Diagnostic Traffic Analytics

## Revision History

|  |  |
| --- | --- |
| Revision | Description |
| 0.1 | Initial version |
| 0.2 |  |
| 0.3 |  |

## TODO

* Collect and Analyze more traffic files
* Add responses as well as requests
* Traffic example

# The goal

Extract various diagnostic CAN traffic traces based on a pre-defined normal diagnostic cases. Use these traces to test firewall for false positives.

# The general way of implementation

Use log service to get the requests and responses. Then save the results into traffic files as firewall testing traffic.

## Pre-defined diagnostic normal cases

The normal cases will be used to test firewall for false positives. CAN Bus traffic traces need to be extracted from diagnostic traffic. The normal cases include:

* *VIN scan*
* *ECU version reading*
* *DTC codes reading*
* *Clean DTC codes*
* *Successfully Security Access request*
* *Read memory data from ECU*
* *Read ECU identifier*
* *OTA ECU memory flushing*

## VIN scan

Send diagnostic VIN request packets to get VIN of vehicle. The whole process should include:

1. Request: request packet
2. response: response packet
3. packets including VIN: payloads of VIN info

*traffic example:*

## ECU version reading

Get the ECU version number, serial number, or vendor info. The whole process should include:

1. Request: request packet
2. Response (positive or negative):
3. Any security access requirement?
4. packets including ECU info: payloads of ECU info

*traffic example:*

## DTC code reading

To be added

### Normal Traffic Traces

|  |  |
| --- | --- |
| Normal Case Name | Traffic description |
| *VIN scan* | File name: size: traffic description |
| *ECU version reading* |  |
| To be added |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Attack scenarios:**

**CAN Bus system probing**

Purpose: scan CAN Bus for available CAN ID list

Attack method:

Scan range: a large range of CAN IDs

Service ID range: 0x10 diagnostic session control

Sub-function range: 0x01 default session

Request message: CAN ID - 0x02 0x10 0x01 \* \* \* \*

Response: positive 0x50

**ECU Service scan**

Purpose: find supported service list of the ECU IDs found above.

Attack method:

Service ID scan range: 0x00-0xff

Other bytes: all zeros

Request message: Supported CAN ID - 0x02 0x00 0x0 0x0 0x0 0x0 0x0 0x0

Supported CAN ID - 0x02 0x01 0x0 0x0 0x0 0x0 0x0 0x0

Supported CAN ID - 0x02 0x02 0x0 0x0 0x0 0x0 0x0 0x0

…..

Supported CAN ID - 0x02 0xff 0x0 0x0 0x0 0x0 0x0 0x0

Response: positive = service ID +0x40