RADIO EXPLOITATION 101

MATT KNIGHT // MARC NEWLIN // BASTILLE NETWORKS

CHARACTERIZING//CONTEXTUALIZING//CLASSIFYING RF ATTACKS
WHO ARE THESE GUYS

▶ Matt Knight
  ▶ Software Engineer and Security Researcher @ Bastille
  ▶ Reverse engineered the LoRa wireless protocol in 2016
  ▶ BE & BA from Dartmouth

▶ Marc Newlin
  ▶ Security Researcher @ Bastille
  ▶ Discovered Mousejack vulnerability in 2016
  ▶ Finished 3rd in DARPA Spectrum Challenge in 2012
  ▶ Finished 2nd in DARPA Shredder Challenge in 2010
WIRELESS EXPLOITATION METHODS
WIRELESS VS. WIRED
DO IT LIVE (DEMO)
AGENDA

1. **Evolution** wired and wireless network attacks

2. RF technical overview

3. **Methods of** Wireless Exploitation
   - Techniques, impact, and defenses
   - **Analogues** to wired networks
   - Examples and **demos** (do it live!)

4. How to **apply** this information
HISTORICAL BACKGROUND // CONTEXT

EVOLUTION OF NETWORK SECURITY
Packet sniffing in the 1990s
$8,000+ (in 1990s dollars)

**NETWORK GENERAL PACKET SNIFTER**

Installed on a Dolch lunchbox computer
Packet sniffing in 1998
Wireless* sniffing in the 2000s*non-802.11
EARLY SDRS

>>$100K
Wireless sniffing in 2012
RTL 2832 USB STICK

$8

(not pictured: promiscuous mode driver)
Wireless sniffing in 2017
$8 -> $1150

ALL THE SDRS
ON PROTOCOLS...
Key protocols in the 1990s
802.3
802.5
Key protocols in 2017
WIRELESS IN 2017

- 802.11 is just one piece of the puzzle
- Explosion of IoT and Mobile means...
  - There’s a PHY for every use case
  - Licensed and unlicensed spectrum
  - Embedded systems require compromises

**EMBEDDED TRADEOFFS**

- Battery powered
- Limited user interaction
- Lack of crypto
- Unsuitable pipes for firmware updates

- Performance, UX, cost, and delivery are more important than best practices
Historical reliance on SECURITY THROUGH OBSURITY means...
[PIÑATAS]
SO WHAT DOES IT TAKE...
FIRST
ENDPOINTS

[ SYSTEMS, DATA AT REST ]
NETWORKS

[ DATA IN MOTION ]
IP NETWORK SNIFFING IS EASY

- Interfacing with an IP network is trivial
- Hardware NICs, monitor mode
- Known Layer 2 // MAC frame protocols
- 802.3 // Ethernet for wired IP traffic
- 802.11 // Wi-Fi for wireless IP traffic
WIRELESS* NETWORK SNIFFING IS HARD

- Network interface is **totally non-trivial**
- Your Wi-Fi NIC can’t sniff wireless traffic from your home security system
- **Arbitrary Layer 1 // PHYs**
  - There are many ways to make a PHY
  - **802.11 // Wi-Fi is just one example**
SOFTWARE DEFINED RADIO (SDR)

- Board with flexible wideband RF frontend
  - Captures raw radio spectrum
  - Shuttles RF I/Q samples to DSP or host
- Implement arbitrary PHYs in:
  - Software
  - FPGA HDL
SDR DOWNSIDES

- Requires:
  - Lots of computing power, power hungry
  - Complex and esoteric domain knowledge
- Left: RTL for one of Matt’s 802.15.4 decoders

- Want to learn about SDR and PHYs? Check out our “So You Want to Hack Radios” series from Shmoocon, Troopers, and HITB2017AMS
OPEN SOURCE SOFTWARE RADIO

- Vibrant open-source SDR community
- GNU Radio
  - Modular signal processing framework
  - Abstracts away hard math
- Ossmann and Balint’s video tutorials
- Wireless Village here at DEF CON
GROSSLY SIMPLIFIED

RF CONCEPTS
PHYSICAL LAYER (PHY)

- Lowest layer in communication stack
- In wired protocols: voltage, timing, and wiring defining 1s and 0s
- In wireless: patterns of energy being sent over RF medium
Spectrogram
a.k.a. “waterfall”

Time

Frequency

Power (z-axis)
MANIPULATING RF

- ... is done with a radio
- Hardware defined
- Software defined radio (SDR)
PHY COMPONENTS

- Modulation
  - How digital values are mapped to RF energy
- RF parameters that can be modulated:
  - Amplitude
  - Frequency
  - Phase
  - some combination of the above
HOW TRANSMITTING WORKS

Layer 2 (MAC)

<table>
<thead>
<tr>
<th>MAC Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW Addresses</td>
</tr>
</tbody>
</table>

Layer 1 (PHY)

<table>
<thead>
<tr>
<th>PHY Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
</tr>
</tbody>
</table>

Modulation

Maps 1s and 0s to electrical phenomena

(to antenna)
HOW RECEIVING WORKS

Layer 2 (MAC)

<table>
<thead>
<tr>
<th>MAC Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW Addresses</td>
</tr>
<tr>
<td>Sequence Number</td>
</tr>
<tr>
<td>(other stuff)</td>
</tr>
<tr>
<td>Layer 3 Frame</td>
</tr>
</tbody>
</table>

Layer 1 (PHY)

PHY State Machine

- Wait for Preamble
- Look for SFD
- Check CRC
- Demodulate N bits
- (optional) Inspect PHY Header
- Present to Layer 2

(from antenna)
KEY CONCEPTS

- Radios are **state machines**
  - They process environmental inputs
  - They are **deterministic**: not magic

- RF is a **lossy analog medium**
  - Developers know this! Radios are resilient
  - Tons of contention

- Opportunities for **tactical abuse**
CLASSIFYING RF ATTACK METHODS

METHODS OF EXPLOITATION
CLASSIFYING RF ATTACK METHODS

- For each attack classification, we’ll show:
  1. Method: how the attack is performed
  2. Impact: what the attack enables
  3. Analogue: equivalent attack on wired/IP network, if one exists
  4. Example: A relevant example of this type of attack
  5. Proof: demo
RECONNAISSANCE // DATA LEAKAGE

SNIFFING
SNIFFING OVERVIEW

- Method
  - Capture traffic from nearby networks

- Impact
  - Data loss // leakage
  - Reconnaissance for further development

- Wired Analogue
  - None! Unique characteristic of RF
SNIFFING APPLIED

- Example
  - Key extraction during ZigBee lock pairing
  - Keystrokes from unencrypted keyboards

- Limitations
  - Physical range, availability of PHY interface

- Demo Scenario
SNIFFING DEMO

- HP Classic Wireless Desktop
- Unencrypted wireless keyboard
- Keystroke sniffing demo
- Proprietary 2.4 GHz wireless protocol
  - MOSART Semiconductor transceiver
  - No encryption
  - No official protocol documentation
  - Over-the-air compatible with nRF24L USB dongles
WARDRIVING OVERVIEW

- **Method**
  - **Scan** for identifying features on a protocol of interest
  - (Optional) **actively beacon** to induce traffic

- **Impact**
  - **Discovery** of exploitable devices or networks

- **Wired Analogue**
  - Port scanning, service discovery
WARDRIVING APPLIED

Example

- 802.11 AP discovery
- Beaconing for a ZigBee network coordinator

Limitations

- Physical range, number of channels, easy to spot if defender knows to look for it
WARDRIVING DEMO

- 802.15.4 wardriving
- Killerbee exploitation framework
  - Crafts broadcast beacon requests
- ApiMote hardware radio board
  - Sends out requests
  - Records responses from 802.15.4 network coordinators
DEVICE CONTROL // STATE MANIPULATION

REPLAY
REPLAY ATTACK OVERVIEW

- Method
  - Re-transmit a previously captured PHY frame

- Impact
  - Change the state of a device or something on the network

- Wired Analogue
  - Same (replays exists on wired networks too)
REPLAY ATTACKS APPLIED

- Example
  - 2017 Dallas tornado emergency siren attack

- Limitations
  - Defeated by freshness (sequence number) or authentication (cryptography/trust)

REPLAY ATTACK DEMO

- Fortress Security Safeguard panic button
- 433 MHz on-off keying
- No freshness or authentication
- Raw IQ replay OR decode/resynthesize

https://www.amazon.com/Fortress-Security-Safeguard-All-One/dp/B01K4OMDWW/ref=cm_cr_arp_d_product_top?ie=UTF8&th=1
DENIAL OF SERVICE // NETWORK STATE DISRUPTION

JAMMING
JAMMING OVERVIEW

- **Method**
  - Transmit noise or conflicting traffic within target network’s RF channel (same frequency)

- **Impact**
  - Blocks traffic on network
  - Network state disruption

- **Wired Analogue**
  - Denial of Service
JAMMING APPLIED

- Example
  - Home security system jamming
- Limitations
  - Jam detection mechanisms
  - Self-denial: difficult to simultaneously jam and monitor network traffic
JAMMING DEMO

- Home security system jamming

- 433 MHz on-off keying protocol
  - Transmit wideband noise at 433 MHz
  - Device jam detection mechanisms will detect after several seconds, so...
SMART JAMMING

EVADING DETECTION
DUTY CYCLED JAMMING

- Problem: Hardware radios implement “clear channel” detection features to avoid talking over other radios
  - Sampling CCA is a zero-cost jam detector
- Solution: pulse jammer on and off at appropriate rate to evade jam detection functions
- Examples: Matt’s done this to defeat 802.15.4 jam detection, but doesn’t know of any public examples
**REFLEXIVE JAMMING**

- Problem: Continuously jamming makes offensive network monitoring hard
- Jamming denies both the attacker and the defender
- Solution: detect beginning of frame and reflexively jam to target either specific packets or trailing checksums
- Examples: Samy Kamkar’s RollJam (left), reflexive jamming built into Killerbee//802.15.4 ApiMote
DENIAL OF SERVICE

LINK LAYER RESERVATION
LINK LAYER CONGESTION OVERVIEW

- **Method**
  - Abuse channel reservation features to reserve channel... forever
  - See 802.11’s virtual carrier sensing

- **Impact**
  - Denies legitimate Wi-Fi devices access to the RF channel
  - Wired Analogue
  - None (virtual carrier sense is unique to wireless protocols)

Left: 802.11 MAC header
Link Layer Congestion Applied

- Example
  - No recent examples of a virtual carrier sense abuse attack

- Limitations
  - Network allocation vector constraints
  - [802.11]: Max 32ms effect per malicious packet

$5 attack platform (ESP8266)
LINK LAYER CONGESTION DEMO

- 802.11 virtual carrier sense abuse
  - One-line Scapy script
- 802.11 frame with:
  - Empty payload
  - Maximum frame duration
- Prevents other 802.11 devices from communicating on the channel

*cheers to Bastian Bloessl for original research
NETWORK MEMBERSHIP // IMPERSONATION // ROUTING

EVIL TWIN
EVIL TWIN OVERVIEW

- **Method**
  - Impersonate a network participant by assuming its address and configuration
  - (Optional) actively deny the real thing

- **Impact**
  - Man in the Middle: Intercept and actively tamper with network traffic
  - Tamper with routing, if eligible

- **Wired Analogue**
  - ARP spoofing/cache poisoning

(evil facial hair)
EVIL TWIN APPLIED

- Example
  - WiFi Pineapple
  - IMSI Catchers // Stingrays

- Limitations
  - Defeated by authentication (cryptography/trust)
  - Attacker may need to deny legitimate twin
EVIL TWIN DEMO

- 2G GSM/GPRS IMSI catcher/man in the middle
- Presents as an arbitrary network
- Phones downgrade and connect if they believe it’s the best option
FIRMWARE UPDATES

PERSISTENCE // ADDED VALUE
FIRMWARE ATTACK OVERVIEW

- **Method**
  - Modify firmware image to your liking
  - Deliver it via OTA firmware update mechanism
- **Impact**
  - "Added value": modifying device behavior
  - Persistence
  - Denial of Service: see BrickerBot
  - Self-propagation, worm style
- **Wired Analogue**
  - Endpoint malware, worms, etc.
FIRMWARE ATTACKS APPLIED

- Example
  - Phillips Hue ZigBee Light Link worm
  - Cesar Cerrudo’s hypothetical traffic light sensor worm (also ZigBee)
- Limitations
  - Defeated by code signing, network encryption, etc.
FIRMWARE ATTACK DEMO

- No demo for this one
- Check out ZigBee lightbulb worm at http://iotworm.eyalro.net/
- Great paper and video
- Best part is firmware signing key recovery through a side channel
PHY LAYER PROTOCOL ABUSE
PHYSICAL LAYER ABUSE OVERVIEW

- **Method**
  - Chipsets implement PHY state machines differently – various degrees of error tolerance
  - Send transmissions that are **not strictly compliant** with a receiver’s PHY

- **Impact**
  - Targeted receiver evasion (IDS evasion)
  - Device fingerprinting

- **Wired Analogue**
  - Same (demonstrated on 802.3 chipsets)
  - Far more practical in RF domain

---

PHYSICAL LAYER ABUSE APPLIED

- Example
  - 802.15.4 receiver evasion

- Limitations
  - Network participants must be on different chipsets
  - Not all chipsets are vulnerable
PHYSICAL LAYER ABUSE DEMO

- Selectively evasive 802.15.4 packets
- Transmitter: ApiMote w/ CC2420
- Receivers: ApiMote w/ CC2420
  RZUSB stick w/ AT86RF230
- Both receivers receive everything, until they don’t :)
NON-EXHAUSTIVE

[OBVIOUSLY]
CONCLUSIONS
## WIRELESS ATTACK METHODS SUMMARY

<table>
<thead>
<tr>
<th>Method</th>
<th>Analogue</th>
<th>Complexity</th>
<th>Ease of Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sniffing</td>
<td>Unique!</td>
<td>Easy</td>
<td>Hard</td>
</tr>
<tr>
<td>Wardriving</td>
<td>Port Scanning</td>
<td>Easy</td>
<td>Hard</td>
</tr>
<tr>
<td>Replay</td>
<td>[same]</td>
<td>Easy</td>
<td>Moderate</td>
</tr>
<tr>
<td>Jamming</td>
<td>Denial of Service</td>
<td>Easy</td>
<td>Hard</td>
</tr>
<tr>
<td>Link Layer Congestion</td>
<td>Unique!</td>
<td>Moderate</td>
<td>Moderate</td>
</tr>
<tr>
<td>Evil Twin</td>
<td>ARP Spoofing</td>
<td>Hard</td>
<td>Moderate</td>
</tr>
<tr>
<td>Firmware Attack</td>
<td>Malware</td>
<td>Hard</td>
<td>Moderate</td>
</tr>
<tr>
<td>PHY Abuse</td>
<td>[same]</td>
<td>Hard</td>
<td>Hard</td>
</tr>
</tbody>
</table>
AS ATTACKERS...

- Look for low-hanging fruit first
  - Unencrypted comms, replay attacks, cleartext key exchanges, etc.
  - Complexity goes up in a hurry
- Bridge your traditional wired/IP network skill set wherever possible
AS ATTACKERS, CONTINUED

- Leverage Open Source Intelligence (OSINT):
  - FCC regulatory filings
  - Data sheets
- It will make your life easy
- Marc gave an entire talk on this at HITB2016AMS
AS DEVELOPERS // DEFENDERS . . .

- This is the **Golden Age of RF Hacking**
- Software Defined Radio has been commodity for >5 years
- Security Through Obscurity is a naive stance
TIME TO OWN YOUR AIRWAVES
ADDITIONAL RADIO RESOURCES

- “So You Want to Hack Radios” series (all about RF Physical Layers)
  - Shmoocon: [https://www.youtube.com/watch?v=L3udJnRe4vc](https://www.youtube.com/watch?v=L3udJnRe4vc)
  - Troopers: [https://www.youtube.com/watch?v=OFRwqpH9zAQ](https://www.youtube.com/watch?v=OFRwqpH9zAQ)
  - HITB2017AMS Commsec: [https://www.youtube.com/watch?v=QeoGQwT0Z1Y](https://www.youtube.com/watch?v=QeoGQwT0Z1Y)
- Matt’s LoRa research
  - 33c3: [https://media.ccc.de/v/33c3-7945-decoding_the_lora_phy](https://media.ccc.de/v/33c3-7945-decoding_the_lora_phy)
- Marc’s OSINT techniques
  - HITB2016AMS Commsec: [https://www.youtube.com/watch?v=JUAiav674D8](https://www.youtube.com/watch?v=JUAiav674D8)
- Dallas siren attack research
ACKNOWLEDGEMENTS

- Balint and Logan from Bastille’s Threat Research Team
- Bastille at large
- DEF CON 25 team!
- See you at DEF CON 50!
THANKS

github.com/BastilleResearch
QUESTIONS?

github.com/BastilleResearch

matt@Bastille.net
@embeddedsec

marc@Bastille.net
@marcnewlin