2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)

Vulnerability Analysis of an Automotive Infotainment System's WIFI Capability

Edwin Franco Myloth Josephlal and Sridhar Adepu
iTrust, Center for Research in Cyber Security
Singapore University of Technology and Design
Email: edwin_franco@sutd.edu.sg and adepu_sridhar@mymail.sutd.edu.sg

***Abstract*—Automobiles of the current era are heavily computerized which makes them highly susceptible to attacks that were unheard of with traditional automobiles. In the past with traditional automobiles, one would require physical access to the automobile to compromise it. Today however, computerization has allowed remote accessibility of automobiles. Remote compromise is feasible by utilizing a vast range of attack vectors such as mechanics tools, automotive infotainment system, Bluetooth and cellular radios etc. In addition, wireless communication channels have made it possible to have long distance vehicle control, location tracking, in-cabin audio ex filtration etc. One of the electronic components in a modern automobile is its automotive infotainment system. This paper focuses on identifying the vulnerabilities of the automotive infotainment system with respect to its WIFI capabilities by conducting structured vulnerability tests on the WIFI capabilities of an automotive infotainment system. To do this, we analysed the WIFI attack surface and constructed test environments and used appropriate tools such as (Nmap (open port scan), Nessus (vulnerability scan), Metasploit) to generate a penetration testing plan to search for vulnerabilities. The vulnerability findings are well documented in this paper.**

***Index Terms*—Automotive infotainment system; Critical Infrastructure; Cyber Physical Systems; automobile security; Industrial Control System; cyber attacks; cyber-physical attacks; Android; Nmap; Nessus; Metasploit**

I. INTRODUCTION

People nowadays spend an extensive amount of their time on mobile phones. They are so accustomed to the wonderful features offered by their mobile phones that many expect the same with other devices including their car infotainment system. Hence there has been immense pressure on the car infotainment industry to develop car infotainment systems [5] that have similar features with mobile phones. The infotainment system provides a wide range of services such as AM/FM radio, music playback, video, hands-free communication with the mobile phone, navigation, internet access etc. Automotive infotainment systems originated with simple car audio systems that consisted of radios and cassette/CD players and have now evolved to include other features such as navigation systems, video playback, Bluetooth and WIFI connectivity, USB and SD card input and internet accessibility.

One of the features that customers want is the ability to customize the user interface and applications installed in their car infotainment systems[5]. This includes the installation of third party applications which is supported by the Android Operating system [5]. Due to convenience, infotainment systems are allowed to communicate with other devices in the car through CAN (Control Area Network) buses. According to [4], devices in an automobile that are interconnected by CAN buses can be classified into 2 categories: convenience devices and vehicle-critical communication devices. Devices such as the Infotainment system, air conditioning, door locks and windshield vipers are considered to be convenience devices while devices such as the engine and braking system are considered as vehicle-critical communication devices. This is illustrated in the Figure 1.

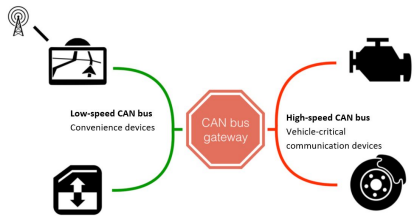

Fig. 1. Schematic overview of CAN bus system according to [4]

The convenience devices are inter-connected by low-speed CAN buses while the vehicle-critical communication devices are inter-connected by high-speed CAN buses. The low-speed CAN buses and high-speed CAN buses are connected to each other through a gateway called the CAN bus gateway. The reason for this connection is because there are certain instances when data needs to flow between the low-speed CAN buses and the high-speed CAN buses. For example the Infotainment system needs to communicate with the engine to receive status information and errors from the engine to display it to the driver on its display screen. Similarly, the door lock sensors need to communicate with the engine in order to enable and disable the immobilizer. These features have the potential to make automobiles susceptible to remote compromise through its infotainment system.

The CAN bus gateway is supposed to act as a firewall to the traffic that passes between the two types of CAN buses. However, there is always the possibility that malicious traffic could cross through this gateway unnoticed if the gateway firewall is not familiar with such malicious content. This was demonstrated in [6] where hackers managed to compromise a Jeep Cherokee through the infotainment system. Therefore, attackers who manage to compromise the Infotainment system may be able to navigate themselves to other vehicle-critical

978-1-5386-8540-2/19/$31.00 ©2019 IEEE
DOI 10.1109/HASE.2019.00044

communication devices such as the engine and brake system and cause them to malfunction. This could result in accidents and car theft that could lead to many social, economic and political problems and even loss of lives. This is the reason why security of the automotive Infotainment system is of great importance.

Contributions: In this paper, we carried out vulnerability analysis on an automotive infotainment system that is running on the android 5.1.1 operating system.

Organization: The structure of this work is as follows: In section II, we introduce the automotive infotainment system and its functionality and its interaction with users. We introduce the WIFI capability of the infotainment system as well. In section III, we look at various vulnerability analysis that were carried out on the WIFI capability of the infotainment system such as Nmap port scan, Nessus vulnerability scan, android attack and denial of service (DOS). In section IV, we propose certain recommended security upgrades. In section V, we detail some of the related works to this paper and finally we conclude in section VI.

II. Context: Infotainment System

Product Functionality: For our experiment, we were provided with an automotive infotainment system from a well known manufacturer whose products are used by many reputable auto makers. The product that we ran our tests on are currently being used in various car models on the roads. The automotive Infotainment system that we are testing upon supports the following functionality: WiFi, Bluetooth, CAN connectivity, GPS, USB 2.0 Host, SD card reader, UART port, HDMI 1080p.

The system boots-up with Android 5.1.1 customized for in-vehicle Infotainment functionality. Upon boot-up, the system displays a dashboard containing application icons. The touch screen on the display unit enables users to easily launch applications such as the following: Navigation System, Music Player, Image Gallery, Internet Browser, Video Player, File Manager, Settings.

For this experiment, we carried out an attack vector analysis according to [1] where we planned out which parts of the Infotainment system needed to be reviewed and be analyzed for vulnerabilities. This would help us to identify which parts are open to attack and try to attack them. Only if the attack is successful, we can find ways to mitigate the attack and prevent such an attack from happening again in the future. We will define attack vectors as all parts of the Infotainment system that can be used by an attacker to infiltrate into it. The attack vectors would include all routes for foreign data or commands to enter and exit the Infotainment system's control system.

The infotainment system that we are carrying out our experiments runs on the Android 5.1.1 operating system. Therefore, we would notice certain similarities in the attack surfaces between itself and an android mobile phone. This should not be surprising as both devices are running on the same operating system. By carrying out careful analysis, we found out that the following are the attack vectors on the Infotainment system due to the reasons [3] stated: WIFI, Bluetooth, USB wired connection, SD card reader.

WIFI: It allows the Infotainment system to communicate with external devices in the same network. Therefore, data or commands can be sent to the Infotainment system by other devices existing in the same network. At the same time, the other devices can obtain data from the Infotainment system. Overall, we focused on the WIFI attack vector during our vulnerability analysis as the points of entry for any malicious codes, packets, requests etc.

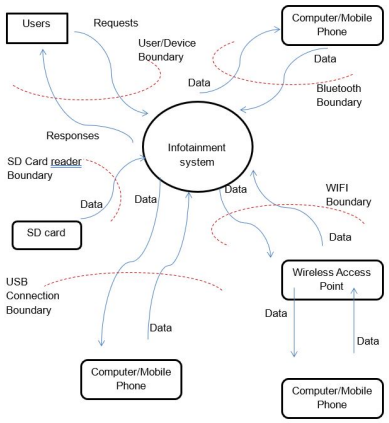

Fig. 2. Interaction with Infotainment system

As shown in Figure 2, users of the Infotainment system will interact with it by sending request commands by touchscreen of the display. The Infotainment system's silver box will process these requests and deliver responses onto the display screen. The boundary between the user and the Infotainment system is the user/device boundary.

Computers or mobile phones can communicate with the Infotainment system by pairing with it and sending and receiving data. The Infotainment system can send and receive data with a wireless access point when it is trying to browse the web. The boundary between the Infotainment system and the wireless access point is known as the WIFI boundary. The wireless access point will have other devices such as computers or mobiles phone that are also communicating with it by exchange of data. Since these boundaries are the paths through which data traffic enters the Infotainment system, it should be well monitored and filtered.

III. WIFI

The Infotainment system has WIFI capabilities which allow it to connect with other WIFI capable personal mobile devices such as mobile phone and laptops which can act as a hotspot. This allows users to get their Infotainment system connected to the internet and be able to stream contents online such as videos, music, pictures, files etc. The WIFI capability being

a point of connectivity to provide great convenience can also cause vulnerability to the Infotainment system by serving as an attack vector. This is due to its ability to connect the Infotainment system to the outside network and hence the outside world where adversaries are lurking. The following are some of the vulnerability assessment that we managed to carry out on the Infotainment system:

A. Nmap port Scan

Nmap is a security scanner that is used to scan for hosts and services running on a computer network. By connecting ourselves to the same network as the Infotainment system, we managed to run Nmap and found the IP address of the Infotainment system to be 192.168.1.144. We also found its MAC address to be F4:5E: AB: 5B:5F:5E and its net card manufacturer to be Texas Instruments. Furthermore, UDP Nmap scanning of the port 5353 reveals that this port is either open or filtered and that the service running on it is Zeroconf. The zeroconf service is responsible for allocating addresses without a DHCP server, carrying out IP to name translation without DNS server and for service discovery such as printers etc. Hence we are able to obtain information about the Infotainment system by doing this Nmap scan. This information may be useful to cybercriminals who can use the IP address or MAC address to launch an attack on the Infotainment system remotely.

B. Nessus Vulnerability Scan

The Nessus vulnerability scanner is an open source vulnerability scanner that can detect certain vulnerabilities in devices. Some vulnerability that may be found using Nessus are as the following:

- Vulnerabilities for remote compromise of device within a network
- Misconfiguration
- Test the strength of login credentials such a login passwords
- Denial of service vulnerabilities

Although Nessus can find vulnerabilities in target devices that could be exploited, it does not actively prevent attacks. Prevention still needs to be done by the device administrator by patching up the identified vulnerabilities. Nessus consists of two main parts which are the Nessus daemon (Nessusd) and the web server Nessus. Nessusd carries out the scanning while the Nessus web server controls scan and display the scan results.

We followed the steps [10] undertaken to perform the Nessus Vulnerability scan on the infotainment system and obtained the results shown in Figure 3. From the previous Nmap scan, we found out that the IP address of the infotainment system is 192.168.1.144.

The medium risk vulnerability is: *mDNS Detection (Remote network):* An attacker can discover information about the Infotainment system such as its operating system (OS) and version, its hostname and the list of services running on it.

Sev	Name	Family	Count
MEDIUM	mDNS Detection (Remote Network)	Service detection	1
INFO	Ethernet Card Manufacturer Detection	Misc.	1
INFO	Host Fully Qualified Domain Name (FQDN) Resolution	General	1
INFO	ICMP Timestamp Request Remote Date Disclosure	General	1
INFO	Nessus Scan Information	Settings	1
INFO	Traceroute Information	General	1

Fig. 3. Nessus Scan Results

This is because the service running on the Infotainment system on port 5353 understands the mDNS protocol. During the scan, Nessus was able to find out the mDNS hostname of the Infotainment system which was Android.Local. Solution: Filter incoming traffic on UDP port 5353.

The information leak vulnerabilities are:

Ethernet card manufacturer detection: The Ethernet card manufacturer from the Infotainment system was identified as Texas Instruments and its 24-bit Organizationally unique Identifier (OUI) was also identified as f4:5e:ab:5b:5f:5e.This OUI is registered with IEEE.

Host Fully Qualified Domain Name (FQDN) Resolution: The FQDN of the Infotainment system was fully resolved as android-5b6a0fc1bb50ecf7.SECURITY-LAB-SG.

ICMP Timestamp Request Remote Date Disclosure: The Infotainment system responds to ICMP time-stamp requests with 23536 seconds difference.This will allow attackers to know the date that has been set on the Infotainment system. This will allow an attacker to defeat time-based authentication protocols. Solution: Filter out incoming ICMP timestamp requests and outgoing ICMP timestamp replies.

Traceroute Information The traceroute information retrieved by Nessus shows that the traceroute from the scanning PC to the Infotainment system is a single hop.

C. Android Attack

Since we know that the Infotainment system runs on the Android operating system, we could try to take advantage of vulnerabilities that exist in the Android operating system to compromise the device. For this attack we created a malicious APK file that requests and gains special privilege of the Android services in the Infotainment system upon installation. We will use social engineering methods such as those stated in [7] to send over the APK file to the Infotainment system and trick the user into installing the malicious APK file. Social Engineering is a method by which we can manipulate the user of the Infotainment system into downloading the malicious files such as clicking on a link etc. We named this malicious APK file as Android_upgrade.apk file in order to trick the Infotainment system user to download and install this file thinking that it is actually an upgrade file for its operating system. Once the malicious APK file has been installed and activated, it can be used to gain access to the critical file

system, photos, music, videos etc. We even demonstrated this by stealing a picture that was in the Infotainment system to the attackers computer remotely. The following method details how we managed to create the malicious APK file, send it over to the victim (Infotainment system) and make him install it on the system and steal sensitive information from the Infotainment system.

Creating the malicious APK file: We can create the malicious APK file by using msfvenom [8] which is a standalone payload generator for Metasploit. We create the reverse TCP payload and set the IP address and port of our attacker PC as the address for the reverse TCP payload to send back information from its host. We name the malicious APK file as Android_upgrade.apk to seem harmless to the unsuspecting user.

Set up of listening port in attackers PC: We will need to set up a listening port on the attackers computer to listen to connection requests from the malicious APK file once it has been installed in the Infotainment system (victim) as an application. This can be set up using Metasploit using the multi handler exploit. We shall set the payload to be reverse TCP and the listening host to be the attackers computer which has an IP address of 192.168.1.180. Once the parameters are set, we can run exploit to have the listening port running.

Sending the malicious APK file to victim: We will have to use social engineering methods [7] to send the Malicious APK file to the victim which is the Infotainment system. The file can be hosted online and trick the victim into downloading and installing the file thinking that it is an upgrade file for the android device.

Another method of sending the APK file to the victim will be to store it in a SD card and insert the SD card into the SD card reader of the Infotainment system. Since the Infotainment system does not do any authentication of the SD card we can install the malicious APK file into the system by switching off the security properties. To switch off the security properties, we need to connect to the silver box using the USB connection and view the terminal emulator. In the terminal emulator, we just need to input the following two commands to switch off the security in place against installation of third-party applications.

 # setprop persist.sys.security.install false
 # setprop persist.sys.security.check false

This method only works if the attacker has physical access to the Infotainment system for a short while. Once the file has been downloaded into the Infotainment system, it is installed. During installation, it can be seen that the malicious application is gaining permission to sensitive applications in the system like its file system, pictures, songs, videos etc as shown in Figure 4. The unsuspecting user would simply click install and install the application on the Infotainment system.

Connection established with listening port and information theft: Once the malicious APK file has been installed in the Infotainment system, a meterpreter session starts in the attackers computer which allows the attacker to view the entire file system of the Infotainment system. From this file system, the attacker can navigate to the particular picture stored in the DCIM folder and download it. Once the download is complete, he can then upload it into his computer home page and view it under the home folder.

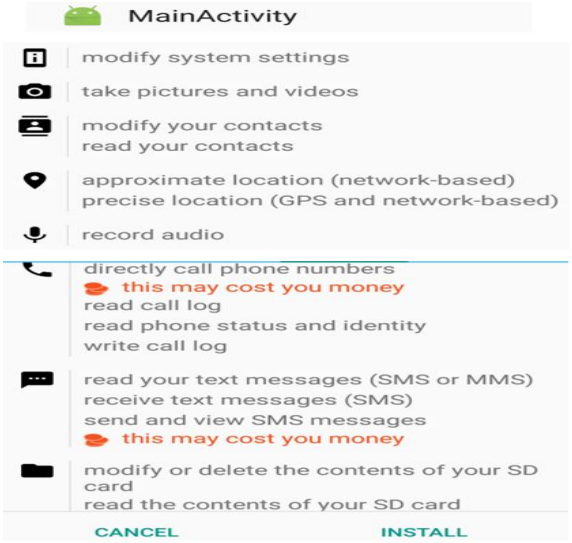

Fig. 4. Permission request by malicious app

D. Android Attack on an Android Mobile phone

When we performed the android attack on the Infotainment system, we were only able to gain access to its file system, pictures, songs, videos due to the limited functionalities of the Infotainment system. However, the same android attack can expose other functionalities as well if they were present in the device such as in figure 5.

Functionalities	Requirement
Record voice conversations happening within the car	Mic in the Automotive infotainment system
Provide live streaming of video call	Inbuilt webcam in the Automotive infotainment system
Take photos remotely	Camera on the Automotive infotainment system

Fig. 5. Compromise other functions of Android

In order to demonstrate this, we used the same attack on an Android device which had more functionalities as described earlier. Hence we performed the attack on an Android mobile phone which had a mic and a camera. Therefore we tried to gain more sensitive information through this attack such as voice recording of private conversation, live video streaming and snapping pictures from the mobile phones camera. Therefore we downloaded and installed the same malicious android_upgrade.apk file on the mobile phone. We open a listening port on the attackers computer using Metasploit and waited for the connection to be established. Once the

connection is established, we can type “help” to see the range of data we can steal as shown in Figure 6.

```
Command         Description
-------         -----------
record_mic      Record audio from the default microphone for X seconds
webcam_chat     Start a video chat
webcam_list     List webcams
webcam_snap     Take a snapshot from the specified webcam
webcam_stream   Play a video stream from the specified webcam
```

Fig. 6. Privacy attack options

Here in the help menu we can see various functionalities such as webcam_stream, webcam_snap, record_mic and geolocate. We tried them out one by one as follows:

webcam_stream: When we initiate webcam_stream, we can remotely view whatever is being viewed by the camera of the mobile phone on the attackers computer. A new browser opens up in Mozilla Firefox and starts streaming the video to the attacker at real time as shown in Figure 7. Similarly, if an automotive Infotainment system had an inbuilt camera, attackers would be able to view activities happening in the car remotely. This would be a serious intrusion of privacy and secrecy.

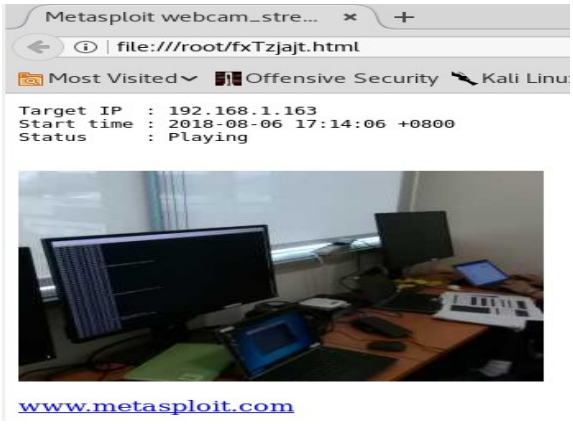

Fig. 7. Webcam stream

record_mic: When we initiate record_mic, we can remotely record and listen to whatever voice conversations can be picked up by the mobile phones microphone on the attackers computer. The voice clip gets stored in the attackers home directory as a wav file. Similarly, if an automotive Infotainment system had an inbuilt microphone, attackers would be able to record and listen to conversations happening inside the car. This is also a serious intrusion of privacy and secrecy.

webcam_snap: When we initiate webcam_snap, we can remotely click pictures using the mobile phones camera and view them on the attackers computer. The pictures get stored in the attackers home directory in JPEG format. Similarly, if an automotive Infotainment system had an inbuilt camera, attackers would be able to snap pictures of the activities that are happening inside of the car. This is also a serious intrusion of privacy and secrecy.

geolocate: When we initiate geolocate, we can remotely find out the latitude and longitude of the location of the mobile and view them on the attackers computer as shown in Figure 8. Similarly, if an automotive Infotainment system had an inbuilt GPS functionality, attackers would be able to locate the car. This attack can be used by attackers for espionage.

Fig. 8. Location display

E. Denial of Service

We were successful in carrying out a denial-of-service (DOS) attack on the Infotainment systems WIFI connectivity by using the aireplay-ng attack [12]. The aireplay-ng attack is used to send multiple deauthentication packets [9] to the Infotainment system which is currently connected with a Wireless access point (WAP) which in our case is the SECURITY-LAB-SG access point. The deauth packets were sent using the Realtek’s WLAN Adapter set up on the TP-LINK’s wireless adapter (TP-LINK, n.d.). The following code is what we used to send deauthentication packets to break the connection between the Infotainment system and the SECURITY-LAB-SG access point.

$\#\ aireplay{-}ng{-}01000{-}a2C:54:2D:38:BB:D8{-}cF4:5E:AB:5B:5F:5Ewlan0mon$

Where:

- -0 refers to deauthentication packets
- 1000 is the number of deauthentication packets that we want to send. In this case we have set the number as 1000 which will send 1000 packets over a long period of time. However in our attack we want to send them continuously. Hence we will use 0 which will send deauth packets continuously forever.
- 1000 is the number of deauthentication packets that we want to send. In this case we have set the number as 1000 which will send 1000 packets over a long period of time. However in our attack we want to send them continuously. Hence we will use 0 which will send deauth packets continuously forever.
- -a 2C:54:2D:38:BB:D8 is the MAC address of the access point “SECURITY-LAB-SG
- -c F4:5E:AB:5B:5F:5E is the MAC address of the Infotainment system which we want to deauthenticate
- wlan0mon is the interface name

When this attack is carried out, the WIFI connection between the Infotainment system and the access point gets

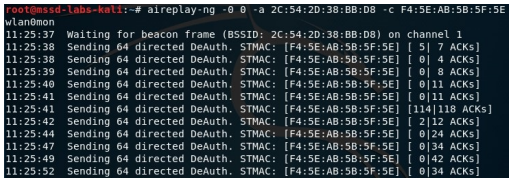

Fig. 9. Sending deauth packets

disrupted. When there is a disruption, the Infotainment system will try to reconnect back by authentication with a WPA2 handshake. However, since we are sending deauth packets continuously as shown in Figure 9, it will continuously disrupt the connection. This results in a denial-of-service (DOS) for us of the WIFI connection for the Infotainment system.

IV. Recommended Security upgrades

We recommend a few security upgrades which could help to harden the WIFI attack surface. These are just suggestions and are not implemented as part of this paper as defence against the above mentioned attacks are beyond the scope of this paper.

The Infotainment system could have a mechanism in place that could filter incoming packets to prevent reply to unnecessary information request packets. Blocking the download and installation of malicious APK files is a good method to prevent the installation of backdoors in the Infotainment system. The Infotainment system should prevent all download and installation of APK file from untrusted sources and make it a necessity that all applications need to be downloaded only from their app store.

V. Related Work

In [5], Garzon discusses about the complexity of automotive infotainment system in the current era. These complexities have lead to the rise of vulnerabilities in the automotive infotainment system. In [4], Computest discusses about how the compromise of an infotainment system can lead to the compromise of other critical devices in the car such as the brakes and engine. In [3], Checkoway addresses about various avenues by which an attack can be carried out on an automobile vehicle as modern automobile are highly computerised and thereby exposed to the network. These avenues are classified into indirect physical access, short range wireless access and long-range wireless. These avenues of attack are possible due to vulnerabilities that exist in certain components in the automobile. The attacks can lead to threats such as theft, surveillance etc.

In [13], Mandal looks in depth on the vulnerabilities posed by apps found on the Google play store which can be downloaded and installed onto Android devices such as the automotive infotainment system. In [2], Bordonali discusses in detail about how the usage of Android-based infotainment systems in cars can lead to multiple security risks. In [11], Kong utilizes something known as the attack tree analysis to assess the threats and vulnerabilities posed to the vehicle due to the various information technology devices in it. One of such device is the infotainment system on which our paper is based on. In[9], Joshi explains abut how we can use the aireplay-ng to send deauthenication packets to disrupt any WPA or WPA2 connection. This is useful for us to attack the WIFI capability of our Infotainment system as it can be used to disrupt its WIFI functionality in a similar way.

VI. Conclusion

In conclusion, automotive Infotainment systems have a wide range of vulnerabilities within its WIFI capability. Some of these vulnerabilities were discovered during our analysis. We were able to find out information about the Infotainment system over the WIFI by using Nmap and Nessus scanners. They returned us with sensitive information about the device such as its open port, physical address, chip manufacturer's name, mDNS hostname, FQDN resolution, timestamp, traceroute information etc. The WIFI vector also enabled us to compromise the file system of the Infotainment system and steal data such as photos, videos, music, files etc. We were able to cause a denial-of-service (DOS) attack on the WIFI functionality of the Infotainment system by sending deauthentication packets to break the connection.

Acknowledgements: This work was partially supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-40) and administered by the National Cybersecurity R&D Directorate.

References

[1] J. Bird and J. Manico. Owasp attack surface analysis cheat sheet. *Open Web Application Security Project*, 2015.

[2] C. Bordonali, S. Ferraresi, and W. Richter. Shifting gears in cybersecurity for connected cars. *Mckinsey Company: New York, NY, USA*, 2017.

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, pages 77–92. San Francisco, 2011.

[4] Computest. Research paper, the connected carways to get unauthorized access and potential implications, 2018.

[5] S. R. Garzon. Intelligent in-car-infotainment systems: A contextual personalized approach. In *Intelligent Environments (IE), 2012 8th International Conference on*, pages 315–318. IEEE, 2012.

[6] A. Greenberg. Hackers remotely kill a jeep on the highwaywith me in it, 2015. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[7] M. Hasan, N. Prajapati, and S. Vohara. Case study on social engineering techniques for persuasion. *arXiv preprint arXiv:1006.3848*, 2010.

[8] J. Jain. Hacking an android device with msfvenom, 2018.

[9] D. Joshi, V. V. Dwivedi, and K. Pattani. De-authentication attack on wireless network 802.11 i using kali linux. *International Research Journal of Engineering and Technology (IRJET)*, 4:1666–1669, 2017.

[10] T. Klosowski. How to use nessus to scan a network for vulnerabilities, 2016.

[11] H.-K. Kong, M. K. Hong, and T.-S. Kim. Security risk assessment framework for smart car using the attack tree analysis. *Journal of Ambient Intelligence and Humanized Computing*, 9(3):531–551, 2018.

[12] V. Kumkar, A. Tiwari, P. Tiwari, A. Gupta, and S. Shrawne. Vulnerabilities of wireless security protocols (wep and wpa2). *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(2):pp–34, 2012.

[13] A. K. Mandal, A. Cortesi, P. Ferrara, F. Panarotto, and F. Spoto. Vulnerability analysis of android auto infotainment apps. In *Proceedings of the 15th ACM International Conference on Computing Frontiers*, pages 183–190. ACM, 2018.