D5.3 – Extended Modules of the Connected Vehicle System

SAFER TEC
SECURITY ASSURANCE FRAMEWORK FOR NETWORKED VEHICULAR TECHNOLOGY

Security Assurance Framework for Networked Vehicular Technology

Abstract

SAFERtec proposes a flexible and efficient assurance framework for security and trustworthiness of Connected Vehicles and Vehicle-to-I (V2I) communications aiming at improving the cyber-physical security ecosystem of "connected vehicles" in Europe. The project will deliver innovative techniques, development methods and testing models for efficient assurance of security, safety and data privacy of ICT related to Connected Vehicles and V2I systems, with increased connectivity of automotive ICT systems, consumer electronics technologies and telematics, services and integration with 3rd party components and applications. The cornerstone of SAFERtec is to make assurance of security, safety and privacy aspects for Connected Vehicles, measurable, visible and controllable by stakeholders and thus enhancing confidence and trust in Connected Vehicles.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

D5.3 & Title:	**D5.3 – Extended Modules of the Connected Vehicle System**
Work package:	WP5 - Assurance Framework Evaluation
Task:	T5.2 Simulation Based Evaluation and Extended Modules
Due Date:	February 2020
Dissemination Level:	PU
Deliverable Type:	R

Authoring and review process information	
EDITOR Andras Varadi / CMS	**DATE** 01/2/2020
CONTRIBUTORS Guillemette Massot / CCS Konstantinos Maliatsos / UPRC Sammy Haddad / Oppida Panagiotis Pantazopoulos / ICCS	**DATE** 15/02/2020 27/04/2020 30/03/2020 28/04/2020
REVIEWED BY Christos Lyvas / UPRC	**DATE** 29/04/2020
LEGAL & ETHICAL ISSUES COMMITTEE REVIEW REQUIRED?	
NO	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Document/Revision history

Version	Date	Partner	Description
V0.1	26/02/2020	CMS	First draft
V0.2.1	30/02/2020	CMS, OPP, CRF, CCS, ICCS	Stable draft
V0.3	26/04/2020	UPRC	Inputs to Section 3
V0.4	28/04/2020	CMS	Internal review version
V0.5	29/04/2020	UPRC	Review, Minor Edits
V0.6	06/05/2020	OPP, ICCS	Comments in all sections
V1.0	11/05/2020	CMS	final version submitted

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Figures

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Acronyms and abbreviations

Abbreviation	Description
ASN.1	Abstract Syntax Notation One
C2C-CC	Car2Car Communication Consortium
CA	Certificate Authority
CAM	Cooperative Awareness Message
CC	Common Criteria
CRL	Certificate Revocation List
CTL	Certificate Trust List
CU	Communication Unit
CVE	Common Vulnerabilities and Exposures
CVS	Connected Vehicle System
D	Deliverable
DENM	Decentralized Environment Notification Message
DoS	Denial of Service
ETSI	European Telecommunications Standards Institute
HMI	Human Machine Interfaced
HSM	Hardware Security Module
ITS	Intelligent Transportation Systems
ITS-S	ITS Station
IVN	In Vehicle Network
LDM	Local Dynamic Map
LLC	Logical Link Control
LVI	Local Vehicle Information
MAC	Medium Access Control
MAP	MAP is the intersection geometry message name as defined by ETSI TS 103 301
OBU	On Board Unit
PKI	Public Key Infrastructure

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

PP	Protection Profile
RSU	Road Side Unit
SAF	SAFERtec Assurance Framework
SFR	Security Functional Requirement
SPaT	Signal Phase and Timing
T	Task
TOE	Target of Evaluation
V2X	Vehicle to everything
V-ITS-S	Vehicle – ITS Station
WP	Work Package

Table 1: List of Abbreviations

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Executive Summary

This deliverable provides evidence of the ability of the SAF framework to adapt to system or context changes. Based on the first evaluations feedback presented in D5.2 the document describes the upgrades made to the Connected Vehicle System (CVS, chapter 2) and the upgrades, methodologies and high-level results of the SAF framework and particular testing methods (chapter 3).

Apart from the general updates on the bench, the risk analysis changes led to the extensions of the security requirement:

- Identification and authentication of users on the HMI: This change was required in order to avoid unauthorized access to the AppOBU (through the HMI).
- Misbehaviour detections functions: The second requirement extension was made to cover the case where an authenticated external source is not to be trusted and thus such data is verified despite its sender being a trusted source.

The second part of the document - following the same approach as D5.2 – describes attacks that have been simulated in a close to real environment to evaluate the impact of the added modules validated by SAF. By the obtained analysis of the overall security, the SAF capabilities to adapt to risk analysis changes are shown. Results suggest that the earlier identified attack vectors and their associated residual vulnerabilities were not present and exploitable anymore.

Overall, the document collects proof on how systems can adapt to results and that the SAF can change easily its evaluation objectives, being able to easily verify them. It concludes that the SAF can adapt easily to a changing security context, meaning the change of Security Target's (ST) scope, new security objectives validation can be added or required assurance level can also be increased to deal with more sensitive use cases or more hostile environment.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1. Introduction

1.1 Purpose of the Document

This document presents the outcome of task T5.2 which is related to extended modules.

Relying on the performed simulation-based evaluation exploring additional test cases and additional configuration attributes (i.e. with respect to attacks, threats and security controls), the results in this document prove how well the proposed assurance framework fits under certain changing conditions and reveals its sensitivity to capture the assurance levels with respect to the simulated conditions. It also introduces new modules or extensions of the Connected Vehicle System that adds or modifies specific features. Changes made help identify how the SAFERtec Assurance framework reacts and also shows the adaptability and granularity of the framework and its methods, techniques and tools developed.

1.1 Intended readership

Besides the project reviewers, this deliverable is addressed to any interested reader (i.e. Public dissemination level).

1.2 Inputs from other projects

This deliverable does not use any inputs from other projects.

1.3 Relationship with other SAFERtec deliverables

This document is coupled with D5.2 (Simulation Based Evaluation of SAFERtec Assurance Framework) the other result of T5.2 and presents the outcome of the task with respect to the simulation-based evaluation.

Based on the D5.2 observations which identified limits in the first set of security requirements, the risk analysis performed and validated by the STs defined in the task 3.3 made some assumption about the security of the user devices and how critical data manipulation would be.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2. Upgrading the Connected Vehicle System

The following chapter presents the additional test cases and additional configuration attributes (i.e. with respect to attacks, threats and security controls) that have been applied during T5.2. The changes proposed and introduced intend to prove how well the proposed assurance framework can adapt itself under certain changing conditions.

The first round of testing by the SAFERtec Assurance Framework have identified a set of vulnerabilities or weak points in the Connected Vehicle System. These results have been identified by the respective partners and changes have been developed in order to avoid or mitigate the vulnerabilities.

As already mentioned, the first simulation tests done for the D5.2 helped us to identify

- Vulnerabilities in the scope of the security targets defined in D3.3
    - Those were not treated due to project resource limitations and objectives, but are to be corrected in the case of a full SAF execution.
- Vulnerabilities in the scope of the TOEs of D3.3 but not in the scope of the STs
    - Some residual vulnerabilities have been identified in the TOE
        - Vulnerabilities exploitable outside the scope of the STs assumptions and the evaluation attacker's profile
- Vulnerabilities of elements outside the scope of the STs

Concerning the first set of vulnerabilities, it is actually beyond the project scope and objectives to treat them. They will not provide any further information on the SAF and while costly to address it would require large amount of efforts that would not help to assess SAF benefits nor limitations. However, let's remind here again that in the context of a full execution of SAF, evaluation iteration would be performed until those vulnerabilities are corrected, or the evaluation aborted. Those vulnerabilities are identified as necessary improvements that are eventually required in the SAF context.

For the second and third sets of identified vulnerabilities, they helped to see and validate the limits chosen (STs scopes) for the first evaluation made in the context of the T3.3 (D3.3 first STs). The associated risks were accepted in the first place, but for more sensitive use cases those are not acceptable. The risks were identified by the risk analysis of D2.3 and validated by the D5.2 work. This is why we have updated the STs and done a second round of SAF assessment in D3.3.

In this deliverable we evaluate the impact of the STs increased scopes and the associated extended modules and systems updates that follows.

In the following sub-section, we first recall the findings extracted from D5.2 together with the SAF adaptation and STs updates to adapt to the new risk analysis. Then we present the bench modules extensions and configuration updates that address those new security requirements.

They will be tested by performing black box vulnerability tests of the CVS, together with the complete SAF adaptation to this new security assurance scope to further validate SAF

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The new version of the bench and SAFERtec product upgrades tested in that deliverable contains:

- updated model configuration on existing components
- new software module features

Changes have also been presented in updated STs and ToE documentation (T3.3). The following section intends to provide a summary of the applied changes between the initial and the upgraded version, furthermore explain their effect on component and system behaviour (e.g. impact on security functionality)

2.1 Updates of existing components

2.1.1 API updates on the VBOX and the OBU platform.

Assessing the outcome of the verification, the V2X docker image (implemented by Commsignia) received a configuration update that closed remaining open ports that were unnecessary. The update included the removal of legacy support (support for older standard functions) which was not used within the use cases. This change did not require any software to be updated or upgraded but was restricted to configuration.

In a similar fashion, possible vulnerabilities have been resolved on the V2X software stack running on the V2X Onboard Unit hardware/processors (implemented by Autotalks). The changes on this platform only required configuration modifications in order to achieve the desired result. API changes included the restriction of mid layer payload access which was not required for the use case functionality.

Further changes on the V2X OBU are described at the end of this chapter which introduce new functionalities.

2.1.2 Other updates of the VCS

At first, a short reminder of the VCS structure (presented in full details in D4.2 and 4.3) is presented.

The bench prototype is composed of: (i) the vehicle, i.e. a mobile ITS station equipped with communication technologies; (ii) the road-side ITS Station (R-ITS-S), that is the fixed infrastructure having V2X communication capability to interact with the vehicle and wired connectivity towards the cloud; and (iii) cloud services providing information concerning traffic events and management.

The vehicle prototype of the bench is characterized by the following main hardware and software components: (i) the in-vehicle CAN network hosting the vehicle sensor signals (e.g., speed, acceleration); (ii) the CAN gateway connecting different areas of the CAN network; (iii) the application on board unit (APP OBU), that hosts the project application use-cases; (iv) the V2X board (V2X OBU) that sends/receives ETSI ITS G5 messages; (v) the network gateway that provides the cellular connectivity to the vehicle, the in-vehicle Wi-Fi to connect the HMI devices to the APP OBU, and an Ethernet in-vehicle network that connects vehicle components such as V2X OBU and APP OBU; and (vi) the in-vehicle HMI devices, which for the SAFERtec testbench is a tablet.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Additional details and documentation about the bench architecture, components and functionality have been reported in the project deliverable in D4.2 “Modules and Applications of Connected Vehicle” and D4.3 “Integration of Connected Vehicle System”).

The following modifications have been applied to the software modules of the vehicle bench prototype in order to follow the recommendations of the penetration test teams for fixing security issues emerged during the penetration testing activity. The changes were implemented by CRF.

In the initial bench version, the road-side ITS Station was functional only when actual physical GPS signal. This fact created some issues during the initial tests. Hence, the performed update consisted in allowing the GPSD service of the R-ITS-S station (i.e., the service that captures and interprets the GPS signal) in working with a dual configuration: (i) by using the actual physical GPS signal received by the antenna of the device; and (ii) by using a simulated pre-registered GPS signal thanks to the use of the GPSfake software module, i.e., a service that pass to the GPSD service a recorder or generated GPS signal. This allowed the bench be completely functional even in the absence of the actual physical GPS signal.

The docker image of the V2X OBU was also modified according to the penetration test results with proper changes in the configuration files. Thus, uploading of files was disabled (with the exception of specific folders), the UDP inject API was disabled and the corresponding UDP port was blocked by the OS firewall.

The in-vehicle network gateway device was subject to some issues; thus, the following modifications have been made to apply the penetration test recommendations.

- The firmware of the network gateway underlying the software providing the connectivity functionality has been updated to the last version, available at the time of the tests. The first pre-installed version of such a firmware was subject to some know security threads fixed by the gateway developers in the latest firmware version (i.e., the latest available at the time of the tests). Therefore, the new firmware version released by the gateway developers has been installed in the network gateway of the bench aiming at eliminating the open issue identified and recognized by the component developers.
- Some services provided by the network gateway (i.e., SSH, Web UI based on simple http – i.e., that does not use protected communication, Command Line Interface, and Modbus) have been disabled since no access is necessary in order to reproduce the project use-cases. Instead, other services used for the reproduction of the use-cases or for administrative purpose has been preserved. For instance, the Web user interface over HTTPS has been preserved open for administration operations on the network gateway.
- The password of the administrator user to access to the Web user interface for administration operations of the network gateway has been changed to follow more restrictive indication on the password creation, thus making it difficult to be discovered by, e.g., brute force attacks.
- The port scan possibility has been disabled for preventing scanning and monitoring activities on the gateway opened ports by external entities.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

The in-vehicle application on board unit (APP OBU) was also subjected to some issues, thus the following modifications have been made to apply the recommendations. The APP OBU is hosted on an on-board unit (actually an embedded PC based on Linux OS) that contains all the software modules and applications that implement the project use-cases.

- The world readable access to some sensible files of the Linux OS has been removed, thus preventing other entities to read such files while not explicitly allowed.
- Linux OS kernel has been updated to the last version available at the time of tests, thus patching all known issues and security threads.

2.2 HMI application vulnerabilities addressed

The SAFERtec Android HMI application was developed for the vehicle's head-unit/tablet to support the communication with the vehicle services, implemented as Docker images, in the APP OBU as well as with the V2X OBU (see relevant sections in D4.1 and D4.2).

The application was designed to communicate with the underlying services and displays information to the driver, such as speed advice, traffic warnings and priority notifications, according to what is required in each use-case.

As marked in D3.3 the initial implementation of the HMI application had no authentication mechanism. Essentially, the initial security context (i.e., reproducing day 1 applications with low automation level) and the relevant environment's security requirements allowed the HMI application to operate without such a security control.

As highlighted by the SAF application and the updated Security Target, this turned-out to be a vulnerable point, especially when considered under a higher level of automation. As such a login mechanism was added to the application relying on the operation of a Keycloak providing the authentication service.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

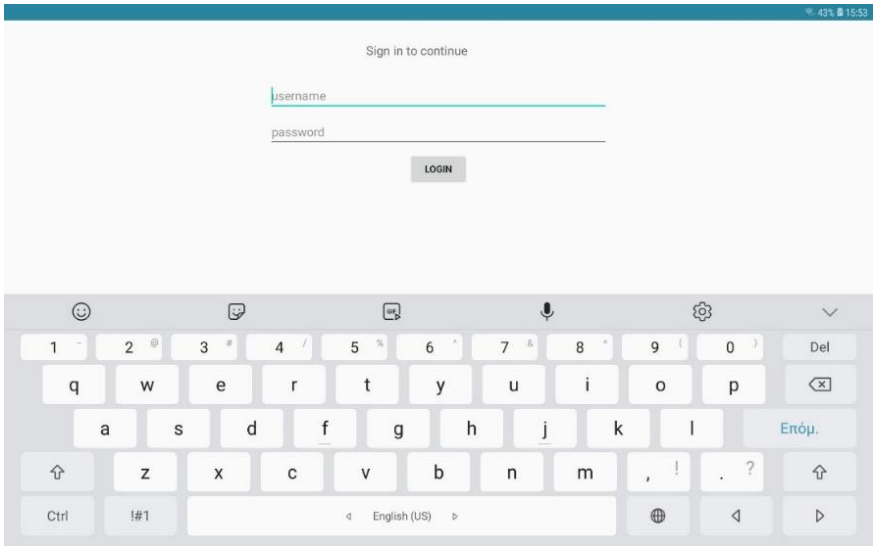


Figure 1 Log-in screen requiring credentials (after the HMI app update)

The addition of the relevant functionality (see Figure 1) as a result of the application of the proposed framework constitutes a validation that SAF can efficiently handle potential updates of the security environment, as captured by ST changes.

2.3 Misbehaviour detection module (MBD)

A major change on the V2X protocol stack running on the vehicle V2X OBU was the addition of misbehavior detection functionality. This module ads an additional protection against threats identified by the project.

Data that come from an external ITS station in form of V2X messages that are signed with a valid certificate and public key may still be considered as a threat as that external entity could be compromised or faulty. To protect against such threats, the misbehavior detection module acts as a step in the processing chain to validate payload at different stages and checks data that otherwise can be considered as standard compliant and trusted.

To protect the in-vehicle system, the MBD module verified data via plausibility checks (e.g. checks speed and the change of reported geo location, signal strength vs reported distance).

A detailed specification, developed by Commsignia, can be found in D3.3.

2.4 Outcome of the update of the Connected Vehicle Platform

With the updates listed in this chapter, the CVS has been improved so that the overall system is protected against the threats identified by the SAF.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3. SAFERtec Assurance Framework evaluation results comparison

The content of the section bares some comments since it is shared up to a significant extent with the confidential SAFERtec deliverable D3.3 content. The work presented here is a result of strenuous experimentation which has exhaustively studied the relevant MD challenges; its completeness essentially makes it span across both D3.3 and D5.3 needs.

Our choice to (partially) reuse here what has been already placed in D3.3 is justified by the great research interest in testing the misbehaviour detection layer and most importantly the fact that our results use simulation means and rely on the actual implementation over the SAFERtec test bench. In fact, the normal approach in CC evaluation would have required the evaluator to make tests in constrained time and technical resources (defined by the target assurance level) which in the case of ITS testing tools state-of-the-art would have been rather limited. Since part of the SAF enhancement is to develop such tools the efforts for simulation environment for misbehaviour tests have been used for the two tasks and thus are fully reused here where it was initially expected.

This section presents a work with considerable innovation that further points to a strong standardization potential (see D7.5). Consequently, it is on purpose that we include again (a large part of) the tests made in the confidential D3.3 in a public document.

In what follows the results of the SAFERtec Assurance Framework assessment of the upgraded Connected Vehicle System is detailed.

3.1 Radio-related validation/penetration tests

UPRC received an implementation of the SAFERtec connected vehicle test bench in the middle of February, thus a set of validation and penetration tests was performed at the final stages of the project. The specific tests had the objective to:

- implement a specific set of attacks.
- apply the attacks on the SAFERtec bench.
- monitor the behaviour of the system.
- validate the existence of security controls that are capable to repel the attacks.
- validate the fulfilment of specific security functional requirements specified by the SAFERtec protection profile (presented in D3.3).

More specifically, the specific set of tests included the simulated/emulated implementation of the following attacks:

- Replay attack: It is a form of network attack (in the SAFERtec case over the radio) in which a valid data transmission is maliciously or fraudulently repeated or delayed.
- Malformed frame attack: The attacker injects malformed frames into the network by violating protocol rules.
- Misbehaviour Detection - False position claim: A legitimate user through valid transmissions claims that its position is different than the real.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Misbehaviour Detection – Sudden position change: A legitimate user (ITS station) through valid transmission claims that it performs irregular/irrational movements that cannot be physically justified in order to confuse other users.

It is noted that for the tests:

- the Target of Evaluation was the Communication Unit (V2X OBU), i.e. the cyber-physical system composed by:
  - the hardware component (implemented by Autotalks) [1].
  - the radio stack and the ITS stack implemented by Autotalks and Commsignia respectively (both SAFERtec partners).
- During the tests, the penetration test team was in communication and coordination with the development teams from the OEM partners in order to gain access to all information and data needed for the completion of the tests. Besides the direct communication between teams, the input used included:
  - The tutorial for the hardware setup of the connected vehicle system bench.
  - The tutorial for the software setup of the connected vehicle system bench – as well as the description of the SAFERtec Use Case implementation on the bench.
  - Instructions and application notes from the development team for the update and the configuration of the ITS kernel of the V2X OBU.
- All tests/attacks were applied through the ITS-G5 radio interface. Thus, the radio interface and the implemented network protocol stack was put under evaluation. Other interfaces (e.g. ethernet, or CAN) of the in-vehicle network (IVN) are assumed reliable and secure.

3.1.1 Description of the UPRC test setup

In order to perform the tests, the following equipment was used:

- The SAFERtec connected vehicle test bench – fully deployed – with the following main components:
  - The V2X OBU hardware – the hardware that hosts the target of evaluation for the specific set of tests;
  - The Vehicle BOX (the vehicle computer) that coordinates all system components and host system applications and services;
  - The Network router and gateway that implements the in-vehicle ethernet network and provides the gateway for the internet. It also operates as an in-vehicle WiFi access point.
  - The Power-over-Ethernet switch that is used in order to power the Roadside Unit and also provides an ethernet port, allowing an external PC to connect to the in-vehicle network.
  - The Roadside Unit (R-ITS) and the respective hosting software and ITS components.
  - A conventional tablet hosting the HMI application.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

  - The CAN Gateway router implementing the in-vehicle CAN. The specific component was not used in the tests – however, it was installed in order to ensure that all applications and services operate as expected by the SAFERtec bench design team.
  - A USRP B210 software defined radio transceiver used to provide the GNSS signal to the bench simulating a moving vehicle with the use of real recorded GPS signals.
- A second software defined radio (SDR) transceiver (both USRP B210 and USRP B200 were used for the tests) that are utilized as the means to deploy the attacks upon the V2X OBU. Software-defined radio is a radio communication system with components implemented by means of software. Thus, components of the radio system that are traditionally implemented in hardware and application-specific integrated circuits can be tested with proper coding. Usually, the code and the radio control application are hosted on a personal computer connected through a high-speed interface with the SDR board.
- Two personal computers (laptops):
  - The one is operating as a host of the radio communication software executed through the SDR board. Generally, the SDR host is a high-performance PC able to handle and support real-time processing of radio waveforms in wideband transmission rates (in the ITS-G5 case, 10 MHz).
  - The second is operating as a monitoring device for the operation of the bench. It is connected with the in-vehicle network through the Power-over-Ethernet switch (conventional ports – non PoE ports) and sequentially with the IVN router. The specific PC is used to connect through SSH with all three main components of the experiments (V2X-OBU, VBOX, R-ITS-S) in order to trigger the proper scenarios, and collect the log-files that are used to verify the applicability and functionality of the evaluated security measures.
  - Two antennas, one for transmission and one for reception, for the SDR board, in order to implement the radio attacks.
  - Proper RF cables.
  - One variable attenuator. Practically, a 40dB attenuator was used serially with a variable attenuator (0~30 dB with an 1dB step). The attenuators were used in order to directly connect the USRP with another radio device (e.g. the R-ITS-S) in order to facilitate the experiment. More details will be provided in the following sections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The software components that are used during the penetration test experiments are the following:

- The V2X-OBU ITS and radio stack hosted by the V2X-OBU, which is the actual target of evaluation for the experiments. It implements the ITS-G5 interconnection; it supports the following types of ITS messages:
  - Cooperative Awareness Messages (CAM)
  - Decentralized Environmental Notification *Message*s (DENM - not used in the specific set of experiments)
  - Signal Phase and Timing messages (SPaT – transmitted from the roadside unit in order to provide a status of the traffic lights in the intersection).
  - Intersection Map data messages (MAP – transmitted from the roadside unit in order to provide information of the intersection topologies, e.g. lanes etc.)
- The R-ITS-S ITS software stack, implementing the functionalities of the roadside unit, hosted on the RSU hardware. During the experiments, the test team was not able to intervene on the R-ITS-S, thus, despite the fact that the roadside messages were used in the attacks, no modification or customization was performed on the specific components. It is noted, that a GPSfake daemon can be activated on the R-ITS-S, in case the equipment is located indoors with no GNSS coverage. Since GNSS data are necessary for the operation of the roadside unit, the GPSfake daemon is used as the means to perform the specific tests.
- The GNSS replay message code, which is executed on the VBOX (implemented by CRF). As an SDR implementation, the GNSS waveforms are recorded in files hosted in the VBOX. The code loads the specific files (representing a route in Trento, Italy) and transmits it through the Tx/Rx RF chain which is connected directly with the GNSS input of V2X-OBU. The execution of the SDR code is initialized through scripts that activate specific use cases.
- A set of scripts – all activated through a main script (in this case start_bench.sh) executed on the VBOX. The script role is to identify and initialize the various components, test and validate their functional health, initialize and coordinate the code execution based on the SAFERtec use case that is appropriate for the purposes of the penetration test.
- The SDR code that attacks the ITS-G5 radio interface and implements the specific threat. For each of the following types of attacks/threats a customized program was developed by UPRC based on the objective of the executed test. Description of the SDR program for each case is provided in the following paragraphs. The SDR code was developed in C++ using the Open Source USRP Hardware Driver (UHD [2]) in order to drive transmission/reception for the SDR board. The program is executed on the SDR host personal computer. In order to run the specific code, the Ubuntu Linux 16.04 operating system and UHD 3.14 (or above) should be installed on the host.
- A GNUradio [3] implementation of the IEEE 802.11p physical layer and medium access control sublayer, that was developed by WIME project[4] and it is provided as open source[5]. More details, on its use for the specific set of tests are provided in paragraph 0.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.1.2 Replay Attack

Objective of the Test:

- To validate that the V2X-OBU provides security controls in order to protect the ITS system from replay and wormhole attacks.
- To evaluate the fulfilment of a set of Security Functional Requirements described in the Communication Unit Protection Profile [6] (details in D3.3).

Security Measure:

The V2X-OBU developers have added a replay attack identifier module. Each packet is examined through its time stamp, message id and sequential number in order to check if it has been received before.

If so, the packet is rejected and a possible replay attack attempt is identified.

Activation of the security measure:

The replay attack protection mechanism is part of the standard security services installed and offered by the V2X-OBU. Thus, it is part of the initial implementation of the ITS stack and it is by default activated. In order to deactivate the replay attack, the security services of the V2X-OBU should be disabled.

Implementation of the replay attack:

The scenario is the following:

- The SAFERtec use case 1 is considered. In this use case, the V2X-OBU transmits CAMs, while the R-ITS-S broadcasts SPaT and MAP messages.
- Since, V2X-OBU is the “victim”, then the replay attacks are performed from the R-ITS-S to the V2X-OBU. More specifically, a “Man in the Middle” setup is established, where the malicious repeater captures the SPaT and MAP messages from the R-ITS-S and replays them.

The role of the malicious repeater is played by the SDR transceiver. When activated, the replay attack program is executed with the task to catch, amplify and forward the messages of the R-ITS-S. The replay attack program developed by UPRC operates as follows:

- The SDR board is assumed to be close to the Roadside Unit and as distant as possible from the V2X-OBU in the laboratory environment.
- The SDR board is programmed to operate as a transceiver:
  - Both Tx and Rx are operating in the 5.9GHz carrier frequency, that is also used by the R-ITS-S and the V2X-OBU.
  - The receiver and transmitter sampling rate are set to 10 MHz, as the ITS-G5 bandwidth.
  - The RF Tx gain is adjusted (through test and trial) to a value that ensures reception from the V2X-OBU without saturating the receiver with power.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

- The SDR program "amplifies and forwards" the ITS message – it does not demodulate it or decode it. This is done because:
  - Implementation of a full IEEE802.11p physical layer will introduce processing delay and thus, there will be increased time difference between the original and the replayed message. However, it is desired to have small time difference between the original and the replayed message.
  - Implementation of a fill IEEE802.11p physical layer will significantly increase computational workload for the SDR code host, that may not be able to handle it properly, leading to an irregular and unstable operation.
- The SDR uses three main input parameters:
  - Threshold θ_1, which is a power threshold that is used from the receiver as an energy detector that identifies a transmission. When the power at the receiver is estimated above θ_1, then an ITS packet is exchanged: a) a CAM is broadcasted from V2X-OBU; b) a SPaT or a MAP is broadcasted by the R-ITS-S. The threshold θ_1 was set to (approximately) -70dBm, which was sufficient for the given laboratory conditions in order to achieve an almost 100% probability of detection and 0% probability of false alarm.
  - Threshold θ_2, which is a power threshold that is used to distinguish messages originating from the Roadside unit and the V2X-OBU. The value depends heavily on the experiment setup in the laboratory. As mentioned before, the SDR board is positioned close to the roadside unit – and as far as possible from the V2X-OBU. This means that the received power of messages originating from the R-ITS-S are expected to have significantly higher power. If threshold θ_2 is exceeded, then it is assumed that the specific transmission is coming from the roadside unit.
  - Delay *d*, which is a time specification in milliseconds that defines the time lag between the detection of the message and the attempt for retransmission. Delay *d* may vary from 50 μsec to several seconds.
- The SDR program operation can be summarized as follows:
  1. The receiver scans the 5.9GHz band.
  2. When the received signal power exceeds θ_1, then an ITS signal transmission attempt is detected.
  3. If the received signal power also exceeds θ_2, then it is concluded that the message originates from the R-ITS-S. This activates the replay procedure and this sets SDR clock time to zero.
  4. The received waveform is stored into a variable, and the SDR transmitter is activated and scheduled to attempt transmission in *d* milliseconds. The content of the variable of the received waveform is fed to the transmitter.
  5. Steps 3 and 4 are repeated until the end of the transmission, which is identified by the fact that the received signal falls below θ_1.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The main problem of the procedure is the uncertainty in the distinction of messages from the R-ITS-S or the V2X-OBU. In order to bypass this problem, the configuration presented in Figure 2 is used. In the specific configuration, we exploit the diversity feature of the Roadside unit. More particularly:

- One of the two R-ITS-S RF chains feeds an antenna. Thus, the message is transmitted only through a single antenna. It is noted that for the laboratory setup, diversity is not necessary since message reception is ensured due to the close proximity between the nodes.
- The second R-ITS-S is connected through cable with the Rx RF chain of the SDR board. Thus, the SDR receiver cannot sense signals over-the-air. It only receives messages from the roadside unit. An attenuator is introduced between the R-ITS-S and the SDR board in order to protect the SDR Rx RF chain from excessive input power. As a conclusion, a single threshold θ_1 can be used in order to detect messages.
- When the messages are received, the signal is forwarded (without decoding it) to the SDR board Tx antenna and it is transmitted after *d* milliseconds. It is noted that *d* should be greater than the duration of the original frame, since otherwise the replayed message will cause a collision (acting as a jammer and not as a replay attack node).

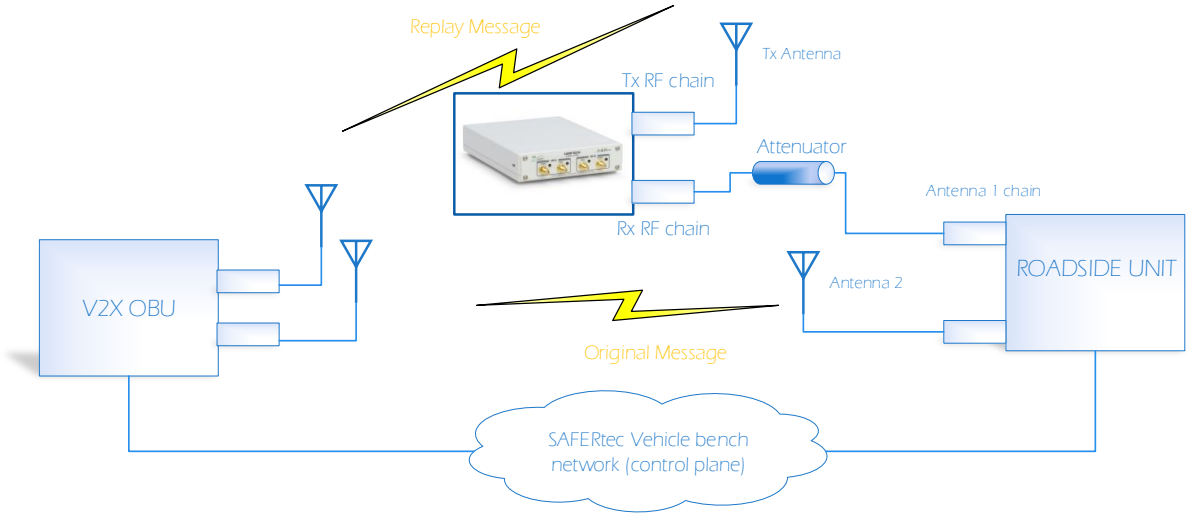


Figure 2: Replay attack experiment configuration

In Figure 3, the measured signal spectrum as captured by the SDR board and visualized by the SDR program is presented. The subfigures clearly depict the technical characteristics of the signal waveform (10 MHz Orthogonal Frequency Division Multiplexing power spectral density), as well as the noise floor during the experiments. For the specific tests, over-the-air operation was performed (no direct cable connection between the R-ITS-S and SDR board was installed.

In the upper subfigure, the signal is transmitted by the R-ITS-S, which is clearly more powerful than the signal of the lower subfigure coming from the V2X-OBU, for the specific laboratory configuration.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Observation of the result can help us visually estimate the thresholds θ_1 and θ_2 (in dBm/Hz) that can be used for signal sensing and signal distinction.

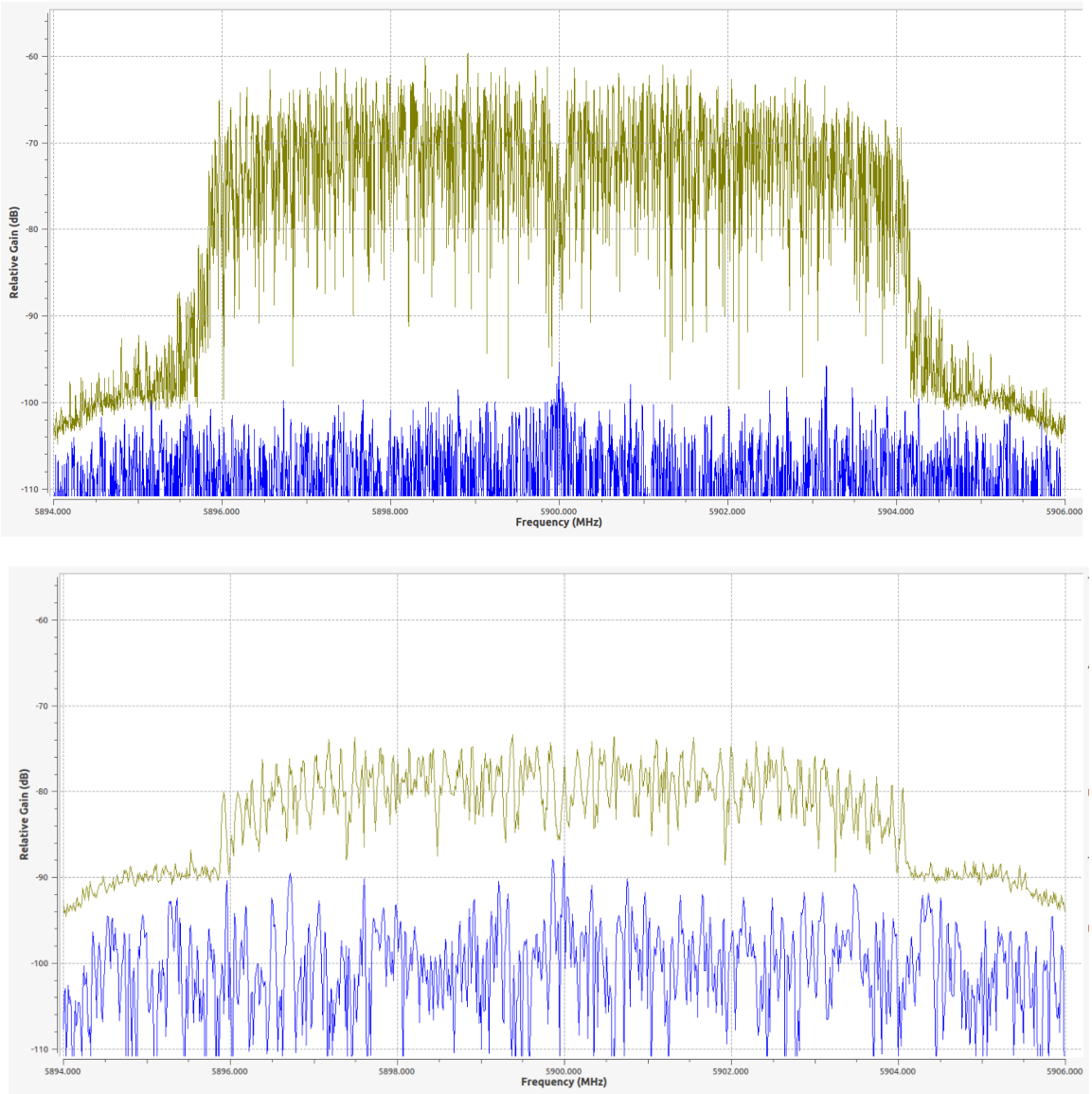


Figure 3: Spectrum of captured signal originating a) from the R-ITS-S and b) from V2X-OBU

In Figure 4, a visual representation of the SDR replay attack module operation is provided. Packets are arriving in the receiving RF chain. When the threshold is exceeded, the signal is "copied" from the receiver to the transmitter with a delay of d milliseconds and the signal is replayed. As a result, the streams depicted in Figure 4 for both RF chains will eventually arrive at the target of evaluation. In the specific example, the time domain signal contains two messages: A SPaT and a MAP from the roadside unit.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

If V2X-OBU has an implemented security control for replay attacks, then only the red-colored packets (from the original source) will be kept and processed, while the replays will be dropped.

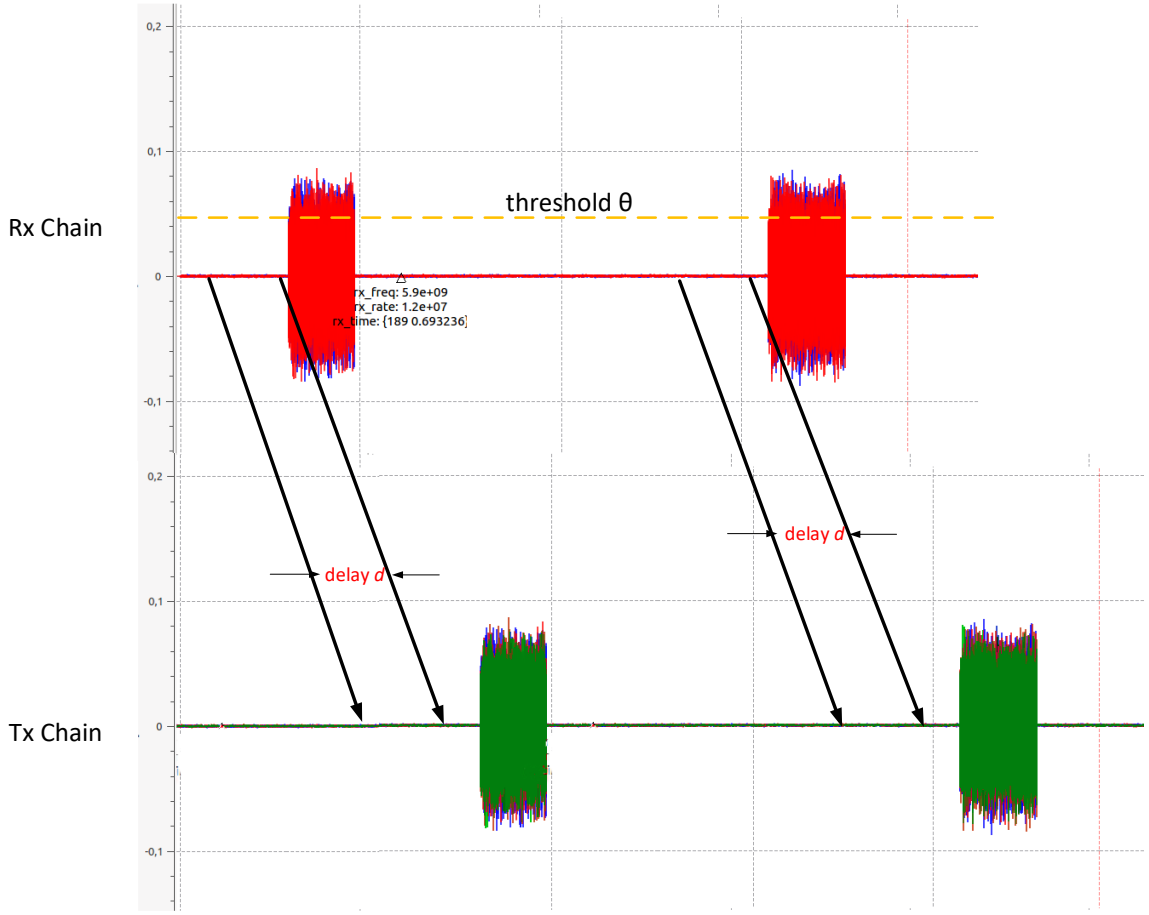


Figure 4: Signals in the time domain a) Rx RF Chain, b) Tx RF chain. Visual presentation of the functionality implemented by the replay attack SDR module.

Conclusion:

It is verified that the V2X-OBU has a functional and effective security control that copes with replay attacks.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.1.3 Malformed Frames

Objectives of the tests:

- To validate that the V2X-OBU provides security controls in order to protect the ITS system from malformed frames, i.e. against the following threats described in the Communication Unit Protection Profile as T.ITS Data Manipulation and T. Malformed Frame Injection.
- To evaluate the fulfilment of specific Security Functional Requirements described in the Communication Unit Protection Profile [6] (presented in D3.3).

Security Measure:

The V2X-OBU developers have added an integrity check module. Each packet is examined through hash checks and CRC checks using the digital signatures and authorization tickets as reference.

If a received packet does not pass the respective CRC and hash integrity checks, then the message is rejected.

Activation of the security measure:

The integrity protection mechanism is part of the standard security services installed and offered by the V2X-OBU. In order to deactivate the protection mechanism, the security services of the V2X-OBU should be disabled.

Implementation of the malformed frame attack:

The scenario is the following:

- The SAFERtec use case 1 is considered. In this use case, the V2X-OBU transmits CAMs, while the R-ITS-S broadcasts SPaT and MAP messages.
- A “Man-in-the-Middle” intercepts the SPaT and MAP message. Then, by intervening in the content of the message, the malicious user changes a set of bits. For example, in the specific test, the “Man-in-the-Middle” tries to alter: the packet header suggesting that the message is longer than expected; the message content is altered by changing timing and phase messages.
- The message is transmitted from the Man-in-the-Middle towards the V2X-OBU.

In this case, and in contrast with the previous tests, the Man-in-the-Middle has to demodulate and decode the message. In order to do that, the open-source SDR program proposed in [7] and provided in [5] was used. There, an open-source SDR-based IEEE 802.11p transceiver can be found, implemented in GNURadio (programmed with C++ and python). The flow diagram of the implemented radio transmitter, as depicted in the GNURadio companion is presented in Figure 5.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

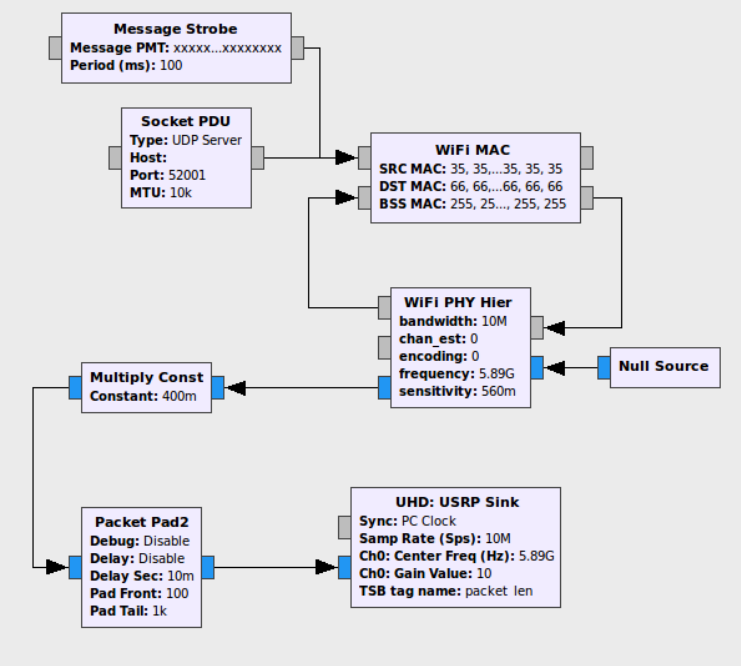


Figure 5: GNURadio open implementation of the IEEE802.11p [5].

The SDR implementation was tested with the real-world messages generated by the R-ITS-S and the V2X-OBU and it successfully managed to demodulate and decode it. It is noted that the open-source program implements the physical layer and does not apply the full ITS stack. In Figure 6, the constellation of received signal for a high-order modulation example is presented.

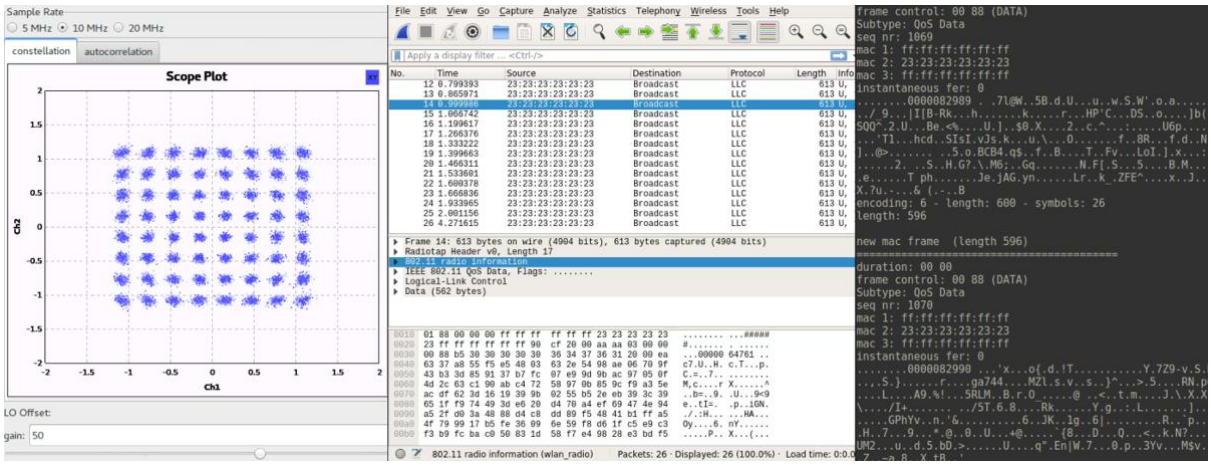


Figure 6: Example of the open-source IEEE802.11p receiver operation [5]

Complementary, an additional function implemented by UPRC is added, where the symbols of the received waveform are altered by changing the fields that contain the packet length (making it greater) and the timing and phase data. The changes were made in respect with the definition of the standards/recommendations [8] and [9].

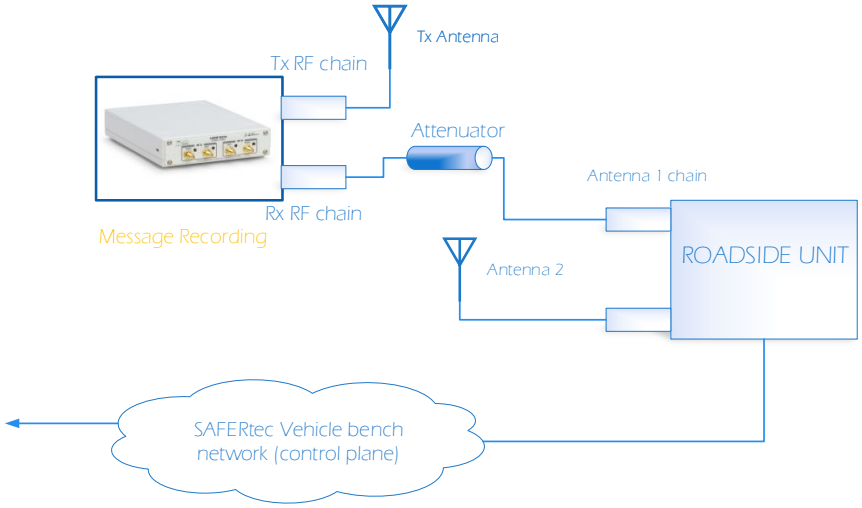
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

However, during the initial tests it was concluded that it was difficult for the SDR transceiver to follow in real-time the operation of the bench implementation. The SDR host needed some extra processing time – leading to missing packets and a non-tractable or observable test. Therefore, a different approach was used (see Figure 7):

1. In the first stage, the UC1 is executed normally. However, the SDR board records the whole procedure and stores all the messages and timings from the R-ITS-S into a file.
2. In an intermediate stage, the signals are demodulated and then the packets are re-generated with the random change of a specific set of bits.
3. In the second stage, the UC1 is executed once again. However, the actual R-ITS-S is deactivated (practically, the antennas were removed). As a replacement, the Man-in-the-Middle transmits the regenerated stream of packets, that are however malformed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

STAGE 1: Run UC1 1st time



STAGE 2: Run UC1 2nd time

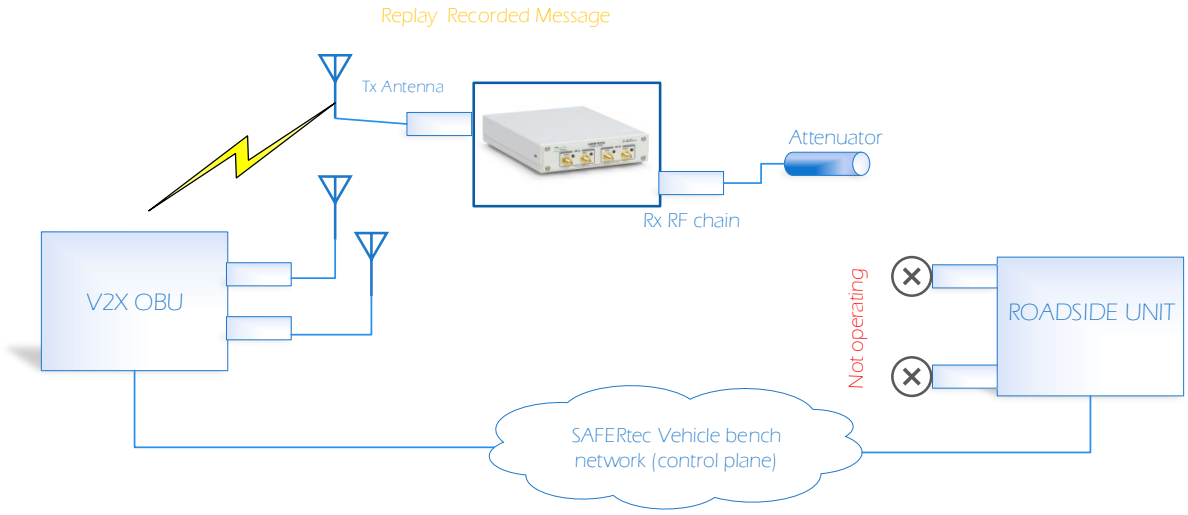


Figure 7: The two-stage experimentation procedure

Conclusion:

It is verified that the V2X-OBU has a functional and effective security control that copes with malicious frame injection.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.1.4 Misbehaviour Detection – The Acceptance Range Threshold

Conventional security assurance and testing does not include misbehaviour detection features. For SAFERtec, misbehaviour detection became part of its objectives during its last year, since: globally, significant research effort was spent for the specification of misbehaviour detection techniques; standardization activities are performed in order to describe a security control mechanism able to detect and prevent misbehavior; any security assurance attempt for connected vehicles without misbehavior detection is incomplete; the inclusion of misbehavior detection techniques into the assurance framework will constitute a significant novelty.

Therefore, new security functional requirements were introduced in the SAFERtec modular protection profile [9]. However, due to the limited time and resources, the development team was not able to incorporate in the system the complete set of plausibility checks defined in the misbehavior detection information flow policy. Nevertheless, Commsignia has provided updates that implement and integrate into the V2X-OBU two of the plausibility checks. In this paragraph, as well as in paragraph 0

However, it should also be noted that due to lack of time and resources, no systematic way in the determination of the thresholds (like the methodologies provided in [10] or [11]) was followed. The used thresholds were extracted by the development team through test and trials.

Objectives of the tests:

- To validate that the V2X-OBU provides security controls in order to protect the ITS system from misbehaviour detection after failing a plausibility check – that correlates claimed position with received signal strength, i.e. against the following threats described in the Communication Unit Protection Profile: Data Manipulation, False Reporting, Extreme Solicitation.
- To evaluate the fulfilment of specific Security Functional Requirements described in the Communication Unit Protection Profile [6] (results presented in D3.3).

The specific validation tasks were performed for the following rule of the misbehaviour detection information flow policy:

> *Acceptance Range Threshold*
>
> *The CU should check if the claimed position of the ITS message source is close to the position of the V-ITS-S. The CU should thus determine a reception area, which is the assumed area from which the messages could have originated. If a position claim is received that is outside the area, it is an indication that this claim is false. It is a simple threshold-based detector on the distance d from which a received claim position is accepted. Thus, a claimed position at distance d is accepted if $d < \theta$ where θ is a threshold defined by taking into account:*
>
> - *The radio propagation environment.*
> - *Acclaimed radio propagation models*
> - *Localization errors and uncertainty.*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

A margin should also be considered in order to minimize possibility of false alarm.

Security Measure:

The V2X-OBU developers have added an Acceptance Range Threshold plausibility check. Each packet is cross-examined and a plausibility check is performed in order to verify that the claimed position justifies the specific received signal strength.

If a received packet does not pass the plausibility check, then the message is not taken into account.

The algorithm checks the RSSI (Received Signal Strength Indicator) values measured during the transmission and compares them to the distance calculated using the coordinates of the communicating vehicles. Using this method, the misbehaving vehicles, which transmit false coordinates in the CAM messages, could be detected.

The RSSI value is depending on the distance between the transceivers, therefore it is possible to estimate the distances from the measured RSSI values.

The algorithm would take into consideration two different scenarios:

- The RSSI value is less, than would be expected based on distance
- The RSSI value is more, than would be expected based on distance

The distance of both scenarios is calculated from the coordinates of the receiving vehicle and the coordinates received in a CAM message.

The algorithm is only valid if the devices have a clear line of sight, because walls and other distractions highly modify the measured RSSI values.

Activation of the security measure:

The initial binaries/ITS software stack that was loaded in the V2X-OBU did not support the RSSI-based plausibility check for misbehaviour detection. Thus, in order to perform tests without the specific security control, the initial binaries were backed up and used on demand.

The security measures were activated with the use of updated executable binaries that were copied onto the V2X-OBU board. Documentation has been integrated into the binaries and was used in order to gain insight in the operation of the binaries.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Implementation of the misbehaving vehicle experiment:

Two binaries were provided by the V2X ITS Stack development team (Commsignia):

- The first binary (misbehaviour receiver) acts as the protected receiver, i.e. it applies the plausibility check at every incoming CAM message.
- The second binary (misbehaving transmitter) implements the attacker, i.e. it is the source that implements the attack.

However, in order to implement the experiment, two V2X-OBU units are needed. In our case, this was not possible, since only one V2X-OBU per bench is available. In order to bypass, this restriction, the SDR board is used. The experiment was performed in the following steps (Figure 8):

1. The misbehaviour transmitter starts sending data. The SDR board records the transmitted messages in order to emulate its behaviour. In order to reduce uncertainty introduced by the radio channel, the SDR board and the V2X-OBU (running the misbehaving transmitter binary) are connected through cable. The attenuation value is set to low (e.g. 40 dB), so that no significant distortion is introduced in the recording procedure.
2. At a second step, the binary with the receiver that also implements the acceptance range threshold plausibility check is executed on the V2X-OBU. Simultaneously, the recorded signal is transmitted emulating the misbehaving transmitter. The two modules claim to have similar position (lower than 50m). However, the attenuator between SDR board and V2X-OBU is set to a high value. More specifically, the attenuator was set to 80dB providing a signal level around -90dBm (in conjunction with the attenuation introduced at stage 1).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

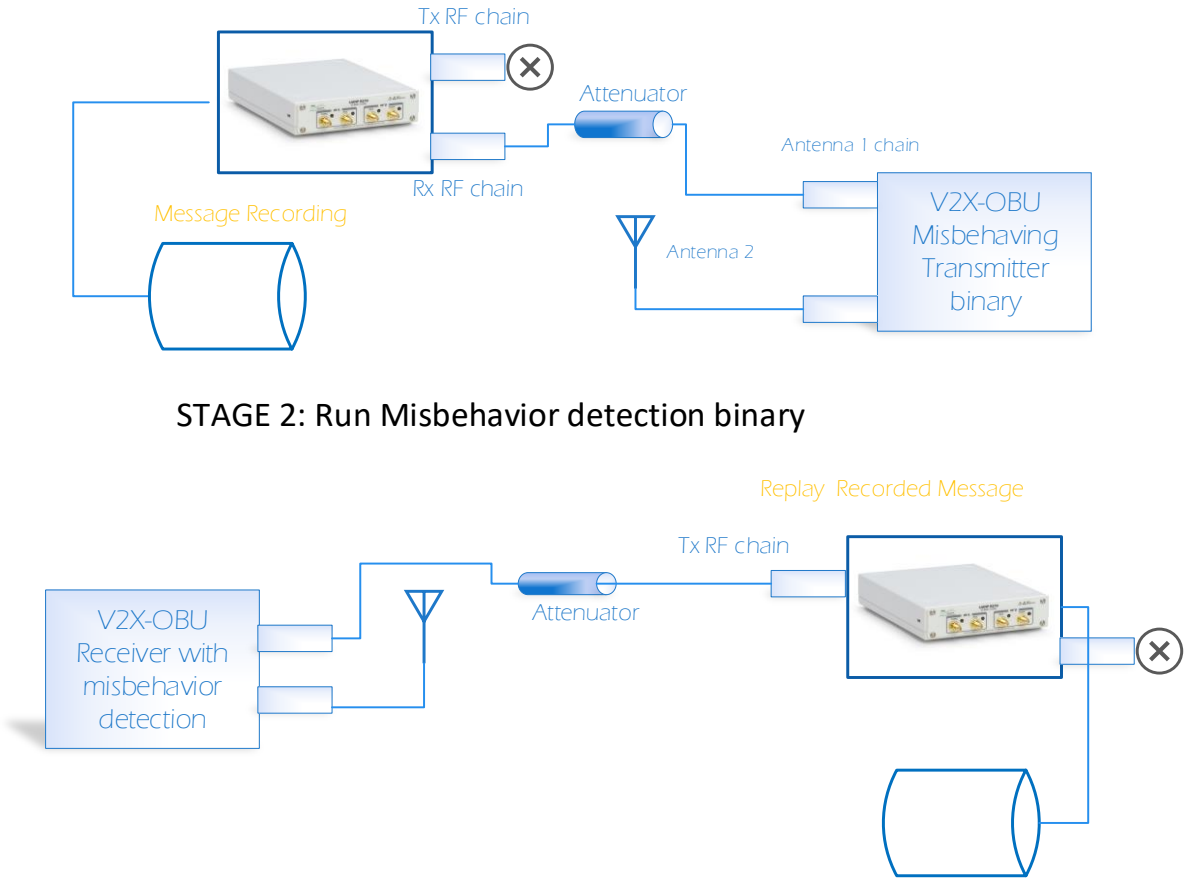


Figure 8: Misbehavior detection experiment setup

Conclusion:

It is verified that the V2X-OBU uses a plausibility check that correlates distance with received signal power in order to identify possible misbehaviour. The use of dynamic and flexible thresholds will improve the performance of the detector.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.1.5 Misbehaviour Detection – Distance Moved Verifier

Objectives of the tests:

- To validate that the V2X-OBU provides security controls in order to protect the ITS system from misbehaviour detection after failing a plausibility check – that checks the consistency in the mobility of the remote nodes (if the claimed distance variations in the reported path are justifiable by the laws of physics) , i.e. against the following threats described in the Communication Unit Protection Profile: Data Manipulation, False Reporting, and Extreme Solicitation.
- To evaluate the fulfillment of specific Security Functional Requirements described in the Communication Unit Protection Profile [6] (results presented in D3.3).

The specific validation tasks were performed for the following rule of the misbehaviour detection information flow policy:

> *Distance moved verifier:*
> *The distance moved verifier validates whether the target has moved a threshold distance ϑ based on its previous claim. As an example of the DMV detector, a vehicle moving with a similar speed with the V-ITS-S claims to have the same position at both t_0 and t_1, while in reality it has moved.*
>
> *The detector tolerance should take into account the traffic conditions and the breaking capabilities of the vehicle.*

Security Measure:

The V2X-OBU developers have added a Distance Moved Verifier plausibility check. Each packet is cross-examined and a plausibility check is performed using the reported path history in order to verify that the distance difference between consecutive packets can be justified. If a received packet does not pass the plausibility check, then the message is not taken into account.

The distance moved verifier or *speed validator misbehaviour module* checks the CAM messages in order to decide whether the vehicle's path history is corrupted or not. The algorithm detects the distance hops in the past, using the path history received within the CAM messages.

The speed calculation using distance and time parameters from the CAM's PathHistory data frame is performed as follows:

All values are scaled SI with a good enough precision to represent all possible values. Notably:

- Latitude / Longitude: 0.1 microdegrees
- DeltaLatitude / DeltaLongitude: 0.1 microdegrees
- PathDeltaTime: 0.01 second

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The module requires a speed parameter which is calculated form the following:

$$v = \frac{\text{deltaDistance(from newest PathHistory PathPoint)}}{\text{PathDeltaTime(from newest PathHistory PathPoint)}}$$

The speed is given in m/s. Calculating the velocity values from the whole path history is also possible, but it is considered as a future work for the development team.

In order to decide whether the path history has been corrupted, this v should be compared to the maximum speed of vehicles according to standardization. This value is 163.82 m/s, which is 589.752 km/h.

At this point, it should be noted that the threshold is quite high and unrealistic, since in most cases the conventional vehicle speed cannot exceed e.g. 220 km/h – without taking into account speed limits and traffic conditions. However, the specific threshold was set as definite threshold. According to [10], the possible outcomes are:

- The vehicle is misbehaving and claims to move in speeds above 163.82 m/s – therefore it is detected by the plausibility check module.
- The vehicle is misbehaving and claims to move in speeds irregularly high (not compatible with current traffic, speed limits or car model), however, it is not detected, since the claim speed is under 163.82 m/s. The process for the definition of a robust and accurate algorithm with adaptive threshold is considered as future work.
- The vehicle is not misbehaving and, thus the algorithm accepts the incoming messages as valid.
- The vehicle is not misbehaving but due to an uncertainty, the plausibility check detects the node as suspicious. Nevertheless, due to the extremely high threshold value, the probability of false alarm during tests was 0.

In order to exclude the possibility of errors in connection with reverse driving, the absolute value of speed must always be used.

Activation of the security measure:

The initial binaries/ITS software stack that was loaded in the V2X-OBU did not support the speed validator misbehaviour module plausibility check. Thus, in order to perform tests without the specific security control, the initial binaries were backed up and used on demand.

The security measures were activated with the use of updated executable binaries that were copied onto the V2X-OBU board. Documentation has been integrated into the binaries and was used.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Implementation of the misbehaving vehicle experiment:

Two binaries were provided by the V2X-ITS stack development team:

- The first binary (misbehaviour receiver) acts as the protected receiver, i.e. it applies the plausibility check at every incoming CAM message.
- The second binary (misbehaving transmitter) implements the attacker, i.e. an assumed vehicle that performs "jumps" on its claimed route.

Once again, in order to perform a complete test, two V2X-OBU units are needed. Since, the test bench has a single V2X-OBU, the architecture of Figure 8 was once again used, with the use of the SDR board as auxiliary hardware. The two stages were:

1. With the use of the misbehaving transmitter binary, and proper configuration according to the experiment goals, the transmission process is recorded into a file (duration of 5-10minutes) with the attenuation value set to low (e.g. 40 dB.)
2. At the second stage, the updated binary is executed in the V2X-OBU while the USRP replays the recorder misbehaving packet flow. It is noted that in this case, the attenuation value is set once again to low, since we do not want to trigger a reaction from the plausibility check.

During the tests, the malicious user generates and transmits a false message every third message. This practically means that, one out of three path points that are recorded in its path history is erroneous.

Following the same testing pattern as before, two tests were performed:

Behaviour with the use of the initial binary for the V2X-OBU

Since there is no plausibility check in place, the V2X-OBU will accept all incoming CAM messages as valid.

Behavior with the use of the updated binary for the V2X-OBU

With the use of the updated binary, the V2X-OBU estimates the speed of the remote vehicle and identifies irregular jumps.
The functionality of the plausibility is verified by the following means:

- The updated binaries produce a standard output message when a misbehavior is detected.
- Investigation of the summary log files indicate that the number of received CAMs is 2x times the number of dropped CAMs, which is consistent with the fact that an erroneous message is produced every three transmitted CAMs from the malicious source.

Conclusion:

It is verified that the V2X-OBU uses a plausibility check that cross checks the path history in order to identify irregular/unnatural behaviour. The use of more strict, robust and adaptive thresholds is necessary in order to improve performance.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.2 Authentication bypass tests

The following tests have be run to tests access to the internal vehicle network through user interface.

Tested interface HMI android application	
Tools used: None	
Initial conditions: • Attacker has physical access HMI's device and try to interact with TOE.	
Test scenario:	
Step 1	**Attacker switch on or unlock the HMI's device.**
Step 2	**Attacker tries to escape login page by clicking everywhere on screen.**
Step 3	**Attacker tries to escape login page by clicking on Android back button.**
Step 4	**Attacker tries to escape login page by pushing application to background (android home button).**
Step 5	**Attacker tries to escape login page by killing application**
Observed results	
For **step 1**, a login page appears after application version 1.2. This login page v1.2 succeeds to pass the **step 2**. However, **step 3 with** version 1.2 fails because android back button allows attacker to return on main "activity" for 1 second. This time is enough for interacting with the TOE. A new version (1.3) of the App was released in order to pass the **step 3**. This new version deactivates the back button. **Step 4 :** with last correction (1.3), the app fails this step. Pushing application to background kill the login "activity". Only main "activity" remains. So, when application returns foreground, attacker is able to interact with TOE. A new version (1.4) was released to correct this last point. Now Login "activity" remains even in background. **Step 5**: killing application and restarting it does not allow attacker to escape login page. Application pass this test.	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

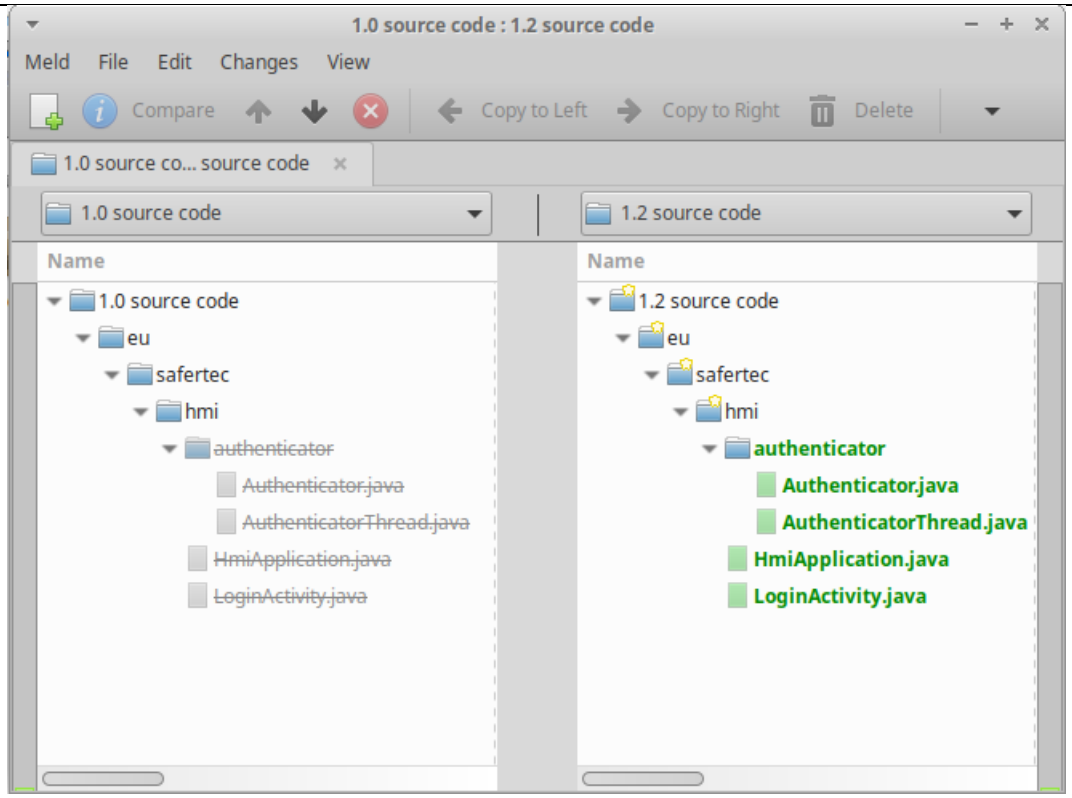


Figure 9 : Source difference between version 1.0 and 1.2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

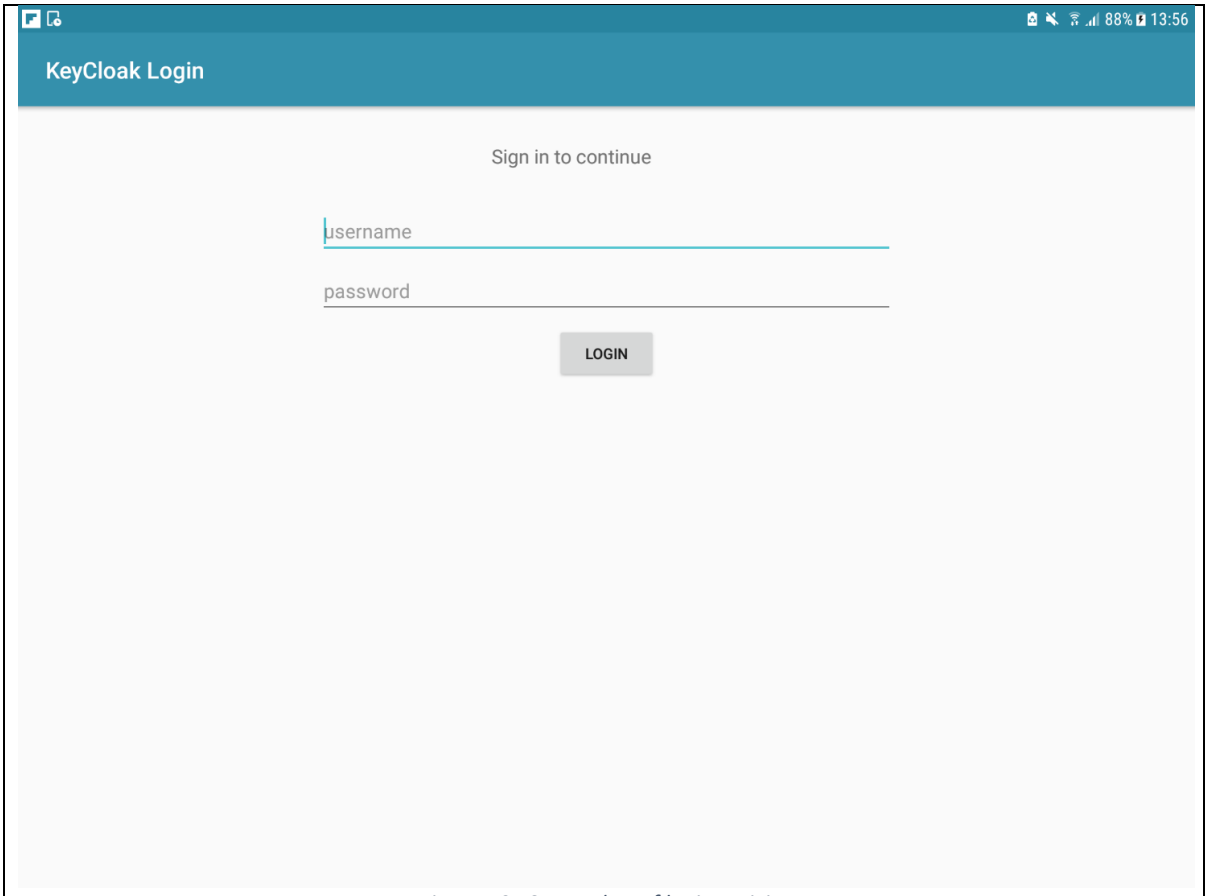


Figure 10 : Screenshot of login activity

Conclusions
Application was updated from version 1.2 to 1.4 in order to pass all the steps. All vulnerabilities are now considered as corrected.

Table 2: HMI android application results

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4. Conclusions

Tests presented in chapter 3 demonstrate that the SAF can change easily its evaluation objectives. As shown, ST updates can be done easily and allow for the definition of new evaluation objectives to cover new threats. Furthermore, it is shown that the evaluation is able to validate these new security properties i.e., in the present tests adding of Identification and authentication of users on the HMI to avoid unauthorized access and Misbehaviour detections functions to cover the case where an authenticated external source is not to be trusted and thus such data is verified despite its sender being a trusted source.

The new vulnerability tests show that the earlier identified attack vectors and their associated residual vulnerabilities were not present and exploitable anymore. This means that the SAF is able to adapt to a changing security context (shaped by objectives and requirements). It can cope with changes of the STs scope by providing new security objectives validation to deal with new threats or raise the required assurance level in view of more sensitive use cases and more hostile environments.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

5. References

[1] Tutorial for the Hardware setup of the SAFERtec connected vehicle bench – part of SAFERtec D4.2

[2] USRP Hardware Drivers and USRP Manual https://files.ettus.com/manual/

[3] https://www.gnuradio.org/

[4] Wireless Measurement and Experimentation, the WIME project https://www.wime-project.net.

[5] https://github.com/bastibl/gr-ieee802-11

[6] H2020-SAFERtec Communication Unit/Protocol Control Module Protection Profile https://www.safertec-project.eu/publications/modular-pp/

[7] Bastian Bloessl, Michele Segata, Christoph Sommer and Falko Dressler, "Performance Assessment of IEEE 802.11p with an Open Source SDR-based Prototype," IEEE Transactions on Mobile Computing, vol. 17 (5), pp. 1162-1175, May 2018.

[8] Signal Phase and Time and Map Data https://amsterdamgroup.mett.nl/downloads/handlerdownloadfiles.ashx?idnv=500795

[9] 802.11p-2010 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments https://standards.ieee.org/standard/802_11p-2010.html

[10] SAFERtec D3.3

[11] SAFERtec D5.2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319