Getting Started with Secure Software

Class 2: Performing a Security Threats Analysis

April 21, 2020

Jacob Beningo

Presented by:

DesignNews

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

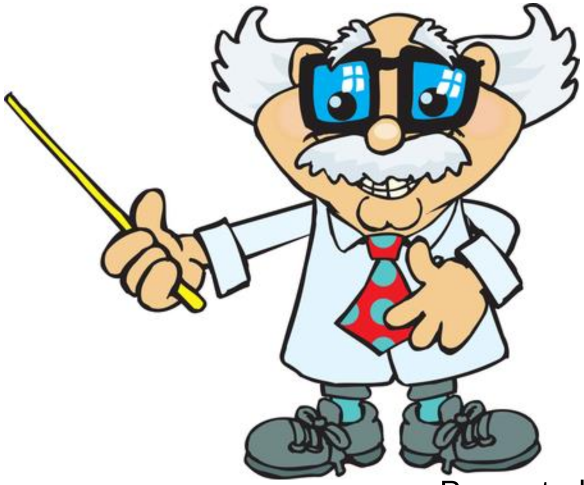
Course Overview

Topics:

- Introduction to Platform Security Architecture (PSA)
- **Performing a Security Threats Analysis**
- Architecting a Secure Solution
- Secure Boot and the Root-of-Trust
- Secure Frameworks and Ecosystems

Session Overview

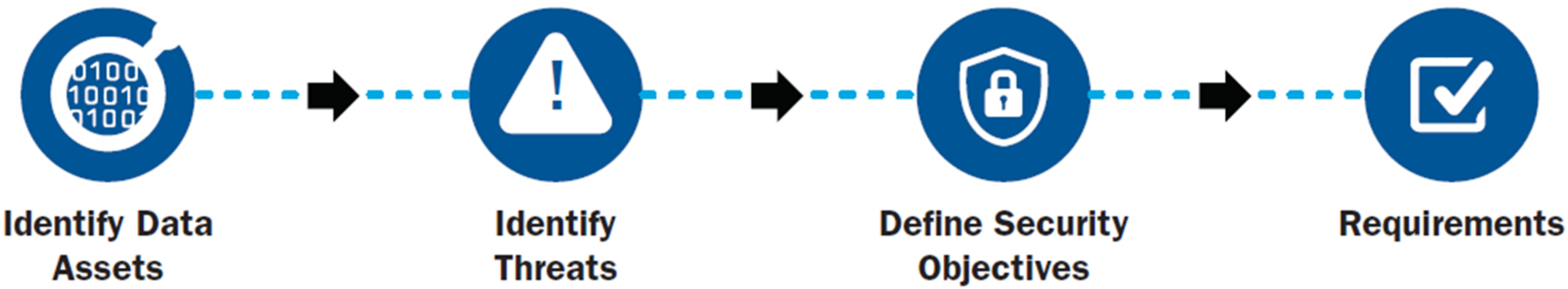
- Threat Based Analysis Overview
- Threat Analysis Steps
- Protecting Data Assets



Presented by:

Threat Based Analysis

The Process



Threat Based Analysis

A Networked Camera

- Used to stream live video to a remote location
- May be used to periodically capture still images or be activated by detected motion
- Video stream is transmitted in a compressed form
- May be used for:
  - Personal use (baby monitors, doorbells, security, etc
  - Enterprise general use (security, event detection)
  - Enterprise high security use (protect high value assets)

Step #1 – Identifying Data Assets

Data assets that exist in nearly all IoT devices include:

- The firmware
- Unique ID
- Passwords (flash, users, etc)
- Encryption keys (to control device, secure communication, etc)

Device specific data assets might include:

- Image data
- Sensor data
- Control data

Step #1 – Identifying Data Assets

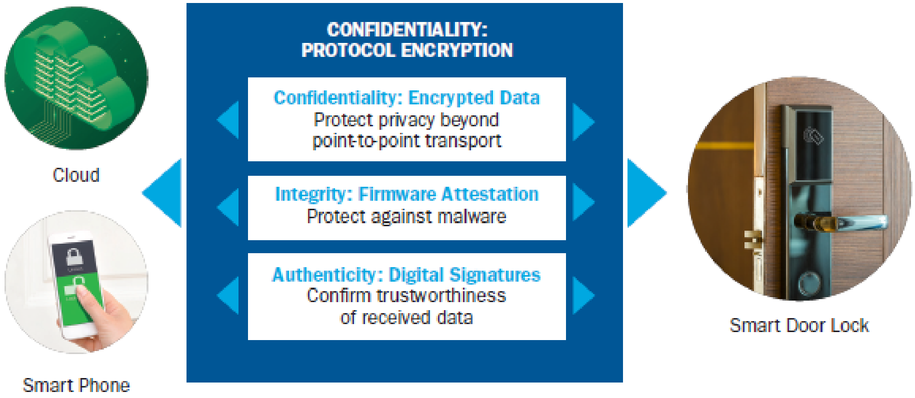
Data assets that exist in a Networked Camera:

Data Asset	Description
Camera Id	A unique identifier for the device
Firmware	Defines how the hardware operates
Firmware Credentials	Used for secure boot
Credentials	Data used for cryptographic operations
Logs	Historic data
Images	Data captured by the camera and sent over the network
Configuration	Data used for configuration, including network configuration

Protecting Data Assets

Confidentiality - the state of keeping or being kept secret or private.

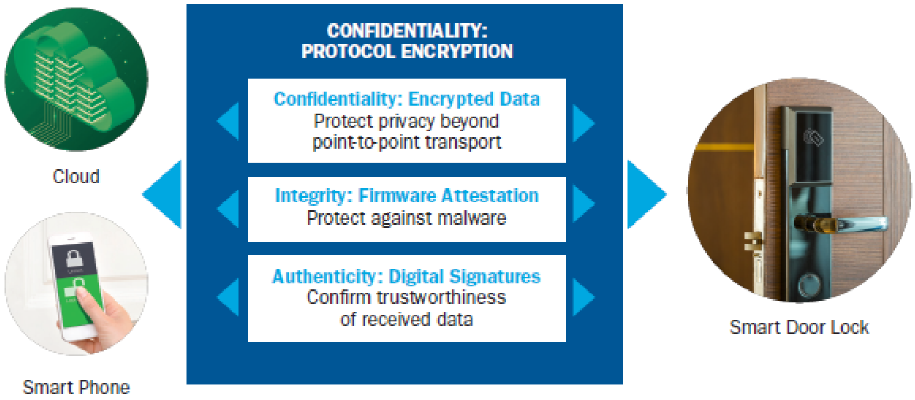
- requires that only authorized people can read the data asset.
- it is kept secret or private
- Data assets that require confidentiality include:
  - Passwords
  - Personal data generated by the IoT device
    - Heart rate
    - Location data



Protecting Data Assets

Integrity – the state of being whole and undivided.

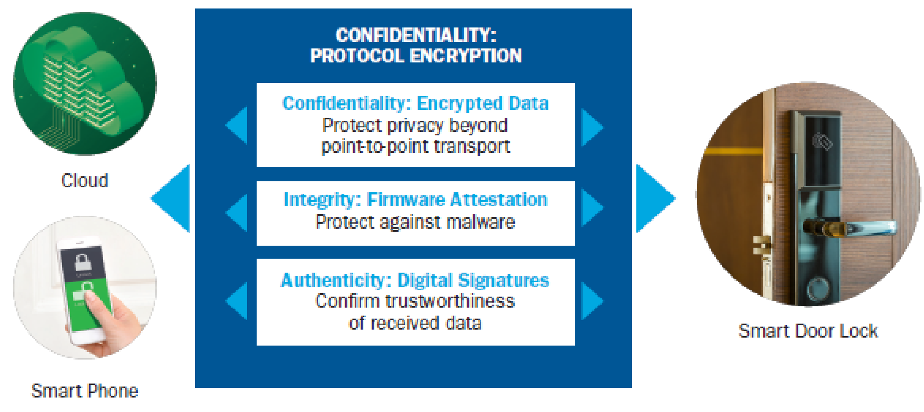
- requires that a data asset remains unchanged through its use or transferal.
- Data assets that require integrity include:
  - Boot firmware (ensures that the MCU initializes to a known initial state)
  - Device configuration
  - Credentials
  - Firmware
  - etc



Protecting Data Assets

Authenticity – the quality of being authentic (undisputed origin; not a copy, genuine)

- Requires that only a trusted actor has established the current state of a data asset
- Example data assets requiring authenticity include:
  - Firmware images



Combining integrity and authenticity establishes trust!

- Digital signature can be used to evaluate and integrity of new firmware
- Digital signature can be used to evaluate and integrity of existing firmware

Step #1 – Identifying Data Assets

Data assets that exist in a Networked Camera:

Data Asset	Confidentiality	Integrity	Authentication
Camera Id		✓	
Firmware	✓	✓	✓
Firmware Credentials		✓	
Credentials	✓	✓	
Logs		✓	
Images	✓	✓	
Configuration	✓	✓	

Step #2 – Identify Threats

Threat	Targeted Data Asset	Confidentiality	Integrity	Authentication
Impersonation	Credentials	✓	✓	
Man in the Middle	Credentials			
	Images			
	Configuration	✓	✓	
	Confidentiality	✓	✓	✓
Firmware Abuse	Firmware	✓	✓	✓
Tamper	Camera ID		✓	
	Firmware	✓	✓	✓
	Credentials	✓	✓	✓
	Logs		✓	
	Images	✓	✓	
	Configuration	✓	✓	

Step #3 – Identify Security Objectives

Access Control - The IoT device authenticates all actors (human or machine) attempting to access data assets. Prevents unauthorized access to data assets. Counters spoofing and malware threats where the attacker modifies firmware or installs an outdated flawed version.

Secure Storage - The IoT device maintains confidentiality (as required) and integrity of data assets. Counters tamper threats.

Firmware Authenticity - The IoT device verifies firmware authenticity prior to boot and prior to upgrade. Counters malware threats.

Communication - The IoT device authenticates remote servers and provides confidentiality (as required) and maintains integrity of
exchanged data. Counters Man in the Middle (MitM) threats.

Secure State - Ensures that the device maintains a secure state even in case of failure of verification of firmware integrity and
authenticity. Counters malware and tamper threats.

Step #3 – Identify Security Objectives

Threats

Security Objectives

	Impersonation	MitM	Firmware Abuse	Tamper
Access Control	X		X	
Secure Storage				X
Firmware Authenticity			X	
Communication		X		
Secure State			X	X

Step #4 – Defining Requirements

Security Objective	Countered Threats	Targeted Data Assets	Security Properties[2]	Design	Mfg	Inventory	End Use	Term
Access Control[1]	Spoofing Malware	Configuration T. Firmware	C I, A	N/A Dig Sign	N/A Dig Sign	N/A N/A	Encryption Dig Sign	Dead[4] Dead[4]
Secure Storage[1]	Tamper	HW ID T. Firmware User Data Configuration Keys	I I, A C, I C C, I	N/A Dig Sign N/A N/A N/A	eFuse Dig Sign N/A N/A SEF[3]	eFuse Dig Sign N/A N/A SEF[3]	eFuse Dig Sign Encryption Encryption SEF[3]	eFuse Dead[4] Dead[4] Dead[4] Dead[4]
Firmware Auth	Malware	T. Firmware	I, A	Dig Sign	Dig Sign	Dig Sign	Dig Sign	Dead[4]
Comm[1]	MitM	User Data Keys	C, I C, I	N/A N/A	N/A SEF[3]	N/A SEF[3]	Encryption SEF[3]	Dead[4] Dead[4]
Secure State	Malware Tamper	T. Firmware HW ID T. Firmware User Data Configuration Keys	I I, A I, A C, I C C, I	Dig Sign N/A Dig Sign N/A N/A N/A	Dig Sign eFuse Dig Sign N/A N/A SEF[3]	Dig Sign eFuse Dig Sign Encryption Encryption SEF[3]	Dig Sign eFuse Dig Sign Encryption Encryption SEF[3]	Dead[4] eFuse Dead[4] Dead[4] Dead[4] Dead[4]

Step #5 – Leverage the Requirements

Select a Secure Microcontroller

Secure MCU Features

- Encryption
- Digital signature
- eFuses
- Isolated execution environment for trusted applications
- Secure element functionality

PSA

Additional Resources

- Beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm
- Platform Security Architecture:
  - www.arm.com/psa
- Threat-based analysis method:
  - www.cypress.com/psoc6security

From www.beningo.com under

- Blog > CEC – Getting Started with Secure Software