Introduction

At beginning of internet, the network and websites were created which people use to share, communicate through. As the technology enriched this machine to person communication changed and become machine to machine communication for eg. GPS device sending location to your phone and giving us exact location. This machine to machine communication is called IoT.

The Internet of Things (IoT) refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.

Some of predication by McKinsey Global Institute reported that the number of connected machines has increased by 300 per cent over the past few years. By 2025, the economic impact of IoT is estimated to range from US$ 2.7 trillion to US$ 6.2 trillion. Wikibon predicts that the value created from the Internet will be about US$ 1279 billion in 2020, growing annually at a rate of 14 per cent.

The pace at which Internet of Things (IoT) continues to gain traction, security becomes a major concern. Businesses are increasingly being breached by Attackers via vulnerable interface which leads to loss of confidentiality, integrity and availability. When it comes to the data flow between devices, there is always a chance that the data can be accessed or read when getting transferred. Tremendous increase in the consumption of IoT products and services are globally spread across different vertical like heath Care, Retail, Automotive, and Hospitality. Not counting mobile phones and traditional network devices like PCs, industry analysts forecast upwards of fifty billion network-accessible "things" by 2020.

Challenges

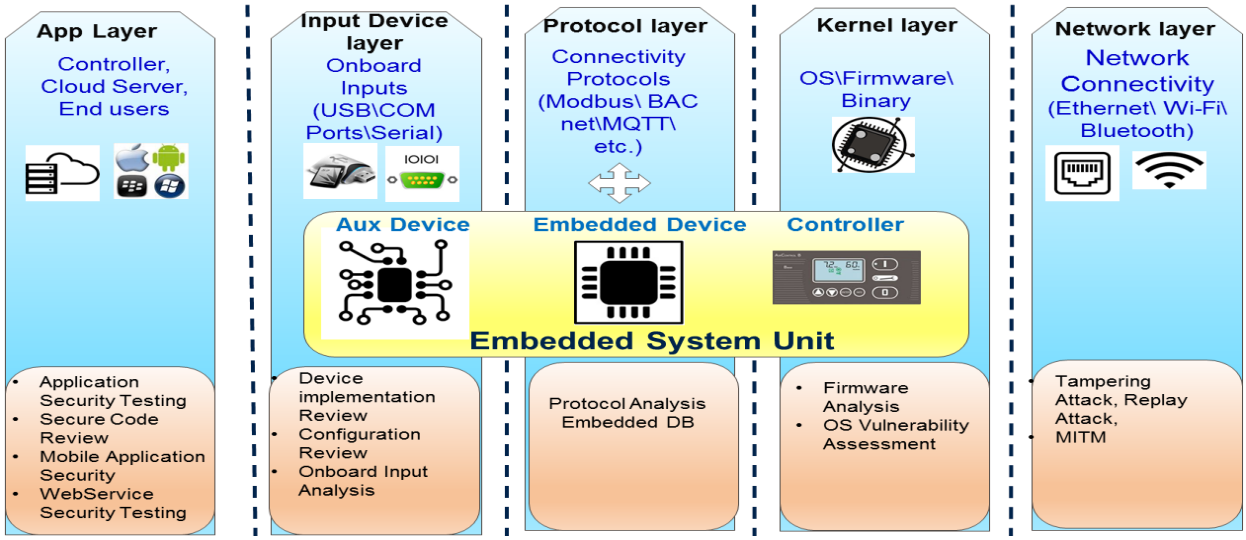
1. **Secure Transmission of Data.** All MQTT, BACnet, XMPP, CoAP traffic is encrypted over an SSL connection. All Console access is exclusively available over an encrypted HTTPS connection. All REST API access is exclusively available over an encrypted HTTPS connection.
2. **Logical Access to Data Store.** All databases should be protected through strict firewall rules from external access and they are only accessible from the mid-tier machines. In the database, data should be segregated by account through a unique tenant Id. At the MQTT broker, broker data and traffic should be segregated between accounts using virtual machine segregation.
3. **Identity and Access Management**. Confidentiality and integrity needs to be ensured through a role based access control model and access control lists which follow the Principle of Least Privilege and are enforced through all the layers of the architecture. Each account manages a list of users and controls the user's credentials.
4. **Vulnerability Management.** Independent certified security firm performs remote vulnerability assessments, including network/host and applications.
5. **Patch Management.** Specialized IoT hardware platforms like **Arduino and Raspberry Pi** Starter Kits come with all the goodies help deploy IoT. Since they are readily available and require less investment than designing and fabricating custom printed circuit boards (PCBs) at each iteration of the design. Often operating system running on such devices don't get download and install these patches and upgrades. Also because of most of the time the **OS/Packages/Protocols used are open source** where we have to wait for patches and upgrades.
6. **Exposure to Sensitive data.** IoT systems are often used to gather sensitive data such as medical biometrics, personal schedules, and inventory and supply chain information, etc. Even

when system is not gathering any data, it store enough contextual data that is useful for analytics tools and cross-reference systems. **Range of protocol** are used to interconnect IoT devices which we have to ensure that the data in motion and data at rest is safe guarded.

7. **Multiple user interfaces**. All the interfaces made to be responsive, intuitive, and built to provide the best user experience while doing this the developer forgot to maintain the best practices which may lead partial or full comprised IoT System. Many of these interfaces have been found to be vulnerable to common and known types of vulnerabilities, including the following:
    a. Use of unauthenticated requests to perform actions (for example, reconfiguration, data retrieval, management functions, etc.)
    b. Ability to perform unrequested firmware upgrades
    c. Command injections
    d. Buffer/heap overflows
    e. OWASP's List of the Top Ten Web Vulnerabilities
8. **Cloud Backend Infrastructure.** IoT systems Cloud infrastructure include a back-end cloud service, depending on the category of the device. Access to the cloud service through a smartphone application or a web portal, where users can log in. Unfortunately, Most of them allow user to choose weak password. Few cloud interfaces have an unsecure password recovery method or reveal too much information during the recovery process, such as displaying the validity of an account and other personal details. Some cloud services have logical errors, which could allow an attacker to obtain sensitive customer information or access devices without authentication. These services also contained common management console vulnerabilities, including those listed in OWASP's List of the Top Ten Web Vulnerabilities.

We have following approach to counter Vulnerability and Configuration Management of IoT solutions. Following layered approach will ensure IoT Solution security at each layer.

Methodology



App Layer:

In this particular layer, we considered the user and admin interfaces to access the Iot device.IoT management Interfaces can be a web/mobile application. We performed automated web/Mobile

application testing followed by a manual assessment to remove false positive. Next phase we conduct exploit research and development to verify the injection vulnerabilities.

Input Device Layer:

In this particular layer, we analyze the onboard input device like USB,COM ports, Serial Ports, SD Card Slots,.etc.

Also perform the Device Implementation and configuration review which consist of IoT Design Review, configuration review set up to handle the onboard inputs.

Protocol Layer:

In this phase we will be testing the protocols used by device to communicate with each other and data sent by the device.

Kernel Layer

Generally, IoT devices memory is split in two parts one is firmware and other is configuration storage. It will have a large amount of RAM, which is usually a few times the size of the flash storage. The processors in the devices vary greatly but they are usually ARM or MIPS based, these are low cost processors. A devices firmware will consist of a firmware header, a boot loader, a Linux kernel and a file system.

In this particular phase we perform, firmware unpacking and modification. Also to check binary protection

Perform below assessments.

- Static binary analysis.
- Checksum Controls
- Code obfuscation
- Certificate Controls
- Firmware reversing

Apart from this it is important to check Firmware downgrade possibility, Sensitive information disclosure, hardcoded credentials etc.

Network Layer

As the most of the IoT system rely on the data gather from the sensor, other IoT device. It becomes important to analyze the data in motion as well as the port/services accessible from the network. To safeguard IoT from network layer we perform vulnerability assessment on the device as well as attacks like MITM, spoofing, etc. Same way it is recommended to conduct wireless audit is IoT System is using Wi-Fi for data sharing.

Take Away

- Build solutions based on **open and industry standards**
- Leveraging **proven** IT/enterprise/Internet class **security technologies and partnerships**
- Including **security**, scalability and resiliency **in design** from day one
- Security technology best practice has **to take into account the specific aspects of distributed, unattended, mobile systems / devices**
- Security has to be implemented **end-to-end and in the individual elements**
- **Encapsulate the complexity** of an end-to-end security solution
- Continuous **testing and auditing**