Cybersecurity for Connected Cars

Exploring Risks in 5G, Cloud, and Other Connected Technologies

Numaan Huq, Craig Gibson, Vladimir Kropotov, Rainer Vosseler

TREND MICRO™ | research

TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

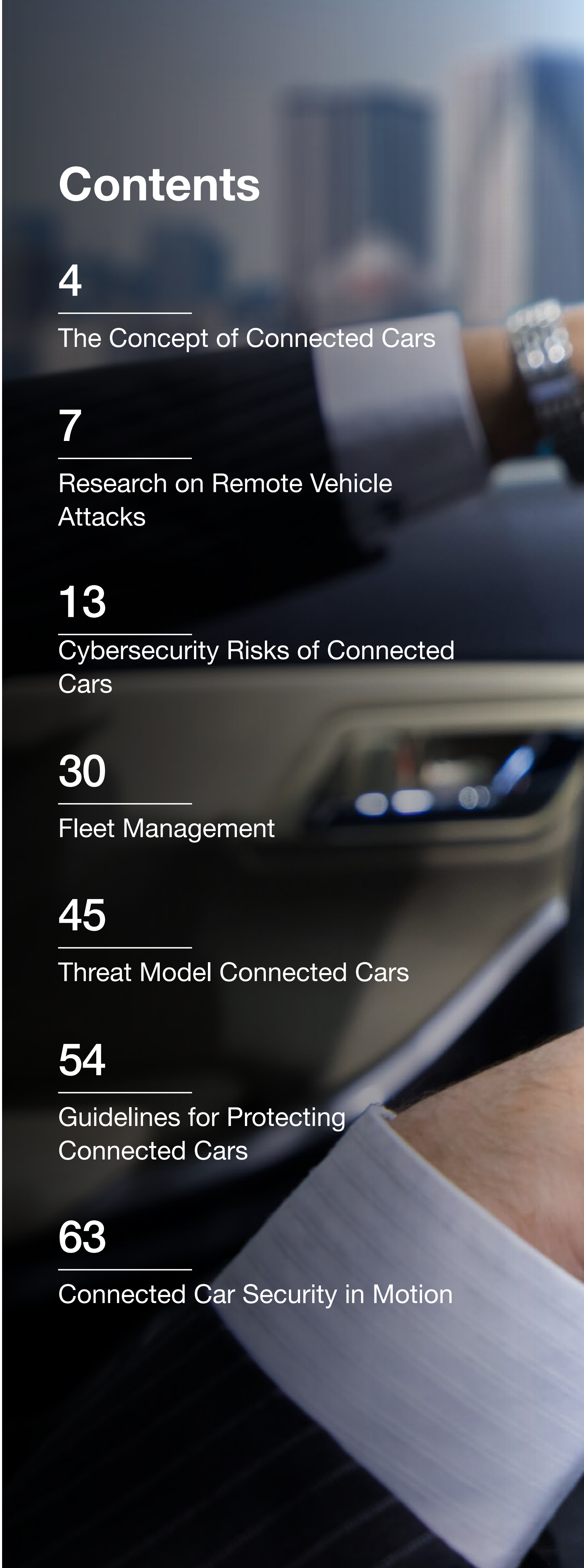
Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an "as is" condition.

Published by
Trend Micro Research

Written by
Numaan Huq
Craig Gibson
Vladimir Kropotov
Rainer Vosseler

Stock image used under license from
Shutterstock.com

For Raimund Genes (1963-2017)



Contents

The use of connected cars has become more and more common, a trend that will continue in the foreseeable future.[1] As the use of these vehicles grow, the scope and weight of risks will continue to increase as well. Similarly, the vehicles' link to connected technologies such as 5G and the cloud can provide heightened speed and improved quality of services but can also pose more risks in the form of attacks by threat actors who aim to exploit these channels for their own gain.

As 5G networks roll out globally, connected cars are expected to heavily utilize the low-latency, high-bandwidth, and network slicing features of these networks. The 5G network backbone, together with advances in artificial intelligence (AI) and machine learning (ML) applications for both on-board and in-cloud data processing, will bring fully autonomous vehicles one step closer to reality. Still, such technology is not immune to threats.

Likewise, the cloud can provide advantages for connected cars; for example, some electronic control unit (ECU) functions can be moved to the cloud to simplify the electrical/electronics (E/E) architecture, expand processing capabilities, and reap many other benefits. However, it also poses risks like denial of service (DoS) attacks, man-in-the-middle (MitM) attacks, and hijacking of services, among others.

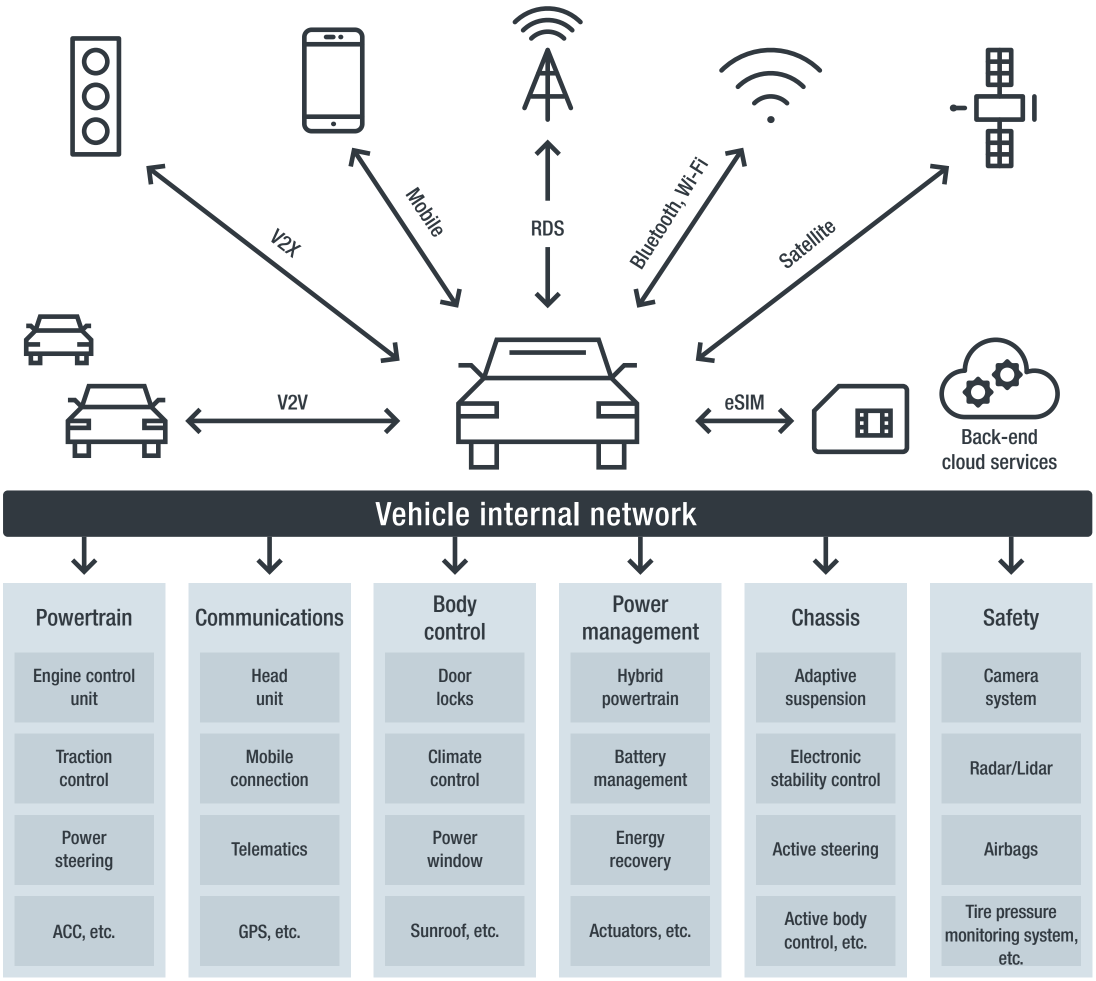
In line with this, this research also delves into fleet management and the risks that come with managing vehicles as a connected unit. Now more than ever, it is important to explore these risks in order to better prepare the proper defenses against threats and as a result, future-proof the security of connected cars. This paper aims to help car manufacturers and security professionals equip connected cars with ample and proactive defenses.

This paper is an expanded version of the research paper released last year titled "Driving Security Into Connected Cars: Threat Model and Recommendations."[2]

1. The Concept of Connected Cars

Connected cars are part of the internet of things (IoT). These vehicles can access and send data, download software updates, and connect with other connected cars or other IoT devices via the internet or WLAN (wireless local area network) connections.[3] They can provide their users with enhanced connectivity and infotainment, and facilitate safer driving.[4] It is estimated that by 2030, the number of connected cars will reach 700 million, while the number of autonomous vehicles will reach 90 million.[5]

Contrary to common belief, connectivity in cars is far from a new concept. In fact, today's basic car already comes with a wide variety of connected technologies. The following is a discussion of some of these connected technologies.



Powertrain	Communications	Body control	Power management	Chassis	Safety
Engine control unit	Head unit	Door locks	Hybrid powertrain	Adaptive suspension	Camera system
Traction control	Mobile connection	Climate control	Battery management	Electronic stability control	Radar/Lidar
Power steering	Telematics	Power window	Energy recovery	Active steering	Airbags
ACC, etc.	GPS, etc.	Sunroof, etc.	Actuators, etc.	Active body control, etc.	Tire pressure monitoring system, etc.

Figure 1. The technologies and functionalities that make up the internal network of a connected car

A typical new-model car runs over 100 million lines of code.[6] The very basic cars have at least 30 electronic control units (ECU), which are microprocessor-controlled devices, while luxury vehicles can have up to 100 ECUs.[7] ECUs are all connected across a labyrinth of various digital buses such as CAN (Control Area Network), Ethernet, FlexRay, LIN (Local Interconnect Network),[8] and MOST (Media Oriented Systems Transport).[9] They operate at different speeds, move different types of data, and enable connections across different parts of the car.[10] ECUs control many critical functions in a car, including the powertrain, the device and system communications body control, power management, the chassis, and vehicular safety. Some of them can be accessed remotely via the head unit.

A modern car can already receive satellite data for connecting to radio stations and getting GPS coordinates. In the future, cars will have cellular-satellite connectivity for data, which is especially useful when driving through regions with poor cellular coverage.[11] With companies like Amazon, OneWeb, and SpaceX racing to launch megaconstellations of internet-beaming satellites into low Earth orbit, cellular satellite connectivity is expected to become mainstream within a few short years.[12]

Most new car models sold in the market have built-in embedded-SIMs (eSIMs), although some of them are not activated. Built-in eSIMs are used to transmit telematics data, communicate with back-end cloud servers, create Wi-Fi hotspots, and get real-time traffic information, among other functions. Examples of cloud-based back-end server applications include smart apps that can remotely start, stop, lock, and unlock a car, and apps that can automatically send current road conditions data to the cloud and transmit to other vehicles subscribed to the same service.

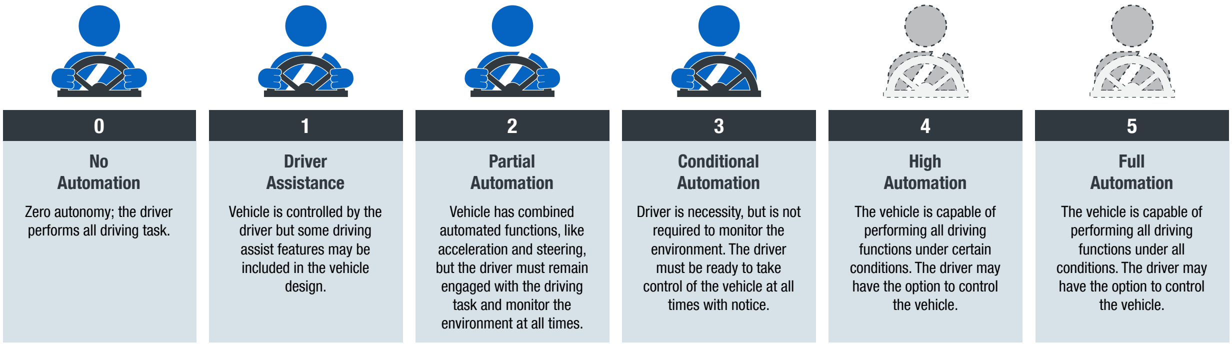
RDS (Radio Data System) is used to embed small amounts of digital information in FM broadcasts. Typically, the name of the radio station, the title of the song, and the time and date of airing are transmitted. Using RDS-TMC (Radio Data System – Traffic Message Channel), a car can also receive real-time traffic alerts, which are then displayed in the head unit.

Bluetooth and Wi-Fi are common in cars nowadays. Users' mobile phones connect via Bluetooth to the head unit of a car to perform activities such as playing music, making phone calls, and accessing address books. Some cars, such as those made by Tesla, can connect to home Wi-Fi networks and download over-the-air (OTA) software update packages for the cars.[13] Many cars can create Wi-Fi hotspots for users to connect to in order to access the internet via the cars' eSIMs.

With the introduction of popular automotive telematics standards, mobile phone connectivity in cars has shifted from simply making phone calls and accessing address books to allowing users to gain access to apps, maps, messages, and music. Even basic cars now have support for standards such as Apple CarPlay and Android Auto, thus making in-car apps available to the masses.

Vehicle-to-everything (V2X) communication is the driving future that the industry is headed toward. Vehicles will be heavily relying on V2X to safely navigate roads. The two major V2X technologies being actively developed are 802.11p, a WLAN-based system,[14] and C-V2X (cellular vehicle-to-everything), a cellular-based system that includes 5G.[15] New cars are being equipped with either of these two technologies, but a full rollout of V2X in every new car is still several years away, especially since the 802.11p and C-V2X camps are competing for market share. Legacy vehicles, or vehicles that are not equipped with V2X technology, will continue to be on the road for decades, and V2X vehicles will have to share the road with them. V2X technology will amalgamate information from multiple network drop points — eSIM, mobile network, RDS-TMC, Wi-Fi, and 802.11p or C-V2X — to build complete road situational awareness.

1.1 Automation and connected cars



0	1	2	3	4	5
No Automation	**Driver Assistance**	**Partial Automation**	**Conditional Automation**	**High Automation**	**Full Automation**
Zero autonomy; the driver performs all driving task.	Vehicle is controlled by the driver but some driving assist features may be included in the vehicle design.	Vehicle has combined automated functions, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.	Driver is necessity, but is not required to monitor the environment. The driver must be ready to take control of the vehicle at all times with notice.	The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.	The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.

Figure 2. Society of Automotive Engineers (SAE) automation levels

Image Credit: National Traffic Highway Safety Administration[16]

The Society of Automotive Engineers has defined six automation levels: L0 to L5, with L0 meaning no automation and L5 meaning full automation. Many "self-driving" cars on the road today like Mercedes Drive Pilot, Tesla Autopilot, and Cadillac Super Cruise, among others, are classified as L2 because the driver must remain engaged with the driving task and monitor the environment at all times. Tesla sells a beta-stage, full self-driving package that it claims can navigate highway merges and exits, lane changes, and interchanges, automatically park, and be summoned from a parked position to where the driver is waiting, not to mention obey traffic lights and stop signs, and autosteer on city roads.[17] All of this maneuvering is done using only the GPS, cameras, radar, and ultrasonic sensors, with no lidar,[18] no V2X, and no cooperative driving. This places Tesla's self-driving package somewhere between L3 and L4 when it exits the beta-stage, but that might take some more time as driver safety concerns[19] need to be addressed first.

One major roadblock for autonomous vehicles is that related legislation needs to be discussed to allow for greater driving automation, after which more L3 and L4 capable vehicles will be seen on the road. As a prerequisite for such legislation, V2X needs to be widely implemented using either C-V2X or 802.11p.

New car models will need to implement V2X modules and road infrastructure needs to be upgraded to communicate via V2X. This begs the question: which standard should be followed, C-V2X, 802.11p, or both? And how can the bill for the enormous cost of upgrading road infrastructure to support V2X standard(s) be paid? In a nutshell, future vehicles need to be always online and able to communicate quickly and efficiently with the ecosystem to make L4+ autonomous driving an everyday reality.

2. Research on Remote Vehicle Attacks

In 2017, we released a research paper that studied cyberattacks on intelligent transportation systems (ITSs).[20] This current research is a follow-up of that earlier research. Now, we study the cybersecurity risks posed by connected cars interacting with other vehicles, cloud services, and road infrastructures.

In this paper, to better understand the cybersecurity risks that connected cars face, we studied past car-hacking attempts. There are a lot of published research papers on hacking cars, but we focused only on attacks carried out *remotely* that successfully compromised at least one ECU inside a car. This gave us a good understanding of the tools, techniques, and procedures (TTP) used to compromise connected cars.

Once we understood the TTPs that are successfully used to remotely attack connected cars, we theorized on possible cyberthreats against the connected car ecosystem, including the challenges of deploying traditional malware against cars, attacking cloud-based services inside cars, visualized traditional IT attacks on connected cars using the MITRE ATT&CK Matrix[21] future connected car architecture in the era of 5G, and theories on how cybercriminals might attempt to monetize attacks against connected cars, among other cybersecurity risks and threats.

Connected cars are "phones with wheels" that combine the risks of cars, telecommunications, and IT while centralizing these risks in connected car fleet management. Notably, phones are some of the most powerful surveillance devices ever made, and thus they expose cars to risks that are related to surveillance, privacy, and fraud.

In our ITS research paper, we did DREAD threat modeling of the entire ITS ecosystem. In this research, we apply DREAD to both the connected car and its ecosystem to determine which areas pose the greatest threats — this is a more focused and in-depth analysis compared to our general ITS threat modeling. In addition to the monetization analysis, based on our expertise in cybercrime investigations,[22] we also theorized which threat actors would pose the greatest threats to the connected car ecosystem and explored other motivations besides money.

Connected cars primarily communicate wirelessly, but there are exceptions. An example is when an electric vehicle is connected to the power supply and communicates with the power grid or another back-end infrastructure over the power line. This is not covered in this research. Instead, this research focuses on wireless attacks as the main attack vector. Hacking via Bluetooth is considered a wireless attack, but the limited range of Bluetooth radio and the need to be within the immediate vicinity of the victim vehicle make it ineffective in compromising a fleet of vehicles. There are many published research papers and articles on hacking cars, but only a small subset explores remotely executed attacks that have successfully compromised at least one ECU inside a car. It is important to study these connected car hacking cases to better understand the cybersecurity risks that connected cars face, and to gain a better awareness of the TTPs used by hackers, which we then apply to our threat modeling analysis.

Finally, the connected car research would not be complete without some suggestions with regard to protective measures for these cars and their ecosystem.

2.1 Connected Car Network Architectures

Modern connected cars have internal network architectures that are as diverse as the cars themselves. The components communicate using standardized network protocols, but no two network architectures are the same. The network architecture can even change between different makes and models from the same manufacturer because the features of the cars will vary based on their prices. Figures 3, 4, and 5 illustrate three examples of car network architectures.

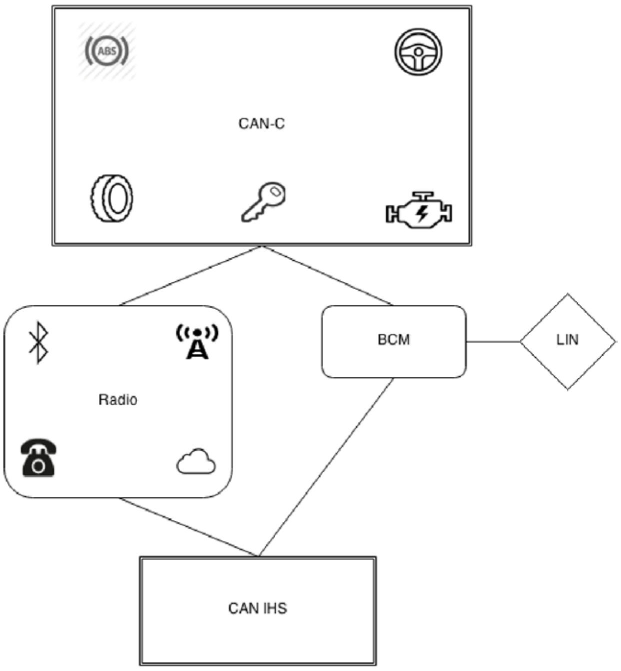


Figure 3. The Jeep Cherokee network architecture that the researchers Charlie Miller and Chris Valasek compromised in 2015

Image credit: Charlie Miller and Chris Valasek[23]

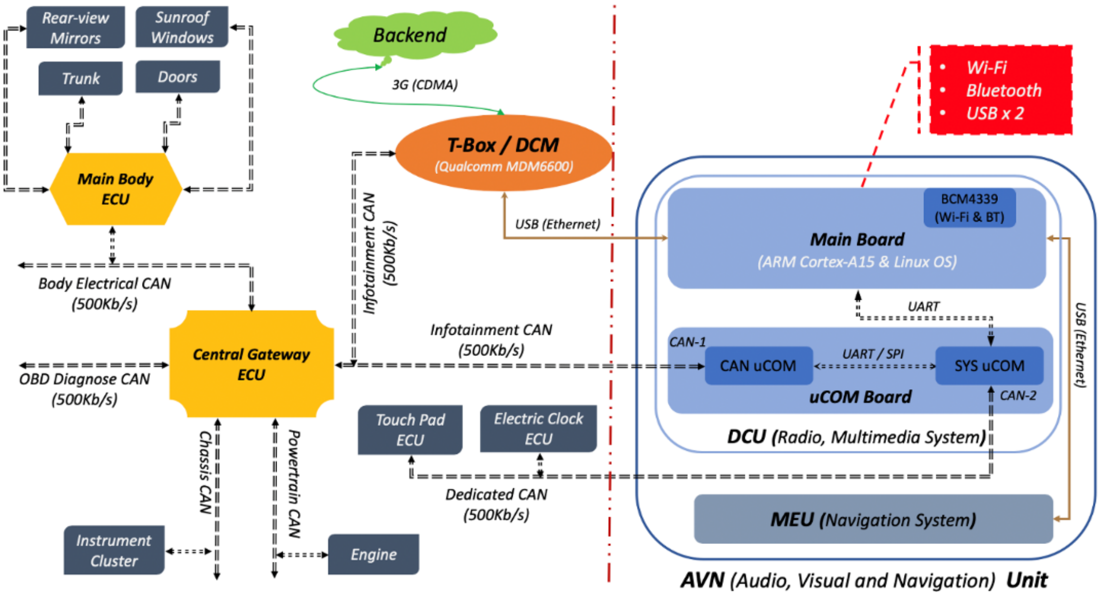


Figure 4. The Lexus network architecture that Tencent Keen Security Lab compromised in 2020

Image credit: Tencent Keen Security Lab[24]

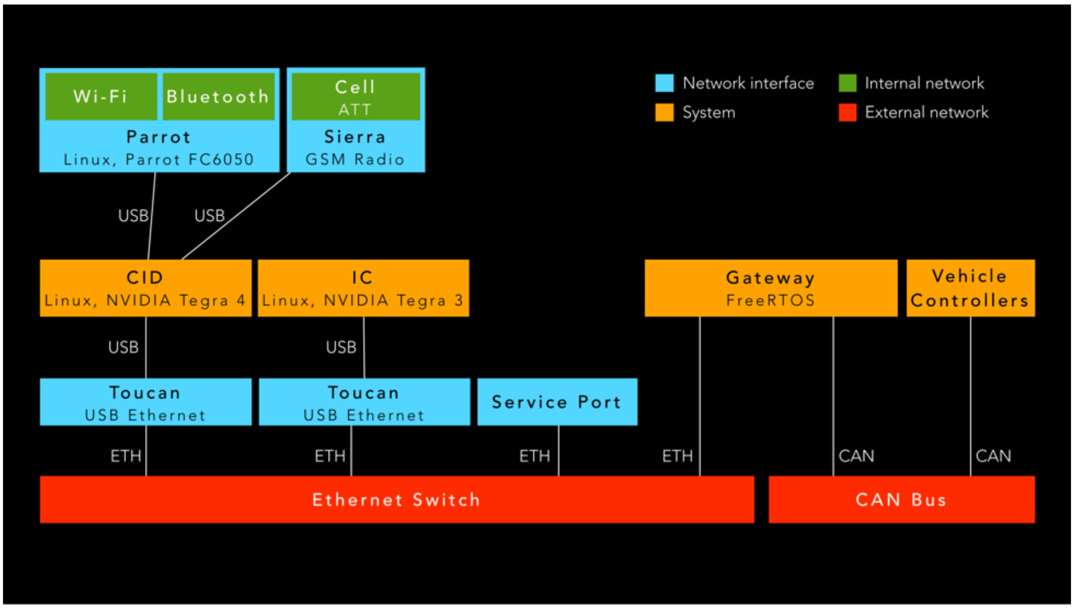


Figure 5. The Tesla network architecture that the researchers Kevin Mahaffey and Marc Rogers explored in 2015

Image credit: Lookout[25]

We observed that while the manufacturers in these examples implement their networks differently, all three architectures have common components such as the gateway, CAN bus, USB, Wi-Fi, and ECUs that perform similar functions and interact in similar ways. To explore the functions and the interactions of these components, we created a generic car network architecture. This is not a network architecture from a production vehicle but rather a theoretical visualization of the network topology and the major components in a connected car's network.

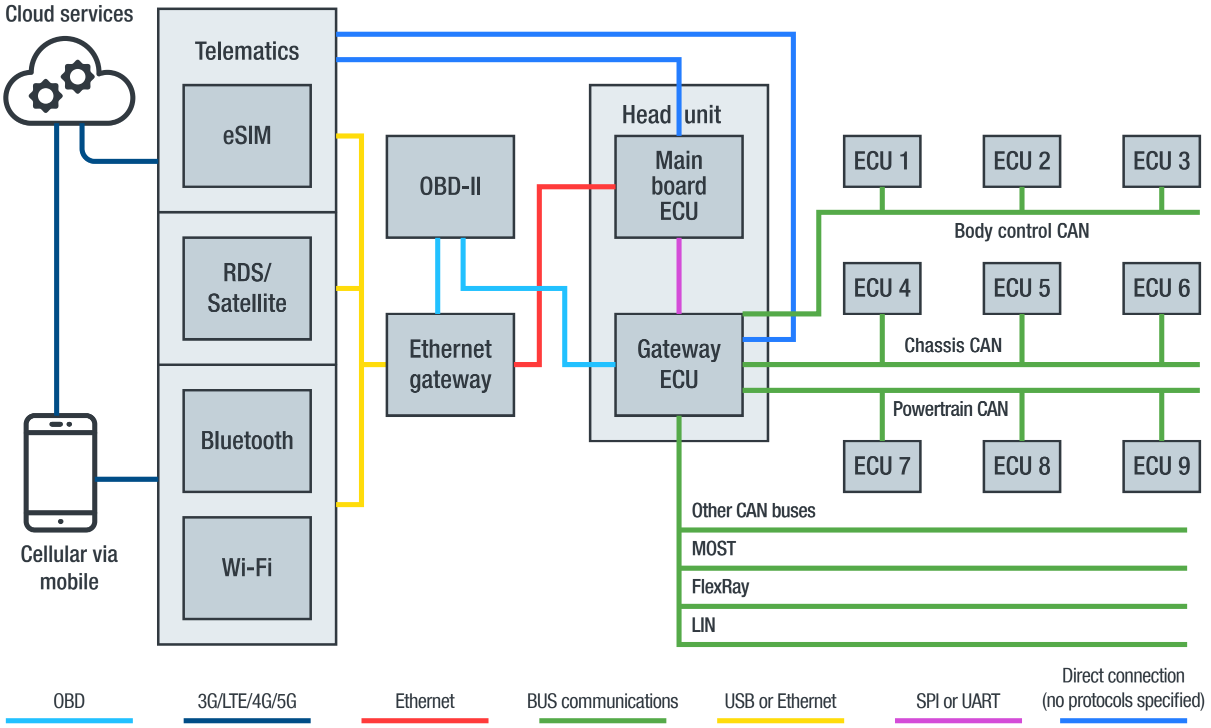


Figure 6. A theoretical visualization of a generic network architecture for a modern-day connected car

The following discusses the major components and their respective interactions in our generic car network architecture:

- The **telematics** unit includes the **eSIM** that allows the car to communicate with 3G, LTE, 4G, and (in the future) 5G networks. It can transmit telematics data, receive real-time data, communicate with back-end cloud servers, and allow access to the internet.

- The **RDS/satellite** unit receives digital information from FM and satellite broadcasts. Using RDSTMC, a car can receive real-time traffic alerts that are then displayed in the head unit. In the future, the satellite component will enable cellular-satellite connectivity for transmitting data as an alternative to 3G, LTE, 4G, and 5G.[26]

- **Bluetooth** and **Wi-Fi** connectivity is common in modern cars. Users can use Bluetooth to connect their mobile phones to a car's head unit in order to play music, make phone calls, and access address books. Some cars can create a Wi-Fi hotspot to provide internet connectivity to users and to connect to home Wi-Fi networks to download OTA software updates. Mobile phones connected to Bluetooth and/or Wi-Fi can tether to give a car access to the internet via 3G, LTE, and 4G networks.

- **On-board diagnostics (OBD-II)** provides a vehicle's self-diagnostics and reporting capabilities. The OBD-II port can communicate with the head unit. It can talk directly to the CAN bus and send and receive CAN messages and commands.

- The **ECUs** in a car communicate via their connected bus and handle functions such as engine control, traction control, door locks, climate control, battery management, hybrid powertrain, airbags, and radar functionalities.

- The **gateway ECU** handles all communications with the different buses: CAN, LIN, MOST, and FlexRay. Other bus protocols exist, but we used these four in our research since they are found in most car models. The gateway ECU ensures that no application can directly communicate with the buses, and it correctly switches messages to the target bus. It also performs validation procedures to make sure that the messages conform to standards.
- The **main board ECU** is the central processor for the head unit. It handles functions such as navigation, display, radio playing, network connection management, and climate control. In our architecture, the main board ECU communicates with the gateway ECU via the SPI (Serial Peripheral Interface) communication protocol[27] or the universal asynchronous receiver-transmitter (UART)[28] to send and receive CAN messages and commands.
- The **Ethernet gateway** handles all of the data switching between the radio frequency (RF) modules and the head unit. In some car network architectures, the Ethernet gateway can directly communicate with the gateway ECU. In our generic architecture, the Ethernet gateway communicates via the head unit.

2.2 Generalized Remote Hacking Techniques for Connected Cars

We looked at four remote car hacking case studies: Jeep Hack 2015, Tesla Hack 2016 and 2017, and BMW Hack 2018 (These case studies are discussed in detail in Appendix A). From these case studies, we found an emerging attack pattern that is repeated across all four attacks to compromise the connected cars and send malicious CAN messages to the ECUs. We illustrate this pattern in the following diagram:

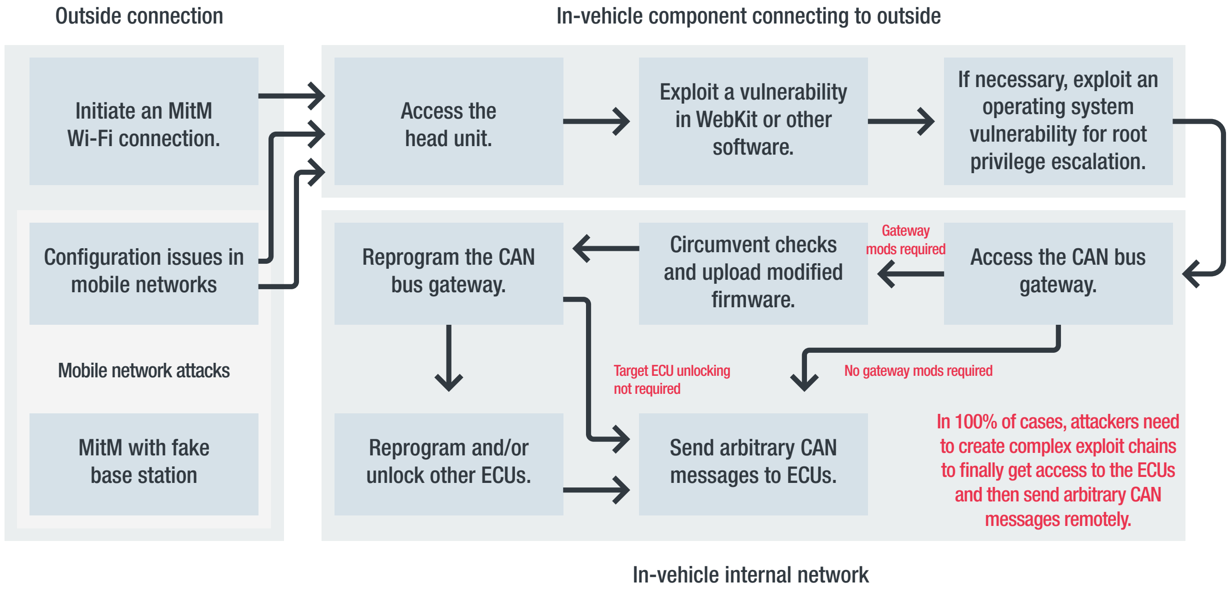

Outside connection
In-vehicle component connecting to outside
Initiate an MitM Wi-Fi connection.
Access the head unit.
Exploit a vulnerability in WebKit or other software.
If necessary, exploit an operating system vulnerability for root privilege escalation.
Configuration issues in mobile networks
Reprogram the CAN bus gateway.
Circumvent checks and upload modified firmware.
Gateway mods required
Access the CAN bus gateway.
Mobile network attacks
Target ECU unlocking not required
No gateway mods required
MitM with fake base station
Reprogram and/or unlock other ECUs.
Send arbitrary CAN messages to ECUs.
In 100% of cases, attackers need to create complex exploit chains to finally get access to the ECUs and then send arbitrary CAN messages remotely.
In-vehicle internal network

Figure 7. A generalized remote hacking attack chain based on the featured remote attack case studies

Examining this attack pattern, we observed that:

- The initial attack vector is either via a mobile network or via a Wi-Fi network. In all four case studies, the hackers attempted some type of a man-in-the-middle (MitM) attack[29] using either the mobile network or the Wi-Fi network.
- The attackers always targeted the head unit, which is the infotainment hub of the vehicle. Head units are found in all modern cars, with different degrees of functionality depending on whether the car is a basic one or a luxury vehicle. All new head units can talk to the gateway ECU, which makes them the go-to entry point into the vehicle's bus network.
- Head units with an LCD screen run a browser agent such as WebKit. The hackers exploited a new or previously existing vulnerability for that browser to get a browser shell.
- The browser shell typically has low privilege. Privilege escalation is required to get root access to the underlying operating system of the head unit, which is commonly Linux.
- The main board ECU talks to the gateway ECU, which then talks to the bus network. To send arbitrary CAN messages to the bus network, the hackers needed to reflash the gateway ECU with their custom firmware. Depending on an ECU's function, it may not require firmware overwriting. If firmware overwriting is not needed, the hackers could immediately start sending CAN messages to the bus network and the connected ECUs.
- Flashing the gateway ECU needs circumventing firmware integrity checks and flashing and restarting the ECU in a reliable manner. Any mistakes in this step risks "bricking" the gateway ECU, requiring an awkward visit to the car dealer to get it fixed by reflashing.
- After they gained control over the gateway ECU, the hackers could send arbitrary CAN messages to the bus-connected ECUs. Some ECUs, however, needed to be reprogrammed and/or unlocked to execute the hackers' CAN messages. Unlocking ECUs might also allow the hackers to put such ECUs in diagnostic mode, so that the malicious CAN messages would not be overwritten with valid CAN messages.
- The gateway ECU handles the routing of the arbitrary CAN messages to their target ECUs based on the target ID of the CAN message frame, which greatly simplifies the attack execution.

3. Cybersecurity Risks of Connected Cars

A modern car has an incredible number of connected technologies such as satellite, cellular, Wi-Fi, Bluetooth, RDS, eSIM-based telematics, and others. The car uses these connected technologies for sending and receiving data that supports user applications, driving applications, autonomous driving, safety features, and many more. All these network-centric applications have created brand new attack surfaces in connected cars. In this section, we apply our expertise in IT cybersecurity to explore potential new network attacks on connected cars.

3.1 Connected Cars and 5G

A connected car comprises of two main components: the connectivity and the car itself. The connectivity portion is based in part on what the connected vehicle will be doing, which determines where they might be expected to go. If a connected vehicle is expected to travel on private property only (such as in an open mineral mine or inside a warehouse), it might be connected with best-effort radio technology like Wi-Fi. If a vehicle is intended to travel at potentially dangerous speeds through areas with a lot of people (such as on a highway) it might use 4G cellular; but since like Wi-Fi, 4G is also a “best effort” technology with unreliable data quality, a human might still be required to drive the vehicle at all times.

In 5G, the data is both high-speed and high-quality. 5G data has a feature called quality of service (or QoS) that allows connected vehicles to travel at high speeds. 5G is an example of a piece of technology that would enable connected vehicles to operate in spaces that are congested and potentially dangerous, such as highways, parking lots, and in the case of autonomous drones, even the air. 4G and other radio technologies are also managed in individual monolithic ways with interactions that are complex to handle. These complex interactions often require continuous maintenance and tuning, making them unscalable to the demands of 5G. In 5G, each of these technologies is automated, and automated in their relationships with each other.

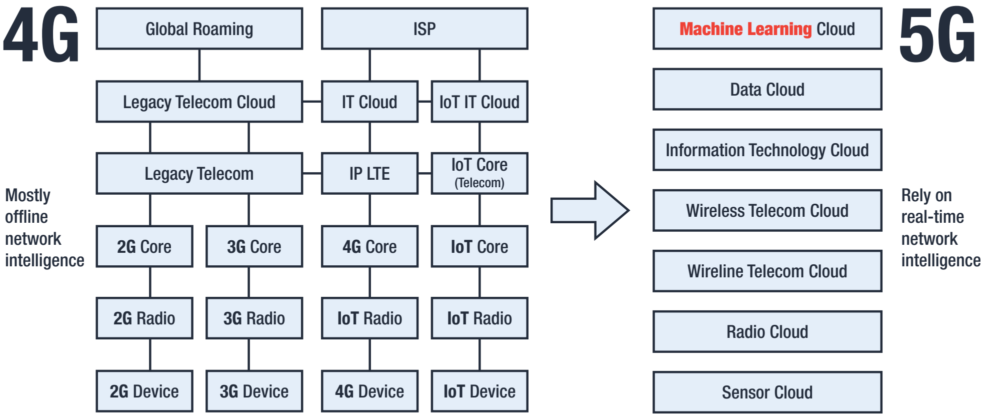


Figure 8. A side-by-side comparison of 4G and 5G networks for connected cars

As "phones with wheels," connected cars have a specific set of phone-like features. In 5G networks, common service-level features are gathered in 5G network "slices." These slices allow a 5G network to be operated more cheaply and effectively, and to reduce the chance of different kinds of devices receiving the wrong data (aka information that provokes automated actions in the cars).

Slices and Slicing

A slice is a virtual network. A slice meets the safety and security requirements of specific regions, laws, and vehicle types. This network slicing is performed at each level of the network and allows cars to function with more service stability.

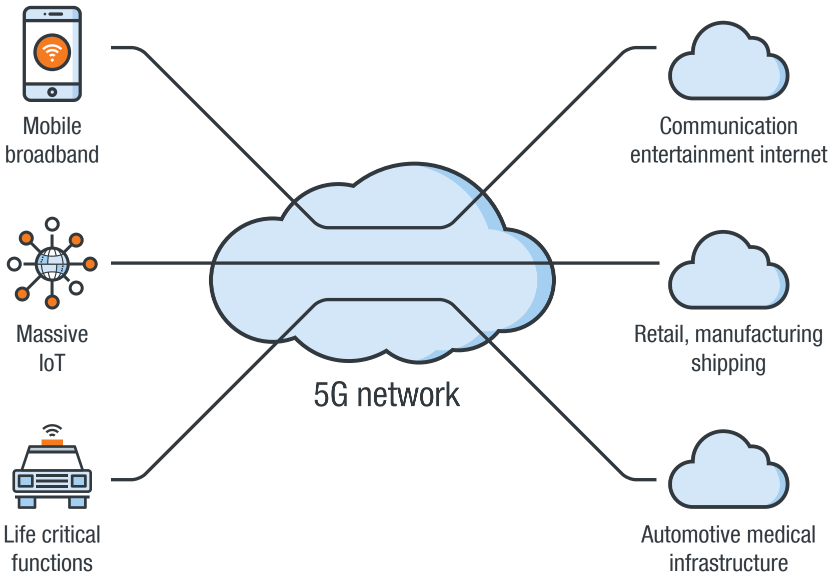


Figure 9. Example of 5G network slices

Image Credit: VanilaPlus news magazine[30]

In a connected car's 5G network, these slicing rulesets are present at each level of the network. The network remains dynamic as a whole but becomes both less expensive and more reliable for cars using the slice.

Datacentric

5G is a datacentric network where specific telemetry values/data provokes network changes at the application network and hardware levels. This is the real strength of 5G: What was once unscalable and manual in 4G is capable of being end-to-end automated in 5G. This common infrastructure that runs customized services leads to a service mesh. Not only does every element in the mesh have its vulnerabilities and patching needs, but every element also has its own concerns that involve access control with topics like Oauth2[31] token sprawl (equivalent to losing control of passwords in an unlogged way) and logging structures that defeat normal detection rules that would trigger a security response (the section on fleet management has more details about this).

C-V2X

Connected vehicles can also link to or at least gather information from the devices around them. Cellular Vehicle to Everything or C-V2X is intended to thoroughly make a 5G car a part of the environment around it, making it capable of reacting to events. While C-V2X does function for 4G-connected vehicles, its primary purpose in this setup is to provide convenience for the driver by giving traffic alerts and other similar functions.

In 5G, C-V2X comes into maturity. C-V2X has a large, stable, and innovative supply chain of integrated technologies. The ability of the vehicle to behave both autonomously and cooperatively based on what it senses from the environment is a step forward from traditional cars such as those in the 4G and pre-4G eras. In 4G, C-V2X is primarily managed near the roadway. In 5G, C-V2X is managed primarily in the network back end for the sake of centralization and reduced cost.

There are several subsets of C-V2X, some of which are discussed here:

- Vehicle to cellular network (V2N)
- Vehicle to device (V2D)
- Vehicle to vehicle (V2V)

Vehicle to Network (V2N)

The V2N approach connects vehicles with the back-end mobile carrier infrastructure. There are several benefits to this approach. These include SIM card subscription management, the ability to receive firmware updates over-the-air, management of the vehicle's digital assets such as subscriptions, enhanced features (such as autonomous driving), anti-theft measures, and fleet management.

V2N technology also provides cooperative coordination to traffic; emergency vehicles can be given priority to drive through at life-saving speeds. Large-scale analytics can be performed, including traffic engineering for smart cities, remotely deployed improvements to vehicle functions like improved braking, and responses to events such as high-traffic conditions, changes in weather, and public safety events such as flooding. Organizational improvements to traffic such as assembling platoons or convoys of three or more vehicles also rely on the 5G cellular mobile network.

Vehicle to Device (V2D)

For V2D, the granular coordination of 5G can relay the location of pedestrians carrying personal devices or cellphones to vehicles, reducing the likelihood of vehicle-pedestrian collisions. But the relationship between vehicles and personal devices is not limited to pedestrian collision avoidance. The vehicle receives navigation information from V2N based on V2D inputs. Passengers can use the vehicle's onboard C-V2N connectivity to receive data inside the vehicle, which the vehicle obtains from the local cell tower. Additionally, V2D provides information that drives activities such as billing and onboard advertising. Information gathered inside the car can also be leveraged by the fleet manager for additional enrichment analytics such as advertising outside of the vehicle.

Vehicle to Vehicle (V2V)

V2V promotes collision avoidance as it enables securely navigating intersections. As vehicles approach an intersection, they communicate by exchanging certificates directly through public key infrastructure (PKI). Since V2V helps avoid collisions, this might lower insurance costs and make pay-as-you-drive insurance a more affordable and suitable option. It also determines which vehicle will take a competing resource such as a parking spot or lane change opportunity. This cooperative driving capability not only improves safety and traffic efficiency but also increases the number of vehicles that can use a stretch of roadway through better organization.

3.2 Cloud-Based Car E/E Architecture

The advent of 5G will play a big part in the connected car ecosystem, so it is expected that the (E/E) architecture of connected cars will evolve to take advantage of the next-generation network. The key 5G technologies that will be important to connected cars include the following:

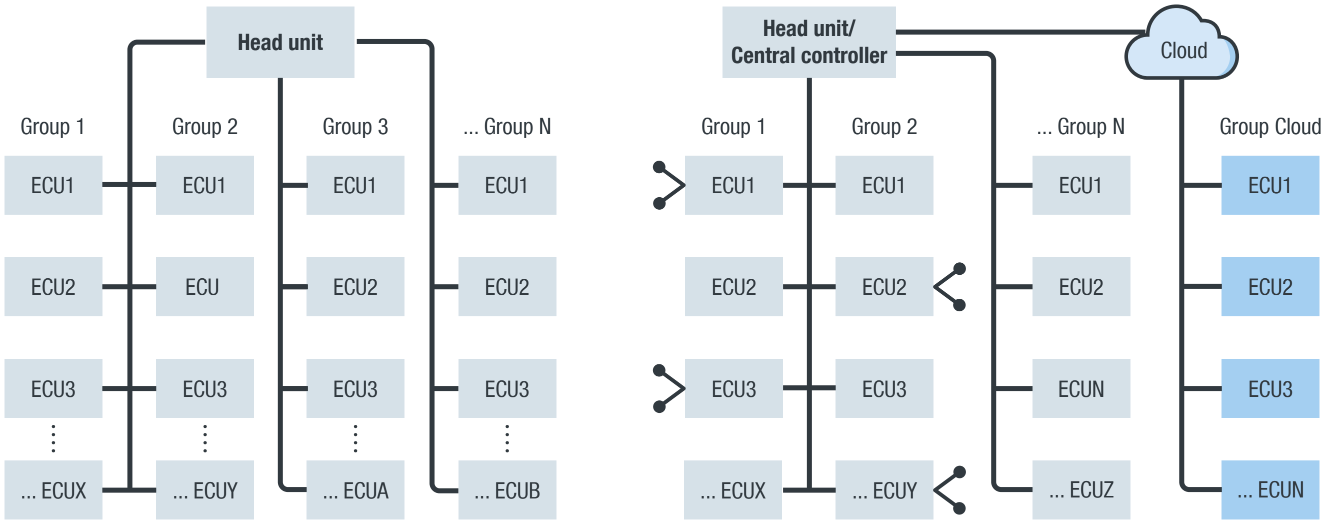
- Software-defined networking (SDN)
- Beamforming for fast communications
- Network slicing to separate applications and QoS
- 5G ML and AI applications
- Multiuser multiple input multiple output (MU-MIMO)

- High availability (99.999%) and low latency (1 millisecond)
- Support for high density of connected devices and ultra-high speed
- Low power consumption

Connected cars will need ultra-low latency more than high bandwidth. Network slicing and SDN can help separate high data rate applications in the car from ultra-low latency applications.

To illustrate the importance of ultra-low latency, here is an example of a car that relies on a cloud-based decision service to shift gears: A car driving at 100 km/h on the freeway travels at 2.77 cm/ms. A supercar like a Ferrari or Lamborghini can automatically shift gears every 50 ms. In between gear shifts, the car will have traveled 138.5 cm or 4.54 feet. To intervene, the network needs to respond in less than 50 ms as the 100 to 0 km/h braking distance is roughly 100 feet or 1.1 seconds. This illustrates that reaction time is critical. On the other hand, the cloud-command to shift gears is probably a few packets long, comprising the command and verification checksum, so it does not need a lot of bandwidth.

With 5G this sort of processing is possible, so why not move some ECU functions to the cloud? Gear shifting and transmission are critical processes in a car, so arguments can be made that transmission and gear-shifting decisions should be processed on-board the car. For example, Luxury car manufacturer Rolls Royce has a production car (Wraith model) that uses GPS data to determine gear shifts according to what the road looks like ahead.[32] Future connected cars could use a hybrid cloud and on-board processing model for the transmission that ensures optimum gear selection either for performance, fuel efficiency, or cruising, depending on what mode the driver selects.



Figures 10. Car E/E architecture today (left) versus cloud-based car E/E architecture (right)

In a cloud-based E/E architecture, which ECUs can we move to the cloud and why? Based on our brainstorming we came up with the following discussion points:

- Move data- or processor-intensive ECUs. A low-latency, high-availability network is needed to make this possible.
- Move ECUs to the cloud, because cross-referenced and processed data in the cloud can increase vehicle fuel efficiency and road capacity. For example, when using current road loads plus GPS data plus traffic prediction, the car can select the optimal gear and speed to improve fuel efficiency and reduce emissions. This decision is made in the cloud and sent back to the car. In electric vehicles, speed is adjusted, which in turn affects the battery levels. This is a centralized adaptive algorithm-based powertrain, chassis and also power management system powered by cloud computing and big data.
- Processor-intensive and overly complex tasks such as image processing and road condition assessment should also be moved to the cloud. Of course, there must be localized processes to handle these tasks as a fail-safe in case the network is disconnected, but the heavy lifting can be done by cloud-based servers. Moore's Law[33] predicts that the processors that will be locally installed in the cars will be several magnitudes faster than what is available today. Similarly, available memory for processing tasks will also scale up to keep up with processor capabilities. We can have cloud computing models where only images that the car fails to interpret locally can be sent to the cloud for further processing, or all processing is done in the cloud except when network connectivity falls below a certain threshold, or local-versus-cloud processing is split 50/50.
- Safety systems should be left in the car, but a cloud backup is possible. The in-car safety systems will always be first priority when making judgment calls on the road. If the cloud judges with 100% accuracy that the safety system is making a mistake because it lacks road situational awareness, it can override the safety system up to a permissible limit. Ideally, the car should have full road situational awareness. Still, the cloud sees all traffic at all times, something that the car does not have the capacity or means to process.

So why move ECUs to the cloud?

- It will simplify the E/E architecture, leaving fewer ECUs to manage in the car itself.
- It will massively expand processing capabilities using cloud computing.
- It will enhance road situational awareness, from local to around 500 m radius.
- It gives the ability to add new and innovative applications flexibly using incremental OTA updates.
- It will improve fuel/battery/emissions/operational efficiency.
- It will expand and flexibly control road load capacity from an ITS point of view.
- It will allow easy integration with third-party service providers, for example, by easily subscribing to revenue-generation services like Uber which operates the car in autonomous mode.

- Distributed cloud servers and a full-coverage, low-latency network will result in attackable surfaces that are more resistant to malicious attacks.

Overall, cloud-based ECUs create exciting possibilities but also new challenges. Therefore, what are the immediate risks with regard to having a cloud-based car E/E architecture? Some of the mainstream cloud attacks that OEMs, suppliers, and drivers need to worry about include:

- **Denial of Service (DoS).** This happens when attackers overwhelm a resource, making it unavailable for users. Imagine the havoc on roadways if the whole cloud infrastructure that runs the E/E architecture becomes unavailable. Depending on whether there is a local processor inside the car that can run the entire car's E/E architecture in the absence of the cloud, cars could get stalled, and crashes and fatalities would become unavoidable.

- **MitM attack.** This occurs when an entity intercepts all network communications between the cloud and the car. This attack can modify, drop, delay the transfer of, or steal data, causing critical malfunction in the car.

- **Hijacking of services.** This takes place when some of the services that are used by the cloud-based E/E architecture are hijacked by an entity, modifying data. This kind of hijacking, because of its small scale, is not easily noticeable and potentially introduces intermittent errors into the car's operating environment.

- **Latency issues.** If the network latency continuously fluctuates (because of network issues or an attack) the car will continuously context-switch between cloud and local processors, which may introduce errors in operations. A sound cloud-based E/E architecture should also require on-board processors that can act as a backup in cases where the network connectivity falls below a certain threshold.

- **Data privacy.** Any cloud architecture is bound to store critical and private data such as driver profiles, car maintenance data, destination data, and financial information, among other pieces of sensitive information. Loss or alteration of this data becomes a contentious issue in the event of a data breach.

- **Authentication and management issues.** This is best described using an example: In 2020, Tesla's global network outage disabled the mobile app for their cars, which left Tesla owners unable to control their vehicles via the app for functions such as changing the temperature and unlocking their vehicles.[34] This demonstrates that, an authentication or cloud management issue can severely disrupt operations for cars with cloud-based E/E architecture, as critical services that they rely on could become unavailable in the middle of an operation, such as because of an ongoing DDos attack against the cloud servers.

- **Incorrect data.** In this case, a car receives incorrect critical real-time data. This could stall the car or lead to an accident and potential fatalities. This could be caused by incorrect processing by the cloud-based server, or it could also be the result of an MitM attack.

- **Misconfiguration issues.** This is a common and recurring issue with cloud-based servers and as such is not unique to connected cars. Misconfigurations lead to malware infection, data theft, loss of control, hijacking, and others. Trend Micro Research has recently published a comprehensive paper that explores more of these cloud security issues.[35]
- **Cloud supply chain issues.** The cloud is an "API economy."[36] One example is the data flow when turning on an internet-connected smart lightbulb via the Google Home app. Clicking "on" in the Google Home app sends a request to Google Cloud, which then forwards the request to the smart bulb manufacturer's cloud, which then sends the turn-on command to the smart bulb. These are all accomplished using APIs. Hence, it is not unrealistic to expect that a connected car's cloud could make API calls to a Tier 1 (T1) supplier's cloud, which in turn could make API calls to a Tier 2 (T2) supplier's cloud, and so on. Any break in this cloud supply chain could adversely affect the connected car.

3.3 Cloud Services Attacks on Connected Cars

At the Black Hat USA 2020 conference, 360's Sky-Go team presented their security research on hacking Mercedes-Benz cars.[37, 38] They succeeded in hacking the Mercedes E-Class in multiple ways and found 19 different vulnerabilities that they reported to Mercedes. What stands out as the most interesting attack vector is that the researchers were able to use the car's eSIM to connect to Mercedes-Benz back-end servers. They found a server-side request forgery (SSRF) flaw in the back-end surface of the car's infotainment system, in a plug-in application that allows users to add their social media accounts to the system.39[39] What makes this interesting is that a flaw in a third-party-developed application installed in the car was exploited to compromise the system. In this section, we discuss what we think connected car cloud services will look like, as well as the new attack surfaces that will be introduced. The research from Sky-Go is early proof that this type of attack can be successful and potentially dangerous.

In inspecting the interiors of the Tesla Model 3[40] and the 2021 Mercedes-Benz S-Class,[41] it becomes evident that luxury automobile manufacturers are doing away with physical buttons and switching to fully digital cockpits. In addition to running applications for regular car features such as climate control, radio, hazard lights, these digital cockpits can also run third-party applications like those for maps, internet radio, web browser, streaming video, social media, and messaging. The modern connected car is becoming a giant smartphone-on-wheels where third-party cloud-connected applications play an important part in the experiences of both drivers and passengers.

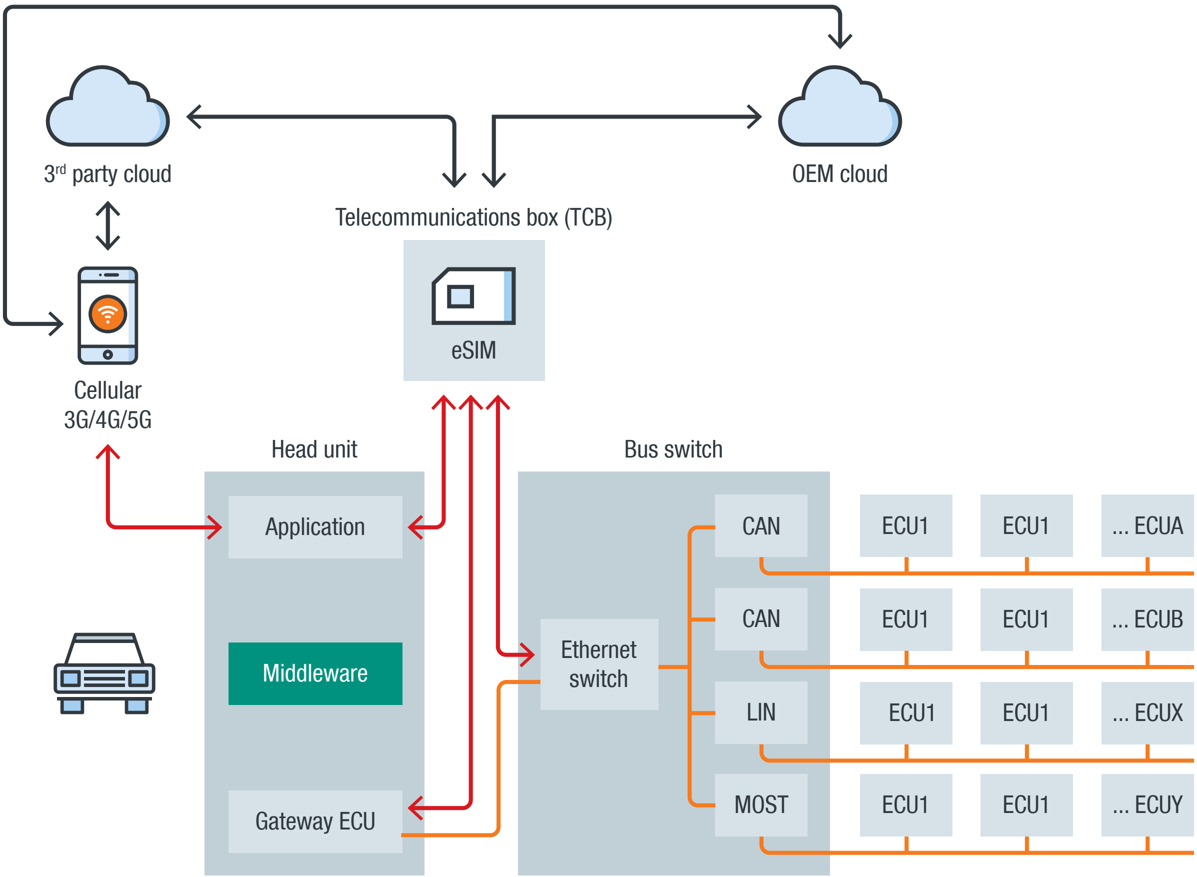


Figure 11. A cloud-connected ecosystem

The diagram in Figure 11 shows what we envision this cloud-connected ecosystem will look like. The head unit will support running applications: There will be a middleware layer that abstracts the E/E details of the car and makes it easier for developers to build car-based applications. The middleware can speak with the gateway ECU and will give API access to applications that need to send messages to the ECUs. The bus switch will route the packets to the target ECUs. The apps talk either to the OEM cloud (manufacturer cloud) or to third-party clouds (like those of video-streaming apps) via a tethered cellular connection from the mobile phone or the built-in eSIM. Depending on the E/E architecture of the car, the gateway ECU can also directly communicate with cloud services. As cars get better connected and smarter, car-specific apps will be developed and T1, T2, and OEM versions of app developers will emerge. OEM apps will probably not need an intermediary to access the gateway ECU, or might even be able to talk to the bus switch directly.

Middleware APIs will create a rich ecosystem for cars with digital cockpits, but they will also present new opportunities for cybercriminals by giving them easy API access to the vehicle's E/E architecture and ECUs. This could give rise to a whole host of architecture-agnostic malware such as phishing attacks on cars that install an architecture-agnostic remote access tool (RAT), ransomware, and botnet, among others. Another plausible vector is using jailbroken phones that are connected to the car, which can be used as a pivot point to install architecture-agnostic malware in the car. In a later section, we discuss how we think criminals will go about monetizing cyberattacks on connected cars. On the flipside, attacks on the car cloud itself will disable functions in the car, cause loss of personally identifiable information (PII), loss of control on the road, and loss of revenue. Cloud APIs can also potentially be used to locate, unlock, start, and steal the car or valuables inside.

Cloud-specific attacks and middleware APIs in connected cars (when they are made available) will become the weapon of choice for cybercriminals, and cars will be easier to compromise using a variety of tried-and-tested malware and attack vectors.

3.4 Challenges of Deploying Malware in Cars

There is this common misconception that just because modern cars share many of the same hardware and software components with everyday IT systems, infecting a car with malware is a straightforward task. This is far from the truth. Here, we discuss the limitations that hackers attempting to infect cars will have to contend with.

- One cannot just walk up to a car, plug in a USB drive, run AutoRun, and install malware. Different car OEMs have vastly different system architecture and software environments that need to be meticulously reverse-engineered for an individual or group to learn how to access systems and execute a random binary.
- Some manufacturers run Linux or even QEMU. A hacker needs to extract the firmware and reverse-engineer the image thoroughly to understand the software internals.
- The cost of development for vehicle malware is high. The malware is also not easily portable, as vehicle hardware, software, and network architectures vary between models and manufacturers. This further increases development costs. Also, the attack vector that the malware uses will determine how effective the attack will be (such as whether it is single-use or can be deployed against a fleet).
- A highly skilled and resourceful hacker who has an understanding of a car's hardware, software, and network architecture is needed to facilitate such attacks. In-depth knowledge of telecommunication technology is also required if a hacker wants to compromise a whole fleet of vehicles.
- Mistakes can be fatal for the driver and passengers, and with malware, such mistakes or fatalities are bound to happen. These will quickly draw the attention of law enforcement and government regulatory bodies, which the cybercriminals would want to avoid.

Even with these shortcomings, it is not impossible to create malware for cars, although such malware would be technically extremely difficult. Supposing that there was a successful ransomware infection for car brand X, for instance, the following points of contention must still be addressed:

- The attackers would need to ensure that the ransomware does not cause fatality as that will attract too much attention.
- The infection can be cleared just by reflashing the ECU(s) at the dealer.
- Ransomware payment cannot guarantee that the car will return to its normal operation; instead of paying, reflashing might be better.
- Ransomware is a best-effort malware versus a targeted attack, so it is logical to expect that no ransomware author will spend a huge amount of time and money developing complex malware that only works with a single car model in specific circumstances.

In the four case studies that we analyzed (detailed in Appendix A), all the attacks ended up using complex chains of exploits and misconfigurations to successfully access the vehicle's internal network.

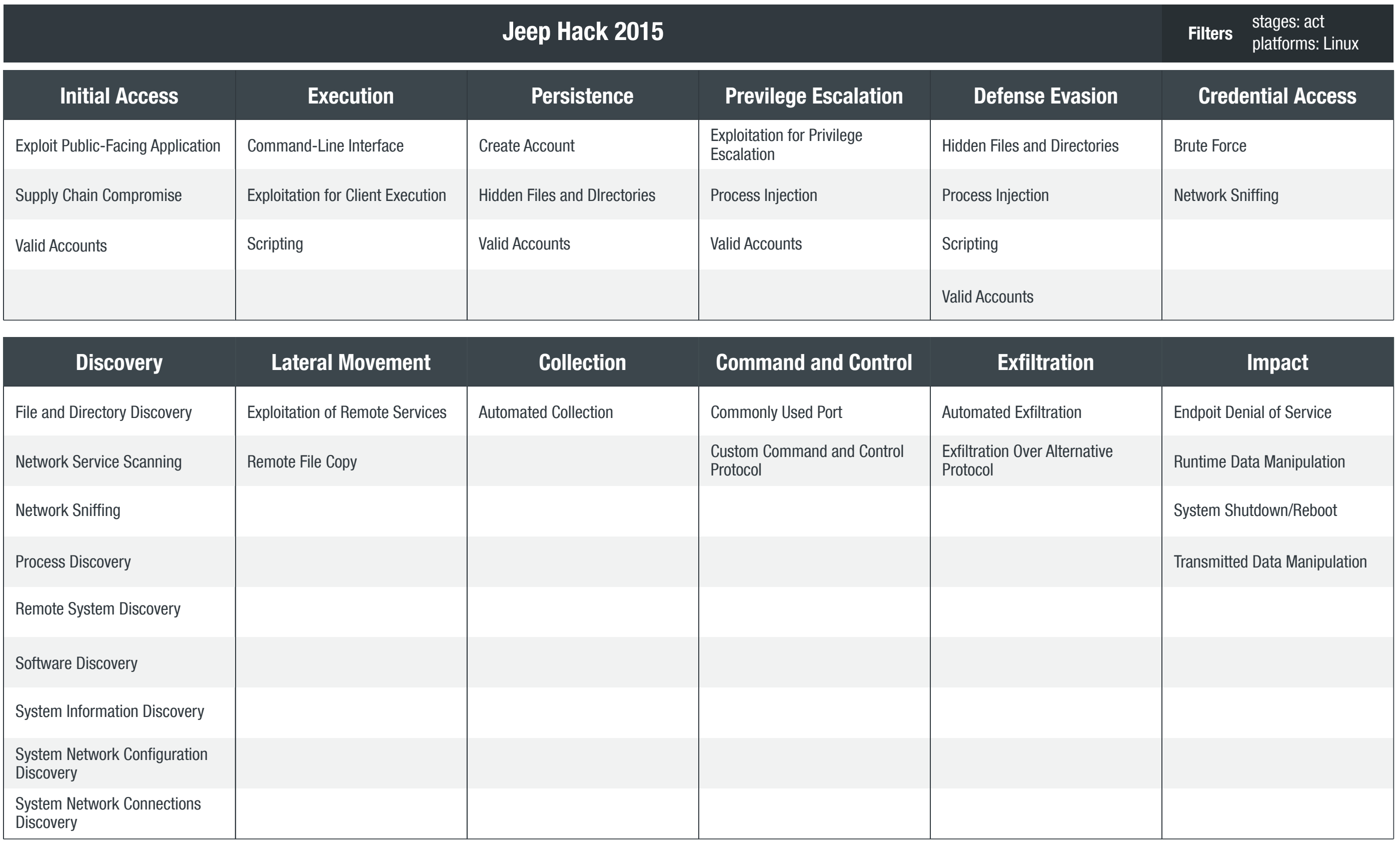
How can cybercriminals infect connected cars? Back in 2016, Trend Micro Research investigated malware attacks against ATMs.[42] The goal of these attacks was to infect ATMs with malware and instruct them to dispense all their stored cash. ATMs from different manufacturers have different hardware and software components, and it would have been impossible for the hackers to take apart every ATM and build custom malware for each brand. Rather, they relied on interacting via APIs with a middleware layer called XFS that runs in most ATMs and that handles all the hardware-specific calls that the malware needs to perform. The same will be true for connected cars where attacks will be made on a middleware layer that runs apps, or on in-car, cloud-based services; these attacks will be architecture-agnostic, similar to how hackers targeted the XFS middleware in ATMs.

In a nutshell, it is not impossible to install malware in cars — just extremely difficult. The only reasonable mitigation for this problem is if the malware becomes architecture-agnostic using middleware APIs, or if it uses an infection-and-spread vector that abstracts the car E/E architecture.

3.5 Traditional IT Attacks on Connected Cars

MITRE ATT&CK® is a "globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, government, and cybersecurity product and service community."[43] MITRE is a commonly used compliance metric used for different ranges of IT security products that are sold in the market today. Standardization makes it easy to identify and assess different classes of cyberattacks and apply the necessary containment and remediation actions to neutralize the threats.

Each of the four remote car-hacking case studies that we analyzed (full details of which can be found in Appendix A) provides technical details of how different parts of their attack chain were executed. Still, none of them provides the full A-to-Z steps of their attack chains, so as not to risk giving cybercriminals an easy-to-follow-guide on how to compromise connected cars. Using our reverse-engineering and programming experience, we attempted to fill many of the gaps to get a near-complete picture of the full attack chains. Given the importance of MITRE to the IT security industry, we thought it would be helpful to plot the attack chains from each of the four case studies into the MITRE ATT&CK matrix. This makes it easy to observe attackers' TTPs, compare attack patterns, and predict how future attacks will take place. It is also a good opportunity to map IT industry security practices to automobiles. As a disclaimer, it should be noted that the ATT&CK matrix plots only serve as approximations of the attack chain based on published information and our analysis.



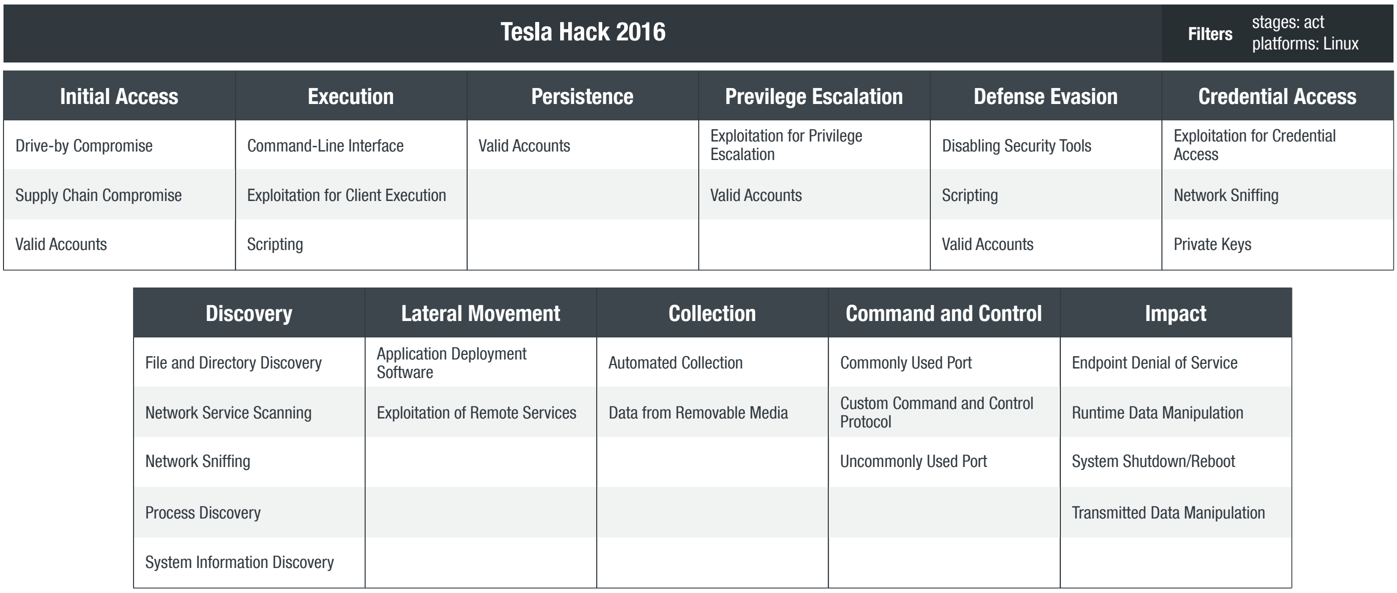
Jeep Hack 2015

Filters stages: act
platforms: Linux

Initial Access	Execution	Persistence	Previlege Escalation	Defense Evasion	Credential Access
Exploit Public-Facing Application	Command-Line Interface	Create Account	Exploitation for Privilege Escalation	Hidden Files and Directories	Brute Force
Supply Chain Compromise	Exploitation for Client Execution	Hidden Files and DIrectories	Process Injection	Process Injection	Network Sniffing
Valid Accounts	Scripting	Valid Accounts	Valid Accounts	Scripting	
				Valid Accounts	

Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
File and Directory Discovery	Exploitation of Remote Services	Automated Collection	Commonly Used Port	Automated Exfiltration	Endpoit Denial of Service
Network Service Scanning	Remote File Copy		Custom Command and Control Protocol	Exfiltration Over Alternative Protocol	Runtime Data Manipulation
Network Sniffing					System Shutdown/Reboot
Process Discovery					Transmitted Data Manipulation
Remote System Discovery					
Software Discovery					
System Information Discovery					
System Network Configuration Discovery					
System Network Connections Discovery					

Figure 12. The Jeep Hack 2015 MITRE ATT&CK matrix



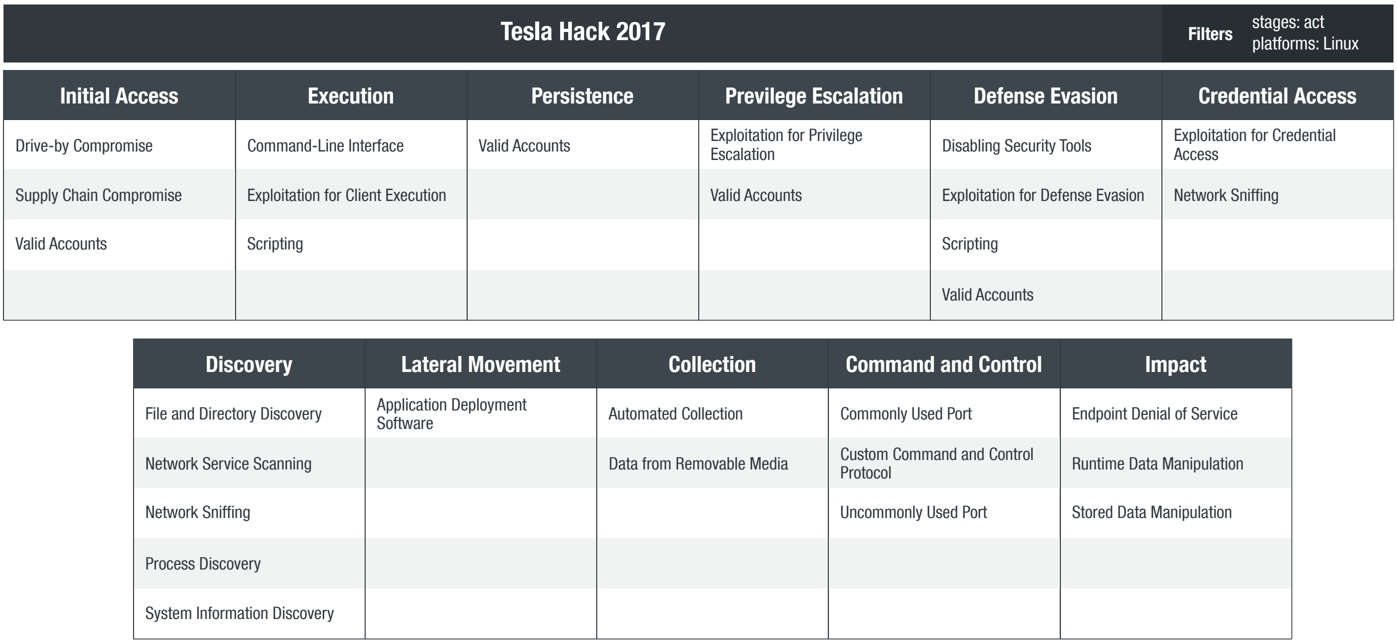
Tesla Hack 2016

Filters: stages: act
platforms: Linux

Initial Access	Execution	Persistence	Previlege Escalation	Defense Evasion	Credential Access
Drive-by Compromise	Command-Line Interface	Valid Accounts	Exploitation for Privilege Escalation	Disabling Security Tools	Exploitation for Credential Access
Supply Chain Compromise	Exploitation for Client Execution		Valid Accounts	Scripting	Network Sniffing
Valid Accounts	Scripting			Valid Accounts	Private Keys

Discovery	Lateral Movement	Collection	Command and Control	Impact
File and Directory Discovery	Application Deployment Software	Automated Collection	Commonly Used Port	Endpoint Denial of Service
Network Service Scanning	Exploitation of Remote Services	Data from Removable Media	Custom Command and Control Protocol	Runtime Data Manipulation
Network Sniffing			Uncommonly Used Port	System Shutdown/Reboot
Process Discovery				Transmitted Data Manipulation
System Information Discovery				

Figure 13. The Tesla Hack 2016 MITRE ATT&CK matrix



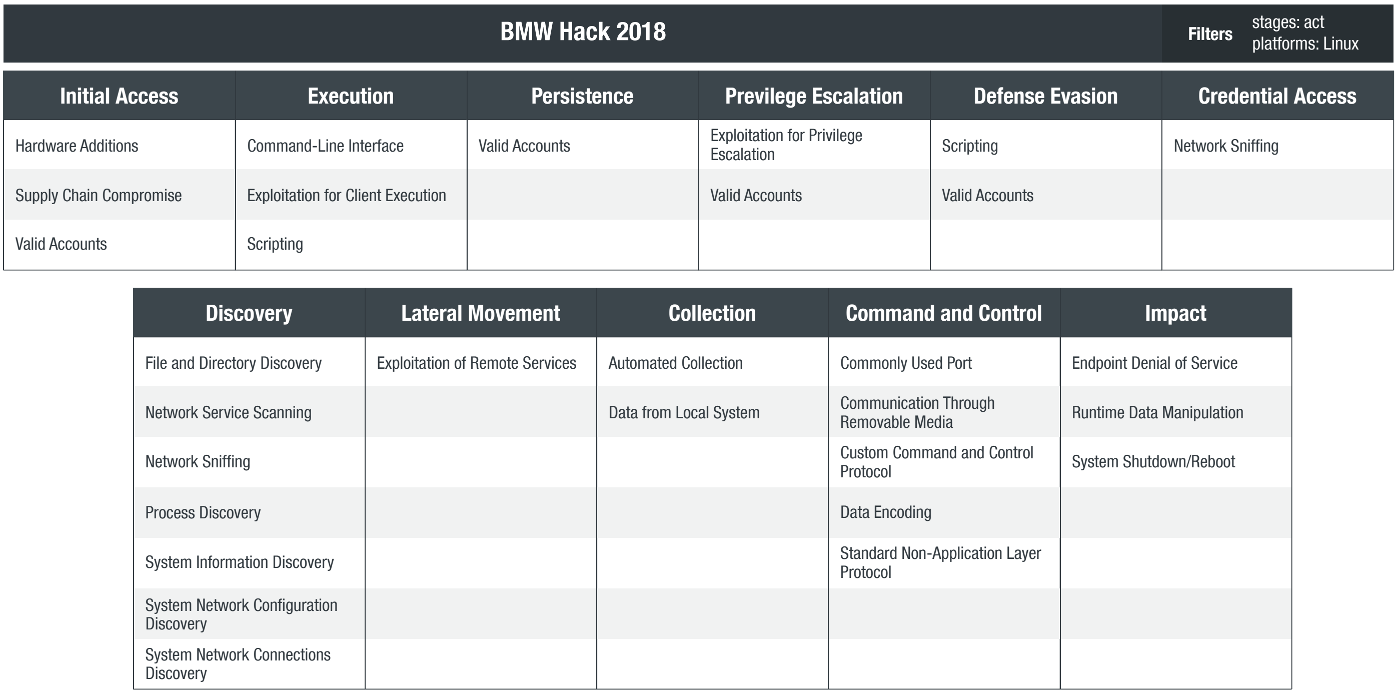
Tesla Hack 2017

Filters: stages: act
platforms: Linux

Initial Access	Execution	Persistence	Previlege Escalation	Defense Evasion	Credential Access
Drive-by Compromise	Command-Line Interface	Valid Accounts	Exploitation for Privilege Escalation	Disabling Security Tools	Exploitation for Credential Access
Supply Chain Compromise	Exploitation for Client Execution		Valid Accounts	Exploitation for Defense Evasion	Network Sniffing
Valid Accounts	Scripting			Scripting	
				Valid Accounts	

Discovery	Lateral Movement	Collection	Command and Control	Impact
File and Directory Discovery	Application Deployment Software	Automated Collection	Commonly Used Port	Endpoint Denial of Service
Network Service Scanning		Data from Removable Media	Custom Command and Control Protocol	Runtime Data Manipulation
Network Sniffing			Uncommonly Used Port	Stored Data Manipulation
Process Discovery				
System Information Discovery				

Figure 14. The Tesla Hack 2017 MITRE ATT&CK matrix



BMW Hack 2018

Filters: stages: act
platforms: Linux

Initial Access	Execution	Persistence	Previlege Escalation	Defense Evasion	Credential Access
Hardware Additions	Command-Line Interface	Valid Accounts	Exploitation for Privilege Escalation	Scripting	Network Sniffing
Supply Chain Compromise	Exploitation for Client Execution		Valid Accounts	Valid Accounts	
Valid Accounts	Scripting				

Discovery	Lateral Movement	Collection	Command and Control	Impact
File and Directory Discovery	Exploitation of Remote Services	Automated Collection	Commonly Used Port	Endpoint Denial of Service
Network Service Scanning		Data from Local System	Communication Through Removable Media	Runtime Data Manipulation
Network Sniffing			Custom Command and Control Protocol	System Shutdown/Reboot
Process Discovery			Data Encoding	
System Information Discovery			Standard Non-Application Layer Protocol	
System Network Configuration Discovery				
System Network Connections Discovery				

Figure 15. The BMW Hack 2018 MITRE ATT&CK matrix

The matrices were generated using ATT&CK v6.3 in Navigator v2.3.2.[44, 45] Detailed descriptions of attack entries can be found at https://attack.mitre.org/versions/v6/. We made the following observations from this plotting exercise:

- All four car platforms were running UN*X (aka *nix) operating systems, which is why we used the Linux layer only.
- Our analysis is only for the act stages because we do not have enough information to plot the prepare stages.
- The Navigator prepopulates entries across multiple verticals when one is selected. For example, Valid Accounts was automatically applied to *Persistence*, *Privilege Escalation*, and *Defense Evasion* after we selected it in *Initial Access*.
- For the BMW attack plot, we also included techniques from the local attack for completeness.
- All four attacks use “supply chain compromise” for initial access. This attack category includes manipulation of software update/distribution mechanisms, which is what all four attack chains used.
- All four attacks exploited vulnerabilities for initial access to the head unit and then escalation privilege to Root.
- Three of the four attack chains did some form of defense evasion (for example, the Tesla attacks disabled AppArmor).

- There were many *Discovery* activities in all four attacks. This is expected as the researchers needed to learn the systems.
- Data collected in these attacks was mostly for learning about the systems and figuring out new attack vectors.
- The command-and-control (C&C) goal for these attacks was to send malicious CAN messages to the different ECUs and get an execution.
- The degree of impact varied between each attack. All of them achieved ECU DoS and manipulated runtime data.
- We used these learnings to predict cloud services and middleware attacks. The findings were discussed in previous sections.

Connected vehicles share hardware, software, and communication protocols with the IT world. This has helped reduce development costs (that is, no need for custom-built hardware and software) and provided access to the rich development environment that already exists in the IT world. Thus, it is not surprising that when we abstract the vehicle attack chains using the MITRE ATT&CK matrix, they look uncannily like everyday IT cyberattacks.

3.6 Monetizing Cyberattacks on Connected Cars

When we think of cybercrime, the primary motivation is almost always direct or indirect financial gain, and this is not any different for cybercriminals who will start attacking connected cars. Other motivation factors for attacking connected cars and the most likely perpetrators who will carry out those attacks will be discussed in a separate section.

Monetization is an interesting topic to explore since, as of writing, as there is not a lot of news about cybercriminals who exploit connected cars to make money. Most car-hacking stories come from security researchers who have discovered and responsibly disclosed ways in which connected cars can be hacked. It is therefore worth pondering how cybercriminals will monetize hacking connected cars. To explore this inquiry, first we looked into underground marketplaces to see what cybercriminals were actively discussing about cars. We found forum threads by people selling stolen accounts from car-sharing and taxi apps, asking about OBD2 hacking, ECU tuning, and resources to learn about the CAN bus. We also found threads by people selling key fob duplicators and antennas to sniff transmissions from car keys. Overall, the results of our search were limited, and this is a good thing because it shows that cybercriminals have not yet focused their efforts on monetizing connected cars. Still, as the number of connected cars on the road multiplies, we cannot expect criminals to ignore this lucrative domain.

To monetize something, there has to be something valuable that has a direct or indirect resale value. So what is valuable in a connected car that would entice cybercriminals to hack it? We drew up a list of valuable data and goods in a connected car:

- **User PII.** The driver's mobile phone connects to the car and can share call history, address book, installed apps, and text messages, among other pieces of information. The car also knows exactly where the driver has driven to, as well as the routes that they used. Overall, the modern connected car collects a lot of user-centric PII that can be sold in underground marketplaces.
- **User data (non-PII).** Modern connected cars let apps run either natively in the head unit or via Apple CarPlay or Android Auto. These apps access cloud services and can be linked to different apps.[46] Cars like the Tesla with Sentry Mode[47] record the car's surroundings when they are activated. Hence, a lot of user data is collected and stored in the car, and this data can prove to be a lucrative theft target.
- **The car itself.** Obviously, the physical car has value, and if someone can remotely unlock, start, and steal the car,[48] then that attack could be repeatable across the same car model, or could even be extendable to other models.
- **Goods inside the car.** These are a favorite for thieves, and with hacking, unlocking the car and stealing things inside will be simplified. If an autonomous car were used to transport goods, then that shipment would be another lucrative target.
- **Driving services.** This will become a valuable commodity when autonomous vehicle services become a reality. Hacking and using cars' services for moving contraband items, committing crimes, performing anonymous movements, and other illegal acts will make criminals difficult to trace for law enforcement.
- **Stored energy.** The world is rapidly moving toward green electric cars. Car batteries have evolved over the years and now have capacities of up to 100 kilowatt-hours, which is a lot of stored energy. Some countries are experimenting with using stored energy in cars to supplement the power grid.[49] In the future, this stored battery energy in cars could potentially become a valuable commodity that is worth stealing.
- **Network and processor resources.** This includes a wide range of things such as free internet,[50] network data usage, access to cloud services, access to V2G networks, access to V2V networks, processor time, and others. Cybercriminals could install a botnet in a connected car and use network and CPU resources while the car is idle at home for the night, or they could use the car as an initial access point to hack the power grid.

These are only a few financial possible motivations, and some of them assume future developments for connected cars. Nonetheless, they help us define what is and will be valuable in connected cars. Based on our analysis, we can classify three major categories of valuable goods and services that can potentially be monetized by cybercriminals:

1. Data collected, generated, stored, and shared by the cars
2. Physical access to the cars, including driving services
3. Network and processor resources of the cars

We fully expect these three categories to expand as more exploitable use cases are found. For example, in the case of the third category, network and processor resources, when there is a middleware layer in the car to support third-party application development and those apps are not strictly controlled by the OEM, then different types of malware that can execute by API calls will be created. We will then see middleware worms, ransomware, wiper malware, bots, trojans, RATs, and many more. In sum, the whole malware monetization machine will be adapted to the connected car's ecosystem to generate revenue for the cybercriminals.

4. Fleet Management

For vehicle management to be cost-effective, the vehicles must be managed as one unit (also known as a “fleet”). A fleet can be composed of two or more similar vehicles that are managed in whole or part by a remote system. Vehicles might be similar in that they are members of a fleet of taxi cars, dump trucks, garbage trucks, ridesharing vehicles, military vehicles such as tanks, or even autonomous vehicles such as delivery drones.

There are many cost-saving opportunities when using fleets of connected vehicles that using unconnected vehicles does not offer, such as identifying what cars have the highest total cost of ownership (TCO), which ones are least used or least useful, which models are the most likely to be stolen or broken into, and which ones have the most cybersecurity risks. Armed with this knowledge, the combination of fleet vehicles with the highest return on investment (ROI) can be selected and improved on as the lowest-ROI vehicles depreciate out of the fleet.

To maximize the ROI of a fleet by reducing its TCO, various factors have to be taken into account. Some of these are tracking the availability of parts, the distance traveled by specific vehicles when driven by specific drivers, the increased cost of repair and maintenance when specific drivers use vehicles or when specific activities are performed, and finally, the maintenance of the “connected” part of the connected vehicle fleet. This final maintenance point includes networking (and breakdowns in networking) such as information security breach or fraud.

One example of lost integrity is the outage of Tesla fleets.[51] On September 23, 2020, a global outage left all Tesla cars separated from their service provider and all Tesla-provided features. Notably, this is not the first time this has happened; in 2017, an API outage left most services unavailable.[52]

Fleet management security is not a topic that can be summed up as a simple dashboard application. It is a collection of company-wide fleet business rules that are enforced by security systems. Different fleets will have different versions of what they consider a security issue, and different risk tolerances promote different security actions. We expound on the 5G telecom network, as well as its risks and security, in our paper “Securing 5G Through Cyber-Telecom Identity Federation.”[53]

In summary, a connected car can be managed as part of a fleet to reduce both operating cost and expense. Fleets and their cars depend on many sources of radio data as they travel, not all of which are trustworthy. It is also difficult for a car or a fleet management function to tell which radio data source is untrustworthy and which is not, since fleet management functions have neither visibility into a global database of trustworthy radio data sources, nor the ability to authenticate them.

One approach to addressing this fractured security landscape is a Zero-Trust Cyber-Telecom Federation, a kind of logging that is harmonized across many sources to allow common security alerting.

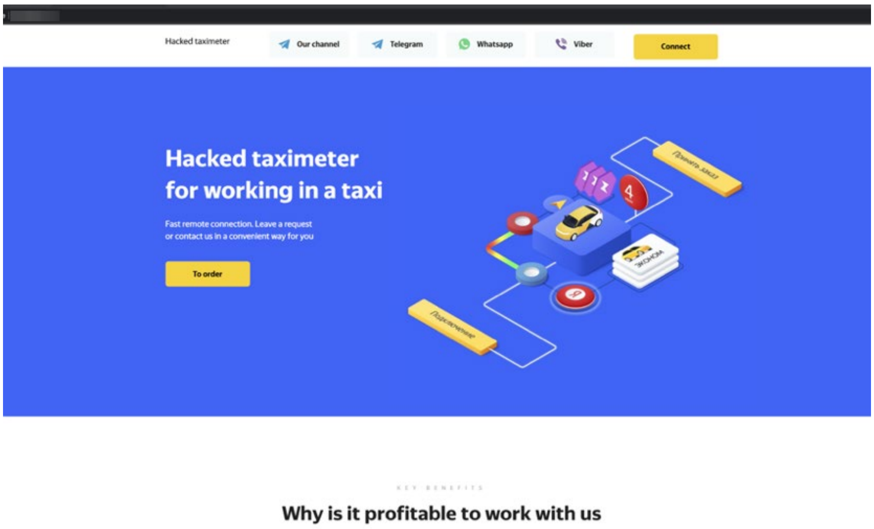
Requirements (including fleet management systems' security requirements) for this are laid out in the United Nations document "Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regard to cyber security and of their cybersecurity management systems."[54]

4.1 Fleet Management Security Risks Today

The following are several examples of fleet management safety and security issues, organized by fleet vehicle type.

4.1.1 Underground Taxicab Fraud

Fraud against fleets and their vehicles is already underway. On hacker underground forums, criminal plans and exploits on how to exploit connected vehicles are shared. A criminal exploit of connected cars includes a means for fleet fraud using fake taxicabs.



Hacked taximeter for working in a taxi

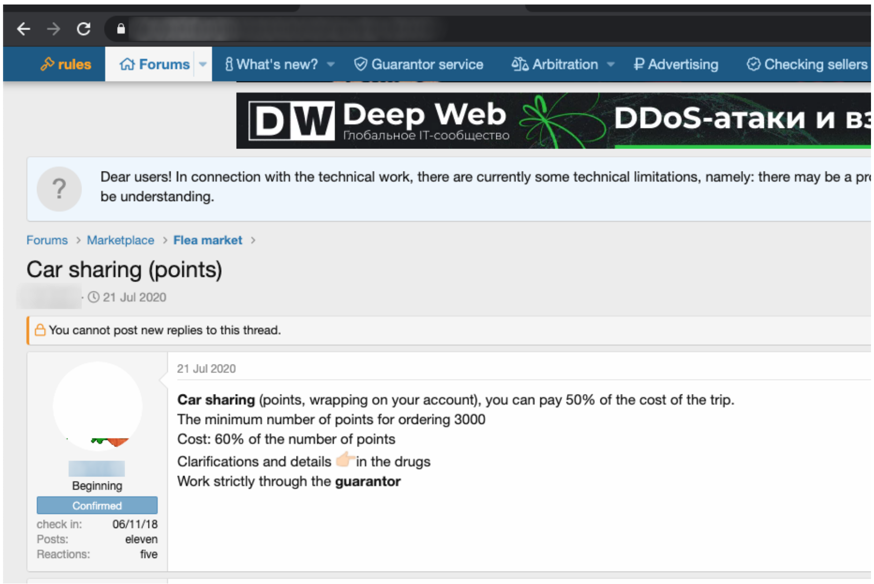
Fast remote connection. Leave a request or contact us in a convenient way for you

KEY BENEFITS

Why is it profitable to work with us

Figure 16. An underground taxi fraud service

The Taxi Simulator falsifies the activity of a legitimate taxi, presenting fake activity. When this software is used in connected taxi vehicles, it can falsify the driving and pickup history of the taxicab to make more money.



Dear users! In connection with the technical work, there are currently some technical limitations, namely: there may be a pr
be understanding.

Forums > Marketplace > Flea market >

Car sharing (points)

[redacted] · 21 Jul 2020

You cannot post new replies to this thread.

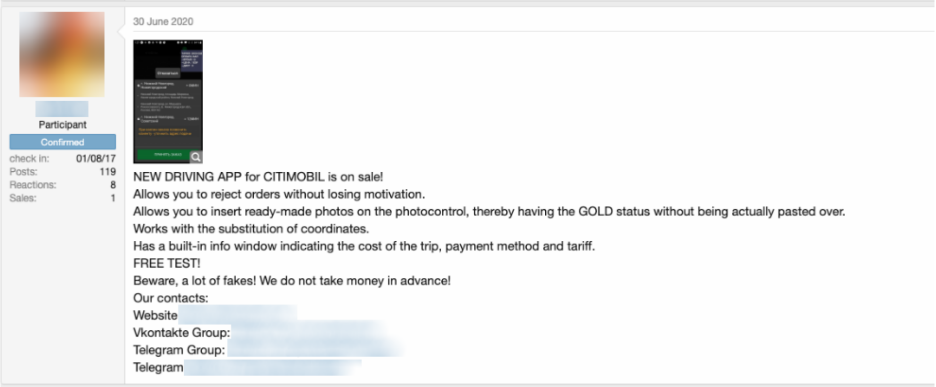
[redacted]
Beginning
Confirmed
check in: 06/11/18
Posts: eleven
Reactions: five

21 Jul 2020

Car sharing (points, wrapping on your account), you can pay 50% of the cost of the trip.
The minimum number of points for ordering 3000
Cost: 60% of the number of points
Clarifications and details 👉 in the drugs
Work strictly through the **guarantor**

Figure 17. A post in the underground showing a forum on taxi fraud

The false history made by the simulator is also useful to criminals in performing a variety of frauds against the company managing the fleet. This false history may also function as an alibi, a statement of false effort by an employee who is home asleep, or a means of hiding the fact that fleet vehicles are being used for non-work purposes. It even includes the ability to falsify drop-off points, the reputation of the driver, and what the customer sees. This can include changing the driver's photo to contain a mark designating the driver as having fake "Gold" status. When these are combined, it is not hard to imagine that they can be chained in sequence to facilitate untraceable kidnapping, or reversed to hide activities such as knowingly driving criminals or contraband items through town to untracked pickups/drop-offs.

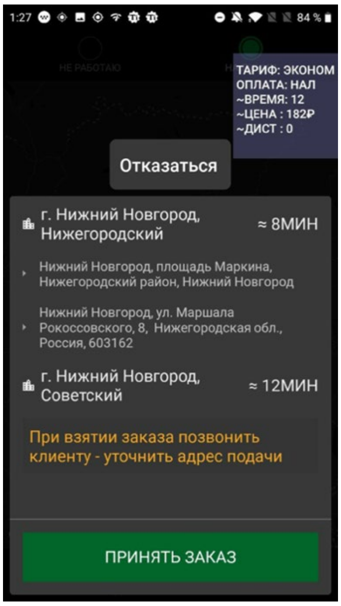

[redacted]
Participant
Confirmed
check in: 01/08/17
Posts: 119
Reactions: 8
Sales: 1

30 June 2020

NEW DRIVING APP for CITIMOBIL is on sale!
Allows you to reject orders without losing motivation.
Allows you to insert ready-made photos on the photocontrol, thereby having the GOLD status without being actually pasted over.
Works with the substitution of coordinates.
Has a built-in info window indicating the cost of the trip, payment method and tariff.
FREE TEST!
Beware, a lot of fakes! We do not take money in advance!
Our contacts:
Website [redacted]
Vkontakte Group: [redacted]
Telegram Group: [redacted]
Telegram [redacted]

Figure 18. A post in the underground showing a sale of a fraud app

The driver in one case uses an app pulled from the underground and uses it just as a ridesharing or taxi app would be used, with the exception of having much deeper control over what the company and customer see.



ТАРИФ: ЭКОНОМ
ОПЛАТА: НАЛ
~ВРЕМЯ: 12
~ЦЕНА : 182₽
~ДИСТ : 0

г. Нижний Новгород, Нижегородский ≈ 8МИН

Нижний Новгород, площадь Маркина, Нижегородский район, Нижний Новгород

Нижний Новгород, ул. Маршала Рокоссовского, 8, Нижегородская обл., Россия, 603162

г. Нижний Новгород, Советский ≈ 12МИН

При взятии заказа позвонить клиенту - уточнить адрес подачи

Figure 19. The user interface of a taxi fraud app

4.1.2 Sale and Purchase of Used Connected and Autonomous Vehicles

Connected vehicles often rely on a SIM card subscription to a mobile internet service provider (MISP) or a cellular internet company to remain connected when away from the fleet garage, which likely uses cheaper Wi-Fi. As even connected cars depreciate out of their fleets, they will have to eventually be sold, normally at city auctions). Connected cars have two value elements, however: the "connected" and the "car." When the "car" is sold, the "connection" (the mobile internet subscription) is not sold with it. When a connected car manufacturer who is reselling the MISP subscription learns that the car has been sold by its original owner, the subscription will be cancelled, making the connected fleet of used cars no longer "connected." The new owner might happen to learn of this when the subscription has not been paid by the former owner. For autonomous connected vehicles, this is especially an issue since they will no longer have their autonomous features and will no longer be able to navigate. Arguments have already been made that selling a car as a connected car (when in fact it is no longer connected) meets the conditions of fraud. Finally, cars that are no longer connected might strand drivers on the side of the road, engage other onboard security mechanisms such as lockout or Lojack for connected cars,[55] and at a minimum, lose their cost-saving benefits to fleet management ROI.

Autonomous vehicles that lose connectivity will seek a safe place to pull over. Without connectivity, the fleet management entity will only know their last position, which might be far from where the autonomous vehicle decides is a safe place to stop. If the autonomous vehicle is on the highway when this happens, it might have to travel several kilometers as it looks for a spot, making it hard to find while driving around visually searching for each of these vehicles within a circle that is kilometers wide. While a fleet's subscriptions could be transferred in some cases, the emergence of mobility-as-a-service (MaaS)[56] lengthens the supply chain and might add expensive hours to the time needed to recover the service.

4.1.3 Fleet DDoS, MiTM, Intercept, and Fraud

The means for revoking connected car connectivity also open the door to several types of connected car fleet management attacks. Since the fleet runs on cellular connectivity when driving around on streets and highways, a range of SIM cards within the vehicles are used to identify and authenticate the vehicles. By extension, these SIMs authenticate the fleet.

This range of SIMs that represents the fleet is managed by the provider of the SIM connectivity through its sales and service organization.[57] The first line for engaging this organization (that is Tesla sales and service for a Tesla fleet or Daimler for a Daimler fleet) is their customer call center. If a person with the right identifiers gets in touch with the call center, they can request for modifications to their account, with the effect of modifying some or all the SIMs and therefore the connectivity of the fleet. This is a connected car fleet management version of SIM-jacking.[58] When modified, SIM-jacking can provide the attacker with the ability to perform multimillion-dollar telecom fraud,[59] intercept traffic by replacing security models, and downgrade the security of the fleet for future attacks. To add, this can all be done OTA.[60]

Since these crimes are high-impact and scalable, even a single instance of fleet-level crime could wipe out all of a fleet's intended ROI. Fleet management security must include adequate methods of preventing social engineering of fleet management accounts, including self-provisioning accounts and online fleet management accounts.

4.1.4 Customer Impacts of Poor Fleet Business Architecture

For fleets that are customer-visible services, customer experience is a critical part of fleet management. Fleet management decisions could have an impact on customers that, if negative, bears an impact on the brand as well. Since the point of fleet management is to manage the fleet, bad security architecture and security design decisions can have a sudden and widespread brand impact. This could be traceable to technical issues such as a system update, or an executive-level interpretation of business logic that is exploitable.

A real-world example of this kind of impact is the aggressive interpretation (and consequent brand impact) of overbroad security definitions. In the case of Kari Paul,[61] a fleet management condition of "no cell coverage" was determined to indicate the car had been taken outside it's allowed area (two slightly different things). Ms. Paul, a The Guardian journalist who rented the car, had unknowingly moved outside an unpublished zone (a geofence[62]) in which the car could travel. Connected car anti-theft functions engaged, and the car refused to be driven. Ms. Paul (who rented the car for a drive through the country) was stranded on a side road in a forest. On calling the rental company to ask for help, Ms. Paul was told she would have to wait up to 24 hours to drive the car again (for a reset to take effect). The customer service representative told her and her partner to sleep in the car and wait for the reset to take effect.

This is a brand impacting design error of both fleet management security and its effect on customer safety. The source was an overbroad security condition (authorized driver leaving an area) and an aggressive response to that overbroad definition (24-hour lockout/reset). These are standard IT security design philosophies that do not work as intended in a vehicle or telecom environment.

4.1.5 Autonomous Vehicle Computer Vision Effects of Fleets and Traffic

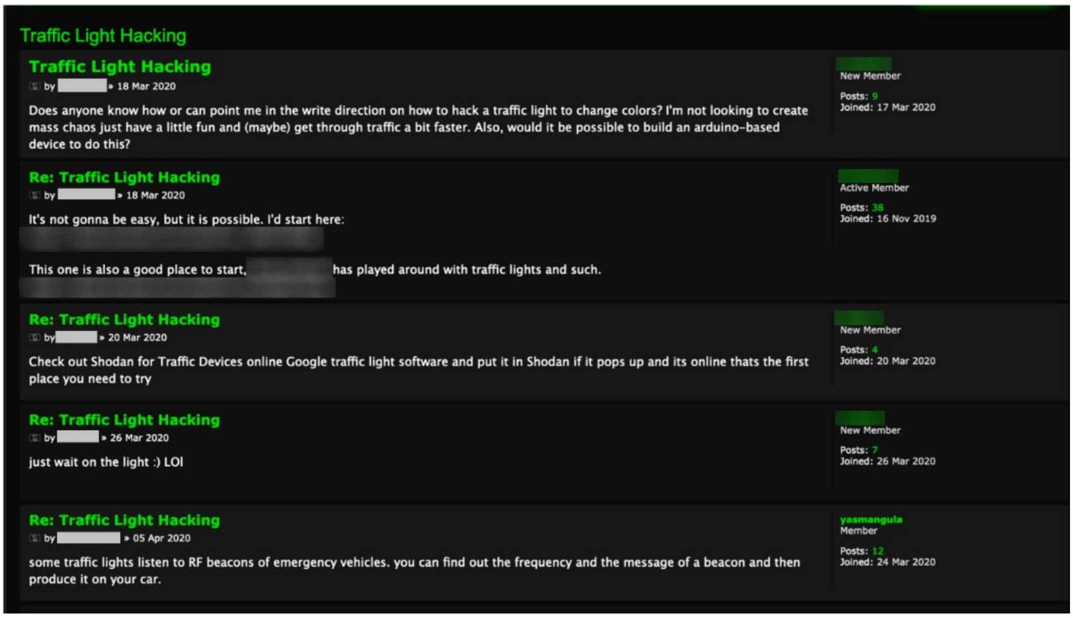
Self-driving is a defining feature of connected cars and autonomous vehicles.[63] When vehicles are driving, they use their computer vision and other navigation systems to avoid accidents and obey traffic laws.[64] This represents a vulnerability.[65] When traffic signs, road lane lines, and laws such as speed limit or direction are modified for mischief or malice, the car will obey the “legal” markings. If the markings are traffic lane lines drawn with a solid circle inside a dashed circle, it will be legal for the car to drive into the circle but illegal to drive out — this becomes a trap for autonomous vehicles. Vehicles could be trapped this way even if the trap is less obvious, such as a circle of “detour right” signs. Since impatient humans who encounter this will simply apply their judgment and “break the law” to escape, eventually, the trap will collect all the autonomous vehicles in the area, if such a trap is large enough and placed in an effective spot.

Similarly, if a marker or piece of electrical tape is added to a traffic speed sign,[66] the vehicle will read the sign and accelerate to unsafe speeds. If delivery vehicles and trucks[67] are involved, they will receive traffic tickets every time that they are clocked speeding, as well as driving at a speed that has been determined to be unsafe.

4.1.6 Effects of False Telemetry on Connected and Unconnected Vehicles

In 2019, the artist Simon Weckert[68] performed an original think piece that consisted of placing 99 activated phones in a small red wagon and walking them across a bridge in Germany. His experiment, a video[69] which he uploaded on Youtube, showed that Google Maps interpreted the wagon full of phones as many vehicles driving slowly. The ML models of Google Maps interpreted this as a high-traffic condition and rerouted real-world vehicles away from Mr. Weckert’s wagon. This incidentally caused increased real-world traffic on the other bridge nearby.

This concept also illustrates real-world abuse. Through the use of several of these wagons involving hundreds of phones, the traffic of an entire road could be arbitrarily redirected to or from different streets. One long-standing trope in movies (as seen in “The Italian Job”[70]) involves a hacker modifying streetlights from red to green (thus enabling escape) or from green to red (thereby hindering police response). The following screenshot from an underground platform illustrates the demand for this service.



Traffic Light Hacking

Traffic Light Hacking
by [redacted] » 18 Mar 2020

Does anyone know how or can point me in the write direction on how to hack a traffic light to change colors? I'm not looking to create mass chaos just have a little fun and (maybe) get through traffic a bit faster. Also, would it be possible to build an arduino-based device to do this?

[redacted]
New Member
Posts: 9
Joined: 17 Mar 2020

Re: Traffic Light Hacking
by [redacted] » 18 Mar 2020

It's not gonna be easy, but it is possible. I'd start here:
[redacted]

This one is also a good place to start, [redacted] has played around with traffic lights and such.
[redacted]

[redacted]
Active Member
Posts: 38
Joined: 16 Nov 2019

Re: Traffic Light Hacking
by[redacted] » 20 Mar 2020

Check out Shodan for Traffic Devices online Google traffic light software and put it in Shodan if it pops up and its online thats the first place you need to try

[redacted]
New Member
Posts: 4
Joined: 20 Mar 2020

Re: Traffic Light Hacking
by [redacted] » 26 Mar 2020

just wait on the light :) LOl

[redacted]
New Member
Posts: 7
Joined: 26 Mar 2020

Re: Traffic Light Hacking
by [redacted] » 05 Apr 2020

some traffic lights listen to RF beacons of emergency vehicles. you can find out the frequency and the message of a beacon and then produce it on your car.

yasmangula
Member
Posts: 12
Joined: 24 Mar 2020

Figure 20. Forums in the underground on hacking traffic lights

The trivial complexity of the red wagon approach used by Weckert shows that a hacker does not need to know anything about the network, but it will redirect traffic at their whim. Several red wagons would push traffic in a smaller range of chosen paths, allowing an attacker greater control. Certainly, this attack could also be performed from the back seat of a car or through any approach with more stealth.

A deeper evolution of this exists as a concept. Mass malicious ridesharing, which involves using many ridesharing apps, emulated phones, and stolen credit cards to swamp an area with rideshare vehicles, can have an extreme effect. If these vehicles are autonomous as well, portions of a fleet could be delayed for some time as they contribute to gridlock.[71]

A conclusion we can draw from these examples and others like them is that business rules must be reviewed with the same or greater level of diligence than that normally reserved for firewall rules, intrusion detection rules, and similar. When managing autonomous and connected vehicles with power over the actions of humans in real-world circumstances, more attention is needed to the negative consequences of planning. The application of a “hacker mindset” to business rules before implementation in technology should be performed so obvious malicious abuses can be detected early when remediation is still less costly and implementation has not yet begun.

4.1.7 Connected Car Self-Provisioning

A means for reducing the manufacturer cost needed to set up connected car connectivity is the use of a self-provisioning portal. A website like this allows customers to manage their accounts and mobile device SIM cards, thus also letting them add or remove vehicles. For fleets, this would be the management of the connectivity (SIM cards) of many vehicles at once.

As a website, it is potentially vulnerable to all the security issues that websites face, such as cross-site scripting, buffer overflow, and others. As an effect of such vulnerabilities, access could be granted to the accounts that are managed by the website: in this case, the fleets of connected vehicles. In the case of SIM-jacking,[72] one of the ways that an attacker can gain control of SIMs is to control exactly this kind of portal. By exploiting the functions of the portal's account management, SIMs and their phone numbers can be reassigned (for controlling the fleet), redirected (for surveillance or malware infection invisible to network-based security), or used to initiate calls to expensive foreign attacker-controlled phone number destinations such as 1-900 numbers. Detailed features such as car heating or others can be activated or deactivated remotely as well.[73]

It can be speculated that a feature that might be abused through this method includes police car coronavirus decontamination,[74] providing false assurances of safety, or even "cooking" detainees through heat exhaustion.

Depending on the features of the fleet management application, it can also be used to designate a car (or all cars) as stolen, or to disable or impair them.

4.2 Fleet Management Security States for Connected Cars

For the fleet to be capable of moving freely, fleet management of connected vehicles must work easily across many types of cell towers and phone companies. This easy and secure fleet management must work even when the car moves from one cell tower to another (having mobility[75]), and from cellular to Wi-Fi (being nomadic[76]).

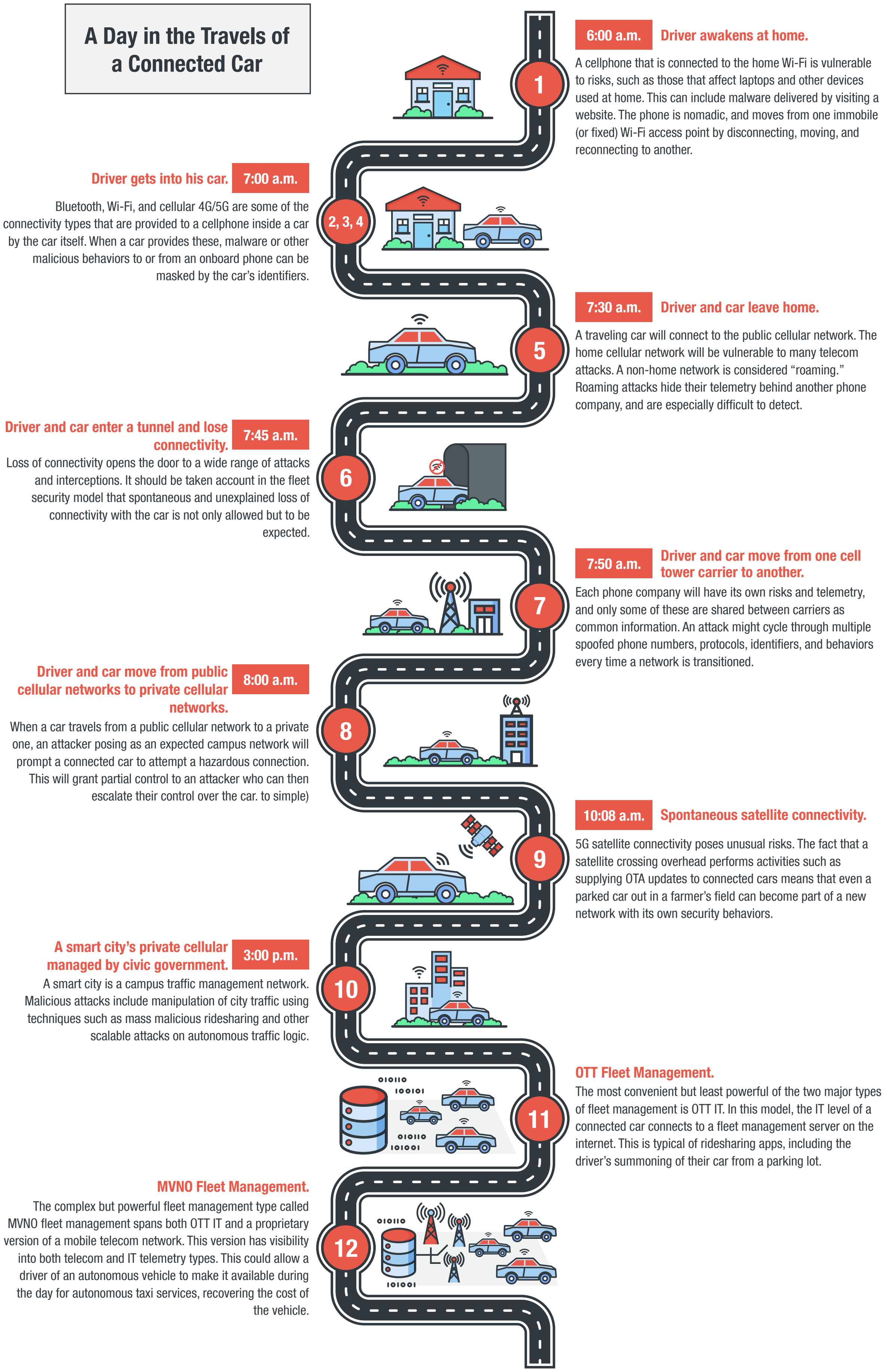
There are two general categories of fleet management. The first is the ridesharing sort whose examples include ridesharing companies such as Uber and Lyft, as well as self-controlled fleets of police cars, transport trucks, delivery vehicles, self-driving warehouse vehicles, and other similar vehicles. These are server-based IT services that lay over-the-top[77] (OTT) telecom while having little visibility into telecom infrastructure. They tend to have mobile phone apps or something similar on the consumer end, and a network of IT servers on the backend. They are very dependent on the security of the telecom network both for the app and the connectivity of the vehicles in the fleet. Security is performed through certificates and the like.

The second general category of fleet management is the virtual phone company of Tesla, ZOOX,[78] or Daimler. These are vertically integrated and have both the ridesharing OTT infrastructure as well as the complex telecommunications supporting it. While the initial investment is much higher, this approach provides both deep visibility and cost savings for the entire fleet through increased data accuracy and automation. Automated control is much deeper for this model as well and is necessary for autonomous road vehicles.[79] This approach also provides additional control and security for the OTT layer by controlling the telecommunications layer beneath it. Security is performed through a combination of SIM card telecom network security and OTT traditional IT methods. The telecommunications name for this type of approach is virtual mobile virtual network operator (aka virtual MVNO[80]).

For the OTT approach, typical IT risks such as hacking, DDoS, DNS hijacking,[81] malware, and other similar traditional attacks are most likely. The OTT approach to fleet management is also vulnerable to MVNO attacks; however, it can only mostly detect the simplest (and arguably clumsiest) attacks. These clumsy attacks will typically all look like lost mobile connectivity of the kind that are caused by mobile network coverage gaps. For the MVNO approach, traditional telecom risks such as SIM-jacking, long-distance fraud,[82], wiretap,[83] and redirection of OTT traffic (that is, for industrial espionage or tracking of individuals) are most likely. OTT attacks such as traditional IT attacks also work against the MVNO method of fleet management.

Both the OTT and MVNO methods identify security events by performing analytics on logging and can follow a course of action similar to this equation: When A + B = C, then do D. To illustrate, when known profiles (A), such as malicious patterns of behavior (B), are found (C), the security system can take a security action (D). In a mobility environment such as that used by a connected car, these behaviors (A) might be spread across multiple phone companies, IP addresses, phone numbers, and other identifiers. When these malicious behaviors (B) are spread around across multiple versions of A, they do not tell a recognizable story (C) that traditional security systems can identify as malicious. In a complex mobile network including the behaviors of a moving connected car, A, B, and C are often not calculated, so D might never occur as a result.

An example can be illustrated by following the activities of a connected car through a typical day in which there are multiple sources of information and therefore multiple versions each of A and B indicating many different use cases that would have to be diligently assembled (that is, federated) to allow the detection of malicious behavior (C). A connected car's day is described in the following diagram, using as an example just one vehicle that might be a member of a fleet. The fleet management system is therefore responsible for addressing the security of all the different potential combinations of these values of A and B so that C can be calculated. These combinations of A and B can be arrived at through a type of tightly unified logging called Cyber-Telecom (CyTel) Identity Federation. More information on this CyTel Federation will be discussed in the section on solutions.



A Day in the Travels of a Connected Car

6:00 a.m. **Driver awakens at home.**

A cellphone that is connected to the home Wi-Fi is vulnerable to risks, such as those that affect laptops and other devices used at home. This can include malware delivered by visiting a website. The phone is nomadic, and moves from one immobile (or fixed) Wi-Fi access point by disconnecting, moving, and reconnecting to another.

Driver gets into his car. **7:00 a.m.**

Bluetooth, Wi-Fi, and cellular 4G/5G are some of the connectivity types that are provided to a cellphone inside a car by the car itself. When a car provides these, malware or other malicious behaviors to or from an onboard phone can be masked by the car's identifiers.

7:30 a.m. **Driver and car leave home.**

A traveling car will connect to the public cellular network. The home cellular network will be vulnerable to many telecom attacks. A non-home network is considered "roaming." Roaming attacks hide their telemetry behind another phone company, and are especially difficult to detect.

Driver and car enter a tunnel and lose connectivity. **7:45 a.m.**

Loss of connectivity opens the door to a wide range of attacks and interceptions. It should be taken account in the fleet security model that spontaneous and unexplained loss of connectivity with the car is not only allowed but to be expected.

7:50 a.m. **Driver and car move from one cell tower carrier to another.**

Each phone company will have its own risks and telemetry, and only some of these are shared between carriers as common information. An attack might cycle through multiple spoofed phone numbers, protocols, identifiers, and behaviors every time a network is transitioned.

Driver and car move from public cellular networks to private cellular networks. **8:00 a.m.**

When a car travels from a public cellular network to a private one, an attacker posing as an expected campus network will prompt a connected car to attempt a hazardous connection. This will grant partial control to an attacker who can then escalate their control over the car. to simple)

10:08 a.m. **Spontaneous satellite connectivity.**

5G satellite connectivity poses unusual risks. The fact that a satellite crossing overhead performs activities such as supplying OTA updates to connected cars means that even a parked car out in a farmer's field can become part of a new network with its own security behaviors.

A smart city's private cellular managed by civic government. **3:00 p.m.**

A smart city is a campus traffic management network. Malicious attacks include manipulation of city traffic using techniques such as mass malicious ridesharing and other scalable attacks on autonomous traffic logic.

OTT Fleet Management.

The most convenient but least powerful of the two major types of fleet management is OTT IT. In this model, the IT level of a connected car connects to a fleet management server on the internet. This is typical of ridesharing apps, including the driver's summoning of their car from a parking lot.

MVNO Fleet Management.

The complex but powerful fleet management type called MVNO fleet management spans both OTT IT and a proprietary version of a mobile telecom network. This version has visibility into both telecom and IT telemetry types. This could allow a driver of an autonomous vehicle to make it available during the day for autonomous taxi services, recovering the cost of the vehicle.

Figure 21. A day in the travels of a connected car

The above use cases can be visualized in the following table:

	Use Case	Mobility	Roaming	Nomadic	OTT IT	Cellular (Public)	Cellular (Non-Public)
Security State 1	Phone – home Wi-FI	No	No	Yes	Yes	No	No
Security State 2	Phone – in-car Wi-Fi	No	No	Yes	Yes	No	No
Security State 3	Phone – in-car Bluetooth	No	No	Yes	Yes	No	No
Security State 4	Phone – in car 4G/5G	Yes	No	Yes	No	No	Yes
Security State 5	Car – public cellular	Yes	Yes	No	No	Yes	No
Security State 6	Tunnel – no connectivity	Yes	Maybe, depending on the tower at the tunnel exit	Yes, upon exiting the tunnel	Maybe, depending on the tower at the tunnel exit	Maybe, depending on the tower at the tunnel exit	Maybe, depending on the tower at the tunnel exit
Security State 7	Car – public cellular	Yes, telemetry unique to the new carrier	Yes	No	No	Yes, telemetry unique to the new carrier	No
Security State 8	Car – campus cellular	Yes	Yes	Yes	Yes	Yes	Yes
Security State 9	Satellite connectivity	Yes	Yes	Yes	Yes	Yes	Yes
Security State 10	Car – Smart city traffic engineering	Yes	Yes	Yes	Yes	Yes	Yes
Security State 11	OTT fleet management	Yes	Yes	Yes	Yes	Yes	Yes
Security State 12	MVNO fleet management	Yes	Yes	Yes	Yes	Yes	Yes

Table 1. Telemetry states affecting fleet management security

4.3 Breakdown of Connected Car States and Risks

The following are the security implications of each of the mentioned states.

Security State 1 – Phone – Home Wi-Fi

A cellphone connected to a home Wi-Fi is vulnerable to risks like those that affect laptops and other devices used at home. These risks can include malware delivered by visiting a website (B.1). The phone is nomadic and moves from one immobile or fixed WI-FI access point by disconnecting, moving, and reconnecting to another. The telemetry generated by the phone is of one kind only, and the IP address is stable. Logs generated from this connection will have a single type of telemetry. Detection of malicious activity is likely since IT security products are often successful in this detection.

Connected cars are improved after the sale using updates, like how cellphones are updated. Some of these updates are huge and would therefore be expensive to load via 4G/5G SIM connectivity. When a car is at its owner's home or the fleet's garage, it can install these updates via the owner's Wi-Fi, like how phones can be updated today. Attacks leveraging home Wi-Fi for malicious updates and car-specific malware are often hidden from connected car fleet management security. To be effective, fleet management security must see this traffic as well, federating it along with cellular traffic from multiple sources including public and campus cellular, mobile and nomadic OTT, and opportunistic attacks from the various networks that the vehicle and its passengers might encounter.

Like home Wi-Fi, homes in some regions can obtain a device called "femtocell," a small cell tower used to provide improved and reduced cost cellular coverage at home. When a connected car is at an owner's home, it can be configured to connect to this immobile (fixed) cellular instead of to the Wi-Fi. When this is true, both the car and the femtocell can be attacked by true telecom attacks, but the vehicle's telemetry will be masked by the femtocell's metadata. Federated logging will need visibility to both IT telemetry, such as IP addresses and telecom telemetry, such as phone numbers.

Security State 2 – Phone – In-Car Wi-Fi

A cellphone connected to a car's onboard Wi-Fi will be masked from upstream systems and provide two sets of telemetry for traditional IT malicious activity. One set will be for the car's Wi-Fi connectivity (B.2), and the other set behind it will be for the cellphone. Security systems attempting to identify malicious activity must be able to identify which is which and create a unified story to identify if malice is present. This requires identity federation[84] (SSO[85] like harmonized logging) between the nomadic phone and the car's nomadic connectivity to the home Wi-Fi. All three (including the home Wi-Fi) must be federated to create a single identifiable set of behaviors for malicious activity. When these three are federated, malicious behavior becomes visible to security systems and capable of triggering a security response. As nomadic devices, telemetry from all three devices will be consistent and stable.

Security State 3 – Phone – In-Car Bluetooth

Malware capable of spreading via Bluetooth (B.3) is another type of telemetry that security systems must federate for logging activities for these systems to be able to detect its behaviors. Bluetooth is also capable of interacting with the SIM cards of mobile devices.[86] This is a way of hijacking SIM-enabled devices such as connected cars and phones.

Security State 4 – Phone – In-Car 4G/5G

Some cars have 4G/5G connections coming from a device that acts as a small cell tower inside the vehicle itself.[87] When a driver enters a car of this type, their preconfigured phone will connect to the car's internal cellular connection, which sits behind the car's outward-facing 4G/5G cellular connection. From this point, the phone is vulnerable to true telecom attacks, including those that spoof a car's cellular connection and allow attacks on the phone, as well as attacks from the phone on the car. Since the car passes the phone's connectivity out to the telecom domain and from there to the internet, federated fleet management logging between telecom telemetry and IT telemetry is necessary to see malicious behaviors spread across these multiple networks and protocols.[88]

Security State 5 – Car – Public Cellular

As a connected car travels away from home, it will connect to a public cellular network. If this is the same network that provides connectivity to the connected car's SIM card, the car will be on its non-roaming home cellular network and will then be vulnerable to many telecom attacks. If the network that the car connects to is not a home cellular, and the car's connectivity is provided by another phone company, this is called roaming. By their nature, roaming attacks hide their telemetry behind another phone company. Roaming attacks are especially difficult to detect.[89] Federating telecom telemetry across multiple phone companies is logistically complex. This information is held in part by the telecom carrier trade association called the GSM Association (originally known as Groupe Spécial Mobile).[90] Federating with GSM Association roaming abuse data supplies valuable insight into other abuses including long-distance fraud performed from within the car, the presence of some kinds of SIM-jacking, and various other telecom-level redirects invisible to OTT IT fleet management security.

Security State 6 – Tunnel – No Connectivity

This use case is a critical distinction. The total loss of connectivity opens the door to a wide range of attacks and interceptions. In nomadic devices, connectivity is lost when the device is purposely disconnected from a fixed network. In mobility, it is lost when the device is unintentionally disconnected from a mobile network. It should be taken into account in the fleet security model that spontaneous and unexplained loss of connectivity in the car is not only allowed but to be expected. This is a source of risk, however, since telecom attacks redirecting connectivity to a malicious telecom OTT network will blind traditional OTT IT security. The traditional OTT IT security will see only that an allowed or expected event occurred (lost connectivity). In the meantime, telecom attacks can modify the security profile of a car and restore connectivity with normal security none the wiser.

Security State 7 – Car – Public Cellular

While public cellular represents a class of telemetry and risk, each phone company will have its own risks and telemetry, and only some of these are shared between carriers as common information. When a car is traveling down a highway, for example, fleet management security that federates telecom and OTT IT logging must recognize and unify malicious telecom activity. This unity must happen across multiple telecom instances representing parts of a single multicarrier session, across which malicious activity might change its telemetry behaviors and identifiers even when continuing the same malicious behavior. One example would be car-originated mobile spam,[91] in which the malicious source code might cycle through multiple spoofed phone numbers each time a network is transitioned.

Security State 8 – Car – Campus Cellular (Non-Public Network)

When a car travels from a public cellular network managed by a cellphone company to a non-public network (an NPN[92] or campus network), the fleet management security model must tolerate yet another hazardous condition. The transition between the public network and the NPN campus network is typically not federated; this means that posing as an expected campus network will prompt a connected car to attempt a connection. This will of course grant partial control to an attacker who can then escalate their control over the car.[93] Campus networks combine the risks of nomadic, mobile, OTT, roaming, and non-public and public networks. NPN campus cellular networks are typically the size of a university or industrial campus but can be as large as a smart city.

Security State 9 – Car – Satellite Connectivity (Campus Cellular From Space)

Satellite connectivity of the sort proposed by Jeff Bezos' Blue Origin[94] and SpaceX's Starlink[95] pose unusual risks. The risk profile is like that of a campus network. The fact that a satellite crossing overhead can perform activities such as supplying OTA updates to connected cars means that even a car parked outside in a farmer's field can become part of a new network. This state means that the operating model of connected cars must always be ready to suddenly receive information from a new network, even if the local cell towers are already connected. This opens the connected car fleet management model to even space-based risks[96] as well as risks that pose as originating from space. As a type of campus network, satellite-based traffic management networks combine the risks of nomadic, mobile, OTT, roaming, and non-public and public networks.

Security State 10 – Car – Smart City Traffic Engineering (Campus Cellular at City Scale)

A smart city is a campus network that provides traffic management input for connected vehicles based on known, predicted, and planned vehicle-traffic-impacting activities.[97] These can include the manipulation of city traffic using techniques such as mass malicious ridesharing.[98] As a type of campus network, smart city traffic management networks combine the risks of nomadic, mobile, OTT, roaming, and non-public and public networks.

Security State 11 - Fleet Management – OTT (Information Technology)

One of the two major types of fleet management is OTT IT. In this model, the IT level of a connected car connects to a fleet management server on the internet. This model is typical of ridesharing apps and their infrastructure. The security of this model is often standards-based according to ISO 21434. More details on this ISO standard and Trend Micro's insights and recommendations can be found in our paper, "ISO/SAE 21434: Setting the Standard for Connected Cars' Cybersecurity."[99]

Security State 12 – Fleet Management – MVNO (Information Technology and Telecom)

The second of the two major types of fleet management security is derived from a common set of security rules that spans both OTT IT and a proprietary version of a telecom network. This version has visibility into both telecom and IT telemetry types. This is a collection of risks called Cyber Telecom (CyTel).[100] Examples include long-distance fraud, wiretapping, and interception of navigation and tracking information that would allow hijacking of a vehicle.

5. Threat Model Connected Cars

Based on the different remote attacks on connected cars, it is evident that there is a need to provide guidance that will help protect connected cars against remote hacks. To that end, we created a threat model for connected cars.

With our threat model, developers and car manufacturers can better assess, identify, classify, and quantify the risks that come with each threat in the threat model.[101] It aims to help them create more secure connected cars from the very early stages of the software development life cycle. By shedding light on connected car attack vectors, their risk levels, and the important security observations based on them, we hope to help keep connected cars running smoothly and securely.

5.1 Connected Car Attack Vectors

The connected car ecosystem is extremely complex, with potentially millions of endpoints and end users. The complexity of this ecosystem, with its immense size and many functions, makes for large and at times unpredictable attack surfaces. Although they primarily communicate wirelessly, connected cars heavily depend on the networked ITS infrastructure for communications. In our threat modeling exercise, we focused on attacks that could be launched remotely against and/or from the victim vehicles. The following, in no particular order, are the connected car attacks that we identified:

- Spoofing V2X messages being broadcast to the ecosystem
- Passively sniffing V2X messages being broadcast to the ecosystem
- Sending incorrect or improper commands to back-end ITSs
- Sending MitM communications and false data to back-end ITSs
- Sniffing network traffic between a connected car and back-end ITSs
- Remotely transmitting and installing malicious firmware and/or apps
- Electronically jamming wireless transmissions to disrupt operations
- Performing an MitM attack with wireless transmission to intercept and modify car data
- Exploiting vulnerabilities in software, hardware, operating systems, and protocols

- Using RF modules to access the head unit via complex exploit chains
- Remotely hijacking vehicles via compromised CAN bus
- Dumping firmware to recover credentials and configurations
- Installing malicious third-party apps in a connected car’s infotainment system
- Deleting local files in a compromised connected car’s file system
- Attacking via a malicious app installed on a connected mobile phone
- Electronically jamming a connected car’s safety systems, such as radar and lidar
- Attacking the camera system’s image processing with specially crafted visuals
- Installing malware or spyware in a connected car
- Identifying and abusing device misconfigurations
- Discovering and abusing vulnerable remote systems using Shodan, a search engine for internet-connected devices[102]
- Conducting social engineering attacks such as creating fake RDS-TMC messages, phishing, and map poisoning
- Launching distributed denial-of-service (DDoS) attacks using a compromised ITS infrastructure
- Launching DDoS attacks on an ITS infrastructure so that it fails to respond to requests
- Credential brute-forcing and abusing weak authentication methods
- Injecting malicious scripts via malvertising
- Performing traditional attacks such as SQL (Structured Query Language) injection,[103] cross-site scripting (XSS),[104] session hijacking,[105] and DNS (Domain Name System) spoofing[106]
- Pivoting a connected car as a trusted entry point to the V2X network
- Compromising a third-party software supply chain to push malicious updates
- Scanning the V2X network from a connected car to discover the topology and nodes

There are overlaps between some of these attacks. An example is the overlap between spoofing V2X messages to the ecosystem and sending incorrect or improper commands to back-end ITSs. In reality, they are different because not all spoofed messages are malicious, but both types of messages might ultimately achieve malicious results.

5.2 The DREAD Threat Model

One of the many benefits of threat modeling is that it allows organizations to look at security in a structured way, enabling them to analyze each possible threat and effectively identify which threats to prioritize in terms of mitigation.[107] The DREAD threat model can be used to perform a qualitative risk analysis,[108] which is opinion-based in that it uses rating values to evaluate the risk level of a threat. We arrived at the risk rating for a given threat by asking the following questions:

- **Damage potential:** How great is the damage to the assets?
- **Reproducibility:** How easy is it to reproduce the attack?
- **Exploitability:** How easy is it to launch an attack?
- **Affected users:** As a rough percentage, how many users are affected?
- **Discoverability:** How easy is it to find an exploitable weakness?

We used the threat rating table shown in Table 2 for our connected car risk analysis.

	Rating	High (3)	Medium (2)	Low (1)
D	Damage potential	The attacker subverts the system and can inflict serious damage.	The attacker subverts the system and can inflict moderate damage.	The attacker subverts the system and can inflict minor damage.
R	Reproducibility	The attack can be reproduced every time.	The attack can be reproduced, but only within set limitations.	The attack is very difficult to reproduce, even with full knowledge of the security hole.
E	Exploitability	The attack requires little or no knowledge of the system in order to exploit it.	The attack requires a skilled operator with fundamental knowledge of the system in order to exploit it.	The attack requires an extremely skilled operator with in-depth knowledge of the system in order to exploit it.
A	Affected users	The majority of everyday users will be affected by the attack.	A good-sized portion of everyday users will be affected by the attack.	A very small percentage of everyday users will be affected by the attack.
D	Discoverability	Published information readily explains the attack. Vulnerabilities are found in the most commonly used applications and systems.	Vulnerabilities are not common and are found only in certain applications and systems. The attack requires skills to discover exploitable weaknesses.	Vulnerabilities are difficult to find and, if found, are very difficult to weaponize. It is extremely difficult to attack applications and systems.

Table 2. The DREAD threat model

The risk rating is calculated by adding the rating values based on the answers to the DREAD questions for a given threat. The overall risk is rated as:

- High if the score is between 12 and 15.
- Medium if the score is between 8 and 11.
- Low if the score is between 5 and 7.

5.3 Measuring the Risks of Attacks on Connected Cars

For each of the connected car attack vectors we have identified, we assigned scores for realistic extreme scenarios and calculated the risk rating using the DREAD threat model, as shown in Table 2.

Attack vector	D	R	E	A	D	Rating	Remarks
Remotely transmitting and installing malicious firmware and/or apps	3	1	1	1	1	Low	Can the attackers remotely download and flash the ECU firmware after they get access to the head unit?
Using RF modules to access the head unit via complex exploit chains	3	1	1	1	1	Low	RF units are Bluetooth, Wi-Fi, and eSIM.
Remotely hijacking vehicles via compromised CAN bus	3	1	1	1	1	Low	This assumes that the attackers are already inside the vehicle. Can they compromise an ECU?
Deleting local files in a compromised connected car's file system	1	1	1	1	1	Low	The firmware is in nonvolatile memory. Some dynamically created user files can be deleted.
Installing malware or spyware in a connected car	2	2	1	1	1	Low	This is difficult to execute because different car models have different network architectures.
Spoofing V2X messages being broadcast to the ecosystem	2	2	2	2	2	Medium	Spoofing is the act of falsifying the identity of the sender in order to gain an illicit advantage.
Passively sniffing V2X messages being broadcast to the ecosystem	1	3	2	1	3	Medium	Data traffic broadcast to or from other cars and the ITS road infrastructure is sniffed.
Sending incorrect or improper commands to back-end ITSs	3	1	2	2	2	Medium	A back-end ITS system is where a connected car sends commands, e.g., a traffic light controller.
Sending MitM communications and false data to back-end ITSs	3	1	2	2	2	Medium	This is when the connected car is reporting roadway or car data to the ITS back-end system.
Sniffing network traffic between a connected car and back-end ITSs	1	3	2	1	3	Medium	Ingress or egress network traffic between the connected car and the back-end ITS system is sniffed.
Performing an MitM attack with wireless transmission to intercept and modify car data	3	1	2	1	2	Medium	A GSM base station or a cloud source is where the connected car receives data from external sources.

Attack vector	D	R	E	A	D	Rating	Remarks
Dumping firmware to recover credentials and configurations	2	2	2	1	2	Medium	The firmware package typically contains updates for multiple ECUs.
Installing malicious third-party apps in a connected car's infotainment system	1	2	2	1	2	Medium	Apps are installed via a mobile network or the TCB (from the OEM).
Attacking via a malicious app installed on a connected mobile phone	2	3	3	1	2	Medium	This assumes that the malicious app is on the mobile phone that is connected via Bluetooth or Wi-Fi to the head unit.
Exploiting vulnerabilities in software, hardware, operating systems, and protocols	3	1	1	2	3	Medium	The head unit typically runs a custom Linux kernel and other common tools such as the WebKit browser.
Attacking the camera system's image processing with specially crafted visuals	2	2	1	1	3	Medium	The firmware is reverse-engineered to find flaws in the image processing logic, or trial and error is used.
Identifying and abusing device misconfigurations	3	2	2	2	2	Medium	The provisioning files for OTA software updates are captured to figure out the running services.
Conducting social engineering attacks such as creating fake RDS-TMC messages, phishing, and map poisoning	1	2	2	1	2	Medium	RDS-TMC could display fake roadway alerts in the head unit and confuse the driver.
Credential brute-forcing and abusing weak authentication methods	2	3	2	1	3	Medium	This could be used to compromise vehicle Wi-Fi or privileged accounts in the operating system.
Injecting malicious scripts via malvertising	1	2	3	2	3	Medium	This could happen via an installed app or via a webpage loaded in the head unit's browser.
Performing traditional attacks such as SQL (Structured Query Language) injection, cross-site scripting (XSS), session hijacking, and DNS (Domain Name System) spoofing	2	1	2	1	2	Medium	This primarily targets the head unit or middleware that runs in the car.
Pivoting a connected car as a trusted entry point to the V2X network	1	2	2	1	2	Medium	The connected car is a trusted endpoint that could be abused to get into the ITS infrastructure.
Compromising a third-party software supply chain to push malicious updates	2	1	2	2	2	Medium	This targets third-party apps installed in the head unit or are running on top of middleware.
Scanning the V2X network from a connected car to discover topology and nodes	1	3	3	1	3	Medium	This is an extension of passive scanning. The goal here is to discover ITS infrastructure topology.
Electronically jamming a connected car's safety systems, such as radar and lidar	3	3	3	1	3	High	Lidar is a detection system based on the principle of radar, but uses light from a laser.
Electronically jamming wireless transmissions to disrupt operations	3	3	3	1	3	High	2G, 3G, LTE, 4G, 5G, Wi-Fi (802.11p), RDS-TMC, or cellular-satellite connectivity is jammed.
Discovering and abusing vulnerable remote systems using Shodan, a search engine for internet-connected devices	3	3	3	2	3	High	This technique could be used to find exposed ITS infrastructures that could then be compromised.

Attack vector	D	R	E	A	D	Rating	Remarks
Launching distributed denial-of-service (DDoS) attacks using a compromised ITS infrastructure	2	2	2	3	3	High	The goal is to overwhelm the connected car with excess data from the ITS infrastructure.
Launching DDoS attacks on an ITS infrastructure so that it fails to respond to requests	3	3	3	3	3	High	The goal is to knock the ITS infrastructure offline so that the connected car cannot send or receive messages.

Table 3. Connected car threat modeling using the DREAD threat model

Based on the results of our connected cars threat modeling exercise, we make the following observations:

- Of the 29 identified attacks on connected cars, about 66% are medium-risk, about 17% are low-risk, and another approximate 17% are high-risk.
- The attacks classified as low-risk are the ones that require a high level of technical skills and an in-depth knowledge of the connected car platform.
- The low-risk attacks, given their specialized nature, would realistically affect only a small percentage of everyday connected cars since the attacks are difficult, albeit not impossible, to execute on a massive scale.
- Surprisingly, malware attacks on connected cars are rated low-risk. This is probably because an attacker needs to understand the low-level electrical/electronic (E/E) architecture of a targeted car prior to launching an attack. It is also not easy to port malware from one car architecture to another as their implementations will be vastly different.
- The high-risk attacks are attacks that require only a limited understanding of the inner workings of a connected car and can be pulled off by a low-skilled attacker, such as the electronic jamming of RF modules.
- The high-risk attacks also include DDoS attacks and the discovery of exposed services and servers using network-scanning services such as Shodan. Even though the 2015 Jeep hack research found the D-Bus daemon running on the exposed port 6667 in Sprint's vehicle network, it still took a high degree of technical prowess to go from finding the D-Bus daemon to compromising the ECUs. Launching a DDoS attack on an exposed ITS infrastructure is comparatively easier and could have devastating consequences especially if connected cars rely on the ITS infrastructure for driving decisions.
- Sensational attacks such as installing malicious firmware over the air, remotely hijacking vehicle controls, sending incorrect or improper commands to the ITS back end, and sending spoofed V2X messages are rated medium- or low-risk. These attacks are difficult to execute because the devices and the systems are not readily accessible for attacking, and expert skills and knowledge are required to successfully compromise connected car platforms.

- Exploiting connected cars as an entry point to the ITS back-end network is, also surprisingly, rated medium-risk. This might be because this attack requires highly skilled attackers who know how to compromise traditional multilayered IT defenses.

Overall, the risk level of successful attacks on connected cars and V2X-connected ITS infrastructures is medium. We think that this is because these hypothesized attacks are assessed using the TTPs used by attackers today. When middleware that obfuscates the internal E/E car architecture is made available to third-party vendors to provide software as a service (SaaS), we expect to see the emergence of new TTPs that will be of a significantly higher risk level. Also, when attackers create viable monetization methods for connected cars and ITS infrastructures, another evolution in their TTPs that will lead to higher-risk attacks.

5.4 Objectives and Profiteering Models for Attacking Connected Cars

The key motivator for the vast majority of cyberattacks that we see daily is financial gain. But not every perpetrator who attacks a car will be motivated by money. Cars are highly visible and attacks on them will be high-impact, which itself is a key motivator for many of the perpetrators. We have identified five broad objectives and profiteering models that motivate perpetrators to attack the connected car ecosystem.

Objectives and profiteering models	Impacts
Ransom	In the future, it is expected that perpetrators will devise methods of sending malware to connected cars and disable the car's functions until the owner pays the demanded ransom. An OTA ransomware attack that happens while the connected car is traveling on the road will severely impact the safety of the vehicle's passenger(s) and other vehicles on the road.
Data theft	Perpetrators can steal proprietary data, intellectual property, business operations data, PII, financial data, sales data, customer information, shipment data, vehicle tracking information, other similar information, and monetize the stolen data in various ways. The stolen data can be used for identity theft, privacy violation, financial fraud, industrial espionage, blackmail, people and/or vehicle reconnaissance, and such.

Objectives and profiteering models	Impacts
Information warfare	This is a broad topic that encompasses everything from hacktivism to data pollution for financial gains. Some of the information attacks that we identified/theorized are: • DDoS of ITS infrastructure to crash systems and cause roadway chaos. • Hacking of OEM apps/websites to post political, protest, or prank messages. Another agenda could be to hurt the OEM's reputation and cause them financial losses. • Transmitting fake V2V messages to create traffic chaos (when cooperative autonomous vehicles become roadway reality). • V2V information poisoning will become a service where unscrupulous businesses will pay criminals to poison V2V channels and forcibly reroute autonomous vehicles to their business locations. • Map-hacking (done via compromising road-based location transmitters, hacking on-board GPS receivers or GPS signal spoofing) will cause autonomous vehicles to veer off course and cause an accident.
System gaming and theft	One of the most attractive profiteering models for perpetrators will be the theft of the vehicles themselves or the goods/valuables inside the vehicles. We identified/theorized the following attacks: • Hacking and rerouting autonomous trucks to some remote location like an empty parking lot outside town, where the criminals can break into the trucks and steal the cargo. • Using hacked autonomous vehicles to anonymously deliver contraband items such as drugs and weapons. • Hacking autonomous passenger vehicles and instructing them to reroute and stop at some obscure location. The criminal's goal is to steal the passengers' valuables or even abduct someone. • Hacking, rerouting, and stopping autonomous vehicles to steal vehicle parts or the vehicles themselves. • Unscrupulous fleet operators (like some autonomous taxi and delivery service providers) attempting to subvert competition by hacking competitors' autonomous vehicles to make them unavailable. • Illegally assigning higher priority to an autonomous vehicle on a dedicated roadway so that other autonomous vehicles move aside. • Making fake orders of autonomous vehicle rideshares to charge unsuspecting customers.
Revenge and terrorism	If the driving functions of autonomous vehicles can be compromised, then there is every possibility that the vehicles will be used as weapons in terror attacks. Terror attacks primarily target people, but we surmise that hijacked vehicles can also be used for attacking critical infrastructures. The terrorists will launch these attacks remotely, which makes tracking the attackers extremely difficult, if not impossible.

Table 4. Objectives and profiteering models and corresponding impacts for attacking connected cars

From a cybersecurity perspective, one glaringly obvious reason for hacking connected cars is to use them as an entry point into the greater ITS ecosystem. If an attacker can successfully exploit the car to gain access into the corporate network, then they can penetrate deep inside the network with minimal effort because the car is considered a trusted node. Once inside the network, the perpetrators can launch any of the attacks that we just described.

5.5 Perpetrators Launching Attacks on Connected Cars

Where there are opportunities, there are also perpetrators who attack, leverage, steal, game, and abuse the system for a wide variety of reasons such as money, revenge, and protest, among others. We have already examined some of the motivators driving perpetrators to attack connected cars, and in this section, we discuss the different perpetrators who pose threats to the connected car ecosystem:

- **Nation states.** Both developed and developing countries gather intelligence using software espionage tools and customized malware. The primary goal for state-sponsored cyberattacks, based on our observation of current attacks, is to steal intellectual property or to gain a competitive advantage. But in certain instances, for example during a war, these attacks can be used to sabotage another nation’s infrastructure. State-sponsored attacks follow one of two modi operandi: either The state directly controls the hacking teams and their resources, or the state outsources the hacking activities to third parties such as criminal gangs to maintain plausible deniability.

- **Criminal hacking groups.** These are composed of highly skilled hacking teams who are funded and controlled by organized criminal gangs. They target victims using different schemes such as ransomware, phishing, and other threats to generate illicit revenue. There are also criminal hacking groups that are contracted by national governments for various political cyberattacks, including cyberespionage and subterfuge.

    1. **Hacktivists.** Hacktivists are internet activists who attack cyber assets to draw attention to their political causes. They tend to choose high-visibility, high-profile targets. Often, their targets and their stated causes do not match.

    2. **Cyberterrorists.** Their goal is to launch cyberattacks to cause destruction of property, loss of life, and the spread of terror.

- **Insiders:** They are people who act against organizations that they are or were part of. Insiders can be motivated by money, ideology, coercion, ego, revenge, and politics; often, more than one of these motives are at play.

- **Unscrupulous operators.** As primary users of vehicles, it is not inconceivable to imagine scenarios where some drivers and commercial operators try to hack and game the system to save on fines and fees, get ahead in traffic, and stifle competition, among others.

6. Guidelines for Protecting Connected Cars

Connected cars are one component of the ITS network, which is a massive, complex, interconnected ecosystem with millions of endpoints and end users. Because of its vastness and the sophistication of its functions, this ecosystem paves the way for large and at times unpredictable attack surfaces. Protecting connected cars from remote attacks is not just about securing the car itself. It is also imperative that the end-to-end data supply chain used by connected cars while on the road be secured.

Cyberattack and data breach prevention strategies should be considered as an integral part of daily business operations for all organizations. Ultimately, no defense is impregnable to determined adversaries — in a nutshell, cyberattacks and data breaches are inevitable. Having effective and active containment and mitigation processes is critical. The key principle of defense is to assume compromise and to take adequate countermeasures:[109]

- Quickly identify and respond to ongoing security breaches.
- Contain the security breach and stop the loss of sensitive data.
- Prevent attacks by securing all exploitable avenues.
- Apply lessons learned to further strengthen defenses and prevent repeat incidents.

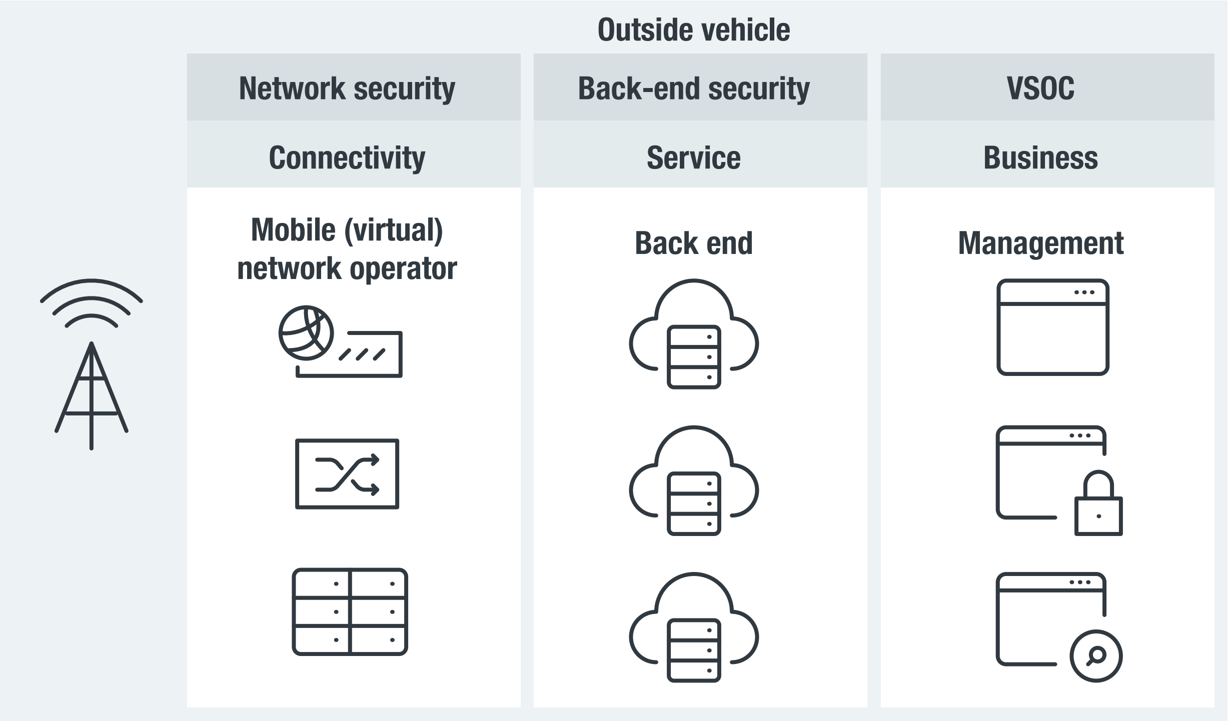


Figure 22. The connected car architecture outside a vehicle

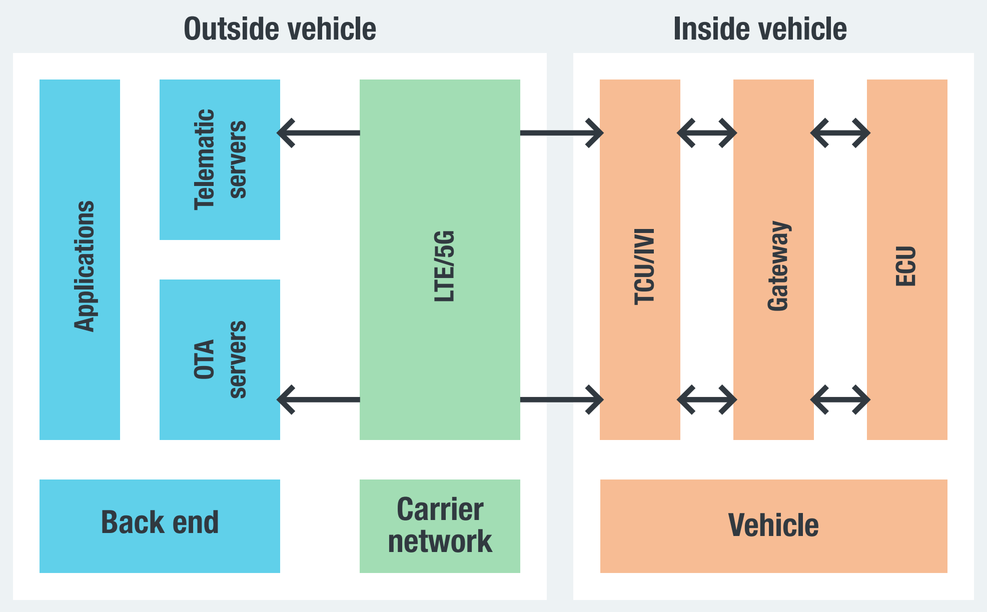


Figure 23. The combined connected car architecture from inside and outside a vehicle

We have identified four critical areas that are needed in the end-to-end data supply chain for connected cars:

- **The E/E network of the car.** This includes the telematics communication box (TCU), the main board ECU or the in-vehicle infotainment (IVI) system in the head unit, the bus gateway, the buses (such as CAN, Ethernet, FlexRay, LIN, and MOST), and the different ECUs. The E/E network also includes external network modules such as the eSIM (which connects to 3G, LTE, 4G, and 5G networks), USB, Bluetooth, Wi-Fi, RDS, and satellite.
- **The network infrastructures.** These handle the millions of connection setup/teardowns and data transmission which includes telecom infrastructure, home network, and mobiles providing network connectivity to the cars.
- **The back-end servers.** These run the applications, services, and databases that a connected car directly and indirectly accesses. Examples include ITS servers, app stores, OTA servers, and more.
- **The VSOC.** This understands context by correlating notifications from the other three critical areas.

Connected car security needs to be designed with an integrated view of these four critical areas to secure the end-to-end data supply chain. Now, a point of contention when considering securing the end-to-end data supply chain for connected cars is: which party will be responsible for data security? Will it be the drivers of the vehicles? Will it be the manufacturers of the vehicles? Will it be the Tier 1 or 2 supplier? Will it be the fleet management operator? (Sub-question, what will be the ownership structure for connected cars?) Will it be a Security-as-a-Service model sold by a third-party security vendor? Will it be a large corporation like Google or Amazon that is entering the connected car space? Will it be the Telco that is managing the data pipelines and builds cybersecurity into the cost of the data? Will it be a government department that implements Security-as-a-Service, similar to national defense, but for connected cars and ITS infrastructure? As is with IT security, the answer is all of the above. The end-to-end data supply chain has many owners who all need to participate in providing holistic security solutions that work to protect the connected car ecosystem.

Our insights and recommendations on ISO/SAE 21434 — cybersecurity engineering guidelines for all processes across different phases of a vehicle's life cycle — are available for reference in our research paper titled "ISO/SAE 21434: Setting the Standard for Connected Cars' Cybersecurity."[110]

6.1 Automotive Defenses Technology Discussion

Currently, the connected car defenses solution space is slightly unknown since the problems and dangers associated with it remain quite unfamiliar. Many companies are working on defense solutions that they think will prevent many of the future threats that connected cars will face. The reality is that many of the automotive defense solutions in the market have not thoroughly been tested to their limits simply because there are not enough connected cars on the road with them, defensive solutions adoption has been slow, and because cybercriminals have not truly found a business case to attack fleets of connected cars. We predict that these will change over the next couple of years, and we best be prepared now to prevent any cyber onslaught against cars.

While thinking about automotive defensive technologies, we decided to start with the different types of attacks (both demonstrated and theoretical ones) that we discussed in this paper. Here is a quick recap of some of the attack vectors discussed (in no specific order):

- Malicious CAN messages transmitted
- SMS used to trigger the telematics unit
- Exploitation of operating systems' vulnerabilities to compromise systems
- Abuse of default system configurations
- Abuse of exposed IT and ITS infrastructure
- Extract firmware from hardware modules
- Upload of modified/malicious firmware and reflash ECUs
- Exploitation of mobile apps and their back-end servers
- Exploitation of third-party apps installed in the head unit
- Use of middleware APIs to create architecture-agnostic malware
- Use of cloud APIs to compromise car (both third-party and OEM clouds)
- MitM wireless data transmissions
- Electronically jamming radio transmissions

So how do we protect connected cars? In this section, we compiled a list of technologies that we think will be important in protecting connected cars and for mitigating and lessening the likelihood of cyberattacks. A solutions technology discussion for securing the connected car ecosystem will yield different requirements for different stakeholders. This is because the needs of the car owner are different from the needs of the fleet management operator or the mobile (virtual) network operator that handles data transmission. Thus, instead of making a case for each stakeholder, we discuss some of the key defensive technologies that will need to be implemented to ensure connected car security:[111]

- **Vulnerability scanner.** This comprises automated tools that scan endpoints, servers, networks, and applications for security vulnerabilities that an attacker can exploit. This can be used to make sure the head unit's operating system doesn't have any unpatched exploitable vulnerabilities. If there is a zero-day that the vulnerability scanner misses, then that will need to be mitigated with an OTA software patch.

- **Code signing.** This is a method of putting a digital signature on a file, software update, or executable so its authenticity and integrity can be verified by an end party.[112] Code signing is a good way of verifying OTA software updates and firmware integrity.

- **Application security.** Security measures at the application level that prevent data or code within the app from getting stolen or hijacked.[113] Application security suites secure against code vulnerabilities, data exfiltration on the server, and other common vulnerability attacks at the application level.[114] Car apps that can be used to lock/unlock, start/stop the car should have strict security enabled to prevent exploit.

- **Threat intelligence.** This is a collection of a wide variety of security data (for example, open-source intelligence, social media intelligence, deep/dark web intelligence, technical intelligence, user endpoint feedback, indicators of compromise, vulnerability data, malware data, and others) that is combined with the expert analysis of a security researcher with the goal of detecting more hidden threats and achieving fast response times against cyberattacks. Threat intelligence drives many of the security solutions used in the car.

- **IDS for CAN.** These are network security systems that examine traffic flow to detect and prevent network attacks. IDS are passive systems that generate a report when a known bad event is identified. IDS for CAN monitors the car's E/E network for suspicious CAN messages by doing deep packet inspection.

- **Antivirus scanner for the head unit.** The head unit run *nix or Windows Embedded. They run applications (both OEM and third-party) and can communicate with the gateway ECU which talks to the car's ECUs. In this critical environment it makes good sense to have an active antivirus scanner running as a service. The main challenge will be to find a reliable method for the head unit antivirus scanner to receive regular OTA updates.

- **Device control.** Systems that regulate access to external storage devices and network resources connected to a computer.[115] Device control is normally used to prevent data loss and leakage, but in cars it can prevent loading and running malicious executables from external storage devices.

- **Firewall.** Firewalls are network security systems that control incoming and outgoing traffic based on an applied rule set. Firewalls monitor ingress/egress traffic from unknown and bad domains and identifies applications or endpoints that generate or request bad traffic. This will be mandatory for V2X traffic policing on the ITS side of the house; it could also be installed on the car side to stop DDoS attacks on the vehicle.

- **Encryption.** In some of the case studies the researchers sniffed and collected CAN messages and then reverse engineered out what the commands were doing by analyzing the collected data. If there is a mechanism to encrypt packets in the E/E network with minimal processing overhead, the sniffing packet data problem would get mitigated.

- **Third-party app review.** Review and test third party apps to verify they are reliable, safe, not malicious, doesn't access ECUs, doesn't break privacy rules, no broken links, doesn't crash or have bug, among others. before allowing them to be downloaded and installed by the user in the car. By strictly controlling the app ecosystem, like Apple does with their iOS devices, the OEMs can prevent third-party apps from introducing new attack surfaces in the car.
- **Read/write protect storage devices.** These are storage devices that download OTA firmware updates and other critical data should be encrypted and have read/write protection enabled to prevent data dumping.
- **Blockchain.** As blockchain solutions for verification become mainstream, it Is possible to have a ledger in the E/E network that records and verifies the status of every ECU in the car. If an ECU is tampered with, then the faulting ECU can be quickly identified, the driver warned, and corrective actions taken.
- **EDR (Extended detection and response).** This collects and correlates deep activity data across multiple points in data supply chain — vehicle, network, and backend servers — enabling a level of detection and investigation. With a view of a combined context, events that seem benign on their own potentially become meaningful indicators of compromise. This quickly contains the impact and minimizes the severity and scope.

6.2 Solution: Cyber-Telecom Federation and Zero-Trust

The use of a cyber-telecom federation allows existing IT security investment to be reused, even when the devices (in this case, cars) being managed are actually from the telecom domain. By managing this harmonized and federated set of cyber-telecom logs using zero-trust architecture, the fractured security landscape can reunited. Attacks that are undetected due to disharmonized logging and multiple sources of radio data can now be seen and alerted as malicious activity.

To manage a fleet of phones-with-wheels at scale, the systems must have visibility into telecom phone data (phone numbers and SIM cards, for example) and cyber IT data (IP addresses and certificates). This cyber-telecom data must also be tied together in a specific way (federation) with the same care and attention that the bolts and other parts of a car are assembled. This cyber-telecom federation allows existing investment in IT security to apply rules to telecom security events that would otherwise be invisible to IT security.

Cyber-telecom federation security should also have privileges and conditions that are realistic for the needs of connected vehicles. As a type of life-critical IoT device, security rules should have recognition for the impact of over-aggressive security responses. Security architects of these privileges should recognize that their rules could result in greater impact on the safety of vehicle passengers. Such impact could include leaving passengers abandoned in a dead car on the side of the road during winter simply because malware was detected and aggressive rules were enforced.

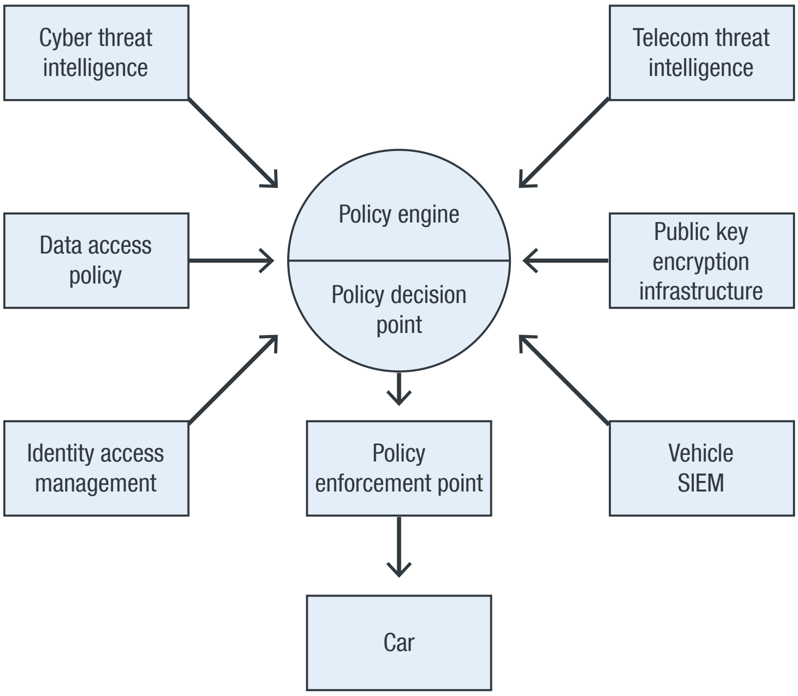
When this data is assembled into a cyber-telecom federation, traditional IT assets can perform security detection on even the telecom (phone numbers and SIM cards) portions of the connected car security model. This is important since telecom threat actors can control the IT security of mobile devices (which is dependent on telecom security) from within the telecom security domain through the use of roaming and home network attacks. The federation can also be used to federate different kinds of IT and telecom telemetry, and even non-cellular types of V2X traffic.

Cyber-telecom federation grants visibility into mobile device issues such as the ability to respond to changes in jurisdiction. Different countries could have security laws on wiretapping, money laundering, and data sovereignty, resulting in business requirements that change as a vehicle drives across country borders. This requires security privileges that will change based on telecom metadata such as location, implying the need for continuous authentication — a technology that is part of the zero trust architecture (ZTA)[116]. These technologies are also part of the build path of 6G.

ZTA is present when all information assets are authenticated and there are no trust zones where authentication does not occur. A version of ZTA is already being used in the telecom domain in systems that touch SIM cards when roaming.

The relationship between cyber-telecom telemetry and ZTA[117] can be used to create unified visibility for connected car fleet management across its various networks and telemetry layers. ZTA guidance has recently been released by the National Institute of Standards and Technology (NIST).[118]

Telecom telemetry and traditional IT telemetry can be brought together harmoniously through the use of a federation. When this cyber-telecom federation is used with ZTA, the method of access control and authorization can be expanded to include telecom infrastructure security events as well.


Cyber threat intelligence
Telecom threat intelligence
Policy engine
Policy decision point
Data access policy
Public key encryption infrastructure
Identity access management
Policy enforcement point
Vehicle SIEM
Car

"Cyberthreat intelligence" refers to typical IT security metadata (such as IP addresses, among others).

"Telecom threat intelligence" refers to typical telecom security metadata, phone numbers, and others.

"Data access policy" refers to information controls and what privileges are needed for access.

"Policy engine and policy decision point" refers to systems responsible for deciding which policies apply to a particular IAM context.

"Vehicle SIEM" refers to security information event management system handling realistic vehicle

"Identity and access management (IAM)" refers to privileges assigned to access data.

"Car" refers to any vehicle managed as a member of a fleet, including cars, drones, or others.

Figure 24. Cyber-Telecom federation

When both ZTA and cyber-telecom federation are used, both telecom crimes and cybercrimes can be made visible to each other's systems, allowing IT security products to secure telecom and telecom security products to secure IT.

A CyTel-federated ZTA can also apply telecom alerting (such as location or jurisdiction) to data sovereignty compliance or other issues such as telecom fraud, SIM-jacking, and other attacks. This can then provide very customized responses that recognize the special considerations needed when managing the security of life-critical mobile devices such as cars.

In this way, a vehicle security operations center (VSOC) can leverage SIEM, SIM card authentication, traditional cyberthreat intelligence, GSMA information, and other sources interchangeably. An enterprise can therefore re-use its investment security investment across multiple infrastructures without having to reinvent the wheel for each.

6.3 Recommended Approach for Connected Car Security

To limit the possibility of a successful remote hack on a connected car, we prescribe a comprehensive cybersecurity strategy that takes the entire connected car ecosystem into account: vehicle, network, back end, and VSOC.

For the Vehicle

The Trend MicroTM IoT Security for Automotive solution[119] monitors and protects critical devices connecting the in-vehicle and outside networks, such as telematic control units (TCUs) and IVI systems, while CAN Bus Anomaly Detection monitors traffic in the CAN bus and reports the status to the VSOC. By linking to Trend Micro's threat intelligence system, they can quickly detect security risks and protect connected cars — along with their systems, applications, and CAN bus — from ever-growing threats.

For the Network

Trend Micro Mobile Network Security (MNS)[120] can be used to keep the network that connects the vehicle, the back-end cloud, and the data center secure. It comprises two key components: First, building on the ETSI NFV framework, it applies appropriate network security protocols to monitor, detect, and take countermeasures against threats. Second, it provides comprehensive endpoint security for IoT devices in two form factors — a physical SIM card and a software Java applet.

For the Back End

Among the various connected technologies used by connected cars are applications and systems hosted on the cloud. And many more of these applications and systems are bound to be built as the adoption of connected cars continues to grow. The Trend Micro™ Cloud One™ security services platform[121] can be used to secure back-end cloud and data center environments without affecting performance. Through the Trend Micro™ Zero Day Initiative™ program, it can detect and disclose vulnerabilities to keep cloud environments secure, especially since it is common for new and evolving technologies to have known and unknown vulnerabilities. The platform also continuously analyzes and identifies new malware, ransomware, and IOCs that could be used in attacks. In addition to ITS back-end systems, ITS endpoints need to be secured. The Trend Micro IoT Security™ solution[122] can be used for this purpose. It uses threat intelligence from the Trend Micro™ Smart Protection Network™ infrastructure to provide risk/anomaly detection and in-system protection for a wide range of IoT devices, including traffic lights and surveillance cameras.

For the VSOC

To ensure that the VSOC is able to correlate events quickly and effectively, the Trend Micro™ XDR® service can be used. It passes analyzed, correlated, and visualized events from the endpoint, the network, and the back end, with individual notifications for each.[123] It provides a comprehensive look at events alongside vital contextual data, thereby helping organizations identify and thwart threats.*

* As of February 2021, Trend Micro XDR is limited to certain Trend Micro products.

7. Connected Car Security in Motion

More than 125 million passenger cars with embedded connectivity are forecasted to ship worldwide between 2018 and 2022,[124] and the annual production of semi- and fully autonomous vehicles are expected to reach more than 14 million by 2025.[125] The sheer volume of network-connected cars will both create and increase new attack surfaces for the ITS ecosystem. In this paper, our goal is to explore potential new cyberattacks against connected cars. By analyzing the technology used in cars today and in the connected car future roadmap, we derived a host of interesting attacks that are not being actively discussed.

Connected cars will be connected via either 5G or dedicated short-range communication (DSRC) to V2X and will probably become one of the heaviest users of cloud infrastructure. Future cars will have part of their E/E architecture transferred to the cloud and communicating back to the vehicle via high-speed, low-latency networks. The connected cars with their all-digital cockpits will support a rich ecosystem of third-party apps that will provide a host of functionalities to the drivers and passengers. This opens up many interesting attack scenarios on connected cars via cloud infrastructure attacks or through architecture-agnostic middleware and/or cloud APIs. These threats do not stop at individual cars, but can also be extended to attack fleets of vehicles potentially hijacking fleet control and causing havoc on the roadways.

The remote attacks on connected cars that we see today use complex chains of vulnerabilities to compromise the vehicles. As the connected car technology stack evolves, we fully expect IT attacks and the everyday malware monetization machine to quickly adapt to a connected car's ecosystem to generate new revenue streams for cybercriminal enterprises.

Finally, by plotting and measuring the risks in connected cars using the DREAD threat model, we determined that the simplest attacks, such as electronic jamming, exploitation of device misconfigurations, DDoS, and others are far more damaging than seemingly sensational attacks such as installing malicious firmware OTA, and remote hijacking of the CAN Bus, among others. In a nutshell, we conclude that cloud and middleware or cloud-APIs will not only be a connected car's greatest assets but also its weakest links.

Given the high level of expectations from connected cars and their potential vulnerabilities, it is important to ensure that automotive cybersecurity technologies stay ahead of adversary TTPs; this is the Red Queen hypothesis[126] where security researchers are in a never-ending arms race with cybercriminals. There is no doubt that there will be a duplication of defensive technologies across the many owners of the end-to-end data supply chain for connected cars. This is a good thing for two reasons: First, data owners will purchase security products from multiple vendors, and multiple vendors of similar products will increase the likelihood of catching malicious activity. Second, multilayered defenses will make it increasingly difficult for attackers to succeed. Ultimately though, almost no defense is impregnable against determined adversaries, but the multilayered approach increases the time, cost, and resources needed by an adversary to mount a successful attack. By making the ROI low compared with the cost of hacking, connected car OEMs and operators can successfully deter most attackers.

This research paper was meant to be a thought paper that aimed to realistically predict what threats against connected cars and their ecosystem will look like. Many of the attacks that we have described have not occurred yet or have only been demonstrated by security researchers as proofs of concept. Our predictions and insights into future threats against connected cars have a two-fold goal: First, we wanted to inform the OEMs, T1 and T2 suppliers, and the general public about the threats and challenges they are going to face on the roadways in the next couple of years; and second, by identifying and addressing the cybersecurity risks faced by connected cars in the early developments stages, we have the opportunity to influence both legislative and technological developments in this domain. Developers and manufacturers must include security as a basic design element in the various technologies that go into connected cars. This will give them better visibility into the connected car ecosystem and enable them to quickly identify, isolate, and mitigate any incoming threats and keep cars running safely on the roadways.

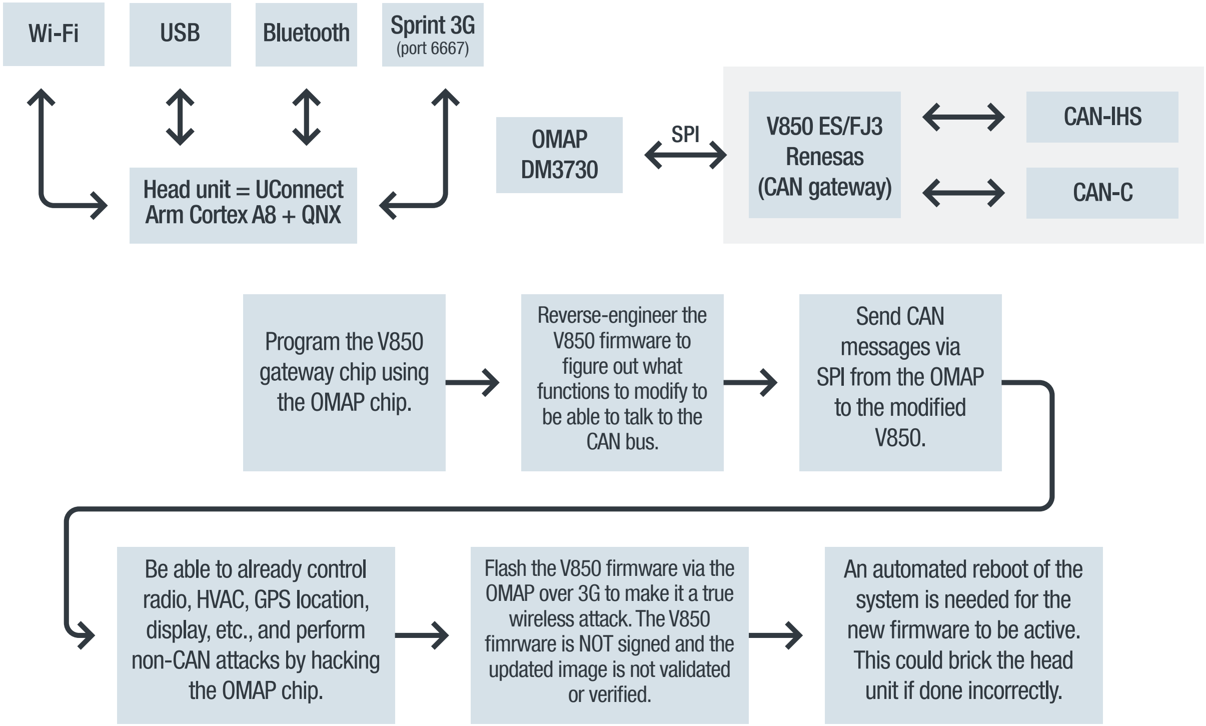
Appendix

Appendix A: Case Studies of Remote Attacks

In this section, we look at four case studies on remote attacks on connected cars where at least one ECU in the target car was successfully compromised, allowing attackers to tamper with vehicle functions. Our goal is to explore the TTPs used by the attackers to remotely compromise their target vehicle ECUs. These TTPs indicate the limitations of what can or cannot be hacked, and the level of difficulty in hacking today's connected cars. These findings, combined with our expertise in cybersecurity, helped us create a threat model for connected cars that original equipment manufacturers (OEMs), their Tier 1 (direct) and Tier 2 (indirect) suppliers, government agencies, and everyday drivers will need to contend with.

The Jeep Hack 2015

"Remote Exploitation of an Unaltered Passenger Vehicle,"[127] a 2015 paper by Charlie Miller and Chris Valasek on car hacking involving Chrysler's Jeep, was a seminal car hacking research that ultimately led to the recall of 1.4 million Chrysler vehicles.[128] The researchers found the Class A address space used by the US telecommunications company Sprint for connected vehicles and discovered that a D-Bus message daemon was running on the exposed port 6667 in the car, which was open to receiving unauthenticated commands via Telnet. The researchers sent commands using remote procedure call (RPC) methods supported by the D-Bus daemon and successfully rooted the head unit of the target Jeep vehicle.



Wi-Fi
USB
Bluetooth
Sprint 3G (port 6667)
Head unit = UConnect Arm Cortex A8 + QNX
OMAP DM3730
SPI
V850 ES/FJ3 Renesas (CAN gateway)
CAN-IHS
CAN-C
Program the V850 gateway chip using the OMAP chip.
Reverse-engineer the V850 firmware to figure out what functions to modify to be able to talk to the CAN bus.
Send CAN messages via SPI from the OMAP to the modified V850.
Be able to already control radio, HVAC, GPS location, display, etc., and perform non-CAN attacks by hacking the OMAP chip.
Flash the V850 firmware via the OMAP over 3G to make it a true wireless attack. The V850 fimrware is NOT signed and the updated image is not validated or verified.
An automated reboot of the system is needed for the new firmware to be active. This could brick the head unit if done incorrectly.

Figure 25. The attack chain of the Jeep Cherokee remote hack of 2015

The following are some of the interesting findings from their paper:

- The researchers selected the Jeep Cherokee 2014 because it offered the best opportunity for a successful hack.[129]
- The head unit can speak with both CAN-IHS (CAN Interior High Speed) and CAN-C (CAN Critical) networks. Critical systems are not on separate buses physically within the Jeep network architecture; the Jeep network architecture is essentially a flat network with no domain segregation.
- The vehicle's head unit, Jeep® UConnect,[130] was found to be "jailbreakable" using the USB, but that method was ultimately not needed in the successful hacking of the car.
- Access via Wi-Fi, while possible, is not ideal, since an attacker needs to be within a specific distance from the vehicle. The researchers cracked the Wi-Fi password by reverse-engineering the password generation algorithm, but this method was found to be tedious.
- Access via cellular (3G) network is best, since an attacker can be outside of visual range and will still be able to control the vehicle. It was found that access via a femtocell,[131] a small and low-range cellular base, is limited to 30 meters, and that it was better to use Sprint's nationwide cellular network.
- The D-Bus message daemon running on the open port 6667 can receive unauthenticated commands via Telnet. The researchers sent commands via command-line injection via RPC methods supported by the D-Bus.
- The researchers found that Sprint's network allows any Sprint device to talk to another Sprint device over any distance as long as both are connected to Sprint's network. No device blocking was found to be active; it was as though the devices were on a national-scale WAN (wide area network). Theoretically, the researchers could create a network worm that could traverse and infect all Sprint-connected cars via the D-Bus daemon running on the exposed port 6667.
- The researchers downloaded the firmware for the Renesas V850 microprocessor and the OMAP (Open Multimedia Applications Platform) processor from Chrysler. They reverse-engineered and modified the V850 firmware. The OMAP is able to update the V850 firmware, and that was how they uploaded the modified firmware.
- The researchers rewrote parts of the SPI in the V850 firmware and inserted their shellcode. This would interpret the SPI messages as CAN messages and broadcast them to all CAN bus-connected ECUs.
- The researchers reverse-engineered the wiTECH mechanics toolkit,[132] a technology that allows technicians to diagnose and fix vehicles remotely, to find out how to unlock ECUs and sniff vendor-specific CAN messages.
- The researchers reverse-engineered the algorithm to checksum CAN messages so that they looked legitimate to the vehicle ECUs. A checksum is used to ensure the authenticity of and check for errors in a message.[133]

- The researchers reverse-engineered the algorithm to unlock an ECU for reprogramming. It turned out that their target ECU — the parking assist module, which reads CAN messages for manipulating the steering function — was also a V850 chip. Because the researchers were familiar with this architecture, the algorithm reverse engineering was relatively easier.
- The researchers figured out the CAN messages that kill the engine, disable the brakes, and turn the steering wheel. They then figured out how to rewrite CAN messages from the real ECUs or deactivate these ECUs so that their malicious CAN messages got executed instead.

The Tesla Hack 2016

A Tesla vehicle is a computer on wheels. It packs technologies for powertrain, battery, user interface, and connectivity that are years ahead of the competition.[134] But even a sophisticated, well-designed computer network has its shortcomings — and that is what a team of researchers at Tencent Keen Security Lab proved with their successful exploitation of a Tesla Model S in 2016.[135] They used a complex chain of vulnerabilities to compromise the components inside the car and ultimately succeeded in injecting malicious CAN messages into the CAN bus.

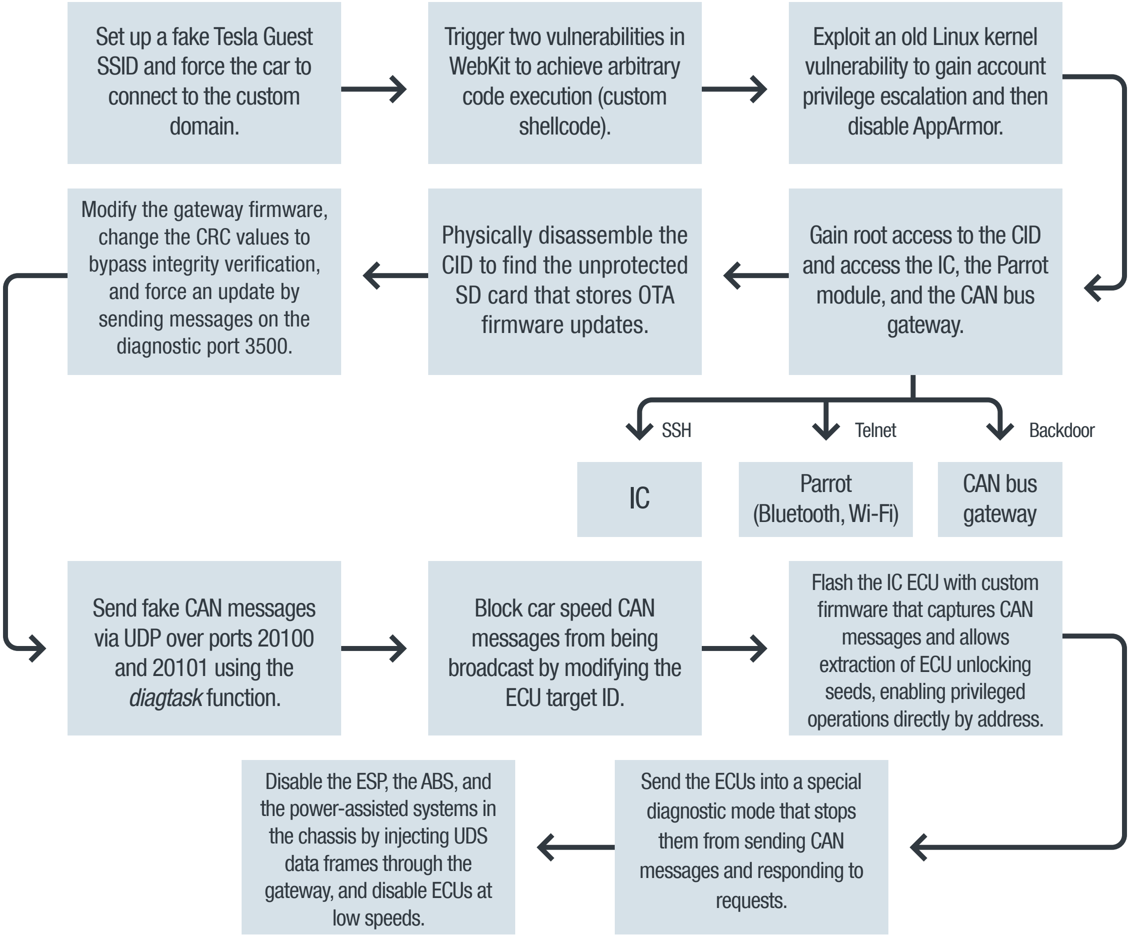

Set up a fake Tesla Guest SSID and force the car to connect to the custom domain.
Trigger two vulnerabilities in WebKit to achieve arbitrary code execution (custom shellcode).
Exploit an old Linux kernel vulnerability to gain account privilege escalation and then disable AppArmor.
Gain root access to the CID and access the IC, the Parrot module, and the CAN bus gateway.
Physically disassemble the CID to find the unprotected SD card that stores OTA firmware updates.
Modify the gateway firmware, change the CRC values to bypass integrity verification, and force an update by sending messages on the diagnostic port 3500.
SSH
Telnet
Backdoor
IC
Parrot (Bluetooth, Wi-Fi)
CAN bus gateway
Send fake CAN messages via UDP over ports 20100 and 20101 using the diagtask function.
Block car speed CAN messages from being broadcast by modifying the ECU target ID.
Flash the IC ECU with custom firmware that captures CAN messages and allows extraction of ECU unlocking seeds, enabling privileged operations directly by address.
Send the ECUs into a special diagnostic mode that stops them from sending CAN messages and responding to requests.
Disable the ESP, the ABS, and the power-assisted systems in the chassis by injecting UDS data frames through the gateway, and disable ECUs at low speeds.

Figure 26. The attack chain of the Tesla Model S remote hack of 2016

The following summarizes the attack chain that the researchers used to compromise the Tesla Model S:[136]

- All Tesla vehicles are configured to automatically connect to SSID Tesla Guest, a Wi-Fi hotspot provided by Tesla body shops and superchargers. The researchers set up a fake Tesla Guest hotspot and forced the car to connect to their custom authentication domain.
- The user agent of the Tesla vehicle uses the web browser engine WebKit.[137] The researchers triggered two vulnerabilities in WebKit to achieve arbitrary code execution by adding a custom shellcode to the script and get a remote shell.
- The researchers exploited an old, unpatched Linux kernel vulnerability, CVE-2013-6282,[138] to gain a higher privilege than the one granted to the browser. The researchers then disabled the kernel security module AppArmor.[139]
- Privilege escalation grants root access to the central information display (CID). Pivoting from the CID, the researchers gained access to the instrument cluster (IC) via SSH (Secure Shell),[140] the Parrot module (Bluetooth and Wi-Fi) via Telnet, and the CAN bus gateway via a custom backdoor.
- The researchers physically disassembled the CID and found a 4 gigabyte (GB) SD card inside it without any read/write protection. This SD card stores the OTA firmware updates sent by Tesla to the Tesla Model S.
- The researchers isolated and modified the gateway firmware in the OTA firmware package, changed the cyclic redundancy check (CRC)[141] values to bypass integrity verification, and forced a firmware upgrade by sending messages to the diagnostic port 3500.
- The gateway sends CAN messages over ports 20100 and 20101. The researchers used their modified gateway firmware to send malicious CAN messages via UDP (User Datagram Protocol)[142] on these ports using the diagtask function.
- As a safety precaution, the Tesla Model S ignores certain CAN bus messages when the vehicle speed is above a set limit. The researchers blocked the vehicle speed CAN messages from being broadcast by modifying the ECU target ID.
- The researchers flashed the IC ECU with custom firmware that captured CAN messages and allowed them to extract the ECU unlocking seeds. ECU unlocking enabled them to perform privileged operations, such as read/write memory, directly by address.
- At this point, the researchers were now able to send the ECUs into a special diagnostic mode that stops the ECUs from sending CAN messages and responding to requests.
- The researchers disabled the electronic stability program (ESP), the antilock braking system (ABS),[143] and the power-assisted systems in the chassis by injecting UDS (Unified Diagnostic Services)[144] data frames through the gateway, and disabled ECUs at low speeds.

Tesla Hack 2017

Keen Security Lab did a follow-up investigation in 2017 to see whether the Model S issues had been resolved after Tesla fixed the reported vulnerabilities. Not surprisingly, they managed to once again compromise the S Model. The truth of the matter is that any complex system, no matter how well-engineered, can have design flaws that an enterprising and dedicated hacker can discover and exploit.

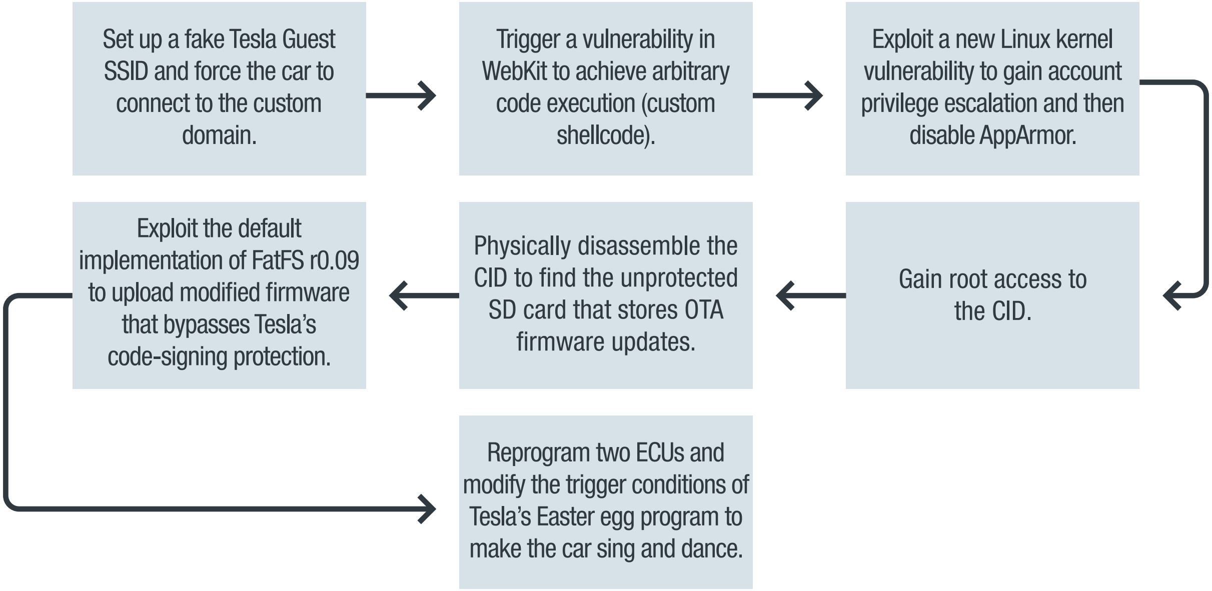

Set up a fake Tesla Guest SSID and force the car to connect to the custom domain.
Trigger a vulnerability in WebKit to achieve arbitrary code execution (custom shellcode).
Exploit a new Linux kernel vulnerability to gain account privilege escalation and then disable AppArmor.
Gain root access to the CID.
Physically disassemble the CID to find the unprotected SD card that stores OTA firmware updates.
Exploit the default implementation of FatFS r0.09 to upload modified firmware that bypasses Tesla's code-signing protection.
Reprogram two ECUs and modify the trigger conditions of Tesla's Easter egg program to make the car sing and dance.

Figure 27. The attack chain of the Tesla Model S remote hack of 2017

The following summarizes the attack chain that Keen Security Lab used to compromise the Tesla Model S and the Tesla Model X:[145]

- The researchers again used the same initial attack vector. They set up a fake Tesla Guest hotspot to force the car to connect to their custom authentication domain.
- The researchers reused another attack vector via WebKit. This time, though, they needed to exploit a single vulnerability in WebKit, instead of two, to achieve arbitrary code execution by using a custom shellcode and get a remote shell.
- Tesla had upgraded the Linux kernel, so using known vulnerabilities would not work. However, the researchers found a new Linux kernel vulnerability to gain account privilege escalation. They were then able to disable AppArmor.
- Privilege escalation also granted the researchers root access to the CID. They physically disassembled the CID again to gain access to the 4 GB SD card. They found that it still did not have any read/write protection.
- Tesla implements code-signing protection to prevent its firmware from getting overwritten. By exploiting the default implementation of FatFS[146] r0.09, the researchers uploaded modified firmware that bypassed or defeated Tesla's code-signing protection.

- Tesla cars have software and hardware Easter eggs that include holiday, video game, and movie themes.[147] The researchers reprogrammed a holiday-themed Easter egg to demonstrate the successful hack. They reverse-engineered and reprogrammed two ECUs and modified the Easter egg trigger conditions to make the Tesla vehicle sing and dance (with the gullwing doors on the Model X). Hacking the Easter egg demonstrated that it is possible to reprogram multiple body control ECUs.
- In addition to reprogramming the Tesla Easter egg, the researchers compromised Tesla's AutoPilot ECU (APE).[148] They wrote and released a separate research paper on it,[149] but it is not covered in our research.

BMW Hack 2018

After hacking Tesla vehicles for two consecutive years, Keen Security Lab shifted their focus to hacking BMW vehicles.[150, 151] They created three attack chains: one for a local attack via the USB/OBD-II[152] port and one each for two remote attacks. Because our research is on connected cars, we focused on the two remote attack chains.

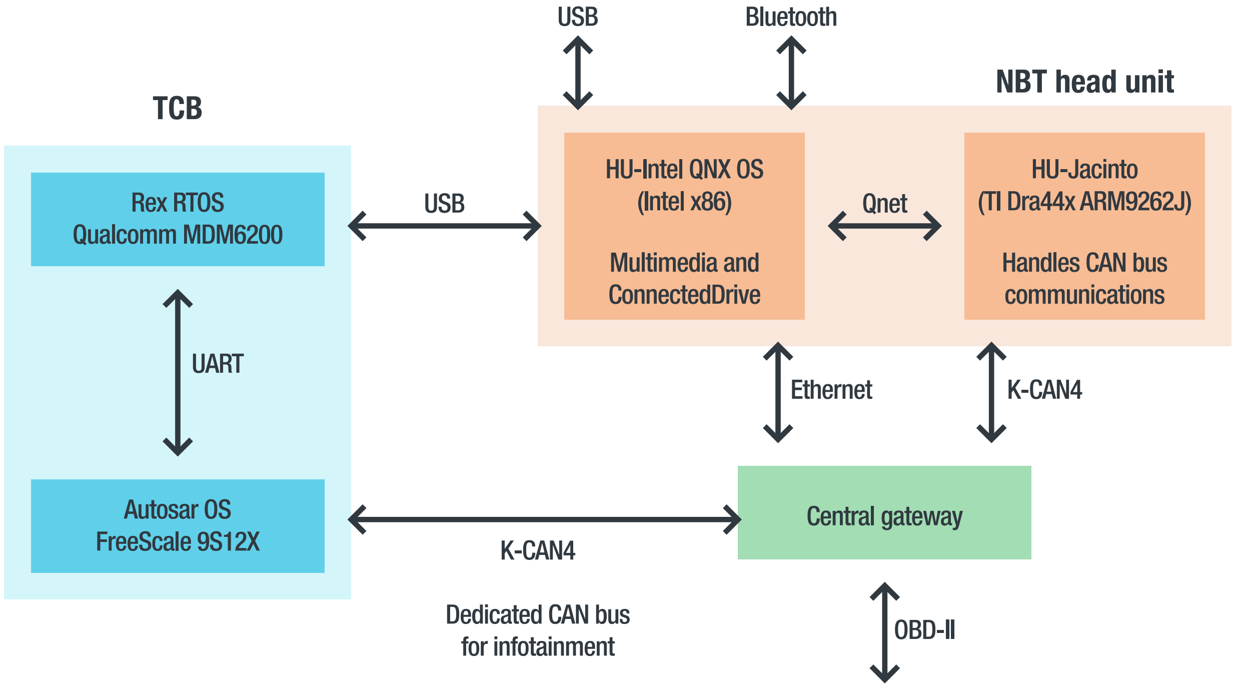


Figure 28. The attack chain of the BMW remote hack of 2018

The first remote attack chain achieved remote code execution (RCE) in BMW ConnectedDrive[153] via HTTP traffic intercept. It worked as follows:

- BMW's ConnectedDrive service in the HU-Intel service periodically polls BMW's back-end servers via 2G or 3G connection of the TCB[154] over HTTP. The researchers set up a fake GSM[155] base station to intercept all GPRS traffic from the vehicle.

- The researchers captured a provisioning file that the vehicle downloads and found the online news URL that ConnectedDrive loads. By forcing the car to connect via their fake GSM base station, they served a custom provisioning file with the online news URL modified with their custom domain.
- The online news functionality was processed by the in-car browser, which ran an older version of WebKit. Exploiting a vulnerability in WebKit resulted in a browser shell. By exploiting a time-of-check-to-time-of-use (TOCTOU)[156] race condition vulnerability, the researchers achieved privilege escalation.
- The researchers gained root access to the firmware via the HU-Jacinto chip, which handles all of the CAN bus communications. They managed to do this by logging in to it from the HU-Intel network through Qnet, a protocol for distributed networking,[157] without any authentication.
- Finally, the researchers dynamically hooked the function CanTransmit_15E2F0 to send arbitrary CAN messages to the ECUs

The second remote attack chain is more interesting and complicated. It exploits the TCB via unsecured SMS as follows:

- BMW's NGTP (Next-Generation Telematics Protocol)[158] allows the back-end server to wake up the car, trigger remote services, and trigger a provisioning update. NGTP messages are encapsulated either in HTTPS or in SMS.
- Using the previously set up fake GSM base station, the researchers sent two SMS messages encapsulating NGTP messages. There was no need to know the TCB's phone number since the car was connected to the hacker-controlled base station. The first SMS message woke up the car's TCB, and the second triggered the provisioning update over HTTP.
- The provisioning file was an XML file that had a signature stored in hex format and was "un-hexified" during signature verification. The researchers crafted a special signature that caused a buffer overflow and allowed them RCE in the TCB's REX Operating System, a real-time operating system (RTOS).[159]
- In the TCB, the Last State Call (LSC) task gathers vehicle status messages via UDS messages stored in a global buffer. Because they could perform RCE, the researchers could overwrite this global buffer with malicious UDS messages. After the LSC task was triggered, they were able to send malicious UDS messages via the TCB to the central gateway.
- The BMW has multiple CAN buses in the network architecture for domain isolation, with the central gateway handling all of the message-switching tasks for the targeted ECUs. The central gateway can forward UDS messages to do remote diagnostics by embedding a UDS message in a CAN message. By changing the target ID of the UDS message, the researchers were able to use the central gateway to send malicious UDS messages to any ECU.

- The researchers were able to reset any ECU of their choosing via the malicious UDS messages while the vehicle was in motion because there are no speed checks for UDS. They could also change the driver's seating position remotely.

Appendix B: V2X Attacks

While most of our discussions have revolved around 5G and C-V2X, there is a competing standard for V2X – IEEE 802.11.p.

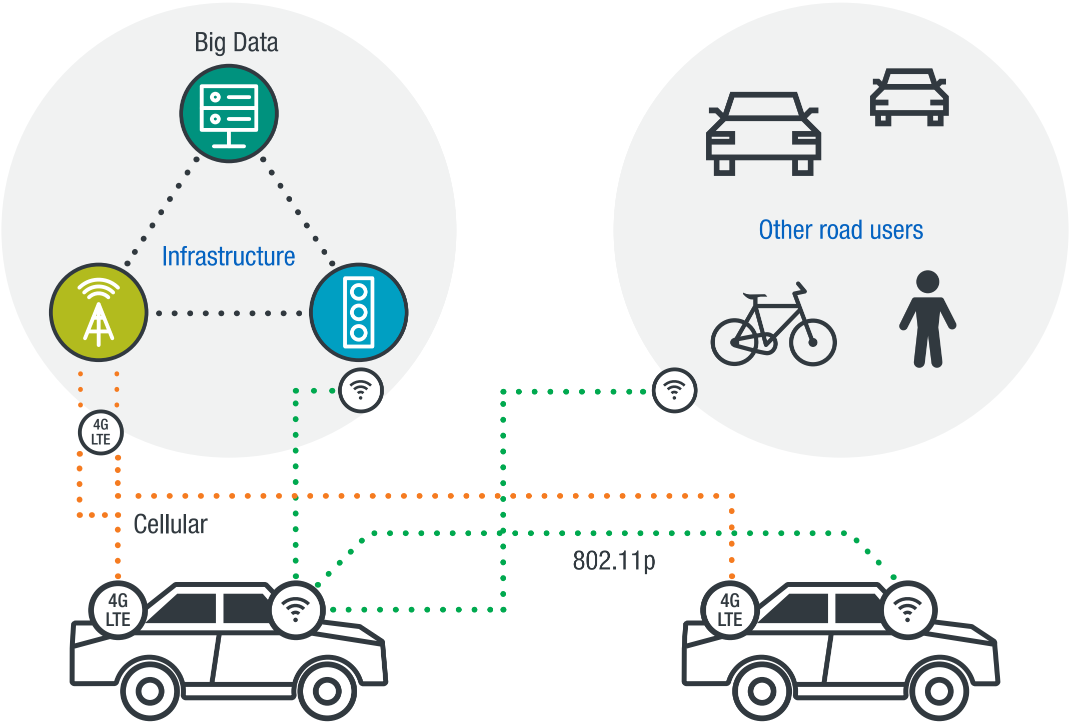


Figure 29. A side-by-side comparison of IEEE 802.11p and cellular connectivity for connected cars[160]

Image credit: Siemens

DSRC based on 802.11p is ready for V2X deployment and addresses most V2X use cases. That means that the cost of deployment is low versus waiting for years to deploy 5G solutions. V2V with cellular would require the car to communicate to the backend to talk to the car in front, whereas DSRC will allow direct communications. We think a hybrid system where the system-on-a-chip supports both C-V2X and DSRC will be the preferred solution. The main drawback for the Wi-Fi system is QoS guarantees and latency requirements might not get fulfilled as they do in 5G based C-V2X solution.

B.1 Use Cases for V2X

Assuming that C-V2X and DSRC (802.11p) are both viable V2X solutions and they are implemented on an SoC and become a standard for connected cars, then the next big question is, what are V2X use cases? Dr. Yunpeng Zang presented five V2X use cases[161] at the ninth European Telecommunications Standards Institute (ETSI) ITS Workshop in March 2018. Michael Gundlach of Nokia also shared insights at the 10th ETSI ITS Workshop.[162] These findings are summarized here:

- **Cooperative maneuvering.** This is the coordination of maneuvers, including intentions and planned trajectories. An example of this would be how vehicles coordinate during the merging of lanes at highway entrances or exits.

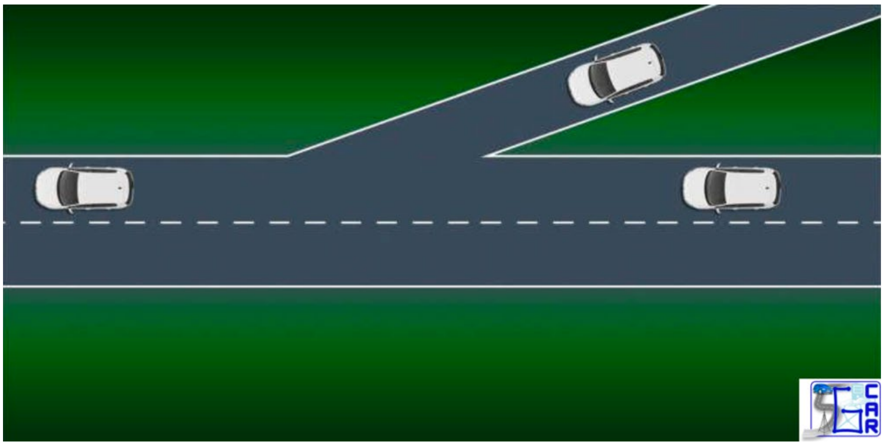


Figure 30. Cooperative maneuvering[163]

Image credit: 5GCAR

- **Cooperative perception:** This is the sharing of data amongst vehicles and/or infrastructure. The data is gathered from various sources like radar, lidar, and on-board cameras, among others, and is shared via wireless connections such as 5G or 802.11p.

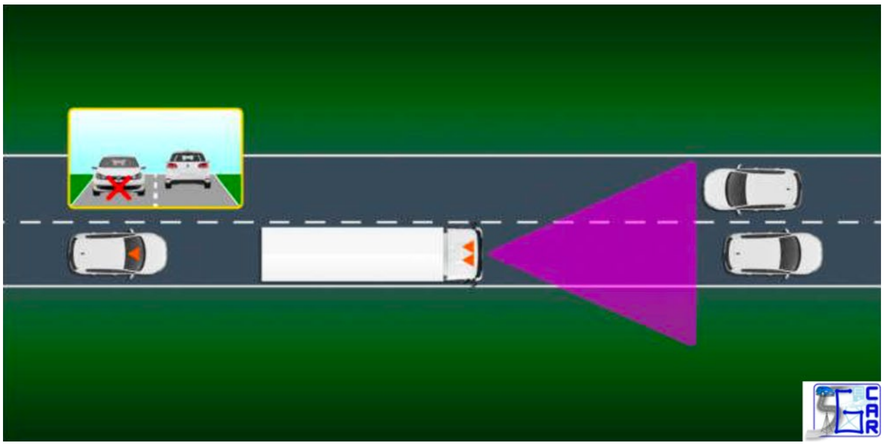


Figure 31. Cooperative perception[164]

Image credit: 5GCAR

- **Cooperative safety.** This pertains to the cooperative sharing of data on the presence of pedestrians on the road. Data is gathered through sensors, cameras, radars, lidar, and other sources. Upon detecting the presence of pedestrians on the street, the drivers can be informed through messages.

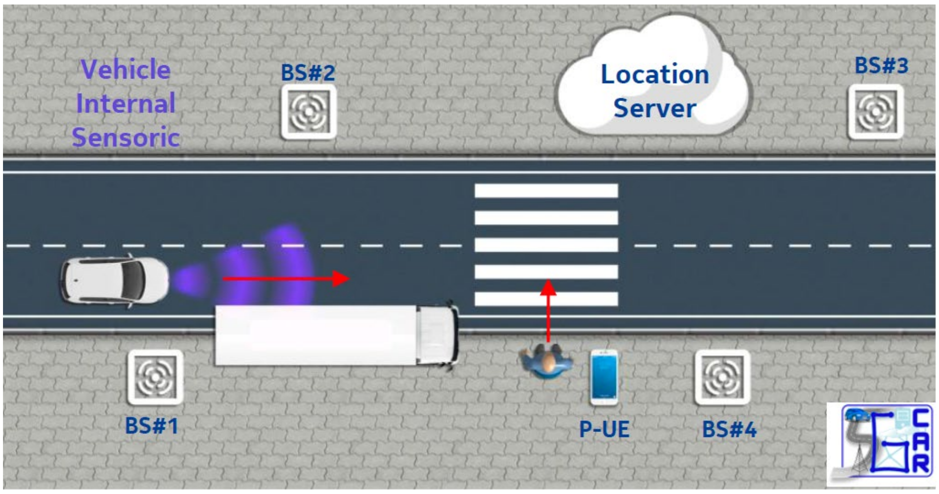


Figure 32. Cooperative safety[165]

Image credit: 5GCAR

- **Autonomous navigation.** This involves building a real-time intelligent HD map through the data collected by vehicles. This data contains specific and detailed information such as road structures and landmarks, among others.

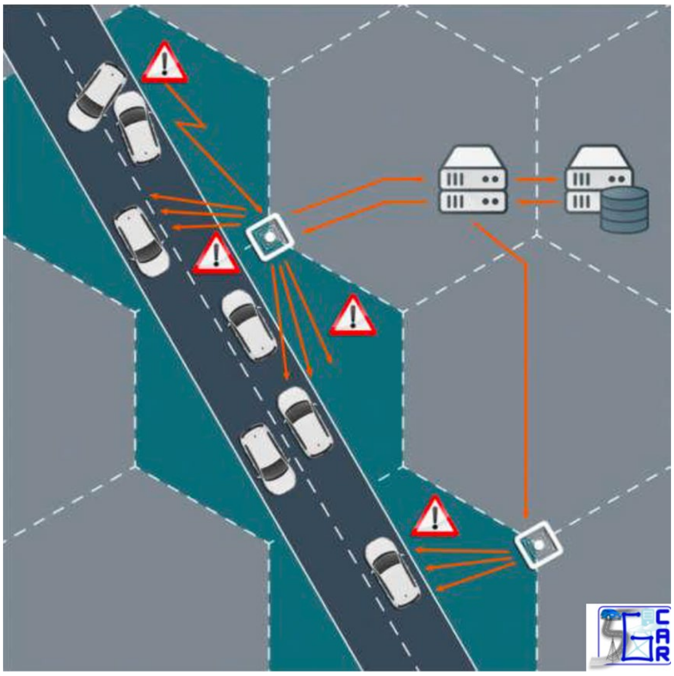


Figure 33. Autonomous navigation[166]

Image credit: 5GCAR

- **Remote driving.** This involves receiving data from vehicle sensors, maps, and infrastructure information. Wireless communication also enables the control of some components of the car, like the steering wheel, from outside the vehicle through wireless communications. Tesla already does something similar with their Smart Summon.[167, 168]

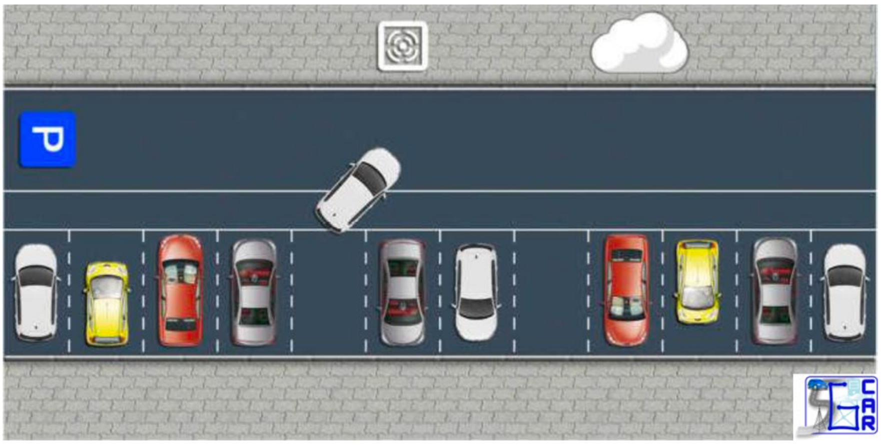


Figure 34. Remote driving[169]

Image credit: 5GCAR

B.2 Attacks on Vehicular Ad Hoc Networks

Connected cars — and in the future, autonomous vehicles — will become the primary roadway users. One of the key technologies connected vehicles will use is vehicular ad hoc networks (VANETs). VANETs are comprised of smart vehicles and roadside units (RSUs), which communicate through unreliable wireless media. Because of their ad hoc nature, VANETs are susceptible to attacks that can jeopardize roadway safety, especially when vehicles depend on VANET data for making critical driving decisions. In this section are the summarized VANET attack vectors described by Fatih Sakiz and Sevil Sen in their paper titled "A Survey of Attacks and Detection Mechanisms on Intelligent Transportation Systems: VANETs and IoV."[170] VANET attacks are primarily network attacks with the connected cars acting as dynamic nodes in the network.

- **Sybil attack.** This is characterized by a node (vehicle) that assumes more than one identity. Due to this, other vehicles in the network cannot verify if the received data originates from one or multiple vehicles. Sybil attacks are difficult to detect.

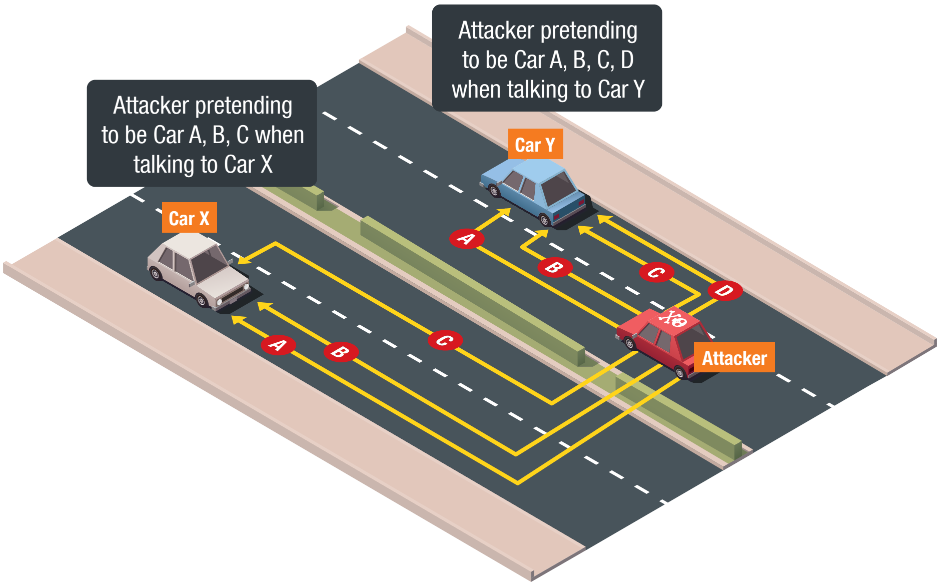

Attacker pretending to be Car A, B, C, D when talking to Car Y
Attacker pretending to be Car A, B, C when talking to Car X
Car Y
Car X
A
B
C
D
A
B
C
Attacker

Figure 35. A Sybil attack

- **DDoS attack.** This overwhelms a system with more requests than it is designed to handle. This causes the system to crash or become unavailable. In VANETs, attackers can interrupt vehicles and RSUs.

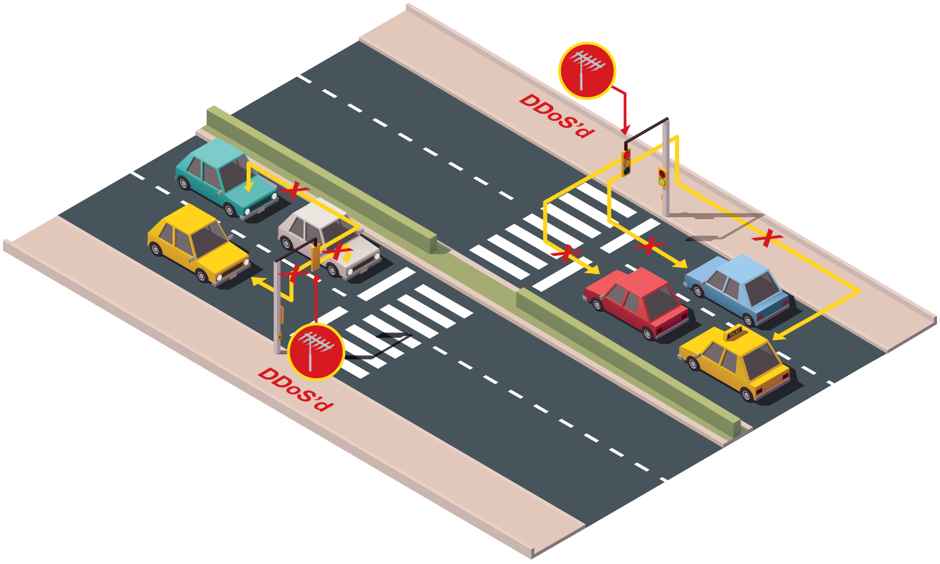


Figure 36. A DDoS attack

- **Blackhole attack.** This attack takes place in an ad hoc co-operative network. The attacker node manipulates other nodes into routing their data packets through the attacker. The attacker then intentionally discards the data packets, resulting in communication loss in the network. Another result is that other vehicles do not receive crucial roadway information.

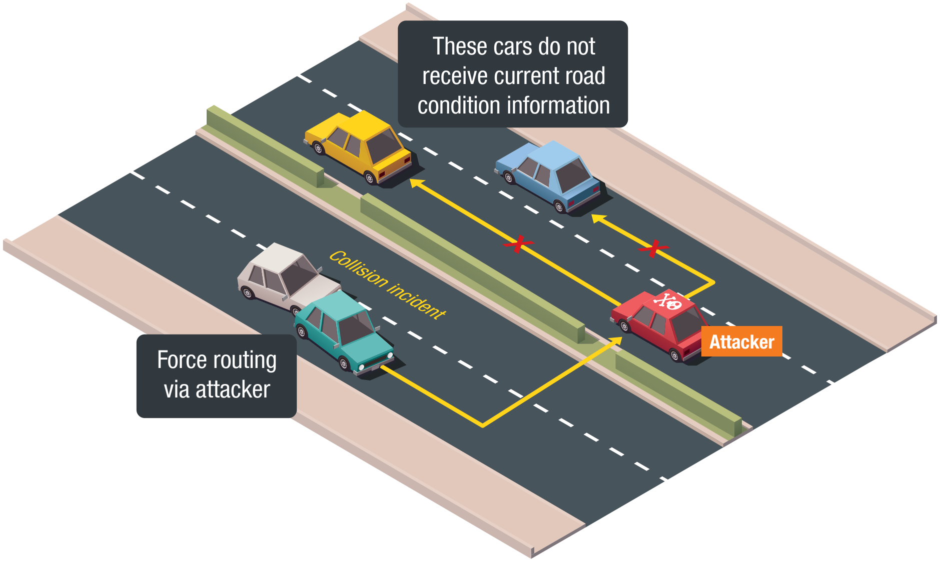

These cars do not receive current road condition information
Collision incident
Attacker
Force routing via attacker

Figure 37. A blackhole attack[171]

Image credit: Fatih Sakiz and Sevil Sen

- **Wormhole attack.** This occurs when two or more compromised nodes involve themselves in many routing requests. They then communicate the false information that they know through the shortest way to a given point. This is designed to alter the topology or arrangement of the network and route all routing requests through themselves to gather and/or control network traffic.

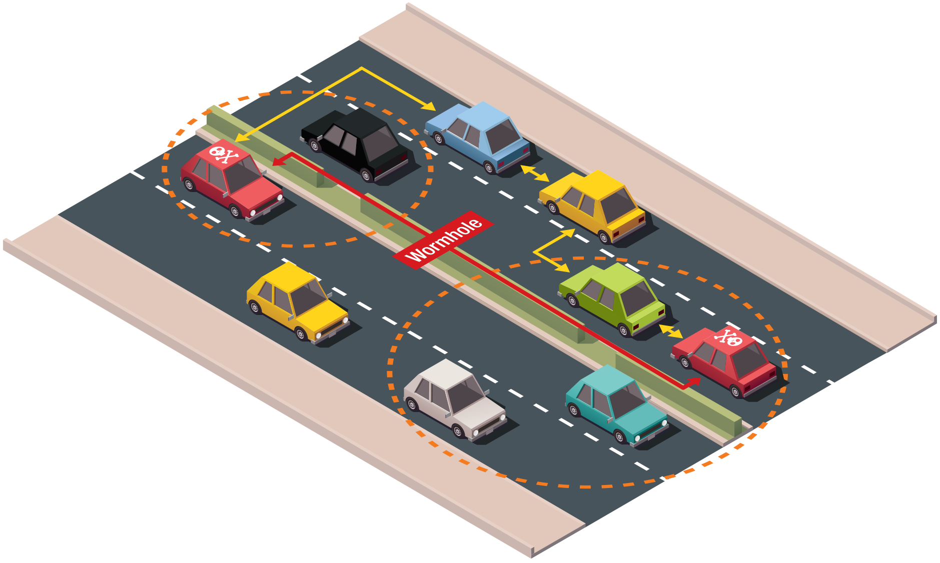


Figure 38. Wormhole attack

Image credit: Fatih Sakiz and Sevil Sen[172]

- **False information attack.** This happens in VANET vehicles that use data generated or forwarded by other vehicles or RSUs. The received or forwarded data might not be true; an attacker vehicle can generate false data and send it to the VANET. Common false information attacks include:
    - **Fake location information,** where vehicles can broadcast fake location data. This is a serious problem because safety-related applications or systems that rely on accurate vehicle location data will respond incorrectly. Also, false location information will result in data packet loss, as packets will be forwarded to phantom vehicles. The vehicles could also broadcast spoofed, fake GPS data with a strong signal that overrides the actual GPS signal, thus causing navigation units to get confused and falter.
    - **Sensor deception,** where by simulating false driving conditions, attackers can deceive in-vehicle sensors. For example, by braking repeatedly over a short distance, the attacker can simulate a traffic jam on the road, causing a car to incorrectly broadcast a traffic jam message.
- **Replay attack.** Here, messages can be stored to be broadcasted later, when the message is no longer valid. This is meant to mislead or deceive other nodes in the network. The attack aims to recreate and exploit the conditions at the time the original message was sent by rebroadcasting the stored message.

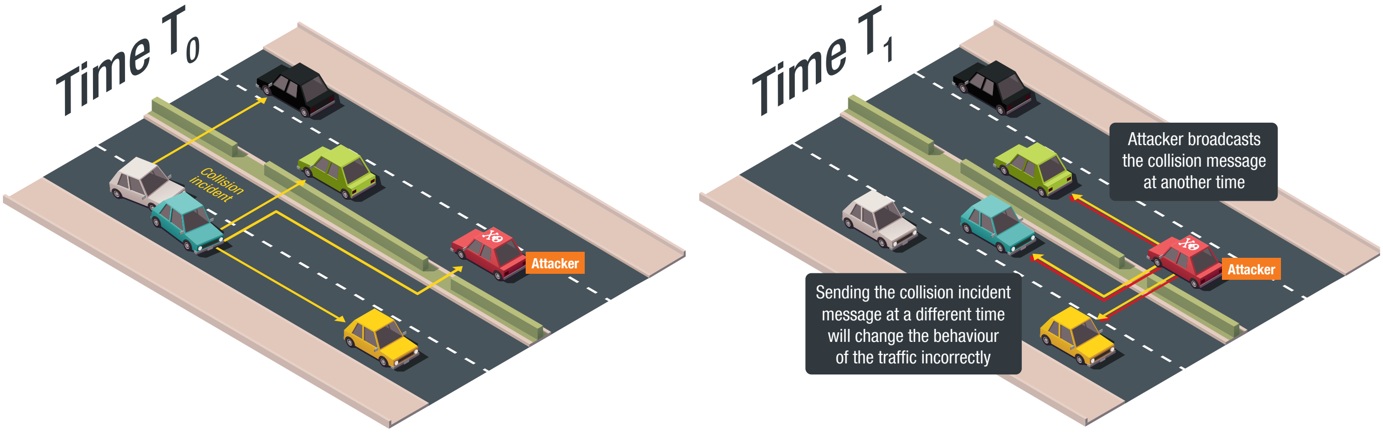

Time T_0
Collision incident
Attacker
Time T_1
Attacker broadcasts the collision message at another time
Attacker
Sending the collision incident message at a different time will change the behaviour of the traffic incorrectly

Figure 39. A replay attack[173]

Image credit: Fatih Sakiz and Sevil Sen

- **Passive eavesdropping attack.** This refers to monitoring the network to track vehicle movement or communications. The attacker node simply intercepts and analyzes the messages that pass through the network. The attacker's goal is to gather information about vehicles and their communication patterns for future use.

References

1 Phil LeBeau. (July 30, 2019). *CNBC*. “Relax, experts say it’s at least a decade before you can buy a self-driving vehicle.” Accessed on June 23, 2020, at https://www.cnbc.com/2019/07/29/experts-say-its-at-least-a-decade-before-you-can-buy-a-self-driving-car.html.

2 Numaan Huq, Craig Gibson, Rainer Vosseler. (n.d.). *Trend Micro*. “Driving Security in Connected Cars.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-driving-security-into-connected-cars.pdf.

3 Kaya Ismail. (Nov. 14, 2018). *CMSWire*. “Connected Car Experiences in 2019: Exploring the Possibilities.” Accessed on July 8, 2020, at https://www.cmswire.com/digital-experience/connected-car-experiences-in-2019-exploring-the-possibilities/.

4 Anshul Axena. (Aug. 17, 2018). *eInfochips*. “Everything You Need to Know About In-Vehicle Infotainment Systems.” Accessed on July 8, 2020, at https://www.einfochips.com/blog/everything-you-need-to-know-about-in-vehicle-infotainment-system/.

5 Sarwant Singh. (Nov. 11, 2019). *Forbes*. “Connected & Autonomous Cars Have Arrived, And They Are Forcing Car Companies To Build New Vehicle Architectures.” Accessed on July 8, 2020, at https://www.forbes.com/sites/sarwantsingh/2019/11/11/connected--autonomous-cars-have-arrived-and-they-are-forcing-car-companies-to-build-new-vehicle-architectures/?sh=415175052cb1.

6 David Zax. (Dec. 3, 2012). *MIT Technology Review*. “Many Cars Have a Hundred Million Lines of Code.” Accessed on June 23, 2020, at https://www.technologyreview.com/2012/12/03/181350/many-cars-have-a-hundred-million-lines-of-code/.

7 Jim Motavalli. (Feb. 4, 2010). *The New York Times*. “The Dozens of Computers That Make Modern Cars Go (and Stop).” Accessed on June 23, 2020, at https://www.nytimes.com/2010/02/05/technology/05electronics.html.

8 Kvaser. (n.d.). *Kvaser*. “Introduction to the LIN bus.” Accessed on June 24, 2020, at https://www.kvaser.com/about-can/can-standards/linbus/.

9 Evaluation Engineering. (June 1, 2014). *Evaluation Engineering*. “Entertainment rides the MOST bus.” Accessed on June 24, 2020, at https://www.evaluationengineering.com/instrumentation/article/13009506/entertainment-rides-the-most-bus.

10 Bob O’Donnell. (June 28, 2016). *USA Today*. “Your average car is a lot more code-driven than you think.” Accessed on June 23, 2020, at https://www.usatoday.com/story/tech/columnist/2016/06/28/your-average-car-lot-more-code-driven-than-youthink/86437052/.

11 Robert Bell. (Oct. 13, 2018). *Space News*. “Satellites and the Connected Car.” Accessed on June 23, 2020, at https://spacenews.com/satellites-and-the-connected-car/.

12 Loren Grush. (March 24, 2020). *The Verge*. “The true impact of SpaceX’s Starlink constellation on astronomy is coming into focus.” Accessed on June 23, 2020, at https://www.theverge.com/2020/3/24/21190273/spacex-starlink-satellite-internetconstellation-astronomy-coating.

13 Tesla. (n.d.). Tesla. “Support: Connectivity.” Accessed on June 23, 2020, at https://www.tesla.com/support/connectivity.

14 B.E. Bilgin and V.C. Gungor. (Nov. 6, 2013). *International Journal of Vehicular Technology*. “Performance Comparison of IEEE 802.11p and IEEE 802.11b for Vehicle-to-Vehicle Communications in Highway, Rural, and Urban Areas.” Accessed on June 23, 2020, at https://www.hindawi.com/journals/ijvt/2013/971684/.

15 Charles McLellan. (Nov. 4, 2019). *ZDNet*. “What is V2X communication? Creating connectivity for the autonomous car era.” Accessed on June 23, 2020, at https://www.zdnet.com/article/what-is-v2x-communication-creating-connectivity-for-the-autonomous-car-era/.

16 National Highway Traffic Safety Administration (NHTSA). (n.d.) *National Highway Traffic Safety Administration (NHTSA)*. “Automated Vehicles for Safety.” Accessed on June 23, 2020, at https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety.

17 Tesla. (n.d.). *Tesla*. “Support: Autopilot and Full Self-Driving Capacity.” Accessed on June 23, 2020, at https://www.tesla.com/support/autopilot.

18 Kyle Field. (Aug. 3, 2020). *CleanTechnica*. “Tesla Achieved The Accuracy Of Lidar With Its Advanced Computer Vision Tech.” Accessed on Jan. 11, 2021, at https://cleantechnica.com/2020/04/24/tesla-achieved-the-accuracy-of-lidar-with-its-advanced-computer-vision-tech/.

19 BBC News. (Sept. 16, 2020). *BBC News*. “Uber’s self-driving operator charged over fatal crash.” Accessed on Jan. 11, 2021, at https://www.bbc.com/news/technology-54175359.

20 Numaan Huq, Rainer Vosseler, and Morton Swimmer. (2017). *Trend Micro*. “Cyberattacks Against Intelligent Transportation Systems: Assessing Future Threats to ITS.” Accessed on June 23, 2020, at https://documents.trendmicro.com/assets/white_papers/wp-cyberattacks-against-intelligent-transportation-systems.pdf.

21 MITRE ATT&CK®. (Sept. 16, 2020). *MITRE ATT&CK®*. “ATT&CK®.” Accessed on Jan. 11, 2021, at https://attack.mitre.org.

22 Trend Micro Security News (n.d.) *Trend Micro Security News*. “Cybercriminal Underground.” Accessed on Jan. 11, 2021, at https://www.trendmicro.com/vinfo/us/security/news/cybercriminal-underground.

23 Dr. Charlie Miller and Chris Valasek. (Aug. 10, 2015). *Illmatics*. “Remote Exploitation of an Unaltered Passenger Vehicle.” Accessed on June 24, 2020, at http://illmatics.com/Remote%20Car%20Hacking.pdf.

24 Tencent Keen Security Lab. (March 3, 2020). *Keen Security Lab Blog*. “Tencent Keen Security Lab: Experimental Security Assessment on Lexus Cars.” Accessed on June 24, 2020, at https://keenlab.tencent.com/en/2020/03/30/Tencent-Keen-Security-Lab-Experimental-Security-Assessment-on-Lexus-Cars/.

25 Kevin Mahaffey. (Aug. 7, 2015). *Lookout Blog*. “Hacking a Tesla Model S: What we found and what we learned.” Accessed on June 24, 2020, at https://blog.lookout.com/hacking-a-tesla.

26 Matthew Greenwood. (June 29, 2020). *Engineering.com*. “Chinese Automaker Plans Satellite Network to Support Autonomous Vehicles.” Accessed on July 20, 2020, at https://www.engineering.com/AdvancedManufacturing/ArticleID/20027/ChineseAutomaker-Plans-Satellite-Network-to-Support-Autonomous-Vehicles.aspx.

27 Scott Campbell. (n.d.). *Circuit Basics*. “Basics of the SPI Communication Protocol.” Accessed on June 24, 2020, at https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/.

28 Scott Campbell. (n.d.). *Circuit Basics*. “Basics of the UART Communication.” Accessed on June 24, 2020, at https://www.circuitbasics.com/basics-uart-communication/.

29 Trend Micro Security News. (July 27, 2017). *Trend Micro Security News*. “InfoSec Guide: Defending Against Man-in-the-Middle Attacks.” Accessed on June 25, 2020, at https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/infosec-guide-defending-against-man-in-the-middle-attacks.

30 Marc-Antoine Boutin. (Jan. 31, 2017). *VanillaPlus*. “Network slicing unleashes 5G opportunities, when service quality can be assured – Part 2.” Accessed on Jan. 11, 2021, at https://www.vanillaplus.com/2017/01/31/24983-network-slicing-unleashes-5g-opportunities-service-quality-can-assured-part-2/.

31 Feike Hacquebord, Robert McArdle, Fernando Mercês, and David Sancho (Sept. 4, 2019). *Trend Micro*. “Ready or Not for PSD2: The Risks of Open Banking.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-PSD2-The-Risks-of-Open-Banking.pdf.

32 David Undercoffler. (March 6, 2013.) *Los Angeles Times*. “Rolls Royce debuts all-new Wraith with GPS-controlled transmission.” Accessed on Jan. 11, 2021, at https://www.latimes.com/business/autos/la-fi-hy-autos-geneva-rolls-royce-wraith-20130306-story.html.

33 Carla Tardi. (Aug. 27, 2020). *Investopedia*. “Moore’s Law.” Accessed on Jan. 11, 2021, at https://www.investopedia.com/terms/m/mooreslaw.asp.

34 Nicolas Vega. (Sept. 23. 2020). *New York Post*. “Tesla’s global network outage disables mobile app functionality.” Accessed on Jan. 11, 2021, at https://nypost-com.cdn.ampproject.org/c/s/nypost.com/2020/09/23/teslas-global-network-outage-disables-mobile-app-functionality/amp/.

35 Morton Swimmer, Fyodor Yarochkin, Joey Costoya, and Roel Reyes. (2020). *Trend Micro*. “Untangling the Web of Cloud Security Threats.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-untangling-the-web-of-cloud-security-threats.pdf.

36 Kelly Sheridan. (March 6, 2020). *DarkReading*. “7 Cloud Attack Techniques You Should Worry About.” Accessed on Jan. 11, 2021, at https://www.darkreading.com/cloud/7-cloud-attack-techniques-you-should-worry-about/d/d-id/1337259?image_number=7.

37 Mercedes-Benz, 360 Group, and Sky-Go. (n.d.). *Black Hat USA*. “Security Research Report on Mercedes-Benz Cars.” Accessed on Jan. 11, 2021, at https://i.blackhat.com/USA-20/Thursday/us-20-Yan-Security-Research-On-Mercedes-Benz-From-Hardware-To-Car-Control-wp.pdf.

38 Mercedes-Benz, 360 Group, and Sky-Go. (n.d.). *Black Hat USA*. “Security Research Report on Mercedes-Benz Cars.” Accessed on Jan. 11, 2021, at https://i.blackhat.com/USA-20/Thursday/us-20-Yan-Security-Research-On-Mercedes-Benz-From-Hardware-To-Car-Control-wp.pdf.

39 Lindsey O'Donnell. (Aug. 6, 2020). *Threatpost*. "Black Hat 2020: Mercedes-Benz E-Series Rife with 19 Bugs." Accessed on Jan. 11, 2021, at https://threatpost.com/black-hat-19-flaws-connected-mercedes-benz-vehicles/158144/.

40 Jon Fingas. (April 12, 2020). *Engadget*. "Elon Musk explains why Tesla's Model 3 has an in-cabin camera." Accessed on Jan. 11, 2021, at https://www.engadget.com/tesla-model-3-cabin-camera-explained-222017515.html.

41 Mr. Benz. (July 10, 2020). *YouTube*. "2021 S-CLASS INTERIOR + New Infotainment & First Look." Accessed on Jan. 11, 2021, at https://www.youtube.com/watch?v=EeuY_B9jQxM.

42 Trend Micro and Europol's European Cybercrime Center (EC3). (Sept. 5, 2017). *Trend Micro*. "Cashing in on ATM Malware." Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-cashing-in-on-atm-malware.pdf.

43 MITRE ATT&CK®. (Sept. 16, 2020). *MITRE ATT&CK®*. "ATT&CK®." Accessed on Jan. 11, 2021, at https://attack.mitre.org.

44 MITRE ATT&CK®. (n.d). *MITRE ATT&CK®*. "Navigator v2.3.2." Accessed on Jan. 11, 2021, at https://mitre-attack.github.io/attack-navigator/v2/enterprise/#.

45 GitHub. (2020). *GitHub, Inc*. "ATT&CK® Navigator." Accessed on Jan. 11, 2021, at https://github.com/mitre-attack/attack-navigator.

46 Apple. (n.d.). *Apple, Inc*. "CarPlay." Accessed on Jan. 11, 2021, at https://www.apple.com/ca/ios/carplay/.

47 Tesla. (Feb. 13, 2019). *Tesla*. "Sentry Mode: Guarding Your Tesla." Accessed on Jan. 11, 2021, at https://www.tesla.com/en_CA/blog/sentry-mode-guarding-your-tesla.

48 Robert Siciliano. (n.d.). *Hotspot Shield*. "Watch this guy hack and steal a Tesla Model S in seconds." Accessed on Jan. 11, 2021, at https://www.hotspotshield.com/blog/tesla-model-s-hack/.

49 Justin Gerdes. (Nov. 8, 2019). *Greentech Media*. "Will Your EV Keep the Lights On When the Grid Goes Down?" Accessed on Jan. 11, 2021, https://www.greentechmedia.com/articles/read/will-your-ev-keep-the-lights-on-when-the-grid-goes-down.

50 Tesla. (n.d.). *Tesla*. "Support: Connectivity." Accessed on June 23, 2020, at https://www.tesla.com/support/connectivity.

51 Nicolas Vega. (Sept. 23. 2020). *New York Post*. "Tesla's global network outage disables mobile app functionality." Accessed on ___, at https://nypost-com.cdn.ampproject.org/c/s/nypost.com/2020/09/23/teslas-global-network-outage-disables-mobile-app-functionality/amp/.

52 Fred Lambert. (March 8, 2017). *Electrek*. "Tesla's app/API has been down and/or spotty for the past 24 hours [Update: resolved]." Accessed on Jan. 11, 2021, at https://electrek.co/2017/03/08/teslas-app-apidown-andor-spotty/.

53 Craig Gibson. (Nov. 15, 2019). *Trend Micro*. "Securing 5G Through Cyber-Telecom Identity Federation." Accessed on Feb. 10, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-securing-5g-through-cyber-telecom-identity-federation.pdf.

54 United Nations Economic and Social Council. (June 23, 2020). *UNECE*. "Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regard to cyber security and of their cybersecurity management systems." Accessed on Jan. 11, 2021, at http://www.unece.org/fileadmin/DAM/trans/doc/2020/wp29grva/ECE-TRANS-WP29-2020-079-Revised.pdf.

55 Lo/Jack®. (n.d.). *Lo/Jack®*. "Connected Car & Stolen Vehicle Recovery." Accessed on Jan. 11, 2021, at https://www.lojack.com/products/.

56 Warwick Goodall et al. (2017). *Deloitte Review*. "The rise of mobility as a service." Accessed on Jan. 11, 2021, at https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/consumer-business/deloitte-nl-cb-ths-rise-of-mobility-as-a-service.pdf.

57 Tesla. (n.d.). *Tesla*. "Support: Connectivity." Accessed on June 23, 2020, at https://www.tesla.com/support/connectivity.

58 Gareth Corfield. (Nov. 16, 2019). *The Register*. "5G SIM-swap attacks could be even worse for industrial IoT than now." Accessed on Jan. 11, 2021, at https://www.theregister.com/2019/11/16/5g_iot_report/.

59 Craig Gibson and Europol. (Oct. 23, 2018). *Trend Micro*. "Toll Fraud, International Revenue Share Fraud and More: How Criminals Monetize Hacked Cellphones and IoT Devices for Telecom Fraud." Accessed on Jan. 11, 2021, at https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/toll-fraud-irsf-criminals-monetize-hacked-phones-iot-devices-telecom-fraud.

60 Aimee Chanthadavong. (June 24, 2020). *ZDNet*. "Nvidia and Mercedes to roll out software-defined autonomous vehicles by 2024." Accessed on Jan. 11, 2021, at https://www.zdnet.com/article/nvidia-and-mercedes-benz-to-roll-out-software-defined-autonomous-vehicles-by-2024.

61 Kari Paul. (Feb. 19, 2020). *The Guardian*. “My smart car rental was a breeze – until I got trapped in the woods.” Accessed on Jan. 11, 2021, at https://www.theguardian.com/technology/2020/feb/18/smart-car-gig-rental-app-trapped.

62 Sarah White. (Nov. 2, 2017). *CIO*. “What is geofencing? Putting location to work.” Accessed on Jan. 11, 2021, at https://www.cio.com/article/2383123/geofencing-explained.html.

63 Loukia Papadopoulos. (June 20, 2020). *Interesting Engineering*. “Tesla’s Head of AI Says The Firm Uses a Harder Approach to Self-Driving for Scalability Reasons.” Accessed on Jan. 11, 2021, at https://interestingengineering.com/teslas-head-of-ai-says-the-firm-uses-a-harder-approach-to-self-driving-for-scalability-reasons?utm_source=Facebookandutm_medium=Articleandutm_campaign=organicandutm_content=Jun20.

64 Kaiser Larsen. (Sept. 13, 2018). *AWS Machine Learning Blog*. “Mapillary uses Amazon Rekognition to work towards building parking solutions for US cities.” Accessed on Jan. 11, 2021, at https://aws.amazon.com/blogs/machine-learning/mapillary-uses-amazon-rekognition-to-work-towards-building-parking-solutions-for-us-cities/.

65 Trend Micro Research, United Nations Interregional Crime and Justice Research Institute (UNICRI), and Europol’s European Cybercrime Centre (EC3). (Nov. 19, 2020). *Trend Micro*. “Malicious Uses and Abuses of Artificial Intelligence.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-malicious-uses-and-abuses-of-artificial-intelligence.pdf.

66 Katyanna Quach. (Feb. 20, 2020). *The Register*. “Researchers trick Tesla into massively breaking the speed limit by sticking a 2-inch piece of electrical tape on a sign.” Accessed on July 20, 2020, at https://www.theregister.com/2020/02/20/tesla_ai_tricked_85_mph/.

67 Chris Young. (June 10, 2020). *Interesting Engineering*. “Tesla Semi Truck Moving Into ‘Volume Production,’ Says Elon Musk in Leaked Memo.” Accessed on Jan. 11, 2021, at https://interestingengineering.com/tesla-semi-truck-moving-into-volume-production-says-elon-musk-in-leaked-memo.

68 Simon Weckert. (n.d.) *Simon Weckert*. “About.” Accessed on Jan. 11, 2021, at http://simonweckert.com/about.html.

69 Simon Weckert. (Feb 1, 2020). *Youtube*. “Google Maps Hacks by Simon Weckert video.” Accessed on Jan. 11, 2021, at https://www.youtube.com/watch?v=k5eL_al_m7Q.

70 Felix Gary Gray (director). (2003). *The Italian Job movie*. “The Italian Job.”

71 Trend Micro Research, United Nations Interregional Crime and Justice Research Institute (UNICRI), and Europol’s European Cybercrime Centre (EC3). (n.d.) *Trend Micro*. “Malicious Uses and Abuses of Artificial Intelligence.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-malicious-uses-and-abuses-of-artificial-intelligence.pdf.

72 Crypto World Club. (Feb. 26, 2020). *CryptoWorldClub*. “CoinDesk Explains SIM Jacking or ‘SIM Hacks.’” Accessed on Jan. 11, 2021, at https://cryptoworld.club/coindesk-explains-sim-jacking-or-sim-hacks/.

73 Andreas Floemer. (May 7, 2020). *t3n Magazin*. “BMW OS 7: Großes Update bringt Android Auto und mehr auf über 750.000 Fahrzeuge.” Accessed on Jan. 11, 2021, at https://t3n.de/news/schlummernde-funktionen-abomodell-1296599/.

74 Cherise Threewitt. (June 3, 2020). *HowStuffWorks*. “Ford Software Update Lets Cop Cars Cook Away Coronavirus with Heat.” Accessed on Jan. 11, 2021, at https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/ford-police-car-coronavirus.htm.

75 Tracy Camp, Jeff Boleng, and Vanessa Davies. (Sept. 11, 2002). *Wiley Online Library*. “A survey of mobility models for ad hoc network research.” Accessed on Jan. 11, 2021, at https://onlinelibrary.wiley.com/doi/pdfdirect/10.1002/wcm.72.

76 Techopedia. (March 6, 2013). *Techopedia*. “Nomadic Wireless.” Accessed on Jan. 11, 2021, at https://www.techopedia.com/definition/2961/nomadic-wireless.

77 Carritech Telecommunications. (Jan. 28, 2019). *Carritech Telecommunications*. “What is OTT and how is it impacting Telecom Service Providers?” Accessed on Jan. 11, 2021, at https://www.carritech.com/news/ott-telecom-providers/.

78 Brad Templeton. (June 26, 2020). *Forbes*. “Amazon Buys Self-Driving Company Zoox For $1.2B And May Rule The World.” Accessed on Jan. 11, 2021, at https://www.forbes.com/sites/bradtempleton/2020/06/26/amazon-buys-self-driving-company-zoox-for-12b-and-may-rule-the-world/?sh=6560149e769c.

79 Brenda Goh and Yilei Sun. (July 9, 2020). *Reuters*. “Tesla ‘very close’ to level 5 autonomous driving technology, Musk says.” Accessed on Jan. 11, 2021, at https://www.reuters.com/article/us-tesla-autonomous/tesla-very-close-to-level-5-autonomous-driving-technology-musk-says-idUSKBN24A0HE.

80 Margaret Rouse. (April 2014). *TechTarget*. “mobile virtual network operator (MVNO).” Accessed on Jan. 11, 2021, at https://whatis.techtarget.com/definition/mobile-virtual-network-operator-MVNO.

81 EC-Council. (n.d.). *EC-Council*. “What is DNS Hijacking and How to Combat It.” Accessed on Jan. 11, 2021, at https://blog.eccouncil.org/what-is-dns-hijacking-and-how-to-combat-it/.

82 Federal Communications Commission. (n.d.) *Federal Communications Commission*. “Cell Phone Fraud.” Accessed on Jan. 11, 2021, at https://www.fcc.gov/consumers/guides/cell-phone-fraud.

83 Cambridge Dictionary. (n.d.). *Cambridge Dictionary*. “Wiretap.” Accessed on Jan. 11, 2021, at https://dictionary.cambridge.org/dictionary/english/wiretap.

84 Derrick Rountree. (2013). *ScienceDirect*. “Federated Identity.” Accessed on Jan. 11, 2021, at https://www.sciencedirect.com/topics/computer-science/federated-identity.

85 Sierra Rogers. (Nov. 11, 2020). *Capterra*. “Single Sign-On: What It Is, How It Works, and Why You Need It.” Accessed on Jan. 11, 2021, at https://blog.capterra.com/single-sign-on/.

86 Bluetooth. (n.d.). *Bluetooth*. “SAP and Remote Network Access.” Accessed on Jan. 11, 2021, at https://www.bluetooth.com/bluetooth-resources/sap-and-remote-network-access/.

87 O’Reilly Media. (n.d.) *O’Reilly Media*. “Home eNB (HeNB).” Accessed on Jan. 11, 2021, at https://www.oreilly.com/library/view/radio-protocols-for/9781118188569/chapter09.html.

88 Marco Balduzzi et al. (Aug. 5, 2020). *Trend Micro*. “Lost in Translation: When Industrial Protocol Translation Goes Wrong.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-lost-in-translation-when-industrial-protocol-translation-goes-wrong.pdf.

89 Terry Young. (Jan. 30, 2020). *ITProPortal*. “Why roaming still matters in a 5G world.” Accessed on Jan. 11, 2021, at https://www.itproportal.com/features/why-roaming-still-matters-in-a-5g-world/.

90 GSMA. (n.d.). *GSM Association*. “GSMA.” Accessed on Jan. 11, 2021, at https://www.gsma.com.

91 Techopedia. (n.d.). *Techopedia*. “Mobile Phone Spam.” Accessed on Jan. 11, 2021, at https://www.techopedia.com/definition/16072/mobile-phone-spam.

92 3G4G. (March. 1, 2020). *The 3G4G Blog*. “5G Private and Non-Public Network (NPN).” Accessed on Jan. 11, 2021, at https://blog.3g4g.co.uk/2020/03/5g-private-and-non-public-network-npn.html.

93 Craig Gibson. (Nov. 15, 2019). *Trend Micro*. “Securing 5G Through Cyber-Telecom Identity Federation.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-securing-5g-through-cyber-telecom-identity-federation.pdf.

94 Blue Origin. (n.d.). *Blue Origin*. “Blue Origin.” Accessed on Jan. 11, 2021, at https://www.blueorigin.com.

95 Starlink. (n.d.). *Starlink*. “Starlink.” Accessed on Jan. 11, 2021, at https://www.starlink.com.

96 Craig Gibson. (June 11, 2018). *Trend Micro*. “IoT and Satellite Security in the Age of 5G.” Accessed on Jan. 11, 2021, at https://blog.trendmicro.com/trendlabs-security-intelligence/attack-vectors-in-orbit-need-for-satellite-security-in-5g-iot/.

97 GSMA. (n.d.). *GSM Association*. “Smart Cities Transport.” Accessed on Jan. 11, 2021, at https://www.gsma.com/iot/smart-cities-resources/smart-cities-transport/.

98 Trend Micro Research, United Nations Interregional Crime and Justice Research Institute (UNICRI), and Europol’s European Cybercrime Centre (EC3). (Nov. 19, 2020). *Trend Micro*. “Malicious Uses and Abuses of Artificial Intelligence.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-malicious-uses-and-abuses-of-artificial-intelligence.pdf.

99 Vit Sembera. (June 29, 2020). *Trend Micro*. “ISO/SAE 21434: Setting the Standard for Connected Cars’ Cybersecurity.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-setting-the-standard-for-connected-cars-cybersecurity.pdf.

100 Trend Micro Research and Europol’s European Cybercrime Centre (EC3). (March 21, 2019). *Europol*. “Cyber-telecom Crime Report 2019.” Accessed on Jan. 11, 2021, at https://www.europol.europa.eu/publications-documents/cyber-telecom-crime-report-2019.

101 SCADAHacker. (n.d.). *SCADAHacker*. “Assessing the Security of ICS Using Threat Modeling.” Accessed on June 25, 2020, at https://scadahacker.com/howto/howto-threatmodeling.html.

102 Shodan. (n.d.). *Shodan*. “Shodan.” Accessed on June 25, 2020, at https://www.shodan.io/.

103 Trend Micro. (n.d.). *Trend Micro*. “SQL injection.” Accessed on June 25, 2020, at https://www.trendmicro.com/vinfo/us/security/definition/sql-injection.

104 Trend Micro. (n.d.). *Trend Micro*. “Cross-site scripting (XSS).” Accessed on June 25, 2020, at https://www.trendmicro.com/vinfo/us/security/definition/cross-site-scripting-(xss).

105 OWASP. (n.d.). *OWASP*. “Session hijacking attack.” Accessed on June 25, 2020, at https://owasp.org/www-community/attacks/Session_hijacking_attack.

106 Justin Jett. (Feb. 15, 2019). *Threatpost*. “Tips on How to Fight Back Against DNS Spoofing Attacks.” Accessed on June 25, 2020, at https://threatpost.com/dns-spoofing-attacks/141880/.

107 OWASP. (n.d.). *OWASP*. “Application threat modeling.” Accessed on June 25, 2020, at https://owasp.org/www-community/Application_Threat_Modeling.

108 David Czagan. (May 21, 2014). *InfoSec Institute*. “Qualitative Risk Analysis with the DREAD Model.” Accessed on June 25, 2020, at https://resources.infosecinstitute.com/qualitative-risk-analysis-dread-model/.

109 Trend Micro Security News. (Nov. 27, 2017). *Trend Micro Security News*. “Securing the Transportation Network of Tomorrow.” Accessed on June 25, 2020, at https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/securing-the-transportation-network-of-tomorrow.

110 Vit Sembera. (June 29, 2020). *Trend Micro*. “ISO/SAE 21434: Setting the Standard for Connected Cars’ Cybersecurity.” Accessed on Jan. 11, 2021, at https://documents.trendmicro.com/assets/white_papers/wp-setting-the-standard-for-connected-cars-cybersecurity.pdf.

111 Nuuman Huq. (March 12, 2015). *Trend Micro*. “Defending Against Pos RAM Scrapers.” Accessed on June 26, 2020, at https://documents.trendmicro.com/assets/wp/wp-defending-against-pos-ram-scrapers.pdf.

112 Venafi. (n.d.) *Venafi*. “Code Signing Certificates [Your In-Depth Guide].” Accessed on Jan. 11, 2021, at https://www.venafi.com/education-center/code-signing/what-is-code-signing.

113 VMware. (n.d.) *VMware*. “What is application security?” Accessed on Jan. 11, 2021, at https://www.vmware.com/topics/glossary/content/application-security.

114 Trend Micro. (n.d.) *Trend Micro*. “Trend Micro Cloud One™ Application Security.” Accessed on Jan. 11, 2021, at https://www.trendmicro.com/en_ca/business/products/hybrid-cloud/cloud-one-application-security.html.

115 Trend Micro. (n.d.) *Trend Micro*. “Device Control.” Accessed on Jan. 11, 2021, at http://docs.trendmicro.com/all/ent/officescan/v10.0/en-us/osce_10.0_olhsrv/osceag/osceag-scan/device_control.htm.

116 Scott Rose et al. (Aug. 2020). *National Institute of Standards and Technology (NIST)*. “Zero Trust Architecture.” Accessed on Jan. 11, 2021, at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf.

117 Scott Rose et al. (Aug. 2020). *National Institute of Standards and Technology (NIST)*. “Zero Trust Architecture.” Accessed on Jan. 11, 2021, at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf.

118 Dave Nyczepir. (n.d.) *Scoop News Group*. “NIST releases finalized zero-trust architecture guidance.” Accessed on Jan. 11, 2021, at https://www.fedscoop.com/nist-finalized-zero-trust-guidance/.

119 Trend Micro IoT Security. (n.d.). *Trend Micro IoT Security*. “Trend Micro IoT Security for Automotive.” Accessed on July 8, 2020, at https://www.trendmicro.com/us/iot-security/product/iot-security-for-auto?solutions=connected-car.

120 Trend Micro IoT Security. (n.d.). *Trend Micro IoT Security*. “Connected Car.” Accessed on Feb. 5, 2021, at https://www.trendmicro.com/en_us/business/solutions/iot/5g-and-iot.html#t4https://www.trendmicro.com/us/iotRE-security/product/trend-micro-virtual-network-function-suite?solutions=retail.

121 Trend Micro. (n.d.). *Trend Micro*. “Hybrid Cloud Security.” Accessed on July 8, 2020, at https://www.trendmicro.com/en_us/business/products/hybrid-cloud.html#t2.

122 Trend Micro IoT Security. (n.d.). *Trend Micro IoT Security*. “Trend Micro IoT Security™.” Last accessed on July 13, 2020, at https://www.trendmicro.com/us/iot-security/product/trend-micro-iotsecurity?solutions=smart-city.

123 Trend Micro. (n.d.). *Trend Micro*. “XDR: Detection and response across email, endpoints, servers, cloud workloads, and networks.” Accessed on July 8, 2020, at https://www.trendmicro.com/en_us/business/products/detection-response/xdr.html.

124 Internet of Business. (n.d.) *Internet of Business*. “Connected cars report: 125 million vehicles by 2022, 5G coming.” Accessed on Jan. 11, 2021, at https://internetofbusiness.com/worldwide-connected-car-market-to-top-125-million-by-2022/.

125 Nick Michell. (Dec. 5, 2016). *Cities Today*. “Self-driving cars to reach 14.5 million by 2025, says new study.” Accessed on Jan. 11, 2021, at https://cities-today.com/self-driving-car-production-reach-14-5-million-2025-says-new-study/.

126 Michael Morand. (March 1, 2019). *Association for Supply Chain Management (ASCM)*. “The Red Queen Hypothesis.” Accessed on Jan. 27, 2021, at https://www.ascm.org/ascm-insights/the-red-queen-hypothesis/.

127 Dr. Charlie Miller and Chris Valasek. (Aug. 10, 2015). *Illmatics*. “Remote Exploitation of an Unaltered Passenger Vehicle.” Accessed on June 24, 2020, at http://illmatics.com/Remote%20Car%20Hacking.pdf.

128 Chris Welch. (July 24, 2015). *The Verge*. “Chrysler recalls 1.4 million cars at risk of being remotely hijacked.” Accessed on June 24, 2020, at https://www.theverge.com/2015/7/24/9032179/chrysler-announces-voluntary-recall-hack.

129 Black Hat. (Dec. 29, 2015). *YouTube*. “Remote Exploitation Of An Unaltered Passenger Vehicle.” Accessed on July 8, 2020, at https://www.youtube.com/watch?v=MAcHkASmXEc.

130 Jeep. (n.d.). *Jeep*. “Uconnect® overview.” Accessed on June 24, 2020, at https://www.jeep.com/uconnect.html.

131 Scott Reeves. (Nov. 11, 2013). *TechRepublic*. “Pros and cons of using femtocells.” Accessed on June 24, 2020, at https://www.techrepublic.com/blog/data-center/pros-and-cons-of-using-femtocells/.

132 Mopar. (n.d.). *Mopar*. “Why wiTECH 2.0?” Accessed on June 24, 2020, at https://www.fcawitech.com/.

133 Kvaser. (n.d.). *Kvaser*. “CAN Messages.” Accessed on June 24, 2020, at https://www.kvaser.com/lesson/can-messages/.

134 Hideyoshi Kume. (Feb. 17, 2020). *Nikkei Asian Review*. “Tesla teardown finds electronics 6 years ahead of Toyota and VW.” Accessed on June 24, 2020, at https://asia.nikkei.com/Business/Automobiles/Tesla-teardown-finds-electronics-6-years-ahead-of-Toyota-and-VW2.

135 Sen Nie, Ling Liu, and Yuefeng Du. (n.d.). *Black Hat*. “Free-Fall: Hacking Tesla From Wireless To Can Bus.” Accessed on June 24, 2020, at https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus.pdf.

136 Sen Nie, Ling Liu, and Yuefeng Du. (n.d.). *Black Hat*. “Free-Fall: Hacking Tesla From Wireless To Can Bus.” Accessed on June 24, 2020, at https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus.pdf.

137 WebKit. (n.d.). *WebKit*. “WebKit.” Accessed on June 24, 2020, at https://webkit.org/.

138 Common Vulnerabilities and Exposures. (n.d.). *Common Vulnerabilities and Exposures*. “CVE-2013-6282.” Accessed on June 24, 2020, at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-6282.

139 Chriss Hoffman. (n.d.). *How-To Geek*. “What Is AppArmor, and How Does It Keep Ubuntu Secure?” Accessed on Jan 11, 2021, at https://www.howtogeek.com/118222/htg-explains-what-apparmor-is-and-how-it-secures-your-ubuntu-system/.

140 1SSH. (n.d.). *SSH*. “SSH (Secure Shell).” Accessed on June 24, 2020, at https://www.ssh.com/ssh/.

141 Jeff Tyson. (n.d.). *How Stuff Works*. “How Encryption Works.” Accessed on June 24, 2020, at https://computer.howstuffworks.com/encryption7.htm.

142 TechTerms. (n.d.). *TechTerms*. “UDP.” Accessed on June 24, 2020, at https://techterms.com/definition/udp.

143 Bosch. (Feb. 15, 2016). *Bosch*. “Preventing skidding: The Electronic Stability Program ESP®.” Accessed on June 24, 2020, at https://www.bosch.com/stories/the-electronic-stability-program-esp/.

144 National Instruments (NI). (n.d.). *National Instruments Corp*. “UDS (Unified Diagnostic Services).” Accessed on June 24, 2020, at http://zone.ni.com/reference/en-XX/help/372140J-01/adcs/udsunifieddiagnosticservices/.

145 Sen Nie et al. (n.d.). *Black Hat*. “Over-The-Air: How We Remotely Compromised the Gateway, BCM, and Autopilot ECUs Of Tesla Cars.” Accessed on June 24, 2020, at https://i.blackhat.com/us-18/Thu-August-9/us-18-Liu-Over-The-Air-How-We-Remotely-Compromised-The-Gateway-Bcm-And-Autopilot-Ecus-Of-Tesla-Cars-wp.pdf.

146 NXP. (February 2011). *NXP*. “Freescale MSD FATFS Users Guide.” Accessed on June 24, 2020, at https://www.nxp.com/docs/en/user-guide/MSDFATFSUG.pdf.

147 Fox Van Allen. (April 27, 2018). *CNet*. “Incredibly cool Tesla Easter eggs.” Accessed on June 24, 2020, at https://www.cnet.com/pictures/tesla-easter-eggs/.

148 Tesla. (n.d.). *Tesla*. “Future of Driving.” Accessed on June 24, 2020, at https://www.tesla.com/autopilot?redirect=no.

149 Tencent Keen Security Lab. (March 2019). *Keen Security Lab*. “Experimental Security Research of Tesla Autopilot.” Accessed on June 24, 2020, at https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf.

150 Tencent Keen Security Lab. (n.d.). *Keen Security Lab*. "Experimental Security Assessment of BMW Cars: A Summary Report." Accessed on June 24, 2020, at https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf.

151 Tencent Keen Security Lab. (n.d.). *Black Hat*. "0-Days & Mitigations: Roadways to Exploit and Secure Connected BMW Cars." Accessed on June 24, 2020, at https://i.blackhat.com/USA-19/Thursday/us-19-Cai-0-Days-And-Mitigations-Roadways-To-Exploit-And-Secure-Connected-BMW-Cars.pdf.

152 The OBD II Home Page. (n.d.). *The OBD II Home Page*. "OBD-II Background." Accessed on June 24, 2020, at http://www.obdii.com/background.html.

153 BMW USA. (n.d.). *BMW USA*. "BMW ConnectedDrive." Accessed on June 24, 2020, at https://www.bmwusa.com/explore/connecteddrive.html.

154 Newtis.info. (n.d.). *Newtis.info*. "Telematic Communication Box." Accessed on June 25, 2020, at https://www.newtis.info/tisv2/a/en/e90-320d-lim/wiring-functional-info/body/audio-video-telephone-navigation-most-ring/telecommunications/documents/Jxfm2y8o.

155 Sascha Segan. (April 7, 2020). *PCMag*. "CDMA vs. GSM: What's the Difference?" Accessed on June 25, 2020, at https://www.pcmag.com/news/cdma-vs-gsm-whats-the-difference.

156 Common Weakness Enumeration. (n.d.). *Common Weakness Enumeration*. "CWE-367: Time-of-check Time-of-use (TOCTOU) Race Condition." Accessed on June 25, 2020, at https://cwe.mitre.org/data/definitions/367.html.

157 QNX. (n.d.). *QNX*. "What is Qnet?" Accessed on July 8, 2020, at http://www.qnx.com/developers/docs/qnxcar2/index.jsp?topic=%2Fcom.qnx.doc.neutrino.prog%2Ftopic%2Fqnet_WQCDFY.html.

158 PressClub USA. (Aug. 1, 2008). *BMW Group*. "Bmw Presents Open Source Telematics Protocol." Accessed on June 25, 2020, at https://www.press.bmwgroup.com/usa/article/detail/T0017923EN_US/bmw-presents-open-source-telematicsprotocol?language=en_US.

159 Takuji Hara, Norio Kambayashi, and Noboru Matsushima. (n.d.). *Google Books*. "Industrial Innovation in Japan." Accessed on June 25, 2020, at https://books.google.com.ph/books/about/Innovation_in_Japan.html?id=TYtoMguVdl0C&redir_esc=y.

160 Alessio Filippi, Kees Moerman, Gerardo Daalderop, Paul D. Alexander, Franz Schober, and Werner Pfliegl. (n.d.) *Siemens*. "Ready to roll: Why 802.11p beats LTE and 5G for V2x." Accessed on Jan. 11, 2021, at https://assets.new.siemens.com/siemens/assets/api/uuid:ab5935c545ee430a94910921b8ec75f3c17bab6c/its-g5-ready-to-roll-en.pdf.

161 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. "Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany)." Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

162 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. "Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany)." Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

163 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. "Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany)." Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

164 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. "Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany)." Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

165 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. "Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany)." Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

166 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. "Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany)." Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

167 Tesla. (Sept. 30, 2019) *YouTube*. "Smart Summon." Accessed on Jan. 11, 2021, at https://www.youtube.com/watch?v=nlCQG2rg4sw.

168 Andrew J. Hawkins. (Sept. 30, 2019). *The Verge*. “Tesla’s Smart Summon feature is already causing chaos in parking lots across America.” Accessed on Jan. 11, 2021, at https://www.theverge.com/2019/9/30/20891343/tesla-smart-summon-feature-videos-parking-accidents.

169 Dr. Yunpeng Zang (on behalf of Dr. Mikael Fallgren). (March 2018). *5GCAR*. “Session 3 5GCAR (Presentation for the 9th ETSI ITS Workshop, 6-8 March 2018, in Berlin, Germany).” Accessed on Jan. 11, 2021, at https://docbox.etsi.org/Workshop/2018/20180306_ITS_WORKSHOP/S03_FRAMING_NEEDS_AUTOMATED-DRIV/5GCAR_ERICSSON_ZANG.pdf.

170 Fatih Sakiz and Sevil Sen. (March 2017) *ResearchGate*. “A Survey of Attacks and Detection Mechanisms on Intelligent Transportation Systems: VANETs and IoV.” Accessed on Jan. 11, 2021, at https://www.researchgate.net/publication/315439003_A_Survey_of_Attacks_and_Detection_Mechanisms_on_Intelligent_Transportation_Systems_VANETs_and_IoV.

171 Fatih Sakiz and Sevil Sen. (March 2017) *ResearchGate*. “A Survey of Attacks and Detection Mechanisms on Intelligent Transportation Systems: VANETs and IoV.” Accessed on Jan. 11, 2021, at https://www.researchgate.net/publication/315439003_A_Survey_of_Attacks_and_Detection_Mechanisms_on_Intelligent_Transportation_Systems_VANETs_and_IoV.

172 Fatih Sakiz and Sevil Sen. (March 2017) *ResearchGate*. “A Survey of Attacks and Detection Mechanisms on Intelligent Transportation Systems: VANETs and IoV.” Accessed on Jan. 11, 2021, at https://www.researchgate.net/publication/315439003_A_Survey_of_Attacks_and_Detection_Mechanisms_on_Intelligent_Transportation_Systems_VANETs_and_IoV.

173 Fatih Sakiz and Sevil Sen. (March 2017) *ResearchGate*. “A Survey of Attacks and Detection Mechanisms on Intelligent Transportation Systems: VANETs and IoV.” Accessed on Jan. 11, 2021, at https://www.researchgate.net/publication/315439003_A_Survey_of_Attacks_and_Detection_Mechanisms_on_Intelligent_Transportation_Systems_VANETs_and_IoV.

TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threat techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com