TU Automotive

REPORT EXTRACT: CYBERSECURITY IN THE CONNECTED VEHICLE

This exclusive report extract from the Cybersecurity for Connected Vehicles Report has been released in conjunction with the TU-Automotive Cybersecurity USA Conference & Exhibition (March 29-30, Novi, MI)

Find out more here: www.tu-auto.com/cyber-security

CHAPTER 3 TYPES OF HACKS AND THREATS THEY POSE

A growing issue is the large rise in keyless theft of vehicles, whether by bypassing any security mechanisms such as the immobiliser or by directly interacting with the vehicle itself (Hsu 2014). In itself, theft fuels organised crime whether through reselling the vehicles or stripping for parts but can also entail some potential economic harm or even danger to life.

It is clear that a significant factor in many of the attacks involve connectivity that enables advanced safety, comfort and convenience features. Connectivity integrated into the vehicle presents arguably the greatest danger: the ability to control a vehicle, or many vehicles *en masse* from a remote location.

Cyber security now has four main areas of concern:

1. **Vehicle theft.** Many tools encapsulating software attacks (for example, immobiliser overrides) are now available on the black market. Yet the biggest risk is the refined techniques now being used to carry out reverse engineering ECUs."
2. **Financial theft and damage.** The targets here are the individuals owning connected cars and the carmakers that is just attacking other large corporations. While these may have a very good grasp of IT security, there can be a disconnect between them and the engineering department developing the in-vehicle systems leading to a lack of awareness of the problem.
3. **Remote surveillance of individuals.** A telematics unit in a vehicle is capable of recording through the in-cabin microphone and exfiltrating data over a (remotely installed) IRC channel to gain location and any private information recorded inside the car.
4. **Attacks on infrastructure:** infrastructure is an increasingly popular target for terrorists, as a 2013 report from the IET points out. (The IET 2013).

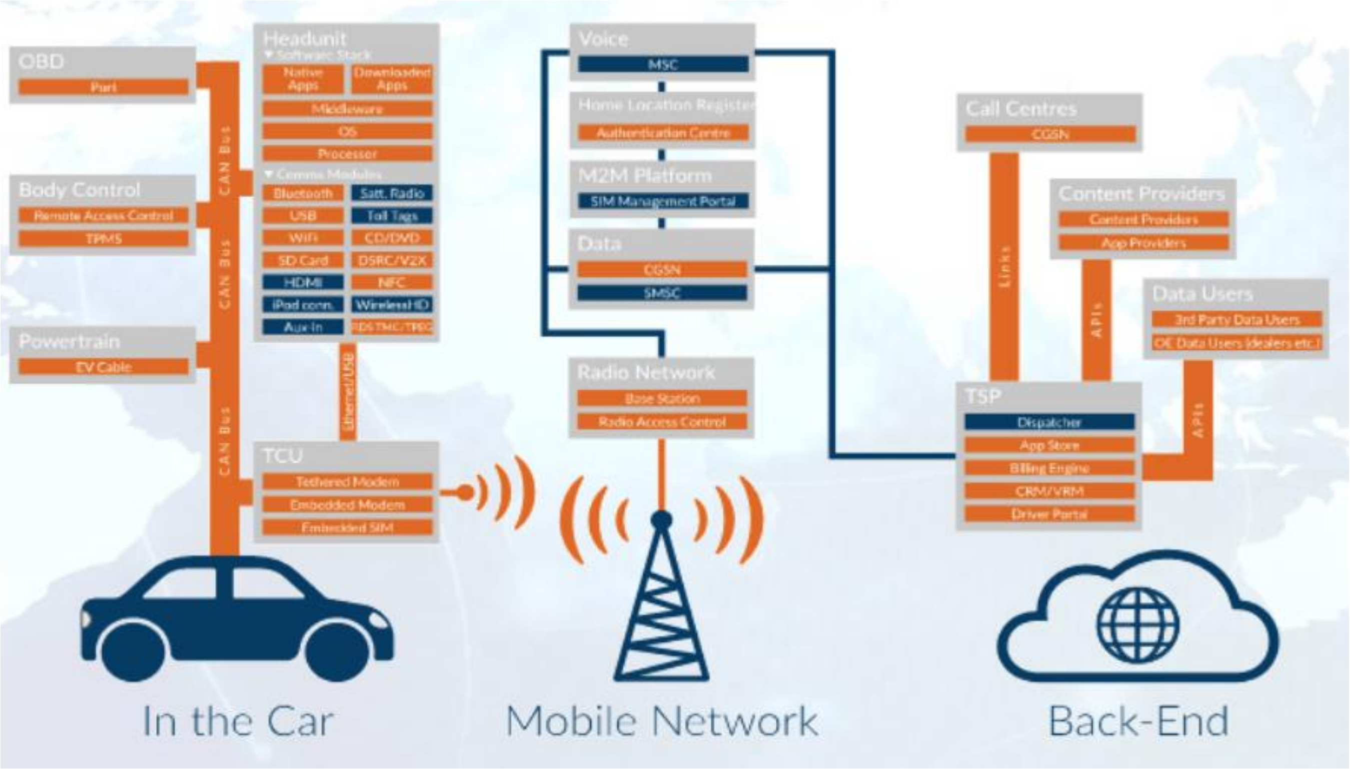

Figure 1: The back infrastructure. Each point represents a possible attack vector. Copyright Mike Parris.

3.2 The potential for remote hacks.

Attack anatomy: In (Miller & Valasek 2014) the authors look at remote attacks on automobiles. They see safety-critical attacks on automobiles as requiring different stages. The first is to remotely gain access to an internal automotive network. “This will allow the attacker to inject messages into the cars networks, directly or indirectly controlling the desired ECU.” Although one can imagine attacks in which simply exercising this control on the entry point ECU is sufficient, attacks that result in physical control of the vehicle will often require interaction with other ECUs. These “cyber-physical” attacks, move from the cyber domain to the physical one, and are potentially far more damaging.

It is likely that the first ECU is there to receive and process radio signals, and therefore does not have control of physical parts of the vehicle, and so a cyber-physical attack will usually require “a second step which involves injecting messages onto the internal automotive network in an attempt to communicate with safety critical ECUs, such as those responsible for steering, braking, and acceleration”. It may well be the case that the two ECUs (the entry point and the target ECU) are on different networks, and this second stage will involve the attacker needing to “bridge” the two networks in order to carry out the attack.

Bridging attacks. A vehicle will usually have multiple CAN Networks, each with its own resident ECUs. There will thus be a CAN for infotainment and a separate one for engine management functions, for example. Communication between these is restricted, and controlled by ECUs which are connected to both networks and act as “bridges” between them. These implement a set of rules to decide which messages should cross the CAN-CAN boundary. This rule set can be changed by injecting code to persistent alter the behaviour of one of these bridging ECUs. This could allow an ECU on a less-critical network to communicate directly with an ECU on a safety-critical network.

An Introduction to Attack Surfaces

NB- Here we have included a sample of attack surfaces mentioned, for full overview of the attack surfaces please refer to the complete report.

Interfaces

Infotainment

In (Checkoway et al. 2011) an attack is discussed using the CD player as an entry point. Two vulnerabilities were identified the first being “a latent update capability in the media player that will automatically recognise an ISO 9660-formatted CD with a particularly named file, present the user with a cryptic message and, if the user does not press the appropriate button, will then reflash the unit with the data contained therein”. They note that this is not a standard manufacturer method, and, therefore, speculate that it is a “vestigial capability in the supplier’s code base.” The second followed from the first. Given that the media player can parse complex files, they reverse-engineered a substantial amount of the media player firmware and through examination of this they were able to construct a buffer overflow attack. Finally, they were able to modify a WMA audio file so that, when a CD containing the file was played on the system, it sent carefully chosen CAN packets to the network.

OBD-II

The On-board diagnostics (OBD) connector offers direct access to all CAN buses through a physical port in the cabin of a vehicle. The fact that the interface and messages are standardised means that there is a plethora of cheap, easily available scan-tools for the OBD port. These scan-tools come in two types: full-featured versions with in-built software, user-interfaces and so on, and dumb tools that must interface with another computing platform such as a phone or conventional PC. At the 2015 Black Hat Asia security conference in Singapore, a programmable device called the CANtact was shown. When available, it will sell for less than $100, and form a physical connection between a vehicle’s OBD port and a computer’s USB port. The device is run on open-source software, and the author has also developed a “Python library designed to make it easy to interact with CAN networks.” (Evenchick 2015). CAN frames can be encoded easily as Python objects and sent, received, logged and inspected. Among others, the OBD II and UDS protocols are supported. Supporting UDS gives the ability to read and write arbitrary memory in the vehicle. Although this device promises to make the hacking of cars much easier, it requires physical access to the OBD and is, therefore, chiefly of interest to vehicle “tuners”.

(Checkoway et al. 2011) notes that “in 2004 the Environmental Protection Agency mandated that all new cars in the US support the SAE J2534 “PassThru” standard – a Windows API that provides a standard, programmatic interface to communicate with a car’s internal buses.” Typically implemented as a Windows DLL, this communicates over a wired or wireless network with a reprogramming or diagnostic tool. They chose the most commonly used device and identified two vulnerabilities. First, if the PassThru device is connected to a car, an attacker on the same WiFi network can connect to it and, obtain control over the car's reprogramming. Secondly, the PassThru device itself can be compromised, and malicious code injected. They were even able to write a worm that automated the attack, spreading itself from device to device.

Bluetooth

(Haataja 2009) presents a through overview of the security architecture and security modes of the Bluetooth protocol before listing the vulnerabilities that Bluetooth networks face. He divides these into three categories, corresponding to the CIA model of security: threat of disclosure of unauthorised information, treat to integrity of information and threat of denial of service. He also notes that "Powerful directional antennas can be used to considerably increase the scanning, eavesdropping and attacking range of almost any kind of Bluetooth attack."

(Miller & Valasek 2014) consider "Bluetooth to be one of the biggest and most viable attack surfaces on the modern automobile, owing to the complexity of the protocol and underlying data. Additionally, Bluetooth has become ubiquitous within the automotive spectrum, giving attackers a very reliable entry point to test."

(Checkoway et al. 2011) also investigated the Bluetooth capabilities built in to their test vehicle's telematics unit. They were able (through reverse-engineering) to "gain access to the telematics ECU's Unix-like operating system and identified the particular program responsible for handling Bluetooth functionality." They were able to verify that it contained "a copy of a popular embedded implementation of the Bluetooth protocol stack and a sample hands-free application" together with a custom-built interface. The interface contained a vulnerability that allowed a buffer overflow attack to be mounted by any paired Bluetooth device, allowing arbitrary code to be executed on the telematics unit.

Internal hardware and software

CAN-BUS

(Hoppe et al. 2011) discuss attacks using the CAN bus, aimed at disrupting the control systems of the car in four areas: the window lift, the warning light, the airbag control system and the central gateway. The initial part of the window lift attack was carried out in a simulation environment, using CANoe, a product from Vector Informatik. In this test, "a few lines of malicious code" were added to "an arbitrary ECU in the simulated comfort CAN subnetwork." This code was deployed when the vehicle speed rose over 200km/h. The window opens and will not close until the end of the attack. Similar results were demonstrated in a corresponding physical environment. In the second scenario targets the warning lights. These are meant to flash in the event of an unauthorised opening of a door but the authors were able to turn them off and ensure they stayed off just by sending CAN commands on the comfort subnetwork. They were able to emulate the behaviour of a fully functional airbag system, including a successful start-up check. This code could be included on the network if the airbag system was broken or had been removed. The final attack was on the gateway

ECU, which “implements basic filtering functions with respect to the internal communication of the car” and forces a degree of separation between internal and external networks. Here, “an implementation flaw in the implementation of gateway ECU could be identified and exploited” inducing the gateway ECU to pass on arbitrary internal CAN messages to the outside.

It should be noted that in each of these cases apart from the last one, the attackers required physical access to the internal CAN network and the ability to insert malicious code in ECUs. The last one required the ability to insert malicious code at the OBD interface. Each of the attacks is analysed using the CERT taxonomy, and a set of short-term countermeasures are suggested. These include intrusion detection and facilitating post-incident analysis through proactive forensics support.

External devices

Telematics: manufacturer and after-market telematics

Manufacturer-provided telematics devices have been shown to have flaws in the scientific literature, notably in (Koscher et al. 2010; Checkoway et al. 2011; WIRED 2015) These, however, have been relatively easy for manufacturers to fix. In terms of security threats, the real threat is aftermarket telematics devices.

Although most TCUs are designed for monitoring, they are often equipped with have the ability to send packets to the CAN bus. As (Foster et al. 2015) show, “transmit access to the CAN bus is frequently sufficient to obtain arbitrary control over all key vehicular systems (including throttle and brakes).” (Foster et al. 2015) go on to carry out a comprehensive analysis of the attack surface of a TCU manufactured by Mobile Devices Ingenierie, separating their analysis into local and remote threat models. In the local threat model they “evaluate how an attacker may be able to attack the TCU directly in an effort to gain control over the device (e.g., for compromising the device after intercepting it during shipping or after obtaining brief physical access to the vehicle such as a valet might have).” In the remote model they “assume that the attacker does not have physical access to the TCU but the TCU is installed in an automobile of a victim and the attacker’s goal is to compromise the vehicle.” Locally the TCU included a mini-USB connector for debugging purposes, which allowed for software updates to be installed. The hardware itself was also insecure: several test points were identified at which a would-be attacker could alter the contents of the chip. The remote threat model in which the authors assume that “the adversary has no physical access to the device or vehicle and may not even know where they are located geographically” is a 2G or 3G modem (depending on the version) that provides connectivity over cellular networks. The device itself uses both SMS and IP data communications for a variety of functions, and “both interfaces will respond directly to requests and thus can be remotely identified if the devices can be remotely addressed.” (Foster et al. 2015).

Locally, a connection to the USB port was sufficient to identify the subnet and IP address, and then to identify that standard ports were used for web/telnet console and SSH access. They were able to expose sufficient commands in this way to change all the software configuration variables. They were also able to remove the memory and extract the contents using a data reader. In this way they were able to extract a variety of public and private keys and certificates. They were also able to identify the private key for the root user. Giving them “the ability to authenticate to the device over

SSH and directly obtain a root shell", then "read and write any file, execute arbitrary commands and download and install additional software to create arbitrary functionality." (Foster et al. 2015). Furthermore, the same key worked in all versions of the device tested. This was what allowed them to carry out the remote exploits they go on to discuss.

The authors found that "the web, telnet console, and SSH servers were bound to all of the network interfaces" and they were thus able to attempt to compromise the TCU using each of these vectors. (Foster et al. 2015) Internet-based attacks were foiled by a quirk of the carrier used by the TCUs but this, they believe, was entirely a property of the carrier, and not the TCU itself. SMS-based attacks were made possible owing in part to the discovery of online documentation for an SMS administration interface. This included commands for, among others, initiating remote updates. The full update procedure was determined from looking at the logs created from initiating an update via SMS.

The next step in the attack was to investigate the capabilities of the TCU and, here, the authors discovered it possible to send and receive arbitrary CAN packets, making arbitrary attacks possible.

(Miller & Valasek 2014) describe telematics units to be "the holy grail of automotive attacks since the range is quite broad (i.e. as long as the car can have cellular communications). Even though a telematics unit may not reside directly on the CAN bus, it does have the ability to remotely transfer data/voice, via the microphone, to another location."

Passive Keyless Entry and Start

Relay attack In (Francillon et al. 2011), the authors demonstrate relay attacks on Passive Keyless Entry and Start (PKES) systems. They build two attack realisations using both wired and wireless physical-layer relays that "allow the attacker to enter and start a car by relaying messages between the car and the smart key." Worryingly for manufacturers, the operation of these relays is completely independent of any protocol, or presence of strong authentication and encryption. They are able to carry out the attack when relaying the signal from the car to the key only. (Francillon et al. 2011)

Remote Key

When restricting their attraction to attacks that all remote code execution on a vehicle, (Miller & Valasek 2015) draw the conclusion that this is relatively difficult to do through a remote key: "the attack surface of a remote keyless entry system is quite small." (Miller & Valasek 2014) Considering the specific example of a Toyota Prius Smart Key, they conclude that it "must have some firmware to handle reading RF signals, encryption/decryption code, some logic to identify data from the key fob, and to be programmed for additional/replacement key fobs. While this is a possible avenue of remote code execution, the attack surface is quite small." (Miller & Valasek 2014). Never-the-less, they note that the remote key ECUs are of necessity on the same network segment as cyber-physical ECUs, and so offer a simpler attack than one that involves bridging.

References

Checkoway, S. et al., 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. *System*, pp.6–6. Available at: http://www.usenix.org/events/security/tech/full_papers/Checkoway.pdf.

Evenchick, E., 2015. An Introduction to the CANard Toolkit. In *Black Hat Asia*. pp. 1–10.

Foster, I. et al., 2015. Fast and Vulnerable : A Story of Telematic Failures. In *Proceedings of the USENIX Workshop On Offensive Technologies (WOOT),*.

Francillon, A., Danev, B. & Capkun, S., 2011. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. *Network and Distributed System Security Symposium*, pp.431–439. Available at: http://eprint.iacr.org/2010/332.

Haataja, K., 2009. *Security Threats and Countermeasures in Bluetooth-Enabled Systems*.

Hoppe, T., Kiltz, S. & Dittmann, J., 2011. Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures. *Reliability Engineering & System Safety*, 96(1), pp.11–25. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0951832010001602.

Hsu, J., 2014. Car Thieves Use Handheld Electronics to Steal Keyless Cars. *IEEE Spectrum*. Available at: http://spectrum.ieee.org/cars-that-think/transportation/advanced-cars/car-thieves-use-handheld-electronics-to-steal-keyless-cars [Accessed October 19, 2015].

Koscher, K. et al., 2010. Experimental security analysis of a modern automobile. *Proceedings - IEEE Symposium on Security and Privacy*, pp.447–462.

Miller, C. & Valasek, C., 2014. *A Survey of Remote Automotive Attack Surfaces*, Available at: http://illmatics.com/remote attack surfaces.pdf.

Miller, C. & Valasek, C., 2015. Remote Exploitation of an Unaltered Passenger Vehicle. , 2015, pp.1–91.

The IET, 2013. *Security and Risk in Transport Systems and Infrastructure*,

WIRED, 2015. Hackers Remotely Kill a Jeep on the Highway—With Me in It | WIRED. *WIRED*. Available at: http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway [Accessed October 14, 2015].

This exclusive report extract from the Cybersecurity for Connected Vehicles Report has been released in conjunction with the TU-Automotive Cybersecurity USA Conference & Exhibition (March 29-30, Novi, MI)

Find out more here: www.tu-auto.com/cyber-security