Driving Security Into Connected Cars: Threat Model and Recommendations

Numaan Huq, Craig Gibson, Rainer Vosseler

TREND MICRO | research

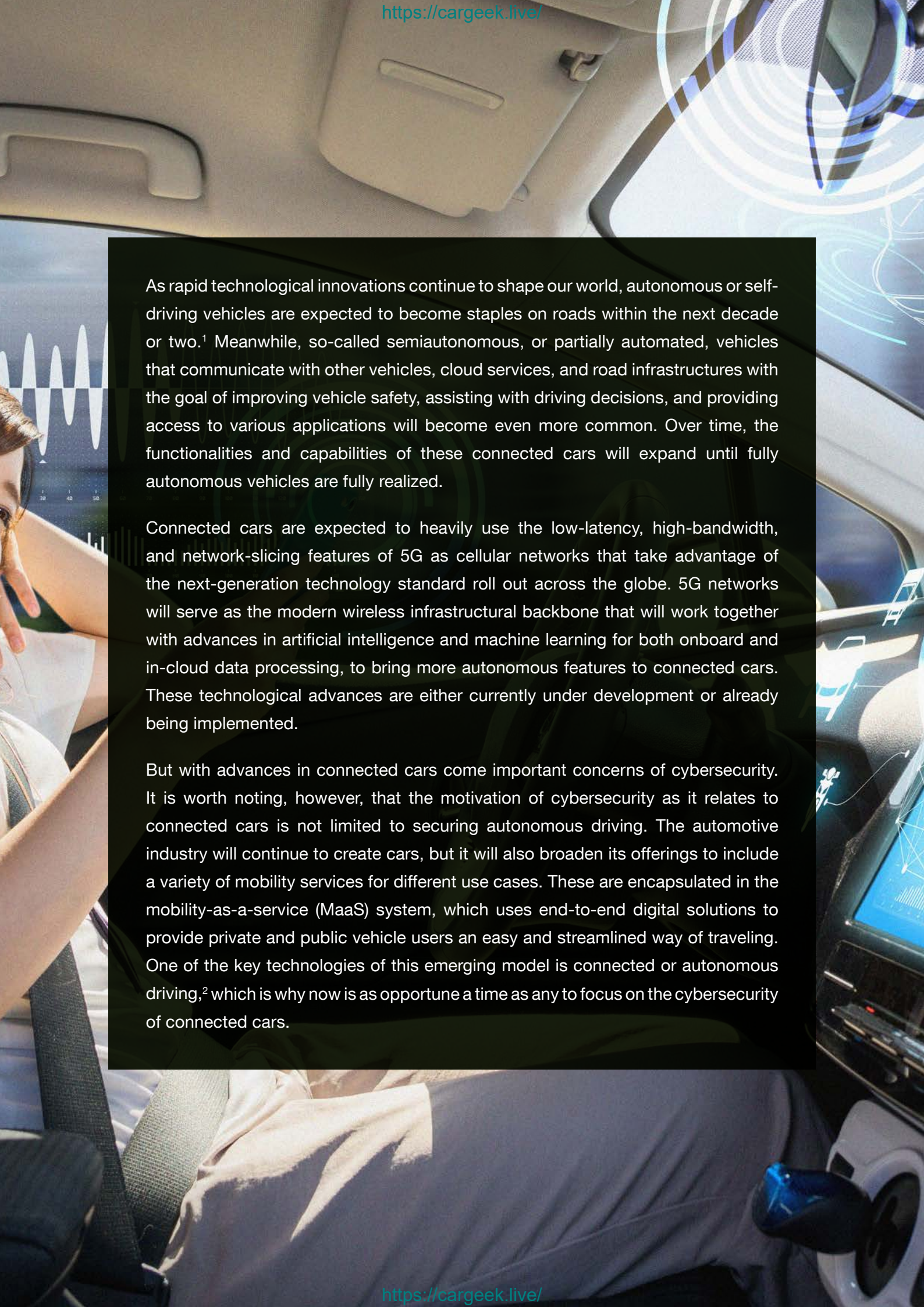
Published by
Trend Micro Research

Written by
Numaan Huq
Craig Gibson
Rainer Vosseler

Stock image used under license from
Shutterstock.com

For Raimund Genes (1963 – 2017)

Contents



As rapid technological innovations continue to shape our world, autonomous or self-driving vehicles are expected to become staples on roads within the next decade or two.[1] Meanwhile, so-called semiautonomous, or partially automated, vehicles that communicate with other vehicles, cloud services, and road infrastructures with the goal of improving vehicle safety, assisting with driving decisions, and providing access to various applications will become even more common. Over time, the functionalities and capabilities of these connected cars will expand until fully autonomous vehicles are fully realized.

Connected cars are expected to heavily use the low-latency, high-bandwidth, and network-slicing features of 5G as cellular networks that take advantage of the next-generation technology standard roll out across the globe. 5G networks will serve as the modern wireless infrastructural backbone that will work together with advances in artificial intelligence and machine learning for both onboard and in-cloud data processing, to bring more autonomous features to connected cars. These technological advances are either currently under development or already being implemented.

But with advances in connected cars come important concerns of cybersecurity. It is worth noting, however, that the motivation of cybersecurity as it relates to connected cars is not limited to securing autonomous driving. The automotive industry will continue to create cars, but it will also broaden its offerings to include a variety of mobility services for different use cases. These are encapsulated in the mobility-as-a-service (MaaS) system, which uses end-to-end digital solutions to provide private and public vehicle users an easy and streamlined way of traveling. One of the key technologies of this emerging model is connected or autonomous driving,[2] which is why now is as opportune a time as any to focus on the cybersecurity of connected cars.

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]

Connected Car Technologies

Contrary to common belief, connectivity in cars is far from a newfangled concept. In fact, today's common car already comes with a wide variety of connected technologies. The following provides a discussion of some of them, including upcoming technologies as well as ones that are already in use.

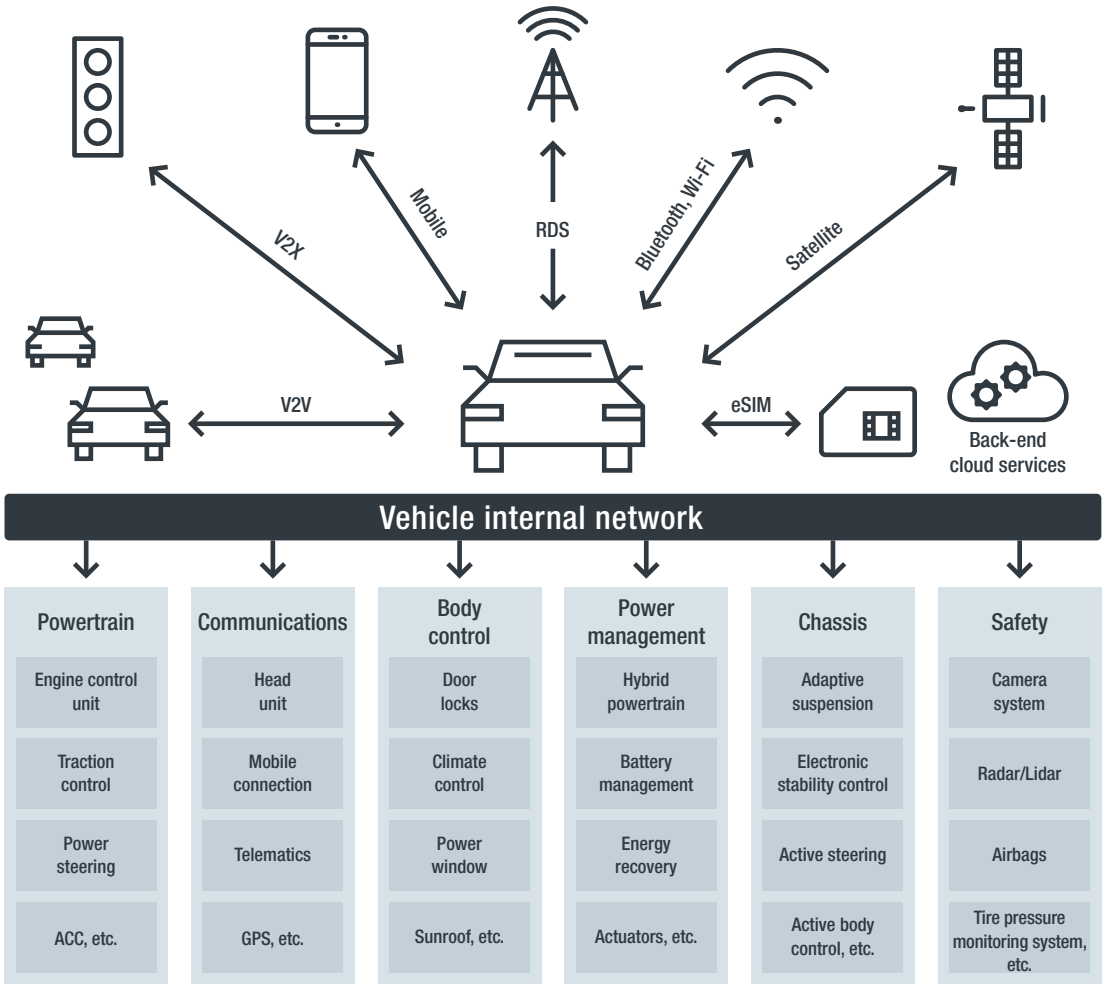


Figure 1. The technologies and functionalities that make up the internal network of a connected car

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, 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.

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 2, 3, and 4 illustrate three examples of car network architectures.

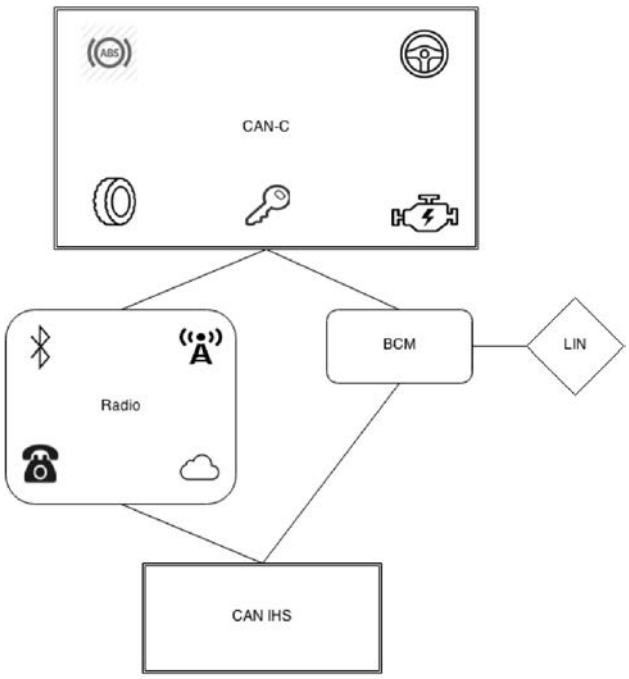


Figure 2. The Jeep Cherokee network architecture that the researchers Charlie Miller and Chris Valasek compromised in 2015

Image credit: Charlie Miller and Chris Valasek[16]

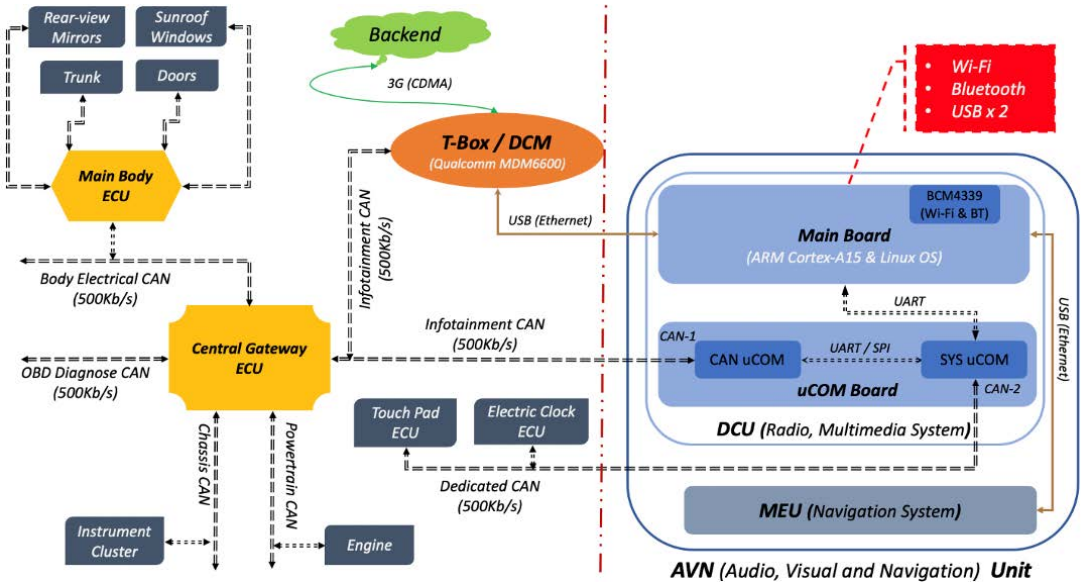


Figure 3. The Lexus network architecture that Tencent Keen Security Lab compromised in 2020

Image credit: Tencent Keen Security Lab[17]

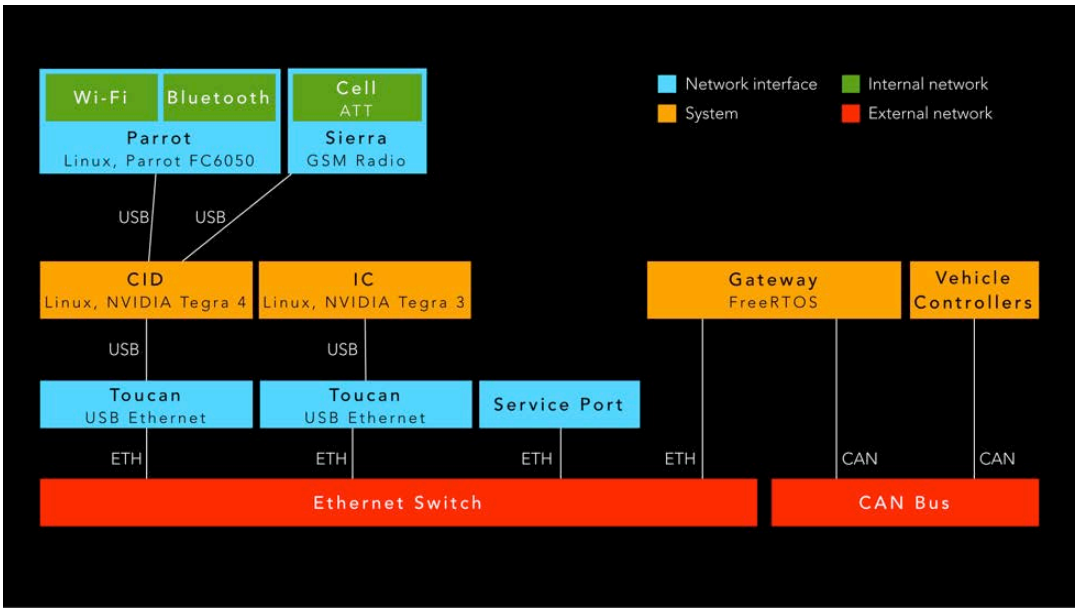


Figure 4. The Tesla network architecture that the researchers Kevin Mahaffey and Marc Rogers explored in 2015

Image credit: Lookout[18]

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.

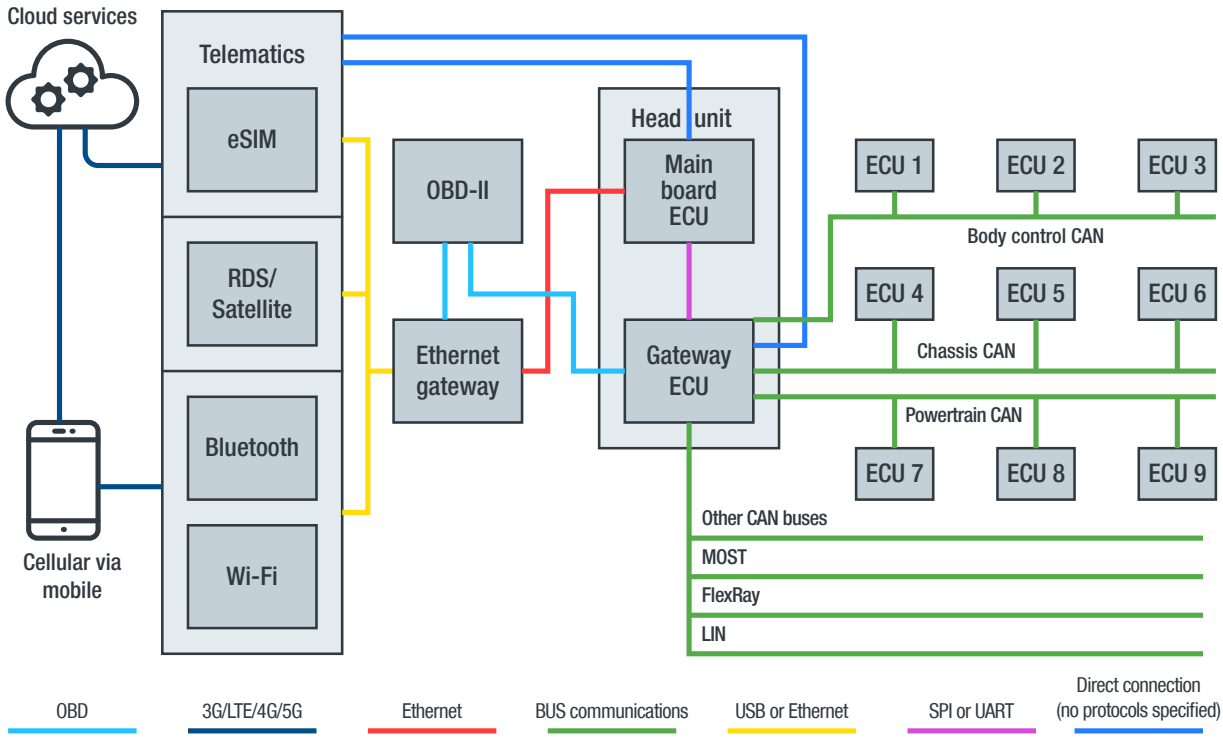


Figure 5. A theoretical visualization of a generic network architecture for a modern-day connected car

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 RDS-TMC, 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.[19]
- **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[20] or the universal asynchronous receiver-transmitter (UART)[21] 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.

Research on Remote Vehicle Attacks

In 2017, we released a research paper that studied cyberattacks on intelligent transportation systems (ITSs).[22] The IT security industry had long been researching on car hacking techniques — exploring attack vectors for modern connected cars that enable malicious actors to seize control of critical vehicle functions and implant and/or steal data. In our research on ITSs, our strategy was to look beyond vehicle attack vectors and instead study cyberthreats to the entire road operating ecosystem. Our motivation at that time was that there was very little published research discussing cyberthreats to ITSs. We applied our knowledge of cyberattacks to hypothesize, develop, and analyze cyberattack scenarios involving ITSs.

This current research is a follow-up to that earlier research. Now, we study the cybersecurity risks posed by connected cars interacting with other vehicles, cloud services, and road infrastructures. Given the continuing expansion of the IoT and the ever-increasing volume of disruptive and destructive cyberattacks, connected car cybersecurity should be made mandatory and should be considered a fundamental aspect of V2X architectures and frameworks. By identifying and addressing the cybersecurity risks that connected cars face in their early development stages, cybersecurity professionals have the opportunity to influence legislative as well as technological developments related to connected cars.

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 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 tactics, techniques, and procedures (TTPs) used by hackers, which we then apply to our threat modeling analysis.

Case Studies on 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 give an indication of the limitations of what can or cannot be hacked, and the level of difficulty in which today's connected cars can be hacked. 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 of 2015

"Remote Exploitation of an Unaltered Passenger Vehicle,"[23] 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.[24] 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.

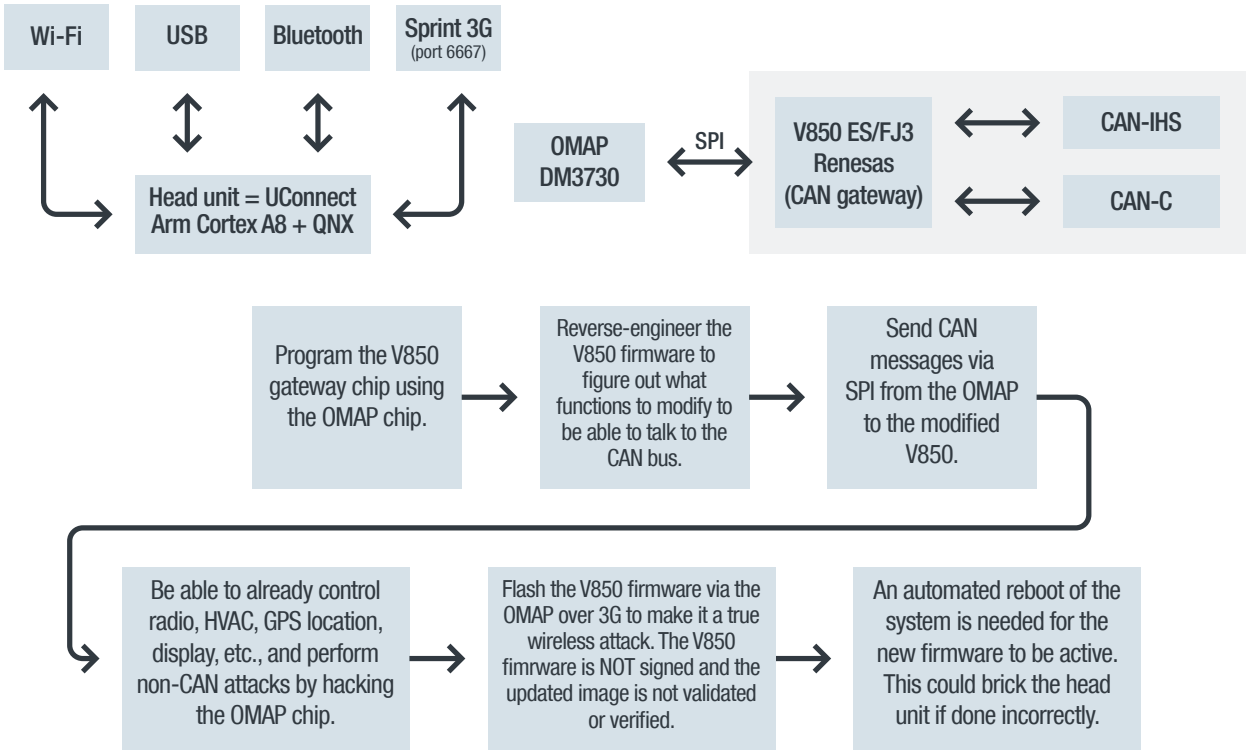


Figure 6. The attack chain of the Jeep Cherokee remote hack of 2015

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.[25]
- 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,[26] 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,[27] 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,[28] 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.[29]

- 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 of 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.[30] 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.[31] 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.

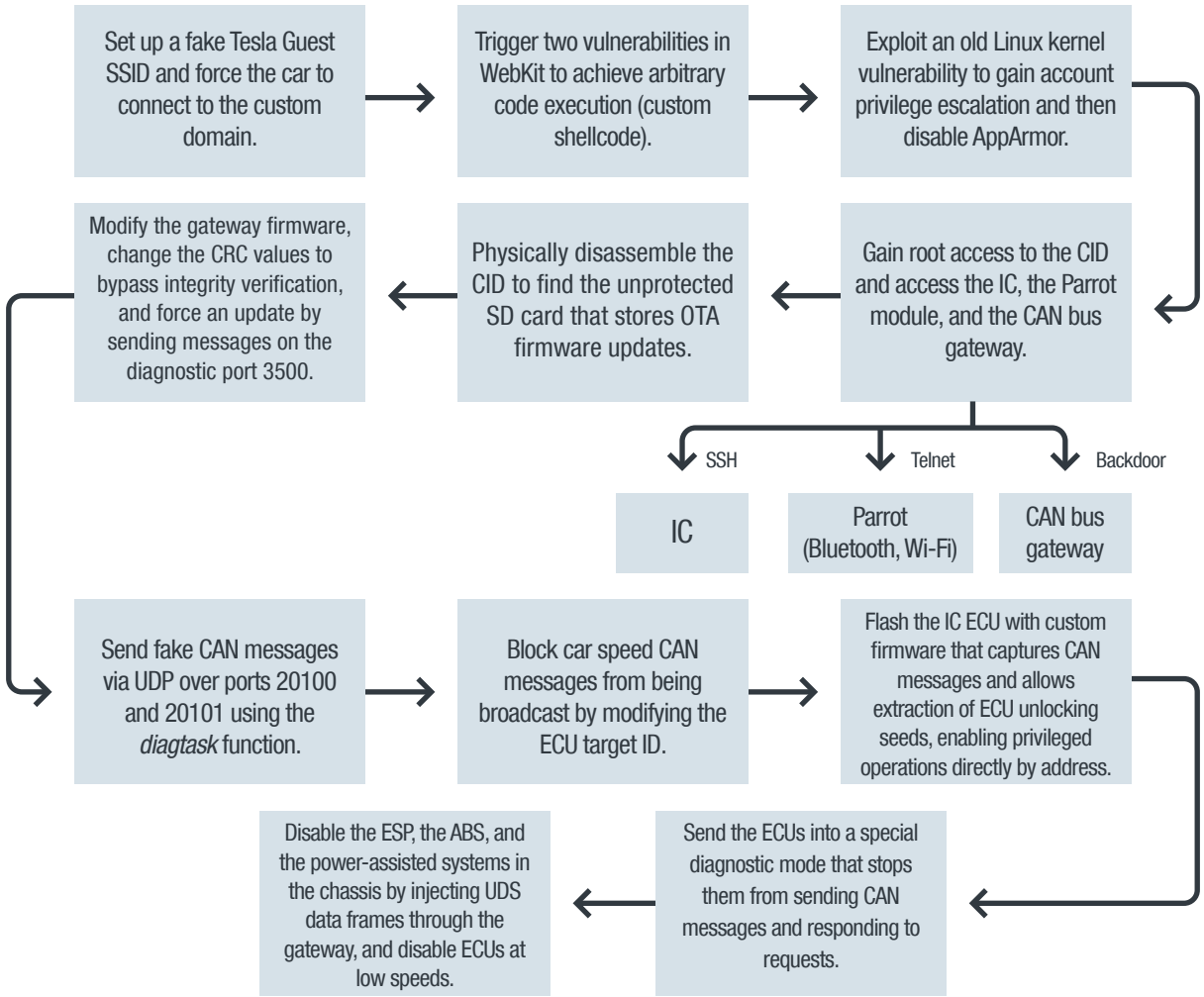


Figure 7. The attack chain of the Tesla Model S remote hack of 2016

The following summarizes the attack chain that the researchers used to compromise the Tesla Model S:[32]

- 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.[33] 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,[34] to gain a higher privilege than the one granted to the browser. The researchers then disabled the kernel security module AppArmor.[35]
- 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),[36] 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)[37] 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)[38] 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),[39] and the power-assisted systems in the chassis by injecting UDS (Unified Diagnostic Services)[40] data frames through the gateway, and disabled ECUs at low speeds.

The Tesla Hack of 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 Model S. 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.

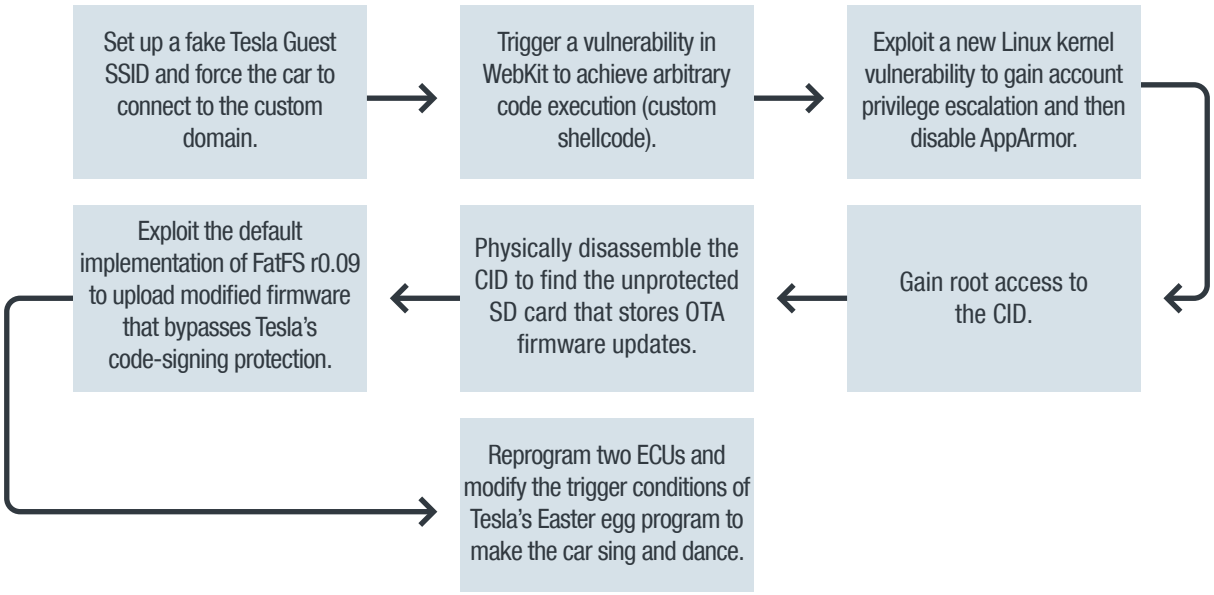


Figure 8. The attack chain of the Tesla Model S remote hack of 2017

The following summarizes the attack chain that Keen Security Lab used to compromise the Tesla Model S and the Tesla Model X:[41]

- 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[42] 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.[43] 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).[44] They wrote and released a separate research paper on it,[45] but it is not covered in our research.

The BMW Hack of 2018

After hacking Tesla vehicles for two consecutive years, Keen Security Lab shifted their focus to hacking BMW vehicles.[46, 47] They created three attack chains: one for a local attack via the USB/OBD-II[48] port and one each for two remote attacks. Because our research is on connected cars, we focused on the two remote attack chains.

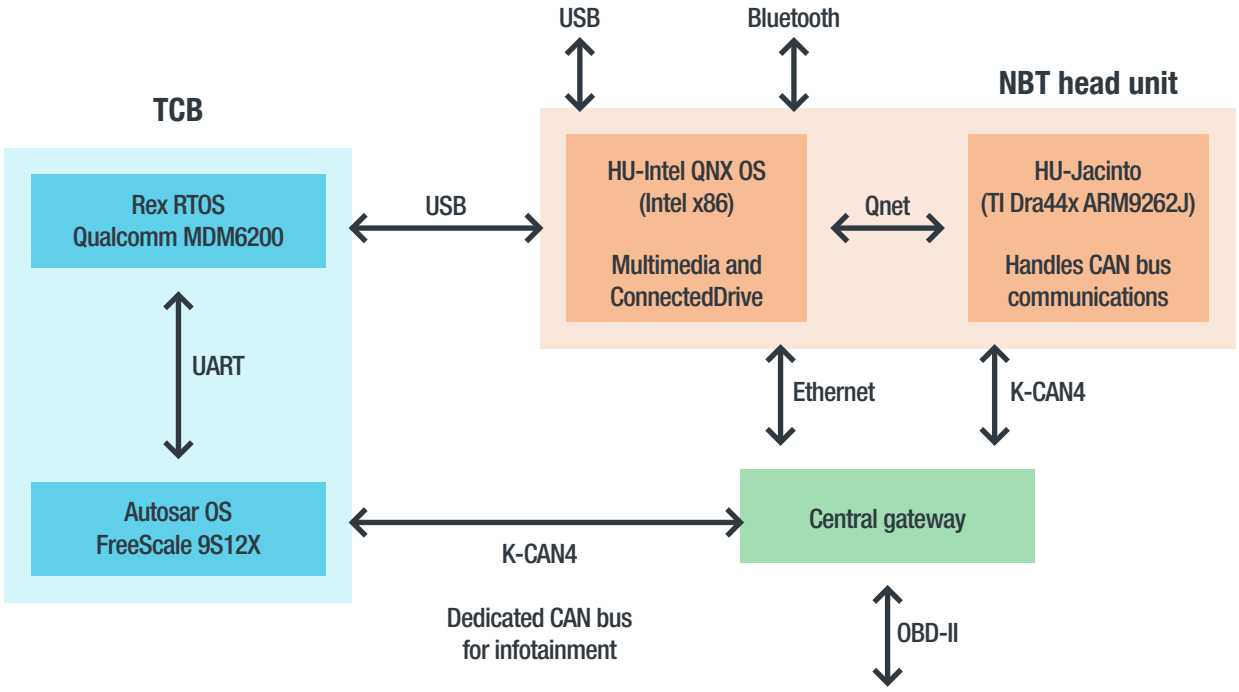


Figure 9. The attack chain of the BMW remote hack of 2018

The first remote attack chain achieved remote code execution (RCE) in BMW ConnectedDrive[49] 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 telematic communication box (TCB)[50] over HTTP. The researchers set up a fake GSM[51] 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)[52] 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,[53] 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)[54] 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).[55]
- 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.

Generalized Remote Hacking Techniques for Connected Cars

From the four case studies, we found an emerging attack pattern, illustrated in Figure 10, that all four hacks used in compromising the connected cars and sending malicious CAN messages to the ECUs.

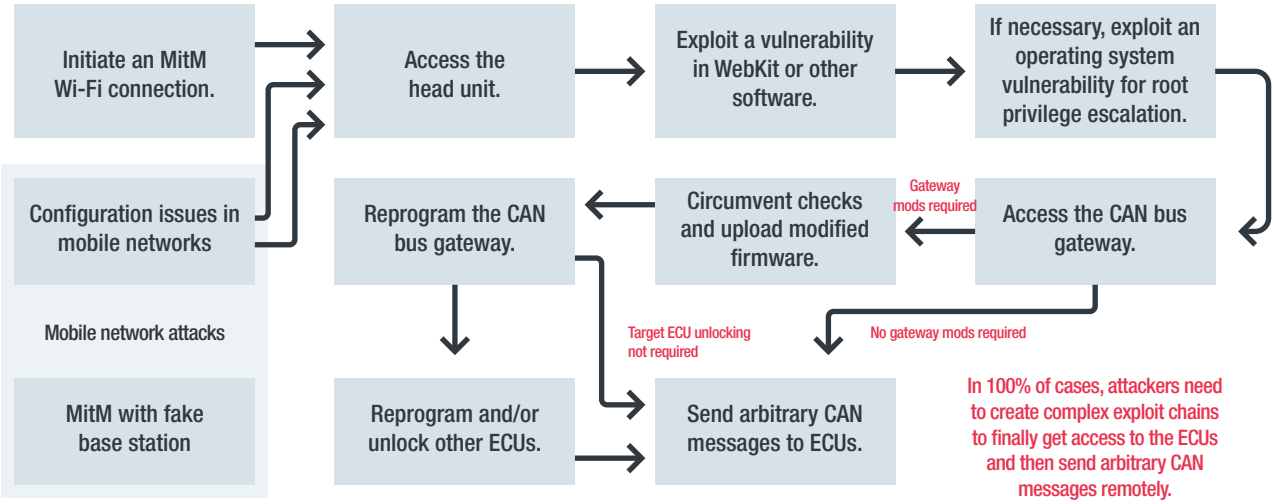


Figure 10. A generalized remote hacking attack chain based on the featured remote attack case studies

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[56] 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 are able to 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.

Threat Model for 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.[57] 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 not just smoothly but securely as well.

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
- Installing a malicious app 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[58]
- 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,[59] cross-site scripting (XSS),[60] session hijacking,[61] and DNS (Domain Name System) spoofing[62]
- 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 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.

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.[63] The DREAD threat model can be used to perform qualitative risk analysis,[64] 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 1 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 1. 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.

Measuring the Risks of Attacks on Connected Cars

For each of the connected car attacks we 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.
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).
Installing a malicious app 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.

Attack vector	D	R	E	A	D	Rating	Remarks
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.
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 2. Connected car threat modeling using the DREAD threat model

Based on the results of our threat modeling exercise, we made 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 the 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 we are assessing these hypothesized attacks 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, we are bound to see another evolution in their TTPs that will lead to higher-risk attacks.

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 — composed of the vehicle, the network, the back end, and the vehicle security operations center (VSOC), which processes and analyzes notifications and data from the first three components — paves the way for large and at times unpredictable attack surfaces. Protecting connected cars against 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 they are 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:[65]

- 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.

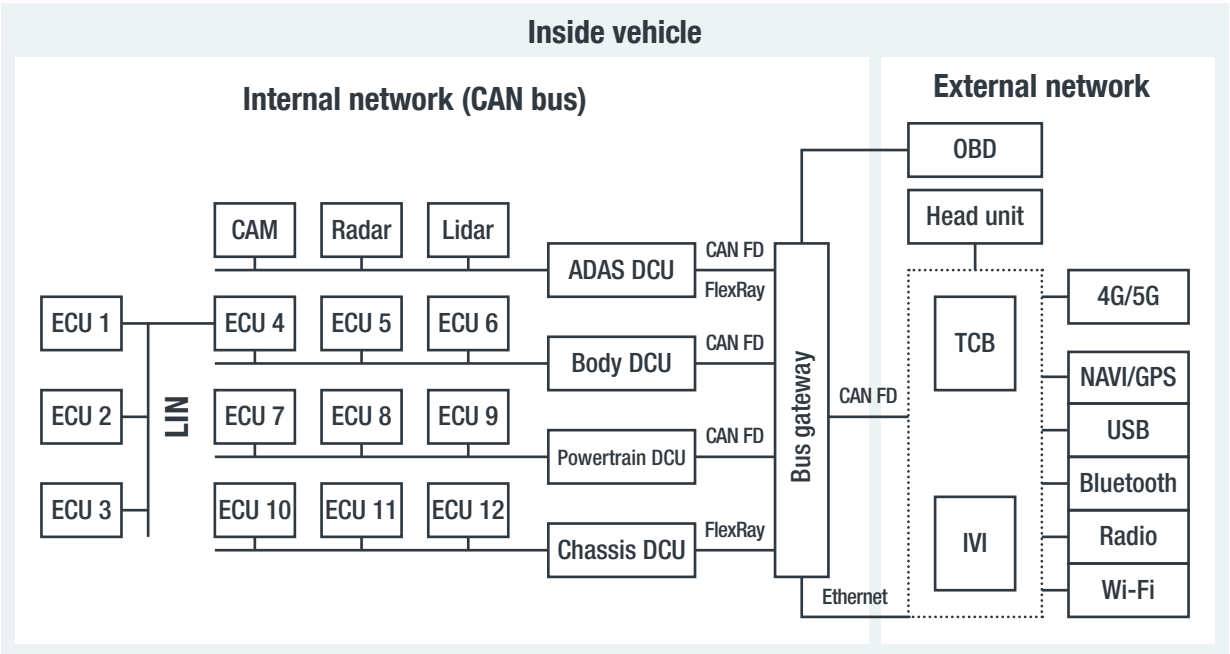


Figure 11. The connected car architecture inside a vehicle

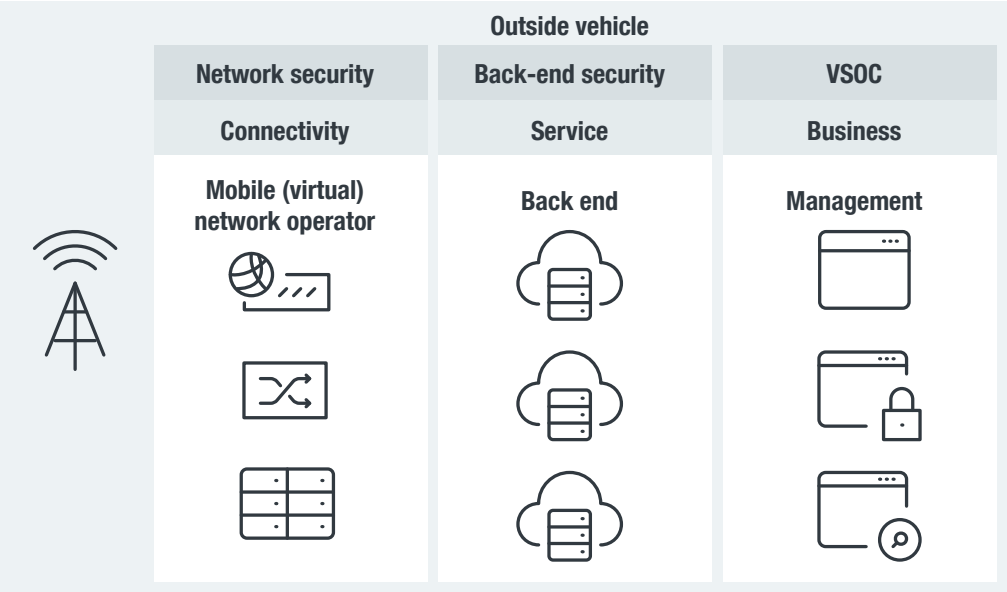


Figure 12. The connected car architecture outside a vehicle

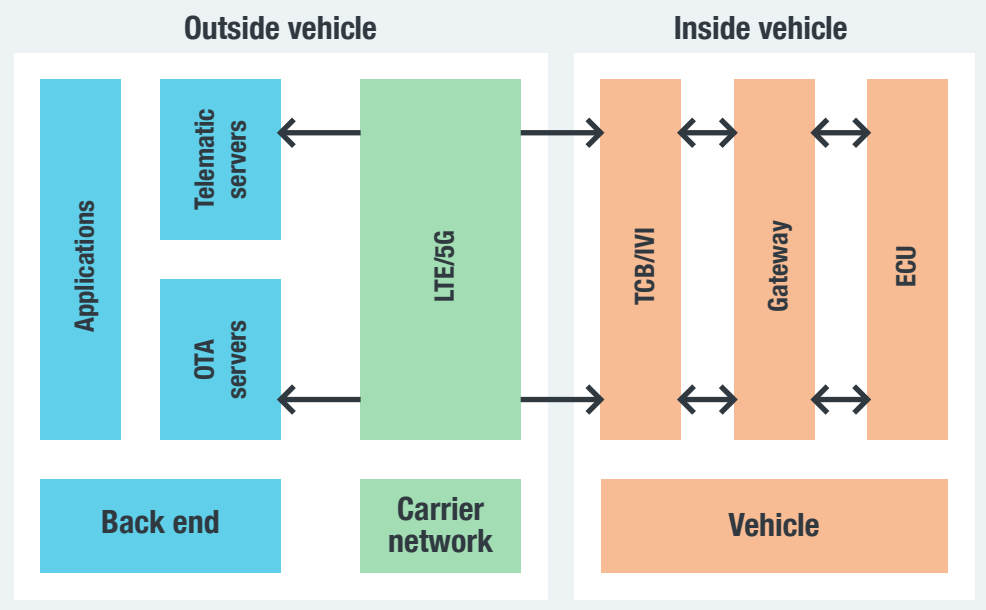


Figure 13. The combined connected car architecture from inside and outside a vehicle

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 itself:** This includes, as explained in the discussion of our generic car network architecture, the TCB, 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 mobile (virtual) network operator infrastructure:** This handles the millions of instances of connection setup and teardown, and data transmission.
- **The back-end ITS servers:** These run the applications, the services, and the databases accessed by the connected cars.
- **The VSOC:** This understands context by correlating notifications from the other 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. 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 tier 2 supplier? Will it be the fleet management operator? Will it be a SaaS 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 mobile (virtual) network operator that is managing the data pipelines and building cybersecurity into the cost of the data? Will it be a government department that implements a SaaS model that is similar to national defense but for connected cars and ITS infrastructures? As with IT security, the answer is all of the above. The end-to-end data supply chain has many owners, all of whom need to participate in providing holistic security solutions that work to protect the connected car ecosystem.

Solutions Technology Discussion

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:[66]

- **Network segmentation:** This refers to splitting a network into multiple subnetworks to reduce congestion, limit failures, and improve security. Putting all of the ITS controllers on a dedicated network that is separate from the corporate network reduces the risk of lateral movement and improves overall security.
- **Firewalls:** These are network security systems that control incoming and outgoing traffic based on an applied rule set. They monitor both ingress and egress traffic from unknown and bad domains, and identify applications or endpoints that generate or request bad traffic.
- **Next-generation firewalls (NGFWs)/Unified threat management (UTM) gateways:** These are network security products that unify multiple systems and services into a single engine or appliance. They can incorporate firewalls, intrusion prevention systems (IPSs), intrusion detection systems (IDSs), antivirus software, web filtering, application control, and other solutions all in the same appliance. These devices analyze network traffic at line speed. UTM gateways generally have lower traffic throughput than NGFWs.

- **Antivirus software:** This is a security solution that scans files to detect, block, and remove malicious software from the system. Antimalware uses heuristics and generic and specific signatures to detect known and unknown malware.
- **Antiphishing software:** These are email-filtering products that scan for and block incoming spam and phishing emails. Spear phishing is one of the top infection vectors. Some antiphishing solutions also use message sandboxes to screen for potentially malicious attachments.
- **Threat intelligence:** This is a collection of a wide variety of security data, including open-source intelligence, social media intelligence, deep web and dark web intelligence, technical intelligence, user endpoint feedback, indicators of compromise, vulnerability data, and malware data. These are combined with the expert analysis of security researchers to detect hidden threats and achieve faster response times against cyberattacks.
- **Breach detection systems (BDSs):** These are security solutions focused on detecting intrusions caused by targeted attacks and other sophisticated threats designed to harvest information from compromised systems. They analyze complex attacks out-of-band, detecting — rather than preventing — network breaches. They can analyze network traffic patterns across multiple protocols, identify malicious domains, and use emulation sandboxing to model the behavior and the impact of malicious files that are being dropped or downloaded.
- **IPSs and IDSs:** These are network security systems that examine traffic flow to detect and prevent network attacks. An IPS rejects the packet when a known bad event is identified. An IDS is a passive system that generates a report when a known bad event is identified. IPS/IDS monitors the entire network for suspicious traffic by analyzing protocols and doing deep packet inspection.
- **Encryption technologies:** These refer to software for the encryption and decryption of data in the form of files, email messages, or packets sent over a network. Encrypted network traffic will defeat MitM network sniffing data theft attacks.
- **Virtual or physical patch management:** This refers to software that keeps endpoints, servers, and remote computers updated by applying the latest security patches and software updates. Virtual patch management uses a security enforcement layer to prevent malicious traffic from reaching vulnerable systems.[67] In a large environment where patches need to be thoroughly tested before being applied, virtual patching provides the stopgap measure of filtering out malicious traffic attempting to exploit known vulnerabilities.
- **Vulnerability scanners:** These are automated tools that scan endpoints, servers, networks, and applications for security vulnerabilities that an attacker could exploit, and identify which among them need to be patched by the IT administrator. One of the tried and tested ways malware does lateral movement is by exploiting vulnerabilities in its target machine.

- **Security information and event management (SIEM):** This includes software products and services that provide real-time analysis of security alerts generated by network hardware, servers, endpoints, and applications. It is used for purposes such as data aggregation, rule-based/statistical data correlation, alerts, dashboards, compliance, log retention, and forensics. In a large network consisting of thousands of connected devices generating alerts, isolating the important alerts becomes increasingly difficult. SIEM provides tools to create rules that filter and highlight the important and relevant alerts. Log analysis helps determine whether there is a data breach in progress inside the corporate environment.
- **Two-factor authentication (2FA):** This is an authentication process in which the user needs to use two separate types of identification before being given access to company resources.[68]

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 Micro™ IoT Security for Automotive solution[69] 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 Virtual Network Function Suite™ (TMVNFS)[70] can be used to keep the network that connects the vehicle, the back-end cloud, and the data center secure. TMVNFS applies the appropriate network security protocols for connected cars to monitor traffic, detect threats, and block potentially risky and malicious connections from network scanners and compromised devices.

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[71] 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 indicators of compromise 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[72] 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.[73] It provides a comprehensive look at events alongside vital contextual data, thereby helping organizations identify and thwart threats.*

* As of July 2020, Trend Micro XDR is limited to certain Trend Micro products.

Connected Car Security in Motion

The number of partially automated or fully autonomous vehicles traversing the roads are expected to reach more than 14 million by 2025.[74] As connected cars continue to evolve and drive digital transformation as well as increase in number, there is a growing need to secure them from an expanding threat landscape.

In order to get a better understanding of how remote hacks could be launched on connected cars, we studied four landmark case studies. From those, we observed an emerging attack pattern used to compromise connected cars, specifically so that they would send malicious CAN messages to ECUs.

We also carried out a threat modeling exercise for connected cars. To highlight the many risks and threats to connected cars, we identified a number of possible attacks and performed a qualitative risk analysis using the DREAD threat model. Developers and manufacturers can use our connected car threat model to build more secure systems from the earliest development and production stages.

There is no doubt that there will be a duplication of defensive technologies across the many owners of the data supply chain for connected cars. This is a good thing for two reasons: First, data owners will buy security products from different vendors, and multiple vendors of the same product will increase the likelihood of catching malicious activity; and second, multilayered defenses will make it increasingly difficult for attackers to succeed. As we have pointed out, no defense is impregnable to determined adversaries. But what a multilayered approach does is it increases the cost, the time, and the resources needed by a malicious actor to mount a successful attack. Ideally, this will deter everyone — even the staunchest and most determined of attackers — from launching remote attacks on connected cars.

The technologies that drive mobility continue to evolve expeditiously. As cars become smarter and more connected, they continue their transformation into computers on wheels that collect and contain critical data — which are always valuable in the eyes of malicious actors. It is therefore important to ensure that cybersecurity is always ahead of the curve, and this can be done by using rich cybersecurity intelligence to protect connected cars from ever-advancing risks and threats. Developers and manufacturers must design security into the various connected technologies and have better visibility over the connected car ecosystem so as to keep connected cars running smoothly and securely.

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 HEREmobility. (n.d.). *HEREmobility.* "What Mobility As A Service Means For Consumers, Cities And Transport Providers." Accessed on July 8, 2020, at https://mobility.here.com/learn/smart-city-mobility/what-mobility-service-means-consumers-cities-and-transport-providers.

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/#5b23e5d2cb14.

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-you-think/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-internet-constellation-astronomy-coating.

13 Tesla. (n.d.). *Tesla.* "Support." 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 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.

17 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/.

18 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.

19 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/Chinese-Automaker-Plans-Satellite-Network-to-Support-Autonomous-Vehicles.aspx.

20 Circuit Basics. (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/.

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

22 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.

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 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.

25 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.

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

27 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/.

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

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

30 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.

31 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-wp.pdf.

32 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-wp.pdf.

33 WebKit. (n.d.). *WebKit.* "WebKit." Accessed on June 24, 2020, at https://webkit.org/.

34 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.

35 Ubuntu Wiki. (n.d.). *Ubuntu Wiki.* "AppArmor." Accessed on June 24, 2020, at https://wiki.ubuntu.com/AppArmor.

36 SSH. (n.d.). *SSH.* "SSH (Secure Shell)." Accessed on June 24, 2020, at https://www.ssh.com/ssh/.

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

38 Tech Terms. (n.d.). *Tech Terms.* "UDP." Accessed on June 24, 2020, at https://techterms.com/definition/udp.

39 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/.

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

41 Sen Nie, Ling Liu, Yuefeng Du, and Wenkai Zhang. (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.

42 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.

43 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/.

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

45 Tencent Keen Security Lab. (March 2019). *Tencent 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.

46 Tencent Keen Security Lab. (n.d.). *Tencent 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.

47 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.

48 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.

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

50 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.

51 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.

52 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.

53 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.

54 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-telematics-protocol?language=en_US.

55 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.us/books?id=WFp9AgAAQBAJ&pg=PT212&lpg=PT212&dq=Rex+RTOS&source=bl&ots=vhs1U8CEmO&sig=ACfU3U1-nWZjA8G0DE_eSxpJVfDQyK2nJA&hl=en&sa=X&ved=2ahUKEwicu7aS-pvqAhUVMd4KHYmVASsQ6AEwCnoECAwQAQ#v=onepage&q=Rex%20RTOS&f=false.

56 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.

57 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.

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

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

60 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).

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

62 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/.

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

64 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/.

65 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.

66 Nuuman Huq. (n.d.). *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.

67 Trend Micro Security News. (June 25, 2019). *Trend Micro Security News.* "Security 101: Virtual Patching." Accessed on June 26, 2020, at https://www.trendmicro.com/vinfo/us/security/news/security-technology/security-101-virtual-patching.

68 Trend Micro Security News. (Aug. 4, 2016). *Trend Micro Security News.* "How to Set Up 2FA: Layered Security for Online Accounts." Accessed on June 26, 2020, at https://www.trendmicro.com/vinfo/us/security/news/online-privacy/how-to-set-up-2fa-layered-security-for-online-accounts.

69 Trend Micro IoT Security. (n.d.). *Trend Micro IoT Security.* "Trend Micro IoT Security for Automotive: Accelerating critical-modules protection against cyberattacks on connected vehicles." Accessed on July 8, 2020, at https://www.trendmicro.com/us/iot-security/product/iot-security-for-auto?solutions=connected-car.

70 Trend Micro IoT Security. (n.d.). *Trend Micro IoT Security.* "Trend Micro Virtual Network Function Suite™: Security offering for communication service providers." Accessed on July 8, 2020, at https://www.trendmicro.com/us/iot-security/product/trend-micro-virtual-network-function-suite?solutions=connected-car.

71 Trend Micro. (n.d.). *Trend Micro.* "Hybrid Cloud Security: Cloud security simplified with Trend Micro Cloud One™ security services platform." Accessed on July 8, 2020, at https://www.trendmicro.com/en_us/business/products/hybrid-cloud.html#t2.

72 Trend Micro IoT Security. (n.d.). *Trend Micro IoT Security.* "Trend Micro IoT Security™: Secure endpoint SDK for IoT device makers." Last accessed on July 13, 2020, at https://www.trendmicro.com/us/iot-security/product/trend-micro-iot-security?solutions=smart-city.

73 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.

74 Nick Michell. (Dec. 5, 2016). *Cities Today.* "Self-driving cars to reach 14.5 million by 2025, says new study." Accessed on July 14, 2020, at https://cities-today.com/self-driving-car-production-reach-14-5-million-2025-says-new-study/#:~:text=New%20findings%20have%20revealed%20that,million%20consumer%20vehicles%20by%202025.

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

TREND MICRO™ | research