How to Test Firewall via CAN Bus Traffic DB

1. Introduction

Purpose: using CAN traffic database to test the performance of connected-car firewall.

a. How to leverage CAN traffic database as the firewall testing feeds
b. Attack detection performance
c. White-listing false positive testing
d. Resource consuming performance testing
e. Firewall integration testing

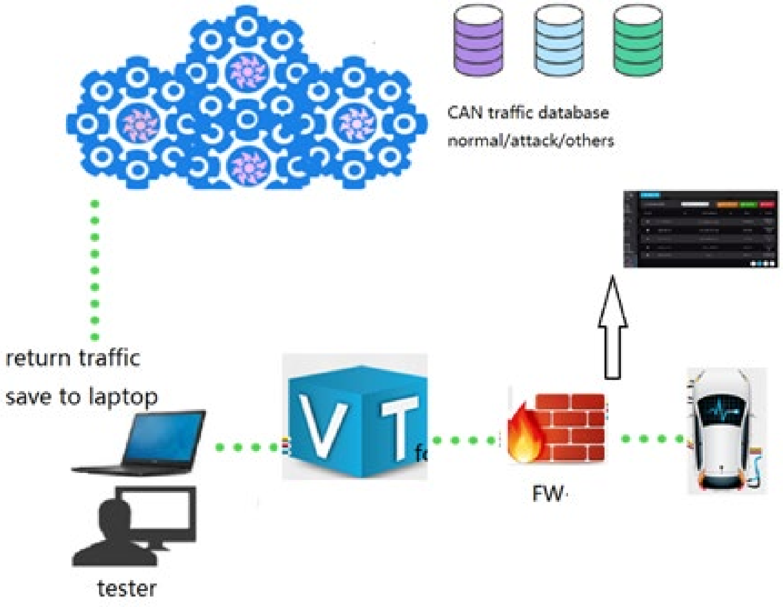

Fig.1 Firewall Testing Flowchart

Fig 1 shows the firewall testing process. The CAN traffic database is hosted in the cloud. Three database schemas represent the white-list/normal, attack/malicious, and others respectively. Based on the vehicle model to be tested, the tester makes a query to the cloud, where the testing traffic will be generated and can be downloaded as a zipped file. CAN traffic files from each schema will be fetched and tested onto the firewall.

A customized CAN traffic replayer plays the downloaded traffic towards the firewall. The firewall needs a log alerting function which can save the detection results into a log file. The tester feeds CAN traffic through the integrated firewall and outputs the testing results. The firewall should also have log function which can log the detected event and statistics. The firewall vendor should provide the integration manual of how to integrate firewall software into ECUs. For example, APIs interface can be used. The cloud firewall portal is preferred for firewall event management.

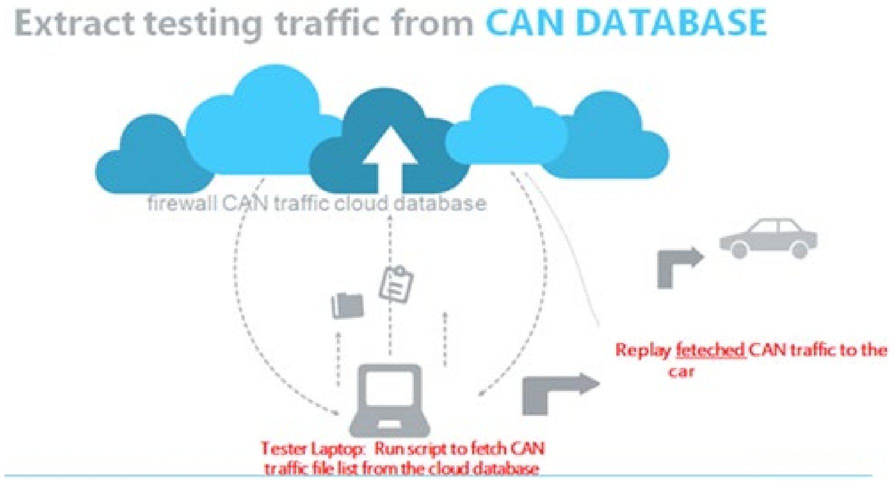
Extract testing traffic from CAN DATABASE
firewall CAN traffic cloud database
Replay feteched CAN traffic to the car
Tester Laptop: Run script to fetch CAN traffic file list from the cloud database

Fig. 2 CAN Bus Traffic DB Extraction

Fig. 2 shows the process to fetch testing traffic from the cloud. One database copy is hosted in the cloud. The database contains CAN traffic collected for each car model. We need implement search process to fetch specific testing traffic set for a specific firewall.

DB searching is used to search CAN traffic files from the raw traffic repository, with the results as a copy from the cloud to the local tester laptop. There are two approaches to implement that: 1) script-based and 2) web portal based. Here we use the second approach, web portal. As a benefit, the tester don’t need to familiar with the query scripts, instead, he/she only need to choose parameters on the webpage to make a query. The web portal should support “query by traffic MD5 hash”. Based on the inputted a MD5 hash, the system can return the corresponding traffic file.

2. FW testing traffic database

It is important to build and maintain the CAN traffic database. The database includes three traffic repositories: the normal (whitelist) traffic, malicious/suspicious (blacklist) traffic and other(customized) traffic data.

--- normal --- collected --- idle/engine off/driving

---- simulated – tools/scripts

---- third party – exchange/public

---- malicious --- attack simulation

--- ECU malfunction -- recoverable/non-recoverable

---- manipulation -- reverse message/ECU reverse/fool

---- third party -- remote test/hackathon/training

---- customized -- gray

--- ECU abnormal

2.1 Normal Traffic

The normal traffic database will take a long time to accumulate. The size will be huge. It is used to test the false positive of FW. The normal traffic database is composed of the following types:

a. normal traffic collection
We need drive different vehicles to collect CAN traffics under a number of conditions: idle, driving, local, highway, turn left/right or the combinations. The firewall should not trigger any alerts on whitelisting traffic. If have, that will be a false positive.

b. Simulated CAN traffic
Based on inter-arrival time, payloads, CAN ID, simulated CAN traffic are generated. They are normally used for performance testing.

c. Other available or public CAN traffic from third-party

2.2 Malicious Traffic

The malicious traffic database is composed of the followings:

a. Attack simulation traffic

Each simulated attack can be considered as a kind of attack. The traffic should include the testing name, target vehicle model, tester name.

b. Testing traffic submitted by third-party via VT APIs.

Third-parties may submit traffic generated by their scripts. We will store them into DB.

c. Reverse engineering or hacking traffic

These traffic are collected when people try to reverse engineering CAN traffic or hack a vehicle.

2.3 Other type of database

This type of database is customized for OEMs. It could be gray traffic (neither malicious nor normal) or ECU abnormal behaviors.

- Collected traffic of vehicle ECU malfunctions or DTC codes

When the vehicle shows the malfunctions or error codes, we need to collect CAN traffic. The traffic is different from the above one because it is not for attack but after-attack malfunction.

3. DB Schema

We implemented a simple portal for uploading traffic. Whenever uploading any traffic, once the cloud receives the traffic, it will automatic generate the hash value of the traffic file and save into the DB.

Filed name	**Value description**
Dataset name	
Traffic Size	
Hash value	MD5 value of the entire traffic data
Submitted date	
Target vehicle OEM/model	
Tester name	
Vehicle status	Idle/drive/engine off
Vehicle source	Dealership/purchase/donation/OEM
Category	Normal or attack
CAN Bus baud rate	
ECU mal-function ID	Which ECU IDs have vulnerability
ECU category comment	e.g. BCM, gateway, powertrain
Description	Testing comments

4. DB Searching

DB searching is used to search specific testing traffic from the raw traffic repository, with the results as a copy from the cloud to the local tester laptop.

There are two approaches to implement that: 1) script-based and 2) web portal based. The script-based solution runs as the standalone command script on the tester's laptop. The tester will run the script app, configure parameters and query a certain a traffic files. On the other hand, the web portal based solution is more automated by extend the current testing portal to support a query page. As a benefit, the tester don't need to familiar with the query scripts, instead, he/she only need to choose parameters on the webpage to make a query.

Search user case:

We will test a OEM vehicle's BCM, so we need search all traffic datasets related with BCM and attack simulation named "Bypass ECU SecurityAccess". In this case, we need search "BCM"in comment field and "SecurityAccess" in attack simulation category. Then based on the list, the script fetch all raw CAN traffic files from the repository and make a copy then send to the laptop.

Notes:

- The CAN traffic files save in the database should not be very large. In the scenarios of poor internet access, it will be hard to download a large amount of traffic data.

- The web portal should support "query by traffic MD5 hash". Based on the inputted a MD5 hash, the system can return the corresponding traffic file.

- The testing process and the traffic downloading can run at the same time. The tester can start the testing once the first traffic file got downloaded, no need to wait till the whole list of traffic files downloaded.