Chip off IoT Devices: Attacks and Mitigations

Gottumukkala Eswar Kumar Raju, Lakshmy K. V

Abstract*: *Internet of Things (IoT) is the most emerging field in Information Technology. IoT will make real-world objects into virtual objects and allows them to be sensed remotely. The IoT biological community is changing and getting to be appealing over a more extensive scope of controls, consequently their development in prevalent innovation. Security and protection are the key issues for IoT applications and still face some huge difficulties. Pervasive IoT gadgets have genuine security suggestions as they happen in extensive numbers, are topographically dispersed and can be hard to physically verify. These gadgets may contain touchy or economically profitable information making them appealing to different types of assault. It is discovered that a large portion of the gadgets is defenseless against assaults on hardware, firmware as well as web interface level. In this paper, vulnerabilities due to Insufficient Security Configurability, Insecure Firmware, Security Misconfiguration of IoT devices, recommendations for those vulnerabilities and an approach to reproduce those attacks is explained so as to provide better security to IoT Devices.

Index Terms*: *Firmware, Hardware, Internet of Things Vulnerabilities.

I. INTRODUCTION

The Internet of Things (IoT) portrays a reality where universal gadgets, for example, sensors altogether fit for speaking with the Internet. This is certainly not another idea and has existed over various territories, for example, retail, car what's more, modern applications for a long time. Numerous wellbeing and security frameworks, for example, medicinal gadgets, CCTV cameras etc. are as of now IoT gadgets, associated with systems which have the ability to process, store and deal with the information delivered by every one of the sensors [1][2][3]. It has become a fundamental part of the present-day epoch as it provides an ease of life with simpler control and vast connectivity through innumerable applications [4][5]. Human life has become more dependent on IoT. As a result, hackers are also focusing more on IoT devices to find the vulnerabilities in the devices and exploit them. IoT has a lot of Applications in various fields like Health Care, Agriculture, Telecommunications,
Automation, Oil and Gas, Transportation etc. [6][7]. Security vulnerabilities in the IoT domain lead to threats and attacks which can compromise critical infrastructures and National security and results in financial and physical loss [8].Generally, most of the IoT devices will have a web interface and supports mobile application. The items that are in the scope of IoT are Physical Device/Hardware, Firmware, Mobile Application, Web Application, Bluetooth and Wi-Fi.

Revised Manuscript Received on July 05, 2019.

Gottumukkala Eswar Kumar Raju, TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham University, Coimbatore, India.
Dr. Lakshmy K. V, TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham University, Coimbatore, India.

The brief attack surface of IoT devices is shown in Figure 1.

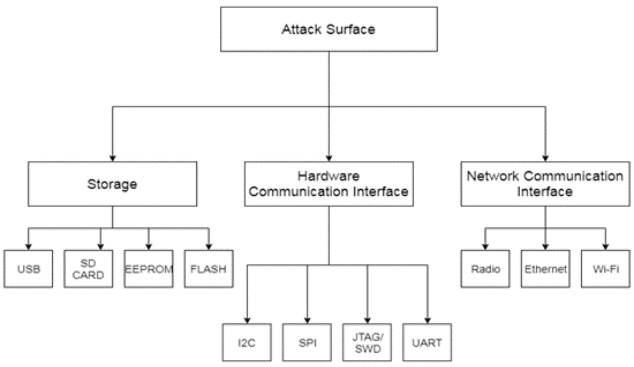

Figure 1: A General Attack Surfaces on IoT Devices

II. GENERAL ATTACK APPROACH ON IOT DEVICES

There is no standard procedure that can be applied for all the devices. The basic concepts of IoT will help to start working with the IoT devices. The general approach to be followed for any IoT device is explained in detail below:

1. Open the IoT device and locate the circuit boards.
2. Identify the components like IC's present in the device.
3. Search the part-numbers/labels for the above mentioned components and download the datasheets if possible.
4. Understand how all the components in the device interact with each other.

Using Multimeter, JTAGulator or other tools, identify the debug ports by cross-verifying the pins/traces with the datasheets available.

III. HARDWARE COMPONENTS

Some of the important components that are required to be examined while dealing with the devices is explained in detail in the following subsections.

1. FLASH

FLASH, EPROM, EEPROM falls under the category of Non-Volatile Memory. Microcontrollers additionally have their own interior memory, which is regularly used to store code. These recollections are generally available while debugging a microcontroller for instance investigating through JTAG. There are many techniques to read FLASH. (Refer Figure 2)

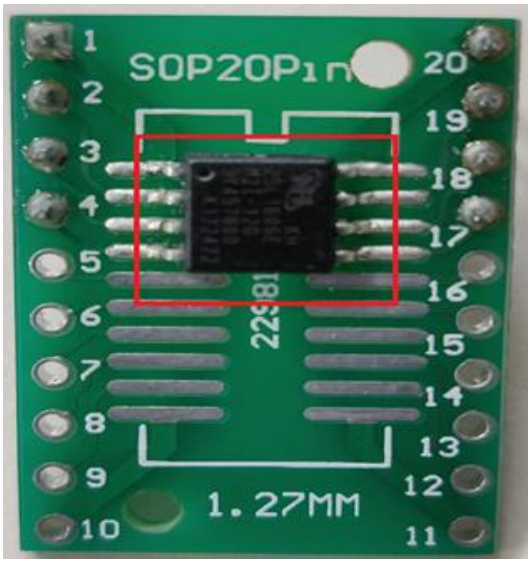

Figure 2: FLASH

2. Joint Test Action Group (JTAG)/Serial Wire Debug (SWD)

Generally, Microcontrollers use the Joint Test Action Group (JTAG) for debugging. Serial Wire Debug (SWD) is a 2-stick (SWDIO/SWCLK) electrical option JTAG interface that has the equivalent JTAG convention to finish everything. SWD utilizes an ARM CPU standard bi-directional wire convention, characterized in the ARM Debug Interface v5. JTAG uses 4 signals (TMS, TCK, TDI, TDO) where as SWD uses 2 signals SWDIO, SWDCLK. SWD is an ARM explicit convention structured explicitly for micro debugging. JTAG is being used for numerous microcontroller/processor structures beside A, but most of the ARM microcontrollers are using SWD because it has some advantages in terms of speed and other aspects in ARM chips debugging. (Refer Figure 3)

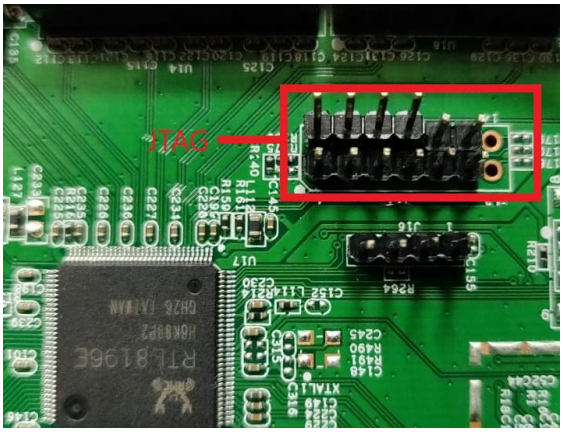

Figure 3: JTAG

2. Universal Asynchronous Receiver Transmitter (UART)

Universal Asynchronous Receiver Transmitter (UART) is one of the most common protocols in the embedded devices. It can be used as a debug I/O. It is a serial Protocol which is used as a direct connection between two devices. A UART's primary intention is to transmit and get sequential information. The name "serial" originates from the way that a serial port "serializes" information. That is, it takes a byte of information and transmits the 8 bits in the byte each one in turn. The preferred standpoint is that a serial port needs just a single wire to transmit the 8 bits while a parallel port needs 8. Standard pin out of UART is V, GND, TX, and RX as shown in the Figure 4

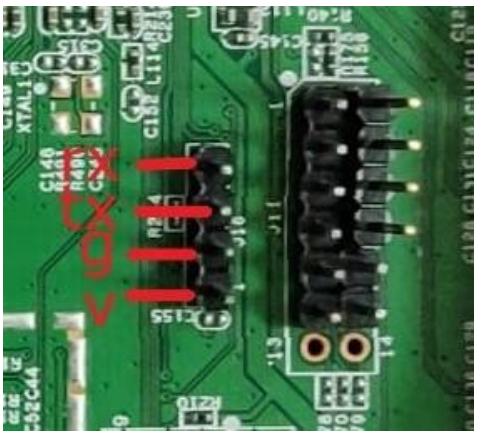

Figure 4: UART

4. Universal Serial Bus (USB)

Universal Serial Bus (USB) is generally used for charging the devices, communication between the devices like data transfer etc., it can also be used for debugging. Generally, the USB port in the devices should be taken care in such a way that attacker should not be able to communicate and get sensitive data by using compatible drivers or software. (Refer Figure 5).

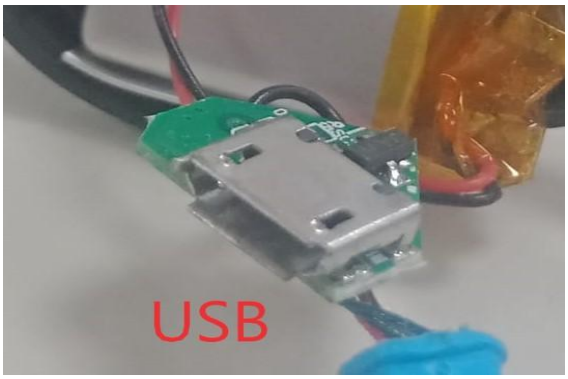

Figure 5: USB

IV. VULNERABILTIES ON IOT DEVICES

After analyzing the vulnerabilities of different IoT devices, in this paper we are trying to address the exploitation and mitigation strategies

A. Firmware Extraction via External Flash Memory Dump

It is possible to desolder the flash memory IC present on the circuit board to read its

content using a flash programmer tool. This allows the attacker to dump the existing firmware [9][10] as well as update it. The attacker will need physical access to the device in order to exploit the vulnerability. This existing firmware is used for further analysis as well as reverse-engineering [11]. The attacker can be able to craft the custom firmware and update the flash memory directly by bypassing into any software level checks if any, during normal firmware update process [12]. The steps to reproduce are listed below:

A. Desolder the Flash IC from circuit board
B. Place it into a flash programmer tool like TNM5000 and read its content

The Supportive Evidence is shown in Figure 6

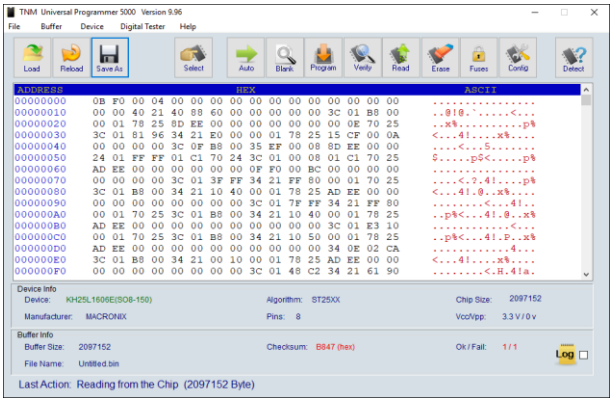

Figure 6: Firmware dumped from flash IC using TNM Programmer

Recommendation: It is recommended to have a secure boot implementation which verifies the integrity and authenticity of the firmware before execution. It also encrypts the firmware present in the external flash IC on the circuit board to prevent the attacker from dumping and reverse-engineering it.

B. Unencrypted Firmware Package

Most of the devices allow the users to update the firmware via web interface. In such cases, the firmware package has to be encrypted. In some devices, the firmware format does not implement the encryption. Since the firmware package itself is not encrypted, it can be easily extracted. This will allow the attacker to analyze and study the binaries for vulnerabilities so it makes it easier to create exploits.

The attacker have to get a copy of the firmware update package itself to exploit this issue. It is trivial to extract its content for further analysis using easily available software tools. Since, the firmware update package format is not encrypted, it allows an attacker to extract contents, reverse-engineer them and this makes it easier to find vulnerabilities [13]. The Supportive Evidence is shown in Figure 7,8.

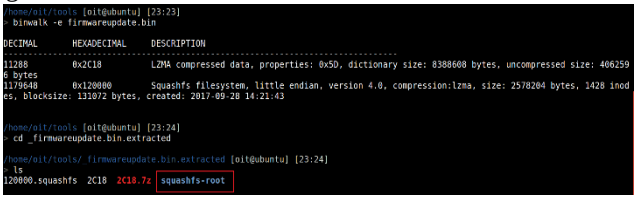

Figure 7: Extraction of Sqaush File System

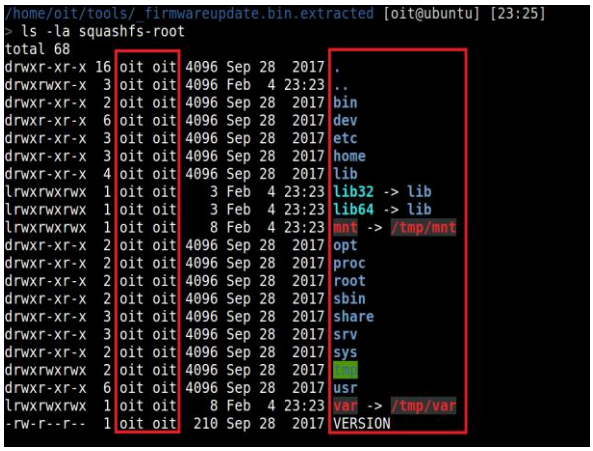

Figure 8: Content of the device's "squashfs" file-system that is extracted from the firmware

Recommendation: It is recommended to encrypt the firmware update package as well as implement the functionality to verify the integrity and authenticity of the package before initiating the upgrade process.

C. Root Shell accessible via UART interface without authentication

In different devices, UART port present on the circuit board is active which prints out the boot time logs and eventually drops into a shell which has "root" privileges and no authentication is required for accessing this shell. The attacker will need physical access to the device in order to exploit the vulnerability.

Once an attacker has “root” access, he/she has complete control over the device. The attacker can then alter the system, add a backdoor, install a malware or do anything they wish. This also allows them the possibility to fuzz the software present on the device and find more vulnerabilities

The steps to reproduce are listed below:

A. Connect to the UART port present on the device circuit board with an FTDI based USB-to-Serial converter
B. Select the correct baud rate settings

The Supportive Evidence is shown in Figure 10. In this case, the account named "admin" has 'root' level privileges.

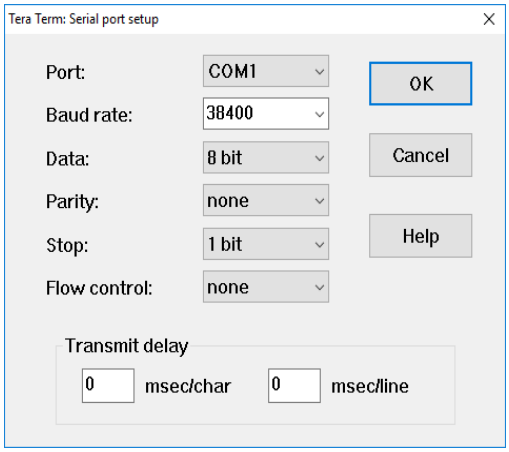

Figure 9: Tera Team Serial Port Setup

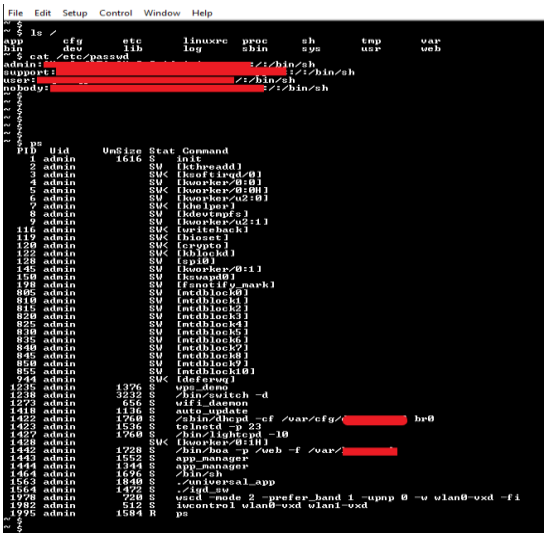

Figure 10: "Root" shell access on the device when connected via UART port

Recommendation: It is recommended to disable UART interface on devices when in production

D. Verbose Server Banner

In several devices, it was observed that the Verbose server information is sent in HTTP responses from the server. This information is commonly included in the server response headers and can disclose some information like server name, type, and version number. By knowing version, the type of web server and how each web server responds to specific commands and by keeping this information in a web server fingerprint database, an attacker can send these commands to the web server, analyze the response, and compare it to the database of known signatures.

By knowing the information about the server, an attacker can plan attacks in future, with the information obtained. There may be publicly known exploits and vulnerabilities associated with the server hosted, whose information gets disclosed in the banner. Verbose server banners provide additional information that allows an attacker to perform targeted attacks to the specific technology stack in use by the application and underlying infrastructure.

The steps to reproduce are listed below:

A. Using Proxy tool like burp-suite, intercept the communication.

B. In http Response, Verify the header called as server

The Supportive Evidence is shown in Figure 11

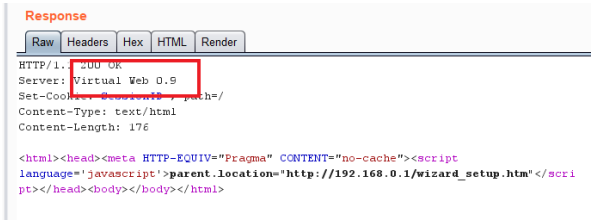

Figure 11: Web Server Disclosing version details of the software used

Recommendation Verbose server information should be removed from all HTTP responses. This can be performed by modifying the server's configuration files or through the use and configuration of a web application firewall. It is recommended to use generic error message response from server, so that server banner is disclosed in the error message response from the server

E. Use of Default Credentials

While testing the devices like routers, It is found that devices like routers has predefined default credentials i.e., username: "admin" \& password: "" for both web-interface as well as telnet service, which is also printed at the bottom of the device. The credentials are very common \& hence very easy to guess if the user leaves it unchanged. It is also interesting to note that once the device is reset, the user-set credentials are erased and the device falls back to the default one

Since the default credentials are very common and easy to guess, it is easy for an attacker to exploit this vulnerability. Since the same default credentials will work on all routers of this model, an attacker can easily automate the attack process by searching the internet/network for this device fingerprint and by trying the default credentials for authentication. Admin account has the highest privileges and can be used to modify device settings, update firmware, install a backdoor etc. The Supportive Evidence is shown in Figure 12

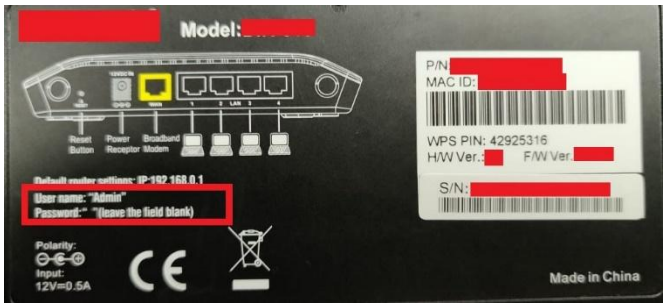

Figure 12: Use of Default Credentials

Recommendation: It is recommended to generate the

credentials using secure random number generation algorithms for each individual device instead of having predefined default values.

V. CONCLUSION

The IoT can possibly drastically increment the accessibility of data, and is probably going to change the organizations and associations in for all intents and purposes each industry around the globe. The primary highlights that separate IoT security issues from the conventional ones are the heterogeneous and largescale articles and systems. This article addressed few vulnerabilities on IoT devices and an approach to reproduce those attacks and recommendations for them.

REFERENCES

[1] Mannilthodi, Nayana, and Jinesh M. Kannimoola. "Secure IoT: An Improbable Reality." IoTBDS. 2017.

[2] Mohan, Lakshmi, et al. "Implementation of Scatternet in an Intelligent IoT Gateway." Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2. Springer, Cham, 2015

[3] Zhang, Zhi-Kai, et al. "IoT security: ongoing challenges and research opportunities." 2014 IEEE 7th international conference on service-oriented computing and applications. IEEE, 2014

[4] Ju, Hongil, Yongsung Jeon, and Jeongnyeo Kim. "A study on the hardware-based security solutions for smart devices." 2015 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2015.

[5] Suo, Hui, et al. "Security in the internet of things: a review." 2012 international conference on computer science and electronics engineering. Vol. 3. IEEE, 2012.

[6] Nawir, Mukrimah, et al. "Internet of Things (IoT): Taxonomy of security attacks." 2016 3rd International Conference on Electronic Design (ICED). IEEE, 2016

[7] Jisha, R. C., Aiswarya Jyothindranath, and L. Sajitha Kumary. "Iot based school bus tracking and arrival time prediction." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017.

[8] Rahman, Fahim, et al. "Hardware-assisted cybersecurity for iot devices." 2017 18th International Workshop on Microprocessor and SOC Test and Verification (MTV). IEEE, 2017

[9] Xie, Wei, et al. "Vulnerability Detection in IoT Firmware: A Survey." 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2017

[10] Stellios, Ioannis, et al. "A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services." IEEE Communications Surveys \& Tutorials 20.4 (2018): 3453-3495

[11] Shwartz, Omer, et al. "Reverse Engineering IoT Devices: Effective Techniques and Methods." IEEE Internet of Things Journal 5.6 (2018): 4965-4976.

[12] Kvarda, Lukáš, et al. "Software implementation of secure firmware update in IoT concept." (2017).

[13] Chandra, Hans, et al. "Internet of Things: Over-the-Air (OTA) firmware update in Lightweight mesh network protocol for smart urban development." 2016 22nd Asia-Pacific Conference on Communications (APCC). IEEE, 2016

Gottumukkala Eswar Kumar Raju obtained his M.Tech in the TIFAC-CORE in Cyber Security from Amrita Vishwa Vidyapeetham. Currently, he is working as a Security Engineer in Philips, Bengaluru

Dr. Lakshmy K. V. obtained her PhD (Cryptography) from Amrita Vishwa Vidyapeetham. Currently, she is working as an Assistant professor in the TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham University, Coimbatore.