Breaking and Fixing Critical Infrastructure

Justin Searle
Managing Partner – UtiliSec
justin@utilisec.com // @meeas

What is Critical Infrastructure?

Assets that are essential for the functioning of a society and economy

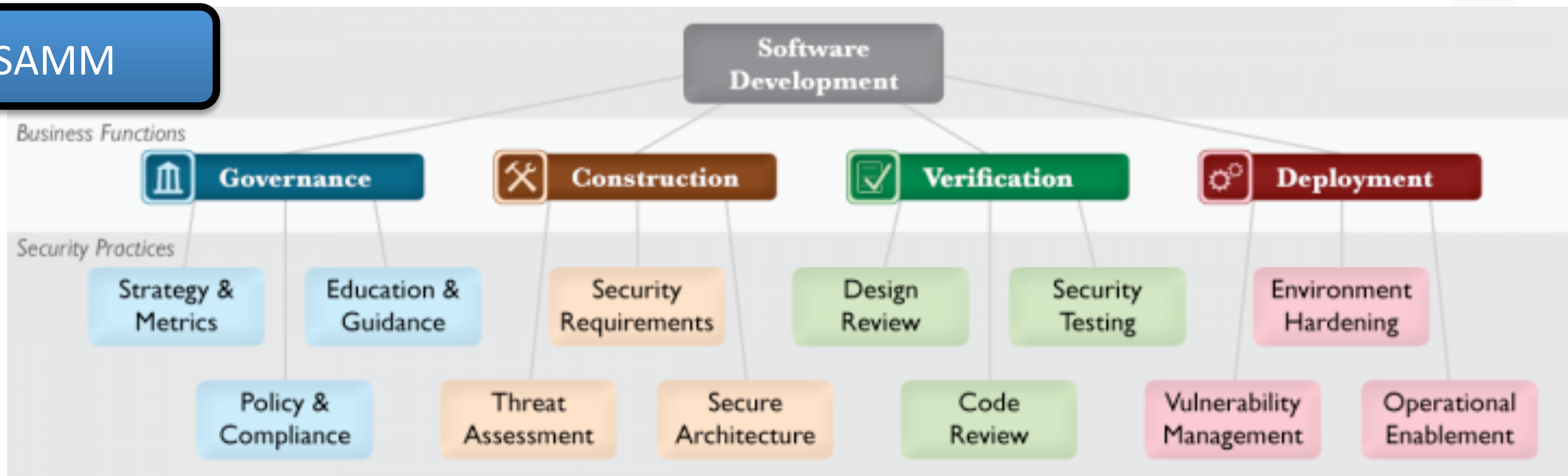
- Food and Agriculture
- Banking and Finance
- Chemical
- Commercial Facilities
- Communications
- Critical Manufacturing
- Dams
- Defense Industrial Base
- Emergency Services
- Energy
- Government Facilities
- Healthcare and Public Health
- Information Technology
- National Monuments and Icons
- Nuclear Reactors, Materials and Waste
- Postal and Shipping
- Transportation Systems
- Water

Traditional Security Models

BSIMM

The Software Security Framework (SSF)			
Governance	**Intelligence**	**SSDL Touchpoints**	**Deployment**
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

OpenSAMM

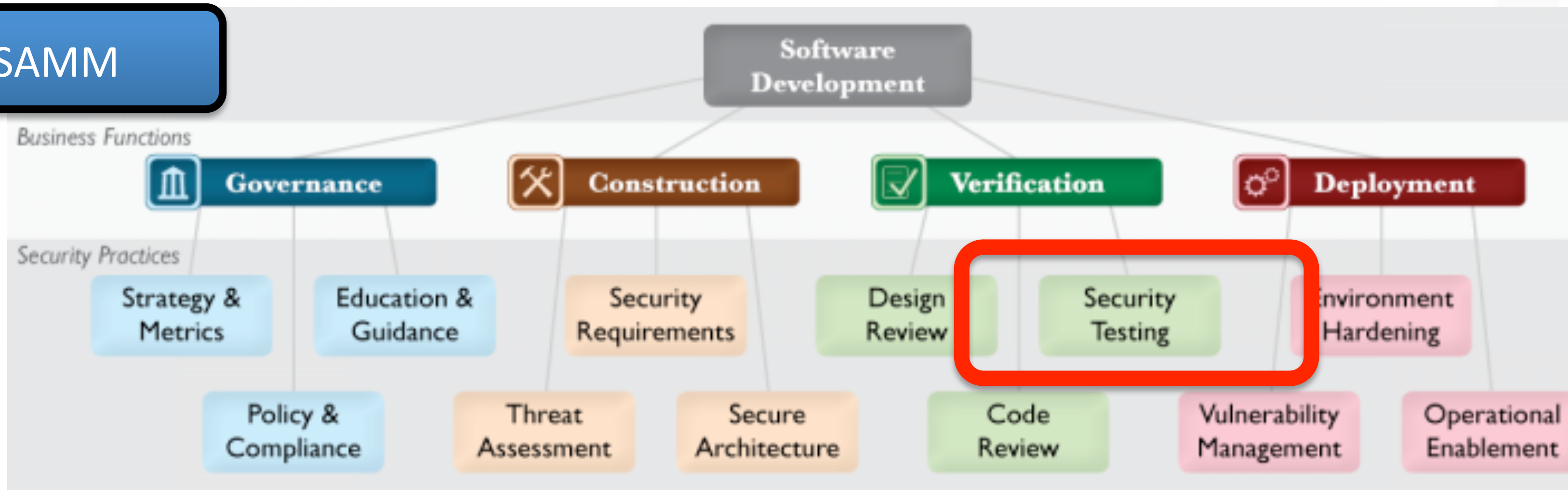


What Does These Mean?

BSIMM

The Software Security Framework (SSF)			
Governance	**Intelligence**	**SSDL Touchpoints**	**Deployment**
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

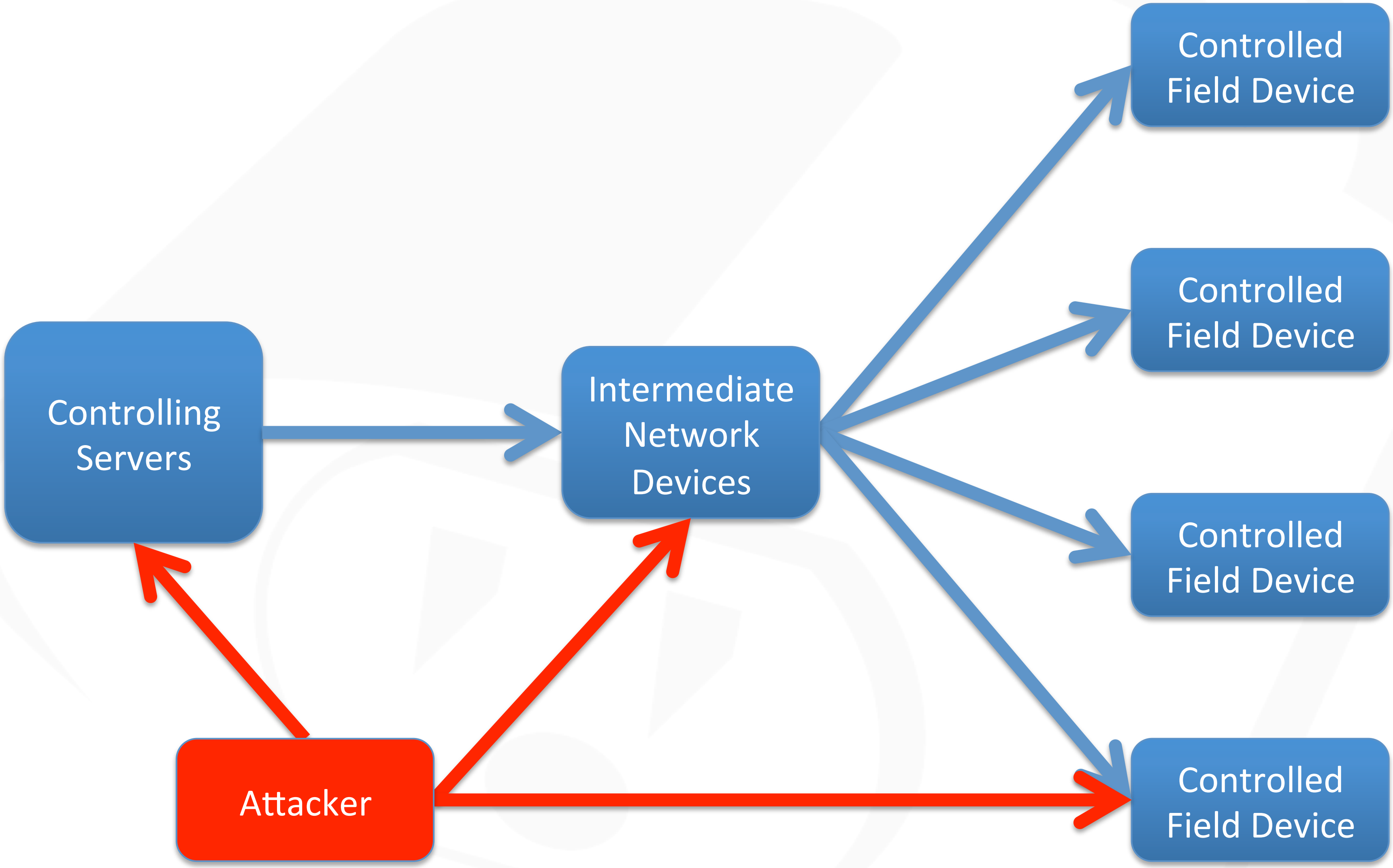
OpenSAMM



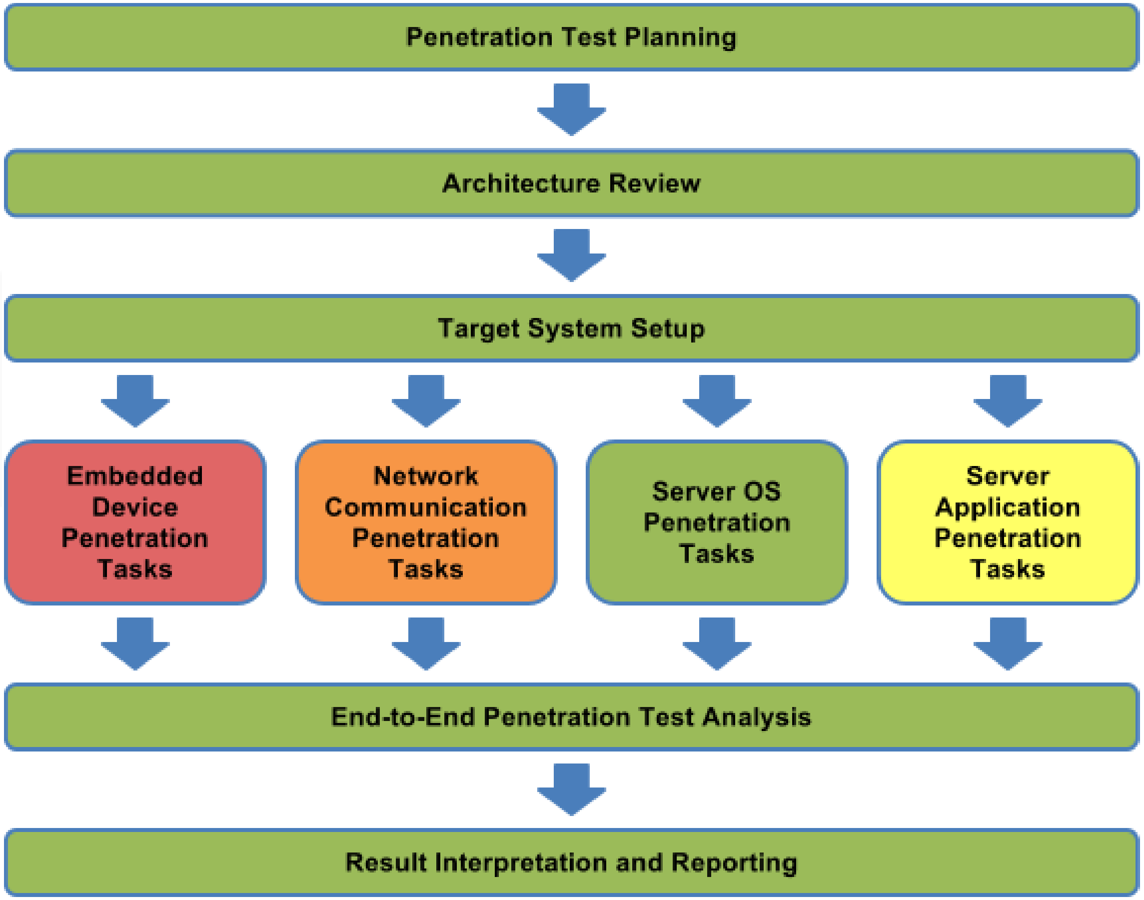
Breaking In Before Attackers Do

- Penetration testing is the MOST underutilized process in these models by most companies across the world!
- Most money and time is spent on governance, awareness, architecture, and defense
  - Most companies may check to see if there their front door is locked, but rarely check to see if they have more than one door, or where those extra doors lead...
  - And companies are always surprised when they are compromised...
- Continuous, agile penetration testing helps to address this
  - This should go far beyond vulnerability scanning, include exploitation
  - Hire or train at least one competent employee
  - Set this as their sole priority, testing both the outside perimeter and inside network segments, with targets changing at least weekly
  - Have them find and follow the paths of least resistance like attackers do
  - Hand off vulnerability remediation and tracking to a different individual

Critical Infrastructure Control



Penetration Testing Methodology



- Green: Tasks most frequently and require the most basic of penetration testing skill
- Yellow: Tasks commonly performed and require moderate penetration testing skill
- Orange: Tasks that are occasionally performed but require higher levels of expertise
- Red: Tasks performed infrequently and require highly specialized skills

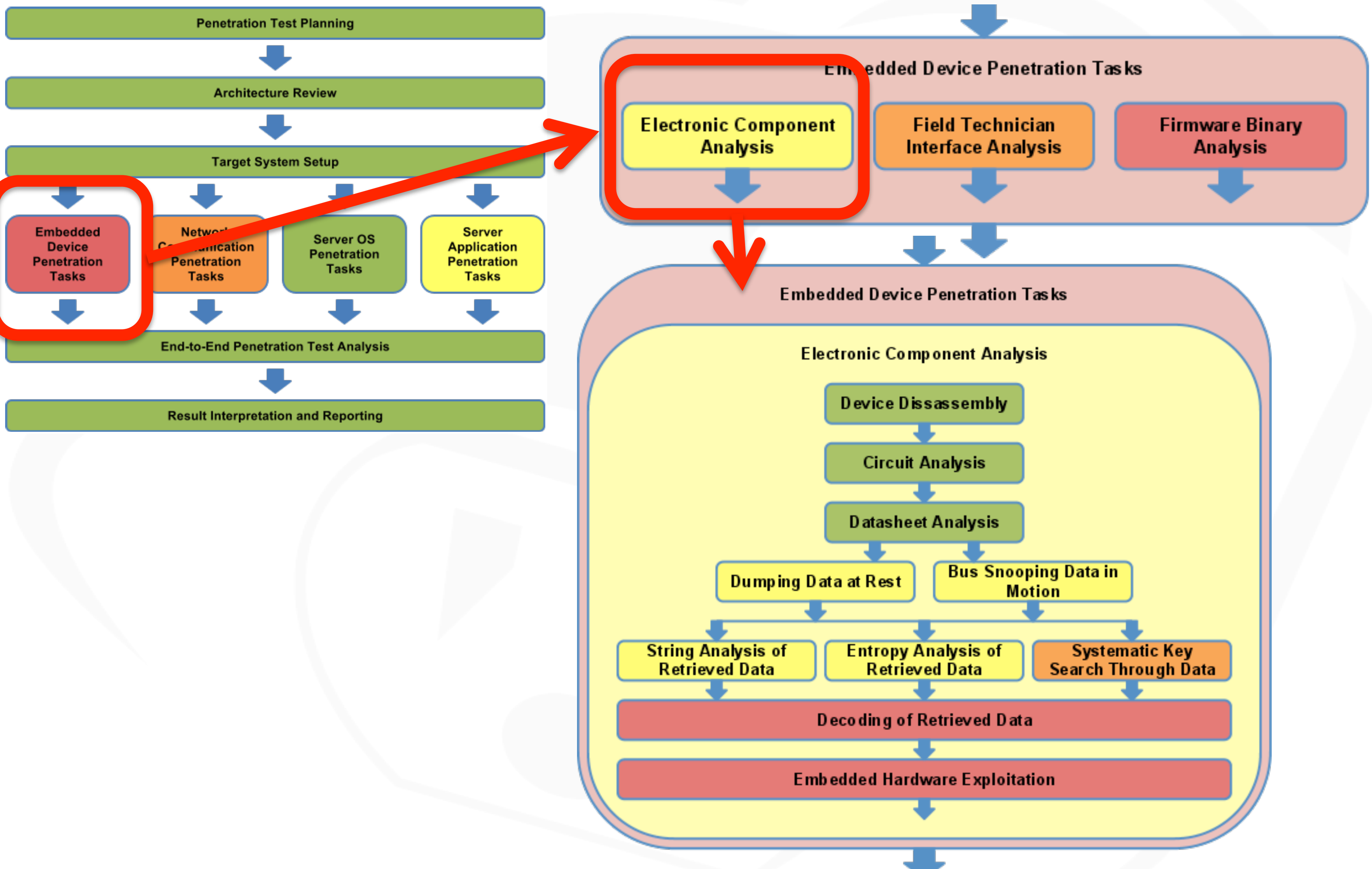
Can Attackers Reach Field Assets?

- Many field deployed Critical Infrastructure devices have tamper detection and tamper protection mechanisms
- Locked cases and enclosures
  – usually trivial to pick
- Tilt and vibration detection components
  – can often be disabled before triggering once you locate them
- Cameras
  – often not monitored
  – long response times if they trigger

What To Do With Field Devices?

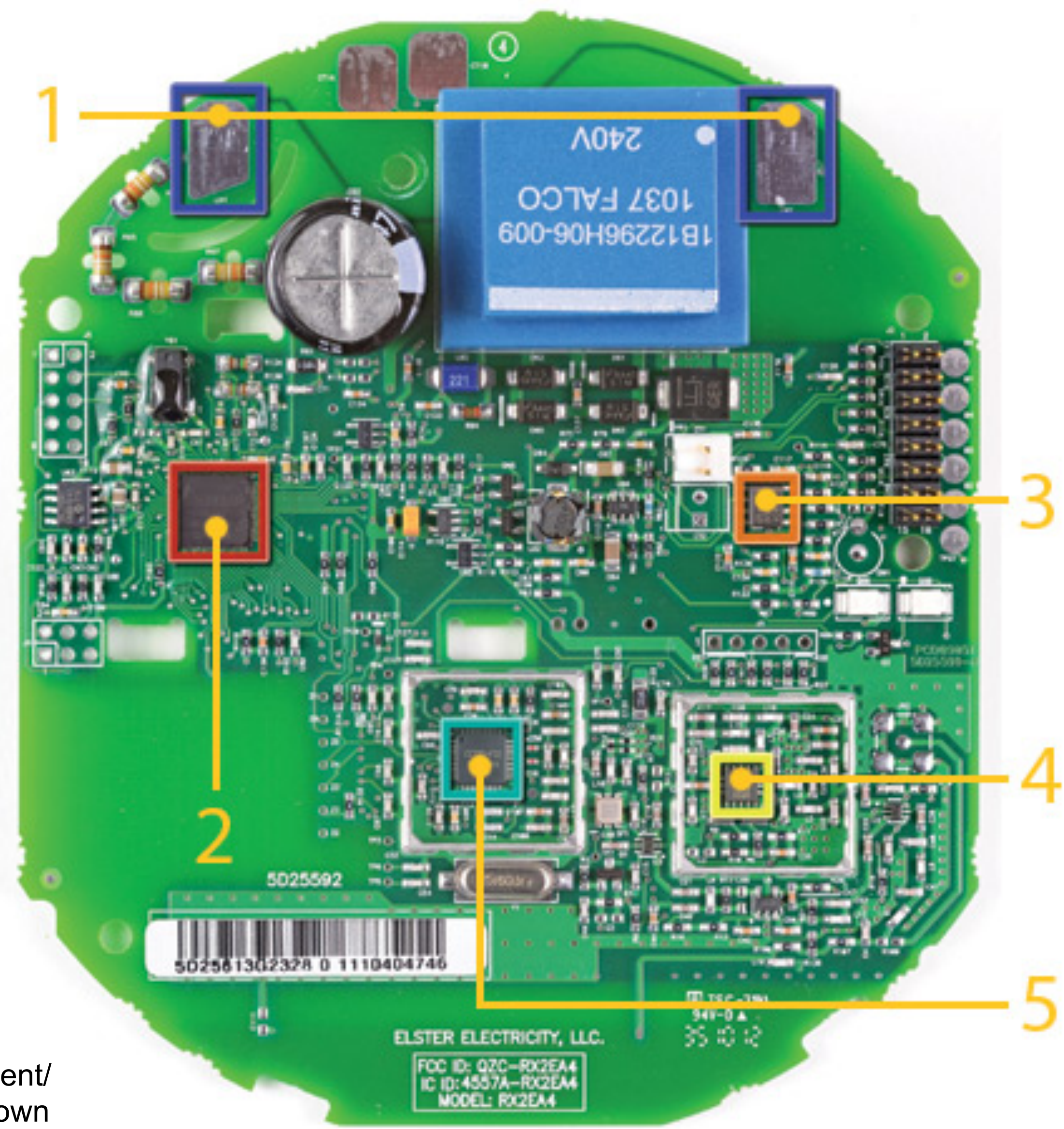
- Attacking data at rest
  - Power down the device, expose its circuit board, and interact directly with each component
  - Extract contents of accessible RAM, Flash, and EEPROM
  - Identify cryptography keys or firmware
- Attacking data in motion
  - Boot and normally operate the device in a lab, monitoring bus activity between major chips (MCU, Radio, Flash, RAM)
  - Crypto keys can often be found in key load operations between a microcontroller and crypto accelerator
  - Firmware can often be found in boot processes (between Flash and MCU) and firmware updates (between Radio, MCU, and Flash)

Embedded Device Pentest Tasks



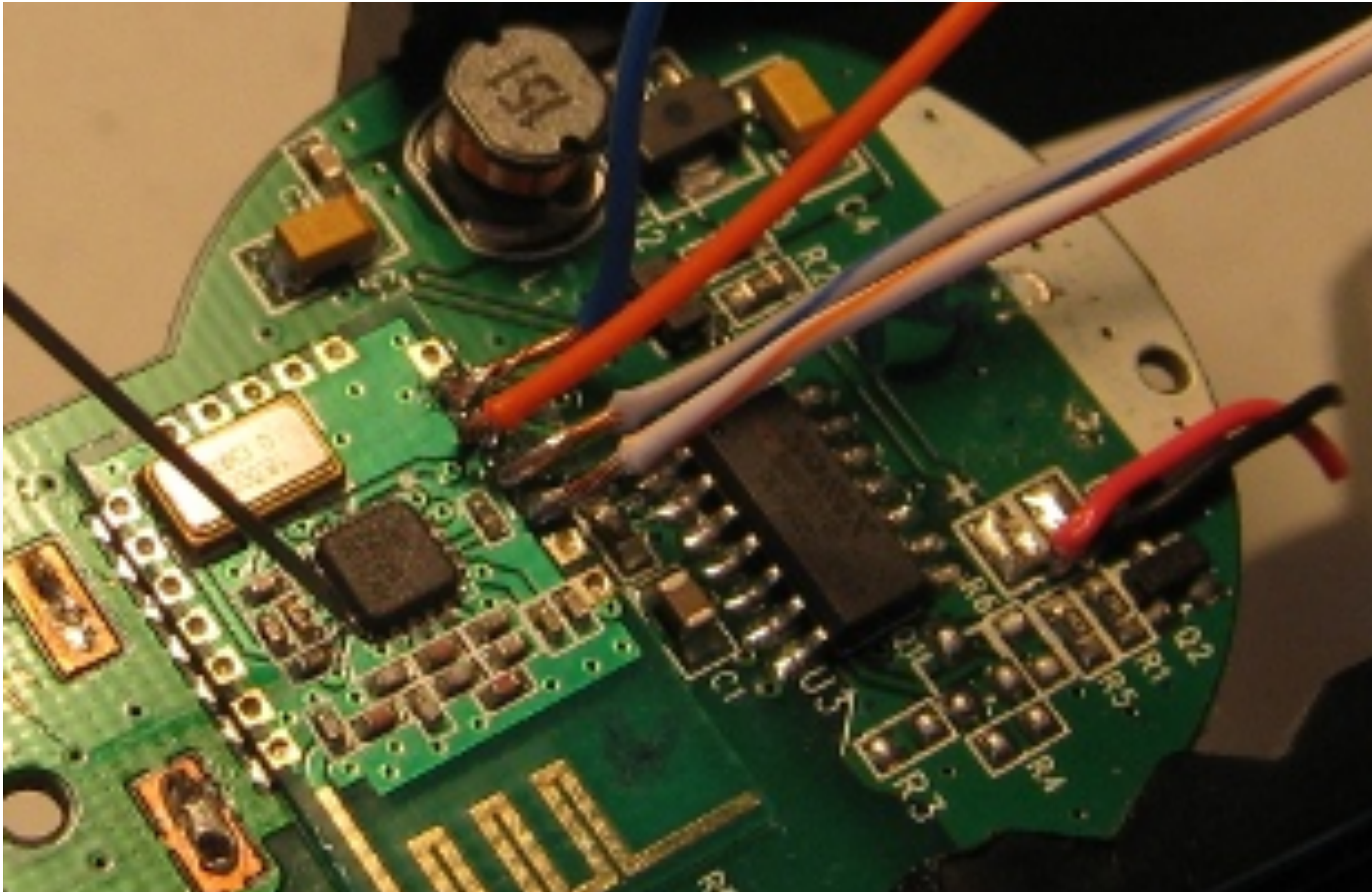
Embedded Field Device Disassembled

1. 240v Connections
2. Microcontroller
   (Teridian 71M6531F SOC)
3. Dual Operational Amplifier
   (LM2904)
4. ISM Band RF Amplifier
   (RFMD RF2172)
5. ISM Band RF
   (TI CC1110F32)



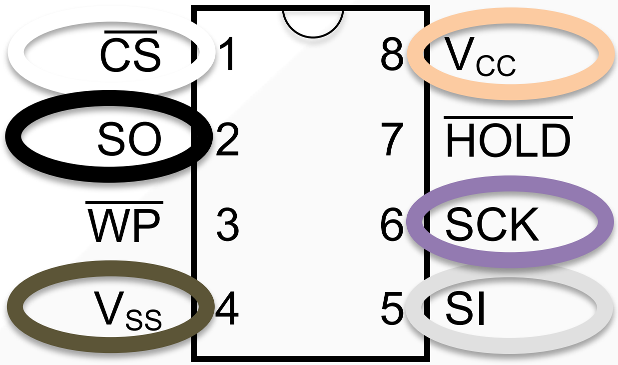
Source - http://www.edn.com/design/power-management/4368353/What-s-inside-a-smart-meter-iFixit-tears-it-down

Embedded Device Testing

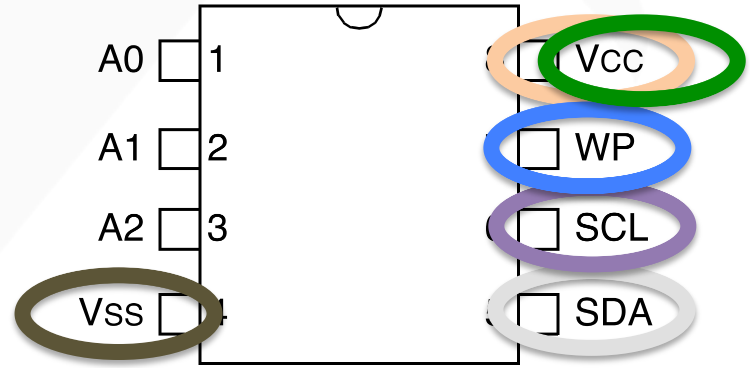


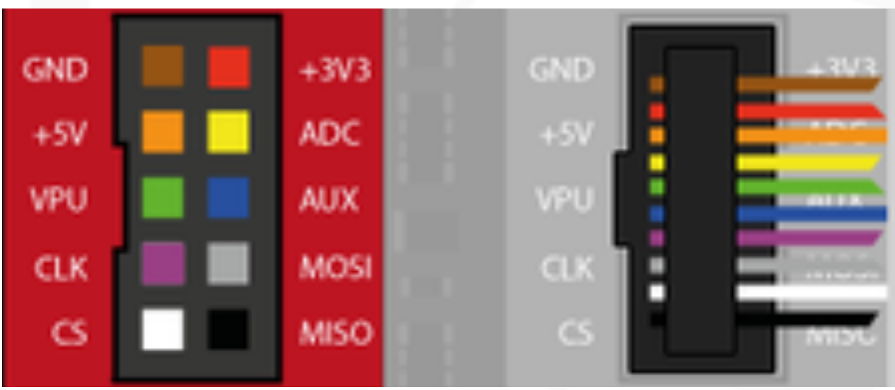
Bus Pirate to EEPROMs

CAT25080 – SPI



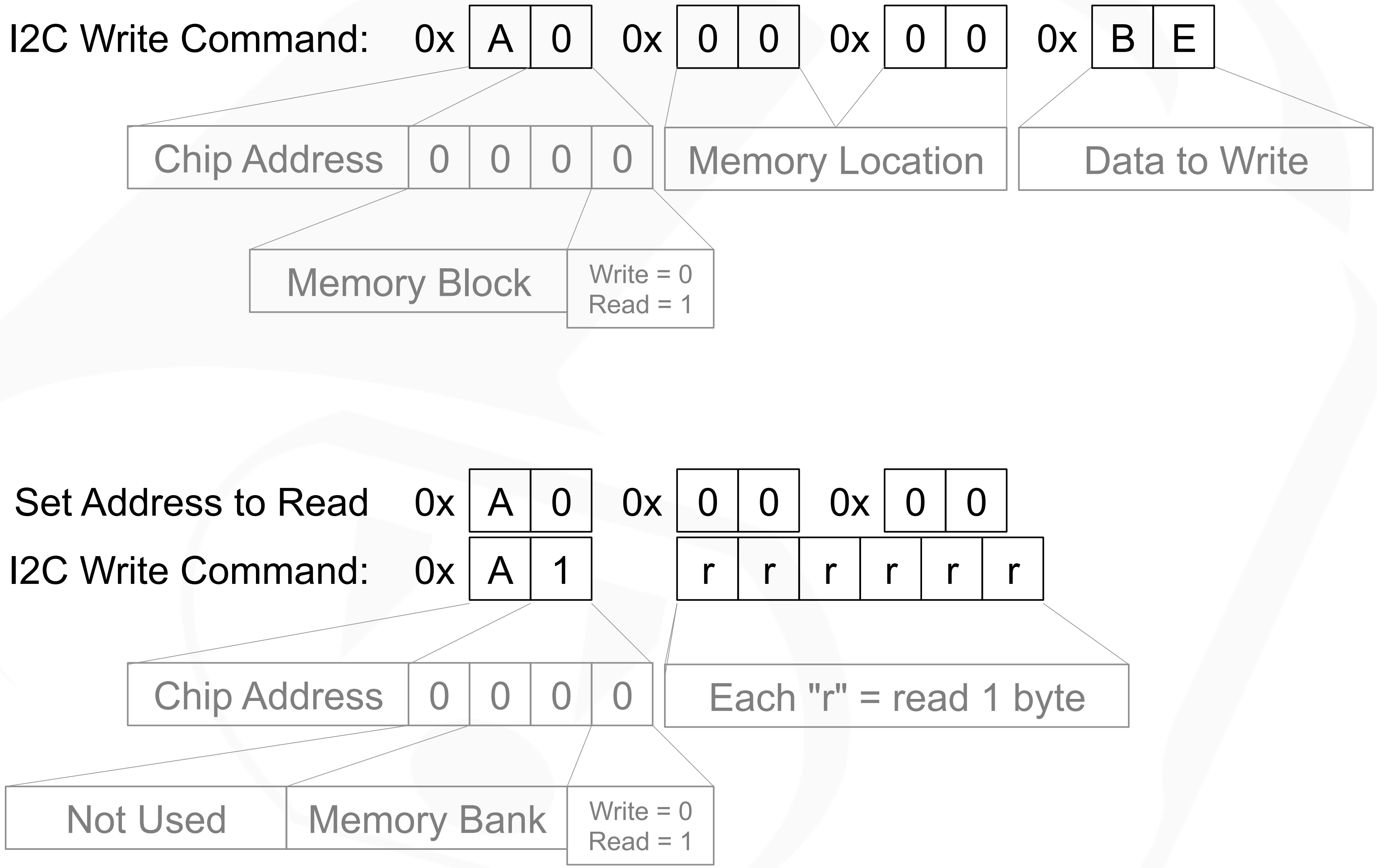
24LC088 – I2C



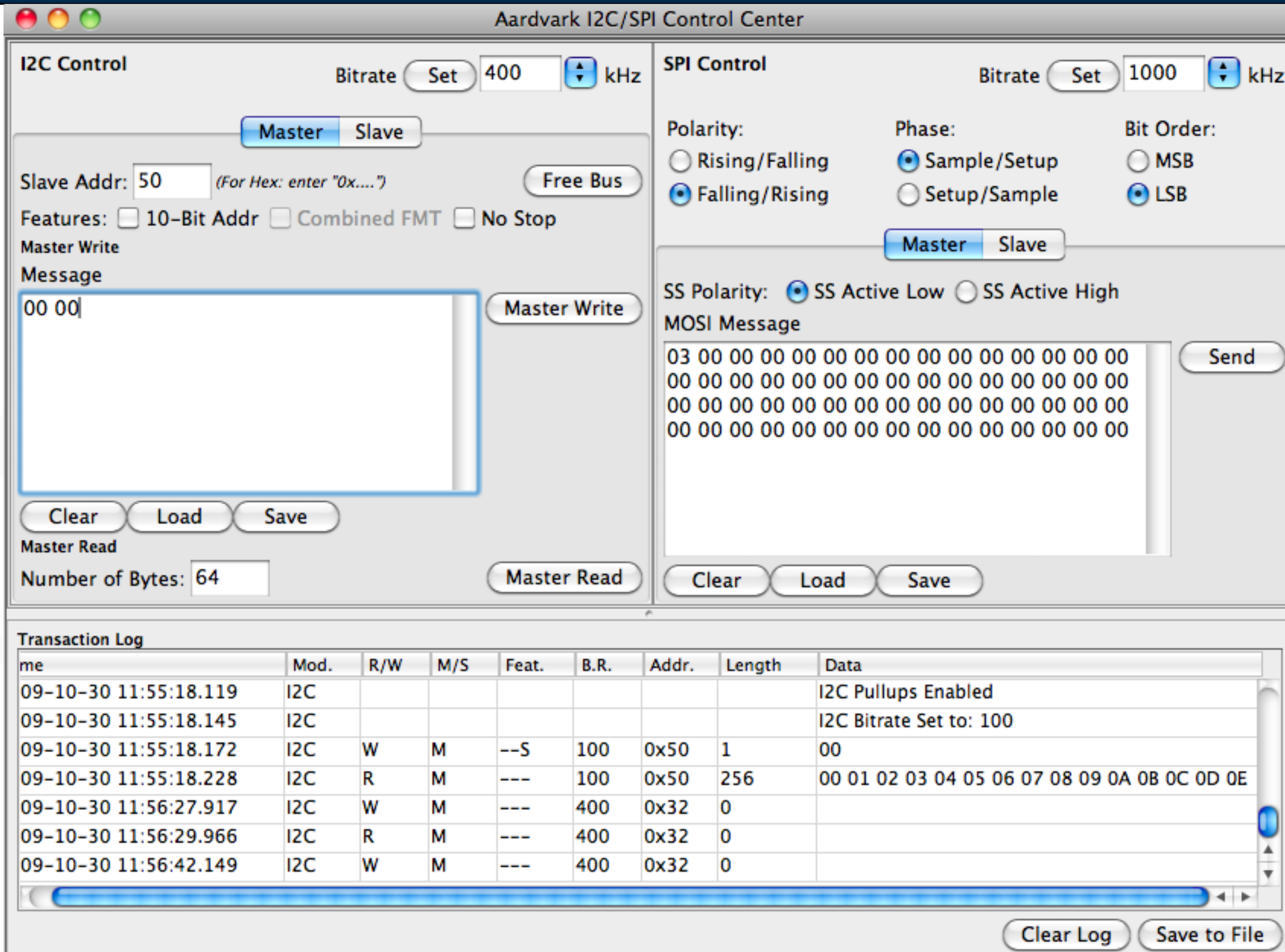


Mode	MOSI	CLK	MISO	CS
HiZ				
1-Wire	OWD			
UART	TX		RX	
I2C	SDA	SCL		
SPI	MOSI	CLOCK	MISO	CS
JTAG	TDI	TCK	TDO	TMS

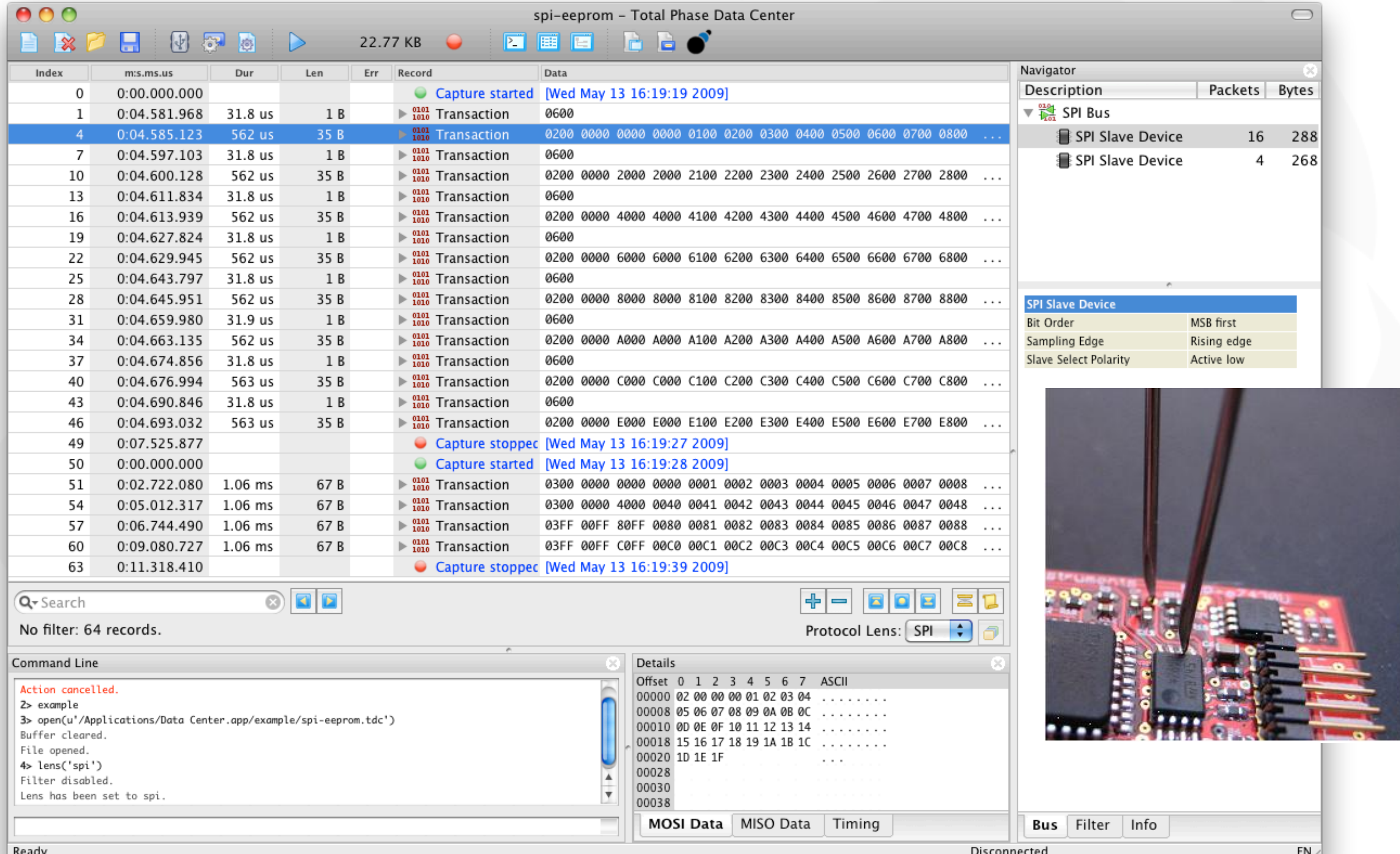
I2C Protocol



Dumping EEPROMs and Flash

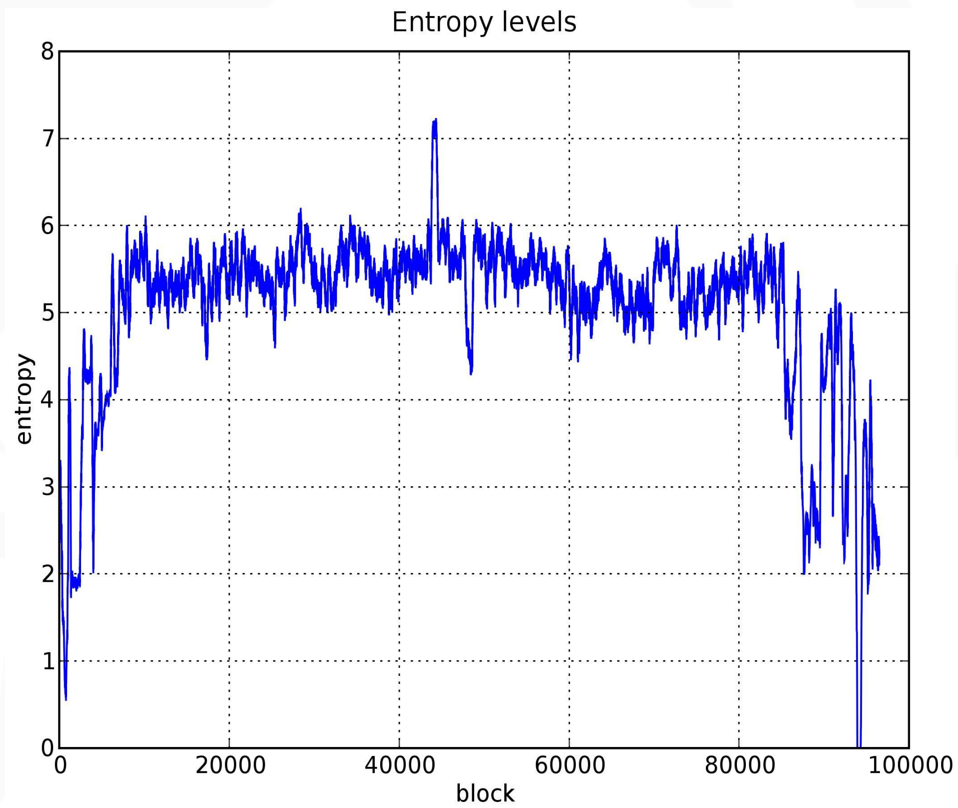


Task: Bus Snooping Data in Motion



Using Entropy to Find Keys

- Asymmetric keys have high entropy (very random)
- RAM and Flash is filled with non-random data
- Graphing entropy of flash reveals a spike in randomness
- This spike is the location of the asymmetric key in flash



Systematic Crypto Key Search

- Perform basic string searches for obvious keys
- Develop custom tools to do more advanced searches:
  - 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
  - Combined they can recover the ZigBee network key

```
$ sudo goodfet.cc dumpdata memdump.hex
Target identifies as CC2430/r04.
Dumping data from e000 to ffff as chipcon-2430-mem.hex.
...
$ objcopy -I ihex -O binary memdump.hex memdump.bin
$ zbgoodfind -R 802154_encr_sample.dcf -f memdump.bin
zbgoodfind: searching the contents of 802154_encr_sample.dcf for
encryption keys with the first encrypted packet in memdump.bin.
Key found after 6264 guesses:  c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc
cd ce cf
```

Termineter & John Commor

- Open a terminal to the "john_commor_c1218" folder and start the serial emulation script

```
sudo ./pts_ports.sh
```

- Open a new terminal and start the virtual smart meter replacing the "???" with the port numbers displayed on the last command

```
sudo python john_commor_c1218.py -e ??? -t ???
```

- Open a new terminal and use Termineter to read the first c12.22 table on the meter. Replace the "???" with the port you typed after the "-t" in the previous command

```
sudo ./termineter2.py
set connection /dev/pts/???
connect
use read_table
set tableid 1
run
```

- Now use Termineter to read tables 2 and 3

INSTRUCTOR LED LAB!!!

Decompiled Code

UtiliSec

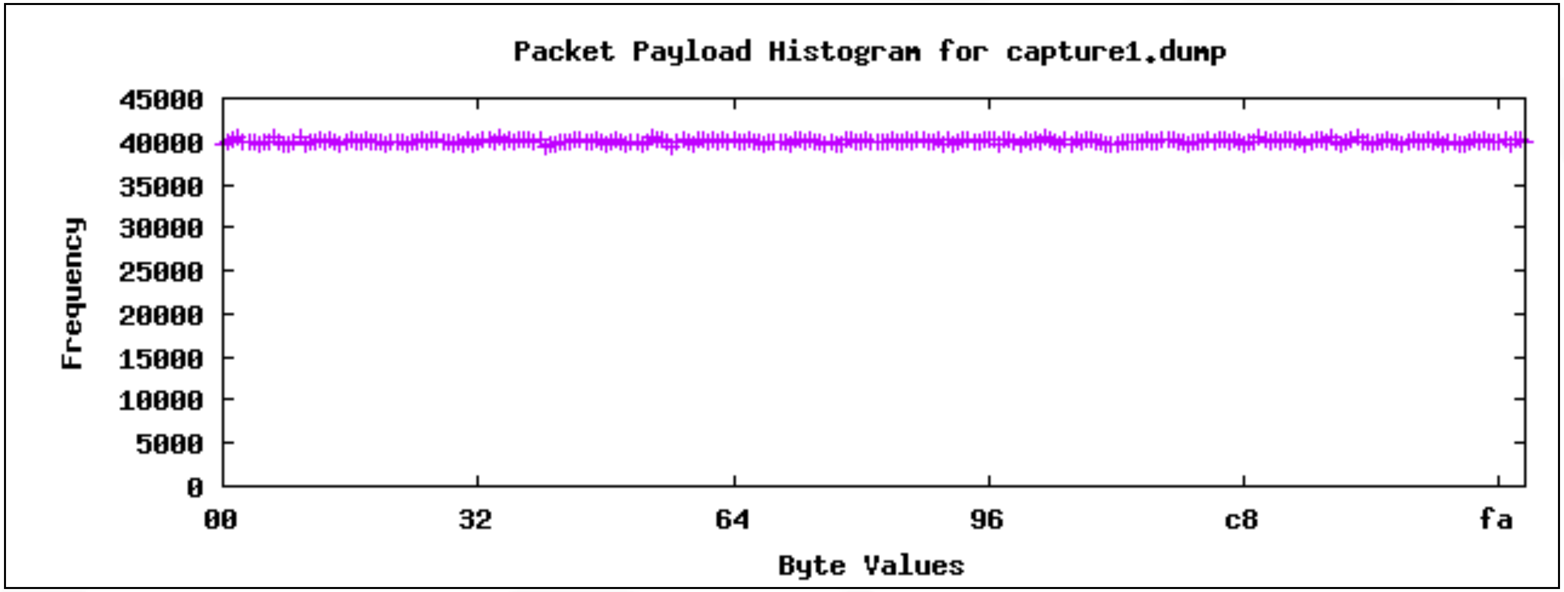
Memory: 0x080497f7

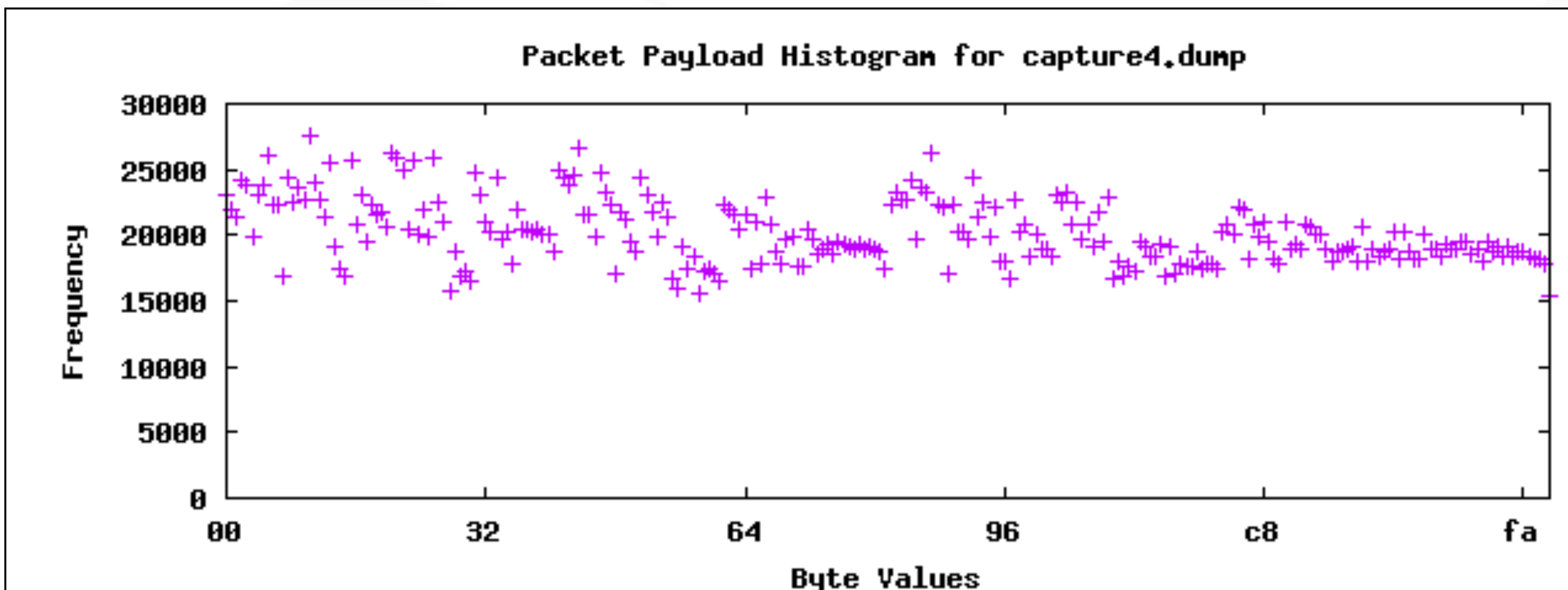
Memory Expression 0x080497f7 Memory Size 1024 viv

```
.text:0x08049814 83ec04         sub esp,4
.text:0x08049817 68ff070000     push 2047
.text:0x0804981c 6800a20408     push input_buffer
.text:0x08049821 ff7508         push [ebp + 8]
.text:0x08049824 e88ff3ffff     call stage3.plt_read      ;stage3.plt_read()
.text:0x08049829 83c410         add esp,16
.text:0x0804982c 8945f4         mov [ebp - 12],eax
.text:0x0804982f 83ec04         sub esp,4
.text:0x08049832 8d85e8fbffff   lea eax,[ebp - 1048]      ;len = 1048 (max)
.text:0x08049838 50             push eax
.text:0x08049839 680e9c0408     push str_bacon:%s_08049c0e
.text:0x0804983e 6800a20408     push input_buffer    ;len = 2047
.text:0x08049843 e8d0f3ffff     call stage3.plt_sscanf     ;stage3.plt_sscanf()
.text:0x08049848 83c410         add esp,16
.text:0x0804984b 8d85e8fbffff   lea eax,[ebp - 1048]
.text:0x08049851 83ec0c         sub esp,12
.text:0x08049854 50             push eax
.text:0x08049855 e8aef4ffff     call stage3.plt_strlen     ;stage3.plt_strlen()
.text:0x0804985a 83c410         add esp,16
.text:0x0804985d 8945f4         mov [ebp - 12],eax
.text:0x08049860 83ec04         sub esp,4
.text:0x08049863 837df400       cmp [ebp - 12],0
.text:0x08049867 740b           jz loc_08049874
```

```
20004b07, (1, 0, 234881040, 'd
        5384:    \x07\x4b\x00\x

4b17681a, (1, 167772160, 23488
        5388:    \x1a\x68\x17\x

d005429a, (1, 16, 234881040, '
        538c:    \x9a\x42\x05\x

681a4b0c, (1, 134217728, 23488
        5390:    \x0c\x4b\x1a\x

19b2380, (1, 0, 234881040, 'da
        5394:    \x80\x23\x9b\x

d100429a, (1, 16, 234881040, '
        5398:    \x9a\x42\x00\x

b0032001, (1, 0, 234881040, '
        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
```

Task: Cryptographic Analysis





Insecure Block Cipher Modes

- AES ciphers using CTR mode effectively become a stream cipher
- Without key derivation and rotation, IV collisions compromise integrity of cipher
- The following example show a failure to use new IV

```
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
```

Conclusion

- Critical Infrastructure needs more testing!
  - Testing needs to be constant and remain agile
  - Testing needs to go deeper and include field controlled assets
- Research is needed in all major areas of security. Apply your talents and find your niche!
- Pentesting methodology for energy sector is publically available
  - Download it at http://www.smartgrid.epri.com/NESCOR.aspx
  - Provides a detailed methodology for performing penetration tests from controlling servers all the way to the controlled embedded field devices
  - Methodology is directly applicable for any critical infrastructure organization and the systems they use to control critical assets
  - Black Hat's *Pentesting Smart Grid and SCADA* course based on this methodology
- Other Energy sector specific documents that may help
  - NIST: http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7628
  - ASAP-SG: http://www.smartgridipedia.org/index.php/ASAP-SG

Contact Information

www.utilisec.com
sales@utilisec.com

Justin Searle
personal: justin@meeas.com
work: justin@utilisec.com
cell: 801-784-2052
twitter: @meeas