Defending Connected Vehicles Against Malware: Challenges and a Solution Framework

Tao Zhang, *Fellow*, *IEEE*, Helder Antunes, and Siddhartha Aggarwal

(Invited Paper)

***Abstract*—Vehicles face growing security threats as they become increasingly connected with the external world. Hackers, researchers, and car hobbyists have compromised security keys used by the electronic control units (ECUs) on vehicles, modified ECU software, and hacked wireless transmissions from vehicle key fobs and tire monitoring sensors, using low-cost commercially available tools. However, the most damaging security threats to vehicles are only emerging. One such threat is malware, which can infect vehicles in a variety of ways and cause severe consequences. Defending vehicles against malware attacks must address many unique challenges that have not been well addressed in other types of networks. This paper identifies those vehicle-specific challenges, discusses existing solutions and their limitations, and presents a cloud-assisted vehicle malware defense framework that can address these challenges.**

***Index Terms*—Antivirus, connected cars, connected vehicles, intelligent transportation system (ITS) malware defense, security, threat defense, vehicle communications, vehicle networks, virus.**

I. Introduction

A GROWING range of vehicle security risks has been revealed recently. With physical access to a vehicle, researchers, car hobbyists, and hackers have been able to use low-cost and commercially available tools to send bogus messages over in-vehicle networks to the electronic control units (ECUs), read and modify ECU software, read ECU memory, and compromise ECU security keys [5], and control a wide range of vehicle functions at ease. When a vehicle does not communicate with the external world, attackers need physical access to the vehicle to exploit its security vulnerabilities and a successful attack impacts only the vehicle; the attacker can access physically.

Today, almost every new vehicle in production in the United States uses embedded short-range radios to support wireless key fobs for keyless entry. Security keys used by vehicle key fobs have been hacked [1]. Researchers have also been able to relay the radio signals from a key fob to the vehicle without compromising its security keys [2]. This can be done even when the key fobs are far away from the vehicle (e.g., when the vehicle is in a parking lot and the driver is inside a shopping mall) and using commercially available low-cost devices. It allows attackers to easily unlock doors to steal or burglarize the vehicle.

Manuscript received October 29, 2013; revised December 19, 2013; accepted January 09, 2014. Date of publication January 23, 2014; date of current version May 05, 2014.

T. Zhang is with Cisco Systems, Inc., Fort Lee, NJ 07024 USA (e-mail: tazhang2@cisco.com).

H. Antunes and S. Aggarwal are with Cisco Systems, Inc., San Jose, CA 95134 USA (e-mail: helder@cisco.com; saggarwa@cisco.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JIOT.2014.2302386

Researchers and hobbyists have been able to compromise tire pressure monitoring systems (TPMS), which are implemented on many vehicles today, to set false "low tire pressure" warnings [3], which can fool a driver into believing she actually has a flat tire while traveling at high speeds. Once again, this can be accomplished easily and using low cost and widely available tools.

The most damaging security threats, however, are only emerging as vehicles connect to the Internet, provide onboard Wi-Fi hotspot services, communicate with other vehicles and Intelligent Transportation System (ITS) infrastructures, and support advanced applications such as over-the-air (OTA) ECU firmware update [4], [6], [32], [33]. Many attacks, which used to be only feasible with physical access to a vehicle, can now be carried out remotely over wireless networks. This allows attackers to compromise more vehicles with relative ease. A compromised vehicle can also be used to attack other vehicles.

One critical threat to connected vehicles is malware—malicious software programs designed to disrupt computer operations or gain unauthorized access to information. Malware can infect vehicles through multiple venues such as Internet connectivity, wireless communication with roadside networks, Wi-Fi hotspots on vehicles, malware-infected consumer electronic devices (e.g., smartphones, laptop computers, memory sticks) wirelessly or physically connected to the vehicle, removable media connected to the vehicle, and files exchanged among vehicles. Known vulnerabilities in the design and implementation of onboard communication systems, hardware, software, and applications [6], [33], [37] can be exploited by malware to infect a vehicle. Once on a vehicle, malware can cause a wide range of disruptions and damages [4]–[6], [32], [33]. Examples include 1) disrupting the normal operation of vehicle features such as locking the in-car radio so the users cannot turn it on, 2) abusing vehicle features to cause driver distractions such as arbitrarily turning on the in-car audio and tuning its volume, 3) locking vehicle features to demand a ransom, 4) deleting or modifying files on the vehicle and on users' brought-in devices connected to the vehicle, 5) consuming memory space and CPU cycles, 6) stealing private data, 7) disabling vehicle safety functions, and 8) using a compromised vehicle to send bogus data to others.

This paper addresses the issue of how to defend vehicles against malware attacks. Protecting vehicles against malware requires addressing unique challenges that have not been well addressed in other types of networks. This paper identifies and discusses these challenges and presents a cloud-assisted vehicle malware protection framework that can address them.

The paper is organized as follows. Section II provides a brief overview of existing in-vehicle networks. Section III

summarizes the main types of malware and how they can infect computer systems. Section IV discusses how malware can infect connected vehicles. Section V summarizes main existing approaches used to protect vehicles and outlines their limitations. Section VI discusses the unique challenges in defending connected vehicles against malware. Section VII describes existing malware detection techniques and their limitations when applied to vehicles. Section VIII presents a cloud-based malware defense framework for connected vehicles. Section IX discusses selected remaining issues. Section X contains the conclusion remarks.

II. In-Vehicle Networks and Connected Vehicles

This section schematically illustrates a typical architecture of existing in-vehicle networks and discusses its characteristics that directly impact vehicle threat defense.

As illustrated in Fig. 1, a vehicle has multiple electronic subsystems, such as powertrain control, body and comfort control, In-Vehicle Infotainment (IVI), vehicle safety, and embedded telematics subsystems. Each subsystem has a set of ECUs. Examples include ECUs for controlling airbags, antilock braking system (ABS), and advanced driver assistance system (ADAS). A modern vehicle has tens to over a hundred ECUs and some vehicles already have over a hundred million lines of software code [32]. Many ECUs, however, have strictly limited CPU and memory resources due to stringent cost constraints.

Different electronic subsystems traditionally use physically separate networks of various types that are specially designed for vehicles. For example, high-speed Control Area Network (CAN) buses are used for time-critical engine control, powertrain and safety subsystems while Local Interconnect Network (LIN) is used for less time-sensitive body control subsystems. Media-Oriented Systems Transport (MOST) and Ethernet are used in IVI subsystems to support audio and video applications and onboard video cameras.

Different in-vehicle networks are interconnected with gateways that control which messages can pass from one network to another. The gateways themselves have usually been interconnected with high-speed CAN buses, which will likely be replaced with Ethernet in the near future [34].

Today, each electronic subsystem typically implements its own dedicated communication modules to connect with the outside world. For example,

1) Users' brought-in devices are traditionally tethered wirelessly (e.g., via Bluetooth) or physically (e.g., via USB) to the IVI system. Tethered smartphones, e.g., have been used to support infotainment and safety applications such as hands-free calling, emergency calls, and to allow smartphone applications to use the vehicle's sound and display systems.
2) The embedded cellular module is usually part of a telematics system separate from the IVI system.
3) Onboard Wi-Fi hotspot access point is integrated either in the IVI or a separate telematics system.
4) Dedicated Short Range Communication (DSRC) radios [4] are being considered to be integrated into active safety systems to support vehicle-to-vehicle safety communications.

Each physically separate communication interface with the outside world has to be protected separately, which leads to duplicate security functions on the same vehicle.

Today, all vehicles in the United States provide a standard Onboard Diagnostic (OBD) port. This interface, although intended primarily for vehicle repair and inspection, enables everyone to access a vehicle's internal networks, diagnose vehicle electronic systems, and update the ECU firmware.

III. Malware and How They Spread

Malware exist in many forms. They include:

1) *Virus*: A virus is a malicious program that can reproduce by copying itself into other computer programs and files. When a virus-infected file runs on a computer, the virus will be activated and will attempt to infect other programs on the local and remote computers. Each newly infected program will in turn try to infect even more programs and computers.
2) *Worm* [7]: A worm is a malicious program that spreads from computer to computer without having to attach itself to other programs.
3) *Trojan horse*: A Trojan horse is a malicious program that gains unauthorized access to a computer while appearing to perform a desirable function.
4) *Spyware*: A spyware is a malicious program that gathers information on a computer and sends it to other entities without the computer owner's knowledge.
5) *Ransomware*: A ransomware is a malicious program that restricts access to an infected computer to demand a ransom to be paid for the restriction to be removed. Recent years have seen a rapid growth in the number of Ransomware [17].
6) *Rootkit*: A rootkit is a software program that hides the existence of malicious programs on an infected computer. This can be done by, e.g., disguising malware as necessary files that antimalware software will overlook. Rootkits can only be installed on a computer with root access privilege. They are difficult to detect. Removing a rootkit often requires operating system reinstallation or even hardware replacement.

To evade detection, many modern malware can modify themselves to look different each time they replicate. Such malware can be polymorphic or metamorphic as explained below:

1) *Polymorphic malware* [23]: A malware that contains a constant malicious code body together with the necessary information to decrypt and encrypt this code body. At each replication, the code body is decrypted and then encrypted again so the new generation of the malware will look different from the previous one. The code body, and sometimes the decryption data, will always remain constant—a characteristic that has been used to detect polymorphic malware.
2) *Metamorphic malware* [24]: A malware that uses code evolution techniques to transform into a new look without any constant part each time it replicates.

Malware can enter a computer through a variety of channels, e.g., coming in downloaded files and applications, in files from a

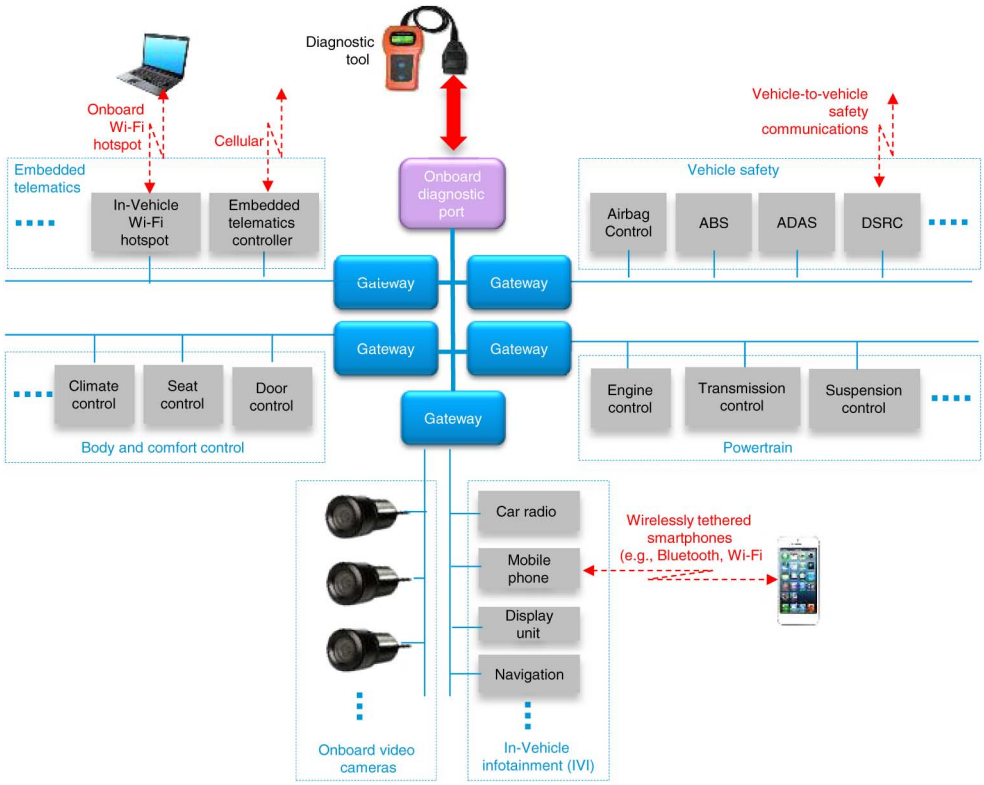

Fig. 1. Schematic illustration of existing in-vehicle network architectures.

removable media, and in e-mail attachments. Just because a malware is on a computer does not mean it can harm the computer. For a malware to do harm, it needs to execute on a computer. A malware may run on a computer through a front door, i.e., by going through the interactive user authorization procedures. A malware may also run a computer through a backdoor, i.e., bypassing the computer's authentication procedures to run without the computer owner's involvement. A backdoor can exist as a result of system design vulnerabilities. A backdoor can also be installed by a malware that is already on the computer. Sometimes, software makers may keep backdoors in software to facilitate remote technical support for the software. The higher access privilege a malware gets, the more damage it will be able to cause.

In addition to attacking computers, malware have been widely used to access private data and communication networks to steal information and to disrupt the normal operation of a networked information system.

IV. Malware and Connected Vehicles

Communications with the external world expose vulnerabilities that can be exploited by malware to infect a vehicle. This section discusses such vulnerabilities of connected vehicles. We will also discuss how Linux operating systems widely used on vehicles impact malware attacks to vehicles. Finally, we will examine what could motivate hackers to mount malware attacks to vehicles.

A. Potential Ways for Malware to Infect Connected Vehicles

Any network interface, physical or wireless, could be exploited by malware to infect a vehicle. The likelihood for a malware to enter a vehicle and run on it depends on many factors. Several of these factors are:

1) vulnerabilities in the design and implementation of hardware, software, applications, and communication systems on the vehicle;
2) users' insufficient ability to safeguard file downloads into the vehicle when users access Websites and download applications from external sources;
3) vulnerabilities in the external data that enters a vehicle (e.g., likelihood that a software update package is infected with malware before it is loaded onto a vehicle);
4) vulnerabilities in the operating systems used on the vehicles.

Next, we discuss several scenarios where malware could exploit these vulnerabilities to infect a vehicle.

1) Onboard Diagnostic Ports: A vehicle's OBD port enables anyone to access a vehicle's internal networks to eavesdrop on messages over these networks, send malicious messages, communicate with ECUs, and update ECU firmware using standard and low-cost off-the-shelf data logging and programming equipment. Attackers can install malware on the ECUs as easily as car enthusiasts can tune and reprogram their ECUs. The diagnostic tools can also be infected with malware without their users' knowledge. These malware-infected diagnostic tools can in turn infect the ECUs when they communicate with each other. The Right-to-Repair laws in the United States, and similar laws in many other countries, require that automakers provide the same information to all independent repair shops as they do to automaker authorized dealerships to create a fair vehicle repair marketplace. This means that any repair shop and personnel, many of which are

not controlled by automakers, can update ECU firmware. Hence, it will be difficult to control the sources and contents of the firmware update packages, creating more vulnerability for malware to infect a vehicle.

2) OTA Firmware and Software Updates: As mentioned in Section II, some vehicles already have over a hundred million lines of software code and the complexity of in-vehicle software continues to grow. Thus, remote OTA ECU firmware update becomes increasingly important and expected, which increase the chances for malware to infect vehicles from remote sites.

3) Embedded Web Browsers: Vehicles have begun to offer embedded Web browsers to allow users to access the Internet and to download media contents and applications to vehicles from remote application stores, which may or may not be provided by an automaker. Accessing the Internet and downloading applications to vehicles provide a channel for malware to be downloaded to the vehicle, in similar ways malware are attracted to our computers and smartphones.

4) Aftermarket Equipment: Aftermarket IVI systems use tethered or embedded devices to provide network connectivity and host third-party applications. A growing security threat is the wide availability of aftermarket devices commonly used to replace factory installed equipment, such as IVI head units. These are often Windows, Linux, or Android-based devices that can be easily modified to run malicious applications.

5) Removable Media Ports: Most modern vehicles provide USB ports for users to connect brought-in devices. These interfaces allow the embedded systems on a vehicle, such as the IVI system, to access data files (e.g., music files) on the removable media. They have also been used by vehicle manufacturers and aftermarket device manufacturers to update ECU firmware. These removable media can be infected with malware, which can in turn spread into the vehicle's embedded systems. Two malware attacks to vehicles using removable media have been identified in [6]. First, a malware on a removable media can be stored under a specific name that can trick the vehicle's embedded system into believing it as a firmware update and will, therefore, install and run the malware when the removable media is plugged in. Second, malware can be added to the music files. The input vulnerabilities of the media player's firmware can then be exploited to allow the malware to run when the music file is played. Researchers have shown an attack by modifying a Windows Media Audio (WMA) file that, when played on the vehicle's media player, sends malicious CAN messages to compromise multiple in-vehicle systems.

Once a malware infects one electronic subsystem on a vehicle, such as the IVI system, it will be able to harm other electronic subsystems in the vehicle because different subsystems are interconnected to enable cross-system functionality. Malware running on one subsystem can send malicious messages to disrupt the normal operations of other parts of the vehicle. It can also perform denial-of-service (DoS) attacks by flooding other subsystems and in-vehicle networks with bogus messages to bring down other subsystems.

B. Impact of Linux on Malware Threats to Vehicles

Many vehicles use Linux operating systems. Linux has experienced significantly fewer malware than other operating systems such as Microsoft Windows and Android. Several factors have contributed to this phenomenon.

1) Most Linux distributions are installed from a limited set of repositories owned or operated by trusted distributors. Software installation packages are cryptographically signed by trusted distributors and the signatures have to be validated before installation is performed.
2) A large number of Linux distributions are currently in use. This makes it difficult for malware to spread because malware designed for one Linux distribution typically will not work on another Linux distribution.
3) A Linux system implements different levels of access privileges for different types of users such as root user, regular users, and guests. Each access privilege, except GUESS access, is protected with a password. Explicit user permission is usually required for a user or program to gain root access privileges. This makes it difficult for a malware to install on a Linux computer with root access.
4) Linux is open source. This enables a worldwide community of security experts to detect and address Linux security vulnerabilities on a routine basis.

However, Linux is not immune to malware [18], [21]. Linux users and applications can be fooled into allowing malware to enter and execute [22], [19]. Recently, Linux malware have been on the rise [19], [20].

More importantly, several factors that have made malware difficult to spread on Linux computers may not be as effective in preventing malware from infecting vehicles. For example,

1) The vehicle's open OBD port allows virtually everyone to access the ECUs and update their firmware. This means that ECU firmware update packages will not come from a restricted small number of well-controlled and trusted sources.
2) Each automaker will likely use a common Linux distribution across as many vehicle models as possible. The automotive industry has been developing an industry-wide Linux-based software framework—GENIVI—for IVI systems. This will make it easier for malware to spread among vehicles.
3) A malware can severely harm a vehicle even with only regular user permissions and not root privileges.
4) As discussed previously and demonstrated repeatedly by many researchers [4]–[6], [32], [33], any successful malware attack to a vehicle can cause significantly more severe consequences than compromising consumer electronics. Thus, even a small number of successful malware attacks to vehicles represent a serious problem.

These mean that malware defense capabilities will also be necessary for vehicles that use Linux operating systems.

C. What Would Motivate Malware Attacks on Vehicles

Understanding what would motivate attackers to spread malware to vehicles is essential to understanding the risks and impacts of malware attacks. Several likely motivations are:

1) *Fun and publicity*: Many security attackers hack computers just for the fun of doing it or as a way to show off their security skills. It is anticipated that many hackers will

view the increasing population of connected cars as highly interesting targets. Hacking cars could generate greater publicity than hacking consumer computers or smartphones.

2) *Breach driver privacy*: By injecting a spyware onto a vehicle, an attacker could retrieve such private information regarding the driver as where he has been, his driving habits, his phone directory and call history, the music he listens to, and much more.
3) *Ransom*: A ransomware could allow an attacker to remotely disable selected vehicle functions (e.g., lock the doors or the in-car radio, immobilize the engine) in a way that the vehicle owner's car keys can no longer activate them. The attackers can then demand ransom to be paid before reenabling these functions.
4) *Theft*: A malware attack could allow an attacker to open a vehicle's doors, deactivate its alarms and immobilizer so that he can steal the vehicle.
5) *Sabotage*: A malware could create a wide range of disruptions to a driver. Examples include remotely locking up a vehicle's infotainment system to specific radio channels, turning up and down the audio volumes, displaying erroneous low tire pressure messages or other messages that require the driver to take urgent actions while driving, posing arbitrary messages or images on the head unit display. Any such disruption could cause the driver to make deadly mistakes while driving, cause traffic accidents, and damage a carmaker's reputation.
6) *Harm people and properties*: A malware could allow an attacker to directly harm a driver and his vehicles. It could, e.g., disable the brakes or apply sudden braking while the vehicle is in motion to cause accidents.
7) *Disrupt transportation*: Malware can cause a large number of vehicles to make dangerous maneuvers to cause transportation chaos across a large region.

Malware attacks can be carried out by individuals, groups, crime or terrorism organizations, or even foreign governments. As more connected vehicles are deployed, more malware targeted to vehicles can be anticipated because vehicles will become more attractive targets for hackers.

V. Existing Approaches to Securing Vehicles and Their Limitations

Conventional in-vehicle networks, such as CAN, MOST, and LIN do not provide adequate security protections. CAN buses, e.g., provide no mechanism for device authentication, message confidentiality protection, or message replay protection. It uses simple checksums to protect message integrity, which is widely known to be insufficient.

This section discusses the typical methods that have been used to secure in-vehicle networks and their limitations.

A. Physical Network Separation

Physically separated networks have been used to isolate different electronic subsystems on a vehicle. This, however, becomes a less effective way to secure in-vehicle networks as different electronic subsystems need to communicate more with each other and with external networks to support advanced vehicle functions. For example, ADAS functions require communications among vehicle safety, power train, body control, and even the IVI subsystems. Close communication among different subsystems is even more important for electrical vehicles. For example, to accurately predict remaining travel distance and smartly dispatch the limited battery power, an electrical powertrain system needs to communicate with the body control system to monitor and even adjust the states of climate control and windows that impact battery power consumption.

B. Message Obfuscation

Automakers often use proprietary messages for ECUs to communicate with each other to support advanced functions beyond what standard messages can support. They seek to keep the proprietary messages known only to authorized parties. However, these messages are known to be easy to decode by reverse engineering.

C. Predefined Messages

Many ECUs on today's vehicles accept only predefined message types. Similarly, onboard gateways will only relay predefined message types. The predefined messages are often hardcoded in firmware. While this approach helps reduce virus infection, it severely limits a vehicle's ability to communicate. It is effective only when all the traffic consists of only predefined messages. It, therefore, cannot be used to control most user application traffic that contains arbitrary data and is a popular venue for malware infections.

D. Limited Applications, Application Features, and User Experiences

Inadequate security in today's vehicles limits the applications available in vehicles. For example, web browsers embedded in vehicles typically do not allow content downloads or even video streaming mainly due to insufficient ability to prevent malware from infecting the vehicle.

E. Control of Applications and Contents

Application-specific authentication is often required before a firmware update procedure can initiate with an ECU. Most existing vehicles, however, use rudimentary challenge-and-response procedures for such authentication, which have been shown to be easy to crack [5].

Some ECUs only accept software updates that are cryptographically signed. However, a rapidly growing number of malware are now also signed with legitimate signatures and digital certificates [17]. This means that code signing alone may no longer be sufficient to ensure malware-free code.

Furthermore, application-specific security procedures cannot control network-layer or link-layer accesses to ECUs and to a vehicle's internal networks.

F. Network-Specific Security Protocols

As Ethernet is introduced into vehicles, IEEE 802.1AE (MACsec), although not yet used on vehicles, could be used to

provide hop-by-hop security protections for the Ethernet-based networks on a vehicle. MACsec supports data integrity, data origin authentication, data confidentiality, and replay protection using symmetric-key algorithms.

However, MACsec currently applies only to Ethernet while vehicles are expected to also use non-Ethernet networks for the foreseeable future. Using MACsec also requires in-vehicle devices to implement neighbor discovery protocols and IEEE 802.1X-based protocols for authenticating devices and for devices to establish security keying materials, which can be overly complex and costly for many resource-constrained in-vehicle devices.

G. Application-Specific Security Protocols

Security protocols have been designed to secure selected categories of automotive applications. For example, IEEE 1609.2 is designed to support primarily the authentication of vehicle safety message broadcasts over DSRC radios. It could be used to support message authentication and protect message integrity over in-vehicle networks. However, it requires all devices to support public-key cryptography for digital signatures and digital certificates. As such, it requires a Public Key Infrastructure (PKI) to supply the digital certificates to the hundreds of millions of vehicles in a large country as the United States. This introduces a new set of technical and business challenges [4].

Furthermore, IEEE 1609.2 does not provide a resource-efficient way to protect the confidentiality of broadcast messages—an important requirement for in-vehicle networks.

VI. Unique Challenges in Defending Vehicles Against Malware

Most electronic devices on a vehicle have limited processing, memory, and communication capacities due to stringent cost constraints. The average age of passenger cars and light trucks on the U.S. roads reached 11.4 years in 2013 [36]. New vehicles manufactured today are expected to last even longer. Over such long lifespans of a vehicle, new malware will be created and hence more onboard resources and more sophisticated techniques will be needed for malware defense. But once a vehicle leaves its manufacturing plant, its onboard resources will be difficult to change. Therefore, the first challenge is how to ensure adequate malware defense for a vehicle over its long lifecycle with minimal onboard resources that cannot be changed.

Onboard threat defense capabilities must be kept up-to-date over time to provide adequate protection for a vehicle throughout its long lifecycle. This up-keep should be accomplished with no or negligible inconvenience to vehicle owners and preferably with no user intervention. However, an onboard malware defense system cannot practically rely only on itself to keep it up-to-date over its long lifespans due to limited onboard resources and the need for new malware defense techniques and software over time. Cloud-based malware defense services can more effectively detect new malware and update the onboard malware defense capabilities. Thus, the second important challenge is how to balance the malware defense processing load on the vehicle versus the load of vehicle-to-cloud communication for the vehicle to maintain up-to-date and adequate protection over time.

A cloud-based threat defense system must rely on information from vehicles to detect threats that are relevant to vehicles. This is because in many occasions only the vehicles have information about the threats they have experienced. However, not all threat-related information from vehicles can be trusted because some vehicles may have been compromised and used to send false information. Attackers with physical access to a vehicle can also fake input data to the vehicle's communication unit to trigger it to send false information without compromising the vehicle's security system. Thus, the third challenge is how should a cloud-based threat vehicle defense system determine how trustworthy threat-related information from vehicles is.

Many malware are irrelevant to vehicles. Without the ability to determine which malware are relevant to vehicles, the potentially large number of vehicle-irrelevant malware can trigger frequent unnecessary updates to vehicles' onboard malware defense systems and result in a heavy waste of wireless bandwidth. Therefore, the fourth challenge is how can a vehicle malware detection system determine which malware are relevant to vehicles and to each specific vehicle model.

The malware defense solution for connected vehicles must be highly scalable. In the United States, a large automaker would sell over two million new vehicles each year and have tens of millions of vehicles in operating condition at any time. Security operations must support the large and changing vehicle population in ways that will not cause significant inconvenience to vehicle owners. These operations include provisioning initial onboard security systems, distributing threat-related intelligence (e.g., malware signatures) to vehicles, updating onboard threat defense system parameters and software, collecting and analyzing threat-related information from vehicles, and performing OTA threat mitigation such as malware removal.

VII. Existing Malware Defense Mechanisms and Their Limitations When Applied to Connected Vehicles

A wide range of malware defense mechanisms have been developed for protecting enterprise networks and personal computers [8]–[11]. They can be classified into signature-based, behavior-based, and heuristic-based techniques. This section discusses the main limitations of applying these mechanisms to protect connected vehicles.

A. Signature-Based Malware Detection

Signature-based malware detection consists of two sequential steps. First, new malware must be identified and a unique representation or a signature of each malware is generated. This process is usually accomplished by a combination of manual and automated analysis of the information collected from networks and user devices [29]. Second, each computer retrieves the malware signatures. It can then detect whether a file or data stream contains malware by scanning the data for malware signatures.

Signature-based detection has been the most widely used detection techniques [15].

Compared to behavior-based and heuristic-based detection mechanisms, signature-based detection is simpler and safer to implement and typically requires lower processing power.

However, it has several limitations when applied to protecting connected vehicles.

First, the already large number of malware, which can further grow significantly over a vehicle's long lifecycle, can result in a prohibitively large malware signature database for a resource-constrained in-vehicle device to maintain and process. Today, a typical malware signature database already contains from hundreds of thousands to over a million malware signatures resulting in tens to hundreds of megabytes of data in a malware signature database [16]. Therefore, a large malware signature database has to be installed on each vehicle when the vehicle is manufactured. The number of malwares detected on the Internet has been growing exponentially over recent years [12]–[14]. Once tens to hundreds of millions of vehicles are connected to the Internet in a large country like the United States, additional malware targeting these vehicles will be anticipated. Therefore, over a vehicle's 11.4 year or even longer average lifetime [36], the amount of storage space required to store malware signatures on a vehicle can grow significantly. When a vehicle is manufactured, it will be difficult to predict how large a database should be installed on a vehicle so that it will be sufficient to handle all the potential new malware over the vehicle's long lifecycle. Consequently, additional storage capacity may need to be added to a vehicle over time. Furthermore, as the number of malware signatures grows, the amount of processing power required to scan files against malware signatures will also increase. That is, the CPU capacity required on a vehicle faces the same issue as faced by the required storage space for malware signatures.

Second, the malware detection functions in a cloud have to rely on information from the vehicles to detect new malware, but such information cannot always be trusted because some vehicles may have been compromised. With physical access to the vehicle, attackers could even manipulate input data to a vehicle's communication device to cause the vehicle to send false information without having to compromise the vehicle's security system. Vehicles already infected by malware could also be used by the malware to send false information to the cloud. Ways to effectively determine the trustworthiness of threat-related information from vehicles have not been well studied. Some existing threat defense systems assign reputation levels to data sources. However, they typically rely on prior knowledge about the reputations of the sources. For example, a device under a network operator's control may be given higher reputation than other devices. Approaches will be needed for determining how reputable vehicles are as safe information sources.

Third, ways for determining which malware detected in the cloud is relevant to vehicles are missing. Consequently, vehicles will have to be updated to handle all threats detected by the cloud, including threats that are irrelevant to vehicles. This will waste wireless bandwidth needlessly.

Fourth, signature-based detection is ineffective in detecting metamorphic malware [24]. This is because the metamorphic malware's mutating code makes it difficult or impractical to create signatures for every instance of the malware a prior.

Fifth, signature scanners cannot detect new malware ("zero-day" malware) for which no signatures have been generated.

Sixth, the malware signature database on each vehicle must be updated timely when new malware are discovered and new malware signatures are generated by the remote entity that detects new malware. However, frequent malware signature updates to an automaker's tens of millions of vehicles will be complex to manage and can also be costly to vehicle owners.

Seventh, with the increasing volume of vehicle communications, scanning the incoming traffic and downloaded files against a potentially very large database of malware signatures can consume prohibitively high CPU power on resource-constrained in-vehicle devices.

B. Behavior-Based and Heuristic-Based Malware Detection

Behavior-based malware detection [31] determines whether a program is malicious by observing what it does when it executes. Heuristic detection [30] examines program files for suspicious characteristics (e.g., rare instructions) or emulates the execution of a program or selected ports of the program to determine whether it will perform malicious acts. Heuristic detection often uses rule-based, data mining, and machine-learning techniques to learn the features of a program to determine whether they are malicious. Many existing antivirus software uses some forms of heuristic-based detection.

Behavior-based and heuristic-based detection mechanisms allow each vehicle to rely on itself (i.e., without depending on off-board systems) to detect malware including zero-day malware that have not been previously seen. They can also detect polymorphic, metamorphic or other types of malware that mutate when replicating themselves.

However, behavior-based and heuristic-based detection mechanisms can have excessively high false positive or false negative rates. They are significantly more complex to implement and resource-intensive to run on each vehicle. Therefore, they may not be suitable for use on resource-constrained in-vehicle devices that also demand ultra-low false positive and false negative rates.

Furthermore, any heuristic-based or behavior-based technique installed on a vehicle today will likely become obsolete and need to be updated or replaced during the vehicle's long lifecycle.

VIII. A Cloud-Assisted Vehicle Malware Defense Framework

An important realization from the analysis in Sections VI and VII is that it will be impractical to rely solely on each individual vehicle to sufficiently protect itself against malware over the vehicle's long lifecycle. Cloud services can help defend resource-constrained devices against malware.

This section presents a cloud-assisted vehicle malware defense framework. We start with the end-to-end malware defense framework with the main design principles. We then discuss in more detail the onboard malware defense functions and architectures.

A. Architecture Overview and Principles

Cloud services have been used to detect malware in Internet traffic destined to computers. A source computer or its Internet

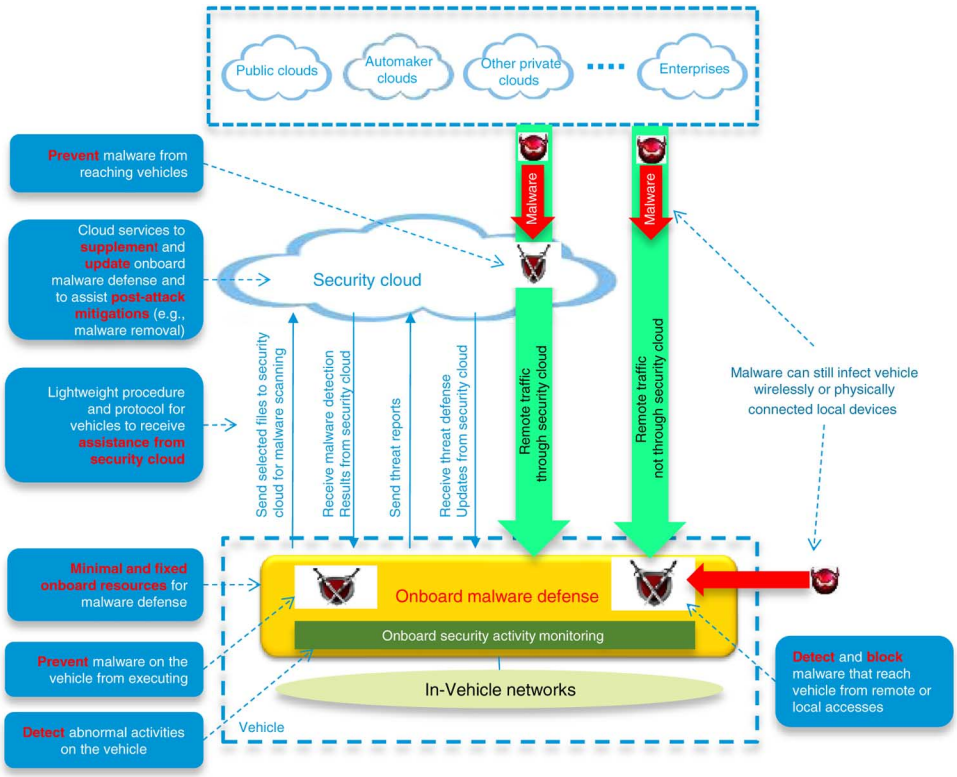

Fig. 2. Cloud-assisted vehicle malware defense framework.

service provider redirects communication requests (such as Web requests) to a security cloud. The security cloud contacts the ultimate destination to redirect the return traffic to the security cloud, where the traffic will be scanned for malware before relayed to the source computer. It has also been proposed that smartphones send suspicious files to a cloud to be scanned for malware [35].

A typical existing cloud-based malware scanning system does not implement any malware detection or scanning capability on the computers to be protected. However, local malware detection capabilities will be necessary on vehicles because malware can come to a vehicle not only from remote sources but also from local devices connected to the vehicle wirelessly or physically (e.g., OBD ports, tethered smartphones, and in-vehicle Wi-Fi).

Therefore, we propose a cloud-assisted vehicle malware defense framework that consists of malware defense functions on each vehicle and allows these functions to be lightweight in terms of processing and storage, and yet have a full spectrum of malware defense capabilities that are kept up-to-date over time. This is accomplished through assistance from a security cloud, as schematically illustrated in Fig. 2.

The onboard malware defense functions protect the vehicle against malware that have reached the vehicle from any attack surface such as from local and remote network connections, physical and wireless network connections, V2I and V2V communications, removable media, and replacement ECUs.

The vehicle maintains local threat information databases (e.g., threat signatures and whitelists) to assist onboard malware detection. However, the total size of such databases will be capped and needs not increase even as the number of malware grows over time. This eliminates the problems of having to install large malware information databases on a vehicle and to add additional storage and processing capacities to keep up with the growing number of malware over time.

The limited size of the onboard malware information databases may constrain the vehicle's ability to detect malware locally. Therefore, the onboard malware defense capability will be compensated by the malware detection and defense capabilities in the security cloud.

The security cloud, as illustrated in Fig. 3, provides several services to assist the vehicles in malware defense:

1) *Examine files received from vehicles for malware*: When the onboard malware defense cannot determine, with sufficient confidence, whether a file contains malware, it will send the file, or in some cases a short unique representation of the file, to the security cloud to be further examined for malware before allowing the file to execute on the vehicle.
2) *Discover new malware that are relevant to vehicles*: Collect and analyze threat-related information from vehicles and other sources to detect new malware that may harm vehicles.
3) *Scan network traffic destined to vehicles for malware*: Some or all V2I communications can be routed through the security cloud that can detect and remove malware in the traffic that passes through it.
4) *Update onboard malware defense*: The security cloud will interact with the vehicles to keep their onboard malware defense functions up-to-date over time.
5) *Support OTA malware removal*: The security cloud will interact with the onboard security gateway to detect and

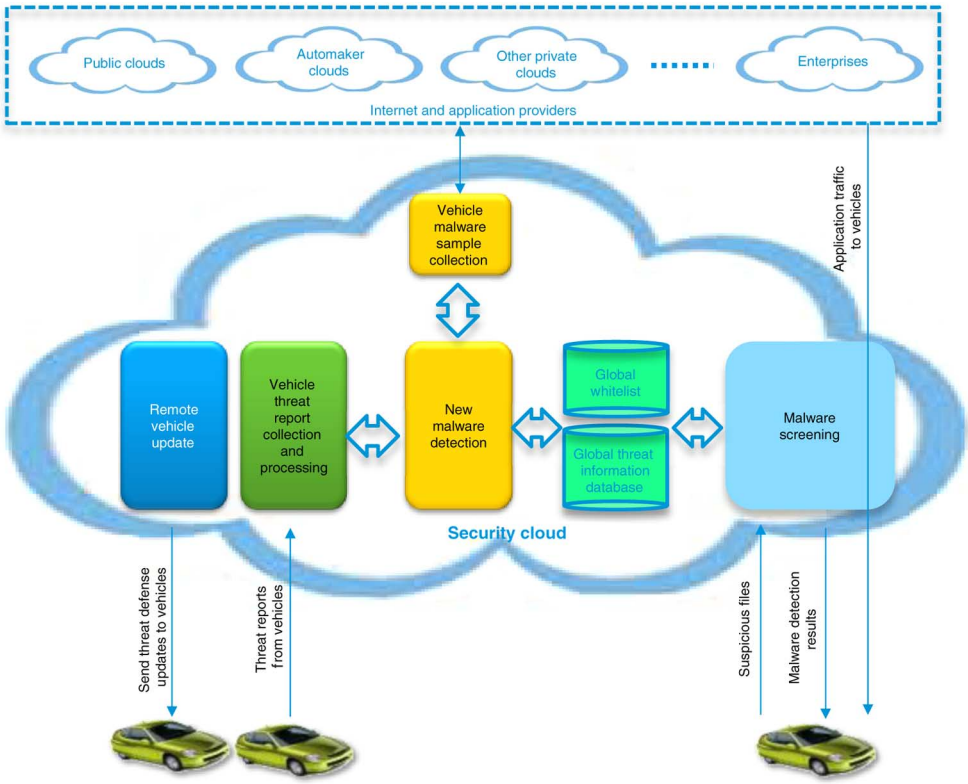

Fig. 3. High-level security cloud functional architecture.

remove malware on the vehicle to minimize the need to bring malware-infected vehicles to service centers for malware removal.

These malware defense functions provided by the security cloud require more resources or need more information than any individual vehicle has, and hence are typically impractical to implement on vehicles.

In addition to analyzing information from vehicles and performing file and forensic examinations of infected vehicles and other computer systems, the security cloud can proactively collect malware samples so it can detect new malware early and even before they spread to vehicles. Malware sample collection can be passive or active [28]. Passive malware collection lures malware to enter a trap in the security cloud [25], [26]. With active malware collection, the security cloud will open suspicious or known malicious URLs and files to discover and collect new malware samples from them [27].

The communication load and the delay for sending files to the security cloud for malware examination should be minimized. To this end, several factors need to be balanced carefully: malware detection techniques used on vehicles, the amount and contents of malware-related information stored on each vehicle, what files should be sent to the security cloud for malware screening and when they should be sent, and ways to identify and represent these files. Some of these factors will be fixed at vehicle manufacturing while others can be adjusted dynamically. For example, the vehicle, in collaboration with the security cloud, can dynamically adjust the contents in its fixed-size local threat information databases (e.g., maintaining information on the most popular and the most damaging malware) to help balance local processing load versus the overhead of communication with the security cloud.

The proposed vehicle malware defense framework does not dictate which malware detection mechanisms should be used on vehicles or in the security cloud. This allows the malware protection capabilities to evolve as new malware detection techniques are created over time. Different vehicle models can also use different malware detection mechanisms to match their specific malware risk profiles and protection needs.

B. Onboard Malware Defense Functions and Architecture

Fig. 4 schematically illustrates the onboard malware defense functional architecture. Its functions can run on an onboard security gateway that connects the vehicle with the outside world or be distributed on multiple in-vehicle devices.

The malware defense functions on the onboard security gateway perform the following main tasks:

1) Detects malware-infected files that have got on the security gateway and prevents them from execution on the security gateway or passing through the security gateway to infect other in-vehicle devices.
2) Detects and blocks malware in the traffic passing through the onboard security gateway to in-vehicle devices to eliminate the need to implement malware scanning on every in-vehicle device.
3) Detects suspicious activities on the vehicle that may indicate a security attack.

For a malware to harm a system, it has to execute. Therefore, any file that attempts to execute on a vehicle is captured before it

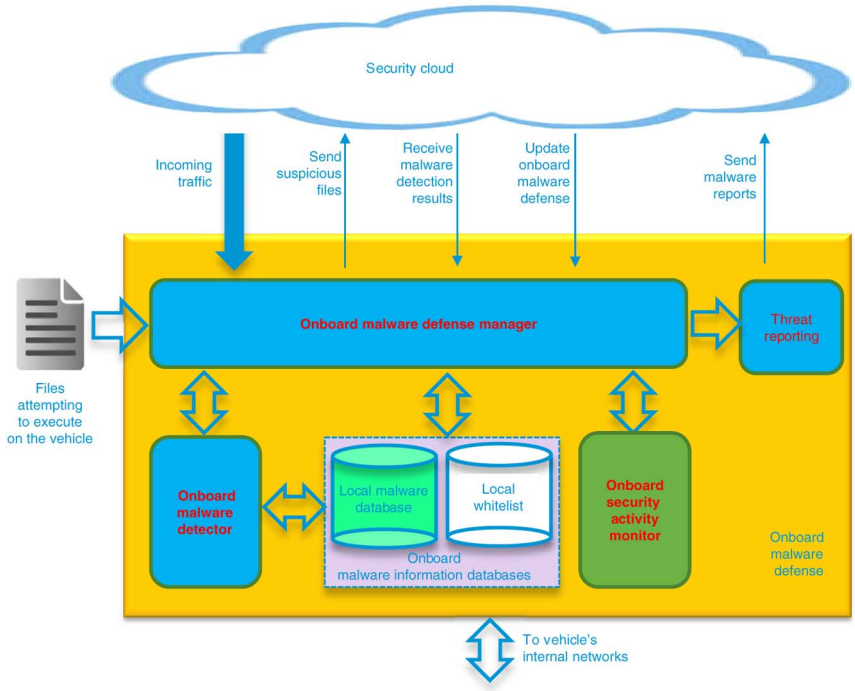

Fig. 4. Onboard malware defense functional architecture.

is allowed to execute and submitted to the Onboard Malware Defense Manager to start the malware examination process. For many operating systems, attempts to execute files that have not been authorized previously need user's explicit authorization (e.g., a pop-up menu to prompt the user to allow a file to run). This user authorization trigger can be used to automatically trigger the Onboard Malware Defense Manager to initiate malware scanning for a file without modifying the operating system.

Malware scanning for nonexecutable files can wait until the system processing load is light. However, nonexecutable files may be infected with malware. If these malware attempt to run (e.g., when the nonexecutable files are used), they will be captured and examined for malware before they are allowed to run.

Important executable files can be cryptographically signed by trusted entities and only allowed to execute on a vehicle after their signatures are positively verified by the Onboard Malware Defense Manager. However, recent years have seen a rapidly growing number of malware that are also signed with legitimate digital signatures [17]. Therefore, to further enhance the protection for a vehicle, mission-critical software programs can be further examined for malware after their digital signatures are positively verified by the vehicle.

Fig. 5 illustrates a process for the onboard malware defense procedure. Each file, which needs to be examined for malware, will be first examined by the Onboard Malware Detector, leveraging information in the Onboard Malware Information Databases.

The Onboard Malware Detector can implement any combination of malware detection techniques. It will interface with the rest of the onboard malware defense system via well-defined application programming interfaces (APIs) so that it can be updated or replaced over the air when it becomes out of date or obsolete.

The Onboard Malware Information Databases maintain a Local Whitelist containing representations of files known not to contain malware, and a Local Malware Database containing signatures of known malware or malware behaviors depending on the malware detection techniques used on the vehicle.

If the onboard malware defense system can determine with sufficient confidence whether a file contains malware, it will make a local decision on whether to allow the file to run on the vehicle without having to communicate with the security cloud. This can be the case when, e.g., when the file is on the Local Whitelist or in the Local Malware Database.

If onboard malware defense cannot determine whether a file contains malware, it sends the file, or a unique short representation of the file, to the security cloud for further malware examination. Digital signature can serve as the file's short representations. Mission-critical files authorized to run on vehicles can be digitally signed by the security cloud or trusted third parties. If a file is not signed, the vehicle can sign the file and use the signature to represent the file.

Upon receiving a signature from a vehicle, the security cloud verifies whether it has generated this signature. For this purpose, the security cloud can maintain a database of all the files it has authorized to run on vehicles together with its digital signatures for these files. If the signature is supposed to have been generated by a trusted third party, the security cloud contacts the third party to verify whether it has generated the signature. A positive verification that the signature was generated by a trusted entity will be a strong indication that the file is safe. If the signature is not generated by the security cloud or a trusted third party, the file on the vehicle has likely been altered or its signature was generated by an attacker, with, e.g., stolen signature keys.

The Onboard Security Activity Monitor function monitors the traffic over the vehicle's internal networks to detect abnormal

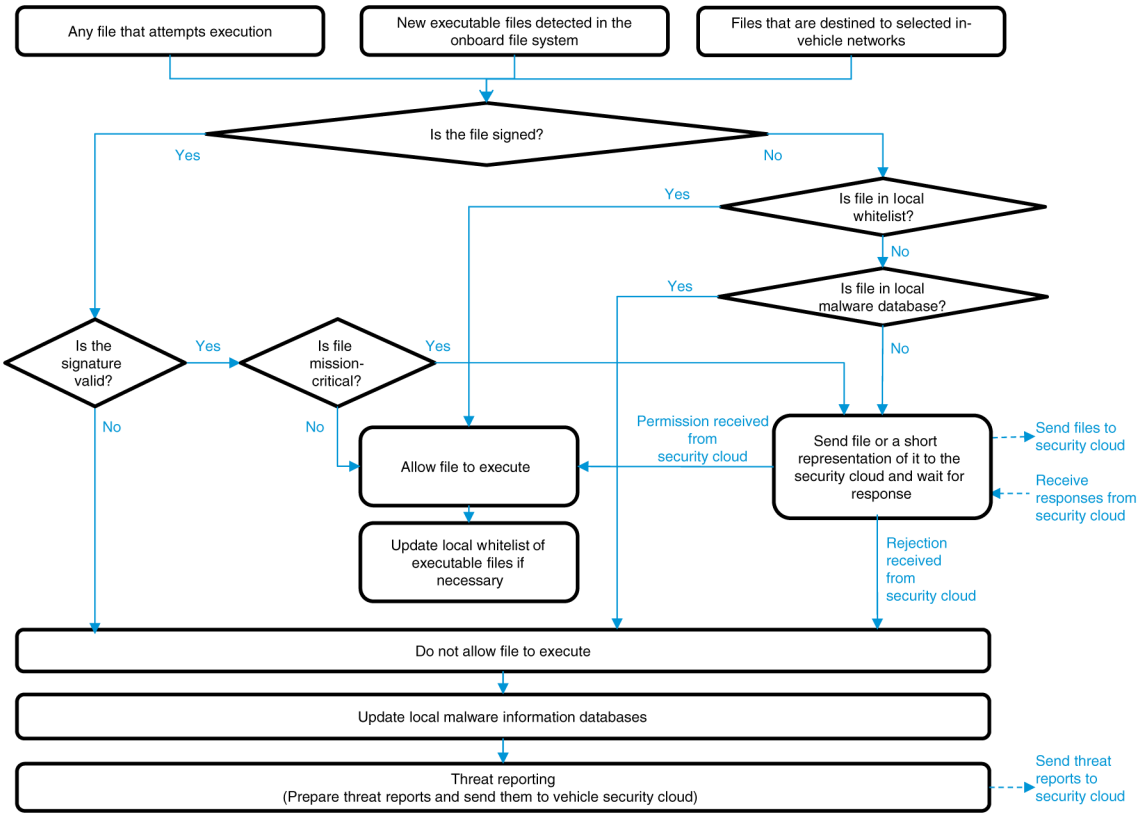

Fig. 5. Onboard malware defense procedure.

traffic patterns and abnormal attempts to access in-vehicle devices, which may indicate a security attack.

The Threat Reporting function sends Threat Reports to the security cloud to inform the security cloud of the malware or other security attacks that the vehicle has experienced. A Threat Report can contain information or claims about the alleged sources of the security attacks. The security cloud must determine the trustworthiness of the Threat Reports because the reports from some vehicles cannot be trusted.

IX. Discussions

Several issues await further investigation to enable large-scale implementation of the proposed framework. One question is which in-vehicle devices should implement the onboard threat defense functions. Ideally, these functions can run only on a single onboard security gateway that controls all the external communication interfaces on the vehicle. Traffic from external sources must traverse this gateway to reach ECUs and will be scanned by the gateway for malware. However, real-time malware scanning of all pass-through traffic can require excessive processing power. Furthermore, the gateway alone, with its limited onboard resources, may not be able to detect all malware in the pass-through traffic. Therefore, mission-critical ECUs should either implement the onboard malware defense functions or be able to capture the files that attempt to run on them and send the files to the onboard security gateway, which can use the process described in Section VIII to examine these files for malware.

Another issue is how to minimize communication overhead and the delay incurred by using cloud services to help vehicles examine files for malware. As discussed earlier, if the security cloud has access to all the files authorized to run on vehicles, a vehicle can send the files' digital signatures, rather than the files themselves, to the security cloud. However, if the security cloud cannot access such a list (e.g., when vehicles can download data from the Internet), more resource-efficient ways will be required for vehicles and the security to collaboratively examine the files for malware.

X. Conclusion

Malware is a serious threat to the increasingly connected vehicles. They can infect vehicles in a number of ways and cause a wide range of severe disruptions and damages. Defending vehicles against malware imposes unique challenges. This paper identified such unique challenges and presented a cloud-assisted vehicle malware defense framework that can address these challenges.

References

[1] L. O'Carroll. (Jul. 2013). *Scientist Banned from Revealing Codes Used to Start Luxury Cars* [Online]. Available: http://www.theguardian.com/technology/2013/jul/26/scientist-banned-revealing-codes-cars. Accessed on: Oct. 18, 2013.

[2] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proc. Netw. Distrib. Syst. Secur. Symp.,* 2011.

[3] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *Proc. 19th USENIX Secur. Symp.,* 2010, pp. 11–13.

[4] L. Delgrossi and T. Zhang, *Vehicle Safety Communications: Protocols, Security, and Privacy.* Hoboken, NJ, USA: Wiley, 2012.

[5] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy,* 2010, pp. 447–462.
[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analysis of automotive attack surfaces," in *Proc. USENIX Secur. Symp.,* 2011.
[7] D. M. Kienzle and M. C. Elder, "Recent worms: A survey and trends," in *Proc. ACM Workshop Rapid Malcode,* 2013, pp. 1–10.
[8] V. Kumar, J. Srivastava, and A. Lazarevic, *Managing Cyber Threats: Issues, Approaches, and Challenges.* New York, NY, USA: Springer, 2005.
[9] N. Idika and A. P. Mathur, "A Survey of Malware Detection Techniques." Purdue Univ., Tech. Rep., 2007.
[10] G. Serazzi and S. Zanero, "Computer virus propagation models," *Perform. Tools Appl. Netw. Syst., Lecture Notes Comput. Sci.*, vol. 2965, pp. 26–50, 2004.
[11] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A Survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, Jan. 2013.
[12] AV-TEST. (2013, Oct.) [Online]. Available: http://www.av-test.org/en/statistics/malware/.
[13] Kaspersky Lab. (2012, Dec.). *2012 By the Numbers: Kaspersky Lab Now Detects 200,000 New Malicious Programs Every Day* [Online]. Available: http://www.kaspersky.com/about/news/virus/2012/2012_by_the_numbers_Kaspersky_Lab_now_detects_200000_new_malicious_programs_every_day. Accessed on: Oct. 18, 2013.
[14] Microsoft. (2013). Microsoft security intelligence report [Online]. Available: http://www.microsoft.com/security/sir/story/default.aspx#!10 year_malware. Accessed on: Oct. 18, 2013.
[15] M. Grace, Y. Zhou, Q. Zhang, S. Zu, and X. Jiang, "RiskRanker: Scalable and accurate zero-day android malware detection," in *Proc. MobiSys'12, 10th Int. Conf. Mobile Syst. Appl. Services,* 2012, pp. 281–294.
[16] X. Hu, T. C. Chiueh, and K. G. Shin, "Large-scale malware indexing using function-call graphs," in *Proc. 16th ACM Conf. Comput. Commun. Secur.,* 2009, pp. 611–620.
[17] McAfee Labs. "McAfee threats report: First quarter," 2013.
[18] A. Ayesha. (2012, Jul. 26). *Meet Linux Viruses* [Online]. Available: http://www.unixmen.com/meet-linux-viruses/. Accessed on: Oct. 18, 2013.
[19] O. H. Alhazmi, Y. K. Malaiya, and I. Ray, "Measuring, analyzing and predicting security vulnerabilities in software systems," *Comput. Secur. J.*, vol. 26, no. 3, pp. 219–228, May 2007.
[20] A. Rogers. (2013, May 24). *Malware Affecting Linux Web Servers Major Trend in 2013* [Online]. Available: http://www.bnamericas.com/news/technology/malware-affecting-linux-web-servers-major-trend-in-2013-eset. Accessed on: Oct. 18, 2013.
[21] T. Bradley. (2010, Jun. 13). *Linux Trojan Raises Malware Concerns* [Online]. Available: http://www.pcworld.com/article/198686/linux_trojan_raises_malware_concerns.html. Accessed on: Oct. 18, 2013.
[22] Foobar. (2009, Feb. 11). *How to Write a Linux Virus in 5 Easy Steps* [Online]. Available: http://www.geekzone.co.nz/foobar/6229. Accessed on: Oct. 18, 2013.
[23] J. A. P. Marpaung, M. Sain, and H. J. Lee, "Survey on malware evasion techniques: State of the art and challenges," in *Proc. Int. Conf. Adv. Commun. Technol.,* PyeongChang, Korea, 2012.
[24] P. Vinod, V. Laxmi, and M. S. Gaur, "Survey on malware detection methods," in *Proc. 3rd Hackers' Workshop Comput. Internet Secur.,* 2009, pp. 74–79.
[25] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, "The nepenthes platform: An efficient approach to collect malware," in *Proc. 9th Recent Adv. Intrus. Detect.,* 2006, pp. 165–184.
[26] L. Spitzner, *Honeypots: Tracking Hackers.* Reading, MA, USA: Addison-Wesley, 2002.
[27] Y. M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King, "Automated web patrol with strider honeymonkeys," in *Proc. Netw. Distrib. Syst. Secur. Symp.,* Feb. 2006, pp. 35–49.
[28] Y. D. Lin, C. Y. Lee, Y. S. Wu, P. H. Ho, F. Y. Wang, and Y. L. Tsai, "How different are malware collected actively and passively," *Computer*, vol. PP, no. 99, p. 1, 2013.
[29] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surveys J.*, vol. 44, no. 2, Feb. 2012.
[30] Z. Bazrafshan, H. Hashemi, S. M. H. Fard, and A. Hamzeh, "A survey on heuristic malware detection techniques," in *Proc. 5th Conf. Inf. Knowl. Technol.* (IKT), 2013, pp. 113–120.
[31] G. Jacob, H. Debar, and E. Filiol, "Behavioral detection of malware: From a survey towards an established taxonomy," *J. Comput. Virol.*, vol. 4, pp. 251–266, Aug. 2008.
[32] R. N. Charette, "This car runs on code," *IEEE Spectrum*, vol. 46, no. 2, Feb. 2009.
[33] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaaniche, and Y. Laarouchi, "Survey on security threats and protection mechanisms in embedded automotive networks," in *Proc. DSN'03, 2nd Workshop Open Resilient Human-Aware Cyber-Phys. Syst.,* Budapest, Hungary, 2013.
[34] P. Hank, S. Muller, O. Vermesan, and J. Van den Keybus, "Automotive ethernet: In-vehicle networking and smart mobility," in *Proc. DATE'13, Des., Autom., Test. Eur. Conf. Exhibit.,* Grenoble, France, 2013.
[35] A. Houmansadr, A. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," *ACM Trans. Manage. Inf. Syst. (TMIS)*, vol. 2, no. 3, pp. 31–32, Oct. 2011.
[36] Polk. (2013, Aug.). *Polk Finds Average Age of Light Vehicles Continues to Rise* [Online]. Available: https://www.polk.com/company/news/polk_finds_average_age_of_light_vehicles_continues_to_rise. Accessed on: Oct. 29, 2013.
[37] Michael Dunn. (2013, Oct. 28). *Toyota's Killer Firmware: Bad Design and Its Consequences* [Online]. Available: http://www.edn.com/design/automotive/4423428/Toyota-s-killer-firmware–Bad-design-and-its-consequences. Accessed on: Oct. 30, 2013.

Tao Zhang (F'11) received the B.S. degree on telecommunications and the M.S. degree on electrical and computer engineering from Beijing Jiaotong University, Beijing, China, and the Ph.D. degree on electrical and computer engineering from the University of Massachusetts, Amherst, MA, USA.

Since 2012, he has been the Chief Scientist for Cisco Connected Cars at Cisco Systems, Fort Lee, NJ, USA. For over 25 years, he has held various technical and management positions, directing research and product development in mobile and vehicular networks. He holds 34 US patents and co-authored two books "*Vehicle Safety Communications: Protocols, Security, and Privacy*" (Wiley, 2014) and "*IP-Based Next Generation Wireless Networks*" (Wiley, 2012).

Dr. Zhang was a founding member of the Board of Directors of the Connected Vehicle Trade Association (CVTA) in the US. He is the Chair of the IEEE Communications Society Technical Sub-Committee on Vehicular Networks and Telematics Applications. He has been serving on editorial boards and as a guest editor for several leading journals. He has been serving on the industry advisory boards for several research organizations and was an Adjunct Professor at multiple universities.

Helder Antunes was born in Angra do Heroismo, Azores, Portugal, on July 6, 1963. He received the B.S. degree in computer science from San Jose State University, San Jose, CA, USA.

He has been the Managing Director of Cisco Connected Cars at Cisco Systems since 2011, and has held other management roles within Cisco during the past 16 years. For more than 28 years, he has been a results-driven Executive, experienced in leading product and solutions development and marketing in networking, security and services for various market segments and industry verticals. He previously worked for Computer Associates and NetManage as a Director of Engineering. He raced cars for many years and designed some of the early data acquisition systems for race cars. He is also a General Partner at Pereira Ventures and a counselor to the Regional Government of Azores, Portugal. He co-authored several publications, "*Cisco Remote Access Design*" (Cisco IT Best Practices, 2005), "Enterprise Solutions: DMVPN Extends Business Ready Teleworker" (*Packet Magazine*, 2004), and "Cisco Rated Top Telecommuting: Cisco Virtual Office (CVO) Solutions" (*CNN Money Magazine*, 2012).

Mr. Antunes received several prestigious awards such as the Cisco Pioneer Award.

Siddhartha Aggarwal received the M.S. degree in computer engineering and M.B.A. degree from Santa Clara University, Santa Clara, California, USA.

Since 2012, he has been a Technical Lead/Engineering Manager for Cisco Connected Cars at Cisco Systems, San Jose, CA, USA. For over 10 years, he has held technical and management roles, directing support and product development of IP networks, and working with customers around the world. Previously, he worked for Foundry Networks/Brocade Communications.