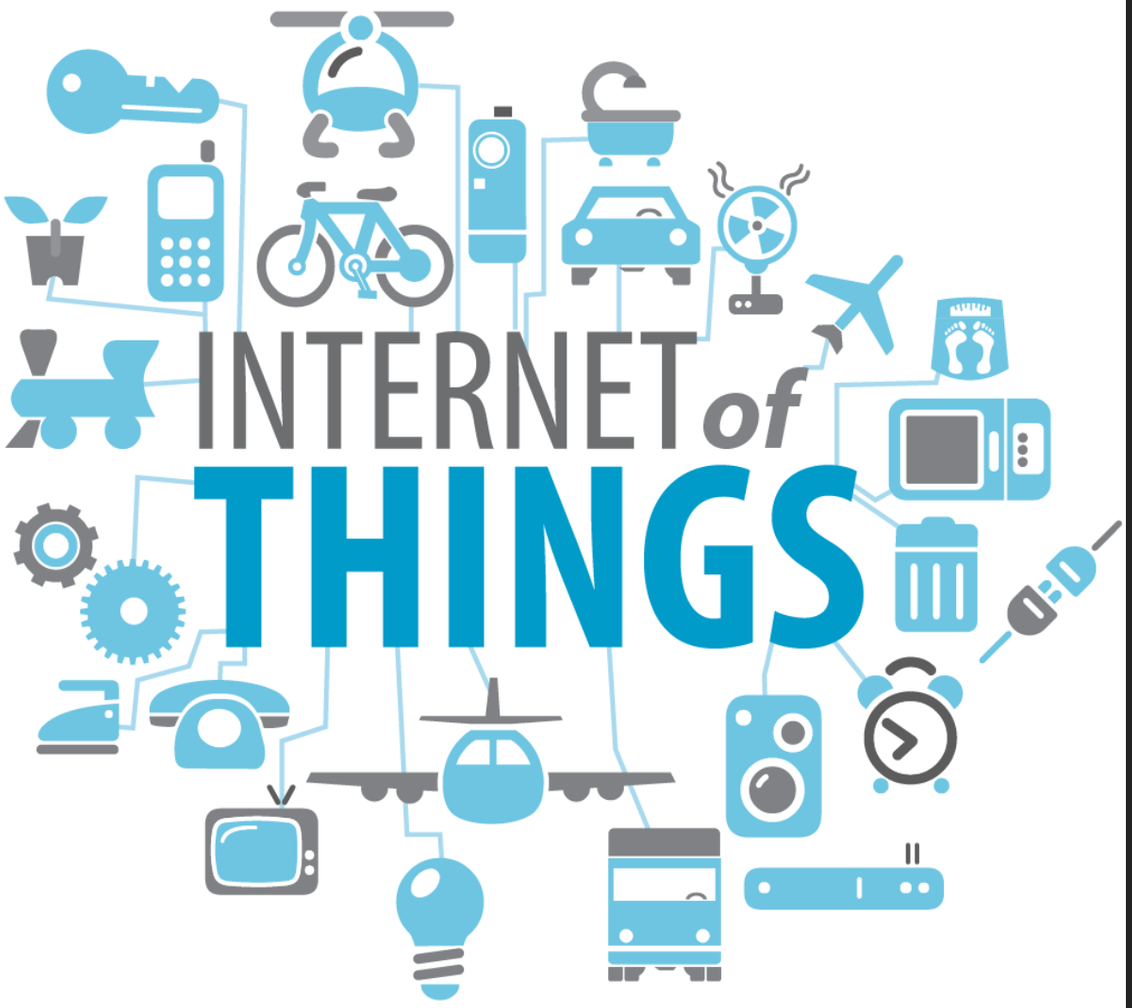
MATT KNIGHT // BASTILLE NETWORKS

WIRELESS LOCKPICKING

EXPLORING 802.15.4 COMMAND INJECTION

WHO AM I

- Matt Knight
- SWE & Threat Researcher @ Bastille Networks
- Passionate about:
  - Wireless sensor networks
  - Information security
  - Finding out what hardware actually does



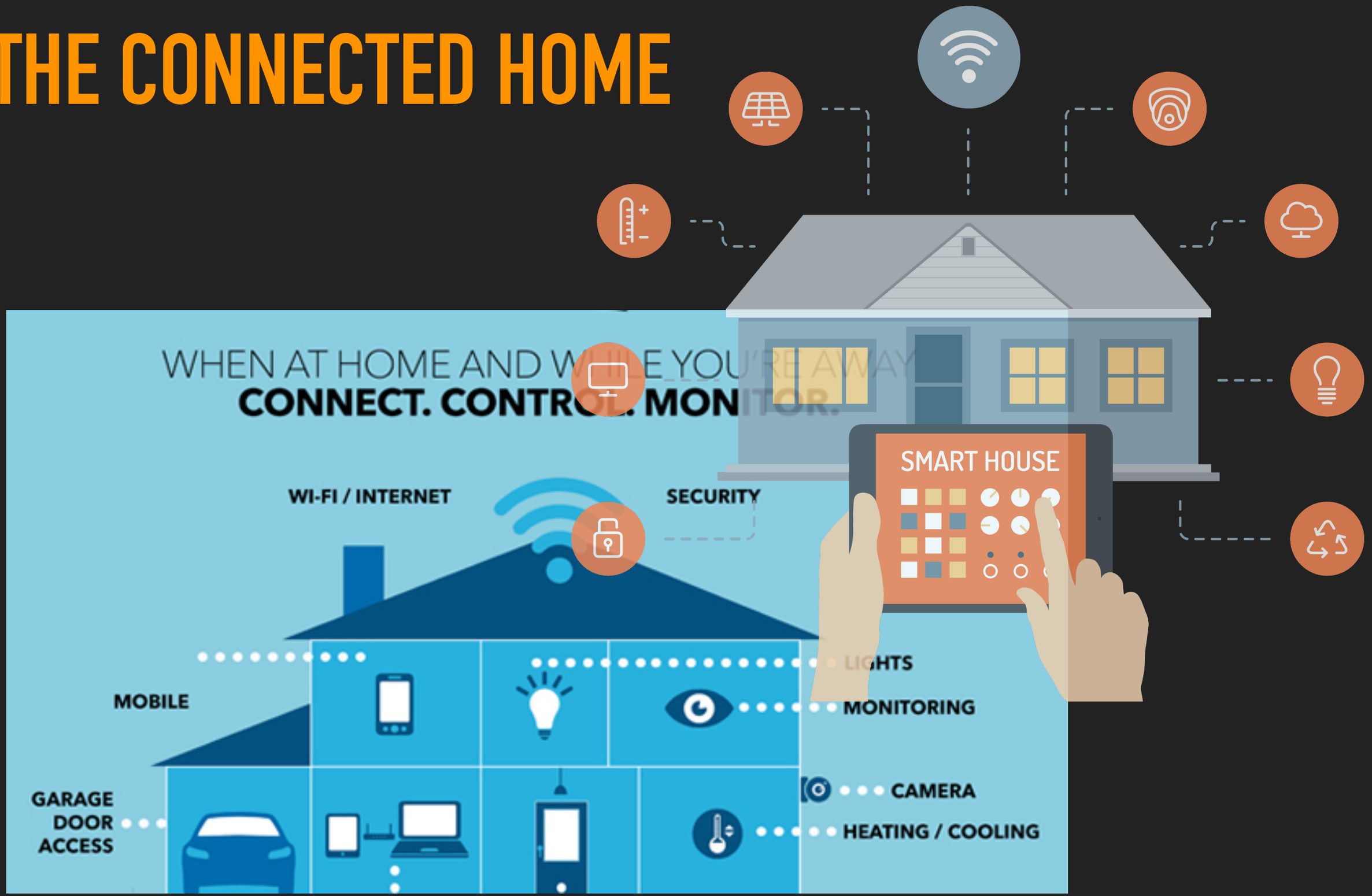
graphic: http://www.fronetics.com/wp-content/uploads/2015/03/Internet-of-things-and-the-supply-chain-industry.png

INTERNET OF THINGS

50 BILLION

BY 2020

THE CONNECTED HOME



http://www.marketsonline.co.za/wp-content/uploads/2015/01/SmartHome.png
https://corporate.bestbuy.com/wp-content/uploads/2014/10/connected-house-main-image1.jpg

RSA IOT SANDBOX

- Coordinated by Balint Seeber, Jesus Molina, and Joe Gordon
- Idyllic IoT environment presented: the smart home
- Common smart home functions relate to physical security
  - Door locks
  - Home security systems

THE HEIST

THE SETUP

- Yale door lock
- DSC home security system
- Internet connected doll
- Wireless shock collar
- IoT Snake!

THE HEIST

THE SETUP

- Yale door lock
- DSC home security system
- Internet connected doll
- Wireless shock collar
- IoT Snake!

THE HEIST

Unlocked wirelessly

THE SETUP

- Yale door lock
- DSC home security system
- Internet connected doll
- Wireless shock collar
- IoT Snake!

THE HEIST

- Unlocked wirelessly
- Jammed

THE SETUP

- Yale door lock
- DSC home security system
- Internet connected doll
- Wireless shock collar
- IoT Snake!

THE HEIST

- Unlocked wirelessly
- Jammed
- Spies on your kids

THE SETUP

THE HEIST

- Yale door lock → Unlocked wirelessly
- DSC home security system → Jammed
- Internet connected doll → Spies on your kids
- Wireless shock collar → Shocks attack dog
- IoT Snake!

THE SETUP

THE HEIST

- Yale door lock → Unlocked wirelessly
- DSC home security system → Jammed
- Internet connected doll → Spies on your kids
- Wireless shock collar → Shocks attack dog
- IoT Snake! → Does snake stuff

THE SETUP	THE HEIST
Yale door lock	Unlocked wirelessly
DSC home security system	Jammed
Internet connected doll	Spies on your kids
Wireless shock collar	Shocks attack dog
IoT Snake!	Does snake stuff

YALE DOOR LOCK

- Made smart by add-on ZigBee module
- Connected to SmartThings IoT hub

top: http://www.bhphotovideo.com/images/images1000x1000/yale_yrd220_ha_619_keyed_touchscreen_deadbolt_with_1172756.jpg
bottom: https://support.smartthings.com/hc/en-us/article_attachments/202949444/Samsung-SmartThings-Hub-Front-Angle.jpg

WIRELESS PROTOCOLS

- Incumbent
  - Cellular, WiFi, WiFi Direct, Bluetooth, BTLE, 802.15.4 (incl. ZigBee, 6PAN, Thread), Z-Wave, ANT, Enocean, etc.

STANDARD	DATA RATE	RANGE	OPERATING SPECTRUM
Bluetooth	0.1-3 Mbps	1-10m (HPA: 100m)	2.4Ghz
Bluetooth Low Energy	1Mbps	5-10m	2.4Ghz
Zigbee / 802.15.4	20-250 kbps	1-100m	868Mhz, 915Mhz, 2.4Ghz
Passive RFID	868 kbps	0.01-3m	860-960Mhz, 13.5Mhz
Active RFID	10's of Mbps	0.01-100m	433Mhz
WLAN	1-54 Mbps	10-100m	2.4Ghz
UWB	53-480 Mbps	3-10m	3.1-10.6Ghz
Proprietary (Sensium)	50kbps	~3m	862-870Mhz 902-928Mhz
802.11a	54Mb/s	90-100ft	5.0Ghz
802.11b	11Mb/s	250-300ft	2.4Ghz
802.11g	54Mb/s	100ft	2.4Ghz

http://cdn.nuvation.com/wp-content/uploads/2014/02/medical_wireless_table.jpg

WIRELESS PROTOCOLS

- Incumbent
  - Cellular, WiFi, WiFi Direct, Bluetooth, BTLE, 802.15.4 (incl. ZigBee, 6PAN, Thread), Z-Wave, ANT, Enocean, etc.
- Emerging
  - LoRa, SIGFOX, Ingenu, LTE-M

STANDARD	DATA RATE	RANGE	OPERATING SPECTRUM
Bluetooth	0.1-3 Mbps	1-10m (HPA: 100m)	2.4Ghz
Bluetooth Low Energy	1Mbps	5-10m	2.4Ghz
Zigbee / 802.15.4	20-250 kbps	1-100m	868Mhz, 915Mhz, 2.4Ghz
Passive RFID	868 kbps	0.01-3m	860-960Mhz, 13.5Mhz
Active RFID	10's of Mbps	0.01-100m	433Mhz
WLAN	1-54 Mbps	10-100m	2.4Ghz
UWB	53-480 Mbps	3-10m	3.1-10.6Ghz
Proprietary (Sensium)	50kbps	~3m	862-870Mhz 902-928Mhz
802.11a	54Mb/s	90-100ft	5.0Ghz
802.11b	11Mb/s	250-300ft	2.4Ghz
802.11g	54Mb/s	100ft	2.4Ghz

http://cdn.nuvation.com/wp-content/uploads/2014/02/medical_wireless_table.jpg

WIRELESS PROTOCOLS

- Incumbent
  - Cellular, WiFi, WiFi Direct, Bluetooth, BTLE, 802.15.4 (incl. ZigBee, 6PAN, Thread), Z-Wave, ANT, Enocean, etc.
- Emerging
  - LoRa, SIGFOX, Ingenu, LTE-M
- Deprecation
  - AT&T 2G GSM shutdown this year!

STANDARD	DATA RATE	RANGE	OPERATING SPECTRUM
Bluetooth	0.1-3 Mbps	1-10m (HPA: 100m)	2.4Ghz
Bluetooth Low Energy	1Mbps	5-10m	2.4Ghz
Zigbee / 802.15.4	20-250 kbps	1-100m	868Mhz, 915Mhz, 2.4Ghz
Passive RFID	868 kbps	0.01-3m	860-960Mhz, 13.5Mhz
Active RFID	10's of Mbps	0.01-100m	433Mhz
WLAN	1-54 Mbps	10-100m	2.4Ghz
UWB	53-480 Mbps	3-10m	3.1-10.6Ghz
Proprietary (Sensium)	50kbps	~3m	862-870Mhz 902-928Mhz
802.11a	54Mb/s	90-100ft	5.0Ghz
802.11b	11Mb/s	250-300ft	2.4Ghz
802.11g	54Mb/s	100ft	2.4Ghz

http://cdn.nuvation.com/wp-content/uploads/2014/02/medical_wireless_table.jpg

WIRELESS PROTOCOLS

- Incumbent
  - Cellular, WiFi, WiFi Direct, Bluetooth, BTLE, 802.15.4 (incl. ZigBee, 6PAN, Thread), Z-Wave, ANT, Enocean, etc.
- Emerging
  - LoRa, SIGFOX, Ingenu, LTE-M
- Deprecation
  - AT&T 2G GSM shutdown this year!

STANDARD	DATA RATE	RANGE	OPERATING SPECTRUM
Bluetooth	0.1-3 Mbps	1-10m (HPA: 100m)	2.4Ghz
Bluetooth Low Energy	1Mbps	5-10m	2.4Ghz
Zigbee / 802.15.4	20-250 kbps	1-100m	868Mhz, 915Mhz, 2.4Ghz
Passive RFID	868 kbps	0.01-3m	860-960Mhz, 13.5Mhz
Active RFID	10's of Mbps	0.01-100m	433Mhz
WLAN	1-54 Mbps	10-100m	2.4Ghz
UWB	53-480 Mbps	3-10m	3.1-10.6Ghz
Proprietary (Sensium)	50kbps	~3m	862-870Mhz 902-928Mhz
802.11a	54Mb/s	90-100ft	5.0Ghz
802.11b	11Mb/s	250-300ft	2.4Ghz
802.11g	54Mb/s	100ft	2.4Ghz

http://cdn.nuvation.com/wp-content/uploads/2014/02/medical_wireless_table.jpg

ZIGBEE

- Defines NWK and APP layers on top of 802.15.4 PHY/MAC
- Mesh routing topology
- Application Profiles: flexibility to suit different applications

http://www.securitymerchants.com.au/presets/product-slideshow/PageFiles/1446985/ImageGallery/130816040947532.jpg

ZIGBEE SECURITY

- AES-128
- Network key shared when device is added to network
- OTA key exchange encrypted with a pre-shared key… the value of which is widely known
- 2015 paper by Tobias Zillner/Cognosec: https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly-wp.pdf

BATTERY POWERED COMMS

- Battery powered devices spend most of the time asleep
- Radios require a lot of power; battery powered devices can’t afford to listen promiscuously
- PANs are designed with power consumption in mind
  - Z-Wave: beaming
  - 802.15.4: indirect data request

802.15.4 INDIRECT DATA TRANSFER

- Battery powered device calls home and asks for updates

```
▶ Frame 141: 12 bytes on wire (96 bits), 12 bytes captured (96 bits) on interface 0
▼ IEEE 802.15.4 Command, Dst: 0x0000, Src: 0xd7b9
  ▼ Frame Control Field: 0x8863, Frame Type: Command, Acknowledge Request, Intra-PAN, Destination Addressing Mode: Short/16-bit, Source Addressing Mode: Short/16-bit
      .... .... .... .011 = Frame Type: Command (0x0003)
      .... .... .... 0... = Security Enabled: False
      .... .... ...0 .... = Frame Pending: False
      .... .... ..1. .... = Acknowledge Request: True
      .... .... .1.. .... = Intra-PAN: True
      .... 10.. .... .... = Destination Addressing Mode: Short/16-bit (0x0002)
      ..00 .... .... .... = Frame Version: 0
      10.. .... .... .... = Source Addressing Mode: Short/16-bit (0x0002)
    Sequence Number: 133
    Destination PAN: 0x3eab
    Destination: 0x0000
    Source: 0xd7b9
    Command Identifier: Data Request (0x04)
    FCS: 0x411c (Correct)
```

802.15.4 INDIRECT DATA TRANSFER

- Battery powered device calls home and asks for updates

```
▶ Frame 141: 12 bytes on wire (96 bits), 12 bytes captured (96 bits) on interface 0
▼ IEEE 802.15.4 Command, Dst: 0x0000, Src: 0xd7b9
  ▼ Frame Control Field: 0x8863, Frame Type: Command, Acknowledge Request, Intra-PAN, Destination Addressing Mode: Short/16-bit, Source Addressing Mode: Short/16-bit
      .... .... .... .011 = Frame Type: Command (0x0003)
      .... .... .... 0... = Security Enabled: False
      .... .... ...0 .... = Frame Pending: False
      .... .... ..1. .... = Acknowledge Request: True
      .... .... .1.. .... = Intra-PAN: True
      .... 10.. .... .... = Destination Addressing Mode: Short/16-bit (0x0002)
      ..00 .... .... .... = Frame Version: 0
      10.. .... .... .... = Source Addressing Mode: Short/16-bit (0x0002)
    Sequence Number: 133
    Destination PAN: 0x3eab
    Destination: 0x0000
    Source: 0xd7b9
    Command Identifier: Data Request (0x04)
    FCS: 0x411c (Correct)
```

- ACK FCF signals whether data is pending

```
▶ Frame 142: 5 bytes on wire (40 bits), 5 bytes captured (40 bits) on interface 0
▼ IEEE 802.15.4 Ack, Sequence Number: 133
  ▼ Frame Control Field: 0x0012, Frame Type: Ack, Frame Pending, Destination Addressing Mode: None, Source Addressing Mode: None
      .... .... .... .010 = Frame Type: Ack (0x0002)
      .... .... .... 0... = Security Enabled: False
      .... .... ...1 .... = Frame Pending: True
      .... .... ..0. .... = Acknowledge Request: False
      .... .... .0.. .... = Intra-PAN: False
      .... 00.. .... .... = Destination Addressing Mode: None (0x0000)
      ..00 .... .... .... = Frame Version: 0
      00.. .... .... .... = Source Addressing Mode: None (0x0000)
    Sequence Number: 133
    FCS: 0xe388 (Correct)
```

UNLOCK SEQUENCE

LOCK

- Lock makes data request
- Lock ACKs unlock message

HUB

- Unlock message queued on hub
- Hub ACKs data request; frame pending flag set
- Hub sends unlock message

UNLOCK SEQUENCE

699	363.673387	0xbcbe	0x0000	IEEE 802.15.4	12	Data Request
700	363.675454	0xbcbe	0x0000	IEEE 802.15.4	12	Data Request
701	363.676000			IEEE 802.15.4	5	Ack
702	368.678182	0xbcbe	0x0000	IEEE 802.15.4	12	Data Request
703	368.678673			IEEE 802.15.4	5	Ack
704	373.681447	0xbcbe	0x0000	IEEE 802.15.4	12	Data Request
705	373.681926			IEEE 802.15.4	5	Ack
706	373.685013	0x0000	0xbcbe	ZigBee HA	48	Unknown Command: 0x01, Seq: 4
707	373.685537			IEEE 802.15.4	5	Ack
708	373.698856	0xbcbe	0x0000	ZigBee	45	APS: Ack, Dst Endpt: 1, Src Endpt: 1
709	373.699353			IEEE 802.15.4	5	Ack
710	373.702966	0xbcbe	0x0000	ZigBee HA	49	Unknown Command: 0x01, Seq: 4
711	373.703472			IEEE 802.15.4	5	Ack
712	373.933852	0xbcbe	0x0000	IEEE 802.15.4	12	Data Request
713	373.934859			IEEE 802.15.4	5	Ack
714	373.938924	0x0000	0xbcbe	ZigBee HA	50	ZCL: Default Response, Seq: 4
715	373.939418			IEEE 802.15.4	5	Ack

THE ATTACK

UNLOCK SEQUENCE

LOCK

- Lock makes data request
- Lock ACKs unlock message

HUB

- Unlock message queued on hub
- Hub ACKs data request; frame pending flag set
- Hub sends unlock message

UNLOCK SEQUENCE

THE ATTACK

LOCK

- Lock makes data request
- Lock ACKs unlock message

HUB

- Hub is idle
- Hub ACKs data request; frame pending flag set
- Hub sends unlock message

UNLOCK SEQUENCE

THE ATTACK

LOCK

- Lock makes data request
- Lock ACKs unlock message

HUB

- Hub is idle
- **FORGED** ACKs data request; frame pending flag set
- Hub sends unlock message

1. **Inject forged ACK w/ frame pending set**

THE ATTACK

LOCK

- Lock makes data request

- Lock ACKs unlock message

HUB

- Hub is idle

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

1. **Inject forged ACK w/ frame pending set**
2. **Inject forged unlock frame**

THE ATTACK

LOCK

- Lock makes data request

HUB

- Hub is idle

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

- Lock ACKs unlock message

0. Sniff encryption key
1. Inject forged ACK w/ frame pending set
2. Inject forged unlock frame

ATTEMPT 1: HOST-BASED USRP

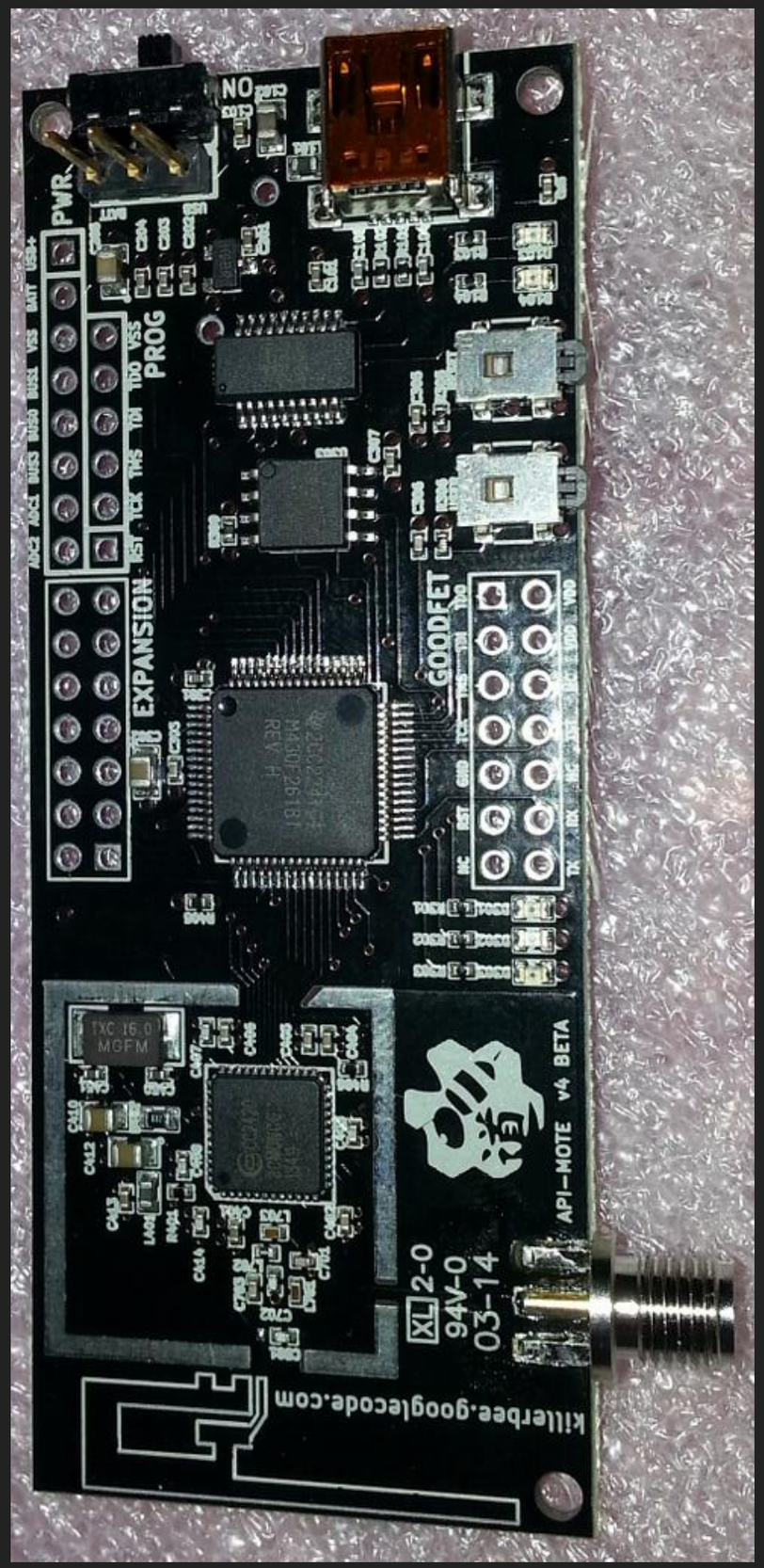
- USRP B210; gr-ieee802-15-4
- Need sequence number from Data Request frame to compose ACK
- 802.15.4 ACK timeout: 864 us
- USB latency: ~ms
- Verdict:

photo: Balint Seeber

ATTEMPT 1: HOST-BASED USRP

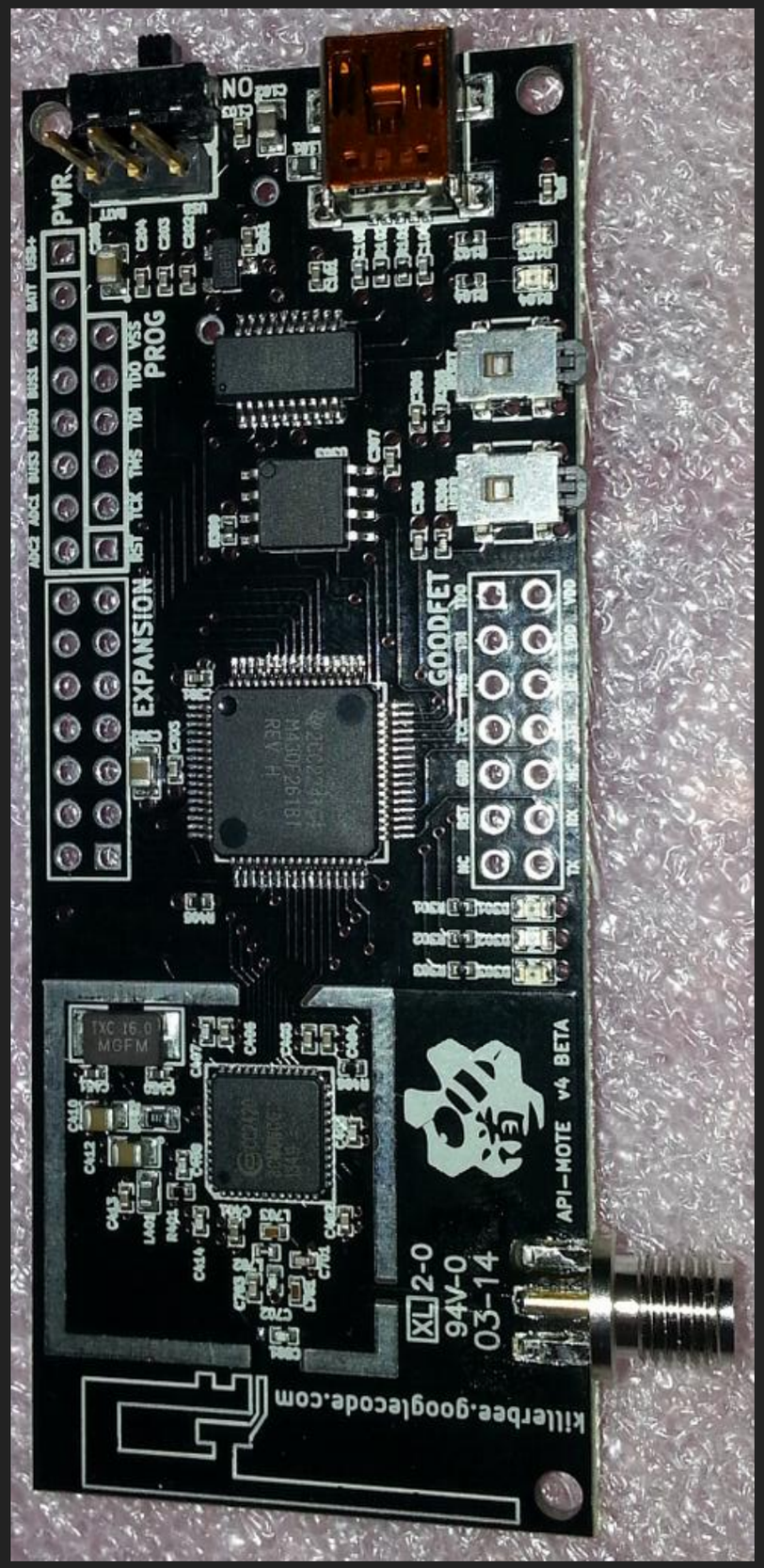
- USRP B210; gr-ieee802-15-4
- Need sequence number from Data Request frame to compose ACK
- 802.15.4 ACK timeout: 864 us
- USB latency: ~ms
- Verdict: too slow!

photo: Balint Seeber



ATTEMPT 2: APIMOTE V4BETA

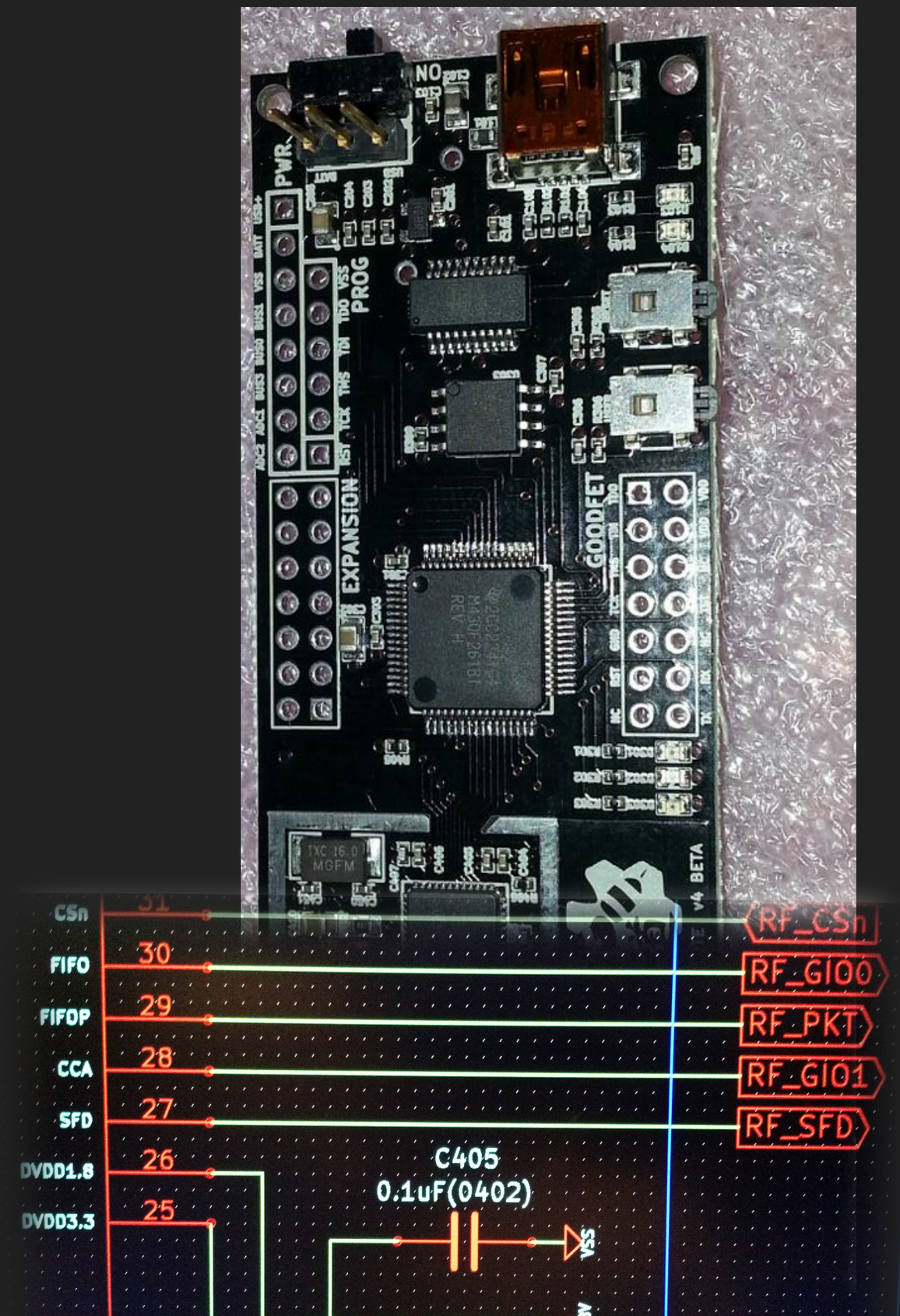
- USB 802.15.4 injection peripheral made by River Loop Security
  - Host-based Killerbee attack framework
  - MSP430 running GoodFET firmware
  - CC2420 commodity RF IC
- USB2 to FTDI; UART to MSP430; bitbanged SPI to CC2420



ATTEMPT 2: APIMOTE V4BETA

- USB 802.15.4 injection peripheral made by River Loop Security
  - Host-based Killerbee attack framework
  - MSP430 running GoodFET firmware
  - CC2420 commodity RF IC
- USB to SPI/JTAG/UART; FTDI; UART to MSP430; bitbanged SPI to CC2420

TOO SLOW



MODIFIED APIMOTE FIRMWARE

- Pre-load unlock command on MSP430, via host
- Reflexively jam Data Request frame from lock
  - Record enough symbols before jamming to get the sequence number
- Generate ACK in firmware
- Inject ACK and unlock command

UNLOCK SEQUENCE

THE ATTACK

LOCK

- Lock makes data request
- Lock ACKs unlock message

HUB

- Hub is idle
- Hub ACKs data request; frame pending flag set
- Hub sends unlock message

UNLOCK SEQUENCE

THE ATTACK

LOCK

- ~~Lock makes data request~~
- Lock ACKs unlock message
    1. **Jam data request**

HUB

- Hub is idle
- Hub ACKs data request; frame pending flag set
- Hub sends unlock message

THE ATTACK SEQUENCE

LOCK

- ~~Lock makes data request~~
- Lock ACKs unlock message
    1. **Jam data request**
    2. **Inject forged ACK w/ frame pending set**

HUB

- Hub is idle
- **FORGED** ACKs data request; frame pending flag set
- Hub sends unlock message

THE ATTACK

UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- Lock ACKs unlock message

HUB

- Hub is idle
- FORGED ACKs data request; frame pending flag set
- FORGED sends unlock message

1. Jam data request
2. Inject forged ACK w/ frame pending set
3. Inject forged unlock frame

UNLOCK SEQUENCE

THE ATTACK

LOCK

- ~~Lock makes data request~~
- ~~Lock ACKs unlock message~~

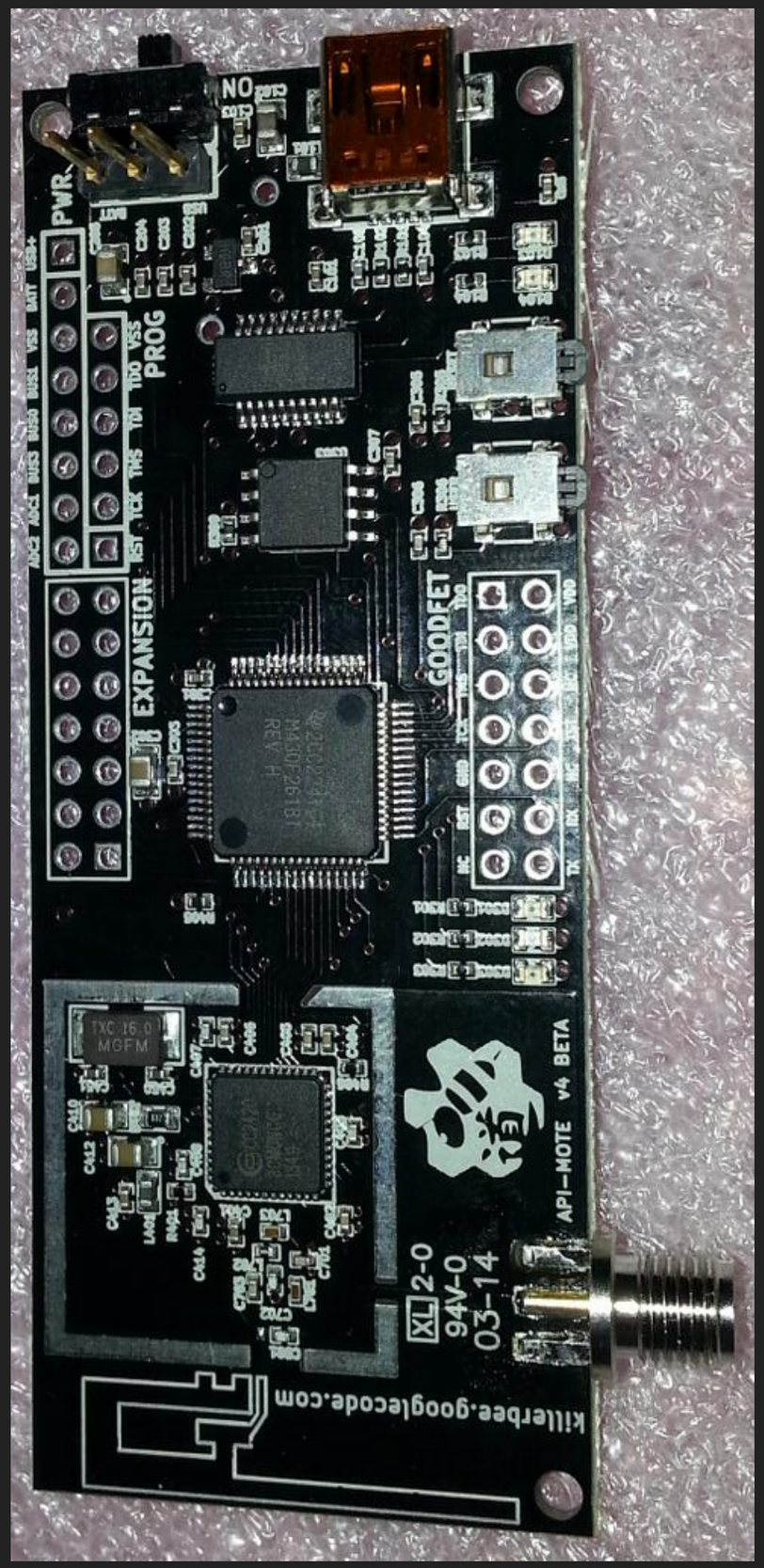
HUB

- Hub is idle

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

1. Jam data request
2. Inject forged ACK w/ frame pending set
3. Inject forged unlock frame
4. Jam ongoing traffic



ATTEMPT 2 RESULT

- Still too slow!!
- Jam works correctly but ACK arrives too late
- SPI latency is too high

ATTEMPT 3: COMMAND PIPELINING

- ApiMote with custom firmware
- Exploits protocol retries to fit in SPI transactions
  - If a frame goes un-ACKed, sender will reattempt after a timeout
  - 3 or 4 attempts typical
- Strategy: Jam initial message and retries while reading/composing ACK

699	363.673387	0xbcbe	0x0000	IEEE 802.15.4	12	23	Data Request
700	363.675454	0xbcbe	0x0000	IEEE 802.15.4	12	23	Data Request
701	363.676000			IEEE 802.15.4	5	23	Ack

THE ATTACK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock ACKs unlock message~~

HUB

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

1. Jam data request
2. Inject forged ACK w/ frame pending set
3. Inject forged unlock frame
4. Jam ongoing traffic

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~

HUB

1. Jam data request, read its sequence number

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~

HUB

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~
- ~~Lock makes data request~~

HUB

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~
- ~~Lock makes data request~~

HUB

FORGED ACKs data request; frame pending flag set

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK
4. Inject forged ACK again

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~
- ~~Lock makes data request~~

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK
4. Inject forged ACK again
5. Inject forged unlock frame

HUB

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~
- ~~Lock makes data request~~

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK
4. Inject forged ACK again
5. Inject forged unlock frame
6. Jam ongoing traffic

HUB

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

Live Demo Time

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~
- ~~Lock makes data request~~

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK
4. Inject forged ACK again
5. Inject forged unlock frame
6. Jam ongoing traffic

HUB

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~

HUB

165	148.410124	0xd7b9	0x0000	IEEE 802.15.4	12	141	Data Request
166	148.410161			IEEE 802.15.4	5	141	Ack
167	153.411391			IEEE 802.15.4	4	255	Data[Malformed Packet]
168	153.413415			IEEE 802.15.4	4	255	Data[Malformed Packet]
169	153.415412			IEEE 802.15.4	5	142	Ack
170	153.415944			IEEE 802.15.4	5	142	Ack
171	153.419493	0x0000	0xd7b9	ZigBee HA	48	4	Unknown Command: 0x01, Seq: 3
172	153.420050			IEEE 802.15.4	5	4	Ack

frame pending flag set

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK
4. Inject forged ACK again
5. Inject forged unlock frame
6. Jam ongoing traffic

FORGED sends unlock message

PIPELINED UNLOCK SEQUENCE

LOCK

- ~~Lock makes data request~~
- ~~Lock makes data request~~
- ~~Lock makes data request~~

HUB

FORGED ACKs data request; frame pending flag set

FORGED sends unlock message

0. Still need that key!

1. Jam data request, read its sequence number
2. Jam data request, load forged ACK frame in CC2420's TXFIFO
3. Jam data request using forged ACK
4. Inject forged ACK again
5. Inject forged unlock frame
6. Jam ongoing traffic

WHAT ABOUT THAT MAJOR CAVEAT…?

- Attacker must possess ZigBee encryption key
- Key is sent in the clear only once, when lock is added to network

WHAT ABOUT THAT MAJOR CAVEAT…?

- Attacker must possess ZigBee encryption key
- Key is sent in the clear only once, when lock is added to network
- Entice user to add lock to network while you are sniffing
  - De-authentication attack
  - Denial of service
  - “Insecure rejoin” (see Zillner’s paper)

CONCLUSIONS

- SDR is awesome, but not always the right tool for the job
- Commodity hardware is a powerful complement
- We have a long way to go to secure the IoT

- ApiMote and GoodFET enhancements to be released on Bastille RFStorm's github: github.com/RFStorm

ACKNOWLEDGEMENTS

- Bastille RFStorm / Balint Seeber
- River Loop Security (ApiMote & Killerbee)
- Tobias Zillner (SecBee)
- Bastian Bloessel (gr-ieee802-15-4)
- Cyberspectrum community

Thanks!

Twitter: @embeddedsec
Email: [firstname]@bastille.io

Questions?

Twitter: @embeddedsec
Email: [firstname]@bastille.io