Car Cybersecurity: What do the automakers really think?

2015 Survey of Automakers and Suppliers
Conducted by Ponemon Institute

Ponemon Institute

RogueWave Software

Security Innovation®

Executive Summary

The Ponemon Institute recently conducted a cybersecurity survey sponsored by Rogue Wave Software and Security Innovation of over **500 automotive developers**, **engineers**, and **executives**, primarily from automotive OEMs and Tier One suppliers. The key findings of the developer survey were:

- Developers are not familiar enough with their company's program to secure software for automobiles.
- Developers do not believe their companies are taking security seriously enough, or empowering them to make software more secure.
- Developers want – but do not have – the skills necessary to combat software security threats and they do not feel they are properly trained.
- Automakers are not as knowledgeable about secure software development as other industries.
- Security is not built into the Software Development Lifecycle (SDLC) in the automotive industry.
- Enabling technologies are not being provided to developers so they can build security into their processes.

Introduction

Recent statistics about automobile safety are disconcerting, both for the automotive supply chain and for the consumer. Media attention has highlighted numerous security and safety issues for the connected car. But as sensational as some of these headlines are, the problem is real. Nobody wants to stifle innovation, much less slow the consumer's access to the benefits of innovation, but this new technology is moving faster than the government's ability to regulate its use. We are now at a tipping point. Innovation no longer starts and ends with a car's mechanical components; electronic components now make up over 50% of the total manufacturing cost of a car with some cars now containing over 100 million lines of code. This clearly represents a source of worrisome security vulnerabilities.

Estimates are that 60-70% of vehicle recalls are due to software glitches. Cars are run by networks of computers, wireless connections, and electronic control units (ECUs), offering the potential for hackers to access critical car controls including the steering and braking. Modern cars can easily connect to smart devices and the internet. This connectivity potentially exposes critical systems to hackers which could lead to remote attacks on cruise control mechanisms, braking systems, and other safety-critical operations.

It's not just new cars that should be considered vulnerable to hackers. Older cars are increasingly connected using devices that are plugged into vehicle diagnostic ports and linked to smartphones. Once a smartphone is connected to the car's network, both the automobile's fundamental safety and the driver's personal information can be compromised. Knowing this, it's obvious that security and safety should be front and center in automotive software development.

Ponemon Institute, the leading independent security research organization, recently reported feedback from over 500 people directly involved with the development of automotive software, with nearly 80% of the respondents coming from OEMs and Tier One suppliers. The goal of this survey was to gather insights on the state of application security practices in the industry.

This new Ponemon survey, which was sponsored by Security Innovation and Rogue Wave Software, provides new insights to help automotive software suppliers understand the current mindset of their developers and build security and safety into their software.

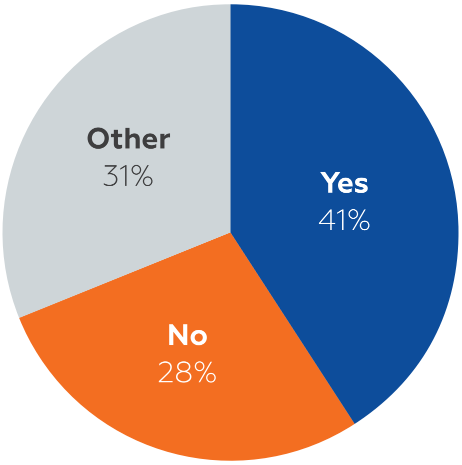
Part 1: What did the survey find about automotive software developers?

Are automakers worried about hackers?

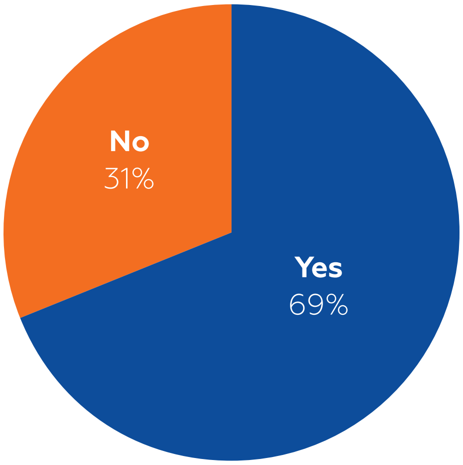
Despite the industry insiders public statements that hackers aren't wasting their time on cars, and that the fear of a car being hacked is simply unfounded, 44% of the developers surveyed believe that hackers are actively targeting automobiles. This could be a positive sign, demonstrating that developers are at least aware of the security and safety vulnerabilities that their code could present, even if the possibility is remote. However, there are plenty of examples of hackers attacking cars directly, so in reality the possibility is not remote. In Montreal, hackers were able to enter cars leaving no visible signs of forced entry, and in California, thieves could buy a $30 kit which allowed them to access many high-end vehicles.

Is security a priority?

Despite the understanding that automobiles are hacking targets, only 41% of developers polled agree (and 28% disagree) that secure software is a priority for their company. Worse, a large number of them (69%) believe that securing the applications are difficult/very difficult and nearly half (48%) believe that a major overhaul of the car's architecture is required to make it more secure. Only 19% think that it is even possible to make a car "nearly hack proof."



"Secure software is a priority for my company."



"I believe that securing the applications needed is difficult/very difficult."

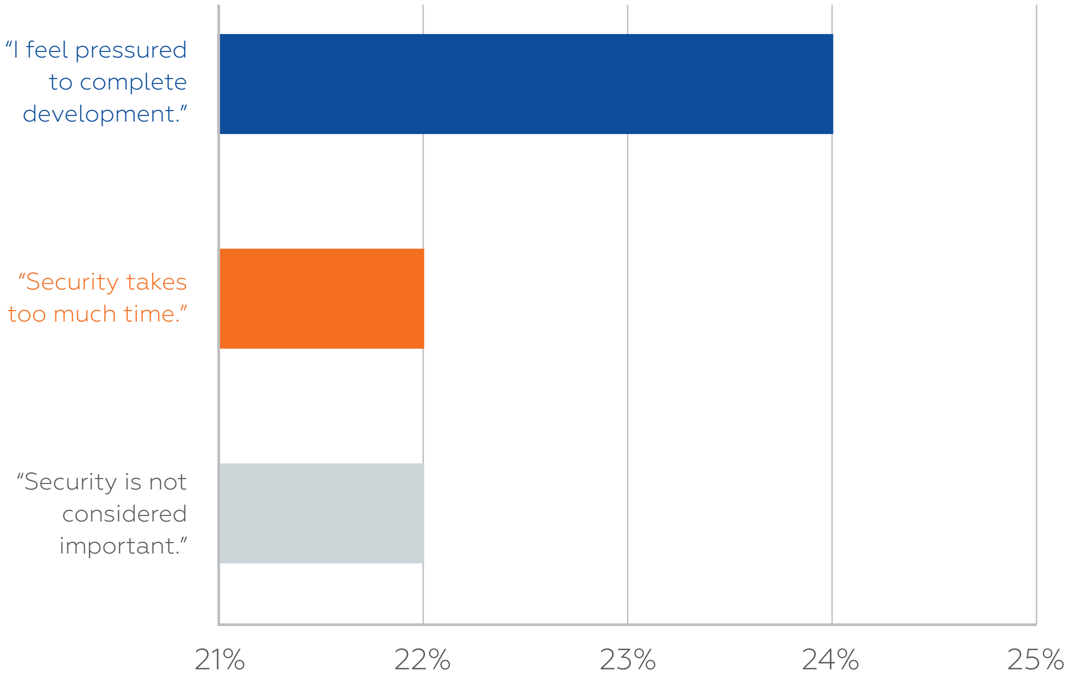
In further interpreting the results, we find that one third of developers are "unsure" whether their own company takes secure software seriously. This is a truly scary statistic. If developers do not believe that secure software development is a priority, what is their incentive to add critical – and what they might believe time-consuming workflows – into their own development cycles?

Why isn't security a priority?

Based on the survey, there is some belief that even if cars are being targeted by hackers it is difficult to stop them, even if their company did place a higher priority on security. This begs the question: Why are companies not putting more emphasis security in their applications? The main reasons they cite are pressure to complete development (24%), security takes too much time (22%), and it's not considered important (22%). These reasons for ignoring secure development practices would indicate that it is imperative that companies formalize security throughout their development lifecycle processes. In fact, over half of the respondents felt that security is not integrated into the development process, but that it is treated an add-on responsibility, usually managed by someone else.

It is clear that individual developers are not aware of their own responsibility, and that a knowledge of security and their role in addressing it should be built into the fiber of the company. This would make security not a nice to have, but a must have. If security is not optional, it is less likely a company will end up in the headlines as the company whose software allowed brakes to be disabled or a cruise control system to stop functioning. Developers must treat security as a design imperative in the same way that they treat quality, functionality, and performance requirements today.

Why are companies not putting more emphasis on security in their applications?



What are the barriers to making security a priority?

Identifying the barriers to achieving security is paramount. To some extent, those on the automotive supply chain have not considered themselves "in the software business" until recently, when well publicized hacks and threats began flooding the media. For this reason, only 28% of those polled believe that automakers are as knowledgeable about secure software development as are other industries. So not only is basic knowledge a barrier, but there is also insufficient training. Less than half of the respondents believe they are adequately trained in secure architecture and coding practices. The good news is they acknowledge the gaps; but it's the responsibility of companies to bridge those gaps in order to avoid life-threatening security failures.

But how can we bridge this gap when 47% don't believe that making an automobile "nearly hack proof" is even possible? This re-enforces the need for companies to accelerate secure development training and create awareness across their organization. Building secure code is possible, given the right processes, tools, and policies.

What are companies currently doing?

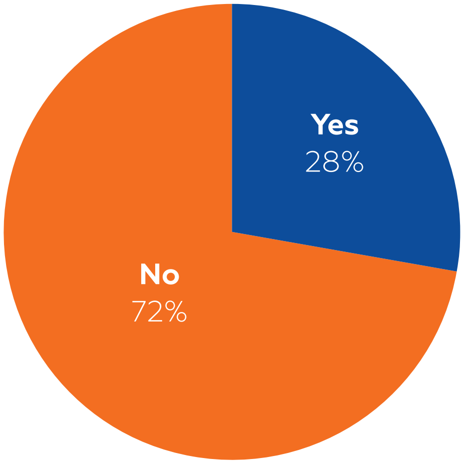
Despite what we've learned about basic coding practices in regards to security, it is important to recognize that, based on our research, companies are not simply sitting back and ignoring the problem. The lack of knowledge on how to move forward doesn't mean that automakers aren't doing anything to secure their applications. Nearly two-thirds (63%) are running automated software scans during development, half are running scans after the application has been released, and slightly more than a third (36%) are conducting penetration tests. But only a quarter of those surveyed say they are adhering to secure coding standards or conducting high-level assessments, such as threat models. They also realize that a change in their software development processes as well as specially trained staff will be required to improve security in automobiles. The key will be for the companies to begin operating and behaving like software companies, using the proven and tested methods of others to avoid recreating the wheel, but at the same time making a major shift in their standard operating procedures.

Learning from other industries

In the survey, Ponemon found that developers and their organizations are also not following other industries' example in working with white hat hackers to discover bugs through programs such as bug bounties. Surprisingly, 43% felt that white hat hackers should be subject to the Digital Millennium Copyright Act (DMCA), which means that hackers could be potentially arrested for experimenting on automotive application code. Of the 42% that believe white hat hackers shouldn't be subject to the DMCA, 54% of these said they shouldn't be encouraged to test car software.

Only 28% of the automakers believe that they are as knowledgeable as other industries with respect to security, but as we mentioned before, they don't appear to be serious about understanding and adapting best practices from other industries.

"I believe that automakers are as knowledgeable about secure software development as are other industries."



The IT world has been struggling with cybersecurity for well over a decade, with varying levels of success, but they have paved the way. The automotive industry does have precedents to lean on rather than re-inventing secure development techniques, should they choose to use the lessons learned from their peers in other industries. The inherent challenge to this IoT problem is, how do manufacturers ensure that the software, deeply embedded and intertwined with their hardware, is secure, even when there are countless potential consumer environments that their product could end up in?

Clearly, the software development processes of old, one-to-one testing systematically, are not feasible, whether you're talking about mobile phones or the connected car of today. The amount of testing that would need to take place, when taking every operating system and hardware possibility into consideration, would be virtually impossible.

Best practice is for companies to provide their engineers with processes and tools that address security throughout the software development life cycle, mitigating security risks up front, well before code is deployed. Benefits are not limited to security improvement though; they extended to speedier development processes and simplified adherence to industry and government-imposed standards. The Automotive industry is at the right juncture now to take a page from those that have gone before them, to begin standardizing processes and procedures to ensure more secure code development.

Part 2: What should companies be doing when it comes to security in automotive?

Measuring gaps

It is important to identify the gaps in an organization so management and individual developers can address the vulnerabilities that lead to security breaches. What the Ponemon survey indicates is a basic lack of understanding of how to secure code – not whether they should or not. There are some basic misunderstandings about how simple process and policy changes can make a big difference. For instance, 18% of respondents indicated that their biggest concern was non-compliance with industry standards, though the majority of those same respondents said they were very worried about vulnerability to hackers and safety of the vehicles they were helping build. This is interesting, since industry standards are inherently designed to mitigate those issues. It would be reasonable to say that management should be providing development tools that build standards-checking into their processes. Using the latest tools available means that developers spend less time staying up to speed with standards, and provides management with the peace of mind that security is simply built-in.

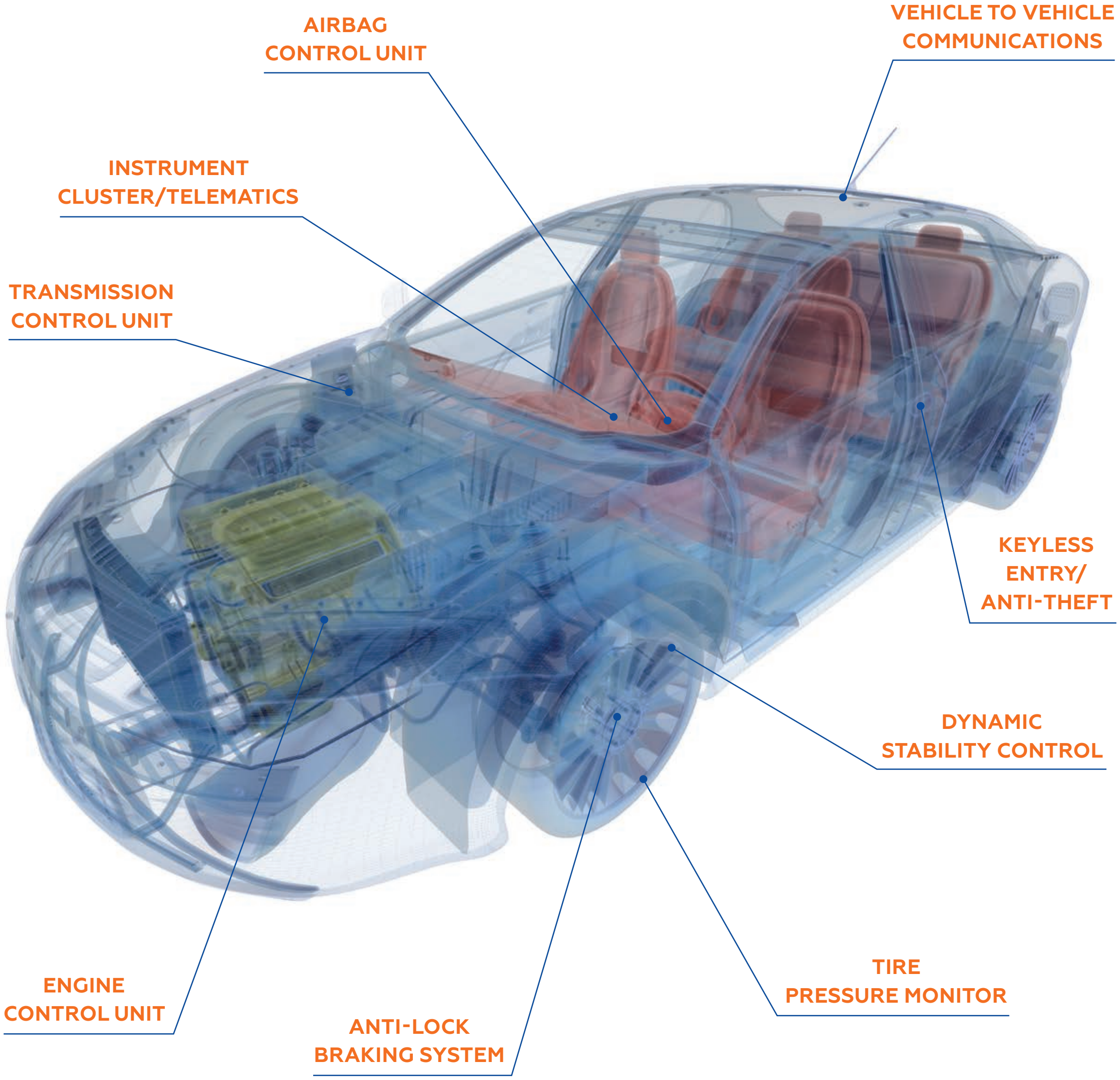
Empowering development teams

Fundamentally, companies should be making it easier for developers to create secure code. Companies need their developers to focus on innovation, and not tedious processes for ensuring security. Providing processes, policies, and tools makes security non-negotiable, and more than an afterthought. Security should be infused in every step of the SDLC, but only 45% of developers indicate that their development processes include any activity that supports security requirements. How can we expect individual contributors to take responsibility for a moving target, or a target that is not even identified? In fact, the majority of our respondents believed the CIO was the person most responsible for security in their organization – but that role rarely if ever even touches the code. When asked to select what one thing they thought should be implemented to make automobiles more secure, developers chose almost equally, change in software development processes and specially trained staff.

Conclusion

Connecting cars to the internet brings a host of new convenience, entertainment and safety features to consumers, but this connectivity also opens up cars to remote hackers. Automakers and their suppliers are struggling to adapt their hardware and software architectures, their code and their processes to deal with this new threat to automobiles. Contrary to public statements by the automakers, the Ponemon survey shows that OEMs and their suppliers do not yet have the desire, skills, tools or processes to make a secure car.

Vehicle Attack Surfaces


AIRBAG CONTROL UNIT
VEHICLE TO VEHICLE COMMUNICATIONS
INSTRUMENT CLUSTER/TELEMATICS
TRANSMISSION CONTROL UNIT
KEYLESS ENTRY/ ANTI-THEFT
DYNAMIC STABILITY CONTROL
TIRE PRESSURE MONITOR
ENGINE CONTROL UNIT
ANTI-LOCK BRAKING SYSTEM

RogueWave
SOFTWARE

Accelerating Great Code

- Find security vulnerabilities as code is being written
- Find risks with run-time debugging
- Identify malware or spyware with smart, predictive analytics
- Ensure open source use is protected

WWW.ROGUEWAVE.COM

Security
Innovation®

Helping Organizations Secure Software Wherever It Runs

- Software security testing (Pen testing, SDL Gap Analysis, Code Reviews, etc.)
- Security Training (Computer Based and Instructor led)
- Embedded Systems Security (Encryption, Communications Security)

WWW.SECURITYINNOVATION.COM