Protecting IIoT Endpoints

Industrial Internet Security Forum
Thursday, October 6, 2016
Overview

What is an endpoint?
Why endpoint security?
Functions of an endpoint
Implementing endpoint security
What is an Endpoint?

The IIoT Landscape: Where are Endpoints?
What is an Endpoint (II)?

IISF and IIC defines endpoints similar as ISO/IEC 24791-1:2010 standard does:

• An endpoint is one of two components that either implements and exposes an interface to other components or uses the interface of another component.

IIC simplified this definition (see IIC Vocabulary, version 2.0):

• An endpoint is a component that has an interface for network communication.

... but added a note for clarification:

• An endpoint can be of various types including device endpoint or an endpoint that provides cloud connectivity.
What is an Endpoint (III)?

The IIoT Landscape: Endpoints are everywhere!
What is an Endpoint (IV)?

Summary:

• Endpoints are everywhere in an IIoT System (including edge and cloud)
• One single (security) model for all locations
• A single computer, even a device, can have several endpoints
  • Example *Router*: One LAN endpoint, one WAN endpoint
  • Frequently shared code/data between multiple endpoints
• Endpoint and its communication are another model
Why endpoint security?

Endpoints are the **only** location in an IIoT system where:

- Execution code is stored, started and updated
- Data is stored, modified or applied (“Data at Rest” / “Data in Use“)
- Communication to another endpoint is initiated and protected
- Network security is analyzed, configured, monitored and managed

Result: An **attack** to an IIoT system typically starts in attacking one or more endpoints:

- Try to access the execution code and analyze to find weak security implementation
- Attack weak communication protection via network
- Modify or replace (“hijack”) the execution code in a malicious way
- ...
Why endpoint security?

Endpoints are the only location in an IIoT system where:

• Execution code is stored, started and updated
• Data is stored, modified or applied (“Data at Rest” / “Data in Use“)
• Communication to another endpoint is initiated and protected
• Network security is analyzed, configured, monitored and managed

Result: An attack to an IIoT system typically starts in attacking one or more endpoints:

• Try to access the execution code and analyze to find weak security implementation
• Attack weak communication protection via network
• Modify or replace (“hijack”) the execution code in a malicious way
• ...

IISF Endpoint Protection Model

Endpoint Protection

- Endpoint Monitoring & Analysis
- Endpoint Secure Configuration & Management
- Endpoint Integrity Protection
- Endpoint Identity
- Endpoint Root of Trust
- Endpoint Physical Security

Endpoint Data Protection

Endpoint Security Model & Policy
Threats and Vulnerabilities to an IIoT Endpoint

1. Hardware components
2/3. Boot process
4. Operating System
5. Hypervisor/Sep. Kernel
6. Non-OS Applications
7. Applications and their API
8. Runtime Environment
9. Containers
10. Deployment
11. Data at Rest, Data in Use
12. Monitoring/Analysis
13. Configuration/Management
14. Security Model/Policy
15. Development Environment
Endpoint security: Solutions

• Start with a **clean design** of the security **model** and **policies**
  • Define endpoint **identity, authorization, authentication**
    • How other endpoints see me? What can they do with me?
  • Define proper **data protection** model
    • Integrity and confidentiality, especially of shared data-in-rest but also data-in-use
  • Define secure **hardware, BIOS, roots of trust**
    • Includes lifetime of hardware, BIOS update, consistent root of trust
  • Select secure **OS, hypervisor, programming language**
    • Consider lifetime of (open source?), dynamic of programming language
    • Consider isolation principles (4 different models explained in IISF)
  • Plan **remote code update** and provide **code integrity**
    • Security has an unspecific expiration date: needs update
    • Code integrity prevents malicious remote code-hijacking
Endpoint security: Solutions (II)

• Plan “**beyond the basics**” security *instantly*
  • Plan security *configuration* and *management*
    • For example: defining, replacing and updating of keys and certificates
    • User-friendly setting of access rights and authorization
  • Plan endpoint *monitoring* and *analysis*
    • For example: log all security configuration changes
    • Log all unexpected remote activity
    • Provide user-friendly analysis, alerts etc.

• Implement “**state of the art**”:
  • Have a team of *experienced* security implementers
  • Use *latest* versions of development tools, OS, hypervisors, libraries
  • *Test* a lot, including *malicious* attacks
  • Prepare and test your first *remote update*