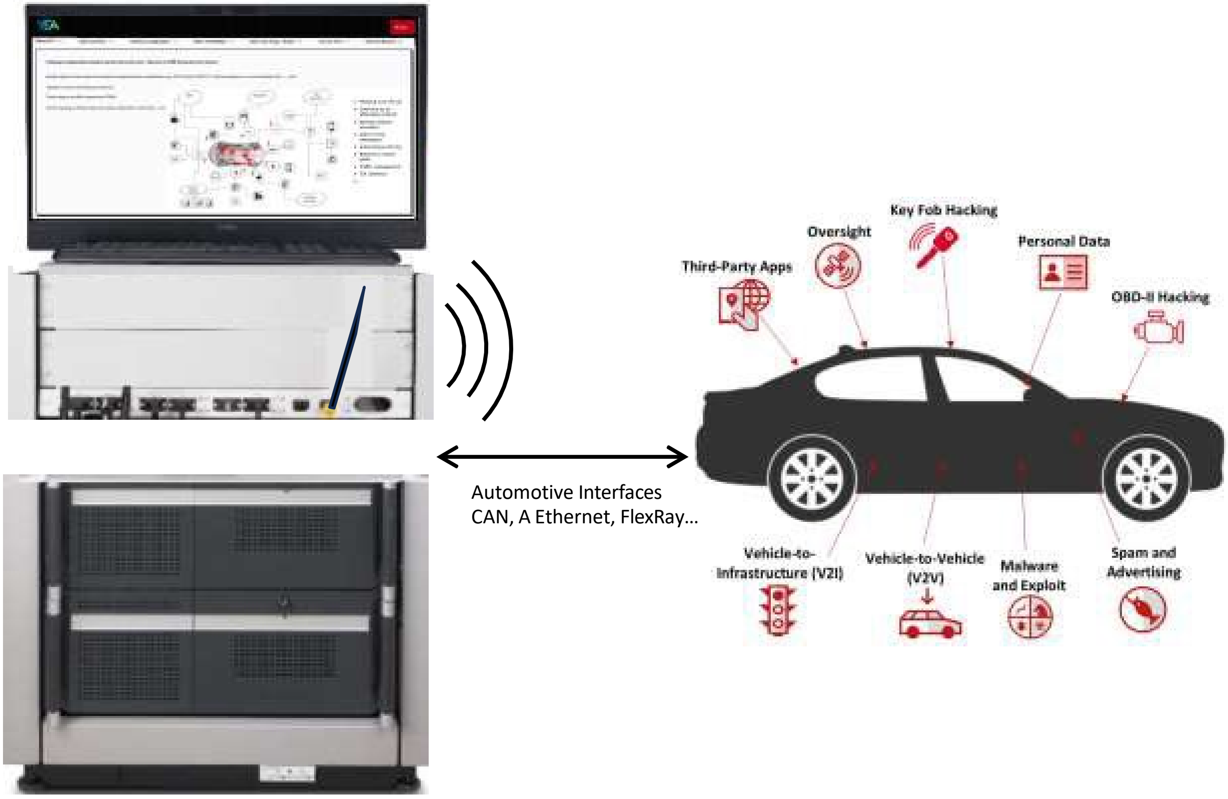
Automotive Cyber Security Test System

Datasheet

Introduction and Functional Overview

Modern vehicles have multiple connectivity features and functional blocks that make them vulnerable to external cyber-attacks.

The software environment of Automotive Cyber Security (ACS) Test System can simulate multiple kinds of cyber-attacks to test the resilience of the vehicle or its internal components to the external cyber-attacks.



Key Features

The ACS software has the following features:

- **Configurable Report Generation:** User configuration data can be entered in the User Interface to generate reports. Description of the System-under-Test, System topology, Interface types, Encryption Algorithm, CAN matrix and Test Reporting format according to ISO 2134, UNECE WP29, Threat Analysis and Risk Assessment (TARA), acceptable risk thresholds and many others are examples of acceptable user configuration data.
- **Sniff and Record:** The software can sniff the data and record it in a file upon request. It can also detect the specific payload data change when the user needs to know which payload data is responsible to a specific function in the vehicle (i.e., User wants to know which data is responsible for brake-request signal or Mirrors-folding signal.) so he can use it later for penetration test.
- **Replay:** Replaying the recorded log file with the same timestamps and send back to the SUT.
- **Man-In-the-Middle Attack:** Reads the data through the selected hardware interface and makes changes on the payload data through existing UI (i.e., adding Message Authentication Code (MAC) to the payload, or injecting fake signal in the data section) and resends it using the same or another physical interface.
- **Text-based Script Editor:** Writes user-defined scripts in Python or Linux Shell and test-sequences by the user according to the architecture of the System under Test (SUT) to perform functional security tests (i.e., AUTOSAR SecOC, Custom Crypto modules, Vendor specific functions)
- **Fuzzing:** Injects random and erratic frame headers and payload data to the System under Test (SUT) and checks the response.
- **Denial of Service attack:** Injects high priority CAN messages to prevent other modules in the vehicle being able to communicate.
- **Penetration Test:** Sends specific and structured fake data to the System under Test (SUT) and checks negative and positive responses (user visual inspection is needed).
- **Upgradable Data base:** Selects and stores well-known attacks and test scripts in library
- **Modular Interfaces:** Interacts with new hardware interface modules to execute cyber-attacks (i.e., WI-FI, Automotive Ethernet, FlexRay, Bluetooth, LTE) upon request.

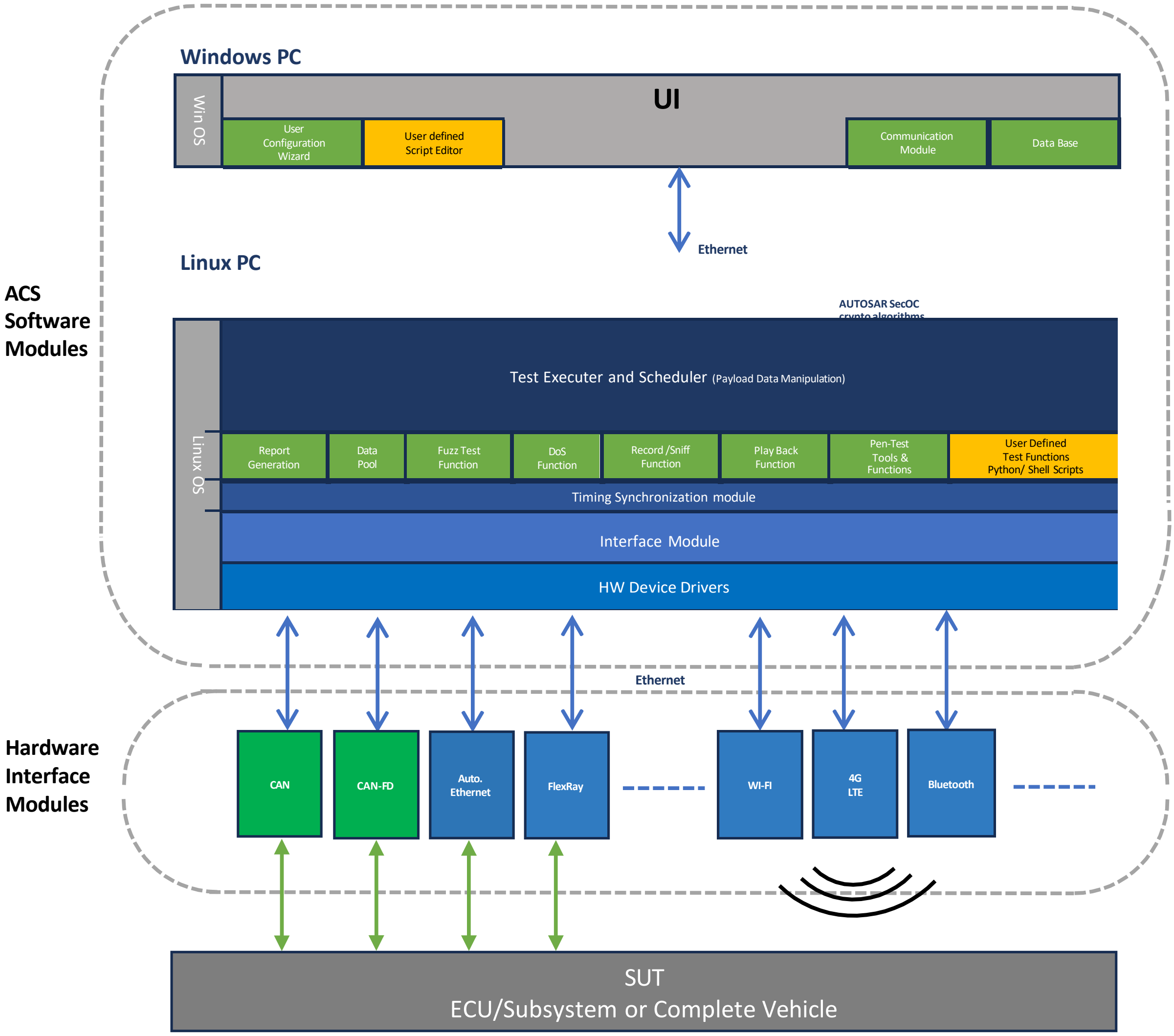
Additional Features

In addition to the above-mentioned features, ACS software can use off-the shelf attack and penetration tools (i.e., attack and penetration libraries in Linux).

For vendor-specific functional security modules, ACS software can provide a text-based script editor for implementing custom functions. These functions can be called by the Test Executer to get the required codes (Hash / Authentication) and send it via a physical interface to execute the attack.

High Level Architecture

The ACS Test System has the following components and functionalities:



- **User-Interface:** Installed on the Windows PC, its main functionality is to run the user configuration wizard and send the configuration data to the Linux machine via ethernet communication module, displaying status and test results.
- **Test Executer:** Runs on the Linux PC. Its main functionality is:

1. Receives all user configuration data through ethernet communication module.
2. Initializes both interface and report generation modules according to the user configuration.
3. Checks for selected attack methods and calls the proper implemented functions to run.
4. Schedules and synchronizes multiple functions-calling and attack-sequences for user defined scripts according to their timestamps.
5. Sends the ongoing software status through communication module to the UI.
6. Reads and stores physical interface traffic in a temporary log file before and after each attack execution and detects the SUT responses by comparing the files.
7. Opens a pop-up text window after each attack allowing for addition of operator’s visual inspection notes and comments.
8. Saves all test results mentioned in points (6.) and (7.) in the database module and deletes Temp files.
9. Generates Test Reports based on the User configuration and test results (8.)
10. Sends the Test reports from the Linux PC to the in Windows PC UI to be displayed on the UI.

User Guidelines

The user needs to follow the steps below to use the ACS Software

1. Configuring the software and report generation format by importing the System-under-Test specific data through operator interface (UI) (i.e., System topology, Interface types, Encryption Algorithm for cryptographic interfaces, CAN matrix, acceptable risk thresholds and Test Reporting format according to ISO 2134)
2. Select and configure the physical interfaces that should be connected to the vehicle or specific modules inside the vehicle.
3. Select and configure one of the mentioned attack methods in page 2 (i.e., Fuzzing, Pen-T, DoS.)

4. In case of the need to test a specific cryptographic module or authentication mechanism in the vehicle, the operator should write a vendor specific script inside the text-based script editor in Python or Linux shell to implement the Message Authentication Code (MAC) functions. The ACS software can then use the calculated MAC to manipulate the frame data for the selected interface and execute the penetration attack accordingly.
5. Execute the test/attack and watch the status on the UI. The visual detection by the user is needed to determine positive/negative responses of the SUT and add his comments to be used in the report generation.
6. Generate test report based on the configuration data imported in Step (1), SUT Feedback, as well as the visual user comment (5.).

Data Security

All vendor specific and user-defined written test scripts are stored locally inside the data base module in the local device storage.

Technical Specs

- CAN 2.0 interface module: Number of ports 2, Max speed 500kbps.
- CAN-FD interface module: Number of ports 3, Max speed up to 2Mbps.
- WI-FI interface module: HW 802.11 b/g/n, Max speed up to 300Mbps at 2.4Ghz. supports WEP, WPA, WPA2, TKIP and AES encryption.
- Bluetooth v5.0 Interface module: Distance up to 10m, transfer rate up to 2Mbps, compatible with all previous old versions 4.0, 3.0, 2.0 and 1.0
- Automotive Ethernet IP attack module: 2 ports 1000 BASE-T1 physical interface supporting UDP PTP protocols.
- 4G,5G LTE Interface module: LTE Release 12,13 and 14, up to 20Mhz Bandwidth, 5.9 Ghz frequency.

Note: Additional physical ports or custom interfaces can be added upon requirements.