EC601 Project 1 Report: Network Security for IoT Devices

- **Author:** Zihao Diao <zhdiao@bu.edu>

Introduction

Internet of Things, or simply **IoT**, is used to describe a vison of how people will use the internet now and in the future. It integrates different technologies including the communication networking, sensors, control and intellegence system together. The specific use of IoT may vary depends on the different scenarios, including the home, industrial, businness, healthcare and so on. But the basic of them is the same. IoT wants to connect machines, sensors together, enabling them to interact with the physical world, without or with little intervations from human.

Generally, IoT requires more than thousands of devices to be connected to the internet. Those devices can be very diverse, in the sense of many ways including the functionality, computational power, the way it is connected to the internet and so on. The scale and diversity of the whole system impose greater challenge to the security of the whole system. This challenge can be described at least in 3 ways:

- the architectural level, which includes problems of desgning a secure architectures for the IoT network;
- the implementation level, which includes problems of design and validate a secure implementations of the system and protocols used in the IoT network;
- the adminstrative level, which includes problems of allowing human (esp. the system administrators) to efficently manage the whole system, preventing the human errors.

Keoh et al. borrorws the concept of layering from the classic network layering model, proposed the 4 layering of IoT security: infrastructure security, communication network security, application security and general system security. In this report and in the whole project, the focus will be on the communication network security of a IoT network.

Real World Challenges

In this section, I will case study some of the typical network security problems faced by a modern IoT network.

- **Distributed Denial of Service Attack**: Despite the fact that the IoT devices are typically of limited power consumptions and thus have very limited computational power and network bandwidth, the massive amount of them can really be used by attackers to launch massive DDoS attack to the network infrastructure. This is especially a problem when take the problem in the adminstrative level into consideration: in practice, many IoT devices is leaking essiental updates to the software running on it. The out-of-date software may have many vulunerbities that can be used by attackers to gain controll of the device which means ease of creating a robonet that can be used for DDoS attack.
- **Man-in-the-middle Attack**: The end to end encryption principle is widely accepted by many people and is

implementated in most modern systems, including IoT devices. E2E encryptions provide IoT devices with a sound way to allow private and authentic communication. But there are still devices that does not ship with E2E encryption during communicating. Also, some devices are using outdated protocols and algorithms when doing communications. In both cases, the private and authentic promise by the E2E encryption will be no longer true. The communication content may be compromised to a third-party or the content of the communication may be altered. The data leak is especially harmful when considering that IoT devices are used to interact with real world. A leaked sensory data, like live camera stream, or altered control signal may have huge impact on real world.

- **Unauthorized Access**: On top of the DDoS and MITM attacks, there is one starting point from which the problems arise. That is the relative ease of being able to gain unauthorized access to an IoT device. The architecture of modern internet makes it easy to launch large scale port scanning. Due to the large scale of the deployment of many certain IoT devices, one vunlnblity in the software or hardware design may be used with the data gained from the part scanning to gain unauthorized access to thousands of devices on the internet. Some misconfigurations in the network system (i.e., the leak of proper firewall, unneeded port forwarding rules on gateways) may make it easier for a hacker to gain access to massive ammount of devices.

Also, there is challenges that make building a secure IoT network more difficult than building a secure ordinary network. Those challenges includes:

- **Power consumptions & limited computational power**: Many IoT devices are expected to operate for months and years on a single AAA battery without user intervention. The power and computational limit makes it to be almost impossible to implement the existing secure protocols used by more powerful devices. Thus some new protocols needs to be proposed and used or changes must be done to existing protocols.
- **Heterogeneous nature**: the IoT device itself is heterogeneous. The working environment of the devices is also heterogeneous. This makes things like the protocol design and implementation, firmware update and many other aspects more complicated.
- **Private Protocols**: Vendors tend to use their private protocols in their IoT device products. Some of those protocols may be poorly designed, leaking some key security features. Some implementation of those protocols may be buggy thus decrease the security of the system.

Requirements

The following requirements of network security is also true for IoT networks from a high level prospective:

- **Authenication**: The identity of a device/people must be validated before an access could occur;
- **Access control**: Authorization of access must be granted to a device/people according to their identity;
- **Non-repudiation**: Device/people cannot dispute the authorship of an action.

Exsisting Solutions

Solutions are built to provide secure access to the IoT devices. In the following section, I will list some of them by the relative location of it in the network layering.

Physical and MAC Layer

Protocols like Wi-Fi (IEEE 802.11), Bluetooth and celluar network is also widely used by the IoT devices as its physical layer protocol of communication. Some other protocols are designed specifically for IoT devices. Those protocols includes Near-field Communication (NFC), ZigBee and Z-Wave. In some cases, wired connection via Ethernet cable or coax cable is also used.

Those protocols generally provides some kind of authenication and access control for IoT devices. For example, the ZigBee, Bluetooth and Wi-Fi allows authenication and access control of a device via pre-shared keys (PSK), PIN or digital certificate. In some cases, the authenication and access control are granted by the ablity to physially access a certain resource. This is true for the Ethernet and coax cable case. In both of them, the ablity to access the cable (media) implies the authenication and privellage of accessing the media. However, proceeding updates to the Ethernet protocol, including IEEE 802.1x, allow more explicit validates the authenication of a client and grant access control to it.

Network and Transport Layer

For the IoT devices that uses TCP/IP based network and transport layer protocols, existing protocols designed for the IP protocols can be used to achieve the three requirements for IoT network. Those protocols includes IPSec that runs on the network layer and Transportation Layer Security (TLS) on the transport layer. IPSec and TLS can provide all three requirements: authenication, access control and non-repudiation. For other protocols, some similar protocols are used. Most of those protocols are based on the asymmetric encryption algorithms like RSA and some kind of the public key infrastrcture (PKI).

Application Layer

It is very flexiable for vendors to design there application layer protocols. Auuthenication and access control can also be implemented on this layer. This implementation is largely vendor-specific. Those protocols are also largely used for exchange of the authenication informations during the adoption and pairing phase.

Case Study

OpenWiFi

OpenWiFi is an open source IEEE 802.11 implementation on FPGA and software-based radio (SDR). It can serve as an analysis tool for the physical and MAC layer implementation of many IoT device using the Wi-Fi (IEEE 802.11) protocol. The project is now under active development by a small group of less than 10 developers.

Licensed under the AGPL 3.0 license, OpenWiFi allows researchers to modify the existing IEEE 802.11MAC layer implemented running on a FPGA to explore the protocol. OpenWiFi implements the physical layer (PHY layer) and the medium access control layer (MAC layer) of the IEEE 802.11a/g/n protocol on the Xilinx Zynq series FPGA system-on-chip (SoC). Also, OpenWiFi also implements the mac80211 API in the Linux kernel. This allows it to be used as a fully functional network card in the on-chip Linux operating system. The IEEE 802.11 implementation on the open source FPGA provides the advantages of transparent implementation and lower latency to solve the above problems. It should be noted that due to delay requirements (for example, the minimum inter-packet interval needs to be controlled at the level of tens of microseconds), the IEEE 802.11 protocol stack can hardly be implemented in ways other than FPGAs or dedicated integrated circuits.

Home Assistant

Home Assistant is an open source application runs on the application layer. It acts like a bridge over IoT device that runs on different protocols like Wi-Fi, ZigBee and Bluetooth allowing user to create automation across different protocols.

Lisenced under the Apache license, Home Assistant allows users to freely modify the software and extent the functionality of the software. The design of Home Assistant also take into the consideration of network security. Despite the heterogeneous nature of the devices connected to it and protocols it support, Home Assistant provides all those modules with a unified interface for authenication and access control. The implementation of every module is also under active peer review.