IoT Security – Part 3 (101 – IoT Top Ten Vulnerabilities)

Admin-Payatu
19/01/2018

If you haven't read through Part 1 and Part 2 of our IoT Security Blog series I would urge you to go through them first unless you are already familiar with the basics and want to only read about IoT Top Ten Vulnerabilities.

When talking about Top Ten vulnerabilities, the first thing that comes to our mind is OWASP. Why not, after all they are the pioneers in defining top 10 vulnerabilities for web and mobile. I'm an OWASP fan, simply because of the work the OWASP community has done over the years to define Application security issues, provide free tutorials and open source tools for the Industry to mitigate the risks and vulnerabilities. It would be highly unlikely that you haven't heard of OWASP or read content from their website, however if you have not, I strongly suggest that you go through their website https://www.owasp.org

OWASP has also started the IoT security initiative where the community has defined the IoT attack surface and the IoT Top 10 vulnerabilities in addition to web and mobile. They are in the right direction and soon enough it will be an excellent place for IoT security content.

The content relevant to the reader for IoT security on OWASP website is as follows:
1. OWASP Web Top 10 project: –
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
2. OWASP Mobile Top 10 Project:
https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
3. OWASP Internet of things project:

https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project
a. OWASP IoT Attack Surface:
https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Attack_Surface_Ar
b. OWASP IoT Top Ten Vulnerabilities:
https://www.owasp.org/index.php/Top_10_IoT_Vulnerabilities_(2014)

OWASP Internet Of Things Top Ten Vulnerabilities

OWASP has recently defined the top 10 vulnerabilities in IoT. They are quite comprehensive and we would suggest that you go through them and understand what are the threats and issues with IoT ecosystem. As a homework you can map it to the attack surface we defined in the previous blog post. The OWASP IoT top ten vulnerabilities (as per https://www.owasp.org/index.php/Top_IoT_Vulnerabilities):

I1. Insecure Web Interface
I2. Insufficient Authentication/Authorization
I3. Insecure Network Services
I4. Lack of Transport Encryption/Integrity Verification
I5. Privacy Concerns
I6. Insecure Cloud Interface
I7. Insecure Mobile Interface
I8. Insufficient Security Configurability
I9. Insecure Software/Firmware
I10. Poor Physical Security

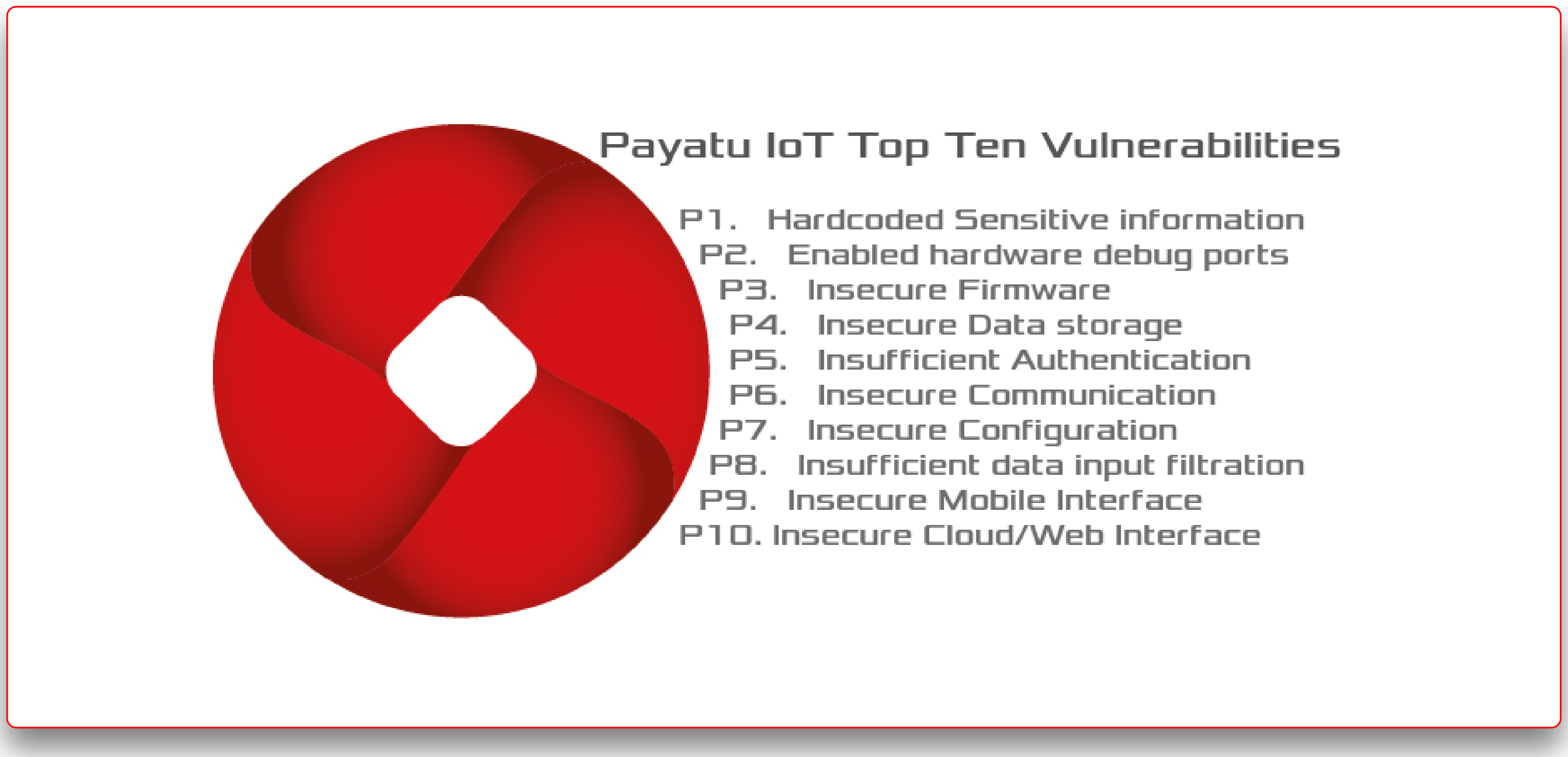
We will not go into the details of each item in the top ten list. The details can be found on the OWASP link (given above). Instead we are going to refine the top ten based on our experience of issues found by us or issues published on the Internet.

Payatu Internet of Things Top Ten Vulnerabilities 2018

Disclaimer: Please note, our aim is not to try to override OWASP Top Ten, the guys have done a great job. Kudos to the OWASP team! This is rather an exercise based on our experiences and focusing more towards the hardware and new IoT technology which deserves its due attention.

We will continue to maintain and update Payatu IoT Top 10 Vulnerabilities. If you have any suggestions, feel free to send us an email (info a..t payatu DOT com). We have combined web and cloud into one item, the reason being that not all sensors or IoT devices will have web interface and cloud is an important part of the ecosystem which is mostly web API based from an attack surface perspective. Also, some of the vulnerabilities may be applicable to more than one component, for example Hardcoding is applicable to both device and mobile app.

we will define the top 10 IoT vulnerabilities that have caused an impact on the IoT security market and product. We will explain all below IoT vulnerabilities to provide an understanding of the basic security issues.



Payatu IoT Top Ten Vulnerabilities

P1. Hardcoded Sensitive information
P2. Enabled hardware debug ports
P3. Insecure Firmware
P4. Insecure Data storage
P5. Insufficient Authentication
P6. Insecure Communication
P7. Insecure Configuration
P8. Insufficient data input filtration
P9. Insecure Mobile Interface
P10. Insecure Cloud/Web Interface

P1. Hardcoded Sensitive information
P2. Enabled hardware debug ports
P3. Insecure Firmware
P4. Insecure Data storage
P5. Insufficient Authentication
P6. Insecure Communication
P7. Insecure Configuration
P8. Insufficient data input filtration
P9. Insecure Mobile Interface
P10. Insecure Cloud/Web Interface

P1. Hardcoded Sensitive information

Hardcoding information during development is common practice as developers hardcode static data within a program. However, the problem occurs when sensitive information is hardcoded. It is very likely to have sensitive information hardcoded within the firmware as well as the mobile App or a thick client. The issue is that it remains the same for all the instances of the product and can be used for attacking any product instance deployed in the field. Some examples of sensitive information that is hardcoded:

1. Credentials – of device services, cloud services etc.
2. Encryption keys – Private keys, symmetric encryption keys
3. Certificates – client certificates etc.
4. API keys – Private/paid APIs
5. URLs – development, firmware related, user related, backend etc.
6. Configuration

P2. Enabled hardware debug ports

The device hardware may have debug ports open for interaction with the system. In simple terms, it is a set of pins-outs on the PCB which are connected to the micro-controller/microprocessor pins and you can use a client software to connect to these pin-outs to communicate over a hardware communication protocol that allows you to interact with the system. The level of interaction and privilege is dependent on the type of protocol and its usage. There may be pin-outs for UART interface, for example, which may give you access to high level software/application i.e. command shell, logger output etc. You can also get a low-level interaction with the micro-controller using protocols such as JTAG, SWD etc, these give you direct control over the micro-controller so you can test and analyze the microcontroller pin values, read/write the internal flash, read/write register values, debug the OS/base firmware code and much more. If these ports/pin-outs are enabled on the device an attacker can hijack the device and/or extract sensitive information from the device including the firmware and data. These ports are usually enabled for troubleshooting/debugging issues in production devices.

P3. Insecure Firmware

The term "insecure" here refers to the way firmware is managed and not specifically firmware code vulnerabilities themselves. Firmware contains the business logic of the device and is mostly proprietary i.e. IP (intellectual property) of the vendor. If an attacker has access to the plaintext firmware he/she can reverse engineer it to find security issues or to clone the logic and ultimately the product itself. The vulnerabilities depend on the way firmware is stored and updated on the device. If care is not taken to properly encrypt the firmware in storage or in motion (updates), the attackers can get hold of it. Some of the issues with firmware are (but not limited to):

1. Firmware is stored in plaintext on memory chips
2. Firmware is not signed and/or bootloader does not verify integrity of the firmware before loading
3. Firmware updates are transported in plaintext from the cloud or mobile to the device.
4. Firmware updates are transported over a plaintext communication protocol, for example, http.
5. Firmware encrypted with a single symmetric key for all the device instances.
6. Firmware encryption keys transferred along with the update to the device.

A properly implemented PKI based system can ensure optimum security, however most low power sensors lack the computation power to implement PKI effectively. Also, if updates are secure but the key can be extracted from the device using other vulnerabilities, then the whole exercise is futile.

P4. Insecure Data storage

This issue is prominent in devices as well as mobile apps. It is more apparent in device hardware probably due to the assumption that reversing hardware is difficult. Sensitive data, if not stored securely, can be extracted and utilized by an attacker to subvert the system. In addition to security issues, it may also have privacy implications if users' personal data is not protected properly. Some of the common issues:

1. Sensitive data stored in plaintext on memory chips
2. Sensitive data stored encrypted but encryption key is accessible
3. Custom encryption used to encrypt data
4. No access control for modifying data
5. Insecure data storage on mobile app (refer "P9. Insecure Mobile Interface")

P5. Insufficient Authentication

The devices may use improper or no authentication mechanisms, which allow the attacker to bypass the authentication mechanism altogether, if it is implemented poorly and send unauthorized commands to the device. This is a serious problem for critical IoT devices as anyone on the network (TCP/IP or radio) can override the normal operations and control the device. Some of the authentication issues that occur on the devices are (but not limited to):

1. No client authentication
2. Authentication over plaintext communication channel
3. Improper encryption used for credentials
4. Predictable credentials
5. Default credentials

P6. Insecure Communication

The communication within the IoT ecosystem may be insecure if attackers are able to sniff, analyze, replay and extract sensitive information from the communication. The vulnerabilities may be due to using insecure communication protocols or protocol deficiencies themselves. To keep things simple vendors may choose to use insecure methods of communication. Since, IoT is a new and evolving technology, many IoT protocols do not define proper security mechanisms or vendors implement default insecure modes. The issues include (but not limited to):
1. Unencrypted communication while sharing sensitive information
2. Using custom encryption
3. Using custom/proprietary protocols
4. Improper encryption used
5. Using protocol default (weak) security mode
6. Using protocols with known issues
7. Replay issues

P7. Insecure Configuration

This issue occurs when the device configuration is insecure or if the device does not allow users to modify configuration parameters. This issue also occurs in mobile app and cloud configuration. To keep things simple or shipping the product fast, developers may opt to use simple but insecure configuration and/or disallow changes. Some apparent issues are (but not limited to):
1. Using default insecure configuration
2. Disallowing integrators and/or users from modifying the configuration
3. Insecure low-level protocol and hardware configuration in release products
4. Insecure encryption modes and settings
5. Low or no visibility on user personal data shared or stored

P8. Insufficient data input filtration

This is going to be a major issue going forward as more IoT protocols are implemented in the IoT ecosystem. The telemetry data coming from the device, for example, may be trusted by the cloud or an IoT gateway, leading to known and unknown security issues such as remote code execution, web based attacks like SQL injection, cross site scripting and many more. We expect this to move up in priority in the future. While mature implementations do filter data for traditional technologies, the same is yet to pick up for new IoT protocol implementations.

P9. Insecure Mobile Interface

As mobile technology is mature compared to sensor technology from security perspective, we have grouped all mobile security issues into one. This does not mean they have less priority as you can see some of the high priority vulnerabilities are applicable to mobile as well. However, owing to the maturity of the technology, it already has plethora of information on security issues and secure implementations. Being a fan of OWASP, we recommend starting from OWASP Mobile Top Ten vulnerabilities that will take care of majority of the security issues.

P10. Insecure Cloud/Web Interface

As discussed in "P9. Insecure Mobile Interface", the same applies to cloud and web. In case a device has a web interface, you may still be able to own the device via web attacks, however these security issues are already well defined and understood. Again, we would recommend starting from OWASP Web Top Ten vulnerabilities for understanding and mitigating web security issues and documents from Cloud security alliance for cloud security. Please note, this is not the only knowledge base available and one should look at tools and research papers available on the internet for the same. It is important to note that the cloud forms the data storage and communication backbone for the IoT ecosystem. If the cloud is compromised, it may lead to the compromise of the whole IoT ecosystem including all deployed products around the world and the Universe.

That's all for this blog post. Keep an eye on the next blog in the series. Feel free to get in touch with us if you have any suggestions.

Continue to the next part- IoT Security – Part 4 (Bluetooth Low Energy – 101)