Cyber Security - IT Security Meets OT Security

Paul Didier, Solution Architect Cisco

October 10, 2018
Abstract

This session will look at how CIP Security can be used in-conjunction with current and in-development IT Security tools and technologies to significantly improve plant floor security and protection. We will discuss how to apply network access control and Plant and Cell-Area zone segmentation policies in a structured, scaleable manner. As well, we will review the value of monitoring network and security health through the use of traffic flow monitoring (NetFlow/IPFIX, IETF RFC 7011-7015) features and security monitoring applications. Lastly, we will look at some advanced, under-development standards to automatically apply trust and policy based on IETF standards such as Bootstrapping Remote Secure Key Infrastructure (BRSKI) and Manufacturer Usage Description (MUD) specifications.
• Why are we talking about this?
• A look at some of the standards and how they line up security-wise
• BRSKI – Bootstrapping Remote Secure Key Infrastructure
• MUD – Manufacturers User Descriptions
• How it could work
  – Scenario 1
  – Scenario 2
Gratuitous IoT Growth Chart

IoT Units Installed Base
Grand Total

2014: 3.8b
2015: 4.9b
2016: 6.4b
2017: 9.5b
2018: 12b
2019: 18b
2020: 25b+

The Network Administrator’s Problem

Number of Types of Things
A snapshot of Security Standards

**Onboarding (ID & Trust)**
- CIP Security: Pull & Push
- 802.1AR (DevID)
- Enrollment over Secure Transport (EST RFC7030)
- BRSKI

**Classification and Authorization**
- 802.1X Network Access Control
- xDS device descriptor
- Manufacturers User Description (MUD)

**Segment & Protect**
- VLANs
- ACLs
- TLS and DTLS
- Secure Group Tagging (Draft IETF)

**Monitor**
- Syslog
- SNMP
- IPFIX
Bootstrapping Remote Secure Key Infrastructure (BRSKI)
Zero-touch provisioning of IoT devices

• We want to encourage IoT device vendors to embed an X.509 certificate in their devices, and support IEEE 802.1X for authentication

• It would be even more valuable if we could safely automatically authorize IoT devices to be on the IT or OT network
  • IoT devices don’t have a UI, and manually adding large number of IoT devices onto the network won’t scale

• But when the X.509 certificates come from a variety of manufacturers, we aren’t able to easily share a common trust root, and thus can’t easily authenticate them
The necessary components

• The IoT Device is manufactured with an X.509 certificate and private key that conform to IEEE 802.1AR.
  • This is a standard definition of a “manufacturing certificate”

• The IoT device & Production network support a multi-vendor bootstrapping protocol (BRSKI)
  • draft-ietf-anima-bootstrapping-keyinfra-04

• The IoT device manufacturer supports a Manufacturer Authorized Signing Authority (MASA), which keeps a log of which devices have been installed in which IT or OT domains

• At the end of the BRSKI protocol, the IoT Device and Production network have a mutual trust, and the IoT device can be admitted to the network.
• The IoT Thing requests some assurance that it should join this network.

• The request is forwarded to the MASA server, which returns a signed “audit token” to the network, which also forwards it to the IoT Device.

• At the end there is mutual trust provided by the MASA server.

 "request for assurance that domain is safe"

 "audit token" returned

 "audit token" returned

 "audit token" returned

 "audit token" returned

 Request sent over IPIP

 Request sent to MASA server

 Manufacturer MASA Server

 Plant Data Center

 AAA/BRSKI Server

 Network

 IoT Thing
Manufacturing User Descriptions (MUD)
Translating intent into config

Any intended use can be clearly identified by the manufacturer

access-list 10 permit host controller.mfg.example.com

All other uses can be warned against in a statement by the manufacturer

access-list 10 deny any any
• **draft-ietf-opsawg-mud-01**
  • Defines the format of a MUD file (YANG/JSON) for describing policy about a device, such as its network access policy. This network access policy can be turned into ACLs and placed on a network access port
  • Defines a new URI for referencing the MUD file, which describes the location of the MUD file on a public web server.
  • Defines protocol extensions that a IoT device uses to forward the URI to network devices
    • X.509 certificate extension
    • DHCP option
    • LLDP TLV
Actors in a MUD scenario

- **IoT Thing**: Emits the MUD URI
- **MUD Controller**: Receives the MUD URI (forwarded from a AAA server), resolves the URI to recover the MUD file, translates the MUD policy into ACLs, and delivers the ACLs to the AAA server
- **Manufacturer Web Server**: Serves the MUD file to all comers
The preferred method of URI distribution

- MUD URI is placed in the IoT Thing certificate, used by 802.1X/EAP-TLS
MUD without IEEE 802.1X/EAP-TLS

When MUD URI is distributed in DHCP or LLDP, the port starts with a default policy that only allows packets necessary to provision the IoT Thing (e.g., DHCP or LLDP)
Results: Micro-segmentation

- Access limited to devices based on manufacturer recommendations
- Policy choices easily identified by MUD file
- Hacked devices can’t probe for holes
- An additional layer of security
  - BUT- manufacturers should still **always** secure their devices
Status of MUD

- draft-ietf-opsawg-mud-25 is an IETF Proposed Standard
- Additional capabilities to be introduced over time
Scenario 1: Easier Use of LLDP to communicate manufacturer usage descriptions
Initial Configuration (the same as before)

Plant Network

- Capable network connectivity
- MUD file server would sit on the Internet
- Hacked system maybe authorized but not known to be a “Production” Thing (perhaps a laptop)
Onboarding Process
MAC authenticated Bypass (MAB)

Plant Network
• Device does not do any authentication
Device States What It Is via LLDP MUD-URL

Plant Network
- https://example.com/.well-known/mud/v1/mudfile
- LLDP information forwarded to ISE
- Address information collected
Retrieve Manufacturer Information

Plant Network

• AAA server gets profile information from MUD controller
• Controller retrieves the information from the Manufacturer
Approval and Access List Generation

Plant Network

- Approval, if required for initial device types
Issue Change of Authorization

Plant Network

AAA Server

MUD Controller

Manufacturer Server
Scenario 2: Harder
Full scale manufacturer certificate authentication and authorization (the big one)
Initial Configuration

Plant Network

- Capable network equipment
- MUD file server would sit on the Internet
- Hacked system is authorized but not known to be a Thing (perhaps a laptop)
Enterprise Network

- Assume hacked device has general access
- Onboarding device is initially isolated
Bootstrap
Find Registrar

Enterprise Network

Registrar
MUD Controller
Manufacturer Server

mDNS Registrar Discovery
• Device transmits its manufacturer certificate
• This contains URL of manufacturer (MUD file) server
Bootstrap

Retrieve MUD File from Manufacturer, to Find MASA Server
Request a Voucher to Send to the Device
Administrative Approval, if Required
(Probably Only First of This Type of Device)
Install Trust Anchor and Perform EST Registration to Obtain A Local Certificate

Yay!! At the end of this step, the device is registered!
Next up: determine its access
Now Device Authenticates Using 802.1X

Enterprise Network
- 802.1X from device to the switch
- EAP-TLS over radius from switch to the AAA server
Determine Appropriate Authorization

- AAA server requests profile from MUD controller
- No need to retrieve MUD file (we already did that)
Effect Change of Authorization

Enterprise Network
- RESTful interface to MUD controller
- Radius back to switch
More information

- draft-ietf-opsawg-mud-25.txt (MUD)
- https://mudmaker.org
- draft-friel-brski-over-802dot11-00.txt (Problem statement)
- Information list: mud-interest@cisco.com
- IETF WGs: opsawg@ietf.org (MUD) anima@ietf.org (BRSKI)
THANK YOU