black hat USA 2020
AUGUST 5-6, 2020
BRIEFINGS

Reverse Engineering the Tesla Battery Management System to Increase Power Available

By Patrick Kiley

RAPID7

#BHUSA @BLACKHATEVENTS

Patrick Kiley – Principal Security Consultant - Rapid7

- Member of the Penetration Testing team at Rapid7
- Performed research in Avionics security
- Internet connected transportation platforms
- Experience in hardware hacking, IoT, Autonomous Vehicles, and CAN bus

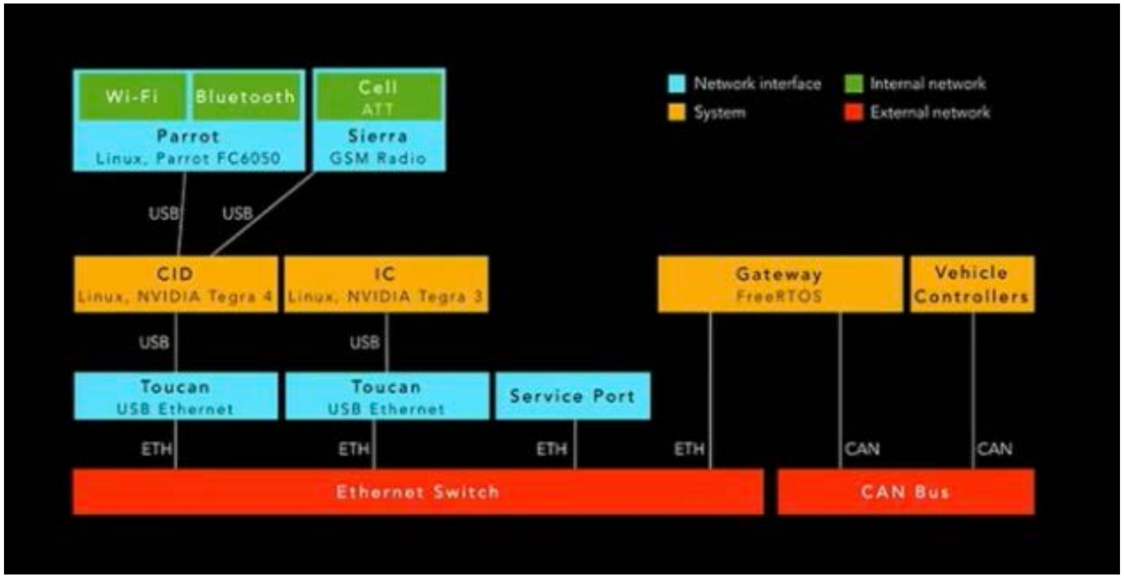
RAPID7

Topics

- Architecture of the Model S and Battery Management System(BMS)
- Performance and Ludicrous timeline
- Hardware changes
- Data stored in toolbox
- Firmware changes
- Shunt modification
- Upgrade process
- Failure and what I learned
- Next steps

Model S Architecture

- Central Information Display (CID): Nvidia Tegra based
- Gateway: a security component, stores vehicle configuration, sits between the various CAN buses and the CID
- Powertrain (PT) CAN bus, contains the BMS, Drive units, charging, thermal control and other powertrain related controllers
- PT CAN runs at 500 kBit/sec and is a standard vehicle CAN bus (differential signaling, 11 bit arb ids, etc)
- PT CAN supports UDS standard

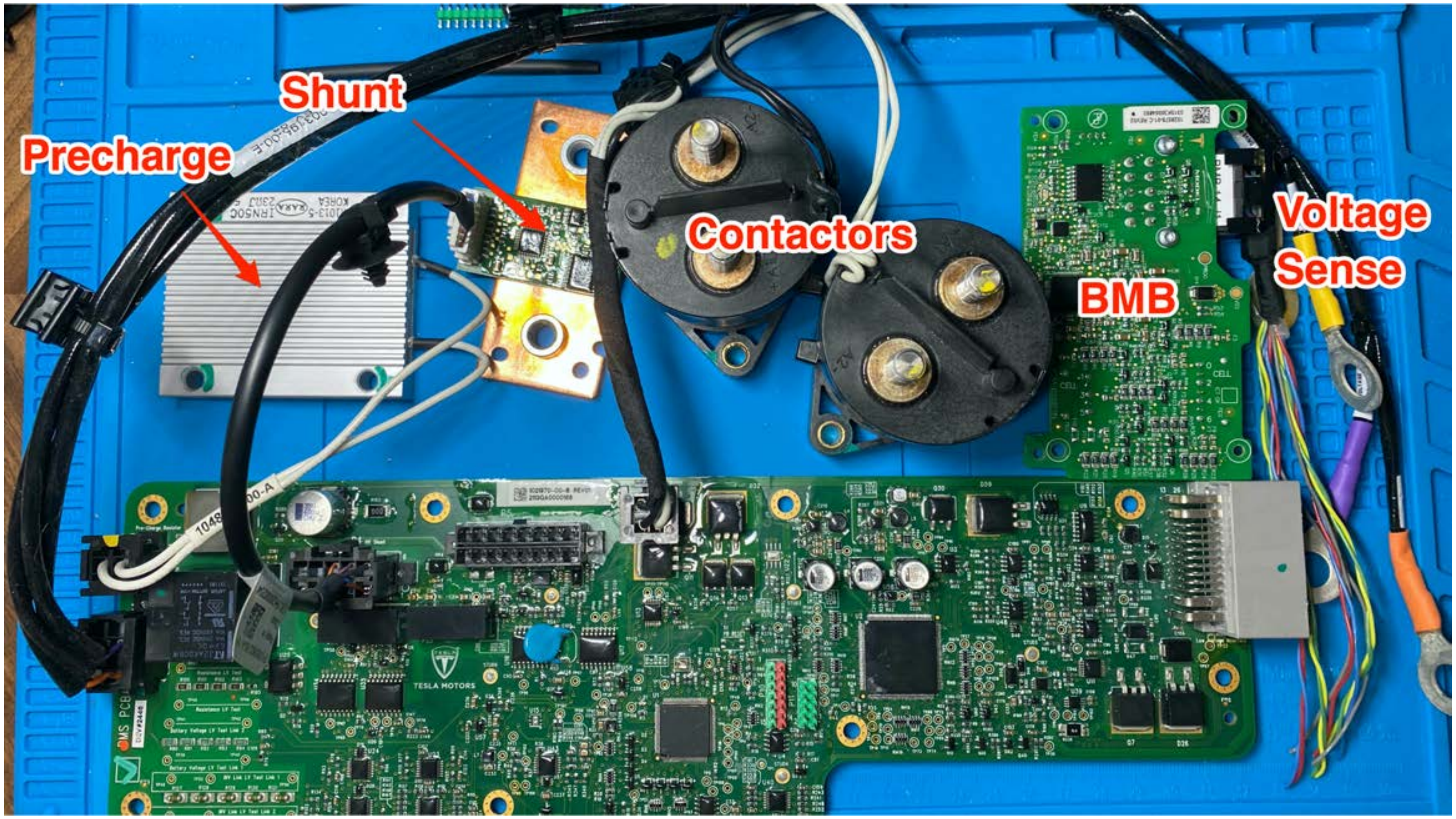


BMS Overview

- TI TMS320C2809 – Main microprocessor
- Altera CPLD – Hardware backup for TMS320
- Current Shunt with STM8 , measures current coming from the battery
- Precharge Resistor, prevents inrush current damage
- BMB boards on each battery pack, these include bleed resistors to balance packs

All the firmware changes are on the TMS320
Some settings are changed on the shunt, in addition it has a small physical modification

Full reversing of all the components is an ongoing project, so if you want to help, I am lacking in some of the skill areas


Shunt
Precharge
Contactors
BMB
Voltage
Sense
TESLA MOTORS

Ludicrous History

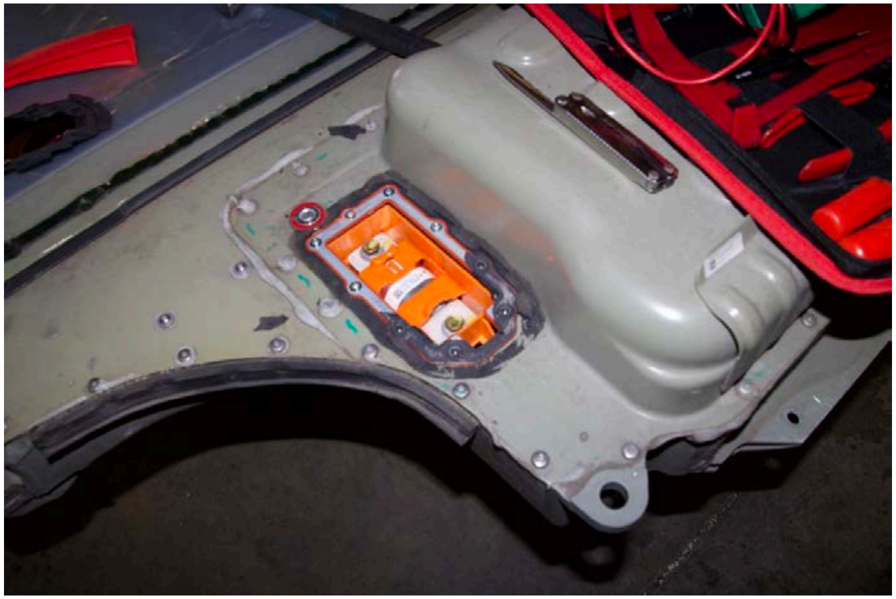
- P85D announced on Oct 10, 2014
- Ludicrous announced on July 17, 2015
- 10K for new buyers, 5K for existing P85D owners
- Upgrade involved new contactors and pyro fuse.
- Many performance battery packs would come standard with new components
- They were “ludicrous capable”
- All 100kWh performance battery packs are “ludicrous capable”
- Ludicrous capable means add “performanceaddon 1” to internal.dat on the gateway

black hat USA 2020

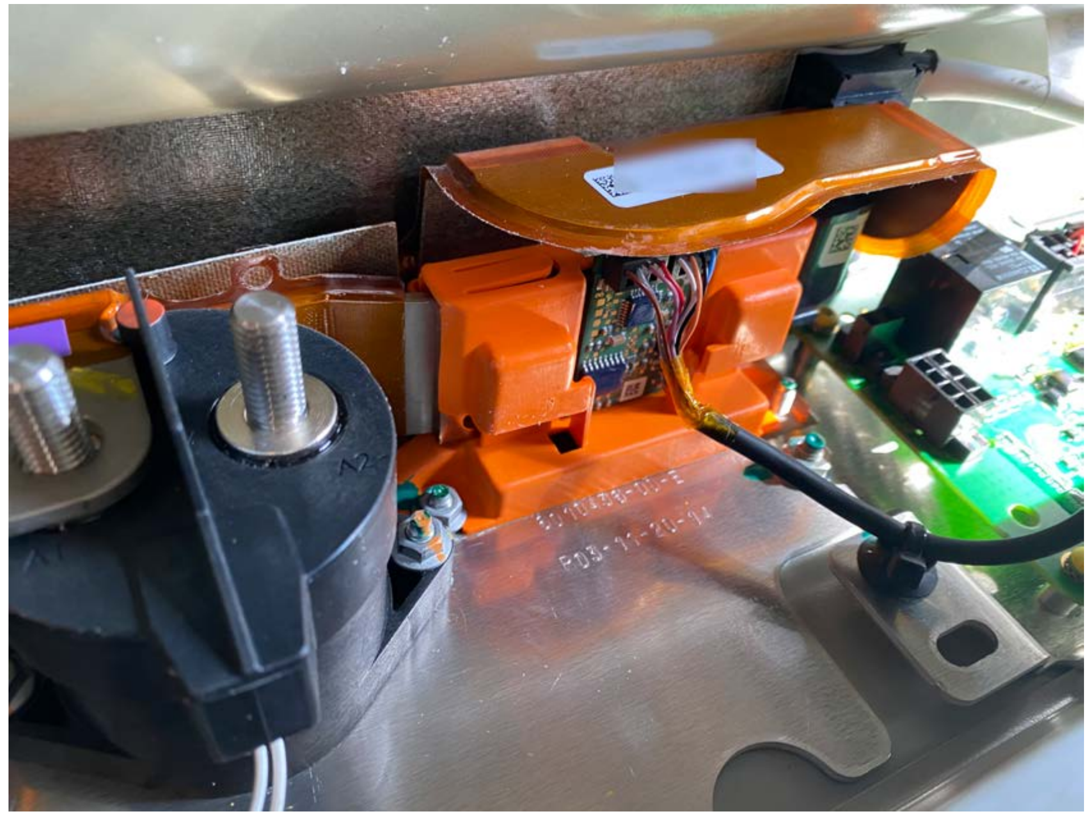
I upgraded a donor vehicle

Pack Dropped

Fuse and Contactor Bay



Shunt and Contactor Close up



What about firmware?

- For this we need to dig into some python
- Tesla makes a diagnostic tool called toolbox, runs on windows, uses encrypted and compiled python modules
- The important files are contained as individual plugins with the .scramble extension
- All of the information needed to decrypt the scramble files are on a machine that is running toolbox
- Some of these scramble files include firmware as well as many other useful items
- Once decrypted, we can use Uncompyle6 to give us source code
- Tesla left all the source code comments in place. Thank you!

Name	Type	Size
hci-2018.0.6-win32.scramble	SCRAMBLE File	548 KB
tbx_chademo-2018.0.1-win32.scramble	SCRAMBLE File	311 KB
tbx_coyote_cam-2018.0.1-win32.scramble	SCRAMBLE File	140,060 KB
tbx_dev-2018.0.5-win32.scramble	SCRAMBLE File	184 KB
tbx_driver_assist-2018.0.8-win32.scramble	SCRAMBLE File	142 KB
tbx_engineering-2018.0.1-win32.scramble	SCRAMBLE File	232 KB
tbx_fw_update_ext-2018.24.1-win32.scramble	SCRAMBLE File	458 KB
tbx_fw_update-2018.0.2-win32.scramble	SCRAMBLE File	19 KB
tbx_gen2_firmware-2018.0.1-win32.scramble	SCRAMBLE File	22,107 KB
tbx_gen2_meta-2018.0.1-win32.scramble	SCRAMBLE File	3,415 KB
tbx_gen2-2018.36.26-win32.scramble	SCRAMBLE File	2,780 KB
tbx_key_pairing-2018.0.4-win32.scramble	SCRAMBLE File	1,102 KB
tbx_meta_18_2_23-18.2.23-win32.scramble	SCRAMBLE File	13,573 KB
tbx_rest-2018.0.3-win32.scramble	SCRAMBLE File	1,205 KB
tbx_restraint-2018.0.2-win32.scramble	SCRAMBLE File	1,341 KB
tbx_security-2018.0.4-win32.scramble	SCRAMBLE File	102 KB
tbx_service-2018.33.4-win32.scramble	SCRAMBLE File	1,257 KB
tbx_steering-2018.0.1-win32.scramble	SCRAMBLE File	75 KB
tbx_suspension-2018.0.1-win32.scramble	SCRAMBLE File	58 KB
tbx_testing-2018.36.1-win32.scramble	SCRAMBLE File	43 KB
tbx_third_party-2018.0.2-win32.scramble	SCRAMBLE File	5,381 KB
tbx_uss-2018.0.1-win32.scramble	SCRAMBLE File	92 KB

Toolbox Uncompyled

```
# uncompyle6 version 3.3.2
# Python bytecode 2.7 (62211)
# [GCC 5.4.0 20160609]
# Embedded file name: build\bdist.win32\egg\vehicle\core\uds\data.py
# Compiled at: 2017-07-27 11:08:07
__author__ = 'Eric Hulser'
__email__ = 'ehulser@teslamotors.com'
__copyright__ = 'Copyright Tesla Motors Inc. 2013'
import logging
from xqt import QtCore
from .object import UdsObject
from . import errors
log = logging.getLogger(__name__)

class UdsData(UdsObject):
```

Helpful Comments

```
# Compiled at: 2017-07-26 15:43:06
"""
Defines the a VehicleTest to change the performance addon config.
"""
__authors__ = [
 'Otto Chiu']
__author__ = (',').join(__authors__)
__credits__ = []
__copyright__ = 'Copyright Tesla Motors Inc. 2015'
from tbx_gen2.testing import Gen2VehicleTest
import logging
log = logging.getLogger(__name__)

class PerformanceAddonConfig(Gen2VehicleTest):

    def exec_(self):
        """
        First verify that a vehicle can be configured in the desired config.
        For Standard mode, there is no checks; for Ludicrous mode, the vehicle
        needs to be AWD and has a battery pack config that supports 1500A+ current discharge.
        """
```

Data Structures – Extract and Binwalk

```
# uncompyle6 version 3.3.2
# Python bytecode 2.7 (62211)
# Decompiled from: Python 2.7.12 (default, Nov 12 2018, 14:36:49)
# [GCC 5.4.0 20160609]
# Embedded file name: build\bdist.win32\egg\tbx_gen2_firmware\resources\pyside_tbx_gen2_firmware_rc.py
# Compiled at: 2018-01-23 14:15:28
from xqt import QtCore
qt_resource_data = '\x00\x00\x00\x10<\xb8d\x18\xca\xef\x9c\x95\xcd!\x1c\xbf`\xa1\xbd\xdd\x00\x00\x00\x10<\xb8
qt_resource_name = '\x00\x11\n|*E\x00t\x00b\x00x\x00_\x00g\x00e\x00n\x002\x00_\x00f\x00i\x00r\x00m\x00w\x00a\
qt_resource_struct = '\x00\x00\x00\x00\x00\x02\x00\x00\x00\x01\x00\x00\x00\x01\x00\x00\x00\x00\x00\x02\x00\x0

def qInitResources():
    QtCore.qRegisterResourceData(1, qt_resource_struct, qt_resource_name, qt_resource_data)


def qCleanupResources():
    QtCore.qUnregisterResourceData(1, qt_resource_struct, qt_resource_name, qt_resource_data)


qInitResources()
# okay decompiling /home/can/Desktop/tbmaster/Roaming/Tesla/plugins/service_alpha//tbx_gen2_firmware-2018.0.1
```

Bootloader

We already know from the donor vehicle's config that it had a pack id of "57"

These are the files we need from the extracted firmware

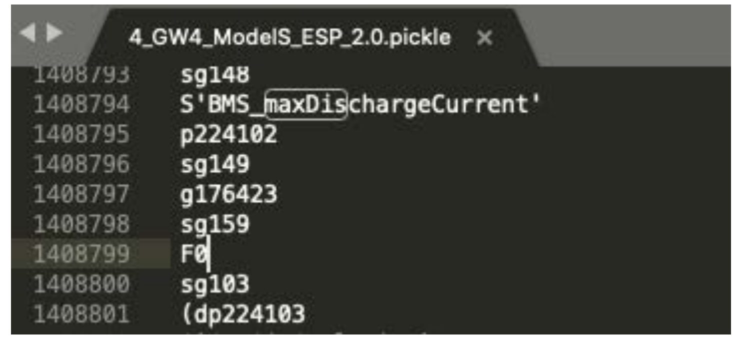
Pack id 57 becomes pack id 70 after the changes

```
# Changes HWID from 57 to 70

# http://artifacts.teslamotors.com/jenkins-job/bootloaders/git/21c44cbe0713a1beaa105b00cf
- name: 57 Gateway Application
  description: HWID 57 Gateway Application for Shunt Calibration
  filename: firmware/bms/withSecondaryBoot_UDSBoot_BMS_GATEWAY_APP_HWID-57.hex
  ludicrousable: True
  hwid: 57
  calibrateShunt: True
  linkedPackageName: 57 Updater

- name: 57 Updater
  description: HWID 57 Updater File
  filename: firmware/bms/withSecondaryBoot_UDSBoot_BMS-R57-CSM_UPDATER_SVN-68454.hex
  flashBootUpdater: True
  linkedPackageName: 70 Bootloader
  ludicrousable: True

- name: 70 Bootloader
  description: HWID 70 Bootloader
  filename: firmware/bms/withSecondaryBoot_UDSBoot_BMS-R70-CSM_SVN-71214.hex
  changeBootHWID: True
  module: UDS_FLASH_BOOTLOADER
  linkedPackageName: 70 Application
```

black hat USA 2020

Firmware Upgrade

- All the instructions and files needed for the upgrade process were stored in Toolbox files
- DBC files to help understand signals on the PT CAN bus
- ODX files that defined how to calibrate the shunt, grant security access and upgrade the firmware
- Files that stored calibration data and firmware
- Text comments and text data structures that offered clues on the process



CAN and UDS

Sitting on top of the CAN network stack is a protocol called UDS, or “Unified Diagnostic Services”, this protocol can be used to help technicians:

- Diagnose problems
- Read values from sensors
- Update firmware

CAN networks use a descriptor file called a DBC file

UDS networks can use a scripting file called ODX or GMD

Used commercial tool Vehicle Spy to assist in the research

ARBS 7E2 and 202 from BMS identify max current as a static value. 232 (BMS), 266 (DI) and 2E5 (DIS), identify max power in watts, which varies based on SOC, temp, and power recently used


```
<STRUCTURE ID="DLC.TESLA_BOOT.BV.TESLA_BOOT.STR.SHUNT_CALIBRATION_DATA_READ">
  <SHORT-NAME>SHUNT_CALIBRATION_DATA_READ</SHORT-NAME>
  <LONG-NAME>SHUNT_CALIBRATION_DATA Read</LONG-NAME>
  <BYTE-SIZE>11</BYTE-SIZE>
  <PARAMS>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>SHUNT_HWID</SHORT-NAME>
      <LONG-NAME>Shunt HWID</LONG-NAME>
      <BYTE-POSITION>0</BYTE-POSITION>
      <DOP-REF ID-REF="DLC.TESLA_BOOT.BV.TESLA_BOOT.DOP.UINT_1BYTE"></DOP-REF>
    </PARAM>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>CGI1_DATA</SHORT-NAME>
      <LONG-NAME>CGI1 Data</LONG-NAME>
      <BYTE-POSITION>1</BYTE-POSITION>
      <DOP-REF ID-REF="DLC.TESLA_BOOT.BV.TESLA_BOOT.DOP.UINT_2BYTE"></DOP-REF>
    </PARAM>
    <PARAM SEMANTIC="DATA" xsi:type="VALUE">
      <SHORT-NAME>CAU1_DATA</SHORT-NAME>
      <LONG-NAME>CAU1 Data</LONG-NAME>
      <BYTE-POSITION>3</BYTE-POSITION>
      <DOP-REF ID-REF="DLC.TESLA_BOOT.BV.TESLA_BOOT.DOP.U
```

DBC Turns this

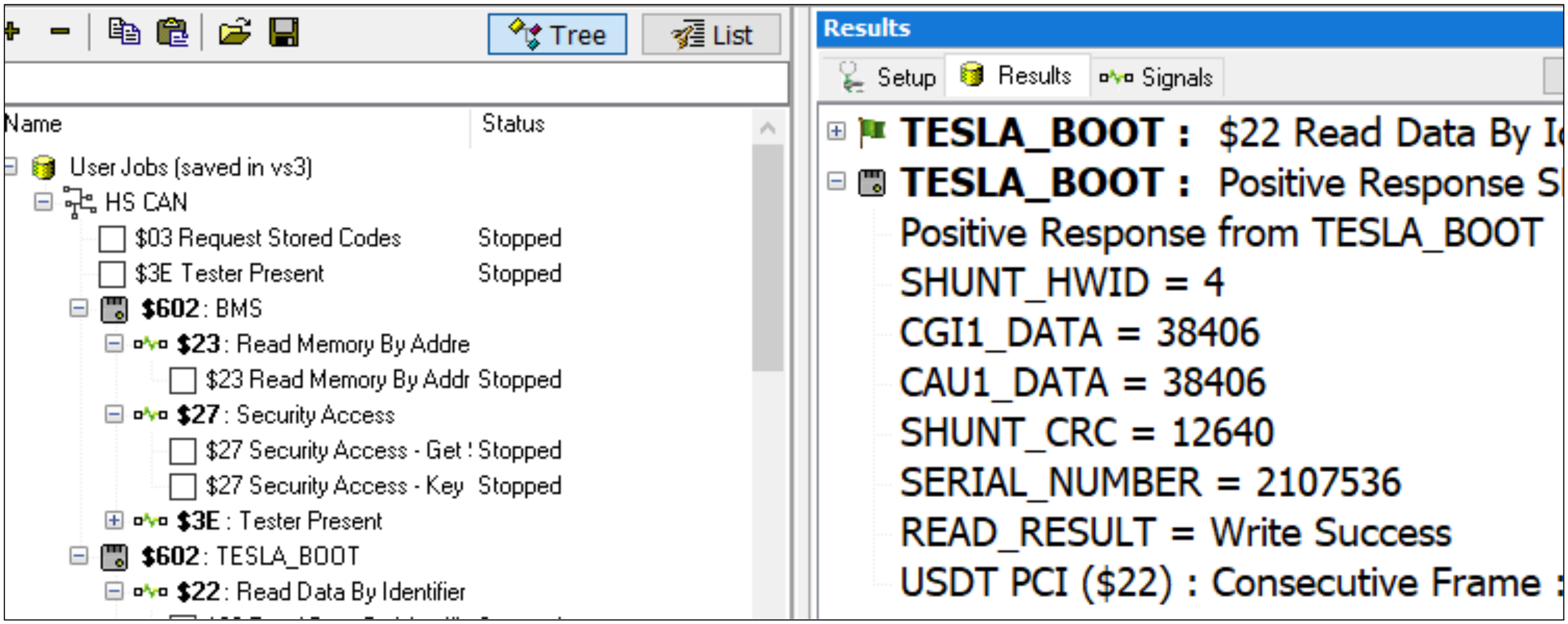
	Count	Time (abs/rel)	Tx	Er	Description	ArbId/He...	Len	DataBytes	Network
Filter									
	3298	10.000 ms			HS CAN $102	102	8	00 00 32 A6 64 4C 08 00	HS CAN
	330	100.006 ms			HS CAN $202	202	8	80 9D 1E 98 00 00 00 00	HS CAN
	330	100.004 ms			HS CAN $212	212	5	40 04 70 FF 00	HS CAN
	33	1.000002 s			HS CAN $218	218	8	7D 00 FF 27 FF DF 7F 37	HS CAN
	330	100.238 ms			HS CAN $222	222	6	00 00 00 00 00 10	HS CAN
	330	100.474 ms			HS CAN $232	232	4	88 26 5E 07	HS CAN
	33	1.000020 s			HS CAN $242	242	8	00 00 00 00 00 00 00 00	HS CAN
	65	648 µs			HS CAN $246	246	5	DB 00 F5 03 49	HS CAN
	330	99.986 ms			HS CAN $248	248	7	28 29 00 2F 01 31 00	HS CAN
	330	101.181 ms			HS CAN $252	252	7	00 00 00 00 02 2B 00	HS CAN
	330	101.417 ms			HS CAN $262	262	8	00 00 00 00 00 00 00 00	HS CAN
	330	101.655 ms			HS CAN $272	272	8	00 E0 43 FB 00 00 00 00	HS CAN
	825	40.006 ms			HS CAN $2BF	2BF	8	FF 0F FC FF FF FF 9F 61	HS CAN
	330	100.002 ms			HS CAN $2C8	2C8	8	60 B0 03 FF 20 24 20 40	HS CAN
	330	101.664 ms			HS CAN $2D2	2D2	7	00 00 00 00 06 00 00	HS CAN
	33	1.000022 s			HS CAN $302	302	8	00 00 10 00 5F 67 40 00	HS CAN

black hat USA 2020

Into This

	Count	Time (abs/rel)	Tx	Er	Description	ArbId/He...	Len	DataBytes	Network
Filter									
⊞	43	1.000973 s			BMS_energyStatus	382	8	5C 00 00 00 00 00 A0 10	HS CAN
⊞	4326	10.000 ms			BMS_hvBusStatus	102	8	00 00 32 A6 64 4C 08 00	HS CAN
⊞	4	10.001053 s			BMS_iSensorInfo	532	8	00 00 00 00 00 00 00 00	HS CAN
⊞	31	1.000975 s			BMS_info	5D2	8	0D 00 00 00 A3 2A A2 39	HS CAN
⊞	43	1.001201 s			BMS_kwhCounter	3D2	8	17 6E 68 00 31 DB 61 00	HS CAN
⊞	4	10.001059 s			BMS_odometerSta...	562	4	DC D4 45 02	HS CAN
⊟	432	100.003 ms			BMS_powerAvailable	232	4	88 26 5E 07	HS CAN
	BMS_maxRegenPower				= 98.640 kW	[2688]			
	BMS_maxDischargePower				= 18.860 kW	[75E]			
⊞	432	100.003 ms			BMS_ptNm	402	2	00 00	HS CAN
⊞	4	10.001053 s			BMS_serialNumber1	542	7	54 31 35 4C 30 31 31	HS CAN
⊞	4	10.001053 s			BMS_serialNumber2	552	6	39 37 31 39 00 00	HS CAN
⊞	43	1.000973 s			BMS_socStatus	302	8	00 00 00 00 E9 D0 10 00	HS CAN

ODX routines for shunt calibration



Shunt Modification

- Shunt also needed a hardware modification
- Single wire connecting the shunt to the CPLD
- If this wire remained connected after the firmware update then the BMS would generate an alert and refuse to close the contactors
- Discovered ran through the upgrade process on a bench version of the components
- Made a breakout board to monitor the signals from the shunt
- This also meant that the hardware and firmware both had to be updated before the car was driven

Upgrade Process

- Had access to garage and lift in Southern California
- Drove there to do upgrade, arrive with low SOC
- Drop pack, do hardware stuff
- Reinstall pack, carefully (image is from borescope)
- Flash BMS with special firmware for shunt modification
- Flash BMS to new packID
- Update internal.dat to add ludicrous and change packID
- Redeploy firmware due to changed battery packID
- Drive away and enjoy the ridiculous amount of torque?

Final Steps

- Using known techniques that I have used before, I tried to redeploy the firmware, also tried to upgrade since I had access to several versions
- The car failed using every method I tried
- Had to Tow the car from Rancho Cucamonga to Las Vegas so I could continue to work on it
- Cost me $360 or 3.6 hundred dollars, not great, not terrible right?

Learned something cool

- Gateway uses a file called firmware.rc
- Gateway uses this as a validation check for the components
- Calculated during upgrade/redeploy
- When the BMS changed, so did its CRC
- Changed the CRC based on CAN and value from “signed_metadata_map.tsv”
- Final CRC line is a JAMCRC based on overall file
- Car woke up, errors cleared and car could be driven
- Eventually figured out the reason for the earlier failure

firmware.rc

```
fileFormatVersion 1
platformType 1
platformVersion develop/2018.14.2-6-a88808ee6a
gtw 9acc071b
bms a0637e09
bmscpld 93.0.0
...(removed for clarity)
dhrp 3.11.0
dhfp 3.11.0
dhrd 3.11.0
dhfd 3.11.0
fileCrc 271d96ad

```

Power Before and After Upgrade

Before Upgrade
1300 Amps

4	2:46.994872	BMS_Debug_1Hz_89	7E2	8	89 00 00 7B 2A AA CB 02
BMS_1HzDebug_Id	=	BMS_1HZDBG_CTR_WOT_COUNTER [89]			
BMS_DRIVE_ctrWotCounter	=	0 [0]			
BMS_DRIVE_ctrWotCurrentLimit	=	1305.0 A [2A7B]			
BMS_DRIVE_ctrWotDeratingActive	=	0 [0]			

After Upgrade
1500 Amps

1		BMS_Debug_1Hz_89	7E2	8	89 00 00 59 31 85 00 00
BMS_1HzDebug_Id	=	BMS_1HZDBG_CTR_WOT_COUNTER [89]			
BMS_DRIVE_ctrWotCounter	=	0 [0]			
BMS_DRIVE_ctrWotCurrentLimit	=	1516.0 A [3159]			
BMS_DRIVE_ctrWotDeratingActive	=	0 [0]			
1		BMS_Debug_1Hz_8A	7E2	8	8A 00 00 00 DE 91 00 00

Actual
Available
Why Lower?

3762	99.895 ms	BMS_driveLimits	202	8	EA 5D 6D 9D 9D 09 8B 2D
BMS_minBusVoltage	=	240.420 V [5DEA]			
BMS_maxBusVoltage	=	403.010 V [9D6D]			
BMS_maxChargeCurrent	=	246.10 A [99D]			
BMS_maxDischargeCurrent	=	1492.4 A [2D8B]			

Further Research

- TMS320F2809 is supported in IDA Pro
- ARBS 7E2 and 202 define max current
- Seems possible to increase speed beyond ludicrous, it has been done by others (1000 HP RWD P85)
- Just need to find the variables and “bump them up a bit”, also might need to modify DU firmware
- Would be extremely dangerous to do so
- Could end up blowing up the Drive unit or battery pack, or worse, cause a fire and injury
- Still it would be interesting to reverse engineer, hit me up if you would like to assist, I have a dug a lot deeper than the information I am presenting here
- Would like to understand shunt parameters CAU1, CGI1

Referenced Material, Acknowledgements

Spaceballs movie, inspiration for Tesla Ludicrous https://www.imdb.com/title/tt0094012/

P85D announcement https://www.tesla.com/blog/dual-motor-model-s-and-autopilot

Ludicrous announcement and P85D upgrade offer https://www.tesla.com/blog/three-dog-day

What is a current shunt? https://youtu.be/j4u8fl31sgQ (electroboom)

TMS320 datasheet https://www.ti.com/product/TMS320F2809

Intrepid Control Systems, makers of Vehicle Spy software https://intrepidcs.com/

Bitbuster, for allowing use of lift and garage

The people who helped with the Toolbox reversing, you know who you are

Tesla security team for letting me do this talk