RESEARCH SUMMARY

Intel IoT
Intel Transportation Solutions Division
Wind River Systems

Intel Automotive Security Research Workshops

Summary of Findings

> "Most of the smart minds—many in this room—may know security or cars, but not both."

TOP 7 vulnerabilities of in-vehicle infotainment systems

The Research Workshops

Over the past year, we've seen several high-profile automotive compromises whereby security researchers were able to attack safety-critical systems using external network interfaces as an entry point.[1] This led Intel to become interested in using a hands-on approach to identifying vulnerabilities that could be found in in-vehicle infotainment (IVI) systems.

Intel hosted two automotive security research workshops, the first of which took place January 12–14 in San Diego, California, and the second February 2–4 in Barcelona, Spain. Individuals from around the globe were invited and given the opportunity to work hands-on on an Intel® Linux*-based IVI simulation platform, with the goals of advancing knowledge of threat areas, identifying vulnerabilities and, more importantly, potential mitigation strategies, as well as identifying topics for future research. Protecting human lives and increasing automotive safety is a collective social responsibility that Intel takes seriously.

This summary of findings can be used to help understand the security implications of current Linux-based IVI designs.

Participants were recruited from universities, consulting firms, technology, and automotive manufacturers.[2] Teams were formed to evenly distribute technical competency and varied software and hardware expertise between groups. Researchers had access to an IVI simulation platform for reference that was representative of a configuration that could be used in an automobile. It contained hardware components typical of IVI systems (Wi-Fi, audio, TFT display, CAN bus, etc.) and software (a Linux-based OS, device drivers, communication stack, etc.). Participants were also given access to design documentation, binaries, and some source code. They were allowed to use any tools they wanted.

Why cars?

Cars are ubiquitous—and increasingly technically complex. Competitive features in automotive now overlap with consumer electronic products, largely driven by consumer expectations for convenience and interoperability. Features such as in-car Wi-Fi access points, the ability to play MP3s, and touch screens barely existed a decade ago. These features come with other complexities: more advanced processors, feature-rich operating systems, consumer device interoperability, and software applications.

Hackers are often interested in attacking commonly used products; the motivations can range from increasing security in an altruistic sense, personal or professional recognition, damage to established global brands, or financial rewards. The value of compromising safety-critical systems adds other nefarious motivations that may be desired by enemy nation states or organized criminals.[3]

What's behind the increase in car hacks?

The similarity to existing consumer electronics means hackers have access to years of prior hackers' experience and exploits.

There has also been an explosion of network features that can be considered as new entry points: Wi-Fi, 3G/LTE, Bluetooth*, DAB, CAN, wireless entry, and even TPMS.

The software stacks required to drive this comprise a large attack surface, such as operating systems and applications. With this complexity comes the requirement to conduct in-field updates, which is another entry point.

Threat observations

The following is a summary of observations and developer considerations that were found during the workshops.

1. Car systems integrity

A common approach to breaching security is to replace trusted software components with malware that can be used to affect the confidentiality, integrity, and availability of the system. This can be done either by a persistent change (such as installing new binaries) or dynamically (with in-memory changes perhaps). This has been seen in phone jailbreaking and other consumer electronics.

The impact is significant. This approach could be used to attack other locally connected systems. For instance, a compromise of the IVI stack could be used to attack a CAN bridge—or tamper with the input going to the CAN bridge. Or, it can infect other connected components.

There are several mitigation strategies:

- Secure boot mechanisms can prevent unsigned software from being loaded.
- Userspace binaries can be verified before load using Integrity Measurement Architecture (IMA).
- Keys used for integrity measurement can be stored behind barriers, such as secure enclaves or hardware security modules (HSM).

There are limitations to these approaches. Simply verifying the signature of the code does not mean the code doesn't have other vulnerabilities (such as buffer overflows or network misconfiguration).

A proper design will balance the boot time requirements of IVI systems with the time required for verification before code is loaded.

2. Running software on the CAN gateway

Very often, one can find an ECU in the vehicle which serves as a gateway between the IVI system and the vehicular network that safety-critical ECUs—such as brakes, steering wheel, gas pedal, etc.— are connected to. The gateway ECU is responsible for passing back and forth the CAN messages from the IVI system to the other ECUs on the network. Examples of these interactions are climate control of the vehicle, seat position, and seat heating.

AUTOMOTIVE SECURITY REVIEW BOARD (ASRB)

Building on interest received since its fall of 2015 introduction, Intel is collaborating with industry leaders to form the Automotive Security Review Board (ASRB), an open and independent global research organization. The ASRB will consist of a board of directors and a Technical Steering Committee (TSC) to cooperatively work toward a shared vision of self-healing vehicles. Intel's Automotive Security Research Workshops served as a pilot for potential future workshops to be conducted as part of the ASRB, and the output from these workshops will help to seed future ASRB research topics.

Other possible approaches are to attempt to inject arbitrary CAN messages to the CAN gateway in an effort to communicate with other ECUs on the network, or even to use various reverse-engineering techniques to disassemble CAN gateway firmware and use update mechanisms to replace the original firmware with a tampered version. This can potentially allow full access to the gateway to send arbitrary CAN messages.

Even when unable to replace safety-critical ECU firmware, the compromise of safety-critical vehicle features can be accomplished by sending denial of service (DoS), diagnostic messages, or sniffing the CAN bus to understand the CAN message format that safety ECUs send or receive.

The following mitigation strategies can be used:

- Implementation of a secure protocol between the IVI system and CAN gateway to make it difficult for an attacker to reverse engineer the communication path.
- Signed CAN firmware update, together with secure boot and deep package inspection.

3. External media

Consumer electronics today are a vital part of the automotive cabin environment. Drivers and passengers can bring various devices to the car and increasingly expect seamless integration into the car multimedia environment. Good examples of this media experience are audio over Bluetooth, Apple CarPlay*, Google Android Auto* solutions, UPnP, and similar technologies. Even old-fashioned USB sticks with MP3 files remain popular.

3G/LTE, Wi-Fi, Bluetooth, NFC, USB, SDCard, and BlueRay—this is just a partial list of external interfaces present in the modern vehicle. Various media services connected to these interfaces can be used to gain access to the IVI system. It can be the first step in a multilevel compromise of the safety ECUs or can enable retrieval of various sensitive user data, vehicle location, etc. Specially prepared media files can be used to tamper media engine services, Bluetooth, and Wi-Fi stacks. An attacker could try to use software update mechanisms via USB stick to run shell scripts or install unauthorized software.

Techniques for mitigation of external media risks:

- All wireless and wired external interfaces need to be properly configured.
- To decrease the level of exposure, all unused Bluetooth profiles should be disabled.
- Only supported file systems should be enabled for mounting by a USB stick, and proper permissions (e.g., read-only, noexec, nodev) should be used.
- Only supported USB device classes should be enabled—e.g., USB MSD (USB stick, USB HID) for iPhone*, USB MTP for Android*, etc.
- For software updates, it is recommended that signed binaries be used, together with an authenticated software update procedure.

4. Use of compromised applications as an entry point

Many automotive OEMs allow users to install applications from dedicated app stores to the IVI system. It can be generic media, browser, or social network applications—such as Google Chrome*, Spotify*, and Facebook*, as well as special OEM applications for car maintenance or fun. HTML5, Android, or proprietary application formats are current alternatives to enrich the native HMI in the IVI system.

However, it offers a range of opportunities to compromise the IVI system. These can include browsing the local file system to determine content, cross-scripting, or any kind of malware that one can find on mobile phones today. These attacks can be considered a first-stage attack to penetrate the IVI system for further exploration, denial of service, or privacy-related attacks.

As mitigation, the following techniques can be used:

- Isolate applications into containers.
- Split the IVI system into different security domains and apply strict role-based access control (RBAC) rules to the application domain.

5. Installation and updating installed software

A complex system must be updatable to allow post-deployment bug fixes, feature enhancements, or security updates.

However, a software update mechanism presents significant exposure. It must run in a privileged mode, since it must update privileged software. It must also be accessible over an entry point that is already susceptible to attack. Often updates are distributed on a USB mass storage device, or through over-the-air (OTA) updates over 3G/LTE.

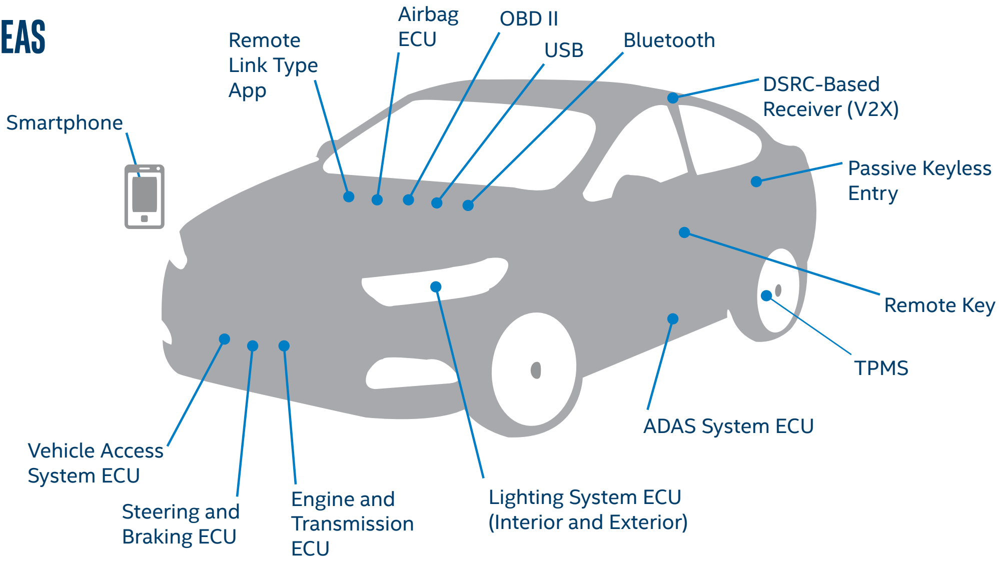
Making downloadable updates available online also allows the ability to conduct reverse engineering. This can help with locating flaws in the update mechanism or with hacking other parts of the system off-line.

Common vulnerabilities include exploiting bugs in package parsing, which can cause arbitrary code execution, or allowing tampered package installation because of compromised encryption methods.

There are a number of ways to prevent the installation of persistent malware to software updates:

- Always sign or encrypt update packages.
- Store verification keys securely on the device, and test the verification process.
- Install a read-only, small-footprint rescue image to fall back on in case of an update failure.
- Protect all boot images with a secure boot mechanism so only properly signed kernels can be loaded.

15 DISTINCT HACKABLE AREAS



6. Configuration of wired and wireless networks

Network interfaces such as Wi-Fi, Bluetooth, Ethernet, and USB are often readily-accessible in vehicles. Ethernet is sometimes used to connect multiple IVI systems and exterior cameras. Each of these interfaces exposes the system to different levels of potential attacks:

- Physical access to USB ports, used for access to audio or video files
- Adjacent access to short-range interfaces such as Wi-Fi or Bluetooth
- Remote threat opportunities of long-range wireless technology such as 3G or LTE, including key fobs

The full range of exploits may be possible on each of these interfaces, such as:

- Denial of service—by stressing an exposed, unprotected interface
- Compromise of configuration—by taking ownership of the interface and changing its configuration
- Penetration of in-vehicle networks—reaching critical components by abusing IVI routing capabilities

Compromise of configuration and penetration of in-vehicle networks could both be used as attack vectors to other parts of the system, such as the CAN buses.

Mitigation techniques to prevent the system from remote attacks include:

- Route only required services to necessary interfaces. For instance, debug services shouldn't be bound to external ports and over-the-air updates.
- Conduct a systematic verification process on all interfaces. Spying the traffic to make sure no unwanted data is leaked at every stage of the system lifecycle will help to prevent misconfigurations in the field.

7. Known vulnerabilities through open source software components

IVI implementations often use open source software to meet the requirements of complex operating system and multimedia applications. These components are frequently used throughout the system, from device drivers, system libraries, and multimedia applications. Open source software vulnerabilities are frequently publicized, which has serious security implications.

Not all components' vulnerabilities are relevant to an IVI system, but each should be considered. Automotive vendors should have an incident response process that allows for discovery of reported vulnerabilities, triage, resolution, and communication.

An update mechanism should be used that allows deployment of security updates, to reduce the applicability of a given compromise. To minimize risk from the start, only the necessary open source ingredients should be used in the software stack, removing potentially vulnerable packages that add no value or utility to the vehicle.

Conclusion

This paper describes some of the IVI potential threat areas, and identifies vulnerabilities and possible mitigation strategies, as well as potential topics for future automotive security research. It is by no means exhaustive, but these are technology areas that were found to be at risk of compromise by third-party security researchers during an organized hands-on workshop. This may point other bodies into areas of further research.

We sincerely thank our workshop participants for coming together to share their knowledge and expertise. Their invaluable contribution helps identify and understand threat areas to further improve the security of future automotive products. Intel is committed to continuing to collaborate with the automotive security industry and assist with the development of best practices and considerations for development of safe and secure automotive products. We look forward to future opportunities with these experts, to evolve the work through ongoing research and analysis.

PARTICIPANTS: SAN DIEGO WORKSHOP

Cameron Beyer	Ford
Chad Dewey	Saginaw Valley State University
Steven Drewes	SpaceX/RIIS
Shane Fry	Star Lab
Helena Handschuh	Rambus
Hunter James	RIIS
Jonathan Kline	Star Lab
Karl Koscher	UCSD
Joseph Maes	SVSU
Mārtiņš Možeiko	LG
Harsh Patil	LG
Chris Poulin	IBM
Guangzhi Qu	Oakland University
Michael Roof	SVSU
Spenser Solys	RIIS
Armin Wasicek	UC Berkeley
Rob Wood	NCC Group

PARTICIPANTS: BARCELONA WORKSHOP

Néstor Álvarez Díaz	University of La Laguna
Senad Aruc	UL
Daniele Bonomi	Security Researcher
Davide Cioccia	UL
David Clare	NCC Group
Radek Domanski	Security Researcher
Jako Fritz	UL
Florian Gaultier	SCRT
Jos Heijmans	UL
Ryan Hileman	Security Researcher
Neil Jones	NCC Group
Sebastian Koszyk	Secure Net World
Denis Legezo	Kaspersky
Francisco Martín Fernández	University of La Laguna
Daniel Mayer	NCC Group
Jan Nordholz	Technical University Berlin
Nicolas Oberli	Security Researcher
Alexandra Rivero García	University of La Laguna
Iván Santos González	University of La Laguna
Jean-Pierre Seifert	Professor at Technical University Berlin
Adrien Stoffel	SCRT
Paul Wooderson	Horiba-Mira

[1] Presentation of the FCA Jeep attack at Black Hat, August 2015.

[2] Particants appendix.

[3] Stuxnet.

0516/MH/CMD/PDF 334430-001US