Tapping into the Automotive Nervous System

Yuqing He
Cedarville University, yuqinghe@cedarville.edu

Erin E. Eppich
Cedarville University, eeppich@cedarville.edu

Joel A. Ramireddy
Cedarville University, jramireddy@cedarville.edu

Stephen M. Impraim
Cedarville University, simpraim@cedarville.edu

Tapping into the Automotive Nervous System

Faculty Advisor: *Prof. Dudenhofer*
Students: *Erin Eppich, Rainbow He, Stephen Impraim, Joel Ramireddy*

Background

Historically, automobiles were wired using point-to-point connections. However, as technology advanced and the number of internal functions increased, the number of point-to-point connections also increased, leading to more complex and cumbersome wiring systems. Bosch invented the CAN bus to fix this problem.

What is a CAN Bus?
A controller area network (CAN) bus is a type of communication network that enables communication between various electronic control units (ECUs). They avoid point to point connections by using one common bus. All messages on this bus follow the CAN protocol.

What is an ECU?
An electronic control unit is an embedded system that controls one or more functions in a car.

Abstract

An Electronic Control Unit (ECU) is a crucial embedded system in automotive electronics that communicates via the Controller Area Network (CAN) buses in cars. These technologies have both continued to grow increasingly intricate and essential to every modern vehicle. Due to the increasingly integral role of ECUs and CAN buses, securing these entry points must become a top priority. By using Korlan, an OBDII to USB device, a physical communication portal can be successfully set up between a car and laptop. Using various python libraries and other open-source programs, cybersecurity professionals can dig into the potential vulnerabilities that lie in automobiles. This poster will attempt to highlight some of the procedures involved in unearthing these vulnerabilities.

Our Arsenal

Park Assist ECU: Provided by Bosch for research, analysis & testing

CAN-utils: Command line tool for reading, writing, collecting, and replaying CAN messages

Korlan: High quality USB to CAN interface for OBD-II port

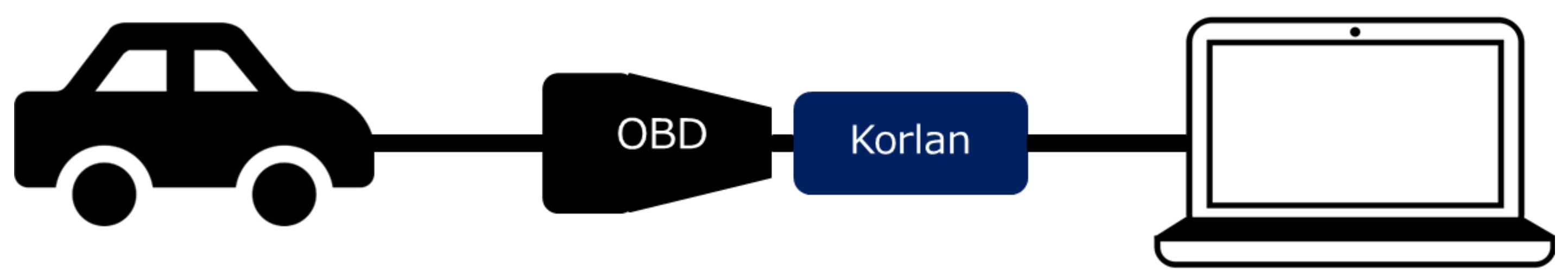
ELM 327 Bluetooth OBDII scanner: A Bluetooth interface for determining the baud rate of the CAN bus.

Research & Analysis

In order to achieve our objective of unearthing automobile vulnerabilities, we started with the OBD-II port as our exploitation access point. On-Board Diagnostics (OBD) is a standardized system that allows consumers to diagnose and troubleshoot internal issues in the car. It connects to a car's CAN bus and receives error codes from all the ECUs connected to it. Our sponsoring client, Bosch, loaned us a Park Assist ECU to aid our penetration testing research. The experience we gained from tinkering with the ECU allowed us to make progress with our car.

To read messages from the CAN bus, we had to first determine the speed of the bus. For most modern cars, the baud rate of the bus is usually 500 Kb/sec. We verified this speed by using a Bluetooth diagnostics device on our test vehicle, a Toyota Highlander 2012. We then connected the Korlan USB2CAN device to the OBD-II port in the car and plugged the other end to our computer. The Korlan comes with a software package that allows us to read and write messages. As soon we turned the car on, there were a flurry of messages. Though the Korlan software was useful for confirming that the bus was active and readable, it was quite difficult to decode any messages that we saw as they flew by. Thankfully, Korlan additionally creates a CAN network interface that we can use with programs other than the original Korlan software. We choose CAN-utils as it provides most freedom, and it can simply be accessed from the Linux terminal.

CAN-utils has a record feature that is useful in analyzing CAN messages. To utilize this feature, we performed a physical action in the car while recording with CAN-utils and then replayed the message to see if we captured it on the CAN bus. The first action we tried was opening a car door and seeing the door-open notification on the dash. We then narrowed down the entire log of recorded messages to a single message responsible for the door-open notification using the BREAD interface. The captured message can be replayed to simulate a door-open notification on the dashboard. Using this technique, we were able to get the doors to lock and unlock, the RPM indicator to move, the speed indicator to move, and other notification lights on the dashboard to turn on.

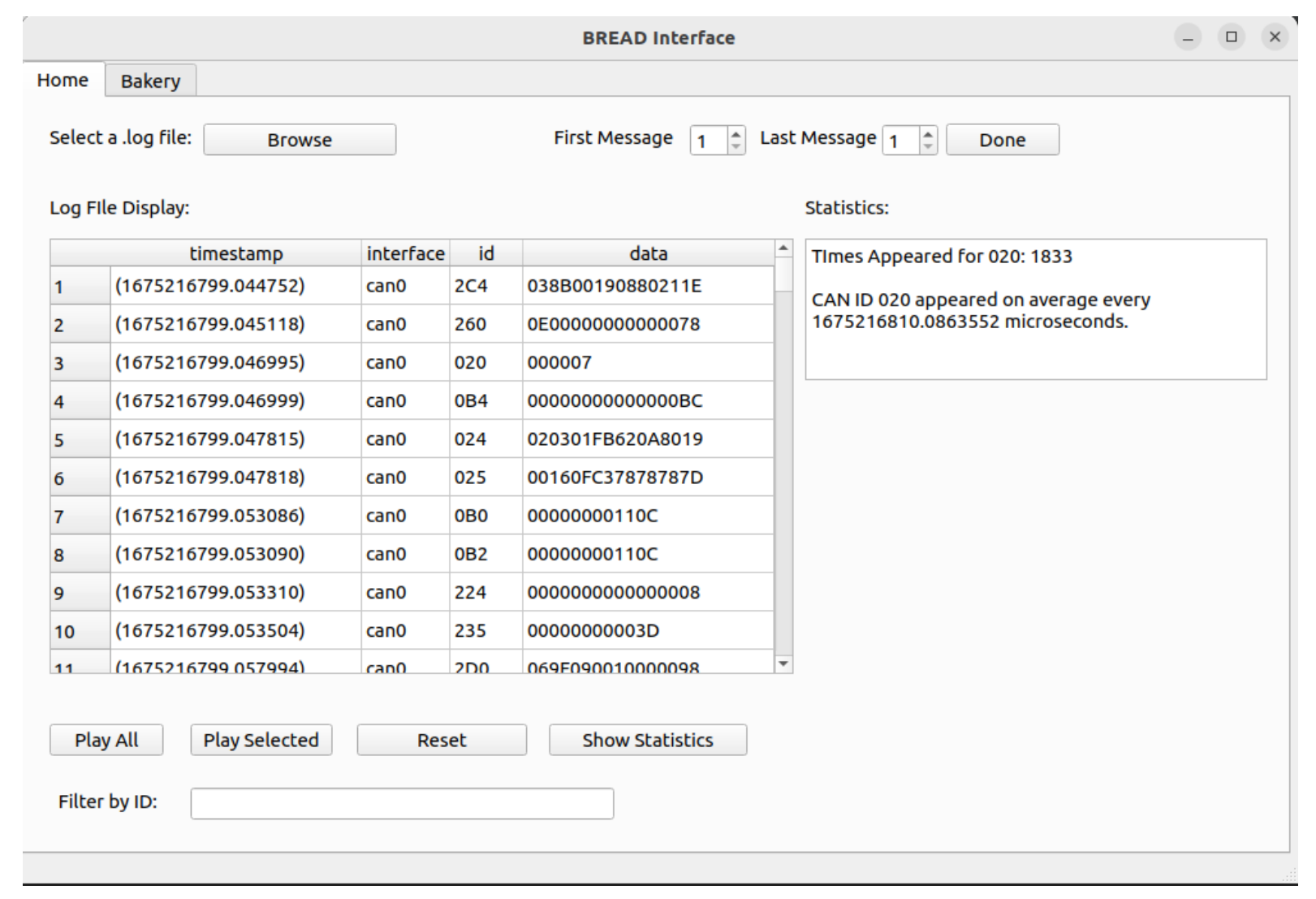


BREAD Interface

Based on what we found to be useful during our research process with the hardware, we have decided to develop a software interface to aid future researchers. The interface will allow us to analyze the CAN messages captured from the car. To achieve this, we have utilized Python QT, which works seamlessly with CAN-utils.

The interface will have various functionalities including:

- Static analysis for captured CAN messages.
- Display of CAN messages within a given range.
- Replay of selected CAN messages from the captured file.
- Replay of the entire captured file.
- Filtering of CAN messages from the captured file by CAN ID.
- Creation of customized buttons for CAN ID and CAN data, which can be sent when clicked.



Select a .log file:

First Message 1 Last Message 1

Log File Display:

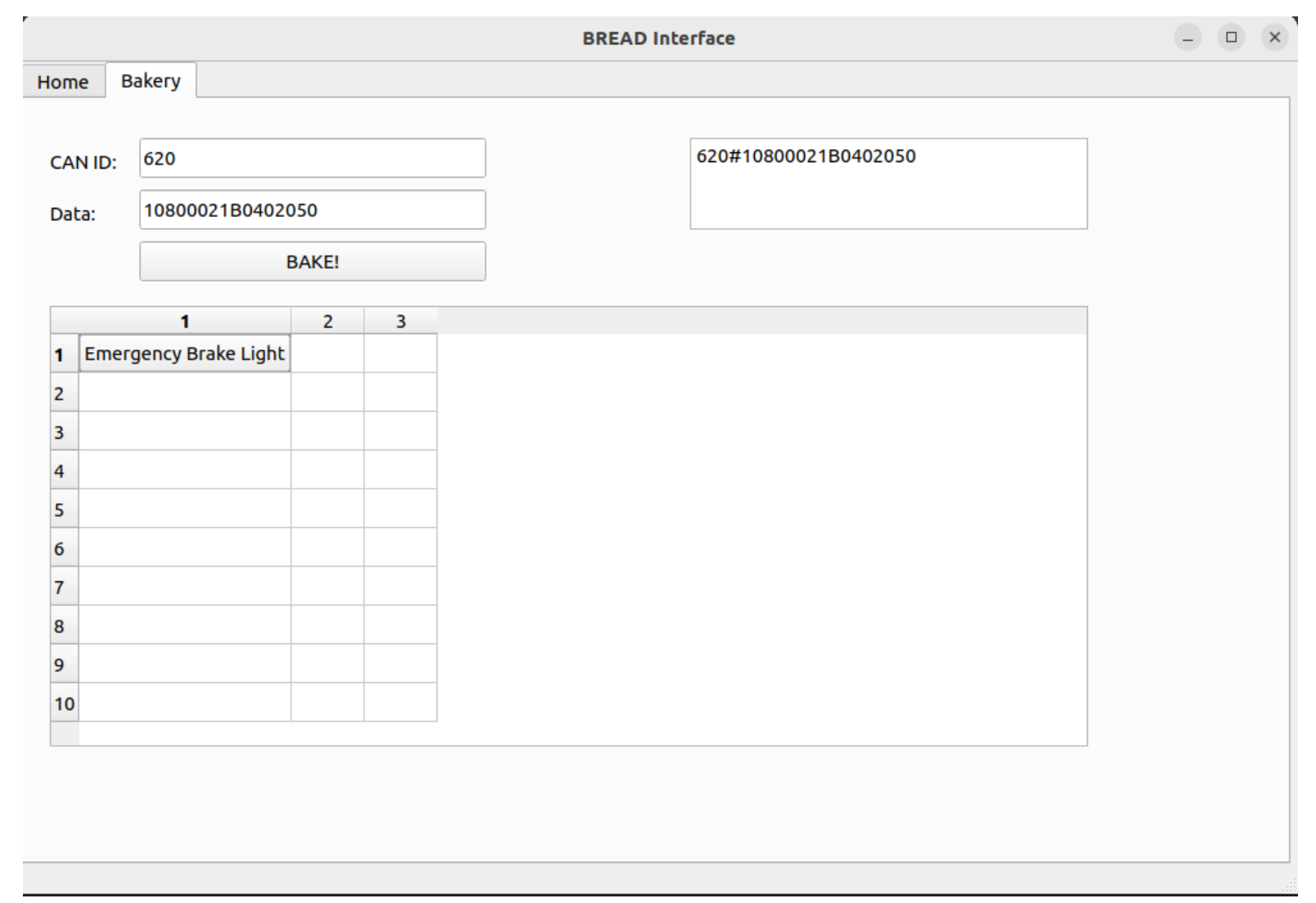
	timestamp	interface	id	data
1	(1675216799.044752)	can0	2C4	038B00190880211E
2	(1675216799.045118)	can0	260	0E00000000000078
3	(1675216799.046995)	can0	020	000007
4	(1675216799.046999)	can0	0B4	00000000000000BC
5	(1675216799.047815)	can0	024	020301FB620A8019
6	(1675216799.047818)	can0	025	00160FC37878787D
7	(1675216799.053086)	can0	0B0	00000000110C
8	(1675216799.053090)	can0	0B2	00000000110C
9	(1675216799.053310)	can0	224	0000000000000008
10	(1675216799.053504)	can0	235	00000000003D
11	(1675216799.057994)	can0	2D0	069F090010000098

Statistics:

Times Appeared for 020: 1833

CAN ID 020 appeared on average every 1675216810.0863552 microseconds.

Filter by ID:



CAN ID: 620

Data: 10800021B0402050

620#10800021B0402050

	1	2	3
1	Emergency Brake Light		
2			
3			
4			
5			
6			
7			
8			
9			
10			