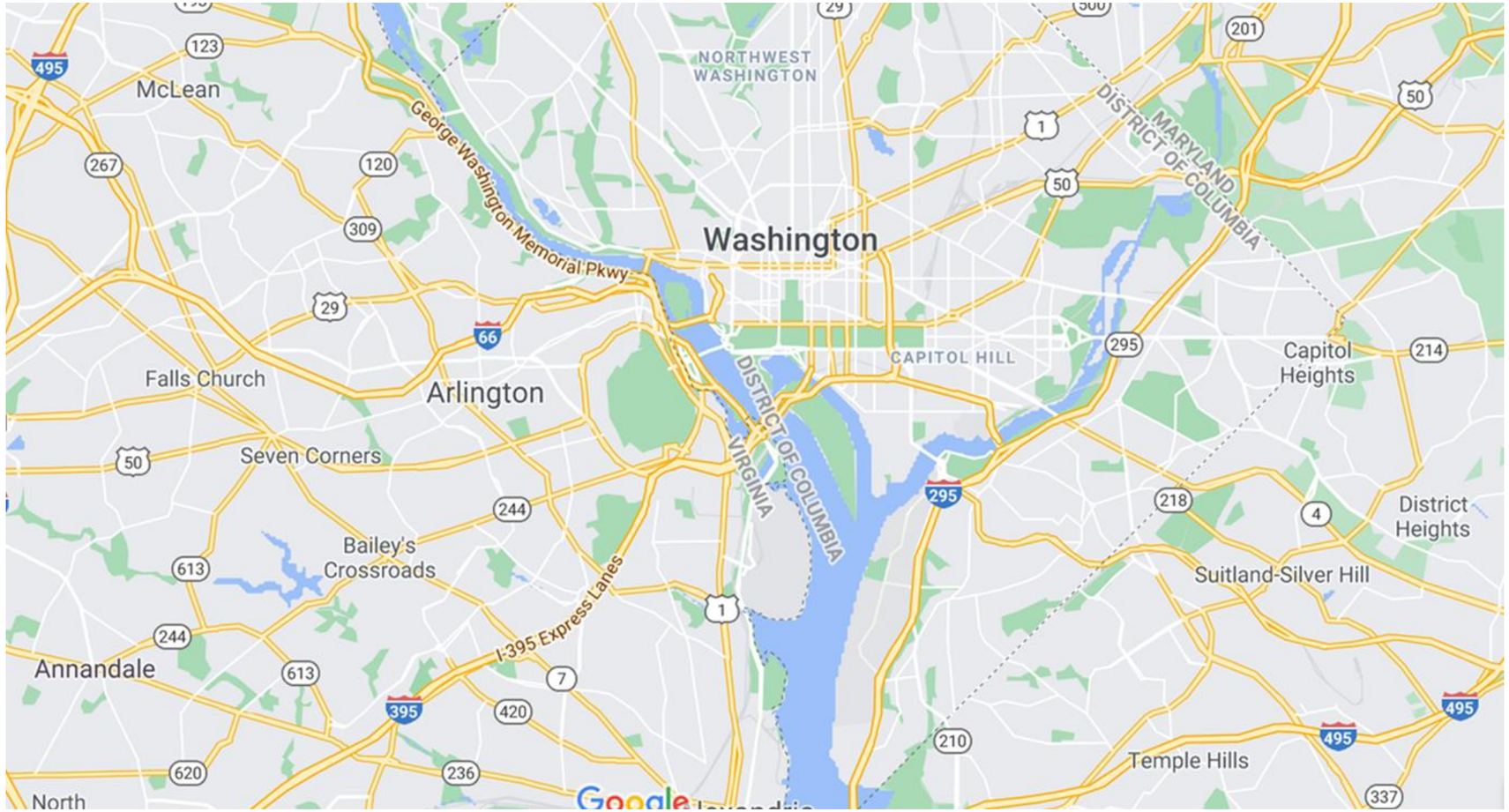
Chasing Your Tail With A Raspberry Pi

Matt Edmondson

Why Are We Talking About This?

- For some people, trying to figure out if they're being followed is a matter of physical safety for themselves or others
- I was approached by a friend in this situation looking for a technical option to use in conjunction with traditional tradecraft

The New (i.e. Really Old) SDR: Surveillance Detection Route



One Option: Looking for Persistent Digital Signatures

- Go to Starbucks to grab a drink
- Hit the gas station to top off the tank
- Head over to the bookstore to look at magazines
- Now… Did I see any devices at all three locations?

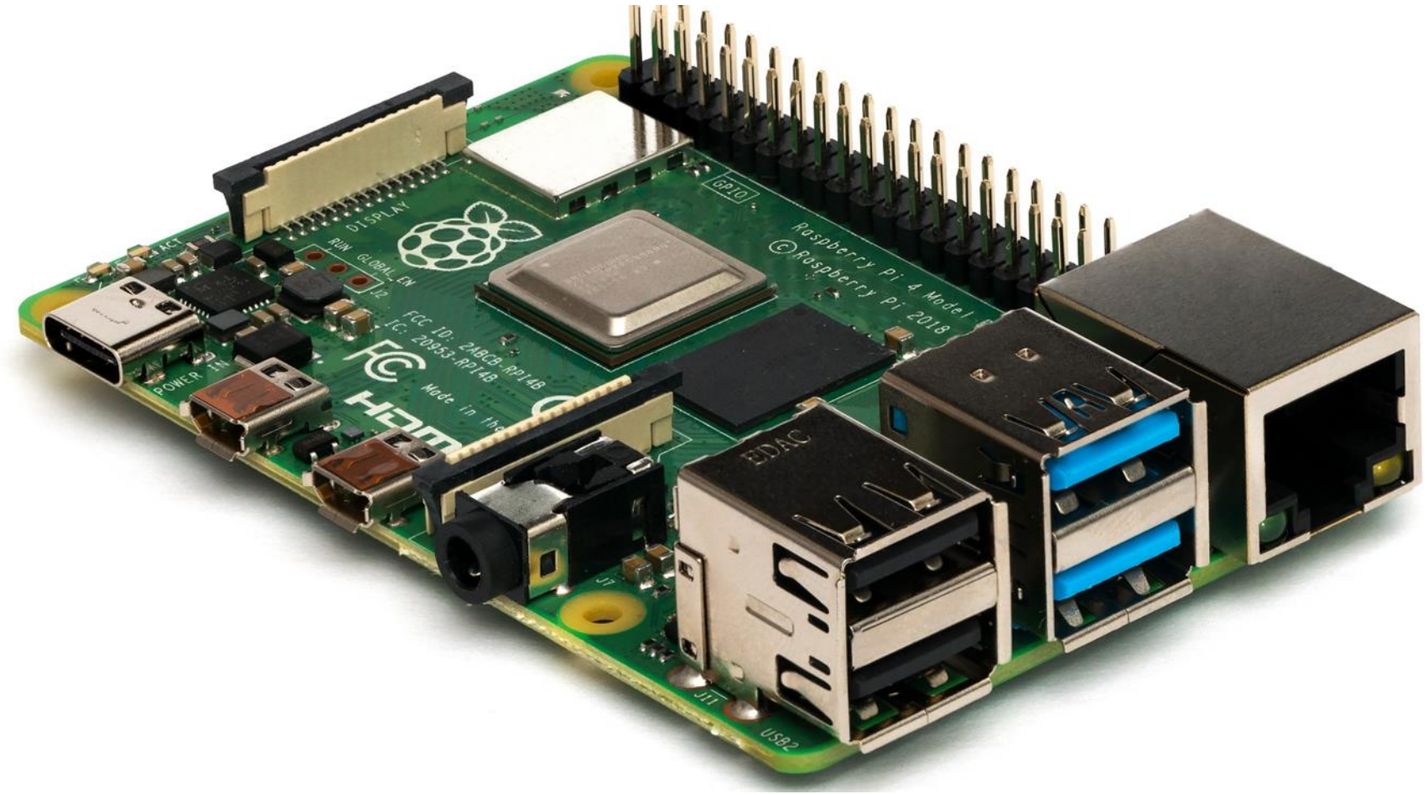
One Option: Looking for Persistent Digital Signatures Cont.

- We'll accomplish this by passively detecting Wi-Fi and Bluetooth devices we observe around us
- The devices could be part of an active connection, or just looking for networks to connect to

Hardware

- For hardware I wanted to use things that I had laying around in a closet
- I imagine many of you have these same things gathering dust somewhere and if not, they're cheap

Platform: Raspberry Pi is One Cheap Option



Wireless: Something You Can Put In Monitor Mode

Power: A Battery Pack or Other Power Source

Display: A Screen For Real Time Monitoring

Software: Kismet

- Open source with a fantastic Discord server
- Easy to install
- Passive
- Supports Wi-Fi, Bluetooth, SDR, ZigBee etc.
- Writes the data into a SQLite database
- Able to generate PCAP and other formats if needed

Everything Else is Just Python or Shell Scripts

```
'{} MACS added to the within the past 5 mins list".format(len(past_fi
 write ("{} MACS added to the within the past 5 mins list \n".format(
itialize macs five to ten minutes ago

_fetch_5_to_10(con):
sorObj = con.cursor()
sorObj.execute("SELECT devmac FROM devices WHERE last_time <= {} AND
s = cursorObj.fetchall()
 row in rows:
 #print(row)
 stripped_val = str(row).replace("(","").replace(")","").replace("'",

 if stripped_val in ignore_list:
     pass
 else:
     #print ("new one!")
     five_ten_min_ago_macs.append(stripped_val)
```

But It's Not Perfect!

- "Perfect is the enemy of good" – Voltaire
- "Give them the third best to go on with; the second best comes too late, the best never comes." - Robert Watson-Watt

Back to Kismet

- By default Kismet starts up a new SQLite file every time you start it (with a .kismet extension)
- We'll just have that output to the same directory and have our Python code parse whatever the newest .kismet file in that directory is

A Slight Change in the Methodology

- Look for any Wi-Fi or Bluetooth devices currently in the area which I also saw 5-10 minutes ago, 10-15 minutes ago or 15-20 minutes ago
- If there are, alert me with the MAC address, device type (Wi-Fi Access point, BTLE etc.) and the time frame I previously saw it

Our First Snag

- Not solvable with a simple SQL query by time frame since Kismet stores the first and most recent time stamps, but not those in between
- But there is a solution…

Enter the List!

- Create 5-10, 10-15 and 15-20 min lists at startup
- Initialize a list of current devices
- Every minute add things you're seeing into the current list
- Every five minutes, make the current devices the 5-10 mins list

Now What?

- Every minute grab the devices that have been seen in the previous 60 seconds, see if any of them are in any of the 5-10 min,10-15 min or 15-20 min lists and print an alert to the screen if so

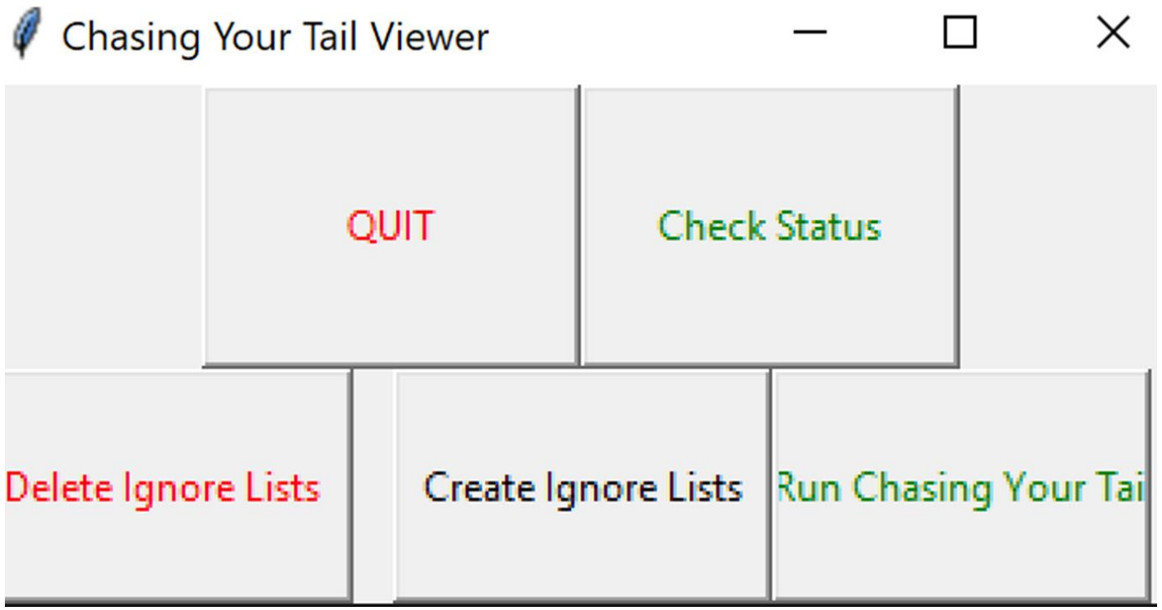
We Don’t Want to Alert on Ourselves…

Let's Make Some Ignore Lists

- At any time you can create an "ignore list" of every MAC address seen in the latest kismet database
- These devices are ignored for the rest of the session
- You can delete or re-recreate the list at any time

Please Don’t Laugh… OK, You Can Laugh

- How do you do all this on a tiny touch screen with no keyboard??



Field Testing

- Worked great in the lab, did everything it was supposed to
- Took it out in the middle of nowhere, turned on a new wireless device and saw…. nothing

MAC Randomization Was WAY More Frequent Than I Thought

Time	Source	Destination	Protocol	Length	Info
1591.471979	3a:48:87:f9:49:d6	Broadcast	802.11	118	Probe Request, SN=2758, FN=0, Flags=........C, SSID=DF
1687.563292	76:f0:92:2e:ed:8b	Broadcast	802.11	146	Probe Request, SN=3435, FN=0, Flags=........C, SSID=DF
1687.941978	96:cd:84:67:4d:3b	Broadcast	802.11	146	Probe Request, SN=1948, FN=0, Flags=........C, SSID=DF
1687.867810	fa:8b:8a:c0:b1:f5	Broadcast	802.11	146	Probe Request, SN=3258, FN=0, Flags=........C, SSID=DF
1730.920716	32:ce:e1:8d:11:9e	Broadcast	802.11	118	Probe Request, SN=2601, FN=0, Flags=........C, SSID=DF
1730.926789	da:ab:73:26:63:ac	Broadcast	802.11	118	Probe Request, SN=2182, FN=0, Flags=........C, SSID=DF
1731.524895	c2:04:f9:50:16:52	Broadcast	802.11	118	Probe Request, SN=3731, FN=0, Flags=........C, SSID=DF
1848.958200	22:9c:ba:45:66:a4	Broadcast	802.11	146	Probe Request, SN=1717, FN=0, Flags=........C, SSID=DF
1849.181210	2a:97:14:a8:bf:52	Broadcast	802.11	146	Probe Request, SN=1554, FN=0, Flags=........C, SSID=DF
1849.470901	ea:4d:a5:a7:38:59	Broadcast	802.11	146	Probe Request, SN=1846, FN=0, Flags=........C, SSID=DF
1849.251121	b2:e5:89:86:8e:af	Broadcast	802.11	146	Probe Request, SN=3871, FN=0, Flags=........C, SSID=DF
1849.491647	ea:4d:a5:a7:38:59	Broadcast	802.11	146	Probe Request, SN=1041, FN=0, Flags=........C, SSID=DF
1856.768172	d2:06:7d:ca:9d:84	Broadcast	802.11	146	Probe Request, SN=3439, FN=0, Flags=........C, SSID=DF
1857.277746	66:f8:ba:15:5d:28	Broadcast	802.11	146	Probe Request, SN=1626, FN=0, Flags=........C, SSID=DF
1870.587798	5a:7b:a6:cf:20:ec	Broadcast	802.11	146	Probe Request, SN=3242, FN=0, Flags=........C, SSID=DF
1870.567386	5a:7b:a6:cf:20:ec	Broadcast	802.11	146	Probe Request, SN=1536, FN=0, Flags=........C, SSID=DF

Is This a Deal Breaker?

- The MAC address logic works and needs to remain, but we obviously can't rely on it alone if we're counting on seeing devices that currently aren't connected to a Wi-Fi network when they're looking for Wi-Fi networks
- We're going to have to dive deeper and look at what was being probed for

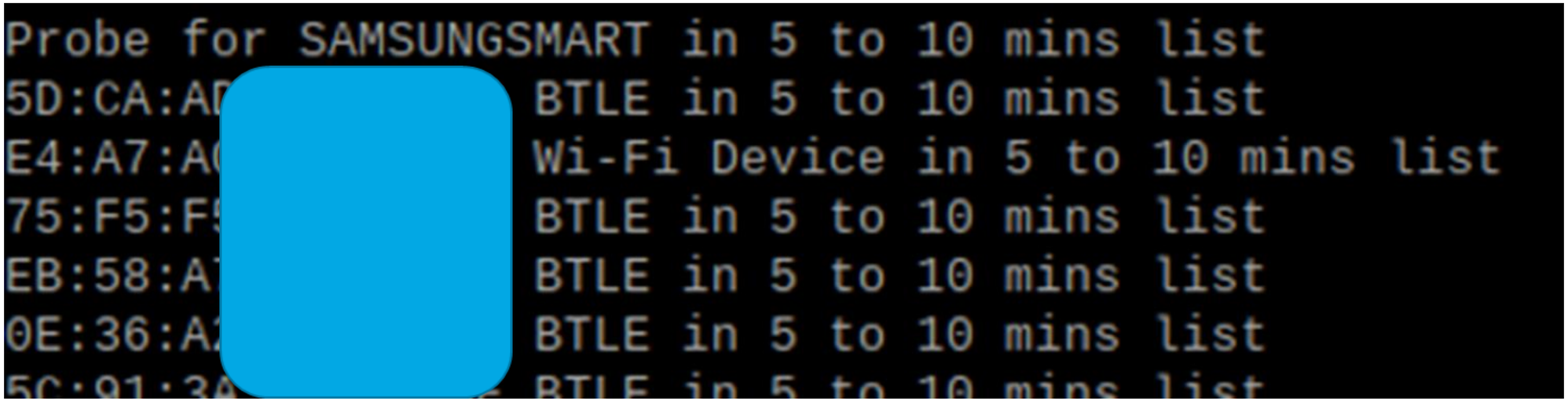
Thank You Kismet!

- Kismet normalizes some fields in the database but has a lot more data in JSON blobs
- raw_device_json["dot11.device"]["dot11.device.last_probed_ssid_record"]["dot11.probedssid.ssid"]
- So now in addition to MACs, we can look for what was being probed for, regardless of the MAC address probing

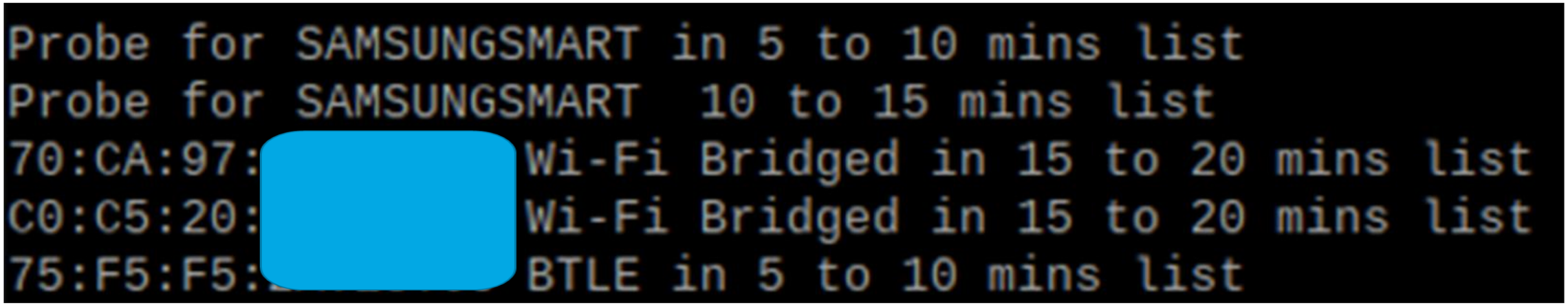
User Display

```
Current Time: 2021-12-02 12:58:13
308 MACs added to ignore list.
27 Probed SSIDs added to ignore list.
Pulling data from: /home/pi/kismet_logs/Kismet-20211202-19-38-08-1.kismet
5 MACS added to the within the past 5 mins list
0 MACS added to the 5 to 10 mins ago list
0 MACS added to the 10 to 15 mins ago list
0 MACS added to the 15 to 20 mins ago list
0 Probed SSIDs added to the within the past 5 minutes list
0 Probed SSIDs added to the 5 to 10 mins ago list
0 Probed SSIDs added to the 10 to 15 mins ago list
0 Probed SSIDs added to the 15 to 20 mins ago list
```

User Display Cont



User Display Cont



The MVP Version

The.... Better...ish... version

We Said Earlier That Kismet Keeps Great Logs

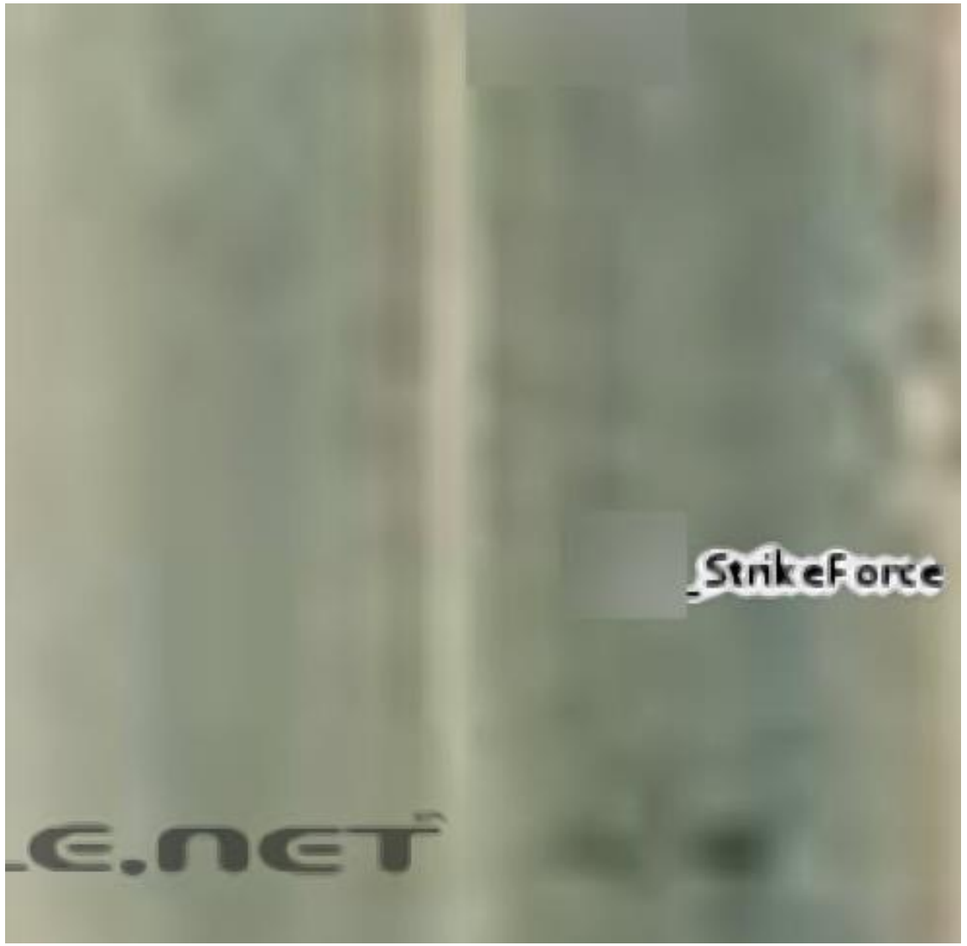
- Can we analyze these logs to find information about the individuals following us?
- Potentially…

Sub-Optimal OPSEC

- Many organizations have a very bad habit of naming their Wi-Fi networks the name of their agencies or specialty units

Wigle.net

- I did this government agency the courtesy of sanitizing the first part of their Wi-Fi network name



Path Forward

- More Wi-Fi adapters
- More wireless protocols
- GPS tracking

Special Thanks

- Mike Kershaw (@kismetwireless)
- Dominic White (@singe)
- Joshua Wright (@joswr1ght)

Thank You!

Matt Edmondson

@matt0177

matt@argeliuslabs.com

www.digitalforensicstips.com