# Table of Contents

1. Acronyms .......................................................................................................................... 4

2. Introduction ........................................................................................................................ 5

3. Cyber Security Standards .................................................................................................. 5
   3.1 Which standard to follow? ............................................................................................. 6

4. Compliance Procedure ....................................................................................................... 6
   4.1 Organization certifying embedded devices ................................................................. 7

5. Embedded Design Considerations ..................................................................................... 8
   5.1 Current State of Industrial Protocols ........................................................................... 9
   5.2 Communication security ............................................................................................... 9
      5.2.1 Spoofing ............................................................................................................... 9
      5.2.2 Data Integrity ..................................................................................................... 9
   5.3 Defense against Denial of service (DoS) attacks ......................................................... 11
   5.4 Using VPN .................................................................................................................. 11

6. Other Defense Measures .................................................................................................. 12
   6.1 Threat Modeling Tool .................................................................................................. 12
   6.2 Identifying security functions and Non secure functions ......................................... 12
   6.3 Operating Systems ..................................................................................................... 12
      6.3.1 Application Partitioning ..................................................................................... 13
   6.4 Audit Logs .................................................................................................................. 13
      6.4.1 Storing of Audit Logs ....................................................................................... 13
   6.5 Coding Guidelines ...................................................................................................... 13
   6.6 External Interfaces ...................................................................................................... 14
   6.7 Other minor considerations but not trivial ............................................................... 14
7 Testing................................................................................................................................. 14
8 Zones & Conduits .................................................................................................................. 15
9 Conclusions .......................................................................................................................... 15
10 References .......................................................................................................................... 15
1 Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy check – an algorithm to check if sent messages are not corrupted</td>
</tr>
<tr>
<td>CERT</td>
<td>Computer Emergency Response Team – a software group that comes out secure C coding guidelines</td>
</tr>
<tr>
<td>ISA</td>
<td>The Instrumentation, Systems, and Automation Society (ISA) is a global, cross-industry organization connecting people and ideas in automation.</td>
</tr>
<tr>
<td>IEC</td>
<td>International</td>
</tr>
<tr>
<td>PLC</td>
<td>Programming Logic Controller</td>
</tr>
<tr>
<td>RTOS</td>
<td>Real Time operating system</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
</tbody>
</table>

Note: Only acronyms that are not expanded in document are listed above.
2 Introduction

Embedded devices are ubiquitous and are used for safety critical applications to leisure. Embedded devices are increasingly prone to security attacks and such attacks can risk human lives and assets. There are specific international standards and organizations in assisting equipment manufacturers to make it difficult to cyber-attacks.

The objective of this document is to provide an overview of cyber security standards and procedures and more importantly the design consideration to be made during the design/requirements stage of embedded device development, specifically embedded devices that are networked, so the device is compliant with standards on cyber security. It assumes the reader is already aware of general embedded product design practices.

One thing to be remembered is “foolproof cyber security protection does not exist”. But by making a device comply with available standards, its immune system gets strengthened and probabilities of withstanding attacks are high. The best way to keep an embedded device or the entire system housing the embedded device, highly secure is to isolate it from any external network.

3 Cyber Security Standards

Currently, there are quite a few cyber security standards in existence, but each with a specific focus or objective

- **ISA 99 / IEC 62443** – ISA 99 standards are intended for Industrial automation devices. IEC 62443 is a superset of ISA99, incorporating all the specification of ISA 99, but in addition there is ongoing work new specification under it. This document describes design consideration required as this standard

- **NERC** (North American Electric Reliability Corporation) is primarily intended to electrical T&D (Transmission & Distribution) equipment. The Standards are called NERC CIP xxx (Critical Infrastructure Protection, xxx is number assigned). ISA99 and NERC CIP have many common requirements.

- **NIST** (National Institute of Standard Technology) standards. NIST is US federal government body and though they have their own set of requirements for control system security (called Special requirement 800-xx), it endorses the ISA99 standards.

- **ISO27001** is intended for security practice to be implemented at organization level, mainly by IT department.

- **IEC 15408 or Common Criteria** is another set of standards or evaluation procedures intended for IT products in general. It could be used an IT middleware system, relations
databases and even for embedded devices. This is also widely used by network OEM to have their routers, switches certified.

### 3.1 Which standard to follow?

For embedded device, more specifically embedded device that goes into manufacturing, industrial machines, controllers used in Industrial automation, ISA99/IEC62443 is most relevant standard to comply with. Typical examples are

- embedded devices used in mining trucks that virtually controls the Truck,
- for the controller used in process automation controlling a refinery plant,
- for embedded device that controls the air conditioning of building
- embedded device that control elevator/escalator movement

Anywhere where an embedded device is not a standalone device and whose operation or mal operation could risk or cause injury to human lives (this definition is very similar to safety critical systems), ISA99 is applicable.

Other than the Electrical T&D devices which need to comply with NERC CIP, for all other embedded devices that need to be cyber secure may follow the ISA99 /IEC62443 standards.

Henceforth in this document, only ISA99 /IEC62443 is considered.

### 4 Compliance Procedure

The procedure to get an embedded device compliant is very much akin to getting a device safety certified.

- First need to have a System/product development process. Though not a “must” to have a mature process, it greatly helps if the organization had used the processes to develop conventional products, if not cyber secure products. Compliance to CMMi, level 3 and above, quality standards is good benchmark of a mature process.

- The device needs to be designed considering the requirements as specified in ISA99 /IEC62443.

- The Processes and design of device have to be approved by an Assessor. The Assessor is usually from an external organization that has track record of certifying compliance to ISA99

- Testing the device based on set of test procedures in an accredited lab.
4.1 Organization certifying embedded devices

There are 2 well recognized organizations certifying embedded devices on cyber security.

- Wurldtech, a GE company [2], offers the Achilles certification for devices. There are 2 levels of certification—level 1 & 2 with level 2 requiring more tests to be passed. Wurldtech do not publicly state the requirements that need to pass to get the certification, they recommend the use of Achilles Test Platform during the development stage to test the embedded device.

- ISASecure [1], an association of Industrial control system users and manufacturers. ISASecure’s EDSA (embedded device security assurance) certification is intended for embedded devices. ISA Secure EDSA certification comprises of:
  - certifying the processes,
  - the design of embedded device and
  - passing the conformance testing.

The ISA Secure have come out with their own specifications on all 3 above mentioned steps and publicly available on their website. It is largely based on IEC 62443 standards and also some specifications from NERC and NIST. There are 3 levels of EDSA certification with level 3 being the stringent.

One thing to be noted is both the organizations offer their own named certification—Achilles certification or ISA Secure EDA certification and do not explicitly certify complying to ISA 99/IEC 62443.

The choice between the 2 certification bodies would be purely based on specific project teams or cost or comfort level or previous project experiences.
5 Embedded Design Considerations

The intent here is bring out what additionally one needs to consider in the design of an embedded device to make it pass the certification compared to conventional embedded design that does not require any need security certification (any of above mentioned certifications).

Figure 1: Hierarchy in secure control system

The diagram above is a representation of Hierarchy of systems in typical control system implementation.

- Layer 0 – Field Devices like Transmitters and control valves
- Layer 1 – Controllers – that handle the Input /output & control the plant
- Layer 2 - Operator (HMI) interfaces for operator to monitor the actions of controllers and override if necessary
- Layer 3 - Advanced controls Implementation. In Some systems layer 2 & 3 are the merged
- Layer 4 - Corporate LAN

The layer 1 controllers are the embedded devices that need to be protected. They are devices that control the plant and intruders target these devices to cause maximum damage to plant.
5.1 Current State of Industrial Protocols

Many of wired industrial protocols like Modbus, Profibus, DeviceNet, and Fieldbus or CAN do not support authentication, encryption or even session management. A server can send message to client without needing to identify itself. As long as the packet format is correct, the slave device would accept the command. They were developed with performance as top priority and defense against intrusion like encryption, authentication were time consuming which real time system could not tolerate. Magnus Sundell et al [3] has published a detail white paper on security analysis of many industrial protocols

However Wireless protocols like Wireless HART or ISA100.11a are secure as they support encryption, key management for joining network, sessions

5.2 Communication security

One way to get to into networked embedded device is through communication protocol and Communication between nodes is one of most important aspect of making an embedded device secure.

Major security threats for a communication system are

- Spoofing – Impersonating as Master and communicating to slave device
- Data Integrity – Assurance the data sent by master is same received by slave

5.2.1 Spoofing

As industrial protocols have no authentication mechanism on their own, it is very much possible that an intruder’s message, from outside the company firewall, can be sent directly to a process controller. The Corporate Firewall does not check for specific protocol messages. The possible defense techniques would be to have additional firewalls that have rules to drop malformed messages or allow access to change only few parameters and drop requests to access /change other parameters. Figure 1 shows it as “Control Firewall”, these devices check each packet passing through it. These can only restrict the intruder’s activity, but will not prevent if intruder message falls within rules set on control firewall. So truly speaking, defense against spoofing is still not yet there for most industrial protocols.

5.2.2 Data Integrity

As in safety critical applications, implementing a safety layer on top of communication layer guarantees the integrity of message. The safety layer is developed on both the transmitter and receiver message as shown in diagram below.
The various defense techniques against potential errors are explained below:

- **Sequence Number** - Each message has a number that is incremented compared to previous message. This allows the message receiver to check the sequence of messages sent by the message sender. The method reduces risk due to message repetition, loss, insertion and if the messages are sent in incorrect sequence.

- **Timeout**: Messages are sent from the transmitter, only if the acknowledgement to previous message is received. If no ack. is got for a message within 2 consecutive cycles then the transmitter is moved to safe state.

- **Safety Code** – 16 bit CRC is used to check for data corruption. Detection of this error will drive the transmitter to Safe state

- **Feedback Message** - After receiving a message the system (transmitter) sends a positive or negative acknowledgement based on content of message. Examples of positive ack. - by providing confirmation of reception of valid and timely messages, example of negative ack. - by providing confirmation of reception of corrupted messages

The Table below gives the potential errors and errors detection mechanism to be implemented in safety layer. The potential errors that can be minimized by each of technique are marked (✓).
5.3 Defense against Denial of service (DoS) attacks

In DoS attacks, Intruders send a flood of messages (or packets) preventing the system from performing normal communication. Ideally a system should drop such intruder messages and let only genuine message pass through. An industrial embedded device is always at edge of network and when there is flood of information request that come to it, its performance slows down and normal operations get delayed. So prevention of DoS is always done at one layer above the embedded device.

Deep packet Inspection (DPI) is one of proven defense techniques against DoS. Deep packet Inspections is a technology that inspects each packet, based on set of pre-configured rules, let go only those packets that passes all rules. The rules include the allowable source IP addresses, the destination IP, the possible content…. Intruder messages are likely not to pass the rules get dropped.

System that execute DPI must run at high speeds (Gigabit) to inspect each packet and same time not degrade system performance. So this is another reason DPI inspection is not at device level, but at level higher. But as processors are becoming more powerful, embedded DPI on a dedicated processor is also being explored and considered.

The Control Firewall, shown in Figure 1, does Deep packet inspection.

5.4 Using VPN

For controllers communicating to remote I/O cards (Input/Output) via the internet, it is recommended to use VPN connection as shown in Figure 1. A VPN provides a secure tunnel between the remote I/O and controller that could be used with any industrial protocols even if that industrial protocol does not support encryption or authentication.
6 Other Defense Measures

6.1 Threat Modeling Tool

During the requirements stage or in early design stage, using a Threat modeling Tool will help to identify the security threats to device. It is similar to an FMEA. Most of tools would start with data flow diagram (DFD) and do a STRIDE analysis (Spoofing, Tampering, Repudiation, information disclosure, Denial of services, Elevation of Privilege) on each module/component. The outcome of exercise is to list all threats, prioritize them. Mitigation of threats is to be handled during design.

Microsoft SDL Threat Tool, freely downloadable, could be used for analysis.

6.2 Identifying security functions and Non secure functions

Security functions are those requirements specifically implemented to make the product secure. For example implementing Public /Private Key, making the communication protocol secure are all secure functions.

It is possible that not all secure functions are identified at requirements stage and product functional requirements may already include some of secure functions.

Classifying as secure and non secure functions helps in segregating the implementation at Hardware and software level

6.3 Operating Systems

It is strongly recommended to use RTOS for embedded systems that need to be secure. Using a RTOS, that too a proven RTOS would handle many of security requirements, by default without requiring any additional programming, such as:

- Providing separate space for data and executable code
- Support for memory protection thereby preventing buffer overflows, protecting private data, kernel and application data
- Providing isolation between different processes /modules
- File system Encryption

RTOS like WindRiver are already Achilles certified. So using such RTOS makes it easier to comply to IEC 62443 standards.
6.3.1 Application Partitioning

As per IEC62443, it is recommended to separate the security functions from non security functions in H/W and firmware implementation. In firmware, a set of processes or Tasks could be implemented only for security functions and another set of processes /task for application logic.

However options for H/W partitioning are limited. Using a Multi-core processor is one way and the RTOS should support secure multi –core processing.

6.4 Audit Logs

It should be possible to record any action done on device to audit logs. Write to logs should include

- When a device get connected /removed /accessed for
- When a user gets authenticated or denied access
- Operator actions
- Booting sequence
- When a service stops /restarted/aborted
- Diagnostics services messages –from warning to error

Also there should be a function to monitor the working of audit service. Should the audit service fail, an alarm should be raised to higher controller /device

6.4.1 Storing of Audit Logs

Audit logs should be encrypted and should not be easily accessible for any user.

6.5 Coding Guidelines

The CERT C Coding Standard has 228 rules, many of which are based on security flaws discovered in real world code and effective enforcement of the rules will lead to higher-quality systems that are robust and more resistant to attack. Additionally during coding, need to consider:

- Static Data to be memory protected
- Some of the attacks include stack-based buffer overflows, heap-based buffer overflows, exploitation of double-free vulnerability, integer errors, and the exploitation of format string vulnerabilities. Static code analyzers should check for overflows
QA-C has rules implemented complying with CERT C coding standards. Rosecheckers from CERT division is another tool.

6.6 External Interfaces

External interfaces to embedded devices like USB port, PCI interface or JTAG interface should be protected when connected to external systems. Public/Private Key authentication is required in such cases before the interface opens up.

Another secure technique is to use authentication chips connected to main processor of embedded device, that handles the certification. This would also prevent cloning of hardware

6.7 Other minor considerations but not trivial

Following are few more considerations, but certainly not trivial, to comply with standards

- On detecting loss of communication, the system should be taken to safe state. The Safe state of system is to be decided during the requirements stage
- Disable unused ports
- For embedded systems that have a user interface (LCD Panel or monitor) for logging in all precautions as would be done for a secure website would be needed.

7 Testing

Robustness testing of device is mandatory to check the device vulnerabilities. Robustness testing of device is of 2 parts.

- Fuzz testing is a black box testing where malformed messages, incomplete or corrupted messages are sent to device and its reaction and normal functionality observed.
- Stress testing by sending valid messages but by ramping the number of messages till the point device can no longer service the request or perform its normal functionality

Obviously a tool is required to bombard the device with malformed messages or for stress testing using the protocol used by device to communicate with its peer or gateway. ISA Secure has a set of test cases that are to be tested as part of robustness testing. Wurldtech recommends the usage of its own Test platform. There are other tool providers like Aegis Consortium or Scadaforce that have specific protocol fuzzers.

One question that could arise while reading this is: why do fuzz testing of protocols that do not have authentication, encryption mechanism? Fuzz testing detects memory leaks and other failures that makes systems unstable, crash, or hang. Fuzzing contributes to software QA and even design in many ways beyond security
8 Zones & Conduits

A zone is a grouping of logical or physical assets that share common security requirements. Devices performing same functionality can be grouped into a zone. For example a series of automation controllers can be in one zone. Usually devices connected on a protocol bus can be grouped into a zone.

A conduit is a path for the flow of data between two zones. Usually a Gateway separates 2 zones and 2 zones operate on different protocols. In Figure 1, Layer 1 devices are in one zone and layer 2 in another zone.

In a control system environment, the security level for PLC zone (or the devices that perform the critical functions) will be need to be greater than HMI as they do critical functions. Threat modeling and Risk analysis help to identify critical zones. As existing industrial protocols as such are not very secure, including an additional firewall in PLC zone is one way of elevating its security level. This additional firewall will perform DPI and rules be based on protocol used in zone.

9 Conclusions

To make an embedded device secure to large extent, more specifically industrial devices that use protocols like Modbus, Profibus, it would require internal and external protection mechanism to be built. The external control firewall is required till these protocols support encryption, authentication and session management. The device is made more secure by designing to comply with other guidelines such as safety layer or using an RTOS, following CERT C coding guidelines.

10 References

3. Magnus Sundell et al, WHITE PAPER ON INDUSTRIAL AUTOMATION SECURITY IN FIELDBUS AND FIELD DEVICE LEVEL