VIEW POINT

FIRMWARE SECURITY FOR IOT DEVICES: DISCOVERING KEY RISKS THROUGH REVERSE ENGINEERING AND BEST PRACTICES IN RISK MITIGATION

Infosys®
Navigate your next

Connected things are connecting human lives

As of 2019, we have over 5 billion active mobile devices, most of which are used for communication, data and applications covering all industry verticals including health care, telecom, retail, media, social media and payments. Industrial IoT has received wide acceptance and adaption in the manufacturing, oil & gas, mining and energy sectors. The usage of connected devices in the IIoT is helping to enhance and optimize the automation in these industries.

Connected devices and their modes: Active devices vs Passive devices

The interconnected devices either communicate with self-generated data or with data collected from other interconnected devices. Any data that a device processes has its own defined mode. Primary difference between active devices and passive devices are that of data generation and processing capabilities.

The following table shows the major differences between active and passive devices.

Active Devices	Passive Devices
Continuously or frequently processes or generates data over IoT connected networks.	External source has to make a call to process or generate data to connected devices.
Active device associated with the microcontroller, memory, transceiver etc.	Passive device may or may not be associated with the microcontroller, memory, transceiver etc.
Own the power source.	Does not own the power source.
Capability to self-trigger the data for send/receive operations.	External device has to trigger the data for send/receive operations.
e.g. Light Sensors, Wireless Sensors, RFID etc.	e.g. ATM Card, Barcode etc.

Top 5 security issues in connected devices

Insecure transport layer communication

Communication in an IoT ecosystem is a major aspect while processing business operations with connected IoT devices. Many times, IoT devices send control commands, API session tokens, collected data from I/O devices to the IoT Cloud for processing.

Failing to use transport level encryption while transmitting such sensitive data puts the IoT ecosystem at a risk of eavesdropping or receiving MiTM kind of attacks. Since this is one of the easiest and well-known forms of attack, anyone with some basic knowledge would be able to manage it with minimal efforts.

Weak cryptography

IoT devices use passwords, cryptographic keys, PKI certificates to protect the firmware state from unauthorized hands. While protecting the estate, firmware developers may fail/forget to store security assets into the firmware. Hardcoding passwords or storing cryptographic keys, PKI certificates into the firmware can give a free hand to hackers to modify existing passwords, replace certificates, download back doored version of firmware over OTA updates thus leading to complete loss of integrity & confidentiality of the firmware.

Listed below are a few security risks:

- Weak cryptographic key size
- Reuse of existing cryptographic keys/ certificates
- Insecure cryptographic key/certificates storage
- Insufficient protection of cryptographic keys (e.g. Encryption Keys, Private Keys, PKI Certificates)

Weak authentication

In the IoT ecosystem, components such as web/mobile applications, IoT cloud, gateways & other connected devices in the network play critical roles for business control & data processing. While doing this, each component needs to verify the identity of the devices from the connected devices network before it collects or processes data from the I/O Systems, thus establishing a chain of security trust in the interconnected device network.

Web UI or mobile applications make use of a number of APIs for performing various user-controlled actions. These user actions are performed by requesting API calls. Requesting APIs from unauthenticated devices/resources to the cloud could lead to potential unauthorized actions.

Given below are a few security risks:

- Unauthenticated API request calls from unauthorized devices
- No interconnected device authentication
- Insecure open cloud instances for access

Security misconfiguration

No matter what or how many security measures you have taken into consideration while designing the components architecture of the IoT ecosystem during implementation, it can still have loopholes while configuring those devices/applications, at the deployment phase.

Below are a few misconfiguration risks:

- Setting up default configuration
- Disabled security event logging
- Using weak cryptographic algorithm from the best available encryption algorithms,
- No alerts or notification on security events
- Using default/no password etc

Insecure operating systems / firmware

The inability of firmware to upgrade to its higher version presents high risk of getting exploited either by newly discovered zero day or existing past security vulnerabilities.

Stated below are a few security issues in the firmware:

- No updated functionality
- Unsigned/unencrypted OTA file updates
- Use of weak or insufficient SSL/TLS encryption channel
- No mutual/2-way authentication between device and cloud update pool

Discovering security risks in firmware through reverse engineering

Every IoT device that we see around us such as camera drones, smart TV, CCTVs, air conditioners are embedded with the firmware. Basically, firmware is binary which has a set of predefined instructions coded/installed on the specific OS (e.g. DOS, Linux, Symbian, CISCO IOS, Windows CE/NT) with limited hardware capability. It also includes bootloaders (e.g. U-Boot, RedBoot, BareBox etc.), common libraries etc. Stated below are the tools and methodologies used for extracting the firmware image:

Tools required for reverse engineering:

1. binwalk
2. dd
3. strings
4. hexdump/xxd

Steps to perform reverse engineering

i. Identify the target IoT device firmware for reversing

ii. Investigate device processor architecture

iii. Understand the firmware header, compressed file system and data (e.g. lzma) using binwalk utility

```
# binwalk <firmware_file>.[bin/zip]
```

-- it detects & lists all identified firmware signatures such as bootloader type & version, firmware version, firmware header, file system, data & its compression type, certificates, library file paths and offset for each identified signature of the firmware

iv. Extract file system from firmware into compressed format using dd utility. Decompress it and mount decompressed file system onto mount point

v. After file system mount, all the firmware specific system directories would be listed

vi. Similarly, for data section of the firmware, human unreadable data can be viewed using strings or hexdump/xxd utility

Recommendations for securing a device's firmware

Underestimating device firmware security could lead to real time hacks, compromising the IoT devices at your home and work network. What are the ways to secure device firmware to protect it from security issues?

Below are the recommendations to secure firmware in the connected devices:

1. Enforce strong SSL/TLS communication between interconnected devices
2. Encrypt & digitally sign the firmware binaries to preserve its confidentiality & integrity for OTA updates
3. Mutually authenticate firmware device and cloud update pool using PKI
4. Establish secure chain of trust among interconnected devices
5. Keep the audit trail of failed, error, anomalous and critical business transaction logs
6. Enforce secure boot process to prevent from modifying/replacing back doored firmware
7. Ensure standard PKI management processes are followed for private keys and certificate store
8. Configure IoT ecosystem components knowing its possible security threats for critical business functions in the firmware
9. Enable authentication on IoT ecosystem API interfaces
10. Place a physical security measures protecting IoT devices from unauthorized physical/local access attacks

About the Author

Suhas Desai, Industry Principal, Infosys

Yogesh Shelke, Associate Consultant, Infosys

For more information, contact askus@infosys.com

Infosys®
Navigate your next