Intro to Car Hacking: Whitehat Car-Napping or is it?

...

Jay Turla @shipcod3

ROOTCON

whoami

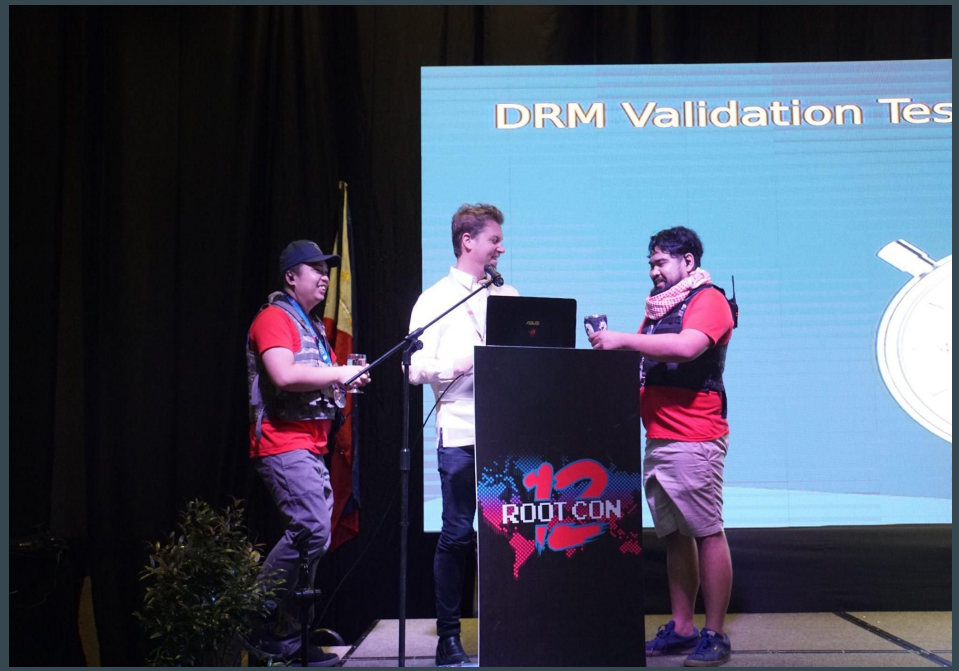
- @shipcod3
- Security Ops Manager (Philippines) at Bugcrowd
- ROOTCON Goon / CFP Review Board
- That pinoy on virtual cons w/ roosters in the background
- One of the main organizers of the Car Hacking Village in ROOTCON and PH → #CarHackVillagePH (https://www.carhackingvillage.com/about)
- msf contributor (auxiliary & exploit modules)
- Speaker at local & international conferences: ROOTCON, HITCON, DEFCON 26 Packet Hacking Village, OWASP Seasides, DragonCon, SarCon, Bsides Myanmar, Bugcrowd LevelUp, Nullcon, PEHCON and TCON

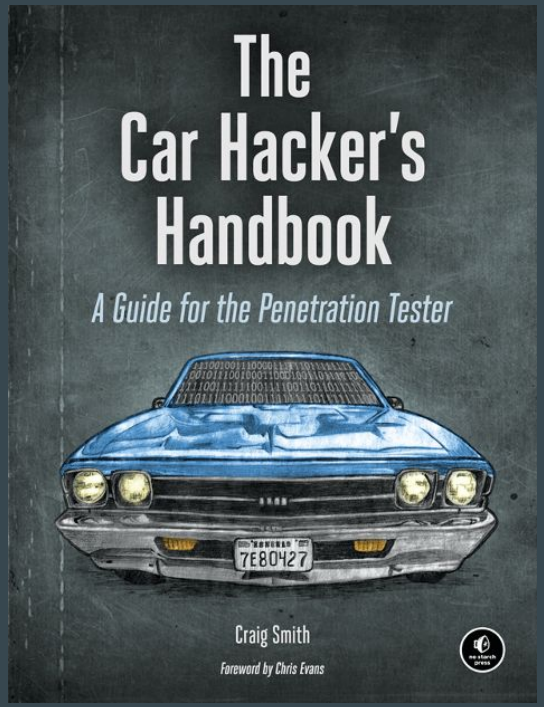
Greetz to Arun Mane and Nikhil Bogam for the Anomaly Detection System

Taken during ROOTCON 13 at the 1st Car Hacking Village PH

Best Car Hacking Book Ever

- Online version: http://opengarages.org/handbook/ebook/
- Written by Craig Smith





Is there money in Car Hacking?

BUG BASH
OCTOBER 12TH 2019

bugcrowd

Rank 1 — Private user — Points 120

Rank 2 — irotem — Points 114

Rank 3 — iangcarroll — Points 105

Rank	Researcher	Points
4	peri	95
5	uDISAn	76
6	fronders	60
7	anvol	60
8	Specters	44
9	BusesCanFly	43
10	Lennert	43

Bug bash is closed

Submissions are still being validated.

0 days : 00 hours : 00 minutes : 00 seconds

Bounties paid

$224000

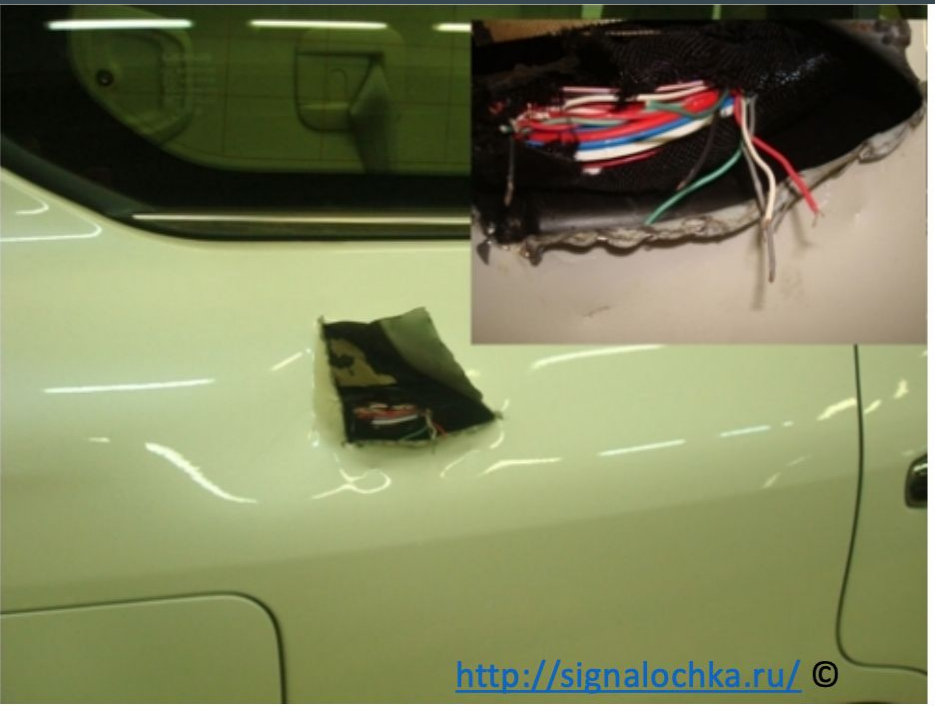
Sample Attacks Carried out on Attack Surfaces

- CAN Injection attacks on the Controller Area Network: OBD-2, sensors, unprotected components, etc
- Radio Attacks (keyfob replay attacks and rolljam)

```
jjt@ubuntu:~/pentot/hackrf-tests$ cat hackrf_listen.sh
hackrf_transfer -r 390_data.raw -f 3900000000 #listen
jjt@ubuntu:~/pentot/hackrf-tests$ cat hackrf_transmit.sh
hackrf_transfer -t 390_data.raw -f 3900000000 #transmit
jjt@ubuntu:~/pentot/hackrf-tests$
```

- Hardware attacks (requires physical access)
- IoT Hacking on infotainment systems, fms, telematics
- Web and mobile attacks on certain endpoints
- DoS (denial of service attacks) and Memory Corruption: Bluetooth, WiFi, Services

Unauthorized CAN Access



Reference:

https://www.blackhat.com/docs/eu-16/materials/eu-16-Sintsov-Pen-Testing-Vehicles-With-Cantoolz.pdf

Did You Know? Airplanes and Broom Broom Have CAN Too

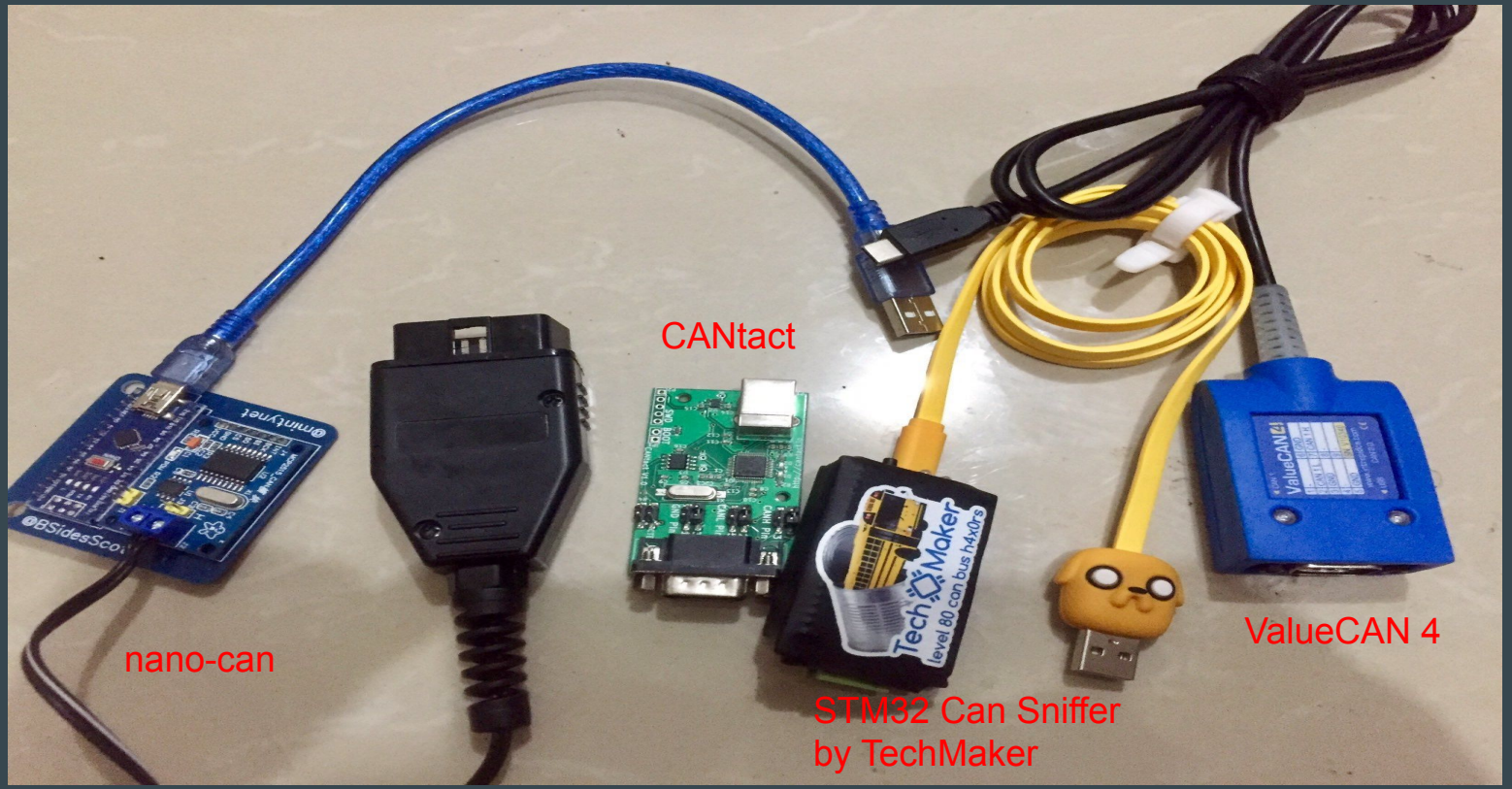
To reduce the number of interconnecting wires from control panels in the flight deck to system computers in the avionics compartment, Airbus deployed CAN bus. - **A380 Application**

Automotive Communication Bus Overview

Bus	LIN	CAN	FlexRay
Speed	40 kbit/s	1 Mbit/s	10 Mbit/s
Cost	$	$$	$$$
Wires	1	2	2 or 4
Typical Applications	Body Electronics (Mirrors, Power Seats, Accesories)	Powertrain (Engine, Transmission, ABS)	High-Performance Powertrain, Safety (Drive-by-wire, active suspension, adaptive cruise control)

Reference:
https://www.ni.com/en-us/innovations/white-papers/06/flexray-automotive-communication-bus-overview.html

My Favorite <3 (My tools)



CarHacking.Tools by jgamblin

- collection of scripts to help jump start car research and hacking
- All the scripts are designed to run on Ubuntu
- Install via Virtual Machine: https://carhacking.tools/install/beta/CarHackingToolsCHVBeta.ova
- Or can be installed via the repo:

```
git clone https://github.com/jgamblin/carhackingtools
cd CarHackingTools
sudo chmod +x *.sh
./toolinstall.sh
```

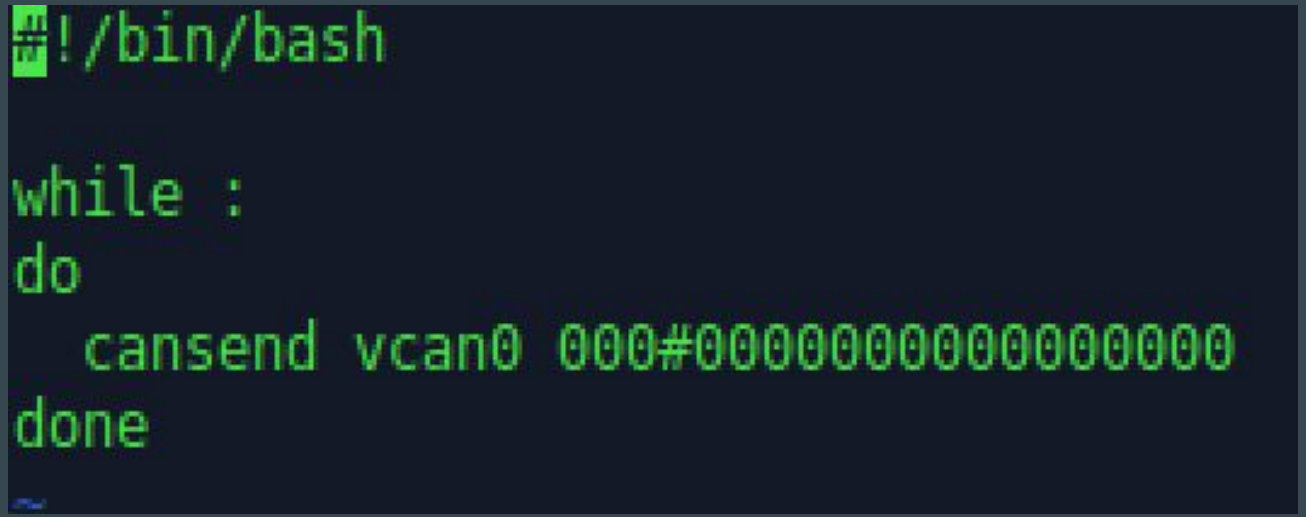
Try to learn how to cansend, cangen, candump, etc

```
jjt@ubuntu:~$ cansend
Usage: cansend - simple command line tool to send CAN-frames via CAN_RAW sockets.
Usage: cansend <device> <can_frame>.
<can_frame>:
 <can_id>#{R|data}          for CAN 2.0 frames
 <can_id>##<flags>{data}    for CAN FD frames

<can_id>:
 can have 3 (SFF) or 8 (EFF) hex chars
{data}:
 has 0..8 (0..64 CAN FD) ASCII hex-values (optionally separated by '.')
<flags>:
 a single ASCII Hex value (0 .. F) which defines canfd_frame.flags

Examples:
  5A1#11.2233.44556677.88 / 123#DEADBEEF / 5AA# / 123##1 / 213##311
  1F334455#1122334455667788 / 123#R for remote transmission request.

jjt@ubuntu:~$
```

Just a good tool you want to try

```
jjt@ubuntu:~/pentot/caringcaribou/tool$ python3 cc.py uds discovery

--------------------
CARING CARIBOU v0.3
--------------------

Loaded module 'uds'

Sending Diagnostic Session Control to 0x0710
  Verifying potential response from 0x0710
    Resending 0x710...  Success
Found diagnostics server listening at 0x0710, response at 0x077a
Sending Diagnostic Session Control to 0x07df
  Verifying potential response from 0x07df
    Resending 0x7df...  Success
Found diagnostics server listening at 0x07df, response at 0x077a
Sending Diagnostic Session Control to 0x07e0
  Verifying potential response from 0x07e0
    Resending 0x7e0...  Success
Found diagnostics server listening at 0x07e0, response at 0x077a
Sending Diagnostic Session Control to 0x07ff

Identified diagnostics:

+------------+------------+
| CLIENT ID  | SERVER ID  |
+------------+------------+
| 0x00000710 | 0x0000077a |
| 0x000007df | 0x0000077a |
| 0x000007e0 | 0x0000077a |
+------------+------------+
```

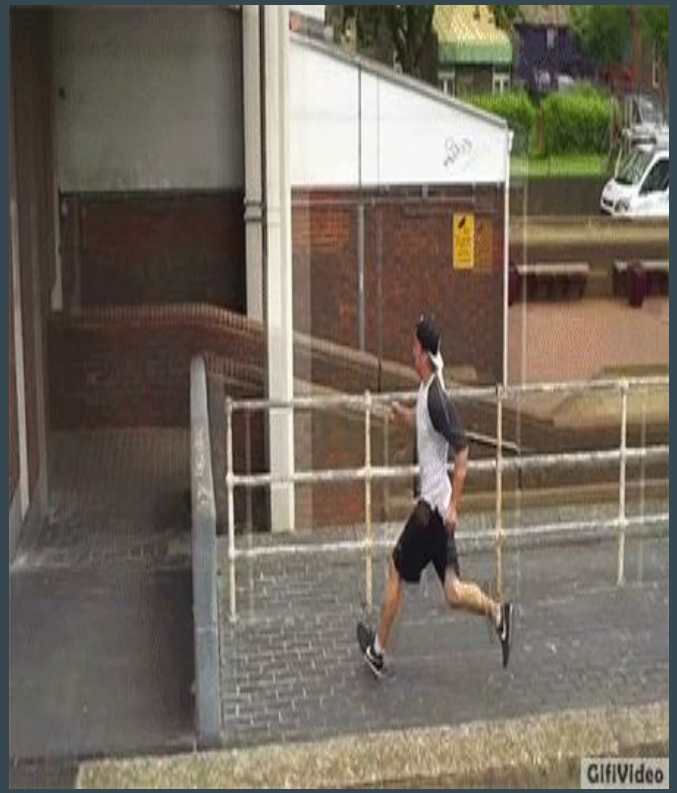
Scripting and Automation (Warning the script below can cause DoS)

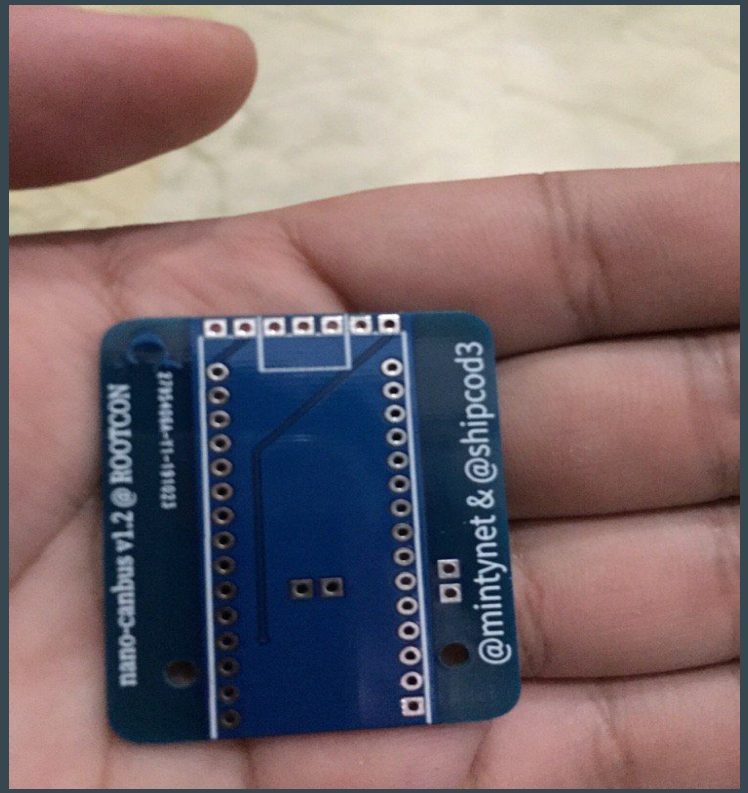


Scripting and Automation: Meet pyvit

```
from pyvit.hw.socketcan import SocketCanDev
from pyvit import can

dev = SocketCanDev('vcan0')
dev.start()

frame = can.Frame(0x7F1)
frame.data = [0x27,0x5F]
dev.send(frame)

frame.data = [0x27,0x60, 0xFE, 0x1C]
dev.send(frame)
```

If you spot me in a conference: if interested, ask me one

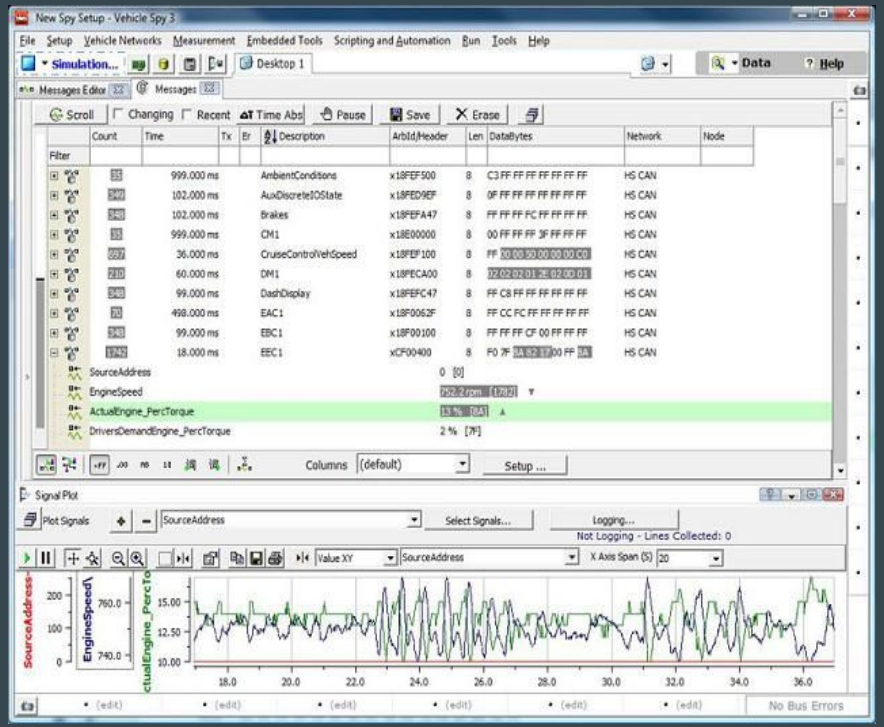


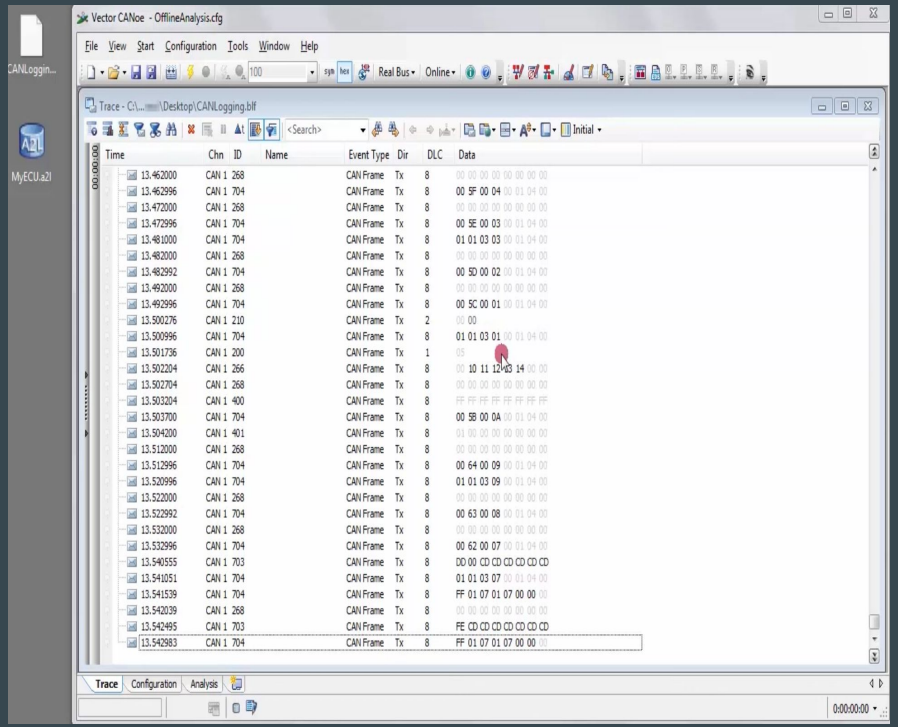


Upload code using Arduino IDE

- Sample CAN Sniffer: https://github.com/mintynet/nano-can/tree/master/can-receive-all (CAN Receive All)
- My other sketches: https://github.com/ROOTCONLabs/tools/tree/master/carhacking/sketches

Common Enterprise Tools on the Block: Vehicle Spy and CANoe





DEMO Time

References and Credits

- Car Hacker's Handbook
- https://github.com/mintynet/nano-can (nano-can)
- Getting Started with Car Hacking: https://www.carhackingvillage.com/getting-started
- CAN Bus Basics With Hands On Fuzzing (Ian Tabor): https://www.youtube.com/watch?v=6mxQFCHwpRI
- Shoutz: Car Hacking Village, Craig Smith, Ian Tabor, carfucar, fronders, techmakerua, ASRG, ASRG-SIN, specters, Jami, and semprix