RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: HTA-F02

EXFILTRATING DATA THROUGH IOT

Mike Raggo

CSO & Threat Research
802 Secure, Inc.
@DataHiding @MikeRaggo

Chet Hosmer

Founder
Python Forensics, Inc. *A Non Profit Research Institute*
@PythonForensics

Speaker Introduction

Mike Raggo

Mike Raggo is Chief Security Officer at 802 Secure and has over 20 years of security research experience. His current focus is wireless IoT threats impacting the enterprise. Michael is the author of "Mobile Data Loss: Threats & Countermeasures" and "Data Hiding" for Syngress Books, and contributing author for "Information Security the Complete Reference 2nd Edition".

A former security trainer, Michael has briefed international defense agencies including the FBI and Pentagon, and is a frequent presenter at security conferences, including Black Hat, DEF CON, Gartner, DoD Cyber Crime, OWASP, HackCon, and SANS.

Chet Hosmer

Is an international author, educator & researcher, and founder of Python Forensics, Inc., a non-profit research institute focused on the collaborative development of open source investigative technologies using the Python programming language.

- A Visiting Professor at Utica College in the Cybersecurity Graduate Program, where his research and teaching is focused on data hiding, active cyber defense and security of industrial control systems.

- Adjunct Professor at Champlain College in the Digital Forensics Graduate Program, where his research and teaching is focused on solving hard digital investigation problems using the Python programming language.

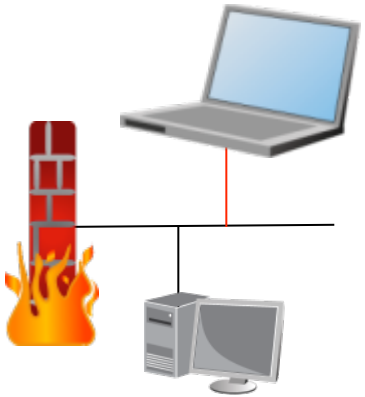
“Apply” Slide

- Our role is to provide insights based on our research/analysis of data exfiltration vulnerabilities found in IoT protocols (i.e. SSDP, P25, Zigbee, Z-Wave, WiFi, uPnP). With an eye toward mitigating weaknesses in current protocols and to impact future protocol designs to eliminate them.
- From a student perspective we hope to first make you aware of these vulnerabilities and weaknesses. Then more specifically delve into the details and demonstrate data exfiltration using IoT protocols.
- The application of this knowledge will allow you to assess and mitigate these risks as you integrate IoT technologies into your production systems, as well as making informed decisions regarding IoT device and protocol selection.

What's really different about IoT?

IoT is more than smart devices

Detecting Exfiltration on the Wire - Old School

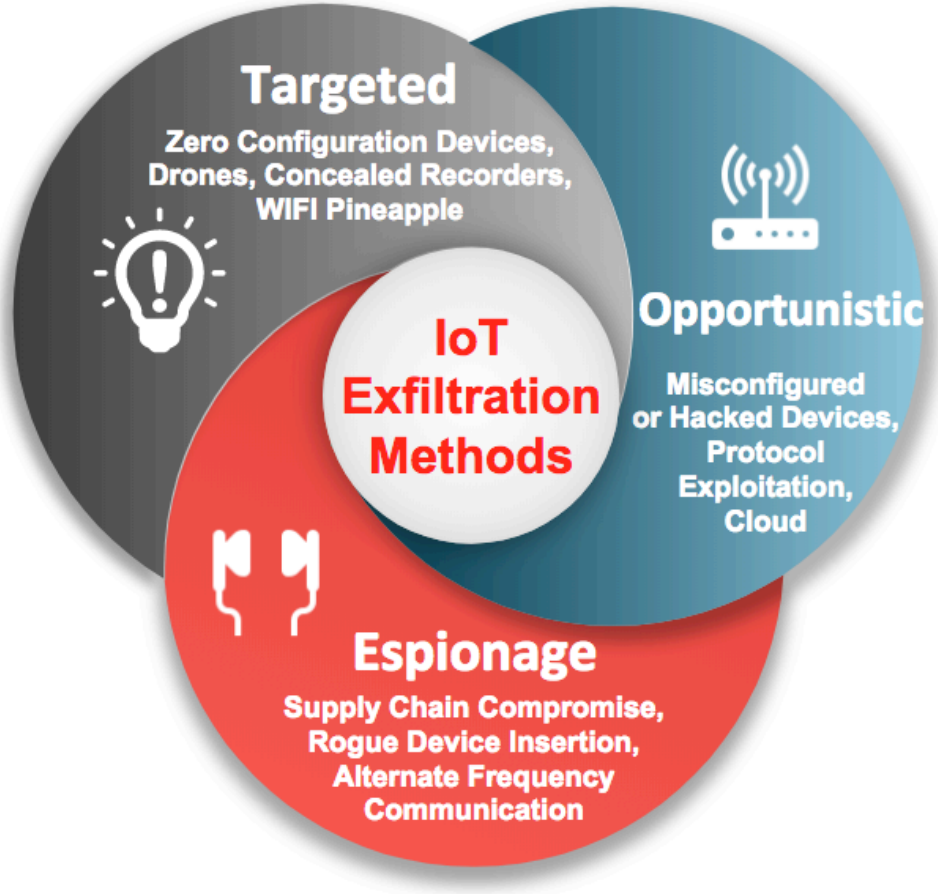


Detecting Exfiltration in an IoT World

IoT exfiltration is easy - No one is monitoring!

802secure

Exfiltration Methods



Zero Configuration Devices

Are obscured from normal network operations - can operate autonomously outside the scope of the enterprise network.

Protocol Limitations

Many IoT protocols lack even basic authentication, integrity and privacy considerations.

Supply Chain Integrity

Data is back channeled to remote country or close proximity listening stations. Communication is obfuscated by the cloud or the use of alternate protocols and frequencies.

Smart Watch Exfiltration

Tested 4 Smartwatches
Apple Watch, Samsung Gear 2 Neo. Moto 360, U8

Samsung
Tizen

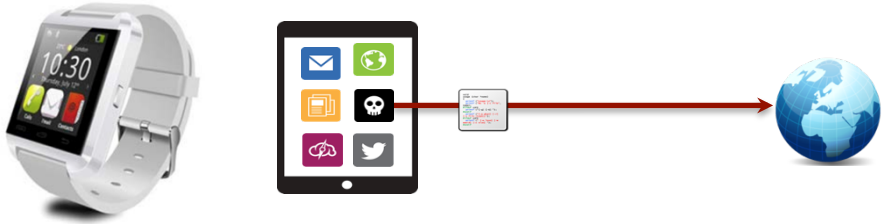
Apple
watchOS

U8 Nucleus

Android Wear
(Moto 360)

Smart Watch Exfiltration

- U8 Nucleus Smart Watch found to be sending data through the app on the mobile device to a random IP in China, over an encrypted channel



- Samsung Gear 2 Neo found with no password and allowed remote privilege escalation (disclosed to Samsung and now patched)
- HackCon Norway, BSidesSF, DEF CON Demo Lab demonstrated SWATtack - python tool for exploiting smartwatches

IoT Hubs and Exfiltration

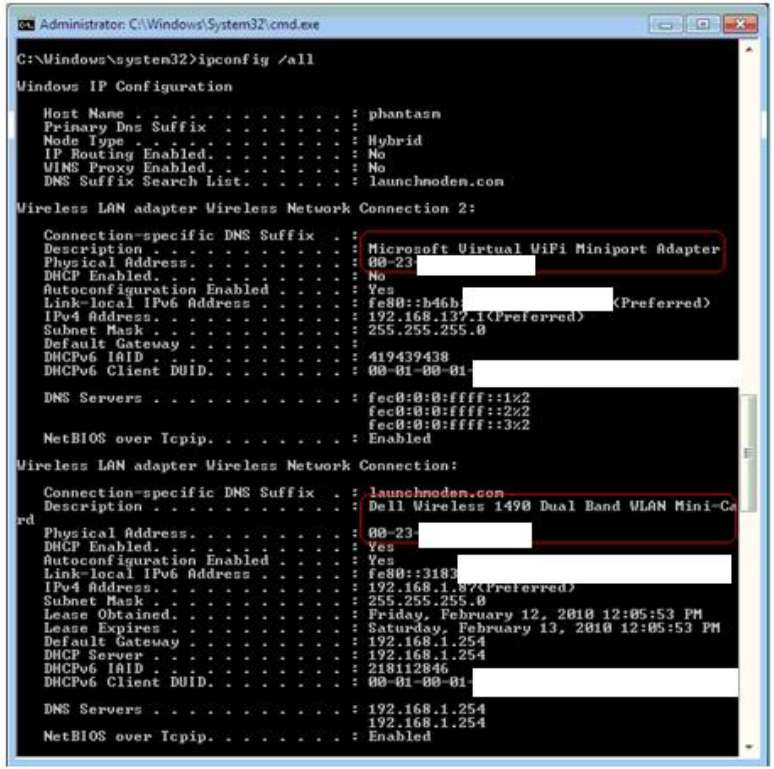
- Many IoT devices and IoT Hubs now have USB ports for data backup
- New USB backup flash drives support Wireless (WiFi, BT, etc.)

Z-Wave Hub with
Ethernet and USB Ports

EXFIL - Steal and exfiltrate surveillance files, data, videos, etc. Nothing seen on the corporate WiFi or Wired Network

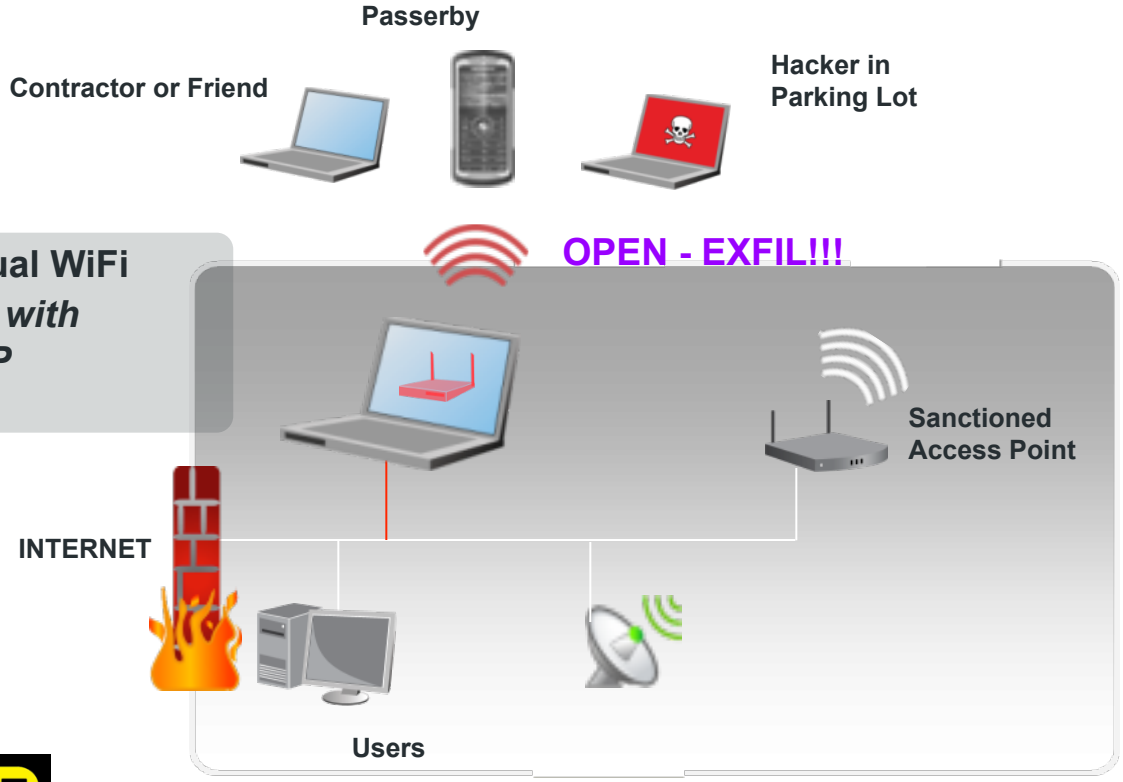
Windows Virtual WiFi (7, 8, & 10)

- This is native to Windows operating system. In all versions of Windows 7, 8, & 10
- Setup at the DOS Prompt
- Share either a Wired or Wireless connection
  The user can share their own desktop ***(like SoftAP, not ad-hoc network)***
- And the user can share their network connection with others - USING THE SAME WiFi CARD!
- Corporate wireless network may use authentication and encryption
  - BUT the user can share that connection with others, allowing those users to connect to the corporate network with weaker authentication & encryption, or OPEN!!!

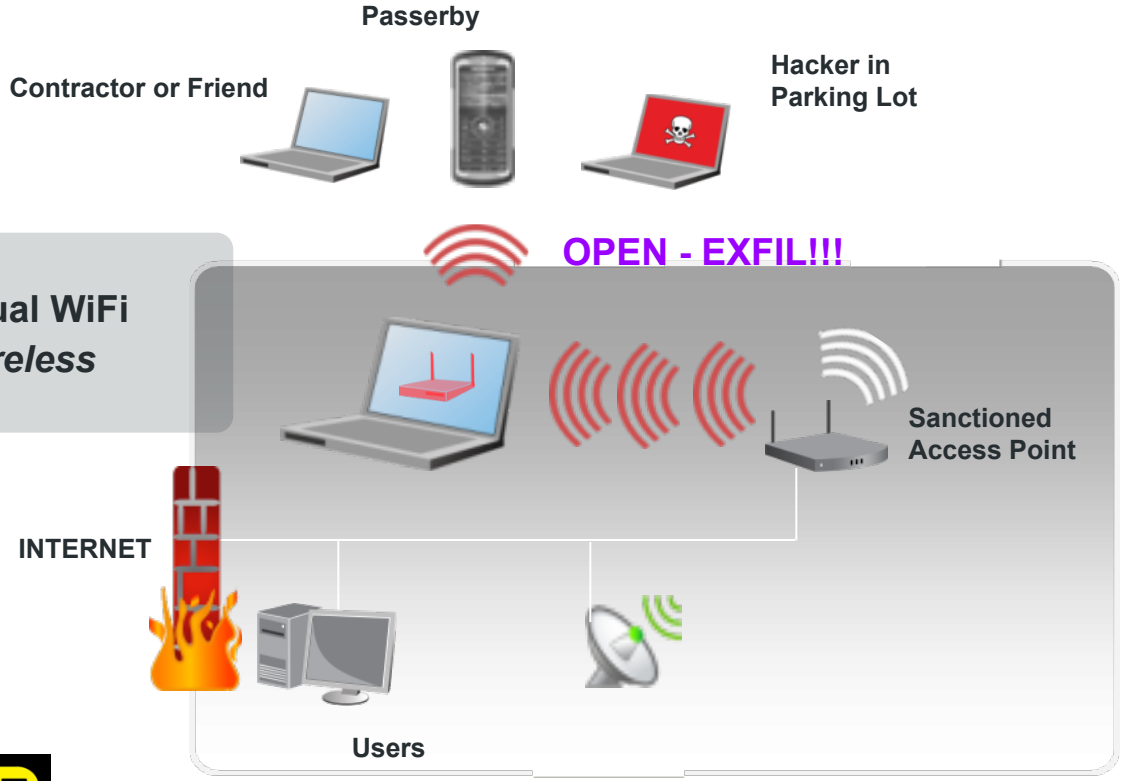


Wireless Rogues on Network - Virtual WiFi

Windows 10 Virtual WiFi
Approved laptop with built-in Rogue AP (hidden)



Wireless Rogues on Network - Virtual WiFi



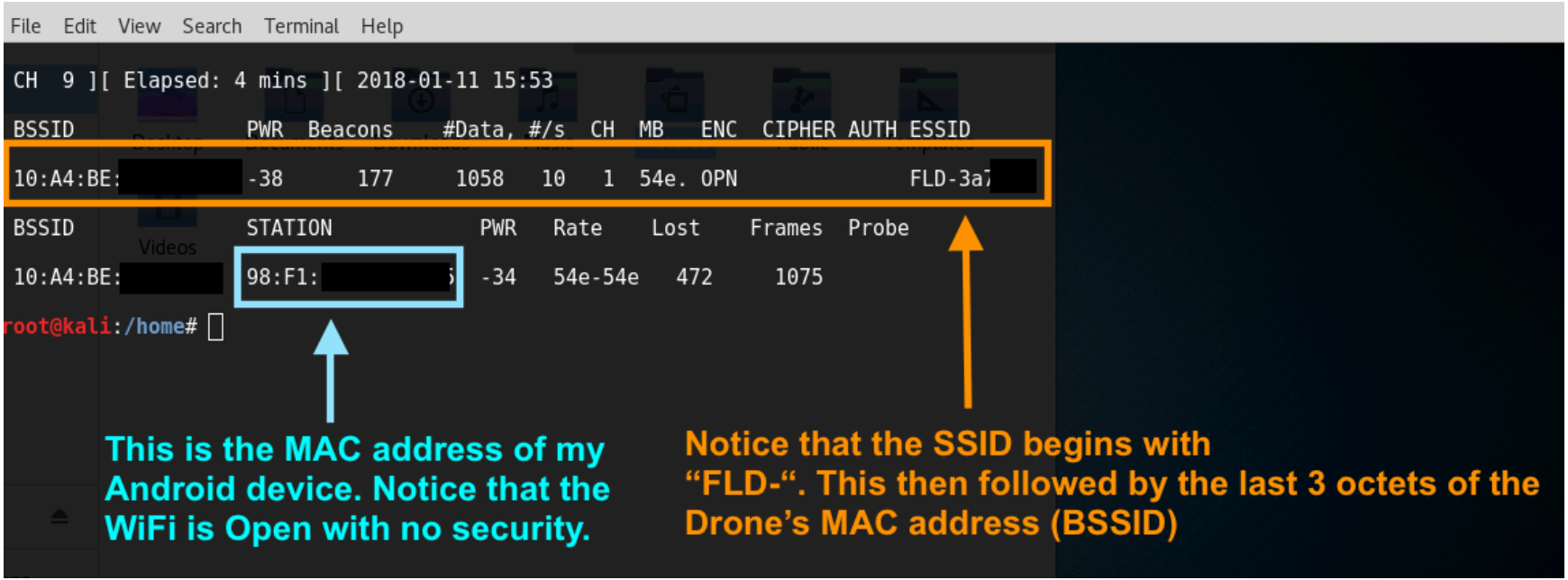
Nearby IoT Threats - Drones

- Video and Audio Surveillance, Wireless surveillance
- Drop cellphones, pathogens, battery operated spy cameras
- Consumer drones pair via WiFi (Virtual AP) with smartphones and tablets
- Most organizations blind to threat (1-2 per day, 3-7 per week)

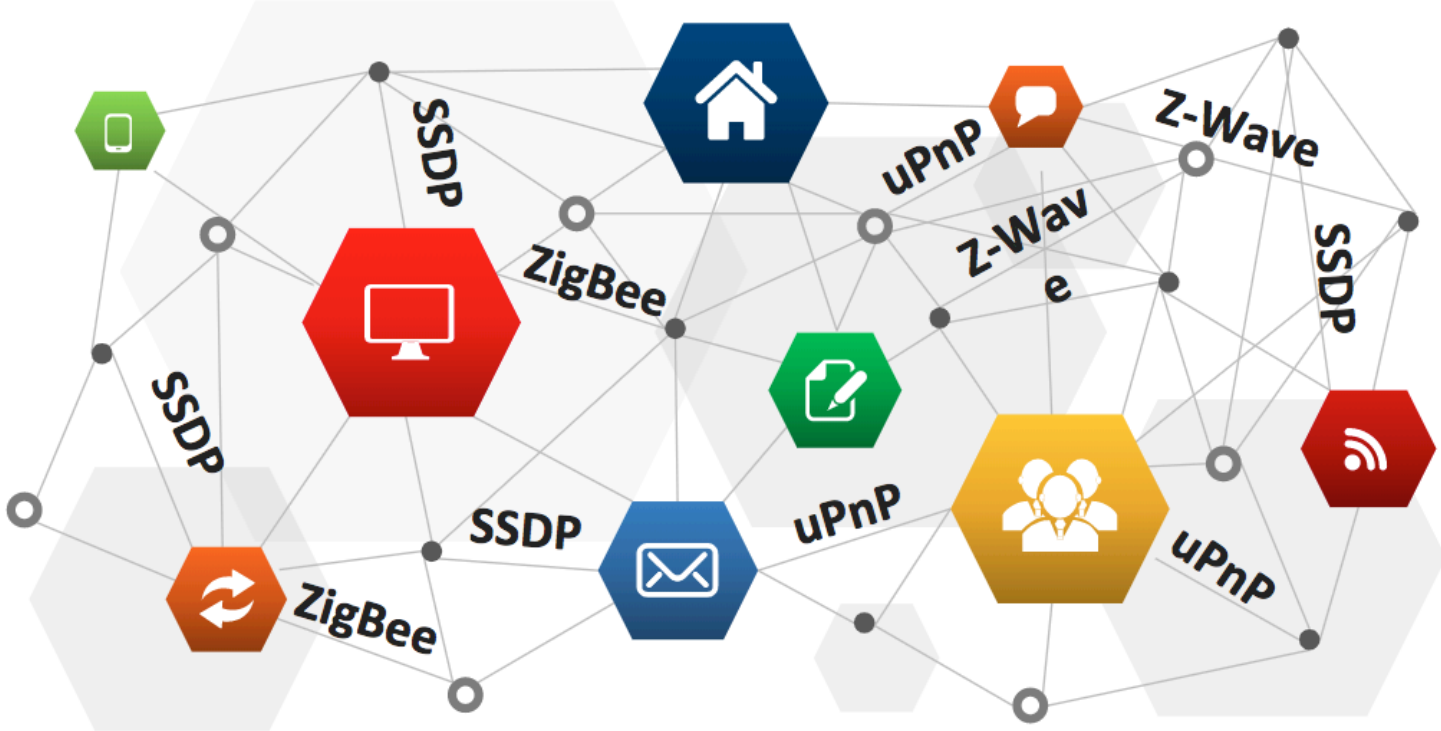
Drones

Drones

- Forensics/Detection - who, what, when, where, how



IoT Protocol Exfiltration



Exploiting Lack of Integrity in IoT Protocols

- What - Many devices support UPnP to allow an app or other devices to discover other devices (M2M)
- Sends multicast packets broadcasted to local network
- SSDP UPNP - Simple Service Discovery Protocol (Part of Universal Plug and Play)
  - M-SEARCH - Discover packet sent by app or another device
  - NOTIFY - Device announces itself on the network, routinely, and also when it leaves

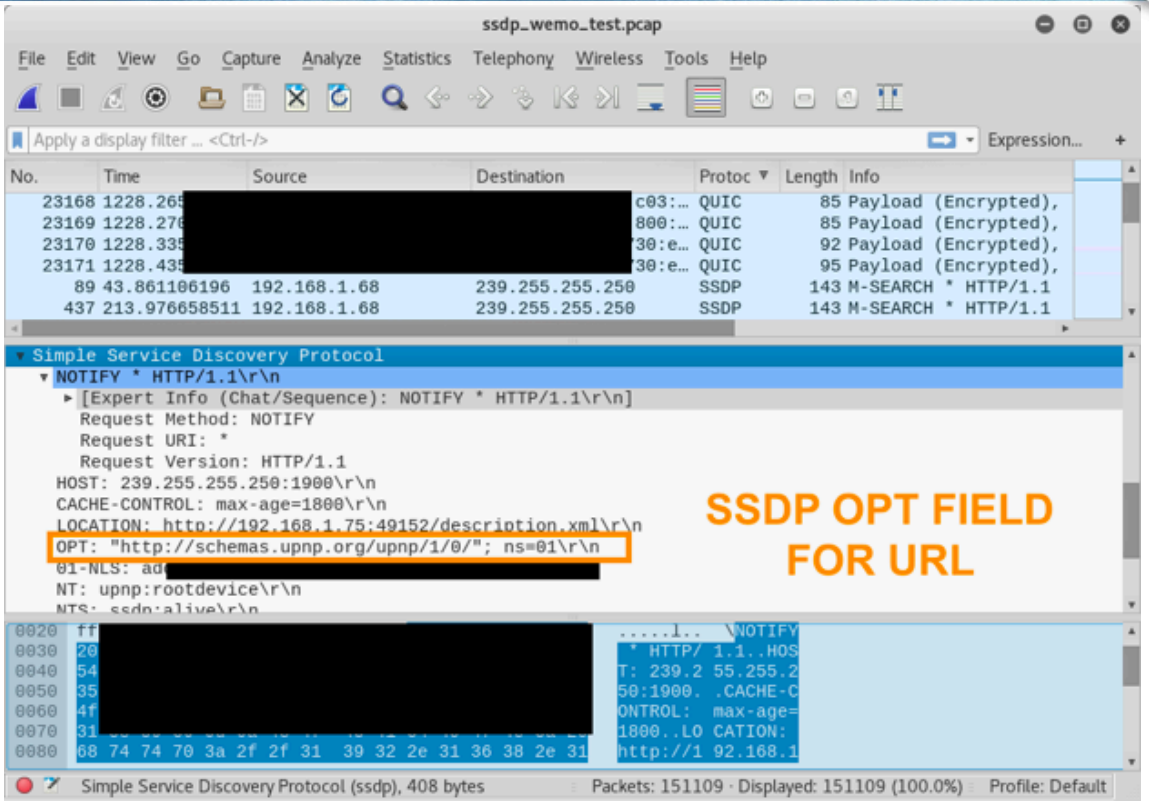
Exploitation of SSDP - ULA OPT Field

- ULA OPT FIELD
  - Unique Local Addresses - Site-Routable
  - Used in NOTIFY and M-SEARCH messages
  - For use in IPv4 and IPv6 (for backward compatibility)

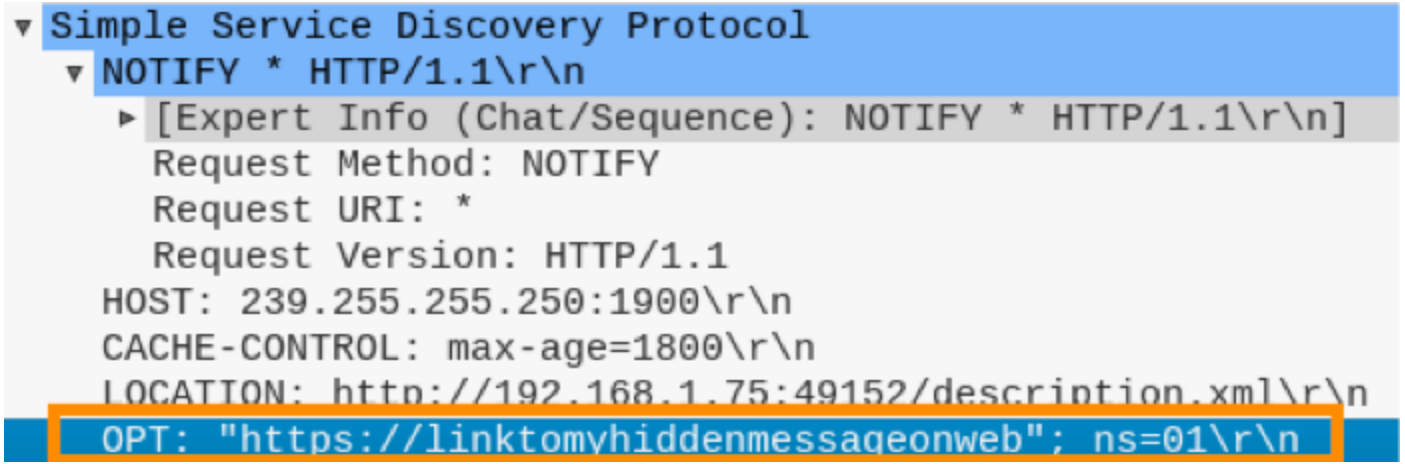
Reference: http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1-AnnexA.pdf

UDP - Exploitation of SSDP

Our Target



UDP - Exploitation of SSDP



- Modify SSDP OPT Field with Hidden Message, URL, etc.
- Covert communications, dead drop, malware callback to CnC for updates, etc.

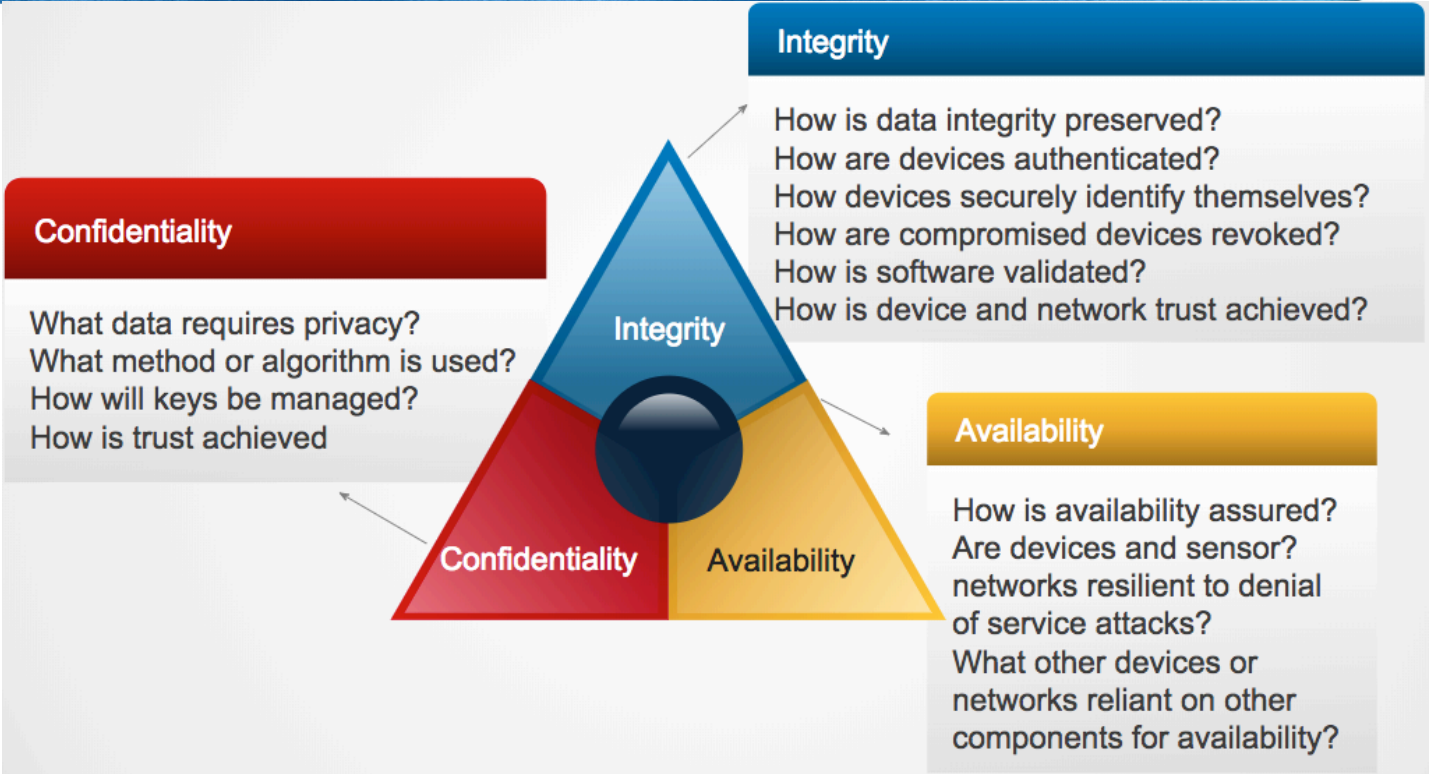
Covert UDP - SSDP

1. Smart Plug sends M-Search
2. M-Search packet embeds hidden message/content or CnC URL in OPT field
3. Received by other IoT devices on network

Two-way conversations using SSDP to hide content (M2M)

4. Smart TV receives M-Search packet and responds
5. NOTIFY packets send a packet back with embedded response

IoT Device Critical Considerations



Exfiltration Demo

Exfiltration Case Study and Demo

- Exfiltration Example
- Typical Broadcast Message

```
broadCastMsg = \
 'M-SEARCH * HTTP/1.1\r\n' \
 'HOST:192.168.86.115:1900\r\n' \
 'ST:upnp:rootdevice\r\n' \
 'MX:2\r\n' \
 'MAN:"ssdp:discover"\r\n' \
 '\r\n'
```

Exfiltration Case Study and Demo

- Exfiltration Example
- Typical Broadcast Message
- Broadcast Message Data Appending Example - Simple

```
broadCastMsg = \
    'M-SEARCH * HTTP/1.1\r\n' \
    'HOST:192.168.86.115:1900\r\n' \
    'ST:upnp:rootdevice\r\n' \
    'MX:2\r\n' \
    'MAN:"ssdp:discover"\r\n' \
    'Hello World\r\n'
```

Plain-Text Insertion

Exfiltration Case Study and Demo

- Exfiltration Example
- Typical Broadcast Message
- Broadcast Message Data Appending Example - Obfuscated

```
broadCastMsg = \
    'M-SEARCH * HTTP/1.1\r\n' \
    'HOST:192.168.86.115:1900\r\n' \
    'ST:upnp:rootdevice\r\n' \
    'MX:2\r\n' \
    'MAN:"ssdp:discover"\r\n' \
    ‘894629\r\n'
```

Index to pre-exchanged lookup table

Python Script to Exfiltrate Data using SSDP

Establish a connectionless UDP Socket → Send Broadcast Message to 239.255.255.255/1900 → Wait for Responses

```python
import socket

broadCastMsg = \
    'M-SEARCH * HTTP/1.1\r\n' \
    'HOST:239.255.255.250:1900\r\n' \
    'ST:upnp:rootdevice\r\n' \
    'MX:2\r\n' \
    'MAN:"ssdp:discover"\r\n' \
    'Hello World\r\n'

# Set up socket to receive IoT responses
udpSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
udpSocket.settimeout(30)
udpSocket.sendto(broadCastMsg, ('239.255.255.250', 1900) )

try:
    while True:
        data, address = udpSocket.recvfrom(65535)
        print "Data:    ", data
        print "Address: ", address
except socket.timeout:
    pass
```

```
================================================================================
Data:     HTTP/1.1 200 OK
ST: upnp:rootdevice
EXT:
USN: uuid:786fad77-04b4-4845-97f4-602b8885aeea::upnp:rootdevice
SERVER: Reciva UPnP/1.0 Radio/1.0 DLNADOC/1.50
CACHE-CONTROL: max-age=1800
LOCATION: http://192.168.86.41:8050/786fad77-04b4-4845-97f4-602b8885aeea/description.xml


Address:  ('192.168.86.41', 1900)
================================================================================
```

Response from Bose Wave Radio

Prescription for better non-Exfil Hygiene

- Monitor for **anomalous network behaviors - wired AND wireless!!!**
  - Strange outbound destinations, and from what IoT source internally
  - Detection of odd (non-WiFi) frequencies and protocols (BT, Zigbee, etc.)
  - Out-of-band & off network WiFi connectivity
- Monitor for **neighboring IoT threats**
  - Drones, spy cameras, wireless attacks
- Monitor wired “and” wireless security postures across **Enterprise Network** and **IoT/IIoT autonomous network deployments**
  - Misconfigurations
  - State change of devices
  - Cloud connections and Outbound Data Storage from IoT devices

Thank You!

Michael Raggo
@MikeRaggo
www.802secure.com

Chet Hosmer
@PythonForensics
www.python-forensics.org