Payatu Case Study

Enhancing the Security Posture for Portable Energy Solutions

Payatu Security Consulting Pvt. Ltd

Project Overview

Clean energy is the need of time, and the solar industry is here for it!

As the world moves towards increased demand for scalable and reliable sources of power, solar companies are amping up their technology to harness abundant sources of energy with optimum efficiency.

One of the largest developers and providers of solar and battery systems wanted to make its PES (Portable Energy System) more secure for its customers to ensure that they only get products of the highest quality.

This is when Payatu was asked to perform a device security assessment of the company's PES device, and the mobile application associated with it.

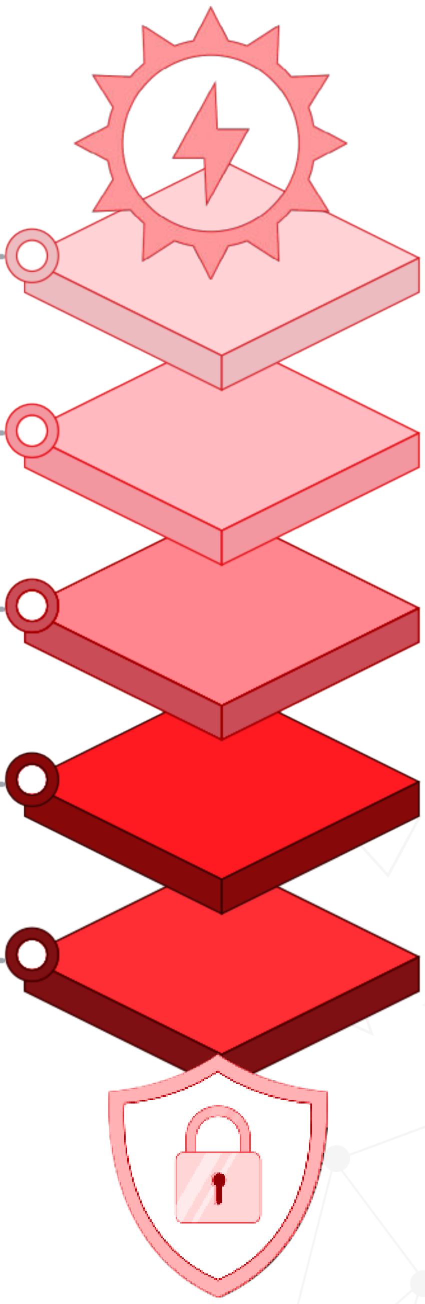
The goal was to identify all vulnerabilities present and report them to the client for timely mitigation in order to enhance the security posture of the device.

Let's take a look -

The Scope

The scope of this project spanned across the assessment of -

- **PES Firmware Analysis**
- **PES Hardware Pentesting**
- **MQTT Protocol Implementation**
- **PES Radio Pentesting (BLE and Wi-Fi)**
- **Android Application**



Process

Hardware Assessment:

The PES device is a bulky device (around 25-30 KG) that comes with a chargeable battery. Since it has a battery that works on a high voltage, doing hardware assessment directly on this device was risky, and thus the client decided to give the Bandits a stripped version of the device (without battery) which comes without any casing on the PCBs, for the hardware assessment.

This was a white box assessment where the client shared all the documents related to the device. Upon going through the documentation, the Bandits realized that at the hardware level, there are 3 separate PCBs working to perform the dedicated functionality i.e. BMU board (Battery Management Unit), SCU board (provides the main functionality of the board like user interface management) and the Gateway (responsible for the BLE and Wi-Fi communication).

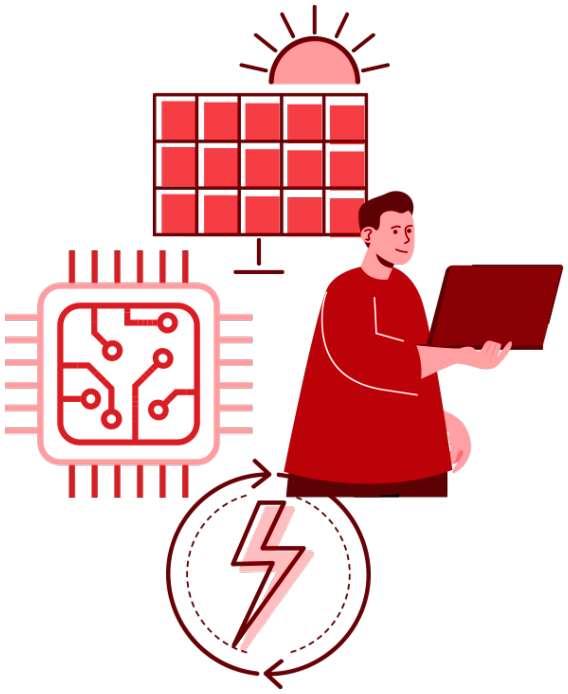
After going through the documentation, the team started working on the device. The process was -

- Recon: Gathering information about the ICs on the board, searching for the debug ports, usage of the external ports on the device.
- Scanning the debug ports: Checking if the debug ports (UART and JTAG) are enabled.
- Accessing the debug ports to capture the booting logs of the device and shell access from UART and extracting/patching the firmware from/to the microcontroller internal memory from JTAG.
- Firmware/data extraction and patching to the external flash memories used on the PCB (SCU and BMU board) of the device.
- Passing the extracted firmware to the firmware team for further analysis and reporting the hardware findings.

Firmware Analysis:

The firmware analysis was done statically where the firmware was analyzed not in the runtime environment. The static analysis aims to find the hardcoded sensitive information in the firmware binary like certificates, encryption keys or hardcoded credentials. It also includes the firmware binary reversing to analyze it for vulnerabilities like memory corruption.

Since this was a white box project, the client shared the firmware binary file of the BMU, SCU and the gateway for the analysis.



The Static Analysis flow was:

1. The Payatu team had the ELF files for both the gateway and BMU firmware. These files weren't stripped, so they could dive into analyzing them right away using the symbols provided.

2. On loading the binary into Ghidra to reverse engineer the device's functionalities, Ghidra offers disassembly, decompiled code, and access to the data within the ELF binary for analysis.

3 Figuring out where to start and what to look for in reverse engineering can be tricky. To tackle this, one should begin by finding the entry point. The team also used the 'strings' command in Linux to find useful strings. Understanding the basics of the ELF file format is essential for making sense of everything.

4 Once the correct flow is identified, the team reverse-engineered the functionalities and hunted for bugs, such as memory corruption or hardcoded sensitive information.

5 Since dynamic access was available, it was leveraged to validate and potentially exploit the bugs identified statically and finally report the firmware findings.

Dynamic firmware analysis is the case where the firmware is analyzed in the runtime environment. The PES device has an open JTAG port on the SCU board which gives us a window to do dynamic firmware analysis on the SCU board. The gateway doesn't have the JTAG interface which restricts the dynamic analysis on the gateway module.

The dynamic analysis flow was:

1. With SCU's SWD ports, the team could initiate dynamic analysis using JLink, OpenOCD, and GDB-Multiarch to delve deeper into the system's behavior.

2. Then, the bugs found statically were by carefully going through each instruction using gdb-multiarch.

Wireless Assessment:

The PES device communicates with user's mobile phone using the BLE and Wi-Fi wireless protocols, which is also under the scope of the assessment.

The module used for wireless communication is ESP32-WROVER-E which has both BLE and Wi-Fi inbuild. BLE version was 4.2 and Wi-Fi supports WPA3 encryption.

After analyzing the configurations, the team began working on the device. The flow then was:

- Recon: Gathering info about the ESP module, searching for the versions and encryptions, features included in user mobile application.
- Initially started with the BLE enumeration of services and characteris-tics.
- Explored for the known vulnerabilities registered for the chipsets (microcontrollers) and BLE versions and tried out their exploits to check if the device is vulnerable to those CVEs or not.
- Few enumerations and DoS attacks were succeeded.
- The findings were reported along with the CVSS scores.

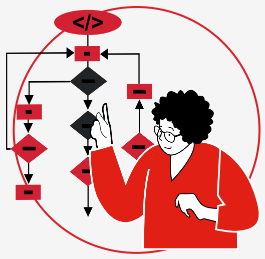
Android Application Security Assessment:

The PES devices can communicate with the mobile application using Wi-Fi. Commands can be sent to the PES device via the Android Application. The Android application offers a dashboard where users can query data from the PES device and manage features like power-saving mode, brightness settings, and monitor total power consumption.

The Android application also falls under the white box security assessment. The team was provided with the APK file for the assessment. Let's take a look at the analysis.

Static analysis involves inspecting the application's code and assets without running it to detect security weaknesses. It involves reviewing the application's source code or compiled files for issues such as hardcoded credentials, insecure API calls, and potential data leaks.

The Static Analysis Phase:



1. Reverse Engineer the APK file to reveal secrets such as hardcoded keys, credentials, sensitive API routes, misconfigured Firebase database, exported components, etc. This process helps identify potential security risks embedded within the application's code and configuration.

2. Assessing Client-side protection features such as root detection, SSL Pinning, emulator detection, Frida detection, code obfuscation, code integrity checks, etc is integral. These mechanisms are critical for preventing tampering and protecting the application from reverse engineering and unauthorized modifications.

3. The team also checked how the application handles data locally and was evaluated to identify risks such as data leakage in logcat and insecure storage practices, including unencrypted database files and shared preferences. These checks are vital for ensuring the proper security of sensitive data stored on the device.

4. Deeplink misconfigurations were checked, along with an analysis of the Network Security Configuration file to identify any potential misconfigurations.

Dynamic analysis involves observing the application's behavior during runtime to identify security vulnerabilities. It focuses on how the app interacts with the backend, detecting issues like access control weaknesses, injection flaws, and improper session management related to the APIs.

The Dynamic Analysis Phase:

1. The Dynamic Analysis begins with inspecting all the API calls sent from the mobile application to the server.

2. The login flow and password reset functionality were thoroughly tested to identify any vulnerabilities or flaws that could potentially be exploited, leading to account takeovers or unauthorized access to user accounts.

3. Issues related to horizontal privilege escalation and Insecure Direct Object References (IDORs) were tested to determine if unauthorized users could access or modify data belonging to other PES devices by manipulating input parameters.

4. The findings were documented and classified according to the CVSSv3 scoring system to assess their severity and impact.

Findings

💣 **Critical**

☠️ **High**

🚨 **Medium**

⚠️ **Low**

FIRMWARE:

- 🚨 Buffer Overflow in Various Functions
- 🚨 Hardcoded RootCA Certificate
- 🚨 Exposed RootCA Certificate
- 🚨 Exposed Operational Certificate
- 🚨 Exposed JWT Token
- ⚠️ Buffer Overflow in 'cell_task()' Function

HARDWARE:

- ☠️ Firmware Extraction and Flashing via SWD Port of the STM32 Controller
- ☠️ ESP32 QSPI Flash Chip Reading and Writing
- ☠️ EEPROM Reading and Writing via STM32 Controller UART Shell
- ☠️ Data Extraction and Patching to the EEPROM Chip
- 🚨 Hardcoded Certificate in the ESP32 UART Log

MQTT:

- 🚨 No Access Control Over Topics
- 🚨 Information Disclosure Using Multilevel Wildcard
- ⚠️ Deprecated TLSv1.0 and TLSv1.1 Protocol Detection
- ⚠️ Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability

RADIO:

- 💣 BLE-Injection: Overwrite URL
- ☠️ DoS Attack - Crash BLE Service
- ☠️ DoS Attack-Restart PES Device Remotely
- ☠️ DoS Attack-Display Error Message
- ☠️ DoS Attack - BLE De-authentication
- 🚨 BLE Read Characteristics-Web Server Banner Disclosure

ANDROID APPLICATION:

- 🚨 Improper Session Management
- 🚨 SSL Pinning Not Implemented
- 🚨 Unprotected API Endpoint Leads to Unauthorized Access of Device Details
- 🚨 Session Tokens Present in URL
- 🚨 Lack of Rate Limiting
- 🚨 Code Integrity Check Not Implemented
- ⚠️ Webview Debugging Enabled
- ⚠️ Application Vulnerable to Runtime Manipulation
- ⚠️ Excessive Permissions in Use
- ⚠️ Root Detection Not Implemented Properly
- ⚠️ Jailbreak Detection Not Implemented
- ⚠️ Web Server Banner Disclosure
- ⚠️ Insecure Webview Configuration
- ⚠️ The App Does Not Remove Sensitive Data from Views When Moved to the Background

Challenges

Access to the hardware was limited

The Bandits were restricted from assessing the internals of the device because of it working on high-voltage

No information provided on the internals of Bluetooth and Wi-Fi communication

Access control on MQTT protocols

Recommendations

- The length field value should be checked whether it is equal to or not greater than the size of the destination buffer.
- The RootCA cert should not be hardcoded in the firmware and should be generated dynamically at runtime.
- The RootCA should be protected or encrypted in the external flash memory.
- The Operational Certificate should be protected or encrypted in the external flash memory.
- The JWT token should be protected or encrypted in the external flash memory.
- The SWD port should be disabled.
- The data stored in the QSPI chip should be encrypted.
- The data stored in the EEPROM chip should be encrypted.
- The sensitive information like certificates should be stored encrypted.
- Have topics and clients grouped into different groups based on their privileges and usage category.
- Filter out the multilevel wildcards (#) and only use in case of a protected high privileged client on the broker (generally used for debugging or monitoring).

- Disable the deprecated TLSv1.0 and/or TLSv1.1 protocols in favor of the TLSv1.2+ protocols.
- Deploy (Ephemeral) Elliptic-Curve Diffie-Hellman (ECDHE) or use a 2048-bit or stronger Diffie-Hellman group.
- Either remove the WRITE access of 0x5c BLE handle or hide the value.
- Thoroughly check the Link Layer and L2CAP layer of the BLE stack.
- Check the BLE stack thoroughly and find out the function responsible for this behavior while the attack is performed.
- Hide or encrypt the data of BLE handles carrying sensitive information.
- Implement session management in such a way that the session is terminated on the server end as soon as the user logs off from the application.
- Implement SSL Pinning in the application.
- Enforce authentication on the affected API endpoint. Also, enforce authorization in a way that users only have access to details related to their PES devices.
- Ensure that sensitive information or session-related information is transmitted using the POST method instead of GET.
- Applications can be configured to utilize alternative mechanisms for transmitting session cookies, such as using HTTP cookies with security attributes or utilizing hidden form fields that are submitted using the POST method.

- Configure robust rate-limiting mechanisms across all APIs to restrict the number of requests a user or entity can make within a specific time frame.
- Implement code integrity checks in the application.
- Disable the webview debugging in the application by setting the "WebView.setWebContentsDebuggingEnabled" to false.
- Implement Frida detection check to mitigate run time hooking.
- Review all the permissions that are defined for the target application and remove the excessive permissions.
- Implement the root detection properly, the application should detect if it is installed on a rooted device. If yes, the user should not be able to use the application.
- Implement Jailbreak Detection in the target application.
- Disable the server banner within the configuration file of the web server.
- Set the SetAllowFileAccess, and SetAllowUniversalAccessFromFileURLs attributes to false.
- Use FLAG_SECURE to hide the screen in the background.

Before and After

Before the Security Assessment ✗	After the Security Assessment ✓
Lack of Physical Hardening or Physical security gaps or unprotected physical interfaces	Physically Hardened
Unsecure radio communication	Secure communication
Unsecure data storage	Secure data storage
Lack of secure firmware implementation	Firmware protection

About Payatu

Payatu is a Research-powered cybersecurity services and training company specialized in IoT, Embedded Web, Mobile, Cloud, & Infrastructure security assessments with a proven track record of securing software, hardware and infrastructure for customers across 20+ countries.

IoT Security Testing 🔗

IoT product security assessment is a complete security audit of embedded systems, network services, applications and firmware. Payatu uses its expertise in this domain to detect complex vulnerabilities & security loopholes to guard your IoT products against cyberattacks.

Mobile Security Testing 🔗

Detect complex vulnerabilities & security loopholes. Guard your mobile application and user's data against cyberattacks, by having Payatu test the security of your mobile application.

Product Security 🔗

Save time while still delivering a secure end-product with Payatu. Make sure that each component maintains a uniform level of security so that all the components "fit" together in your mega-product.

Web Security Testing 🔗

Internet attackers are everywhere. Sometimes they are evident. Many times, they are undetectable. Their motive is to attack web applications every day, stealing personal information and user data. With Payatu, you can spot complex vulnerabilities that are easy to miss and guard your website and user's data against cyberattacks.

Security Operations Center 🔗

Cyber threats are everywhere, often operating in the shadows. Their goal: to breach networks, compromise systems, and steal critical data. With Payatu's SOC service, you can uncover these hidden threats, bolster your defenses, and protect your data from relentless cyber attacks.

AI / ML Security Audit

Incorrectly implemented AI/ML systems can lead to security and privacy issues. The severity of which depends on how critical the use case is. The repercussions of the same include misclassification of unauthorized entities, theft of intellectual property such as application train models, etc. With a dedicated team capable of effectively assessing and strengthening AI/ML systems, we can provide specific methods to prevent potentially damaging threats before they potentially derail your project.

Cloud Security Assessment

As long as cloud servers live on, the need to protect them will not diminish. Both cloud providers and users have a shared. As long as cloud servers live on, the need to protect them will not diminish.
Both cloud providers and users have a shared responsibility to secure the information stored in their cloud Payatu's expertise in cloud protection helps you with the same. Its layered security review enables you to mitigate this by building scalable and secure applications & identifying potential vulnerabilities in your cloud environment.

Code Review

Payatu's Secure Code Review includes inspecting, scanning and evaluating source code for defects and weaknesses. It includes the best secure coding practices that apply security consideration and defend the software from attacks.

Red Team Assessment

Red Team Assessment is a goal-directed, multidimensional & malicious threat emulation. Payatu uses offensive tactics, techniques, and procedures to access an organization's crown jewels and test its readiness to detect and withstand a targeted attack.

DevSecOps Consulting

DevSecOps is DevOps done the right way. With security compromises and data breaches happening left, right & center, making security an integral part of the development workflow is more important than ever. With Payatu, you get an insight to security measures that can be taken in integration with the CI/CD pipeline to increase the visibility of security threats.

Critical Infrastructure Assessment

There are various security threats focusing on Critical Infrastructures like Oil and Gas, Chemical Plants, Pharmaceuticals, Electrical Grids, Manufacturing Plants, Transportation systems etc. and can significantly impact your production operations. With Payatu's OT security expertise you can get a thorough ICS Maturity, Risk and Compliance Assessment done to protect your critical infrastructure.

CTI

The area of expertise in the wide arena of cybersecurity that is focused on collecting and analyzing the existing and potential threats is known as Cyber Threat Intelligence or CTI. Clients can benefit from Payatu's CTI by getting – social media monitoring, repository monitoring, darkweb monitoring, mobile app monitoring, domain monitoring, and document sharing platform monitoring done for their brand.

More Services Offered

- Trainings

More Products Offered

- EXPLIoT

Payatu Security Consulting Pvt. Ltd.

www.payatu.com

info@payatu.com

+91 20 41207726