"CAN -Controller Area Network Exploiting"

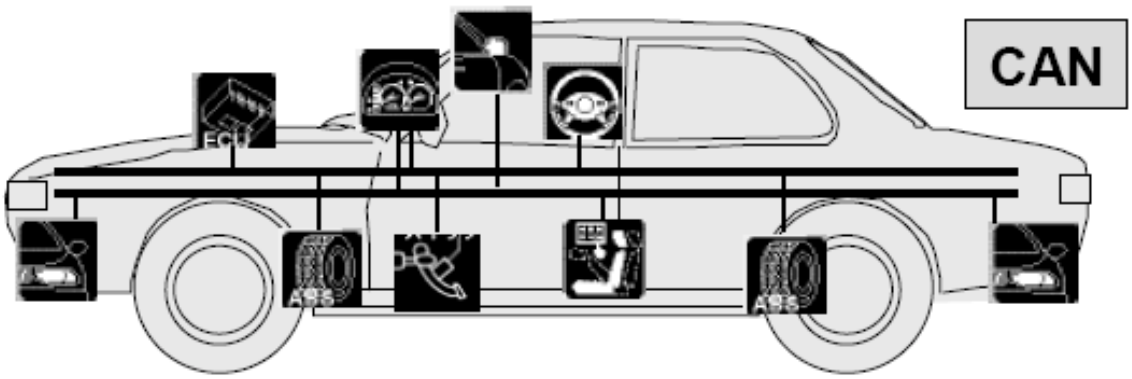
"A powerful computer with Steering and wheels"

1. Introduction

CAN bus

A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each others' applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but can also be used in many other contexts. For each device the data in a frame is transmitted sequentially but in such a way that if more than one device transmits at the same time the highest priority device is able to continue while the others back off. Frames are received by all devices, including by the transmitting device.

"CAN IS center Nervous system of car"



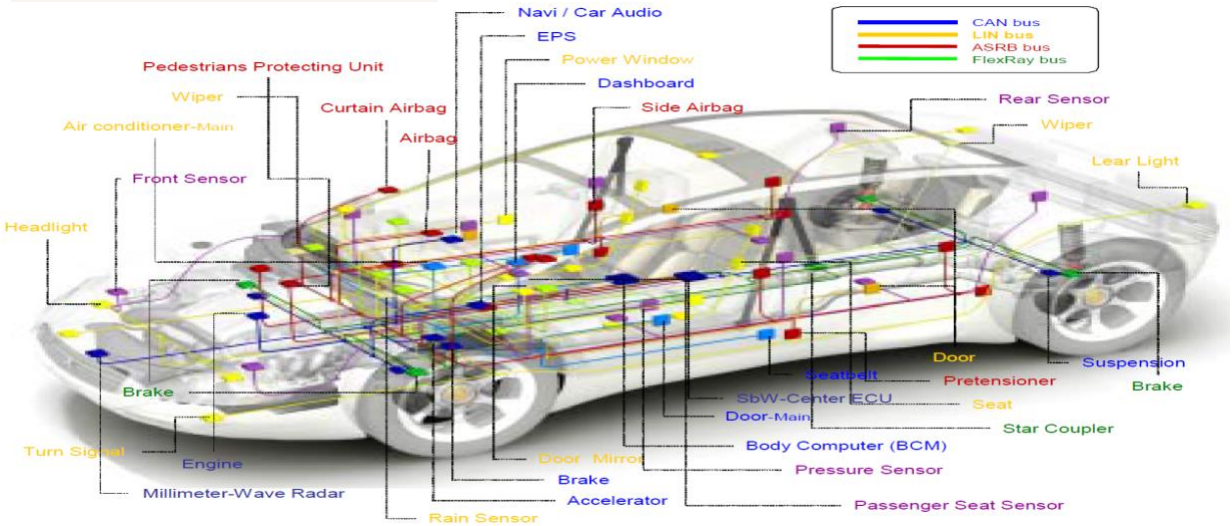
Application

- Passenger vehicles, trucks, buses (gasoline vehicles and electric vehicles)
- Agricultural equipment
- Electronic equipment for aviation and navigation
- Industrial automation and mechanical control
- Elevators, escalators
- Building automation
- Medical instruments and equipment

CAN Bus use in car

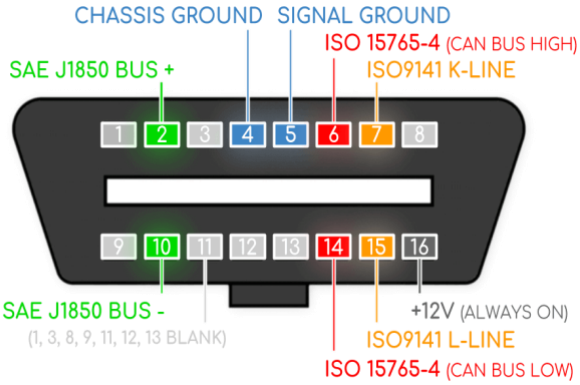
- The CAN bus protocol has been used on the Shimano DI2 electronic gear shift system for road bicycles since 2009.
- Various sensor inputs from around the vehicle (speed sensors, steering angle, air conditioning on/off, engine temperature) are collated via the CAN bus to determine whether the engine can be shut down when stationary for improved fuel economy and emissions.
- Electric Power brakes, Rain sensor, Auto lane assist/collision avoidance systems:
- Parking assist systems: when the driver engages reverse gear, the transmission control unit can send a signal via the CAN bus to activate both the parking sensor system and the door control module for the passenger side door mirror to tilt downward to show the position of the cur

"*you can get physical access to CAN via OBD-II using can to USB2CAN, this is located somewhere near the passenger's seat or driver's seat. And this should be accessible without the need of a screwdriver*
"



For Reference_ https://en.wikipedia.org/wiki/CAN_bus

2. CAN Ping Structure

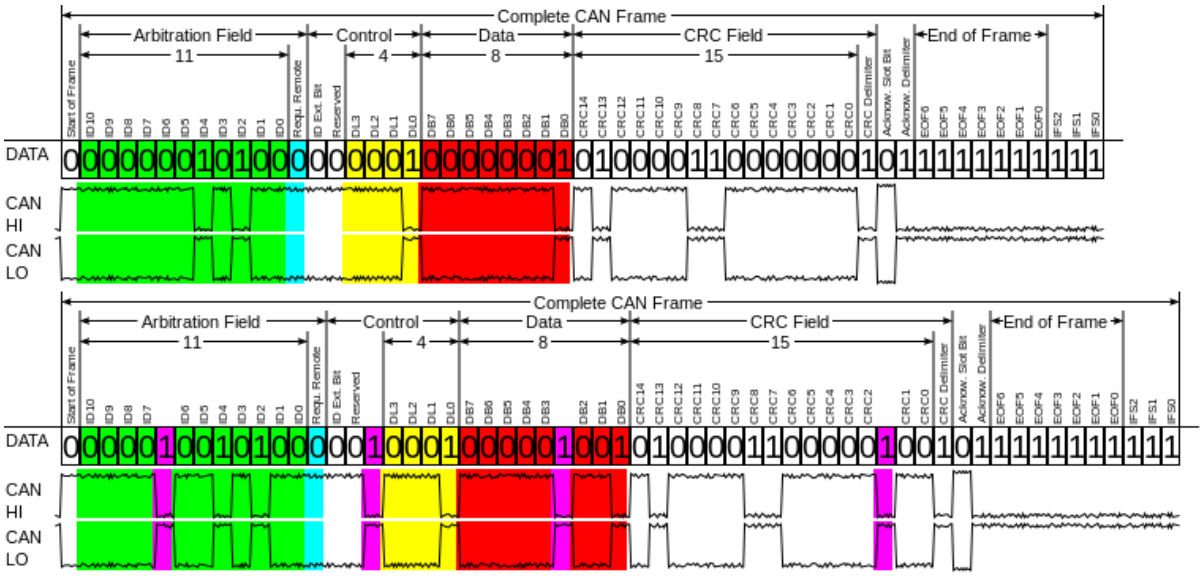


Basic working

A can have multiple nodes which are able to send and/or receive messages. Which contain essentially an ID(priority of the message) and also it can contain CAN message that can be of eight bytes or less at a time. Messages with numerically smaller value IDs are a higher priority and are always transmitted first.

“Message(break)>Message(audio)”

CAN Bus has 3 data frame: Arbitration Identifier, Data Length Code Data field



3. Setting Up Testing environment

- Operating System: Ubuntu 18
- Virtual Car interface
- CAN frame analysis(ICSim)
- Tool :canutill (In order for us to send, receive and analyze CAN packets)

CANUTILL (apt-get install can-utils -y)

- **cansniffer** for sniffing the packets |
- **cansend** for writing a packet |
- **candump** dump all received packets |
- **canplayer** to replay CAN packets |
- **cangen** to generate random CAN packets

ICSIM for generating CAN traffic

Download and install ICSim from (https://github.com/mrnamp/CAN) for virtual simulator in case or real car skip (step 1-2)

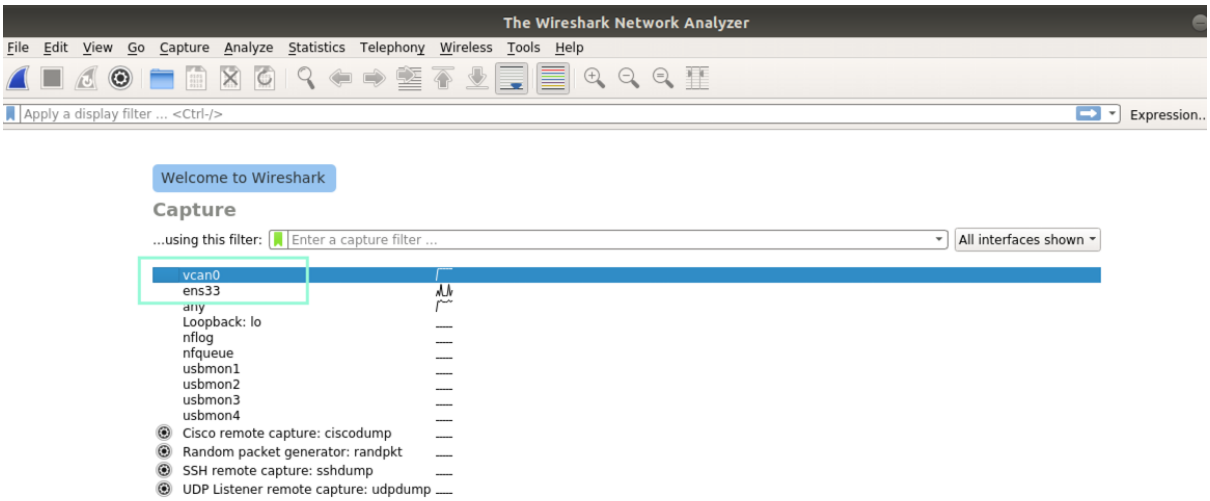
Step 1 ./setup.sh. {check vcan adaptor via ifconfig /iwconfig }

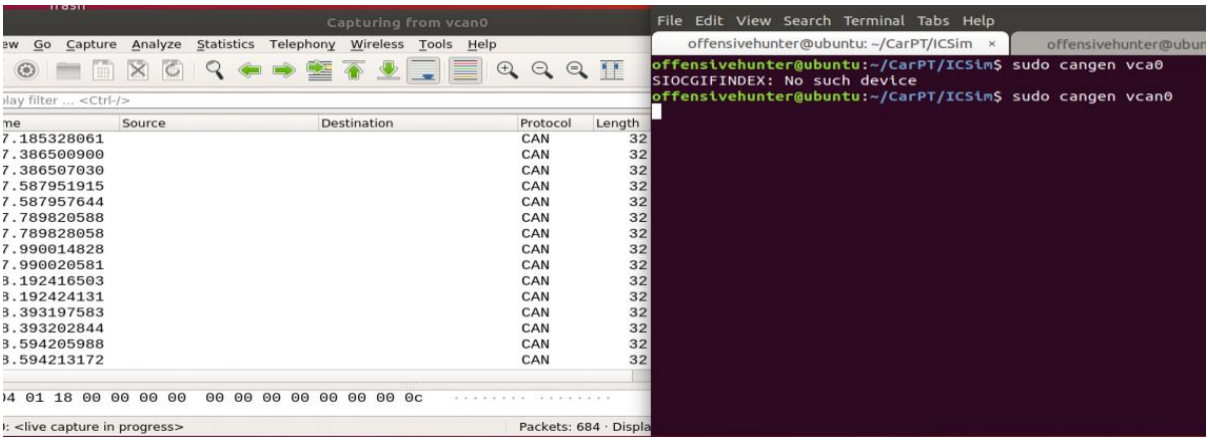
Step 2 ./icsim vcan0

Step3 ./controller vcan0

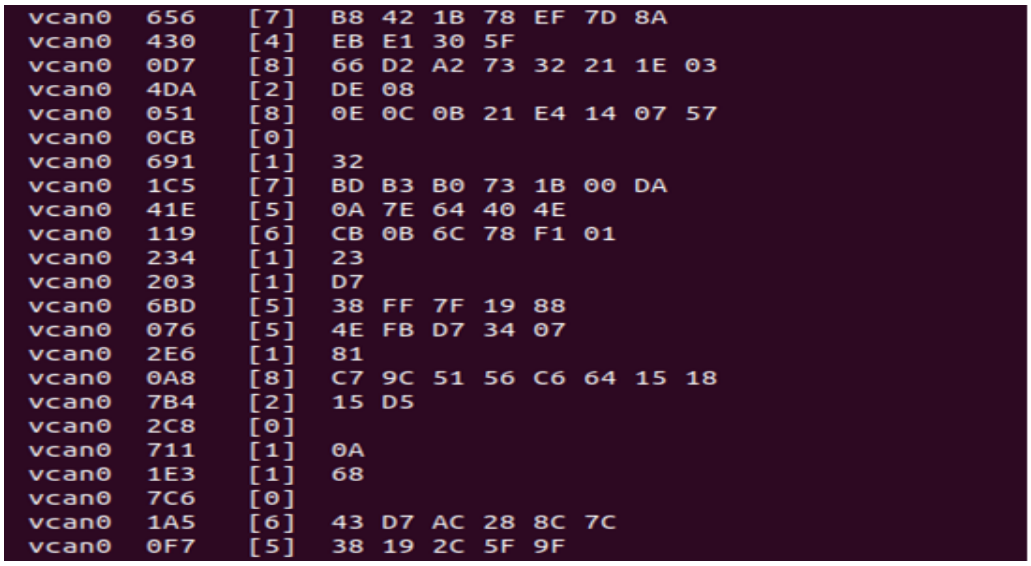
Using CAN UTILL (run those command in terminal) also run wireshark and capture traffic from vcan 0

a. cangen ($ cangen vcan0) -> generate random can packaet frame

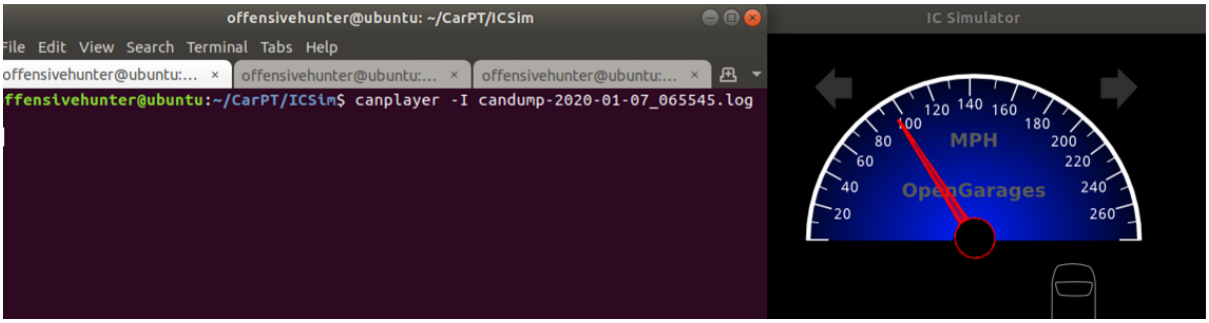




b. candump ($ candump vcan0) ->dump can trafic



c. canplayer for replay attack from capture frame
   eg
   canplay -I xyz.log



d. cansniffer to sniff change can traffic
   cansniff -c vcan0. Or type -00000 “enter”

e. cansend to send crafted packet to can interface (work like burp proxy)
   cansend vcan0 crafted_packet
   eg cansend vcan0 188#03. (tunr left(01) & right(02) both (03))