

An abstract graphic consisting of several thin, black, overlapping lines that form a complex, geometric pattern. The lines intersect to create various shapes, including triangles and polygons, some of which are nested within others. The overall effect is a sense of depth and complexity, resembling a stylized map or a network diagram.

ATTACKING VEHICLE FLEET MANAGEMENT SYSTEMS

Ramiro Pareja Veredas / Yashin Mehaboobe

ABOUT US



Ramiro Pareja Veredas
Principal Security Analyst
IoActive



Yashin Mehaboobe
Security Consultant
Xebia

The Current Automotive Hacking Scene

Entrepreneur



Tesla Owners Beware: Your Car Could Get Hacked With a \$340 Device You Can Buy Online

Researcher Josep Pi Rodriguez published a white paper in August showing how two people could trick their way into Tesla Model Y with relatively accessible technology.

By [Gabrielle Bienasz](#)

September 15, 2022

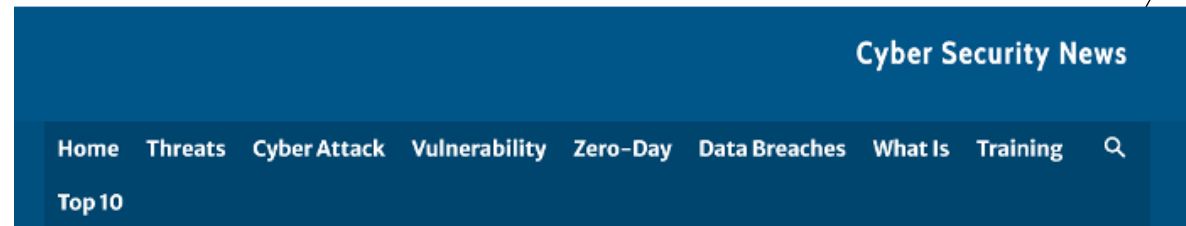


TECH

Honda key fob hack could leave all vehicle models since 2012 vulnerable: reports

By [Thomas Barrabi](#)

July 12, 2022 | 4:22pm | Updated



Home > Cyber Security News > Police Arrested Hackers Group Exploiting Keyless Technology to Steal Cars

Cyber Security News

Police Arrested Hackers Group Exploiting Keyless Technology to Steal Cars

By [Guru](#) - October 19, 2022

THE CURRENT AUTOMOTIVE HACKING SCENARIO

Our impressions:

Researchers ⇒ want to hack cars!
Expensive ones! (Are security researchers underpaid?)

Infotainments == usual attack vector

(Immobilizers / keyfobs == also common attack vector)

Vendors ⇒ main effort in securing infotainments

Consequence:

The security of many other automotive connected systems could have been neglected!



OUR RESEARCH

- Focus automotive embedded devices
- Permanently connected to Internet.
- No infotainments.
- Have potential to launch massive, scalable attacks.



MASSIVE, SCALABLE ATTACKS?

What do we mean?

- Remote attacks
- Can affect entire fleets
- Zero marginal effort/cost
- Could have a big impact



RESEARCH RESULTS

- Started in 2020 as a side project. Still working on it in our free time.
- >15 devices/systems tested, fully blackbox.
- Almost every device tested had high-impact vulnerabilities that could be exploited remotely.
- All the devices analysed are used for fleet control/management. Massive scale attacks are possible.
- No in-depth evaluations. Just enough to find “low hanging fruits”.
- None of the identified attack paths was very complex or elaborated. These are relatively simple hacks. Most of the devices were compromised in less than a week.

PREVIOUS PRESENTATIONS



ESCAR Europe 2022

ASRG Secure Our Streets 2023

Yashin Mehaboobe and Ramiro Pareja | Challenges in Automotive Vulnerability Disclosure

Yashin Mehaboobe is Presenting

DISCLOSING THE VULNERABILITIES

- VULNERABILITIES WERE DISCLOSED BEFORE PUBLISHING THE WHITE PAPER
- THREE EXAMPLES OF DISCLOSURE EXPERIENCES:
 - **THE GOOD:** MOVISTAR CAR ODB2 DONGLE
 - **THE UGLY:** QUECTEL RM500Q 5G MODEM
 - **THE BAD:** HOPECHART (SANY) HQT-401 TBOX

PREVIOUSLY PRESENTED: MOVISTAR CAR

Movistar Car is a OBD2 dongle that, for a monthly fee, provides:

- WIFI hotspot
- GPS tracker
- Anti-theft services
- Emergency call

Many major mobile provider around the world offers similar product. Many fleets are controlled with similar OBD2 dongles.

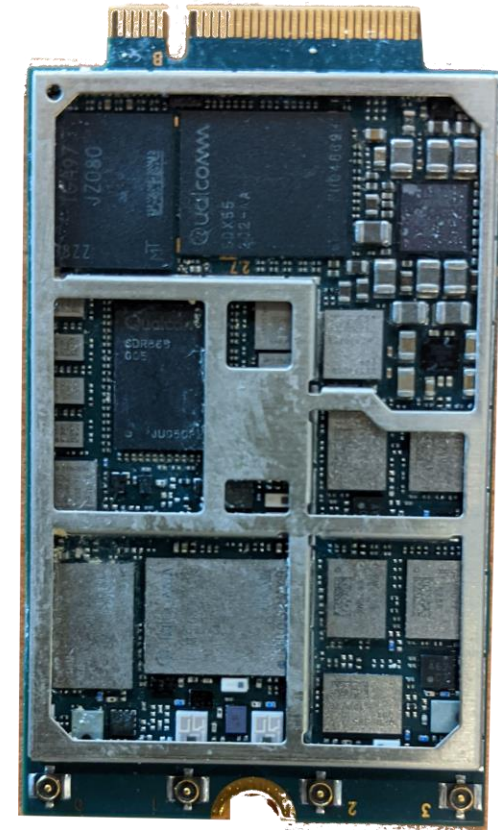
Vulnerabilities:

- Open debug ports
- Buffer overflow in web interface -> allows runtime control of the device
- Broken crypto for authentication -> allows impersonation of the server or the car.



PREVIOUSLY PRESENTED: QUECTEL RM500Q

- Quectel RM500Q is a 5G modem used for IoT and automotive.
- Vulnerabilities:
 - Command injection in the AT command parser -> allows runtime control
 - Insecure OTA communications -> MITM
- Remote exploitation is possible, but not easily scalable.



PREVIOUSLY PRESENTED: MQTT EXPOSED DEVICES

Multiple MQTT brokers expose automotive devices

- EV Cars:
 - Tesla
 - Nissan Leaf
 - Renault ZOE
 - VW ID4
- Aftermarket T-boxes:
 - OVMS (Open Vehicle Monitoring System)
- EV chargers:
 - OpenWB
 - Go-eCharger
 - openEVSE
- ODB2 dongles:
 - VW Connect

The problem are not the devices, but the misconfigured brokers!

More about MQTT hacking later



WHAT DO WE PRESENT TODAY?

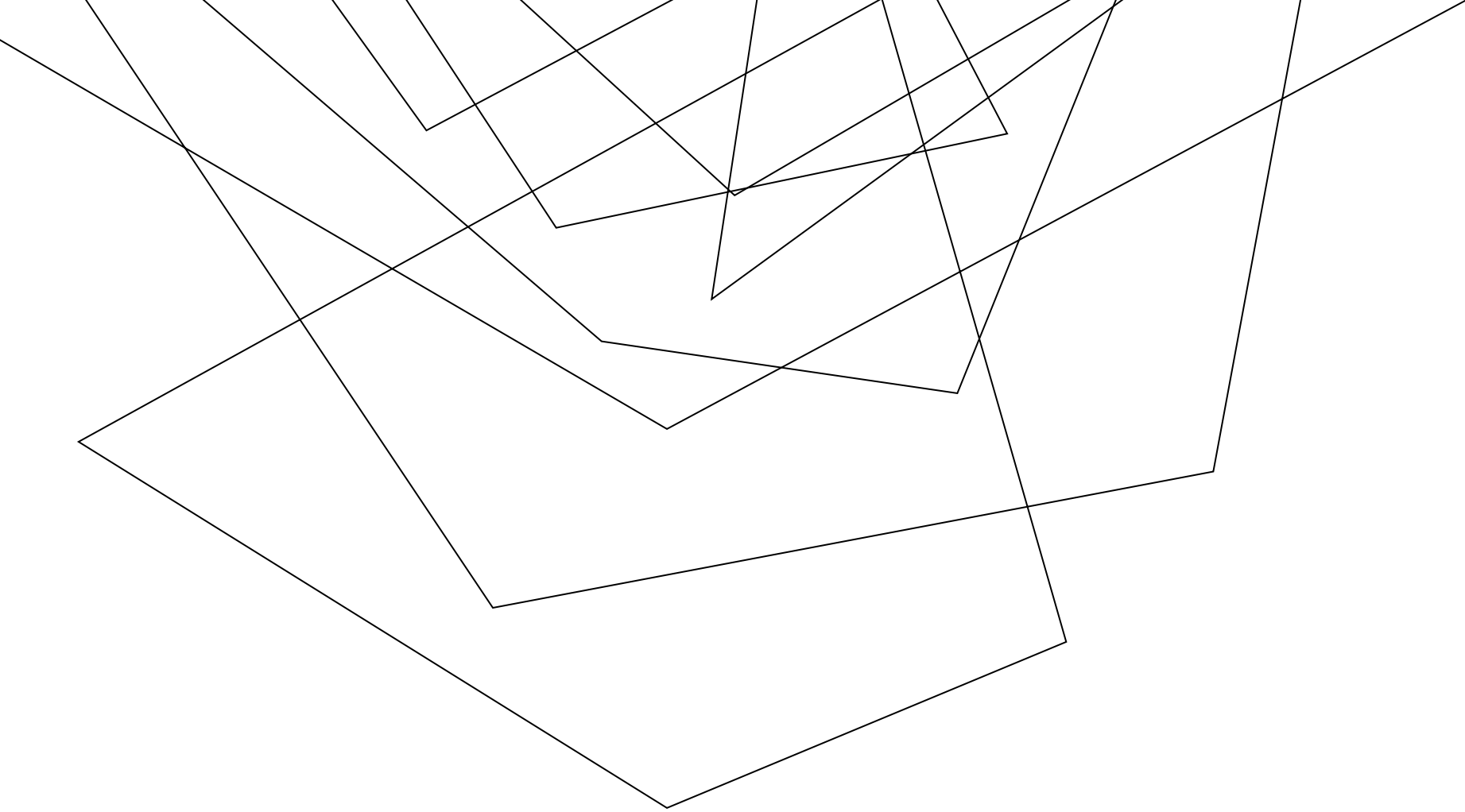
- Today, we focus in two Telematics boxes, used for fleet management.
- We chose them because they represent the **worst case scenario** possible in automotive security:
 - High number of affected vehicles (around 200,000)
 - Very high impact vulnerabilities (full control of the fleets!)
 - Low-to-middle effort to find the vulnerabilities, very low effort to exploit them.
 - Zero response from vendors

TELEMATICS BOX

- T-box == TCU
- Electronic controller designed to collect and transmit data from the vehicle to a backend.
- Used for:
 - V2X.
 - Vehicle tracking.
 - Remote diagnostics.
 - Fleet control.
 - Roadside assistance (e.g. eCall).
 - Etc.
- Most of the modern cars have it.



https://en.m.wikipedia.org/wiki/File:Electric_vans_owned_by_the_University_of_Warwick.jpg



HOPECHART HQT401

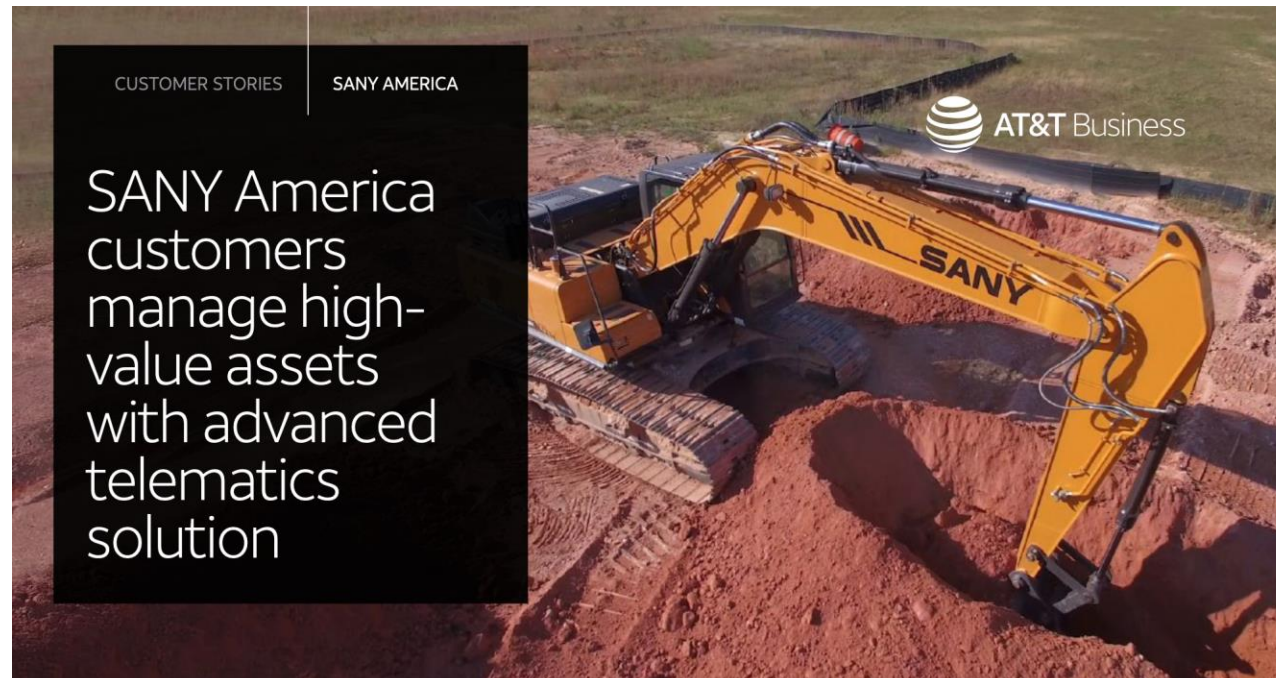
HOPECHART HQT401

- Android-based Tbox with WIFI, BT and 4G
- Used for fleet control:
 - Location
 - Diagnosis
 - Telemetry
 - Remote control (CAN sniffing and injection!)



WHERE IS THIS T-BOX USED?

- Factory installed by **at least** one vehicle manufacturer: Sany
- Sany is the 3rd-largest heavy equipment manufacturer in the world. 1st in excavators since 2020 (>100,000 units/year)



WHERE IS THIS T-BOX USED?

- Also installed after-market



[Home Page](#)

[Product Center](#)

[Solution](#)

[Support](#)

[About Us](#)

[News Center](#)



CN

Products hongquan

Reduce the cost of transportation
Intelligent network connection solution

灵狐

金雕

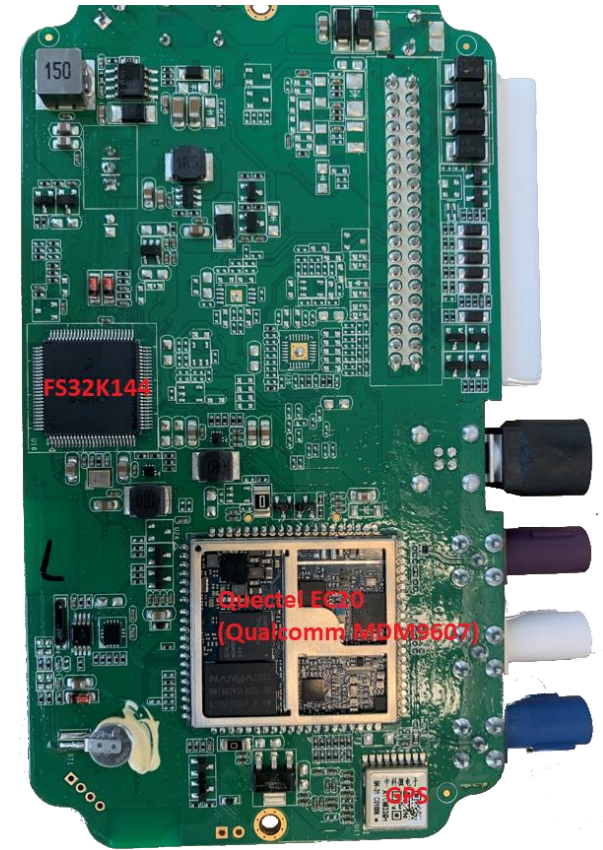
孔雀

海豚

斑马

INITIAL RECON AND IDENTIFICATION

- One T-box bought off Taobao
- Quectel EC20 (Qualcomm MDM 9607) based PCB
- PCB analysis reveals a connector that appears to be USB
 - Oscilloscope measurements point towards the same
- Soldered a USB cable and connected the host side to a PC
- We were able to get an ADB root shell and then dump firmware this way



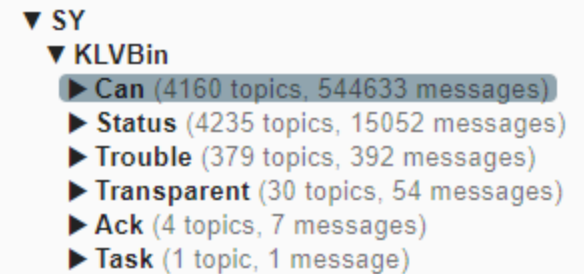
REVERSING FIRMWARE

- Binaries were not stripped or protected in any way
- We followed the usual RE process for Embedded Linux devices
 - Init scripts
 - Running processes
 - Config files
 - Network connections
- Some binary names caught immediately our eye:
 - MqttProxy
 - PlugMqttSanyCrane.so
- MQTT communications!

```
Decompile: Run - (PlugMqttSanyCrane.so)
1 |
2 /* plug_mqtt_sany_crane::TSanyCraneEngine::Run() */
3
4 void __thiscall plug_mqtt_sany_crane::TSanyCraneEngine::Run(TSanyCraneEngine *this)
5
6 {
7     undefined4 uVar1;
8
9     switch(*(undefined4 *) (this + 0x11bc)) {
10    case 0:
11        uVar1 = ModeInit(this);
12        *(undefined4 *) (this + 0x11bc) = uVar1;
13        return;
14    case 1:
15        uVar1 = ModeWork(this);
16        *(undefined4 *) (this + 0x11bc) = uVar1;
17        return;
18    case 2:
19        uVar1 = ModeDone(this);
20        *(undefined4 *) (this + 0x11bc) = uVar1;
21        return;
22    case 3:
23        uVar1 = ModeIdle(this);
24        *(undefined4 *) (this + 0x11bc) = uVar1;
25    }
26    return;
27 }
28
```

MQTT COMMUNICATIONS

- The device connects to an MQTT server for sending Telemetry data and for receiving commands
- We got the connection info from the config file.
- No authentication! No encryption!
- We can see the data from **all** the fleet vehicles.
- Vehicles are identified by the ICCID (kind of SIM serial number).
- We can send data impersonating any vehicle or the backend.



```
▼ SY
  ▼ KLVBin
    ► Can (4160 topics, 544633 messages)
    ► Status (4235 topics, 15052 messages)
    ► Trouble (379 topics, 392 messages)
    ► Transparent (30 topics, 54 messages)
    ► Ack (4 topics, 7 messages)
    ► Task (1 topic, 1 message)
```

WHAT CAN WE DO?

- Data is sent as binary messages, not plaintext.
- Using Ghidra, we reverse-engineered the communication protocol
- The following information is continually reported:
 - GPS position
 - Metrics (Speed, RPM, Gas tank levels, odometer, etc)
- The following information is reported under certain events:
 - Diagnostic errors
 - CAN traffic
- All this information can be **sniffed and spoofed** by an attacker.
- We made a dashboard to show all this information



DEMO



WHAT ELSE CAN WE DO?

- CAN injection:
 - Backend can send CAN packets and the T-BOX injects them into the CAN bus.
 - Used for advanced features like remote vehicle unlocking.
 - An attacker can spoof these backend messages and inject any CAN traffic
- Runtime control?
 - OTA firmware update triggered by a backend command
 - Firmware URL embedded in the command
 - An attacker can spoof the command to point a malicious firmware
 - No firmware verification mechanisms identified, but they could exist somewhere (bootloader?)
- We never tried these two attacks!



DISCLOSURE TIMELINES

- Q2 2021 – Vulnerabilities found
- Q3 2021 – First attempts to contact the vendor by email
- 02/2022 – Second attempt to contact the vendor by email
- 06/2022 – Contact the by phone.
We gave the message to marketing people, but never contacted back.
- 09/2022 – ASRG (Automotive Security Research Group) failed to contact the vendor.
- 11/2022 – ASRG contact in China managed to talk with Sany and Hopechart.
Sany confirms that at least 60,000 vehicles are affected.
- 06/2023 – Vulnerabilities patched according to the vendor.
CVE-2023-3028 assigned

DISCLOSURE TIMELINES

08/2023 – We find out that vulnerability is actually not fixed!

We try to contact Sany or Hopechart using all channels possible, including all the technical employees found in LinkedIn.

10/2023 – We gave up!

The vulnerability probably will not be fixed until somebody exploits in the wild.

POTENTIAL AFFECTED VENDORS



[Home Page](#) [Product Center](#) [Solution](#) [Support](#) [About Us](#) [News Center](#)



Partner

Do a good job of performance down-to-earth and create greater value for shareholders



A series of thin, black, overlapping lines forming various geometric shapes and polygons in the upper-left quadrant of the page. The lines are thin and black, creating a complex, abstract pattern.

MQTT

Vulnerable backend communications

MOTIVATION

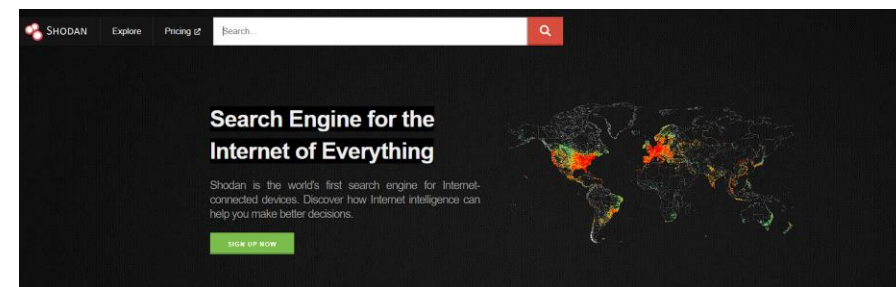
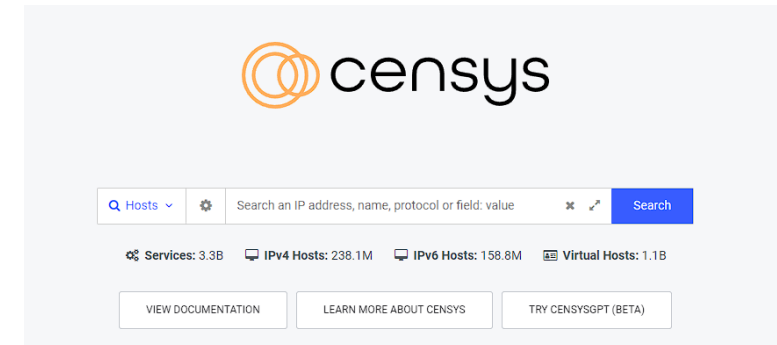
Hacking the T-Box via the MQTT server led to some interesting questions:

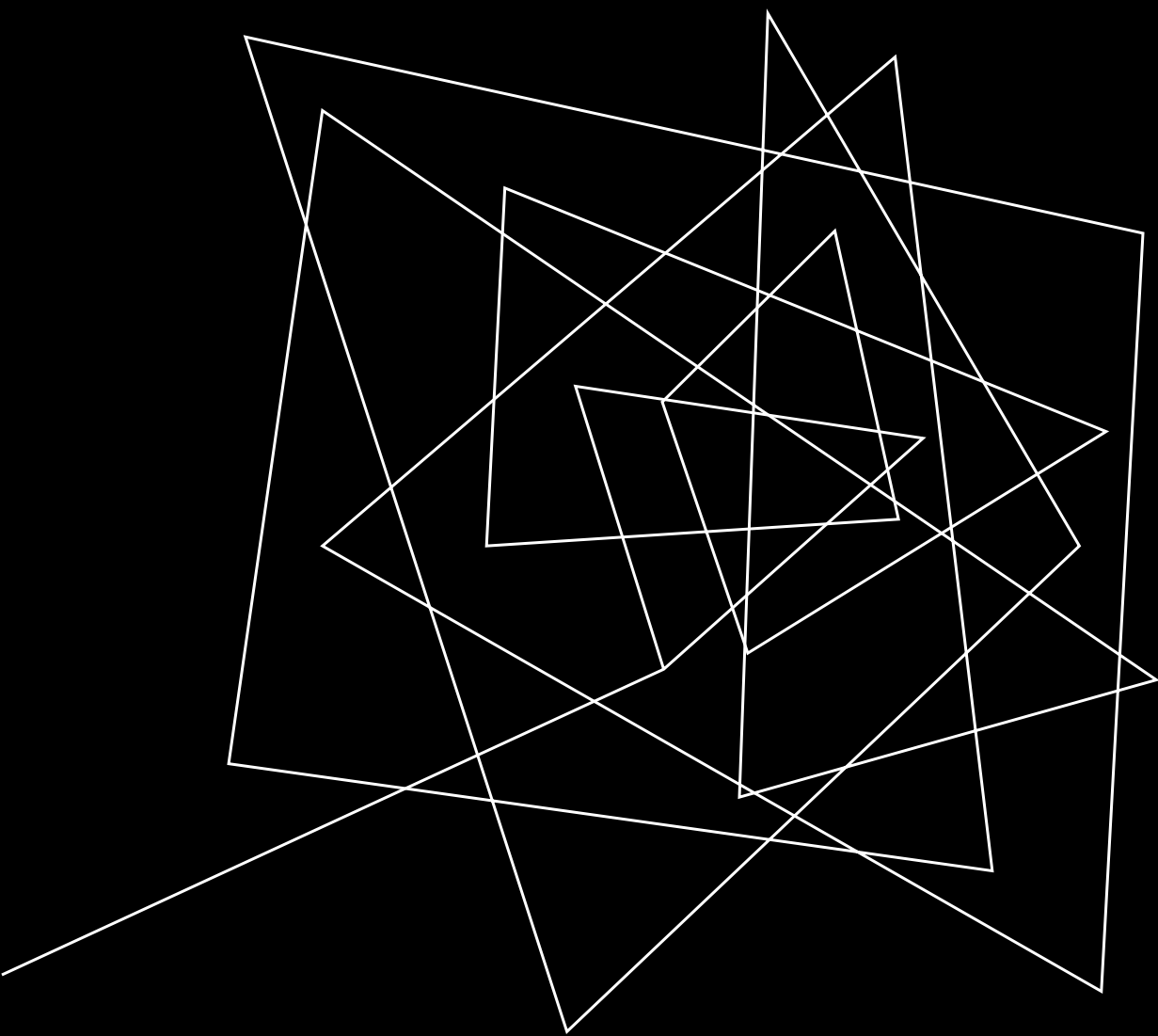
- Are there more misconfigured services like this out there?
- Can we find those services **without knowing** their existence?
- If so, could those services be hacked massively?



OSINT PLAYBOOK

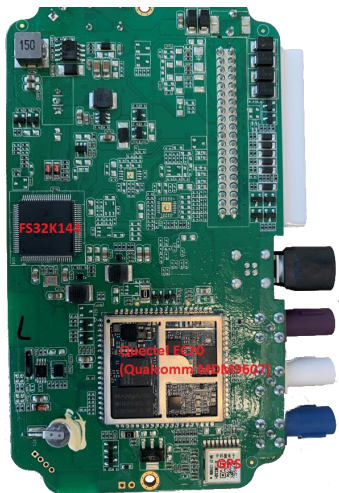
- Started with Shodan and Censys searches
 - Google but for devices
- Censys allows you to search for only MQTT open servers for example
- Narrowed it down to specific automotive terms
- Targeted things like MQTT, Kafka, RabbitMQ etc



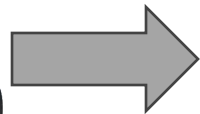


CAN YOU DO
THAT IN
REVERSE?

REVERSE ENGINEERING FIRMWARE

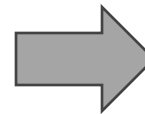


Hardware



```
Decompile: Run - (PlugMqttSanyCrane.so)
1
2 /* plug_mqtt_sany_crane::TSanyCraneEngine::Run() */
3
4 void __thiscall plug_mqtt_sany_crane::TSanyCraneEngine::Run(TSanyCraneEngine *this)
5
6 {
7     undefined4 uVar1;
8
9     switch(*(undefined4 *) (this + 0x11bc)) {
10    case 0:
11        uVar1 = ModeInit(this);
12        *(undefined4 *) (this + 0x11bc) = uVar1;
13        return;
14    case 1:
15        uVar1 = ModeWork(this);
16        *(undefined4 *) (this + 0x11bc) = uVar1;
17        return;
18    case 2:
19        uVar1 = ModeDone(this);
20        *(undefined4 *) (this + 0x11bc) = uVar1;
21        return;
22    case 3:
23        uVar1 = ModeIdle(this);
24        *(undefined4 *) (this + 0x11bc) = uVar1;
25    }
26    return;
27 }
28
```

Firmware

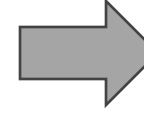


▼ SY

▼ KLVBin

- ▶ Can (4160 topics, 544633 messages)
- ▶ Status (4235 topics, 15052 messages)
- ▶ Trouble (379 topics, 392 messages)
- ▶ Transparent (30 topics, 54 messages)
- ▶ Ack (4 topics, 7 messages)
- ▶ Task (1 topic, 1 message)

MQTT

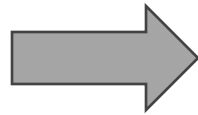


Exploit

REVERSE ENGINEERING PROTOCOL

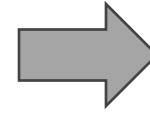
- ▼ SY
 - ▼ KLVBin
 - ▶ Can (4160 topics, 544633 messages)
 - ▶ Status (4235 topics, 15052 messages)
 - ▶ Trouble (379 topics, 392 messages)
 - ▶ Transparent (30 topics, 54 messages)
 - ▶ Ack (4 topics, 7 messages)
 - ▶ Task (1 topic, 1 message)

MQTT



```
Decompile: Run - (PlugMqttSanyCrane.so)
1
2 /* plug_mqtt_sany_crane::TSanyCraneEngine::Run() */
3
4 void __thiscall plug_mqtt_sany_crane::TSanyCraneEngine::Run(TSanyCraneEngine *this)
5
6 {
7     undefined4 uVar1;
8
9     switch(*(undefined4 *) (this + 0x11bc)) {
10    case 0:
11        uVar1 = ModeInit(this);
12        *(undefined4 *) (this + 0x11bc) = uVar1;
13        return;
14    case 1:
15        uVar1 = ModeWork(this);
16        *(undefined4 *) (this + 0x11bc) = uVar1;
17        return;
18    case 2:
19        uVar1 = ModeDone(this);
20        *(undefined4 *) (this + 0x11bc) = uVar1;
21        return;
22    case 3:
23        uVar1 = ModeIdle(this);
24        *(undefined4 *) (this + 0x11bc) = uVar1;
25    }
26    return;
27 }
28
```

Reverse engineering the protocol



Exploit

Are smart homes vulnerable to hacking?



MARTIN HRON | 16 AUG 2018



f
t
in
Avast expert Martin Hron tells you what you need to know about the strengths and weaknesses of IoT security and the MQTT protocol that connects and controls them.

PRIOR RESEARCH

Conferences > 2018 International Conference...

Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs

Publisher: IEEE

Cite This

PDF

M S Harsha ; B M Bhavani ; K.R. Kundhavai All Authors

20
Cites in
Papers

1556
Full
Text Views

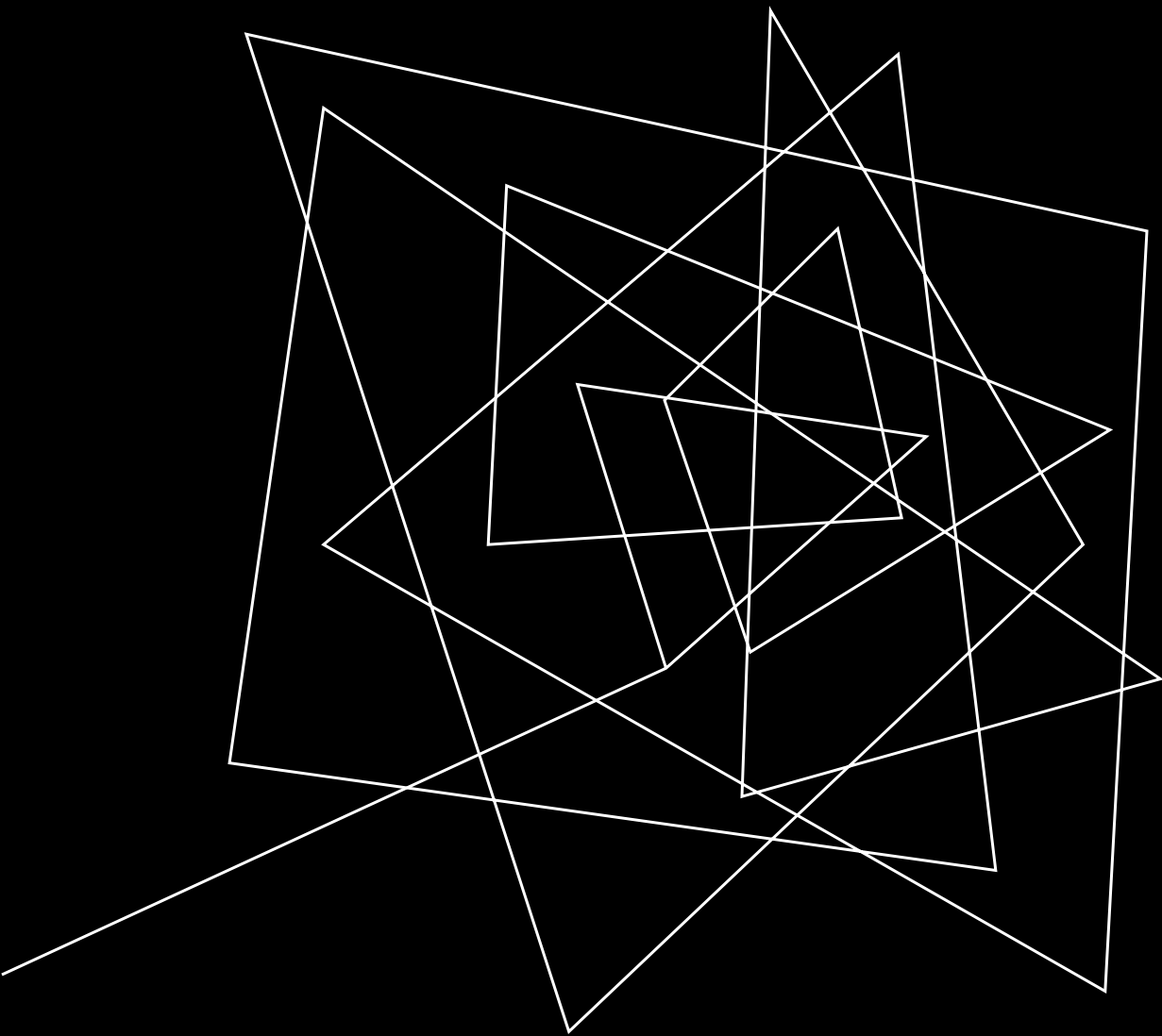


These WiFi garage doors have a major cyber vulnerability

Despite being alerted to these issues, the company has made no attempt to fix things.

BY HARRY GUINNESS | PUBLISHED APR 5, 2023 3:00 PM EDT

TECHNOLOGY



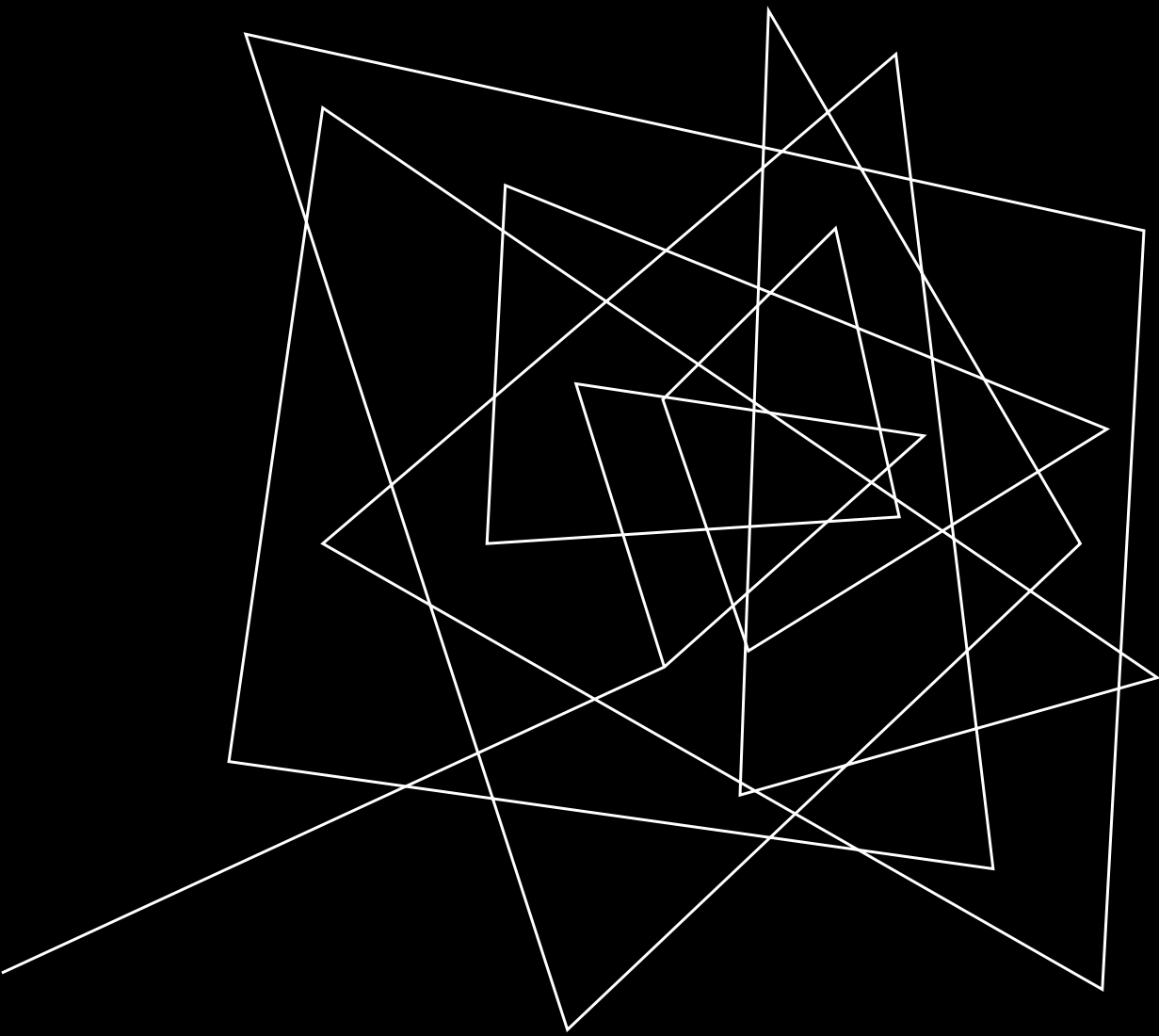
SPOILED FOR
CHOICE

OPEN MQTT SERVERS ARE PLENTY

- There are a lot of unsecured things on the Internet
- We found things we weren't looking for
 - Oil rigs, ship data, license plate readers and more
- Disclosure is still ongoing for all of them
- But we did find some more interesting than others

GENERAL MQTT RECON

- Connect using tooling such as MQTT-Explorer
- Create a dumping script using paho-mqtt
- Use tools like sort, wc and uniq for basic statistics
- Try running it overnight and sometimes on different parts of the day
- Is it returning a lot of data?
- Is the data changing?
- QoS and Retain values



NOTE ABOUT
~~REDACTIONS~~

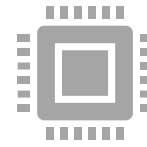
STANDING OUT AMONG TARGETS



**Unauthenticated MQTT
server**



**Seems to contain large
amounts of data**



**Many different diagnostic
data:**

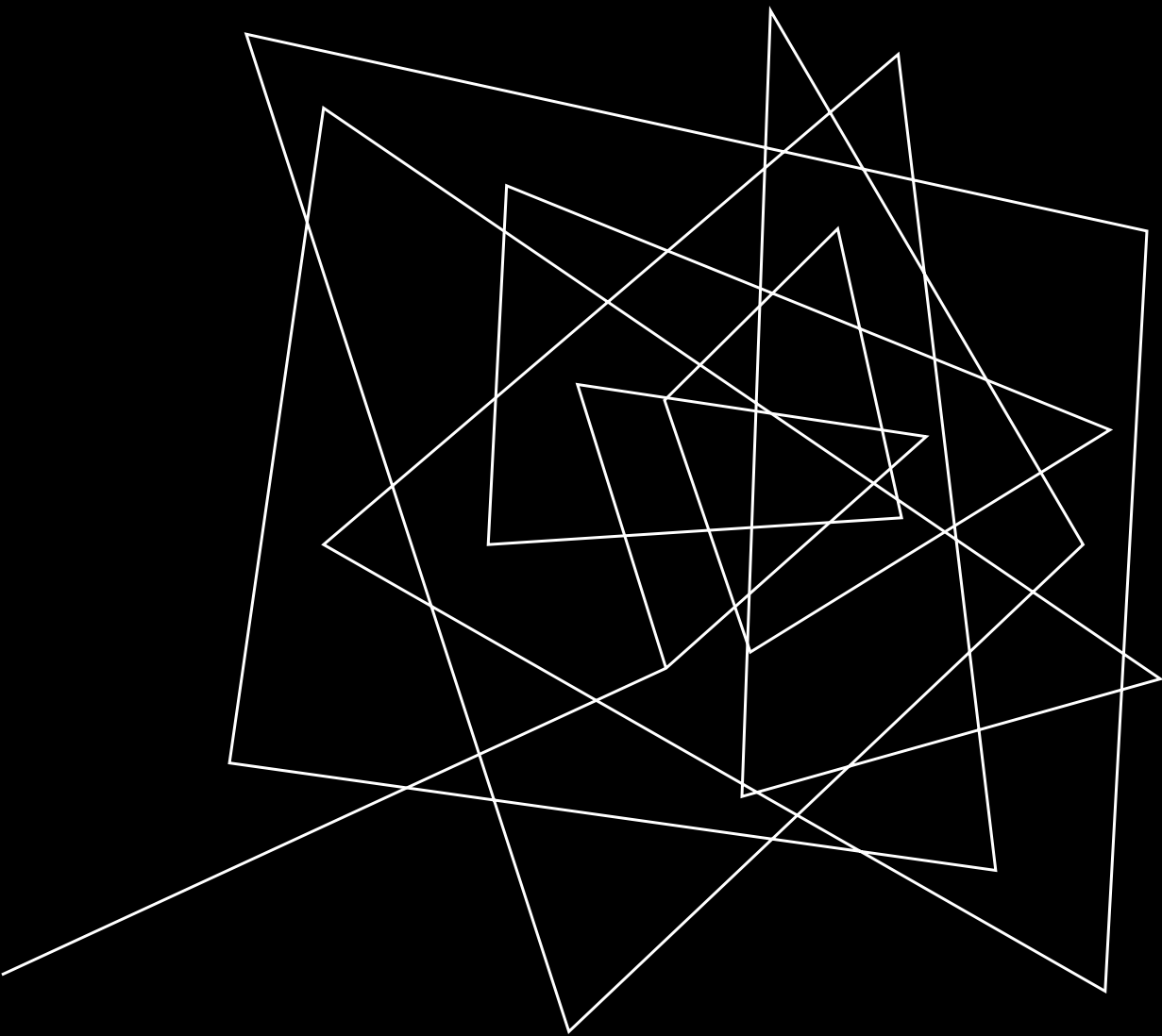
Videos being played?

System information such as CPU,
storage and RAM usage

Some sort of request and response
scheme?



**Keep in mind that we had no
idea who this belonged to
and what it did at the time**




COMPANY F


F stands for Security


WHAT DOES THE DATA LOOK LIKE?

```
1 chyron4/5167062441077/telemetry : b'{"connSt":{"online":false,"type":"dirty"},"epoch":null}'
2 chyron4/7698041094080/telemetry : b'{"connSt":{"online":false,"type":"dirty"},"epoch":null}'
3 chyron4/7698041094080/telemetry : b'{"connSt":{"online":true,"type":"clean"},"epoch":1698530360411}'
4 chyron4/5167062431979/telemetry : b'{"system":{"ramA":397972,"cpuS":[3,7382,110,127,604],"uptime":8229,"loadAvg":[1.06,1.13,1.17],"batt":4.097,"sessionCount":1437},"gps":{"fix":3,"position":[-12.024756,-77.1285],"speed":38},"mobile":{"state":"ON","gsm":1,"gprs":1,"rat":"FDD_LTE","mcc_mnc":"71606","operator":"#ConTodo_movistar","band":"LTE_BAND_4","rssi":31,"ip":"100.102.126.115","simId":"8951064022129512359F"},"netLink":{"name":"ppp0","ip":"100.102.126.115"},"message":{"level":"info","app":"systemd","mess":{"type:clips:CLIP_READY:1698530355-fatigue_alarm-mdsm7.mp4"},"npex_st":{"peripherals":{"mdsm7":{"ip":"192.168.2.12","port":49500,"connected":false,"connection":{"state":"CONNECTED","since":1698529413}}},"epoch":1698529413},"epoch":1698530359760,"full_state_epoch":1698529413}'
```


MAKING SENSE OF THE DATA

 Everything seemed to be under the “Chyron” topic

 The “Chyron” topic had subtopics which were numeric values

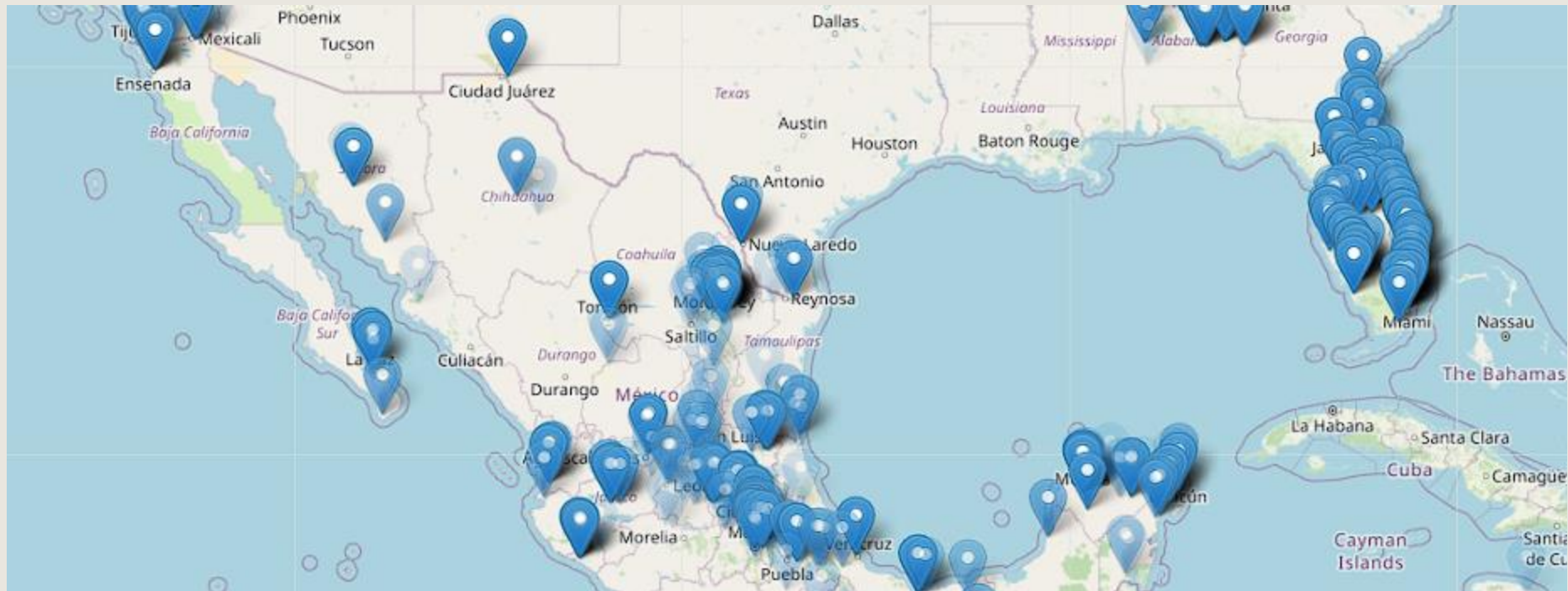
 Each numeric value had diagnostic information, command requests and command responses

This turned out to be the ICCID of the SIM

 Diagnostic information included location data, speed, battery percentage etc.

 Obviously a fleet management system

VISUALIZING THE DATA



FINDING THE OWNER

Reverse DNS gave us some domain names

Also one other MQTT server in the same domain address

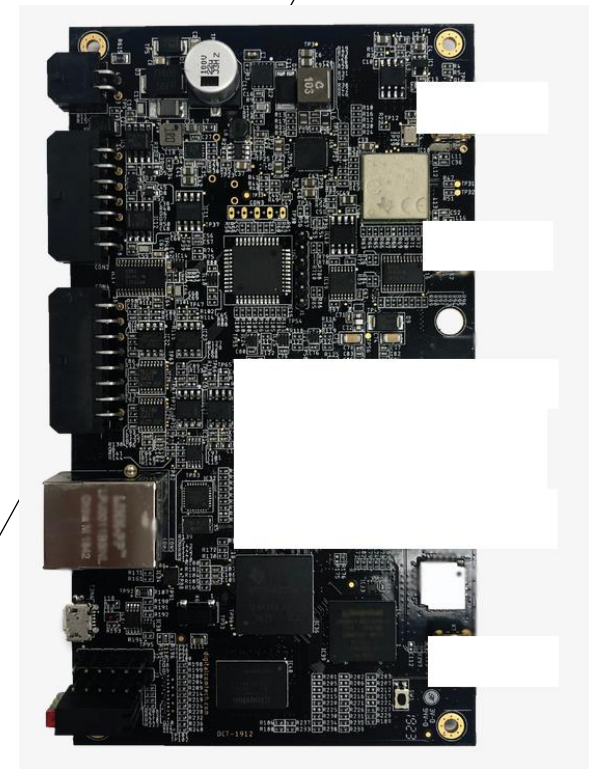
WHOIS plus the web server on the www subdomain gave enough context

Company F provides a fleet management service

They also provide devices to other services to track their vehicles

Very well documented online:

 Made it easier to understand the different systems



CAN YOU GET ROOT?

- We have location data, which means we know where the device/vehicle is
- Each subtopic has a command request/response subtopic
- Each command we saw had a token and an ID
- What happens if we send a command to the subtopic?
- We just reused the token for each command and it works for some commands
 - Others returned an error
- What's happening?

GETTING THE FIRMWARE

- We still don't have firmware
- Each device costs around 600 euros
- Should we buy one, extract the firmware and figure out the token scheme?
- We checked the data I dumped and found a helpful firmware update command containing the URL of the firmware
- Downloaded the firmware and extracted it

```
chyron4/865167066822371/orders/ask : b'{"cmd":"npx-os-update start -f -p https://chyron.companyf.com/npex/releases/npex-11.420.1","id":1698453989632}'  
chyron4/865167066822371/orders/tell : b'{"resp":{"mess":"Starting update from npex-11.420.0 to npex-11.420.1"},"id":1698453989632,"exit_code":0}'
```

REVERSE ENGINEERING THE FIRMWARE

- Firmware was easy to reverse engineer
 - No encryption
 - No complex filesystem
 - Binwalk extracted it on first try
- Grepped the extracted filesystem to find the binary responsible for connecting to the MQTT service
- The chyron-watcher binary seems interesting!

REVERSE ENGINEERING THE WATCHER BINARY

- No token required!
- Sometimes the simplest solution is best
- Credentials for future MQTT security?

```
0x1_0002074 - 0x1_0002074 & 0x11111100,  
syslog(7,"Processing: %s",&input_value);  
iVar3 = FUN_000131d4("run-cloud-script",&input_value);  
if (iVar3 == 0) {  
    *(undefined4 **)((int)aiStack_1770 + iVar1) = &input_value;  
    snprintf(command_to_run,sVar2 + 0x14,"cmd-wrapper %d \'%s\'",4000);  
    local_1744 = fopen(command_to_run,"r");  
    if (((local_1744 != (FILE *)0x0) &&  
        (pcVar4 = fgets((char *)&local_f58,0xf3c,local_1744), pcVar4 != (char *)0x0)) &&  
        ((char)local_f58 != '\0')) {  
        sVar2 = strchrn((char *)&local_f58,"\n");  
        auStack_f54[sVar2 - 4] = 0;  
    }  
    fclose(local_1744);  
    local_1744 = fopen("/tmp/exit_code","r");
```

REVERSE ENGINEERING: OTHER INTERESTING FILES

```
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456 ( [REDACTED]
|
```

/etc/shadow includes easily cracked passwords

Users have sudo rights for some commands

Private keys for SSH to a backend

ECU configuration data

Interesting video capture information

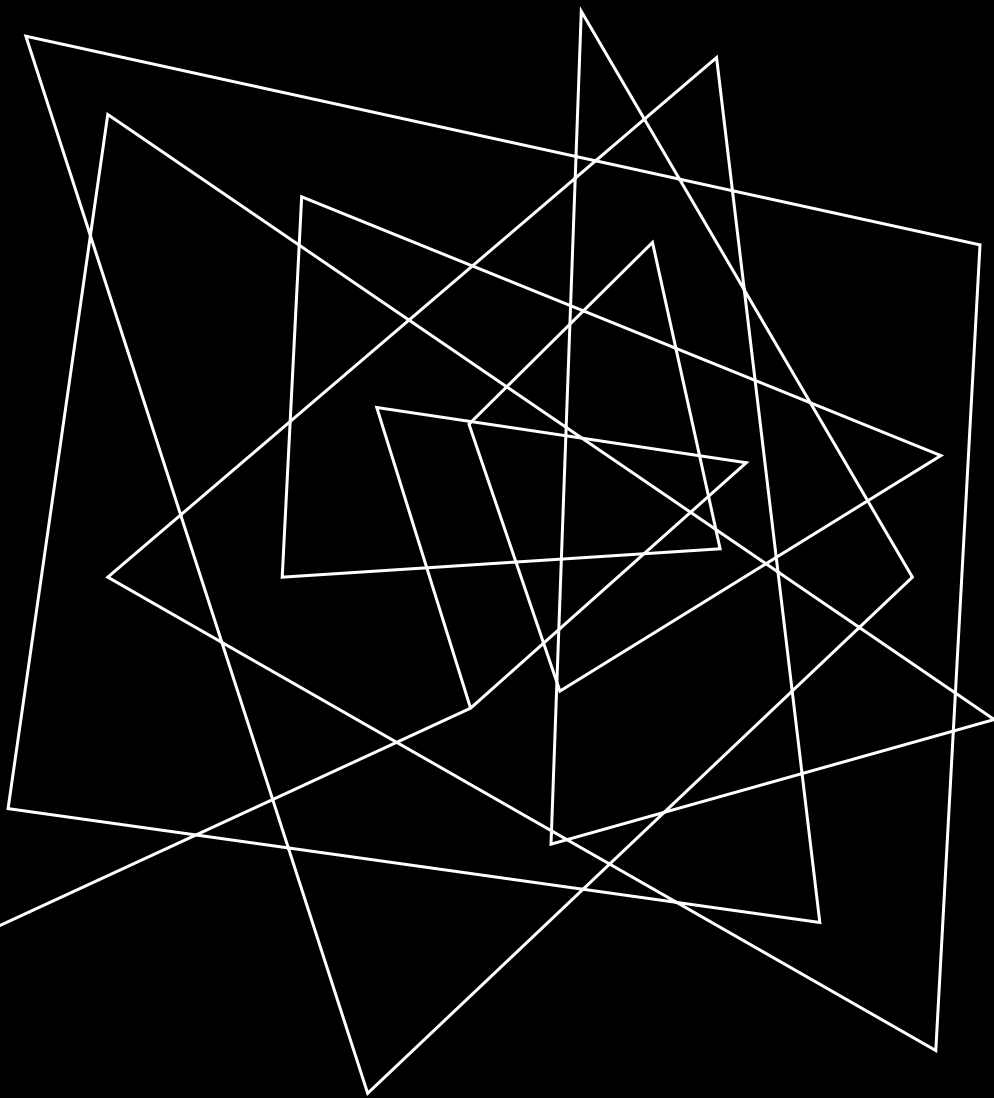
```
-----BEGIN RSA PRIVATE KEY-----
```

```
1 Cmnd_Alias npx_APPS = /bin/systemctl * npx-app*
2 chyron ALL=(ALL) NOPASSWD: npx_APPS
3
4 chyron ALL=(root) NOPASSWD: /usr/bin/npx*
5 chyron ALL=(root) NOPASSWD: !/usr/bin/npx-bist
6 chyron ALL=(root) NOPASSWD: !/usr/bin/npx-ppp
7 chyron ALL=(root) NOPASSWD: !/usr/bin/npx-wl-pwr
8 chyron ALL=(root) NOPASSWD: /bin/umount
9 chyron ALL=(root) NOPASSWD: /usr/bin/port-forwarding
10
11 support ALL=(root) NOPASSWD: /usr/bin/npx*
12 support ALL=(root) NOPASSWD: /bin/umount
```


CAN YOU GET ROOT PART 2?

- Get one of the id values and increment 100
- Send command with that id to the subtopic of the device that you want to hack
- id value turned out to be epoch in microseconds
- Wait for the response!
- Seems we're not root yet

```
publish_cmd(iicid,"whoami")
rc = mqttc.loop_forever()
b'{"resp":"chyon","id":1698613403332,"exit_code":0}'
```



DEMO!

CAN YOU GET ROOT PART 3?

- We have code execution and can execute any `/usr/bin/npx*` app as root
- Now we need to escalate our privileges:
- `npx-bbs` has some nice privilege escalation opportunities
- Can you see it?

```
167 option=$(echo $1 | tr '[:upper:]' '[:lower:]')
168 case "$option" in
169     set)
170         [ $# -lt 2 ] && print_error
171         parse_options "$@"
172
```


EXTRA CREDIT: GET LIVE VIDEO

GPS: -33.750775, 151.183967
Speed: 40.4 km/h



WHAT OTHER THINGS DO WE HAVE ACCESS TO?

- Tire Pressure Management Systems
- Engine Immobilization
- CAN Bus Access
- Send audio messages to drivers

 has a built-in ECU interface, which allows the device to connect to a vehicle's CAN bus and read data.

Safe Immobilization

Tool that activates the device's safe immobilization feature.

DISCLOSURE

Obviously these are critical findings
Developers need to know immediately
No way they wouldn't patch ASAP
Right?
Still unpatched as of yesterday

Dear Yashin Mehaboobe,

Your ticket "4966 Re: Reporting a vulnerability in the : Cloud system" has been Discarded.


We hope that we've helped you to the best of your satisfaction. To re-open this ticket simply reply to this email.

Regards,

--

Support Team

DISCLOSURE

 Ben Koo
CERT/CC


2023-11-02 (8 hours, 52 minutes ago)

[@yashin](#) We have not received a response from them either. At this point in time, we suggest grabbing a CVE for your finding and going with public disclosure. Unless I'm mistaken, the [hardwear.io](#) event will make your finding public before 11/11/23, so feel free to use any material/content from that to publish your finding.

For this case, we won't be publishing a vulnerability report; the November 11, 2023 date is the expected date to make the vulnerability public. Depending on the case, it's not a strict rule, especially given the circumstances for this vulnerability.

[Reply](#)


[Reply](#)

 yashin
Reporter

2023-11-02 (7 hours, 50 minutes ago)

[@Ben Koo](#) Thank you! For [hardwear.io](#) we are redacting some information (the name of the vendor, keywords for Shodan etc) from the presentation to prevent widespread exploitation. We intend to make them public after November 11. Could you help us with applying for a CVE? We are not sure if CERT/CC is a CNA.

[Reply](#)

 Ben Koo
CERT/CC

2023-11-02 (6 hours, 11 minutes ago)

[@yashin](#) Great, thank you for being sensitive to the situation. Good luck on your presentation!

Yes, in terms of going public, feel free to do so after 11/11.

April 16 2023 ————— Initial email to vendor

April 26/27 2023 ————— Second email and post to support channel

1 June 2023 ————— Third request via support channel

August 23 2023 ————— Ticket Discarded

TIMELINE PART 1

Jun 25 2023 — Involving ASRG

September 21 2023 — CERT/CC involved

October 30/31 2023 — Fourth request via support channel. Phone call to vendor HQ

November 3 — Hardwear.io disclosure

TIMELINE PART 2

SUMMING UP

SANY (HOPECHART)

- At least 60K heavy vehicles affected (SANY's estimation)
 - Probably more vendors affected
- Attackers can get:
 - Telemetry data including GPS
 - Impersonate vehicles
 - Read and inject CAN traffic
- Requirements:
 - Access to a single T-BOX device

UNDISCLOSED VENDOR

- 125K devices are potentially affected (based on vendor's website information)
- Attackers can get:
 - Telemetry data including GPS
 - Read and Inject can traffic
 - Runtime control of the ECU
 - Live video streams
- Requirements:
 - NONE! Everything found on internet, without physically accessing the T-Box



THANK YOU