AMI Attack and Defense Showdown

Defining an Attack Methodology for the Security Evaluation of Advanced Metering Infrastructure Devices

Copyright 2009 InGuardians, Inc.

Your Speakers

- Joshua Wright – josh@inguardians.com
  - Senior Security Analyst, InGuardians
  - Senior Instructor, SANS Institute
  - Co-Author, "Hacking Exposed Wireless" (2010), "Emerging Technologies in Wireless LANs" (2008), "Wireshark Packet Sniffing" (2007)
- Matthew Carpenter – matt@inguardians.com
  - Senior Security Analyst, InGuardians
  - Chair: NIST CSCTG Vulnerabilities Group
  - Vice-chair: Utilisec and AMI-SEC
  - Lead: AMI-SEC Red-team

Copyright 2009 InGuardians, Inc.

Outline

- Introduction to AMI Technology and the Attack Methodology
- Principles of AMI Assessment
- AMI Attacks and Countermeasures
- Conclusion

Copyright 2009 InGuardians, Inc.

What is "AMI"?

- Advanced Metering Infrastructure
- Two-way communication between utility and meters
- Meter reading (electric, gas, water)
- Disconnect switch
- Load Control (ex. ZigBee from meter to thermostat/PCT)
- Basis of Smart Grid... it's the base network

AMI has the potential to reach billions of homes and businesses

Copyright 2009 InGuardians, Inc.

Smart Thermostat

- Radio Thermostat of America CT80
  - ZigBee or IEEE 802.11
- Implements Smart Energy Profile
- Load control and demand response capabilities

Copyright 2009 InGuardians, Inc.

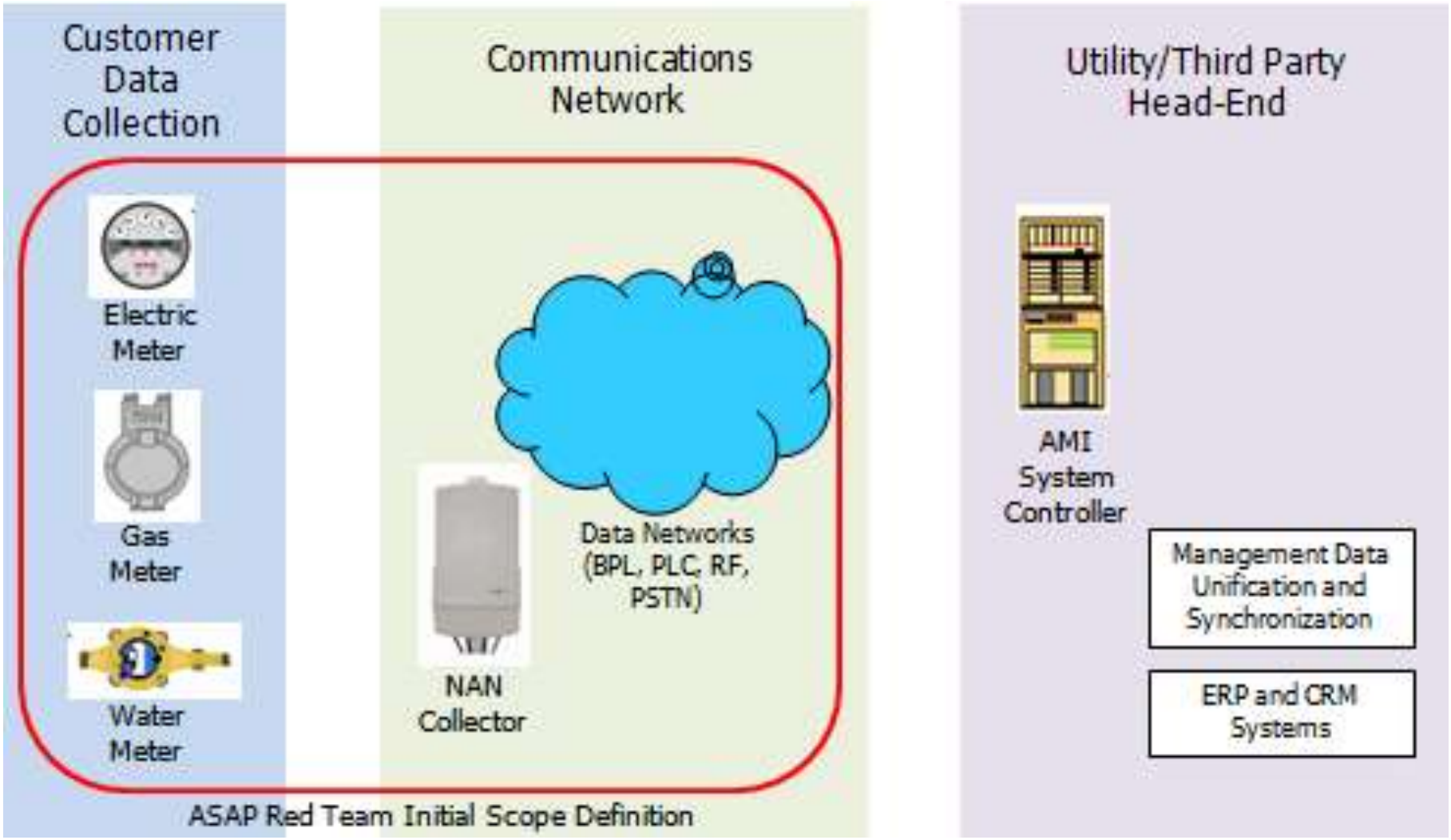
AMI Attack Methodology

- InGuardians contracted by the AMI-SEC task force (UCAIug) to develop a security testing methodology
  - Detail tools/techniques for evaluating embedded AMI components
  - For use by vendors creating products and customers testing networks
  - Help organizations understand attacks and vulnerabilities

A limited amount of information is available regarding embedded AMI attacks

Copyright 2009 InGuardians, Inc.

Attack Methodology Scope



Copyright 2009 InGuardians, Inc.

A Challenging Task

- The analysis of AMI systems covers numerous technology specialties
  - Hardware circuit attack techniques
  - Wireless communication systems
  - Software assessment and attacks
  - Cryptography assessment and attack
  - Reverse engineering proprietary systems
- We identified multiple attack vectors, providing guidance for testing, analysis

Copyright 2009 InGuardians, Inc.

Outline

- Introduction to AMI Technology and the Attack Methodology
- Principles of AMI Assessment
- AMI Attacks and Countermeasures
- Conclusion

Copyright 2009 InGuardians, Inc.

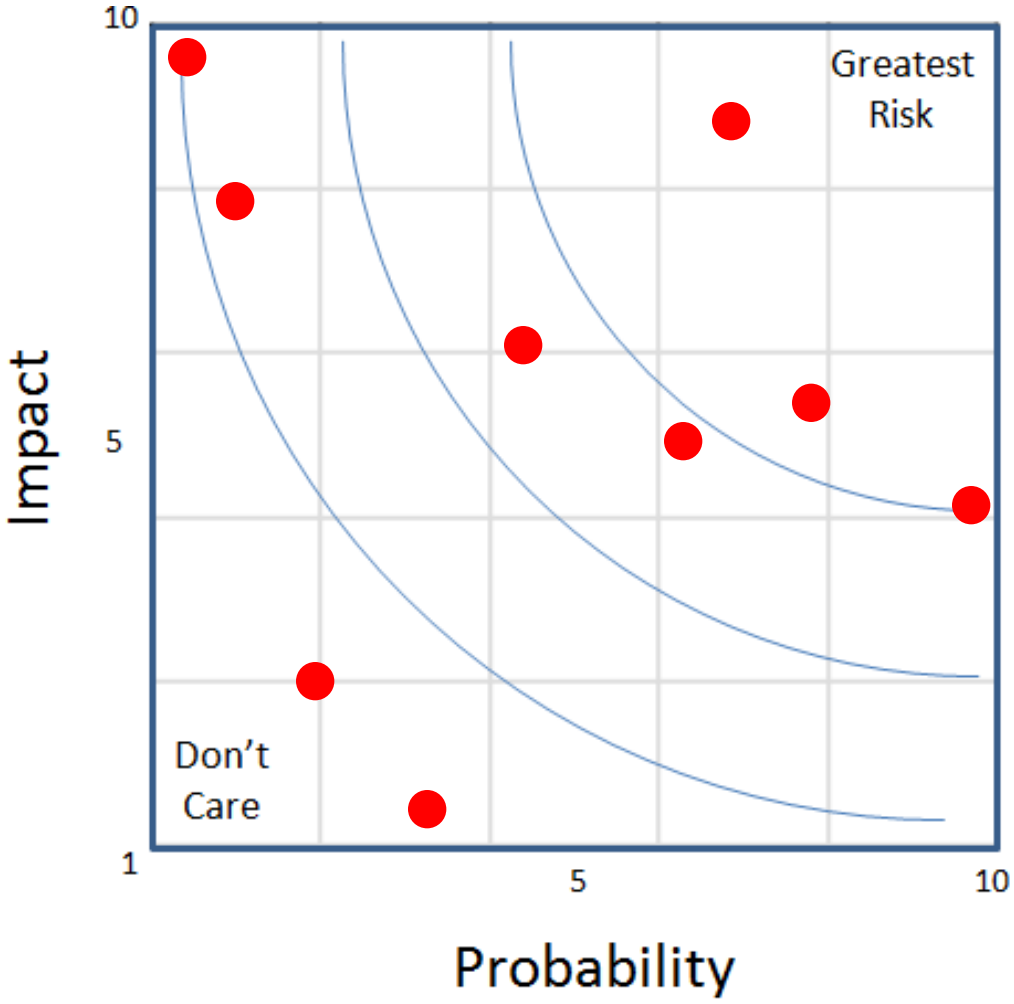
Practical and Pertinent Analysis

- Test procedures must be reproducible
  - Vulnerabilities discovered by a customer must be reproduced by a utility to assess
- Attacks against AMI must be scoped by time and resource
  - An attacker with unlimited resources and unlimited time can defeat any security system
- Vulnerabilities will exist in any system, and must be assessed for applicability
  - What are the resources of an attacker you are willing to defend against?

Copyright 2009 InGuardians, Inc.

Risk Assessment

- Vulnerabilities are assessed to identify impact and probability
  - Scaled 1-10
- Low impact and low probability: don't care
- High impact and high probability: greatest risk



Copyright 2009 InGuardians, Inc.

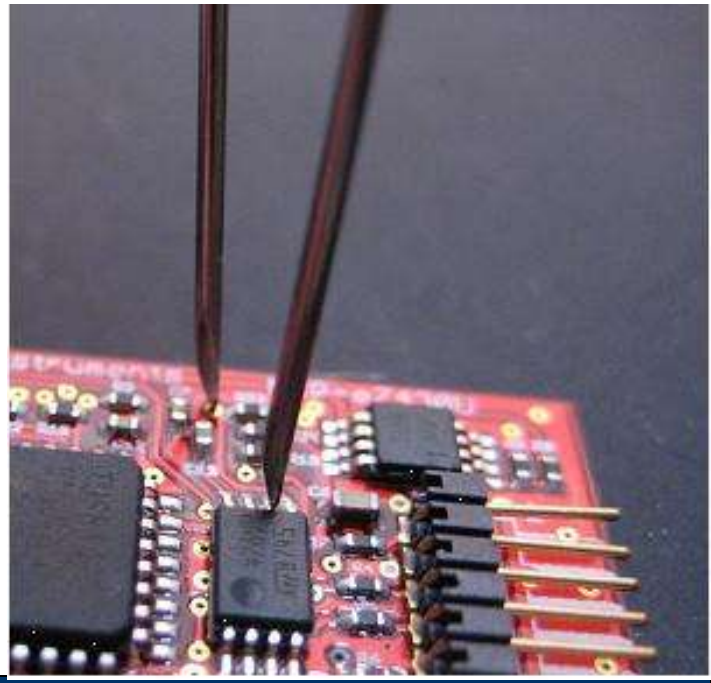
Outline

- Introduction to AMI Technology and the Attack Methodology
- Principles of AMI Assessment
- AMI Attacks and Countermeasures
- Conclusion

Copyright 2009 InGuardians, Inc.

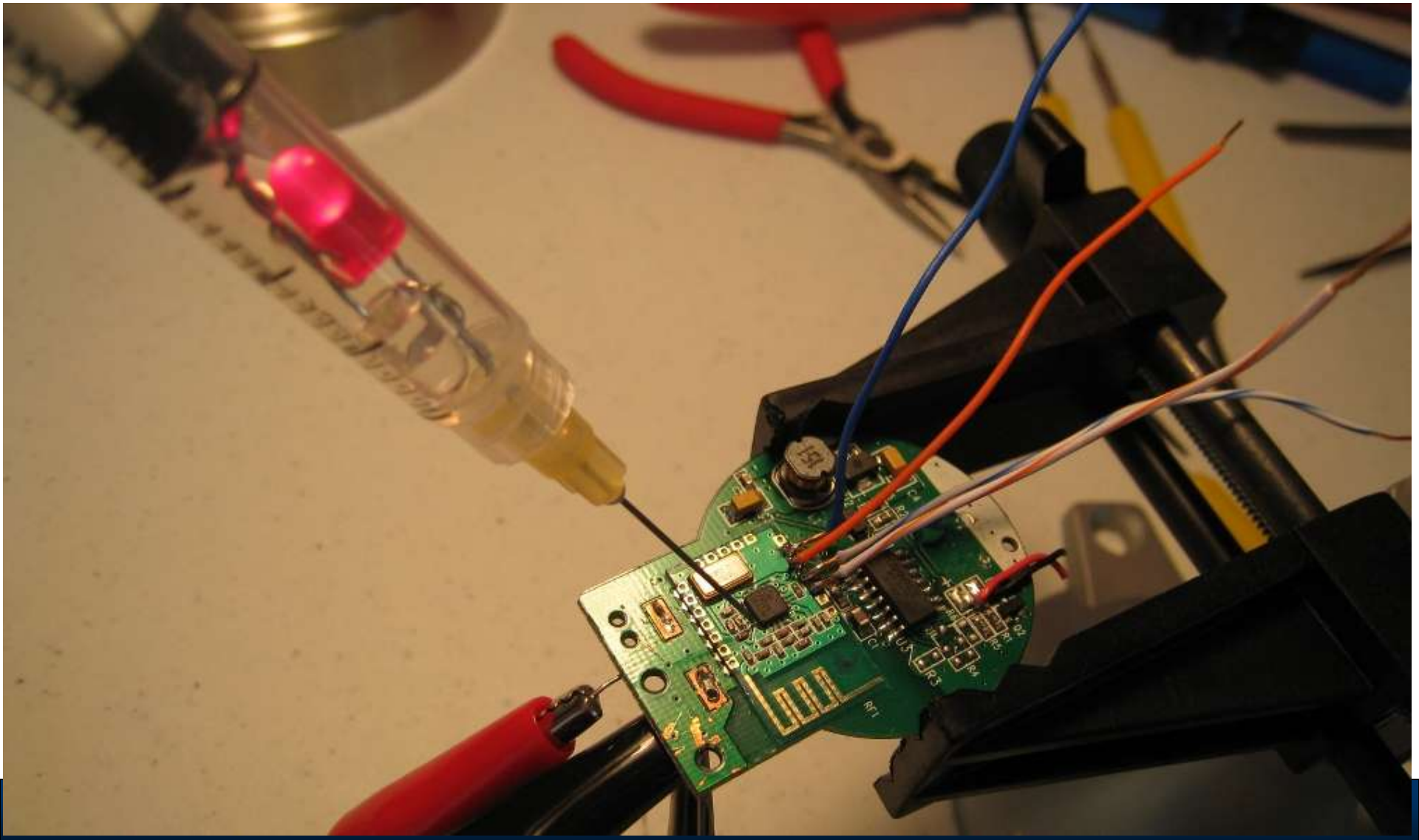
Attack: Hardware Bus Snooping

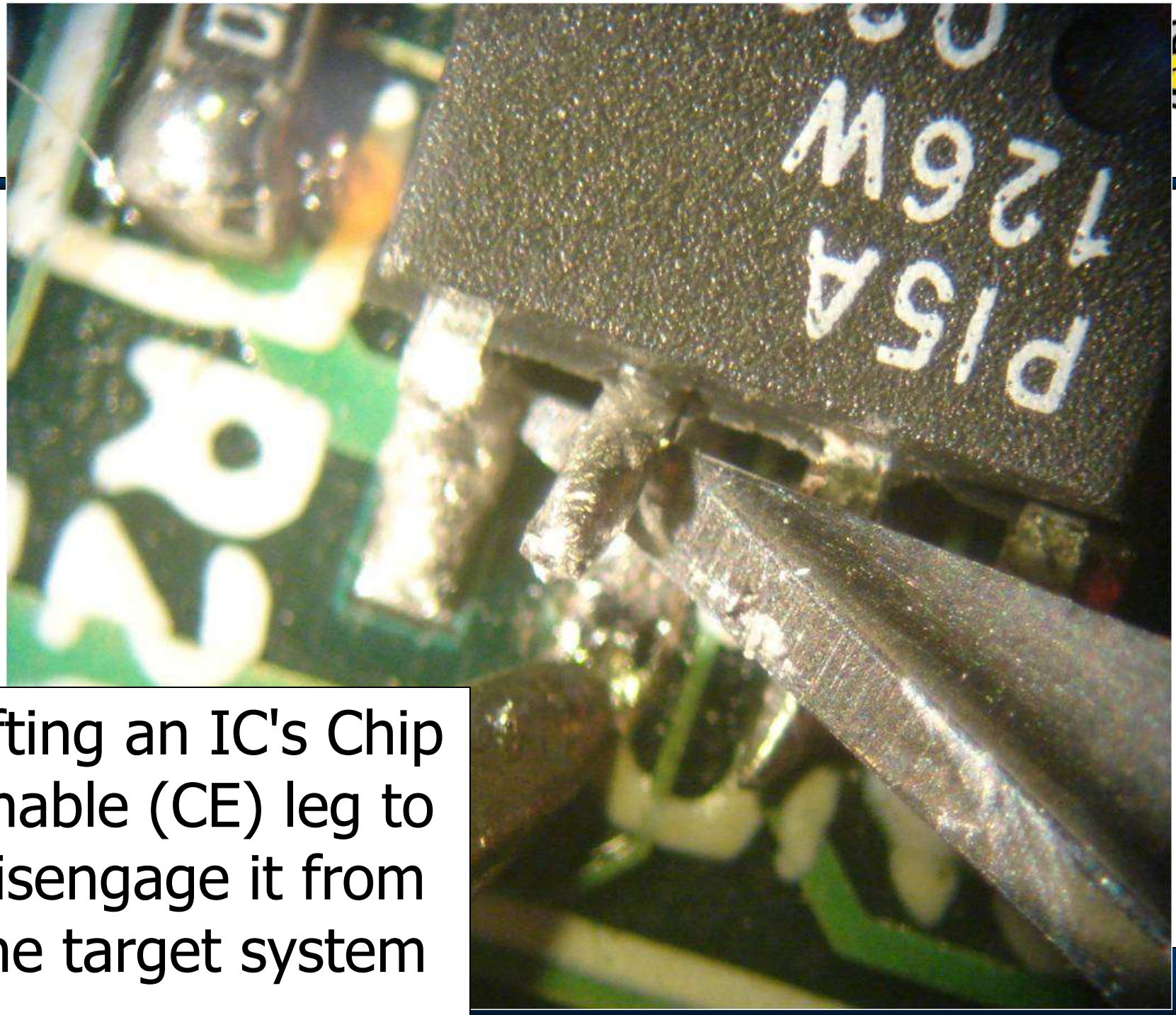
- Intercepting data between circuit board peripherals
  - Key load operations between a uC and crypto accelerator
  - Software updates between radio and flash
- Operate and boot device normally in a lab, monitoring activity



Copyright 2009 InGuardians, Inc.

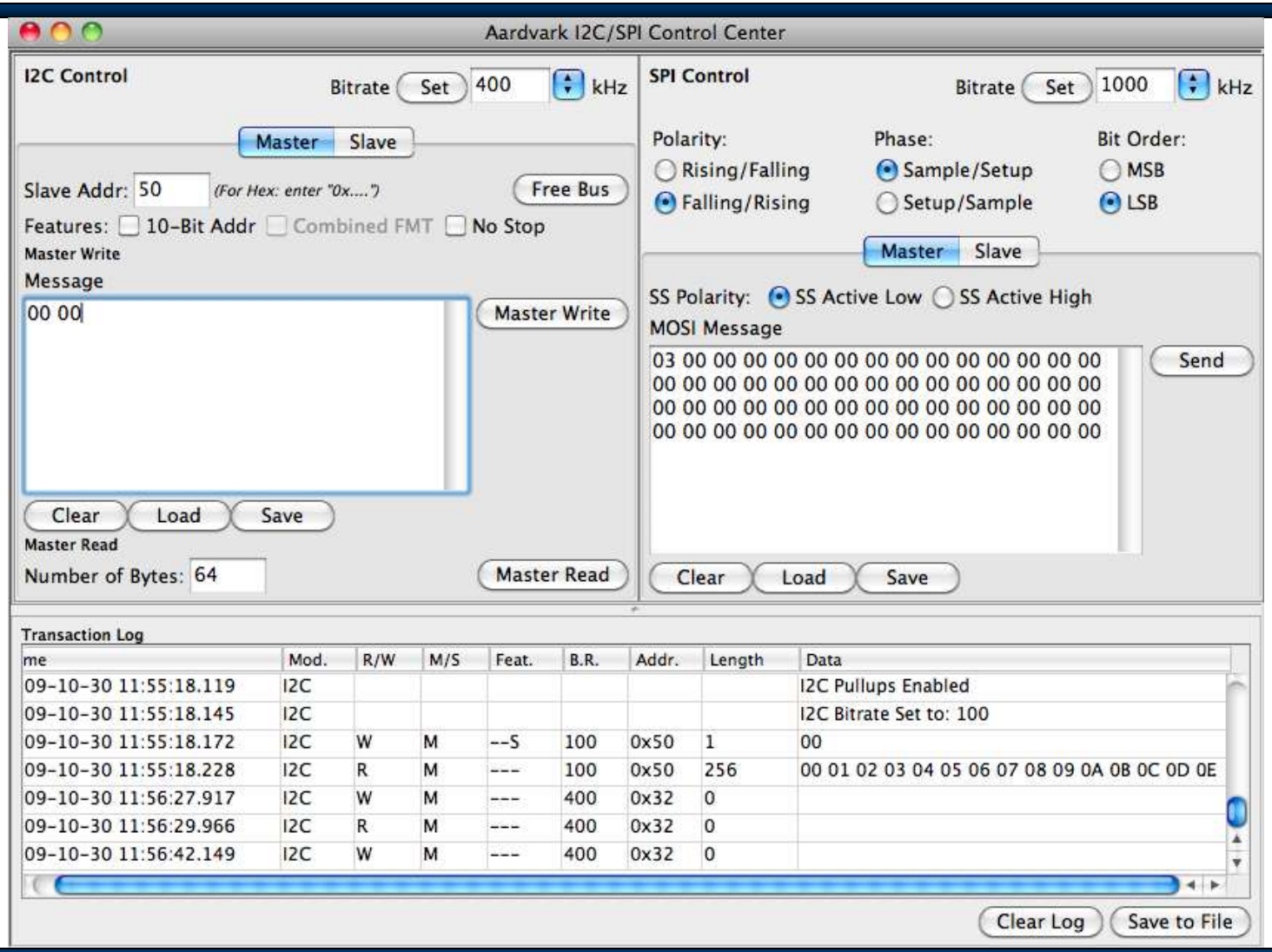
Probing Hardware





Lifting an IC's Chip Enable (CE) leg to disengage it from the target system

Total Phase Aardvark



Copyright 2009 InGuardians, Inc.

Defense: Hardware Bus Snooping

- Assume an attacker can do this: evaluate what they can do with the data
  - Limit key sharing to reduce the impact of key disclosure
- Hardware tamper-proof detection and monitoring mechanisms
  - Trigger device data wipe
- Layered epoxy on hardware slows an attacker down, not a fix
- Utilize integrated devices when possible

Copyright 2009 InGuardians, Inc.

Attack: Key Extraction

- Extract locally stored encryption key information from target device
  - Especially useful when a single key is shared across multiple devices
- Extract contents of RAM, Flash, EEPROM data
- Symmetric Key Recovery: search for stored key by repeatedly decrypting a packet until MIC is valid
- Asymmetric Key Recovery: Entropy analysis over a given dataset

Copyright 2009 InGuardians, Inc.

GoodFET + zbgoodfind

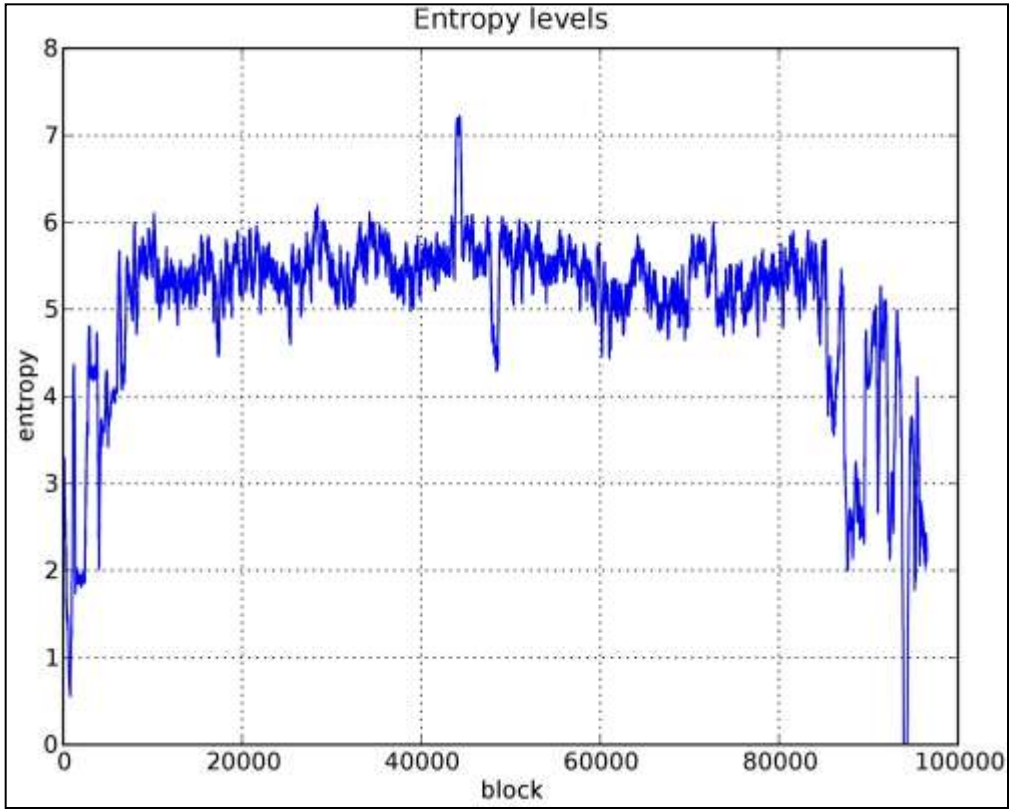
- GoodFET: Abuses vulnerability in TI, Ember radios to access RAM even when chip is locked
- zbgoodfind: Search for ZigBee key using RAM dump as a list of potential keys

```
$ 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
```

Copyright 2009 InGuardians, Inc.

Asymmetric Key Search

- Asymmetric keys have high entropy (very random)
- RAM, Flash are not random
- Graphing entropy of flash reveals high-entropy point
- Indicates asymm. key location in flash



Copyright 2009 InGuardians, Inc.

Defense: Key Extraction

- Hardware tamper-proof mechanism continue to be an important defense factor
- TPM's can protect asymmetric keys
- Limit key distribution
- Implement key rotation mechanisms

Be prepared to answer: What is my remediation strategy once the encryption keys protecting the NAN are compromised?

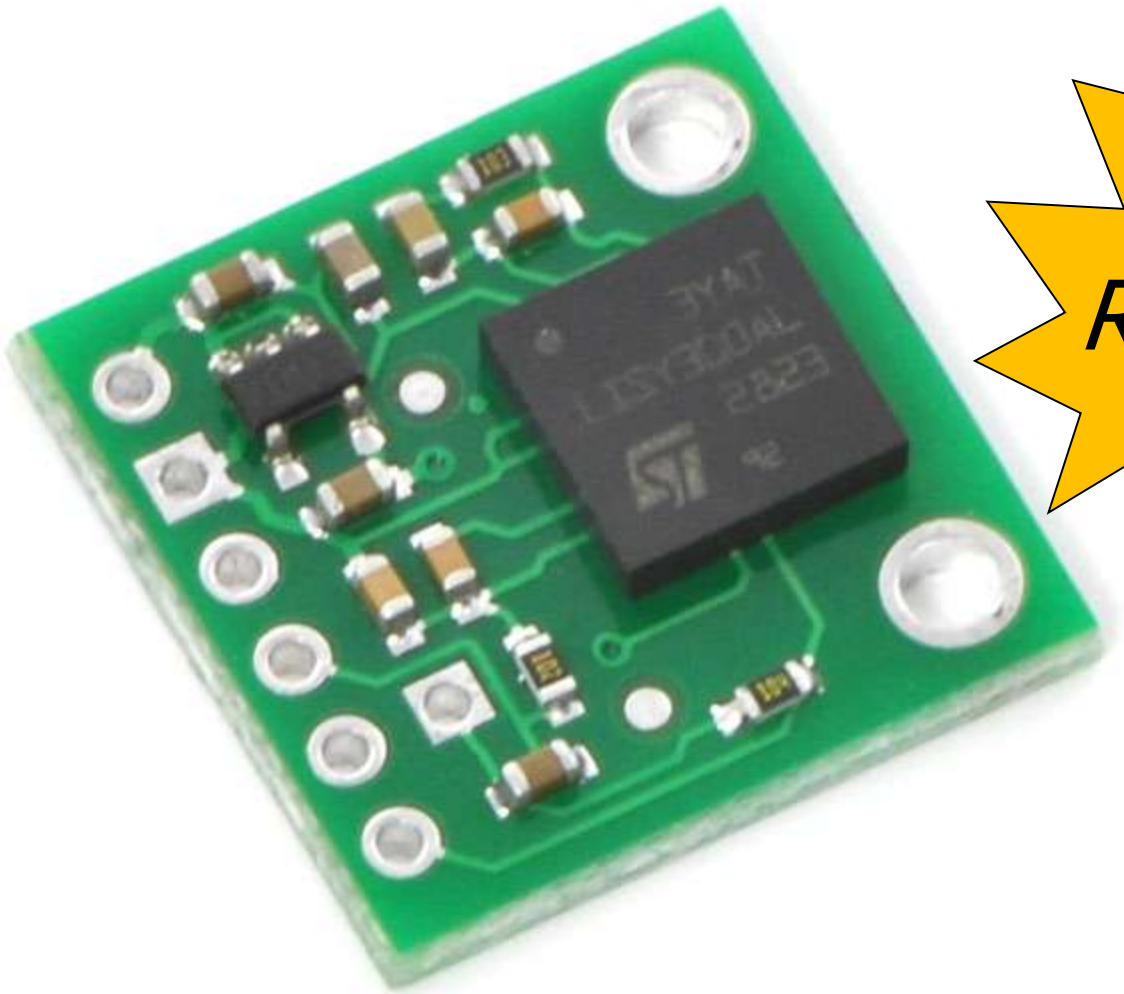
Copyright 2009 InGuardians, Inc.

Attack: Defeating Tamper Protection

- Tamper detection mechanisms aim to identify misuse of hardware
  - Tilt sensors, case opening, last gasp capacitors, epoxy, audio analyzers
  - Tripping can trigger flash/RAM scrubbing
- Attacker has many sacrificial device opportunities to learn about protections
- Attacks include melting or drilling cases to reveal circuitry, bypassing defenses
- Epoxy can be *helpful* to an attacker

Copyright 2009 InGuardians, Inc.

Tilt Sensor Circuit



Just like Rock Band!

Copyright 2009 InGuardians, Inc.

Defense: Defeating Tamper Protection

- These mechanisms are notification defenses for many hardware attacks
- The notifications generated by these systems cannot be ignored
  - Requiring potentially costly investigation to identify causes
- Events identified now could be a prelude to future attacks
  - Information gathering today, exploitation tomorrow (or 6 months from now)
- Vendors must exercise care to avoid false-positives to maintain value of this defense

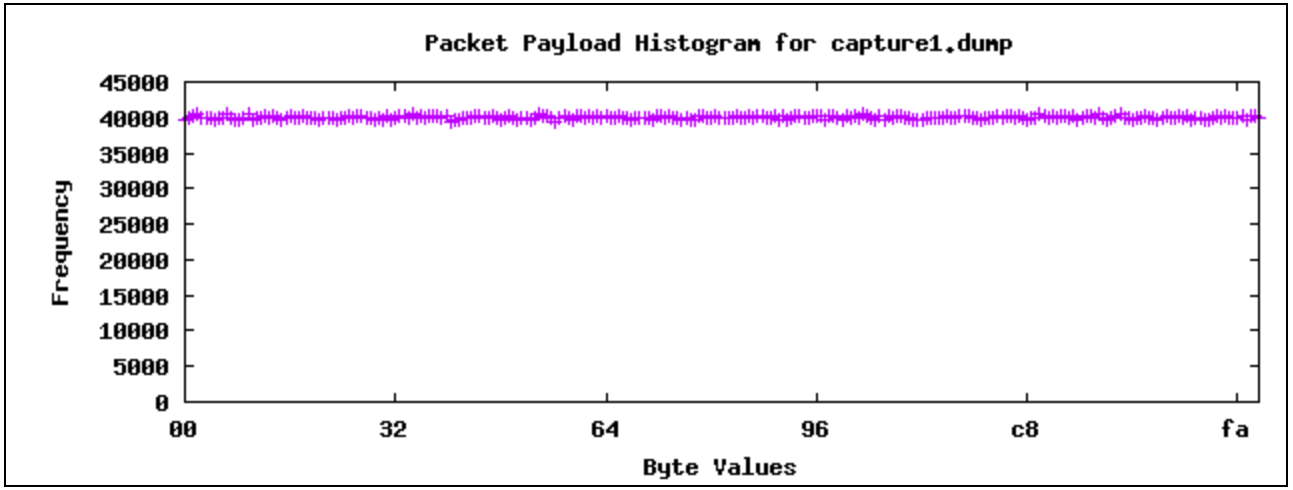
Copyright 2009 InGuardians, Inc.

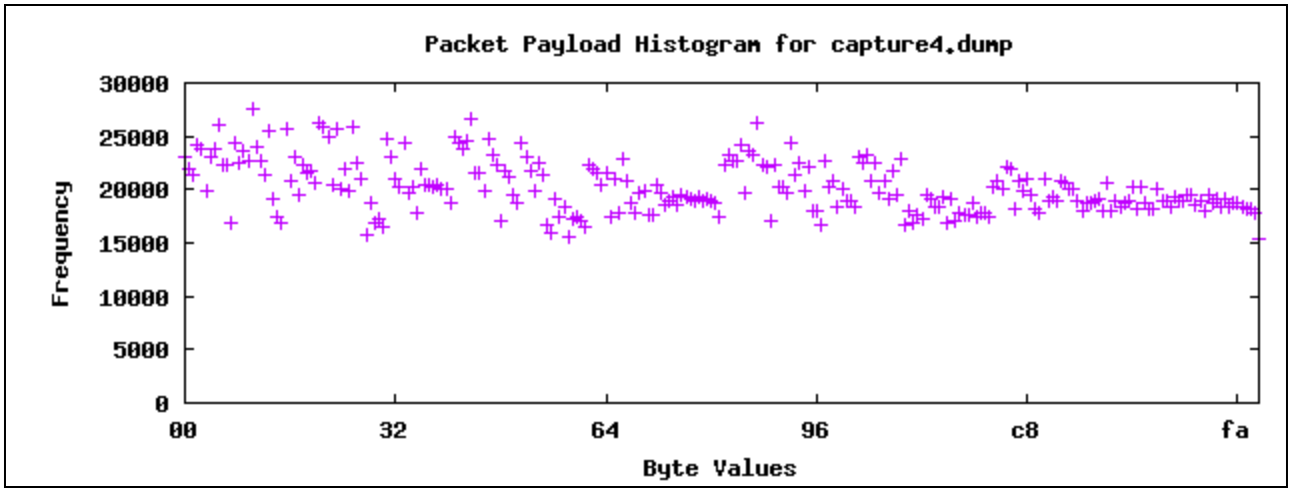
Attack: Weak Cryptography

- Many proprietary systems implement their own cryptography
  - Just because it's "AES" doesn't mean it's secure
- Exploits in insecure cipher modes, weak nonce construction, IV re-use
- Practical attacks include replaying data, decrypting packets, key recovery
- Analysis tools: Ent, visualization of RNG's, cryptographic accelerators, custom scripts

Copyright 2009 InGuardians, Inc.

Histogram Analysis





Copyright 2009 InGuardians, Inc.

AES CTR IV Attack

- AES ciphers using CTR mode effectively become a stream cipher
- Without key derivation and rotation, IV collisions compromise integrity of cipher

```
C:\>type ivcoltest.py
#!/usr/bin/env python
knownplain = "\xaa\xaa\x03\x00\x00\x00\x08\x00\x45\x00\x01\x48\x00\x01\x00\x00"
knowncip = "\x31\xb9\x84\x81\xe1\x96\x6e\x71\xd8\xa3\x39\x0c\xfb\x48\xaa\x61"
unknowncip = "\x31\xb9\x84\x81\xe1\x96\x6e\x71\xd8\xa3\x3d\x0c\xfb\xb5\xaa\x61"
print "Decrypted packet: "
for i in range(0,len(knownplain)):
    print "%02x"%( (ord(knownplain[i]) ^ ord(knowncip[i])) ^ ord(unknowncip[i]) ),
print("\n")

C:\>python ivcoltest.py
Decrypted packet:
aa aa 03 00 00 00 08 00 45 00 05 48 00 fc 00 00
```

Copyright 2009 InGuardians, Inc.

Defense: Weak Cryptography

- Design and implementation of cryptographic systems is extremely difficult
  - Avoid this if possible
  - Leverage vetted third-party encryption stack implementations
- If necessary, model system after proven protocols
  - IEEE 802.11i RSN key derivation
- Expert cryptographic review consulting

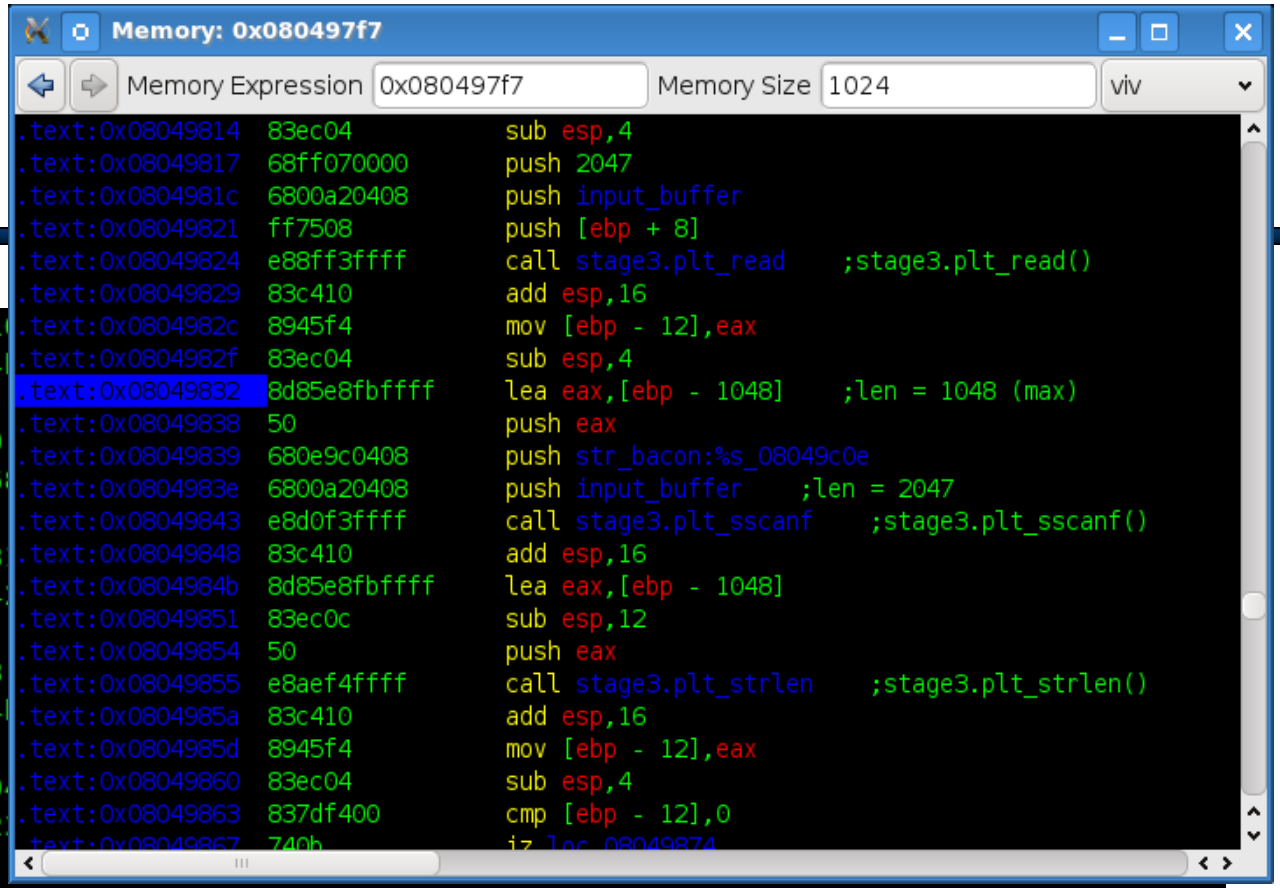
Vulnerabilities in crypto are especially hard to recover from (remember WEP?)

Copyright 2009 InGuardians, Inc.

Attack: Software Deficiencies

- Once firmware is retrieved, assess for insecure programming practices
  - Buffer overflows, off-by-ones, format strings, integer manipulation, etc.
- As a lightweight embedded system, no OS defenses (like DEP)
  - It's Windows 1995 again!
- Custom software analysis tools are required to disassemble code
- Attacker assesses code for vulnerabilities

Copyright 2009 InGuardians, Inc.




```
20004b07, (1, 0, 234881
        5384:   \x07\x4

4b17681a, (1, 167772160
        5388:   \x1a\x6

d005429a, (1, 16, 23488
        538c:   \x9a\x4

681a4b0c, (1, 134217728
        5390:   \x0c\x4

19b2380, (1, 0, 2348810
        5394:   \x80\x2

d100429a, (1, 16, 234881040, 'data processing register shift', 8192, 2)
        5398:   \x9a\x42\x00\xd1        tst     r4      r0      r10/SL  LSL   r2

b0032001, (1, 0, 234881040, 'data processing immediate shift', 8192, 0)
        539c:   \x01\x20\x03\xb0        and     r2      r3      r1  LSL   #0x0

bdf0, (1, 16, 234881040, 'data processing register shift', 8192, 2)
        53a0:   \xf0\xbd\x00\x00        and     r11/FP  r0      r0  RDR   r13/SP

4bc74, (1, 16, 234881040, 'data processing register shift', 8192, 2)
        53a4:   \x74\xbc\x04\x00        and     r11/FP  r4      r4  RDR   r12/IP
```

Copyright 2009 InGuardians, Inc.

Defense: Software Deficiencies

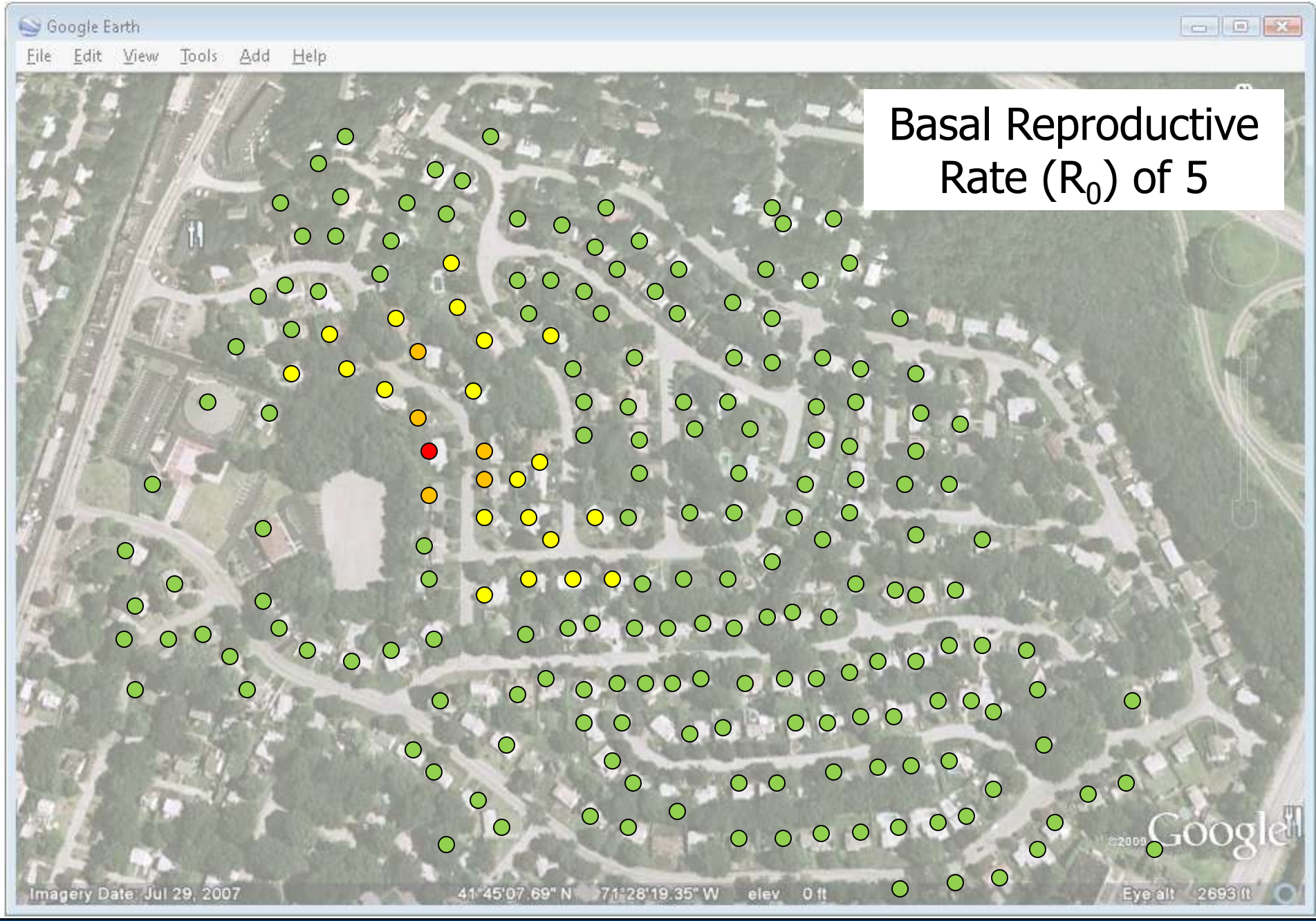
- Prevent access to firmware wherever possible
  - Leverage firmware "lock bits" or hardware fuses to prevent reads
  - Not possible with external storage
- Implement secure programming practices on all development
- Security code review services

Copyright 2009 InGuardians, Inc.

Attack: Firmware Manipulation

- Firmware update to a device is an especially vulnerable operation
  - Impersonate HAN to meter exchange
  - Leverage compromised keys to gain access to NAN
- Opportunity to supplant legitimate software with alternate code
- Reports indicate this could be an automated worm-like attack

Copyright 2009 InGuardians, Inc.



Copyright 2009 InGuardians, Inc.

Defense: Firmware Manipulation

- Botnet of AMI meters is a scary concept
- Use cryptographic signing for firmware updates
  - Public key on all remote devices validate signing key kept centrally
- Consider TPM technology to defeat hardware attacks
  - TPM is responsible for validating image, much harder to manipulate

Copyright 2009 InGuardians, Inc.

Outline

- Introduction to AMI Technology and the Attack Methodology
- Principles of AMI Assessment
- AMI Attacks and Countermeasures
- Conclusion

Copyright 2009 InGuardians, Inc.

Conclusion

- Required skills for assessing AMI cover many areas
  - Hardware, software, wireless, cryptography and more
- AMI Attack Methodology provides a guide for evaluating AMI security
  - Multiple attack vectors to be explored before validating security of AMI technology
- Through efficient testing, we can address vulnerabilities before they threaten deployments

Copyright 2009 InGuardians, Inc.

Questions?

Joshua Wright
josh@inguardians.com
401-524-2911

Matt Carpenter
matt@inguardians.com
202-517-6655

Slides from this presentation to be posted at:

www.inguardians.com

Thank You

Copyright 2009 InGuardians, Inc.