COMPROMISING INDUSTRIAL FACILITIES FROM 40 MILES AWAY

Lucas Apa
Carlos Mario Penagos

IOActive
COMPREHENSIVE INFORMATION SECURITY SERVICES

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

About Us

Lucas Apa

Argentina

Carlos Penagos

Colombia

Vulnerability Research

Exploitation

Cryptography

Reverse Engineering

ICS/SCADA

Agenda

- Motivation
- Industries and Applications
- Wireless Standards
- Journey of Radio Encryption Keys
- Vendor1 Wireless Devices
- Vendor2 Wireless Devices
- Vendor3 Wireless Devices

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

Motivation

- Critical Infrastructures becoming targets
- Insider attacks (Lately)
  - Devices connected to Internet
  - 0days to reach the PLC, RTU, HMI…
- Stealth and precise attacks
- Incident response at hazardous sites

Industrial Wireless Automation

- Copper wires are used to monitor and control
  - Corrosion, Ductility, Thermal Conductivity
  - Cost of wires, trenching, mounting and installation
- Industrial Wireless Solutions
  - Eliminate cost of hardwiring, logistics, installation
  - Heavy machinery involved
  - Remote control and administration (Geography)
  - Minimize Safety Risk & Dangerous Boxes
  - Adds durability

Industries and Applications

- Plunger lift/artificial lift optimization
- Well-head automation
- RTU/EFM I/O extensions
- Cathodic protection monitoring
- Hydrogen sulfide (H2S) monitoring

Oil & Gas

Refined Petroleum
Petrochemicals

- Tank level monitoring
- Pipeline cathodic protection
- Rectifier voltage monitoring
- Gas/liquid flow measurement
- Pipeline pressure and valve monitoring

Industries and Applications (2)

- Transformer temperature
- Natural gas flow
- Power outage reporting
- Capacitor bank control
- kV, Amp, MW, MVAR reading

Energy - Utilities

Waste &
Waste Water

- Remote pumping stations
- Water treatment plants
- Water distribution systems
- Wastewater/sewer collection systems
- Water irrigation systems/agriculture

Industrial Wireless Challenges

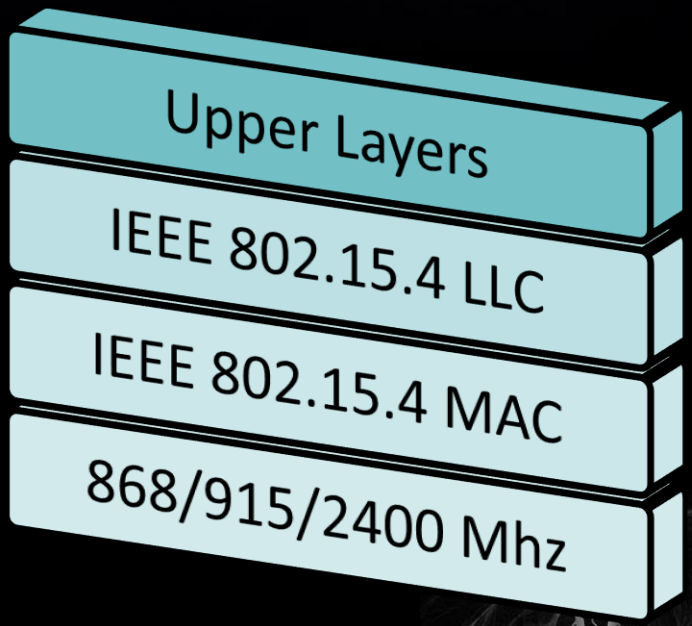
- Defeat electromagnetic interference (EMI)
- Handle signal attenuation and reflections
- Reliability is far more important than Speed
- Higher transmitter power levels
- Site surveys to assess the consistency and reliability of the plant
- Mainly using 2.4Ghz or 900Mhz (ISM Band)
- No “business” protocols

Cryptographic Key Distribution (WSN)

- Distribute secrets on a large number of nodes
- Base stations with clusters surrounding
- Limitations:
  - Deployment in public or hostile locations
  - Post-deployment knowledge
  - Limited bandwidth and transmission power
- Methods for crypto key distribution:
  - Out-of-band
  - In-band
  - Factory pre-loaded

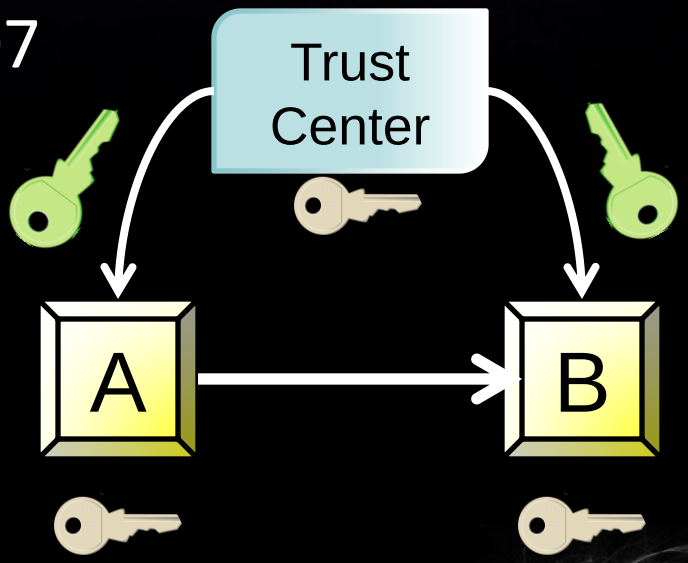
IEEE 802.15.4 Standard

- Wireless Radios (Low Power/Speed)
- Set the encryption algorithm and AES Key
- Upper Layer Responsibility
- Each node can have an ACL
- MAC for upper layers:
  - ZigBee
  - WirelessHart
  - ISA SP100
  - IETF IPv6 - LoWPAN



ZigBee 2007 (Standard Security Mode)

- Suite of high level communication protocols
- Based on IEEE 802.15.4 (Low level layers)
- ISM radio bands
- Trust Center introduced in 2007
- Network Key (AES 128-bit)
  - Pre-installed (Factory Installed)
  - Individually Commissioned (Commissioning tool)
  - Managed by the Trust Center



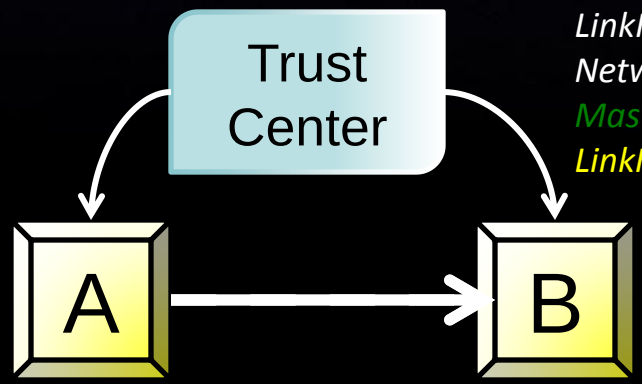
ZigBee Pro 2007 (High Security Mode)

- Many enhancements
- More memory requirements
- New keys introduced

① Master Key
- Unsecured Transport ☹
- Out-of-band Technique ☺
- Secure other keys

② Link Key
- Unicast
- Unique between nodes

③ Network Key
- Regenerated at Intervals
- Needed to join the NWK



MasterKey_TA
LinkKey TA
NetworkKey
MasterKey_TB
LinkKey TB

MasterKey_TA
LinkKey TA
NetworkKey
MasterKey_AB
LinkKey AB

MasterKey_TB
LinkKey TB
NetworkKey
MasterKey_AB
LinkKey AB

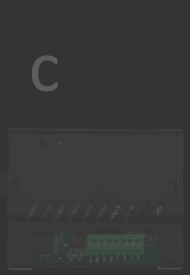
The Journey of Radio Encryption Keys

Radio

DeviceVendorID Key in Firmware

No Encryption Key

Device



Per-Client Encryption Key

Device Company Encryption Key

No Encryption Key

End User

Change Encryption Key

Per-Client Encryption Key

Device Company Encryption Key

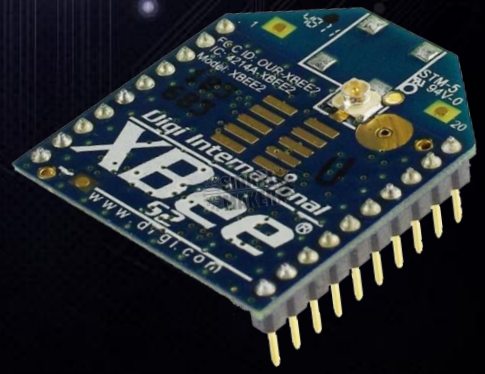
Change Encryption Key

Set Encryption Key

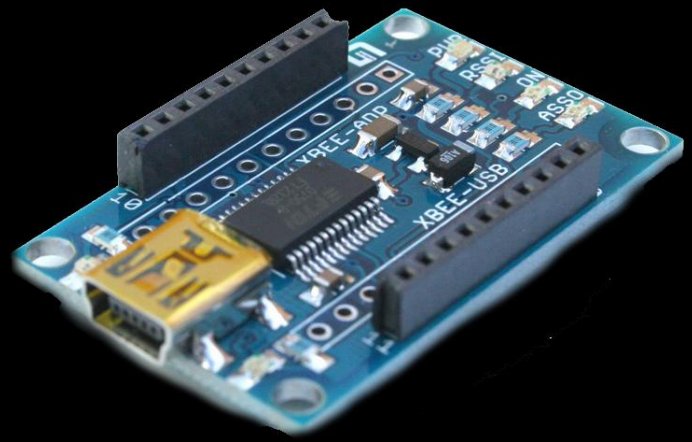
No Encryption key

Reusing Radio Keys

- End-User Node Key Storage
  - Shared Secret
  - Same Firmware or Same Radio Key



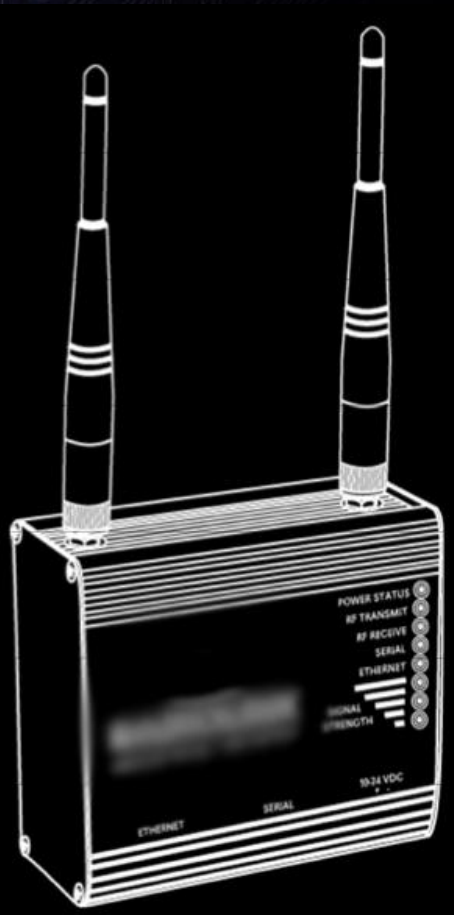
- **Device Company Key** attack
  1. Buy same Device (Buy same Key)
  2. Remove Radio Module
  3. Connect to USB Interface
  4. Interact: API & AT Command Mode
  5. Send frames using the unknown key



Warning: Not possible if exists a Per-Client Encryption Key

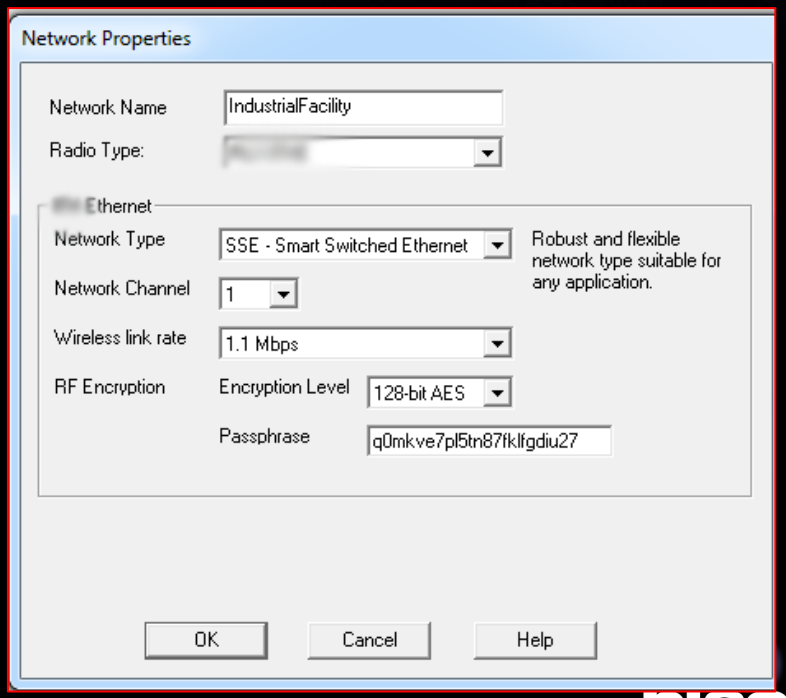
Exploiting Vendor1 Devices

- Company Profile (+1990)
- Frequency Hopping Wireless Devices
  - Great for long or short range wireless SCADA applications
  - Secure proprietary FHSS with 128 bit AES encryption
  - Hazardous location approvals, Perfect for outdoor Ethernet SCADA or indoor PLC messaging
  - 30+ miles point to point with high gain antennas



Vendor1 Key Distribution

"<Vendor1 Tool> is easy to use and intuitive. ***Default values built into the software work well for initial installation*** *and testing making it easy for first-time users. <Vendor1 Tool>* ***manages all important settings to ensure that the network performs correctly." (User Guide)***

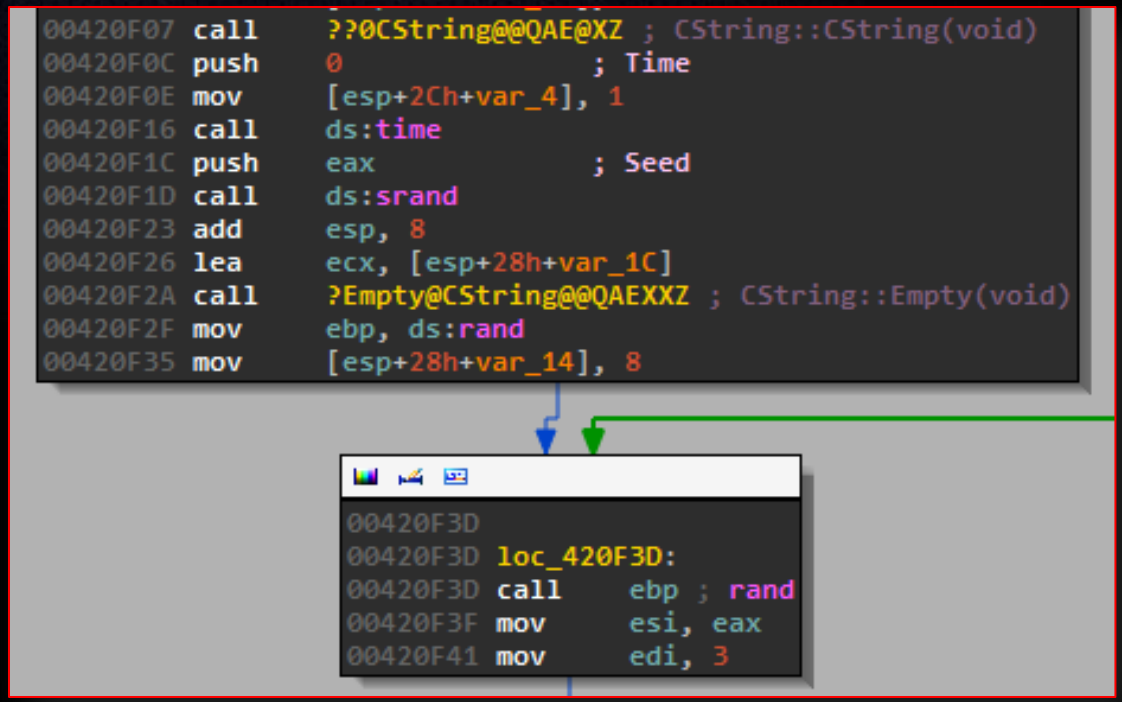


- RF Encryption: A 128-bit encryption level key is suggested for the user.
  - **Blank:** No encrypted packets
  - **5-7 Chars:** Field is translated into a 40-bit encryption level.
  - **15-24 Chars**: Field is translated into a 128-bit encryption level.

Reversing Passphrase Generation

Compiled C++ Binary:

- srand seeds PRNG
- time returns epoch
- srand(time(NULL))
  - Low Entropy Seed
  - Same algorithm
- rand()
  - Bad ANSI C function



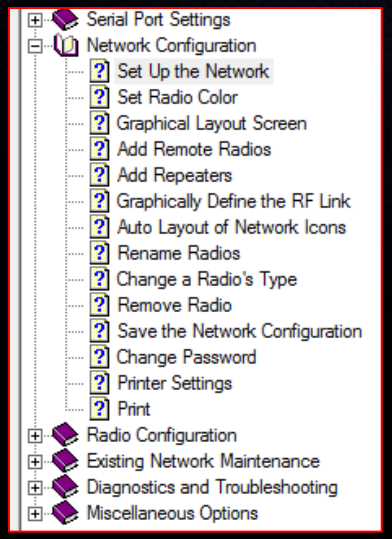
Attacking Weak PRNG

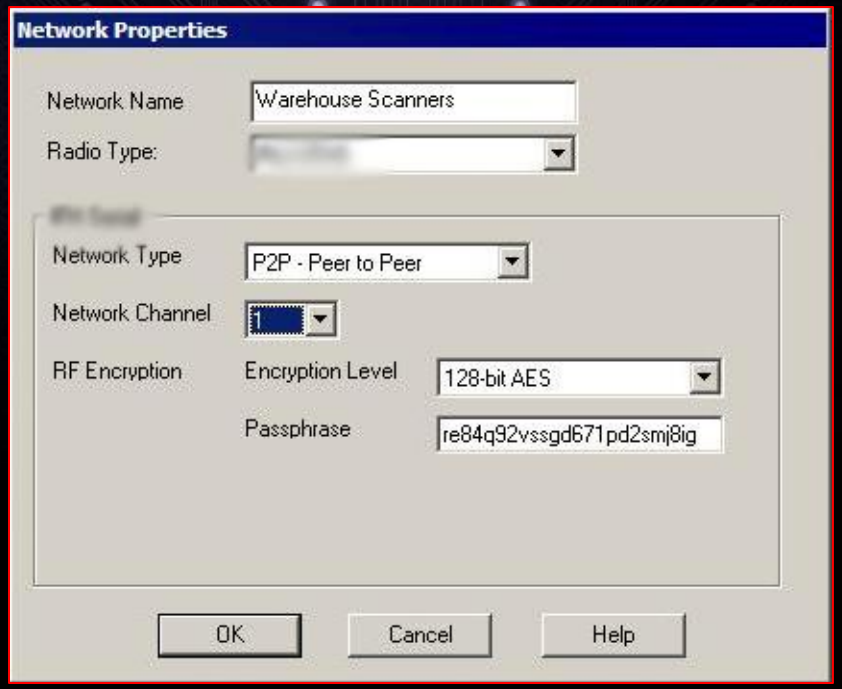
```
void *printPassphrase(time_t epoch)
{
    char buff[100];
    strftime (buff, 100, "%Y-%m-%d %H:%M:%S", (int*)localtime(&epoch));
    printf ("%s => ", buff);
    char passphrase[25] = "\0";
    srand(epoch);
    int block_counter = 8;

    do{
        int i = rand();
        int counter = 3;
        do{
           int i2 = i & 0x1f;
           if(i2 >= 0x0a){
                i2 = i2 + 0x57;
            }else{
                i2 = i2 + 0x30;
            }
            appendchar(passphrase,sizeof passphrase,(char) i2);
            i = i >> 5;
            counter--;
        }while(counter>0);
        block_counter--;
    }while(block_counter>0);
    printf("%d => %s\r\n",epoch, strrev(passphrase));
    return;
}
```

C:\>passgen.exe
2013-04-04 21:39:08 => 1365136748 => knc6gadr40565d3j8hbrs6o0

The Oldest Passphrase

Help File






```
C:\>passgen.exe
2013-04-04 21:39:08 => 1365136748 => knc6gadr40565d3j8hbrs6o0
2013-04-04 21:39:07 => 1365136747 => nir3f1a0dm2sdt41q91c06nt
...
2008-04-17 15:20:47 => 1208470847 => re84q92vssgd671pd2smj8ig
```

Comissioning Tool Audit

Bruteforce Passphrase

- 25^{70} Passphrases
- Mixed lower case alphabet plus numbers and common symbols
- Impossible to calculate all passphrases
- Need to derive AES 128-bit key on realtime

VS

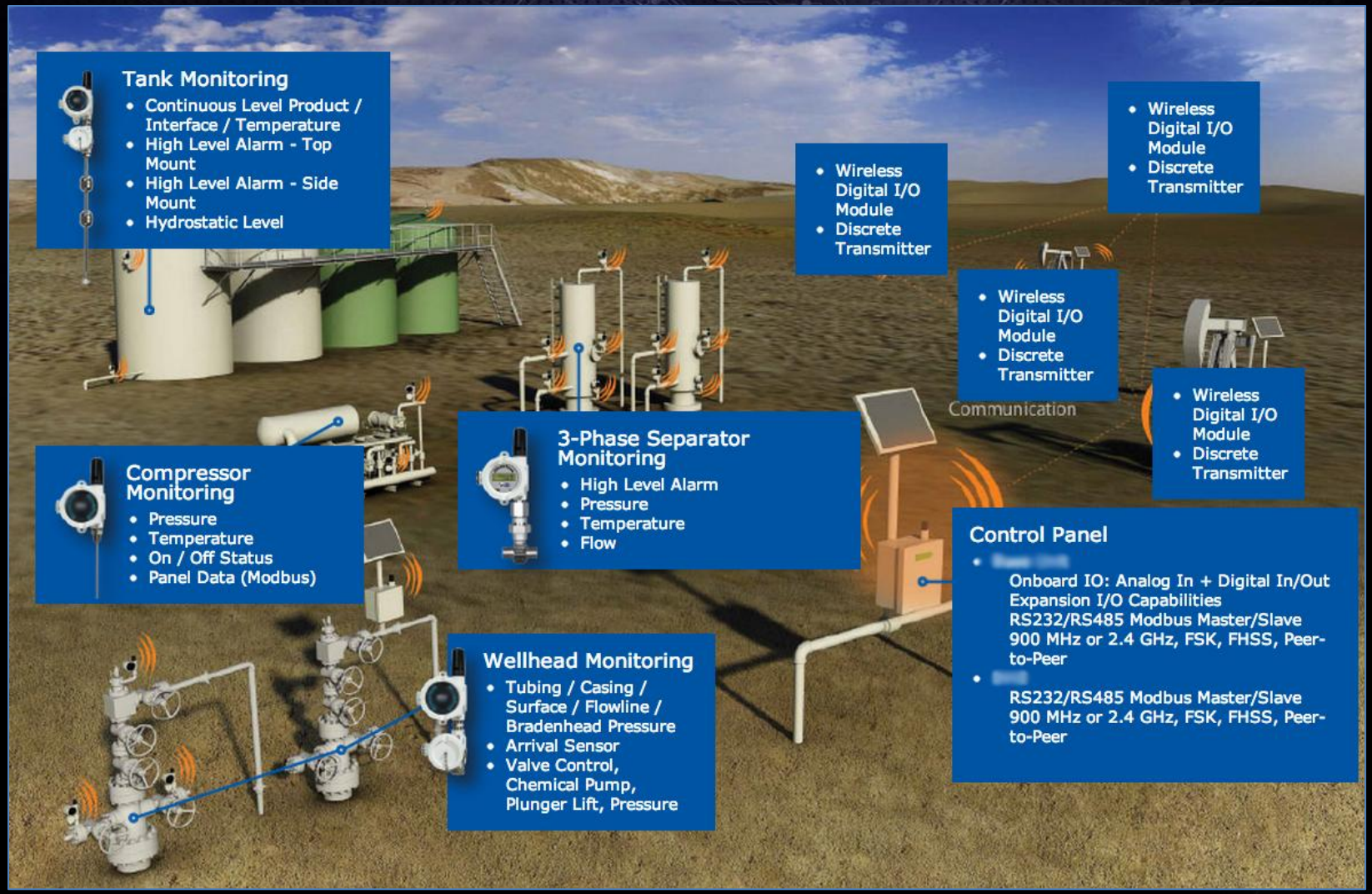
Weak PRNG Attack

- ~156 Million Passphrases
- Every second passed, one more key
- Only a few seconds to calculate all passphrases
- Calculate once and create a database with all possible AES 128-bit key derivations

- Easily breakable by an outsider
- Further Research with the Devices
- Comissioning Tools needs deep testing

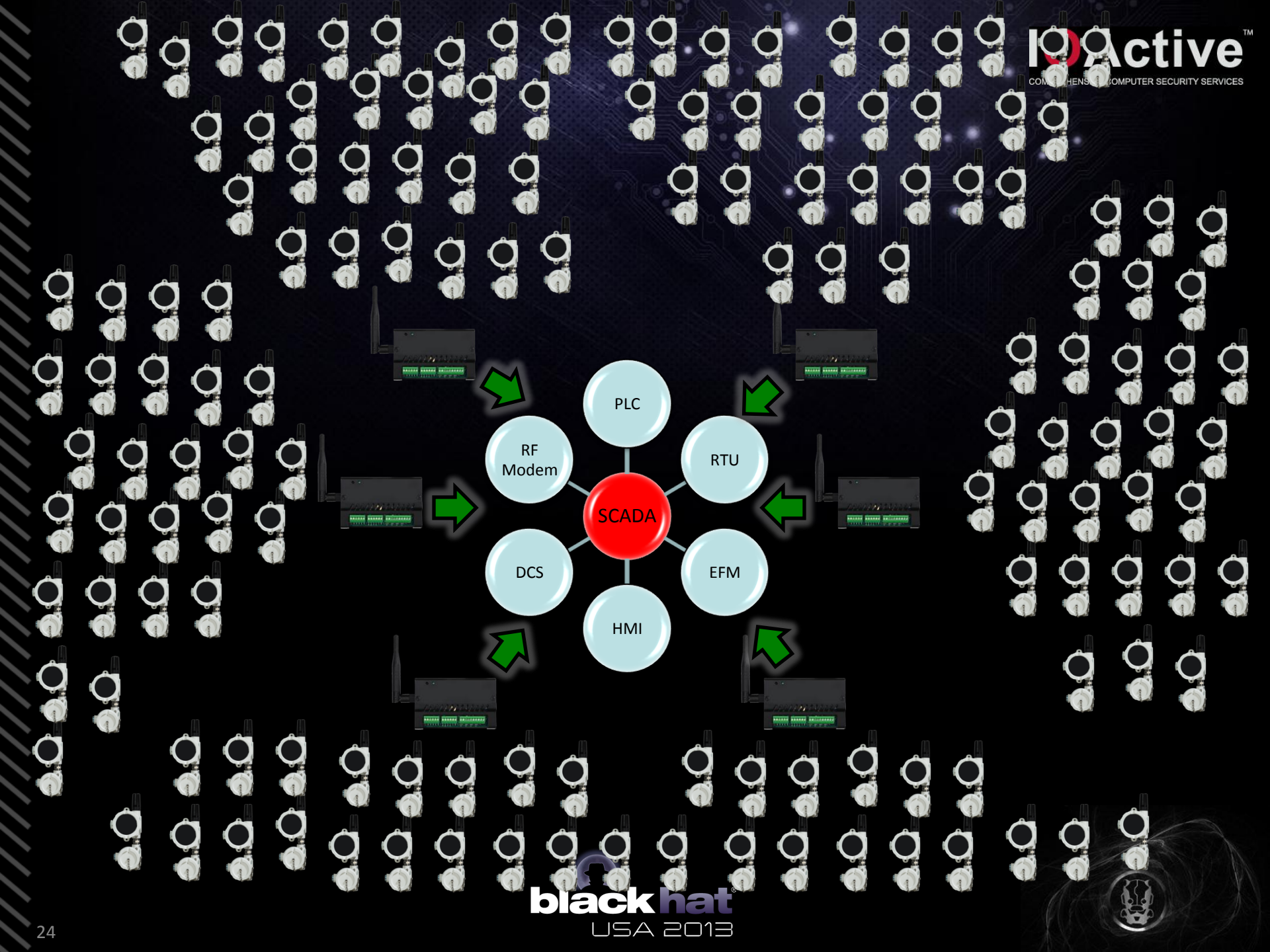
Vendor2 Wireless Devices

- Market leadership: Oil & Gas
- Wireless and wired solutions for the digital oil field automation
- Trusted by top companies in different industries
- **Family System** (Point to Multipoint)**:**
  - **Wireless** Gateways
  - **Wireless** Transmitters
  - I/O Expansion Modules
  - Hardwire Sensors


Tank Monitoring
• Continuous Level Product / Interface / Temperature
• High Level Alarm - Top Mount
• High Level Alarm - Side Mount
• Hydrostatic Level
• Wireless Digital I/O Module
• Discrete Transmitter
• Wireless Digital I/O Module
• Discrete Transmitter
• Wireless Digital I/O Module
• Discrete Transmitter
Communication
• Wireless Digital I/O Module
• Discrete Transmitter
3-Phase Separator Monitoring
• High Level Alarm
• Pressure
• Temperature
• Flow
Compressor Monitoring
• Pressure
• Temperature
• On / Off Status
• Panel Data (Modbus)
Control Panel
• [redacted]
Onboard IO: Analog In + Digital In/Out
Expansion I/O Capabilities
RS232/RS485 Modbus Master/Slave
900 MHz or 2.4 GHz, FSK, FHSS, Peer-to-Peer
• [redacted]
RS232/RS485 Modbus Master/Slave
900 MHz or 2.4 GHz, FSK, FHSS, Peer-to-Peer
Wellhead Monitoring
• Tubing / Casing / Surface / Flowline / Bradenhead Pressure
• Arrival Sensor
• Valve Control, Chemical Pump, Plunger Lift, Pressure

An Extended Family of Devices

- Applications
  - Oil & Gas
  - Refining / Petro Chemicals
  - Water & Waste Water
  - Utilities
  - Industrial Process Monitoring
- Transmitters
  - RTD Temperature Transmitter
  - Analog/Discrete Transmitter
  - Flow Totalizer Transmitter
  - Pressure Transmitter
  - Hydrostatic Level Transmitter
  - Many more..


PLC
RF Modem
RTU
SCADA
DCS
EFM
HMI

Tool and Project Files

- How the devices access the wireless information?
- "Enhanced Site Security Key"

*The Enhanced Site Security feature designed to provide an additional **level of protection** for **RF packets** sent and received between <Vendor2> devices and minimizes the possibility of interference from other devices in this area. This feature is **not available on some older versions** of legacy devices.*

- Security Key == Encryption Key ???
- Legacy Devices Without Encryption???

Key Generation and Distribution

- Create a "Project File" and update all Nodes
- From documentation:

*"If the project file name is **changed**, a new **Site Security Key** will be assigned"*

*Possible Scheme: **Per-Site Encryption***

This Key MUST be somewhere on the Project File.

File Name Change => New Key

```
mov     ecx, [ebx+20B4h]            ; Check if file path has changed
mov     edi, [ecx+esi*4]
test    edi, edi
jz      short not_changed
```

```
push    0                           ; Time
call    __time64                    ; Determine the current calendar time
add     eax, esi
add     esp, 4
mov     [esp+18h+var_4], edx
mov     edx, [edi]
push    eax
mov     eax, [edx+10h]
mov     ecx, edi
call    eax                         ; eax = &update_key
mov     eax, [esp+18h+security_enabled]
```

Project File Binary Diffing

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

ProjectA

\x17\x58\x4f\x51

1364154391

Sun, 24 Mar 2013
19:46:31 GMT

ProjectB

\x51\x58\x4f\x51

1364154449

Sun, 24 Mar 2013
19:47:29 GMT

```
-- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x00000000   0300 0000 0100 0000 0200 ffff 0800 0800  ................
0x00000010   4669 6c65 4e6f 6465 0000 0000 8403 0000  FileNode........
0x00000020   0000 0000 0000 0000 0000 0000 0000 0010  ................
0x00000030   0000 0850 726f 6a65 6374 4103 433a 5c01  ...ProjectA.C:\.
0x00000040   00ff ff03 0004 0053 6974 6517 584f 5101  .......Site.XOQ.
0x00000050   0653 6974 655f 3100 0000 0001 0000 0000  .Site_1.........
0x00000060   0000 0000 ffff 0800 0b00 [illegible]     ..........[illegible]
0x00000070   [illegible] 7402 0000 0084 0300 0001 0000 [illegible]..........
0x00000080   0009 4761 7465 7761 795f 3100 0000 0000  ..Gateway_1.....
0x00000090   0000 0082 0010 0000 0100 0000 1600 0000  ................
0x000000a0   3c00 0000 0000 0010 0000 0000 0000 c842  <..............B
0x000000b0   0300 0000 0010 0000 0000 0000 c842 0300  .............B..

-- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF
0x00000000   0300 0000 0100 0000 0200 ffff 0800 0800  ................
0x00000010   4669 6c65 4e6f 6465 0000 0000 8403 0000  FileNode........
0x00000020   0000 0000 0000 0000 0000 0000 0000 0010  ................
0x00000030   0000 0850 726f 6a65 6374 [illegible]     ...ProjectB.C:\.
0x00000040   00ff ff03 0004 0053 6974 6551 584f 5101  .......SiteQXOQ.
0x00000050   0653 6974 655f 3100 0000 0001 0000 0000  .Site_1.........
0x00000060   0000 0000 ffff 0800 0b00 [illegible]     ..........[illegible]
0x00000070   [illegible] 7402 0000 0084 0300 0001 0000  eUnit...........
0x00000080   0009 4761 7465 7761 795f 3100 0000 0000  ..Gateway_1.....
0x00000090   0000 0082 0010 0000 0100 0000 1600 0000  ................
0x000000a0   3c00 0000 0000 0010 0000 0000 0000 c842  <..............B
```

Component Identification

- Support Center
  - Firmware Images & Documentation
  - Radio Modules, Architectures & Processors

HARDWARE FEATURES	
Device Functionality	· Wireless Gateway
Embedded Controller	· 32-bit Low Power ARM7 Microcontroller with Internal FLASH (Field Upgradeable)

ARM

HARDWARE FEATURES	
Device Functionality	· RTD Temperature Monitor w/ Built-In Wireless Transmitter
Embedded Controller	· Ultra-low Power RISC Microcontroller with Internal FLASH (Field Upgradeable)

RISC

Understanding Firmware Image (RISC)

- Industry Standard Format
  - @Address and content
  - Incomplete Image (Update)
  - Only compiler strings

```
@1200
4F 43 4E 43 00 3C F2 B0 40 00 02 00 FC 27 5D 42
76 00 4F 4D D2 C2 76 00 02 00 7F 90 7E 00 F2 23
30 41 4F 43 7E 40 5A 00 D2 93 AA 07 17 28 F2 90
06 00 AA 07 13 2C 4F 43 03 3C 5E EF AA 07 5F 53
5D 42 A9 07 1D 83 0F 9D F8 3B 5D 42 A9 07 CD 9E
A9 07 02 24 4F 43 03 3C 5F 43 01 3C 4F 43 30 41
4F 43 00 3C F2 B0 40 00 02 00 FC 27 5E 42 76 00
C2 4E A9 07 00 3C F2 B0 40 00 02 00 FC 27 4E 4F
5F 53 4E 4E 5D 42 76 00 CE 4D AA 07 F2 F0 BF 00
02 00 5F 92 A9 07 EE 2B 30 41 0B 12 0A 12 4A 43
5B 42 AA 07 5B 93 0B 20 5F 43 92 12 AC 02 F2 40
06 00 77 00 00 3C D2 B3 71 00 FD 27 21 3C 6B 93
```

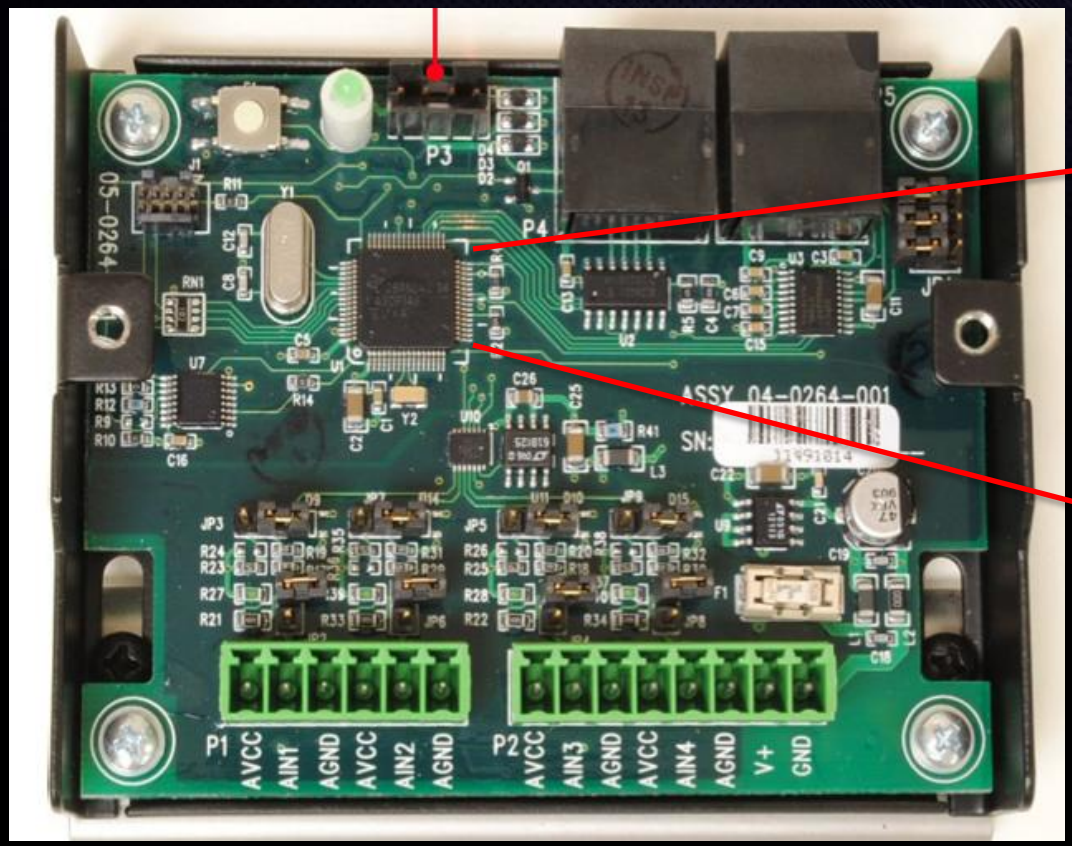
```
L#4A5A6A7A8A9A:A;A0A0A0A0A
^4A5A6A7A8A9A:A;A0A\rC<@
J/B4A5A6A7A8A9A:A;A0A\v
H1R4A5A6A7A8A9A:A;A0A\v
H!S4A5A6A7A8A9A:A;A0A
J4A5A6A7A8A9A:A;A0A\v
J4A5A6A7A8A9A:A;A0A\v
J/B5A6A7A8A9A:A;A0A\v
*4A5A6A7A8A9A:A;A0A\v
```

Online Search

CrossWorks for MSP430

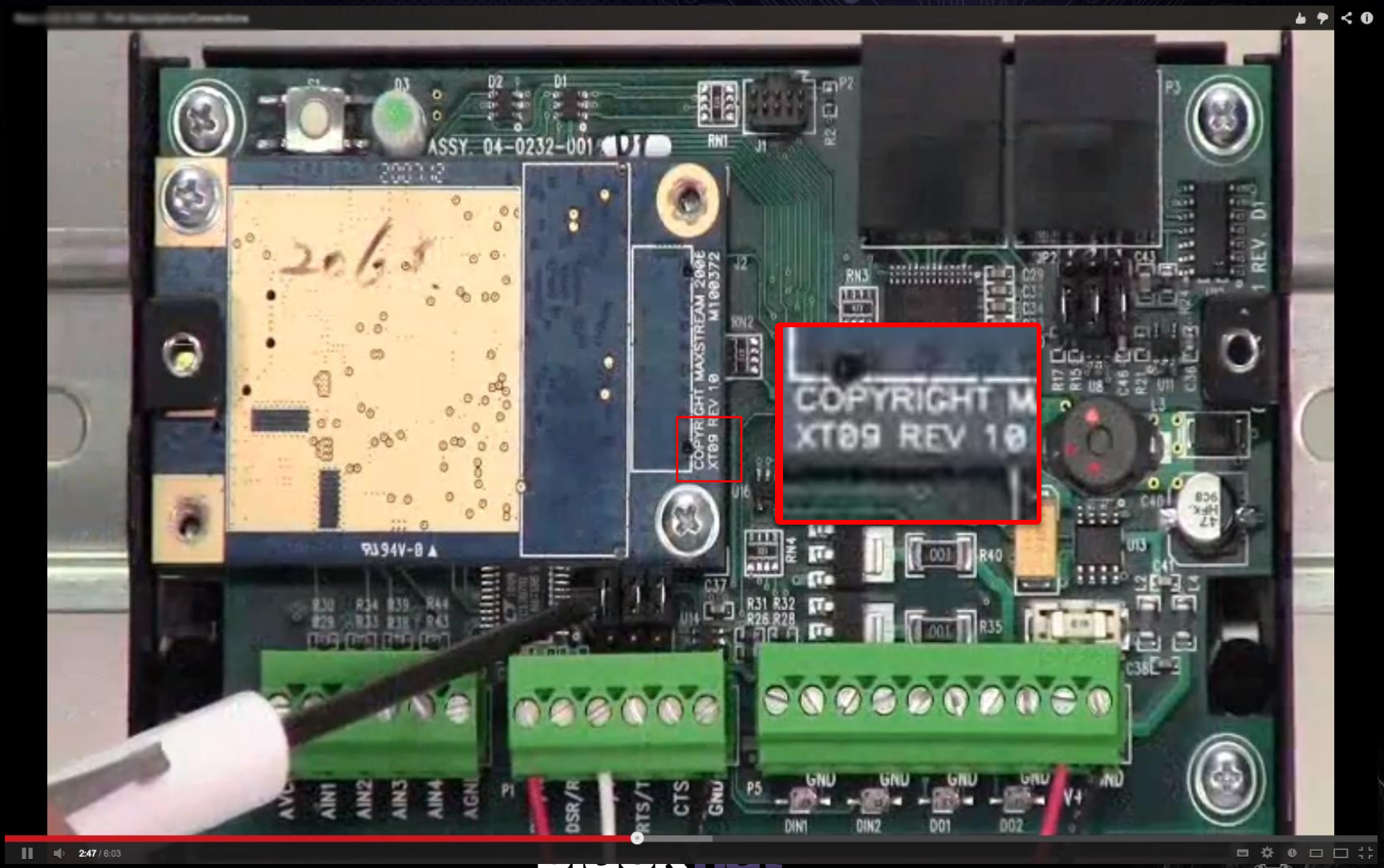
IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

Component Identification



430F149

YouTube (XT09 and 802.15.4)



No Per-Client Key

Dear <<Reseller Sales Eng>>,

*We are going to **borrow a used** "Analog Transmitter" from one of our partners,*

*We are going to test it for a few weeks and let you know if we decide to **buy a new one**.*

***Are there any specific concern we might take into account when deploying this device to connect it with our <Device>?** Or just upgrade all project configuration files?*

Thank you

Lucas,

You just need to upgrade the configuration files.

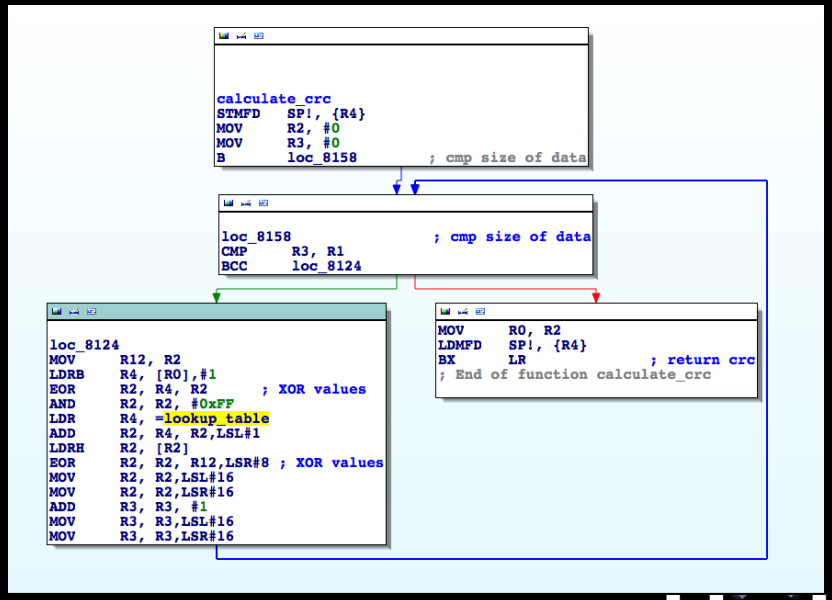
Thanks.

Finding Embedded Keys

- Two kind of Firmwares (ARM and MSP430)
  - One possible hardcoded key in both firmwares
  - Binary Equaling

```
-- offset -  0 1  2 3  4 5  6 7  8 9  A B  C D  E F
0x00000000  4f43 4e43 003c f2b0 4000 0200 fc27 5d42
0x00000010  7600 4f4d d2c2 7600 0200 7f90 7e00 f223
0x00000020  3041 4f43 7e40 5a00 d293 aa07 1728 f290
0x00000030  0600 aa07 132c 4f43 033c 5eef aa07 5f53
```

```
-- offset -  0 1  2 3  4 5  6 7  8 9  A B  C D  E F
0x00000000  18f0 9fe5 ffff ffff ffff ffff ffff ffff
0x00000010  ffff ffff ffff ffff 18f0 9fe5 18f0 9fe5
0x00000020  4000 0000 ffff ffff ffff ffff ffff ffff
0x00000030  ffff ffff ffff ffff 04b8 0000 88b8 0000
```



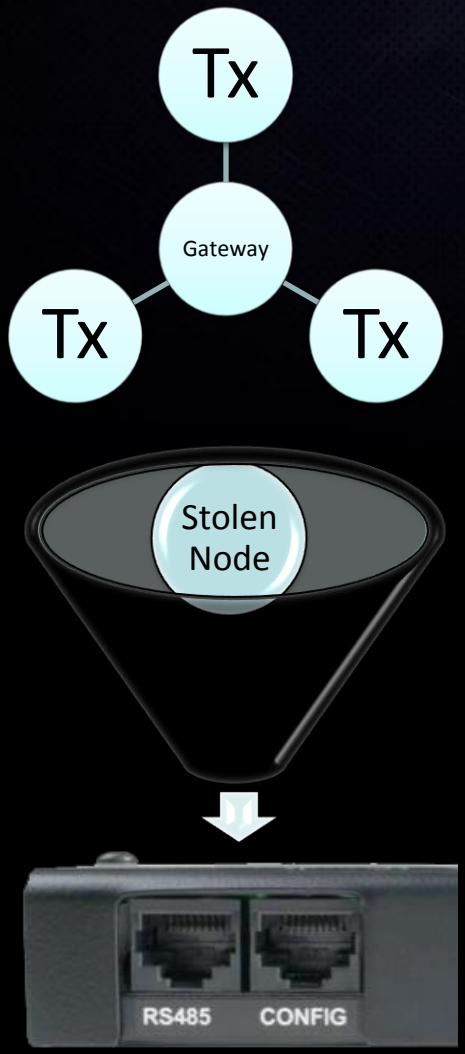
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0000	C1C0	81C1	4001	01C3	C003	8002	41C2	01C6	C006	8007	41C7	0005	C1C5	81C4	4004
1	01CC	C00C	800D	41CD	000F	C1CF	81CE	400E	000A	C1CA	81CB	400B	01C9	C009	8008	41C8
2	01D8	C018	8019	41D9	001B	C1DB	81DA	401A	001E	C1DE	81DF	401F	01DD	C01D	801C	41DC
3	0014	C1D4	81D5	4015	01D7	C017	8016	41D6	01D2	C012	8013	41D3	0011	C1D1	81D0	4010
4	01F0	C030	8031	41F1	0033	C1F3	81F2	4032	0036	C1F6	81F7	4037	01F5	C035	8034	41F4
5	003C	C1FC	81FD	403D	01FF	C03F	803E	41FE	01FA	C03A	803B	41FB	0039	C1F9	81F8	4038
6	0028	C1E8	81E9	4029	01EB	C02B	802A	41EA	01EE	C02E	802F	41EF	002D	C1ED	81EC	402C
7	01E4	C024	8025	41E5	0027	C1E7	81E6	4026	0022	C1E2	81E3	4023	01E1	C021	8020	41E0
8	01A0	C060	8061	41A1	0063	C1A3	81A2	4062	0066	C1A6	81A7	4067	01A5	C065	8064	41A4
9	006C	C1AC	81AD	406D	01AF	C06F	806E	41AE	01AA	C06A	806B	41AB	0069	C1A9	81A8	4068
A	0078	C1B8	81B9	4079	01BB	C07B	807A	41BA	01BE	C07E	807F	41BF	007D	C1BD	81BC	407C
B	01B4	C074	8075	41B5	0077	C1B7	81B6	4076	0072	C1B2	81B3	4073	01B1	C071	8070	41B0
C	0050	C190	8191	4051	0193	C053	8052	4192	0196	C056	8057	4197	0055	C195	8194	4054
D	019C	C05C	805D	419D	005F	C19F	819E	405E	005A	C19A	819B	405B	0199	C059	8058	4198
E	0188	C048	8049	4189	004B	C18B	818A	404A	004E	C18E	818F	404F	018D	C04D	804C	418C
F	0044	C184	8185	4045	0187	C047	8046	4186	0182	C042	8043	4183	0041	C181	8180	4040

Acquiring the Devices

- RTD Temperature Transmitter
  - Integrates Platinum 100 ohm RTD Sensor
  - Ideal for use in various mission-critical industrial applications.
  - Ideal for Monitoring Air, Gas, Water, or Liquid Temperatures

- Wireless Gateway
  - Gateways are responsible for receiving/collecting data from wireless end nodes
  - The collected data can be communicated with third-party Modbus device such as a RTU, PLC, EFM, HMI, or DCS

Resilience and Node Capture



FF 41 06 00 0A 00 00 00 33 2E 1D CC

Serial

Capture

- Extraction
  - Site Security Key
  - Project File

FF 41 0A 00 0A 00 00 00 04 00 AB D0 9A 51 B0 ...

A crypto attack disappointment

- Protocol Reverse Engineering
  - Device has a debug interface
  - Developed a custom tool to receive and send **802.15.4** data
    - 2.4ghz Transceiver (Modified Firmware and Reflashed by JTAG)
    - PyUsb, IPython, Scapy Dissectors, etc.
    - Borrowed **KillerBee** Frame Check Sequence Code
  - Against the perfect scheme: **Per-Site Encryption Key**
    - **Key not really used for data encryption**
    - **Key only used to ”authenticate” devices**
    - **No integrity and confidentiality**

Temperature Injection Live Demo

- Developed an HMI Project
- **C**hemical **S**afety **B**oard (US) background video
- Modbus RTU Driver
- Arduino and SimpleModbus
- Rotary Actuator
- Cost of the attack: $40 USD
- **Live Demo**

IOActive™
COMPREHENSIVE COMPUTER SECURITY SERVICES

KEEP CALM AND GET TO THE CHOPPA!

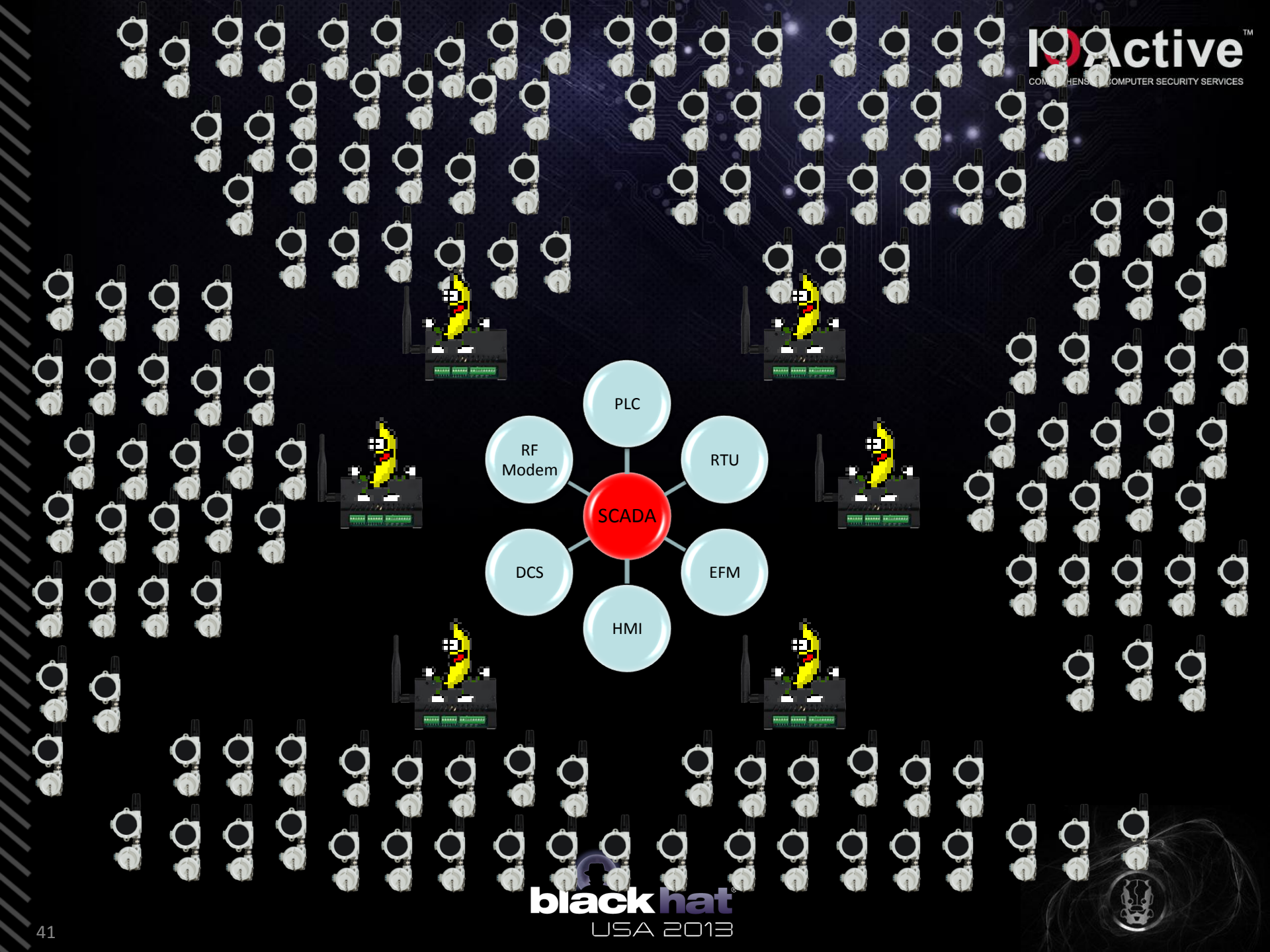
Remote Memory Corruption

- Identify all the protocol fields
- Memory corruption bug using unhandled values.
- Remotely exploitable over the air

- Plant Killer =>

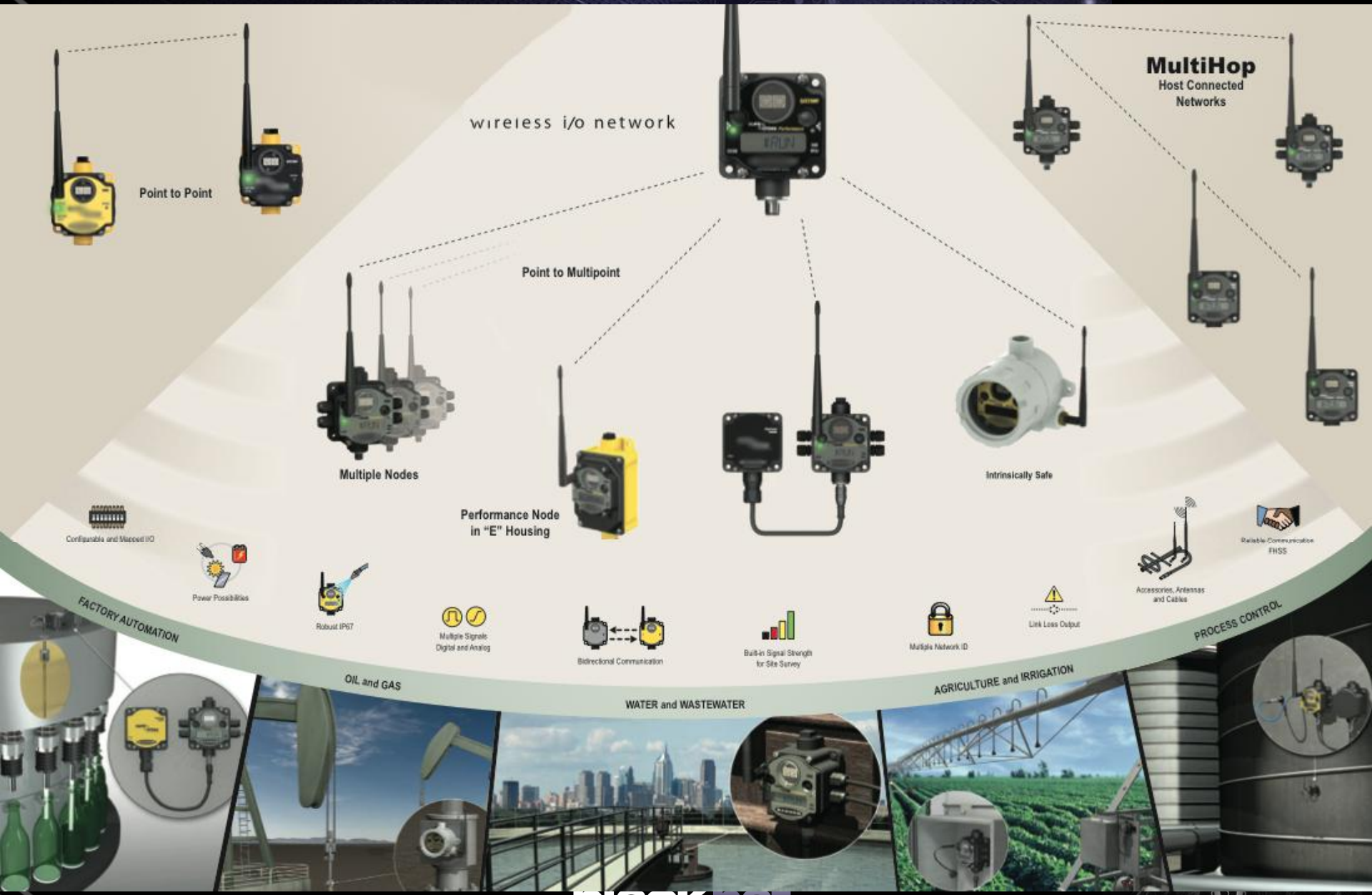
```
17:45:13.608- RF Node Timeout:  Group 0, Node 1, Timeout 00:01:00.
17:46:13.954- RF Node Timeout:  Group 0, Node 1, Timeout 00:01:00.
17:47:14.337- RF Node Timeout:  Group 0, Node 1, Timeout 00:01:00.
17:48:14.662- RF Node Timeout:  Group 0, Node 1, Timeout 00:01:00.
```

- Also could be useful to dump firmware or memory.
- We recorded a demo


PLC
RF Modem
RTU
SCADA
DCS
EFM
HMI

Vendor3 Devices

- Company Profile
  - Self-proclaimed leader in process and industrial automation
  - Clients: Nearly all manufacturing companies from Fortune 500
  - 22.000 different products across 40 industries
- Wireless System (Family)
  - Wireless Gateway
    - Master device used to control network timing and comm traffic
  - Nodes
    - Collect data -> TX Gateway


wireless i/o network
Point to Point
Point to Multipoint
MultiHop
Host Connected Networks
Multiple Nodes
Performance Node in "E" Housing
Intrinsically Safe
Configurable and Mapped I/O
Power Possibilities
Robust IP67
Multiple Signals Digital and Analog
Bidirectional Communication
Built-in Signal Strength for Site Survey
Multiple Network ID
Link Loss Output
Accessories, Antennas and Cables
Reliable Communication FHSS
FACTORY AUTOMATION
OIL and GAS
WATER and WASTEWATER
AGRICULTURE and IRRIGATION
PROCESS CONTROL

Research

- Wireless Family Technical Note:

"Multi-layer security protocol protects your data"

- Network Security
- Data Security
- Data Integrity and Control Reliability

"The wireless I/O systems provide a level of security, data integrity, and reliability far exceeding most wireless systems on the market today"

Quotes (Network Security)

"<Family> is designed to completely eliminate all Internet Protocol (IP) based security threats. Wi-Fi access points have the potential to route any and all data packets, which is why these systems use encryption"

Quotes (Data Security)

"The protocol only carries sensor data values.

Only I/O data is transmitted in the wireless layer."

Quotes (Comm Protocols)

"Widely used open protocols such as Wi-Fi have serious security issues. Even a high degree of encryption **may not protect** your data. It is common for new encryption schemes to be hacked within months of implementation. Proprietary systems are more difficult to hack than an open standard."

Quotes (Comm Protocols)

"<Vendor3> achieves data security by using a proprietary protocol, pseudo-random frequency hopping, and generic data transfer. The <Family> protocol only carries I/O data, making it impossible for a malicious executable file to be transmitted."

Quotes (Comm Protocols)

"This protocol does not operate like an open protocol such as Wi-Fi and is not subject to the risks of an open protocol."

Conclusions (Securing the scheme)

- Out of bands methods
  - Pre-share a strong secret for the initial link (eg: serial comm)
  - Also 802.15.4 AES Encryption at lower layers (MAC)
- Secure the Node Physical Access (Mainly KDC)
- Use hardware Anti-tamper mechanisms
- Audit Source Code // Audit Site regularly
- ICS-CERT Hardening Guides

Conclusions

- Problem space has always been an open topic
- The journey of keys allows practical attacks
- WSN's standards maturity is growing
- Vendors can fail at implementing them
- No evidence of previous security reviews
- Testing the field location is possible with the proper Hardware and open source Software


CC1111


RZUSB


TelosB


HackRF

Aknowledgements

- ICS/CERT – US/CERT
- References: Piotr Szczezechowiak, Haowen Chan, A. Perrig, Seyit A. Camtepe, Bulent Yener, Rob Havelt, Travis Goodspeed, Joshua Wright…
- IOActive, Inc.

THANK YOU !

Lucas Apa (lucas.apa@ioactive.com)
Carlos Penagos (carlos.hollman@ioactive.com)

IOActive
COMPREHENSIVE INFORMATION SECURITY SERVICES