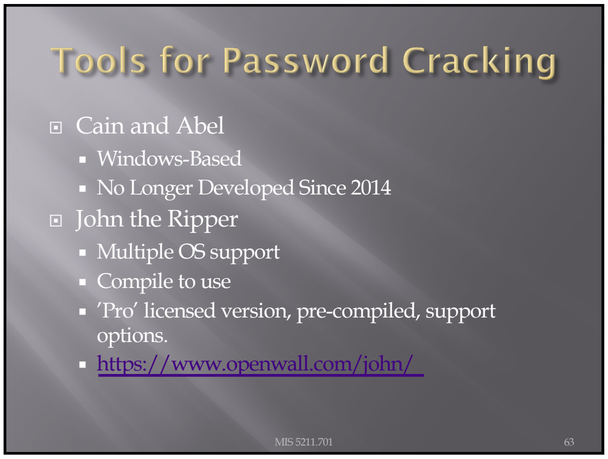
61



Rainbow Tables

- In this instance, Pre-Computed hashes of likely combinations of passphrases, SSIDs, and SSID lengths stored in tables
- These tables use two functions, the hashing function and a reduction function creating a chain and storing only the first and last passphrase (In this case the PMK)
- The table is then sorted for faster lookups
- See: http://en.wikipedia.org/wiki/Rainbow_table

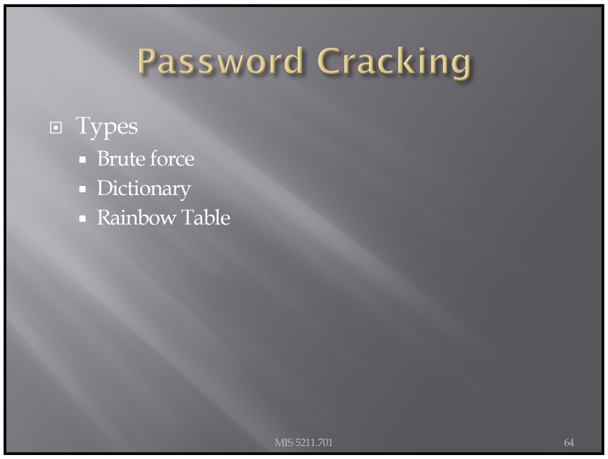
62



Tools for Password Cracking

- Cain and Abel
  - Windows-Based
  - No Longer Developed Since 2014
- John the Ripper
  - Multiple OS support
  - Compile to use
  - 'Pro' licensed version, pre-compiled, support options.
  - https://www.openwall.com/john/

63

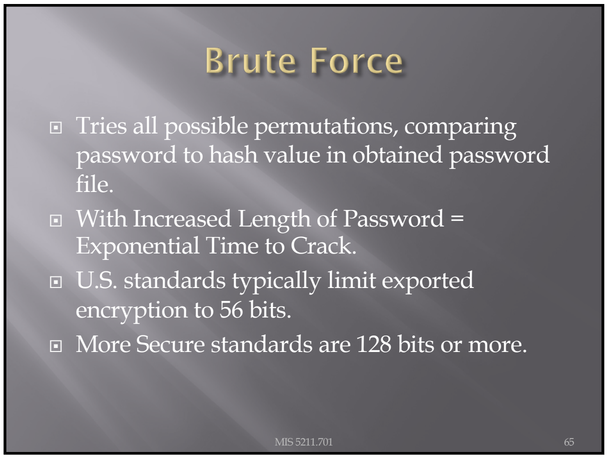


Password Cracking

- Types
  - Brute force
  - Dictionary
  - Rainbow Table

MIS 5211.701 64

64

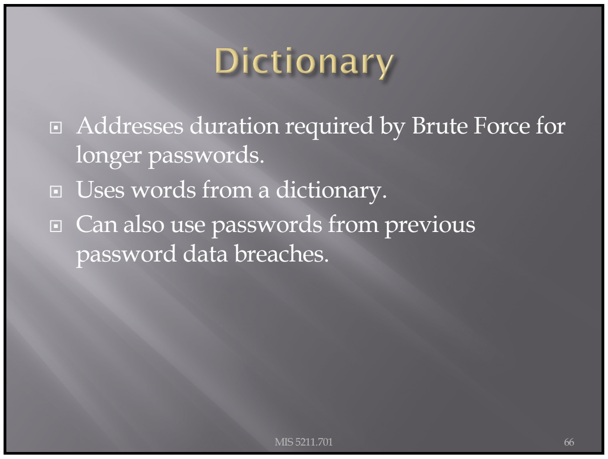


Brute Force

- Tries all possible permutations, comparing password to hash value in obtained password file.
- With Increased Length of Password = Exponential Time to Crack.
- U.S. standards typically limit exported encryption to 56 bits.
- More Secure standards are 128 bits or more.

MIS 5211.701 65

65

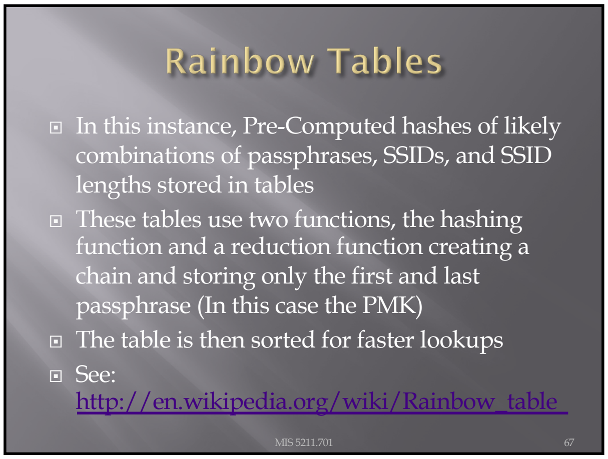


Dictionary

- Addresses duration required by Brute Force for longer passwords.
- Uses words from a dictionary.
- Can also use passwords from previous password data breaches.

MIS 5211.701 66

66

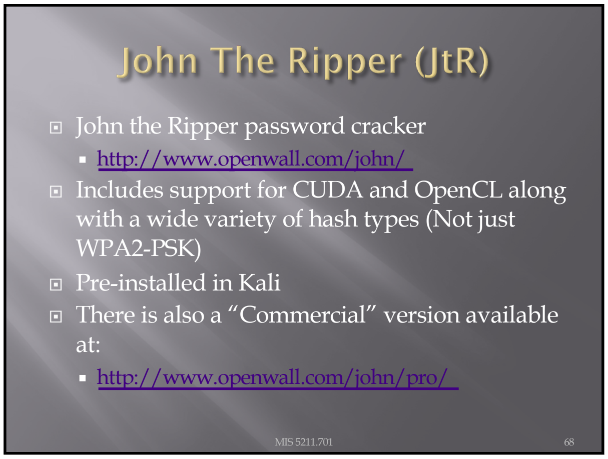


Rainbow Tables

- In this instance, Pre-Computed hashes of likely combinations of passphrases, SSIDs, and SSID lengths stored in tables
- These tables use two functions, the hashing function and a reduction function creating a chain and storing only the first and last passphrase (In this case the PMK)
- The table is then sorted for faster lookups
- See: http://en.wikipedia.org/wiki/Rainbow_table

MIS 5211.701 67

67

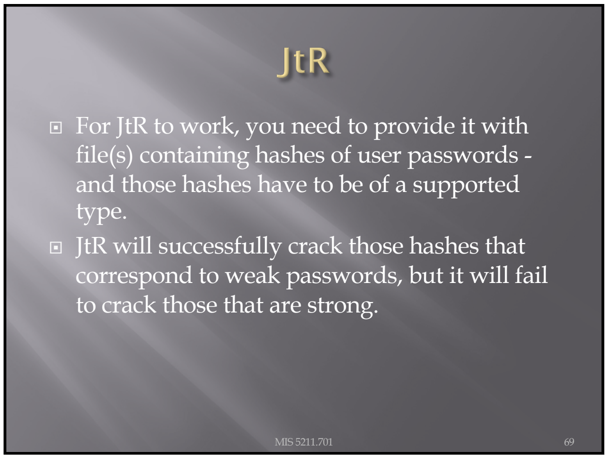


John The Ripper (JtR)

- John the Ripper password cracker
  - http://www.openwall.com/john/
- Includes support for CUDA and OpenCL along with a wide variety of hash types (Not just WPA2-PSK)
- Pre-installed in Kali
- There is also a "Commercial" version available at:
  - http://www.openwall.com/john/pro/

MIS 5211.701 68

68



JtR

- For JtR to work, you need to provide it with file(s) containing hashes of user passwords - and those hashes have to be of a supported type.
- JtR will successfully crack those hashes that correspond to weak passwords, but it will fail to crack those that are strong.

MIS 5211.701 69

69

JtR and Kali

- As several other tools have done, will not launch from drop down
- Open terminal and type:
  - “john --test” this will launch a diagnostic and give you benchmarking numbers for how your system performs
  - Note: this is one instance where running in a VM is a bad idea. Performance will be poor
  - Consider installing directly on a test machine

70

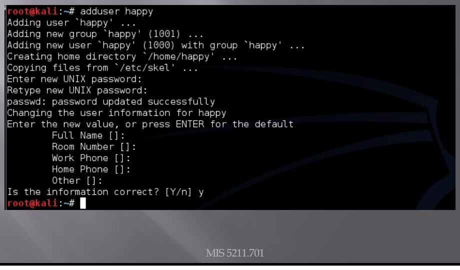
JtR Usage

- At it’s core, very simple
- Find a file with hashes in it
- Run: john passwordlist ~/file

71

First Lets Add a User

- Run command adduser happy
- Use password chess when prompted

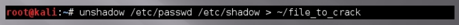


```
root@kali:~# adduser happy
Adding user `happy' ...
Adding new group `happy' (1001) ...
Adding new user `happy' (1000) with group `happy' ...
Creating home directory `/home/happy' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for happy
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@kali:~#
```

72

Now Extract Password File

- Run command unshadow as follows



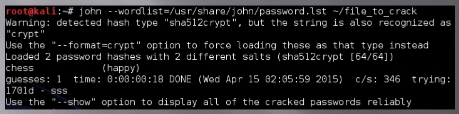
```
root@kali:~# unshadow /etc/passwd /etc/shadow > ~/file_to_crack
```

- This extracts the passwd and shadow file and combines them together to create a file you can go after
- If you were an attacker, this is what is meant by extracting or harvesting password files
- In Windows you would go after the SAM file

73

Now we Crack

- Run the john command as follows



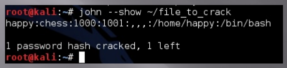
```
root@kali:~# john --wordlist=/usr/share/john/password.lst ~/file_to_crack
Warning: detected hash type "sha512crypt", but the string is also recognized as
"crypt"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 2 password hashes with 2 different salts (sha512crypt [64/64])
chess            (happy)
guesses: 1  time: 0:00:00:18 DONE (Wed Apr 15 02:05:59 2015)  c/s: 346  trying:
1701d - sss
Use the "--show" option to display all of the cracked passwords reliably
```

- This tells john to use a wordlist that is preinstalled in Kali (and has chess as an entry)
- And tells john to apply it against the file: file_to_crack

74

Checking Work

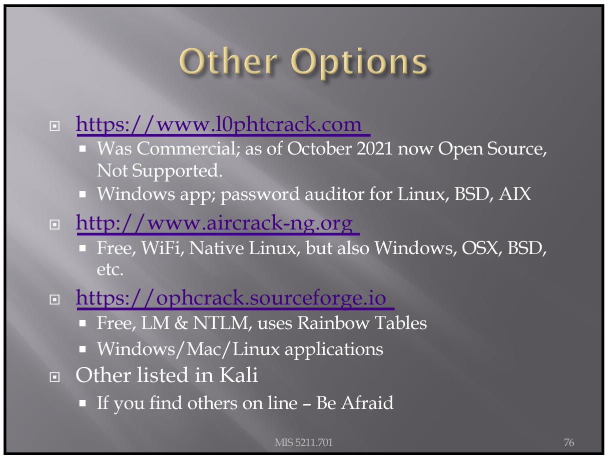
- Using the show command



```
root@kali:~# john --show ~/file_to_crack
happy:chess:1000:1001:,,,:/home/happy:/bin/bash

1 password hash cracked, 1 left
root@kali:~# █
```

- Note: If you have not recently updated Kali 2.0 you may get errors.

75

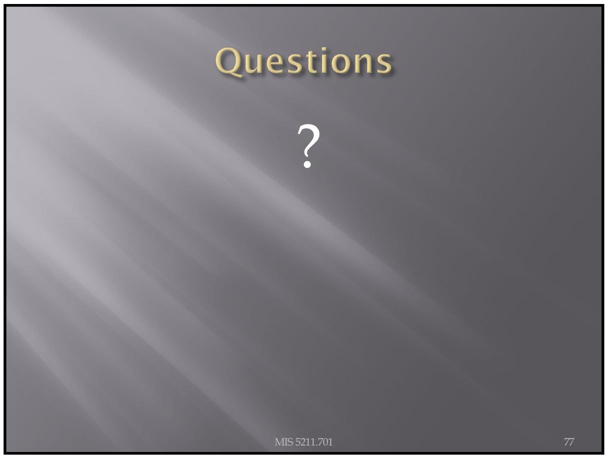


Other Options

- https://www.l0phtcrack.com
  - Was Commercial; as of October 2021 now Open Source, Not Supported.
  - Windows app; password auditor for Linux, BSD, AIX
- http://www.aircrack-ng.org
  - Free, WiFi, Native Linux, but also Windows, OSX, BSD, etc.
- https://ophcrack.sourceforge.io
  - Free, LM & NTLM, uses Rainbow Tables
  - Windows/Mac/Linux applications
- Other listed in Kali
  - If you find others on line – Be Afraid

MIS 5211.701 76

76



Questions

?

MIS 5211.701 77

77