Establishing a Root of Trust (RoT) in EtherNet/IP Devices
Implementing CIP Security

John S. Rinaldi
Real Time Automation

March 4, 2020
I. What is a Root of Trust

II. EtherNet/IP Cybersecurity Issues

III. CIP Security Resolves these Problems

IV. Common Cybersecurity Practices

V. Things Not to Do

VI. Using Software for Root of Trust

VII. Using Hardware for Root of Trust

VIII. Conclusions
WHAT IS ROOT OF TRUST?

One Part of Your Overall Secure Device Solution

Supply Chain Integrity  Secure Boot  Tamper Protection  Root of Trust (RoT)  Access Authorization  Life Cycle Management

Many definitions but a Root of Trust can be thought of as providing confidence that:

1. Your OS is not compromised
2. Your certificates and keys haven’t been compromised
3. Your device memory hasn’t been exploited
THE NEED FOR CIP SECURITY

North Side generally well-protected

South Side to-date largely unprotected

Outside Attacker

Untrained Employee

Disgruntled Employee
EtherNet/IP

An Insecure Industrial Protocol by Design

• No Confidentiality (Unencrypted Messages)
• No Authentication challenges
• No Protection from Message Alteration
• No Authorization process

But…

• Isolated Networks not usually directly connected to the Internet
• Many devices operating on bare metal
CONFIDENTIALITY
Can You Keep a Secret?

AUTHENTICATION
Who Are You?

INTEGRITY
Is it same Msg Received as Sent?

AUTHORIZATION
What Permissions do you have?
“The proper management of cryptographic keys is essential to the effective use of cryptography for security. Keys are analogous to the combination of a safe. If a safe combination is known to an adversary, the strongest safe provides no security against penetration.

Similarly, poor key management may easily compromise strong algorithms. Ultimately, the security of information protected by cryptography directly depends on the strength of the keys, the effectiveness of mechanisms, and protocols associated with keys, and the protection afforded to the keys.”

NIST Special Publication 800-57 Part 1, Revision 3
THE BAD NEWS…

Key Security is Hard to Accomplish

There is no Perfect Solution

More Difficult When You Don’t Know the Device Application
SOME ACTIONS YOU SHOULD CONSIDER:

• Utilize a tamper-resistant enclosure; deadman switches or anti-tamper meshes for intrusion protection

• Implement tamper-resistant circuitry to detect circuit board intrusions

• Implement a secure boot to authenticate firmware prior to execution

• Encrypt and store critical information in secure storage memories

• Implement hardware-based random number generator (for encryption)

• Offload cryptographic processing to hardware accelerators

• Secure your remote update process to ensure factory-authorized updates

• Use a trusted execution environment (TEE) – a secure area of a processor for isolated execution, application integrity and confidentiality
MISTAKES NOT TO MAKE:

• Using encryption keys that are too short
• Creating keys using a flawed, low entropy, random number generator
• Storing an encryption key in application code, file or table.
• Storing an encryption key with poor protection (XOR with weak data, etc.)
• Storing an encryption key with password-based encryption protection
• Failing to secure debug ports so that an attacker can monitor execution
• Failing to provide an audit trail of who accessed what data and when
• Failing to provide a tamper-proof, secure clock
KEY MANAGEMENT OPTIONS

SOFTWARE KEY MANAGEMENT
• Less Secure Devices & Applications
• No Hardware Upgrades Required
• Support Legacy Devices

HARDWARE KEY MANAGEMENT
• More Secure
• Upgrade to Your Hardware Environment
• Rely on Outside Vendor
### SW Method 1 – Hide the Key in Plain Sight

<table>
<thead>
<tr>
<th>Description</th>
<th>Keys stored as clear text in NV RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why Use It:</td>
<td>When the device application is such that consequences of a breach are insignificant such as an input-only sensor with inconsequential data</td>
</tr>
<tr>
<td>Limitations:</td>
<td>This approach relies on inability of most attackers to 1) penetrate the device internals, 2) access non-volatile RAM, and 3) identify which bytes are the device keys.</td>
</tr>
<tr>
<td>Attack Surface:</td>
<td>Large</td>
</tr>
<tr>
<td>Implementation Cost:</td>
<td>Minimal</td>
</tr>
</tbody>
</table>

**NOTE:** THIS APPROACH SHOULD NEVER BE USED ON A CIP SECURITY NETWORK USING PRE-SHARED KEYS
### SW Method 2 – Let the OS Worry About it

<table>
<thead>
<tr>
<th>Description:</th>
<th>Shift the burden of Key Management to the OS Vendor.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why Use It:</td>
<td>Key Management using Advanced Encryption Standard (AES) is now available in popular operating systems. It provides adequate protection without burden of hardware.</td>
</tr>
<tr>
<td>Limitations:</td>
<td>The security mechanisms (and limitations) of a Linux or a Windows OS are well-known by the attacker community. Every unpatched vulnerability increases the attack surface.</td>
</tr>
<tr>
<td>Attack Surface:</td>
<td>Varies with the selected operating system</td>
</tr>
<tr>
<td>Implementation Cost:</td>
<td>Minimal</td>
</tr>
</tbody>
</table>
**SW Method 3 – Multilevel Key Storage**

<table>
<thead>
<tr>
<th>Description:</th>
<th>A master key known as a DEK (Data Encryption Key) or a KEK (Key Encryption Key) is used to protect secrets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why Use It:</td>
<td>These keys are longer and stronger than the private keys and other secrets they protect.</td>
</tr>
<tr>
<td>Limitations:</td>
<td>Instead of protecting private keys, the device must protect the DEK or KEK. Same problem for a different key.</td>
</tr>
<tr>
<td>Attack Surface:</td>
<td>Moderate</td>
</tr>
<tr>
<td>Implementation Cost:</td>
<td>Minimal</td>
</tr>
</tbody>
</table>
### SW Method 4 – Don’t Use a Key

<table>
<thead>
<tr>
<th>Description:</th>
<th>Physical Unclonable Function (PUF) technology. Innate characteristics of a semiconductor component generate keys. Cannot be cloned, guessed, stolen or shared. Keys generated when required. Nothing stored on the system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why Use It:</td>
<td>PUF keys are guaranteed to be unique and unclonable since they utilize the inherent randomness of the manufacturing process.</td>
</tr>
<tr>
<td>Limitations:</td>
<td>PUF technology must be licensed and maintained</td>
</tr>
<tr>
<td>Attack Surface:</td>
<td>Small</td>
</tr>
<tr>
<td>Implementation Cost:</td>
<td>Minimal</td>
</tr>
</tbody>
</table>
## HW Method 1 – Non-Volatile Memory

<table>
<thead>
<tr>
<th>Description:</th>
<th>Encryption keys are in a special NVRAM implemented solely for secrets. Secrets are secure as device firmware provides no mechanism for unauthorized access.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why Use It:</td>
<td>Implementation simplicity and lower cost development, production and support.</td>
</tr>
<tr>
<td>Limitations:</td>
<td>Requires extremely high-quality source code, open source and third-party libraries, as any flaw may compromise secrets. Possible to unsolder NVRAM to discover secrets.</td>
</tr>
<tr>
<td>Attack Surface:</td>
<td>Moderate</td>
</tr>
<tr>
<td>Implementation Cost:</td>
<td>Minimal</td>
</tr>
</tbody>
</table>
## HW Method 2 – External Security Module

| Description: | Dedicated cryptographic module designed for protection of secrets. Located off device, module can be networked, attached, embedded in a PC server, or attached via USB. |
| Why Use It: | Cryptography is managed by an external provider. |
| Limitations: | May be impractical due to cost or latency issues. |
| Attack Surface: | Small |
| Implementation Cost: | Significant though one module can support many embedded modules. |
## HW Method 3 – Onboard RoT Device

| Description: | Semiconductor vendors market several HW RoT devices. Some are known as Trusted Platform Modules (TPMs). Secrets are confined within a security-hardened, tamper-resistant IC, and thus hard to get at directly, even with an unlimited budget and expertise. |
| Why Use It: | Significantly increase the resources and technical expertise to compromise a device. |
| Limitations: | Unit cost and footprint can be problematic. |
| Attack Surface: | Small |
| Implementation Cost: | Varies with the manufacturer |
Root of Trust is vital to securing device and network security

CIP Security is necessary to protecting the "South" side of controllers

Key Management is critical when implementing CIP Security

There is no one "best" solution for protecting keys in a CIP Secure device

All solutions, hardware or software-based, have significant limitations

Vendors may have to support several different security solutions

SUMMARY & FINAL THOUGHTS

• Root of Trust is vital to securing device and network security
• CIP Security is necessary to protecting the "South" side of controllers
• Key Management is critical when implementing CIP Security
• There is no one "best" solution for protecting keys in a CIP Secure device
• All solutions, hardware or software-based, have significant limitations
• Vendors may have to support several different security solutions
THANK YOU

John S. Rinaldi
jrinaldi@rtautomation.com
414-460-6556