Keylabs, Inc.

Foundation Passport Audit

April 29, 2021

Executive Summary 2

Audit limitations 2

Findings 3

- Boot-counter can be defeated 3
- Firmware upgrades can be triggered without PIN approval 3
- Downgrade protection bypass 1: Logic bug 3
- Downgrade protection bypass 2: TOCTTOU 3
- Downgrade protection bypass 3: Failed upgrade attack 4
- Fault injection resistance: Delays 4
- Fault injection resistance: Boolean & Switch Logic 4
- Potential EM side channels 4
- Attacks on ATECC608A 4
- External flash AES-CTR encryption uses weak counter value 4
- External flash contents not signed 5
- Informational: Processor does not provide privilege separation 5
- Informational: Processor potentially vulnerable to fault-injection attacks 5
- Informational: No active tamper detection measures 5

Executive Summary

Keylabs was tasked with performing a security audit of a pre-production version of the Foundation Devices Passport wallet. The main focus of the review was ensuring the security of the boot, firmware verification and login process, ensuring that an unauthenticated attacker can not access stored funds.

The type of processor used in the device is known to be vulnerable to attacks that can allow i.e. a supply-chain attacker to modify the firmware. The threat-model of the device should be adjusted & communicated accordingly.

During the review multiple vulnerabilities in the firmware handling code were found, incl. unauthenticated downgrade attacks. In addition, it was found that certain parts of the code are relatively susceptible to fault-injection attacks, and guidance on improving the defenses were provided.

Foundation Devices quickly provided fixes for the identified issues, which were sanity-checked by Keylabs.

Audit limitations

This audit was conducted on pre-production soft- and hardware, and certain functionality was not yet implemented or could not be tested on the current state of the software. (For example firmware upgrades were not possible on the test devices.)

This document extends on the previous report on the configuration of the secure element and does not re-iterate the findings of it.

1.Findings

1.1. Boot-counter can be defeated

On boot, the firmware increases a monotonic counter on the ATECC. The result of this counter increase is not checked, and as such an on-bus attacker can prevent the counter from being increased. This can allow an attacker to hide boot attempts from the user of the wallet.

```
397     // Increment the boot counter
398     uint32_t counter_result;
399     se_add_counter(&counter_result, 1, 1);
```

1.2. Firmware upgrades can be triggered without PIN approval

The normal UX flow only permits upgrading the firmware after entering the PIN. The reviewed firmware version however checks the external SPI flash to determine whether a firmware upgrade is available. As this SPI flash could be modified by an attacker, it can allow unauthorized firmware upgrades.

1.3. Downgrade protection bypass 1: Logic bug

The reviewed firmware contains a logic bug that disables the downgrade protection:

```
if (verify_current_firmware(false))
{
    internal_fw_valid = true;
    if ((spihdr.signature.pubkey1 != FW_USER_KEY && internalhdr->signature.pubkey1 != FW_USER_KEY ) &&
        (spihdr.info.timestamp < internalhdr->info.timestamp))
    {
        goto out;
    }
    else
    {
        uint8_t *fwptr = (uint8_t *)internalhdr + FW_HEADER_SIZE;
        hash_fw(&internalhdr->info, fwptr, internalhdr->info.fwlength, internal_fw_hash, sizeof(internal_fw_hash));
        hash_board(internal_fw_hash, sizeof(internal_fw_hash), current_board_hash, sizeof(current_board_hash));
    }
}
```

Most of the verify_ functions return 1 on success, however verify_current_firmware returns 0 on success, and as such the downgrade-check will not be executed if the current firmware is valid.

1.4. Downgrade protection bypass 2: TOCTTOU

The firmware upgrade procedure is split into two parts:

- Check signature of firmware on external SPI flash

- Upgrade internal flash with contents of SPI flash

For both steps the firmware is newly read from the external SPI flash. An active attacker can switch-out the SPI flash contents between both steps and flash a different firmware than the one that was checked.

1.5. Downgrade protection bypass 3: Failed upgrade attack

An attacker can trigger the device into a failed-upgrade state, for example by removing power during a firmware upgrade. Once the internal firmware is in an invalid state, the downgrade-check will not be performed, allowing an attacker to downgrade the firmware of the device.

1.6. Fault injection resistance: Delays

To make fault-injection attacks on the firmware harder to perform the code contains randomized delays to make glitching specific instructions more difficult. However in some parts of the code the delay occurs immediately before controlling externally probable IOs, which provide a decent post-delay trigger. It is recommended to place the delay immediately in front of the security-relevant function calls/checks.

1.7. Fault injection resistance: Boolean & Switch Logic

The reviewed firmware contains multiple boolean-checks and switch-statements that are used for security-relevant checks. Branches on boolean logic and switch-statements without default branches are particularly vulnerable to fault-injection, as a single bit-flip can potentially invert the condition. It is recommended to compare against explicit bit-patterns.

1.8. Potential EM side channels

The reviewed, pre-production device has significant coil-whine and other emissions when performing certain actions. It should be ensured that this noise can not be used as a side-channel.

1.9. Attacks on ATECC608A

After the review, new attacks on the ATECC608A and its usage in the COLDCARD firmware were published. Foundation Devices confirmed that these issues were fixed in the latest Passport firmware.

1.10. External flash AES-CTR encryption uses weak counter value

The external flash contents are AES-CTR encrypted. While the key seems to be secret enough, the counter-value is potentially weak and potentially re-used to encrypt different plaintexts. This can lead to situations where an attacker with an old known-plaintext and ciphertext is able to decrypt newer data, or where an attacker can potentially correctly encrypt new data in a way that they will be accepted by the firmware.

1.11. External flash contents not signed

The external flash-contents are checksummed but not signed. It is recommended to sign the external flash contents.

1.12. Informational: Processor does not provide privilege separation

The Passport is based on the STM32H series processor, which is missing some of the advanced privilege separation features found on other STM32 processors (such as the “firewall” on the STM32L4). This feature allows creation of trusted areas that are separate from the main firmware). This can be used, for example, to separate the main firmware from accessing secrets such as the pairing secret directly.

This means that a potential information leak vulnerability (such as an out-of-bounds read) can leak the pairing key. However the reduced attack surface provided by the QR-code based interface lowers the risk of such attacks.

1.13. Informational: Processor potentially vulnerable to fault-injection attacks

The processor used on the Foundation Passport is part of the STM32 family. Research shows that on a lot of STM32 variants it is possible to by-pass the read-out protection using attack-methods such as fault-injection.

Given this the supply-chain security of the device is difficult to ensure, as the firmware might pass the authenticity check even if the device was for example backdoored by a supply-chain attacker.

1.14. Informational: No active tamper detection measures

The device does not have active tamper detection measures, at-rest physical attacks do not trigger a tamper response such as key-deletion.