PAPER 1: ATTACK (reference 1).

DOS ATTACK:

1. False Request to Send (RTS): [1]

**Overview of the FRTS attack: “// (kind of DOS attack with buffer overflow)//”**

J1939-21 standard uses CAN protocol to communicate in some commercial vehicles and in this protocol if they are flooded by RTS messages the protocol ignores the old messages and considers only the latest ones.

**Implementation:**

When a requester receives a RTS message (request-to-send) from the original source, the request is allocated to the buffer it might be of 2-3 bytes, later the requester sends CTS message (clear-to-send) which requests certain packets in sequence.

Now the attacker can exploit it by sending self crafted RTS messages with smaller bytes which looks like original RTS message. But the buffer will allocate these RTS messages, and every message will be registered later the buffer size will be overflowed, making the electronic control unit (ECU) to crash.

**Still working:**

NO

1. Connection exhaustion: [1]

**Implementation:**

In J1939-21 standard each device has 8-byte address which allows an ECU to have 225 connections at once. But in reality, there are usually less than 255 so that an attacker can scan for active devices and make fake connection, so they pretend to be the same and blocks the other devices to connect to it. So other ECUs cannot connect to it and jam’s the network.

**Mitigation:**

Authenticating the ECUs before connection can prevent this type of attacks.

**Still working:**

NO

**Resources needed:**

Connection to CAN bus, data frame understandability.

Paper 2: ATTACK (reference 2)

Reply attack:

**Overview:**

Here the attack is performed on a stimulator, considering 3 nodes acting as ECUs and they are connected in a CAN bus. In order to achieve this in real life, the attacker should some how get connection to the CAN bus of the car and act as an ECU, the reply attack can be done in 2 ways one with transmitting the same and other is with manipulation.

**Implementation 1: without manipulation of spoofed can message.**

The nodes A, B and C communicate over CAN bus, where a node such as C generates and transfers data on CAN bus, nodes A and B receives the data. One of the unit acts as adversary, like node A which saves the message. Here the node B uses the message, when the CAN bus is ideal the node A will transmit the saved message. Now both B and C nodes receive the spoofed message.

**Implementation 2: with manipulation of spoofed message.**

Before attack is done by transmitting the same data frame sent by C. But the one has the same ID as C. Now we are changing the stored message and transmit the manipulated message over the CAN bus. Initially the node C will send the message and node A will save the message. Now the node A will manipulate the message by replacing the ID by its own or some other and transmit over the Can bus after some delay. Node C and B receives the data, node B have very less chances to detect, but there’s a chance for node C to detect it by check all the fields in the data frame.

**Mitigation:**

To mitigate this attack the node C, need to monitor the data and check the data frames it receives, it should check the contents in the frame for ID as it is a stored message which was originally transmitted by node C, now the node C can transmit the error message back and the reply attack can be avoided. Also, a check logic can be implemented to monitor the network and a time stamp can be implemented to avoid the reply to attacks.

**Still works:**

NO.

Paper 3: DEFENCE (reference3).

Attack detection with ML (supervised and unsupervised combo)

**Overview:**

Training a model based on previous data and experimenting on new cars to detect malicious content in CAN messages.

**Implementation:**

The benchmark dataset is collected from the cars KIA soul, Hyundai YF sonata and Chevrolet spark. The data set consists of CAN messages and three different attacks such as Flooding, Fuzzy and Malfunction.

Each CAN message is assigned to one of the four labels as shown in the figure.

A white rectangular box with black text

Description automatically generated

Figure 1. CAN message data.

In order to build feature vector CAN ID, time stamp, and contents of the messages are used.

The Hybrid model is evaluated against pretrained and retrained models which are lower and upper performance bounds.

The hybrid model is an anomaly detector with auto encoder which is combines with pretrained model and implemented using Pytorch.

To evaluate the Hybrid model 2 out of 3 cars data is used as source data sets and the 3rd car acted as target, the process is repeated and changed the target car in each iteration.

The pretrained model is made up of supervised machine learning techniques with mixed of attack and attack free data from the sources. The retrained model is made of attack free data from the target car. The hybrid is made of auto encoder and trained using attack free data from target car and merged with pretrained model.

After the training the models are tested against the target data which has everything included.

The hybrid model showed good results compared to other two, however detecting malfunctions was challenging due to the usage of normal data made the model too cautious.

A white rectangular box with black text

Description automatically generated

Figure 2. Results of detection (accuracy) by Pretrained, Retrained and Hybrid models.

Reference:

1. Practical DoS Attacks on Embedded Networks in Commercial Vehicles Subhojeet By Mukherjee1, Hossein Shirazi1, Indrakshi Ray1(B) , Jeremy Daily2, and Rose Gamble2.

url: https://www.springerprofessional.de/en/practical-dos-attacks-on-embedded-networks-in-commercial-vehicle/11246068

1. Implementation of Replay Attack in Controller Area Network Bus using Universal Verification Methodology Thirumavalavasethurayar P and Ravi T Department of Electronics and communication Sathyabama Institute of Science and Technology Chennai, India

url: <https://ieeexplore.ieee.org/document/9395871>

1. In-Vehicle Network Attack Detection Across Vehicle Models: A Supervised-Unsupervised Hybrid Approach

url: https://ieeexplore.ieee.org/document/9564896