The Intelligent Transportation Society of America (ITS America)

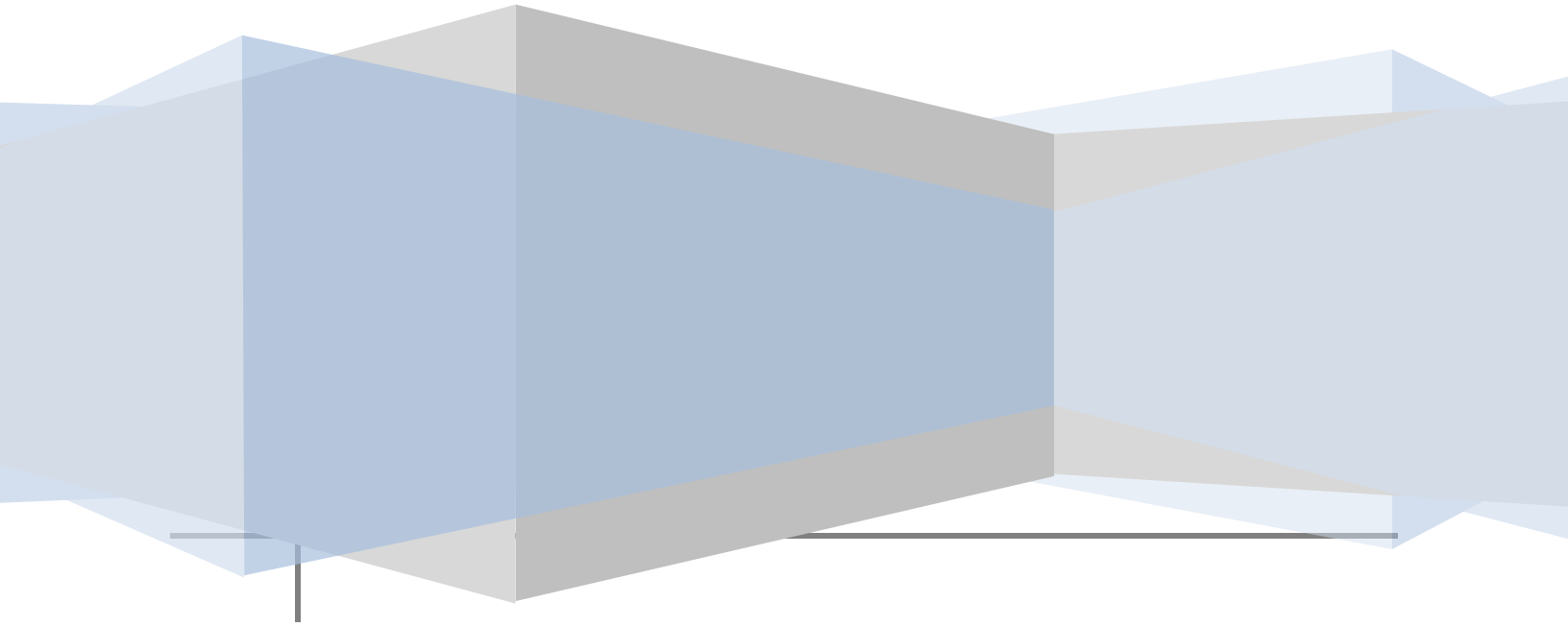
Connected Vehicle Assessment

Cybersecurity and Dependable Transportation

System Assurance, Operations and Reactive Defense for Next Generation Vehicles, Intelligent Highway Infrastructure, and Road User Services (Version 2)

Steven H. Bayless, Sean Murphy, Anthony Shaw

Technology Scan Series 2011-2014



Contents

Introduction

There are approximately five billion "machine" devices connected to the internet, and it is estimated that the "internet of things" will reach nearly 50 billion online connected machines, sensors and appliances by 2020. Likely more than a billion connected M2M (machine-to-machine) devices will be in highway transportation, of which more than half will be road vehicles.

Securing M2M applications, and preventing mischief and mischance, will be major tasks for the transportation sector. The connectivity of billions of new vehicle- or infrastructure-based sensors and "big data" analytics promises to bring new insights into how transportation assets are currently utilized, and how safety and mobility outcomes might be improved. However, there are concerns about the security of these devices and the potential for compromise and misuse.

The growth of M2M in automotive will likely be more rapid than anticipated, as mobile devices will be "aftermarketed" by technology companies onto the end of the automotive supply chain. The expectations of consumers and businesses will result in the introduction of BYOD (Bring Your Own Device) to vehicles, since road users will want to take advantage of useful mobility, logistics, and infotainment "apps" and services. Traffic safety advocates and some auto manufacturers may even cautiously encourage this trend, as they would prefer that mobile devices, absent an effective ability to prohibit or circumscribe their use in cars, be at least managed in a way that eliminates the risk of driver distraction.

Transportation is not only becoming more connected, but more and more dependent on complex computing systems and software. Current automotive electronics and highway Advanced Traffic Management Systems (ATMS) and their component field devices (e.g. traffic signals, ramp meters, roadside sensors, and dynamic message signs), include a number of computing hosts, electronic microcontrollers, and application software that are internally interfaced and networked to each other inside products, but are generally isolated from the rest of the connected world. Security controls for these kinds of systems, in particular for older legacy technologies, typically focused on the simple necessity of reducing the risk of physical product tampering or theft. However, the security and threat environment is beginning to shift around transportation systems as they become externally connected to the wider world.

The root of most technological innovation, and the source most of our vulnerability, is software found in applications, devices and networks. The unintended introduction of design flaws and coding bugs are an inevitable part of the software development process for complex technologies. The demands for more software functionality have outstripped the capacity for software engineers to design systems with any degree of assurance that the software will be reliable or secure. This is especially true for "extensible" software, where a system is designed to avoid early obsolescence by inclusion of mechanisms for expanding/enhancing the system with anticipated capabilities without having to make major changes to the system infrastructure.

The core challenge in cybersecurity is to establish trust. Trust means confidence that software is reliable and free from potential security vulnerabilities, and that it can establish connections with other systems that will not compromise its critical resources or functions. Software vulnerabilities are dangers, and countermeasures must be

applied, or "patches" must developed, tested, distributed, and installed, which is a costly, complex, and burdensome process for government and industry. The best long term strategy is to reduce cyber-vulnerabilities from being introduced in the first place in the early stages of product development. Security over the long term requires more time and investment in upstream software development activities such as requirements definition, design and architecture development to reduce system complexity, and improve software quality.

Applying and maintaining cyber-defenses is typically the responsibility of the end user. In the highway transportation sector, the end user is typically the road operator, automobile manufacturer, fleet manager, or dealer/auto mechanic. Road infrastructure operators (e.g. state and local departments of transportation) and vehicle manufacturers are typically are often driven by much higher standards of assurance in order to preserve safety, security, and privacy. Intelligent transportation systems, like other industrial control technologies (also known under many monikers such as OT, or Operational Technology, "cyber-physical" systems, or Supervisory Control and Data Acquisition Systems [SCADA]), have generally been technically distant from the rest of the larger information technology ecosystem. While most enterprises have adopted commodity computing platforms, common networking standards, software development tools to minimize costs, vehicle and traffic management systems have for the most part maintained costly legacy architectures built upon decades of sector-specific experience, practice and product development. These systems were assumed to be too obscure to garner any unwanted attention from potential adversaries, or were "air-gapped" by design – that is, they were completely disconnected from other networks.

However, the days of "security through obscurity" have likely passed, as transportation systems become more connected. The same business demands that pushes consumer and enterprise IT systems to support greater end-user connectivity, service extensibility and scalability through "cloud" architectures, have gained a foothold in the management of vehicle and highway intelligent transportation systems. Automotive manufacturers, fleet managers and road operators see huge potential efficiencies in increasing centralized remote automated monitoring and control of systems to reduce maintenance or operations costs or increase road user convenience. Second, there is an irresistible desire for technology firms to "aftermarket" devices (PCs, tablets and mobile phones, or other "tethered" peripherals such as vehicle On-Board Diagnostic dongles) onto control systems, without necessarily bringing these devices entirely within a managed network where they can be monitored and secured from potential compromise. Aftermarketed devices and other peripherals therefore threaten to bypass air-gaps, firewalls, and other network perimeter defenses and increases the "attack surface" of legacy vehicle and traffic management systems.

Second, the incentives to attack these systems have increased, as the "black hat" hacker (more appropriately "cracker") culture now possesses the playfulness, motivation, and resourcefulness to reverse-engineer these obscure proprietary systems simply for the notoriety (at best), or to further a particular political agenda or criminal enterprise (at worst). The first bona-fide attack on an enterprise industrial control system occurred when foreign hackers caused a pump at an Illinois water treatment plant to fail in 2011. In 2012, several computer science research centers conducted surveys of the cyber attack surface of conventional light passenger vehicles, and conducted experimental "proof-of-concept" attacks. There are several documented attacks on transit and road operators, such as the hacking of websites and in-the-field traffic control devices, limited primarily to hacks to roadside Dynamic Message Signs (DMS) where there is likely very little impact on public safety. There is greater

concern that the as the sophistication of attacks grow, the costs of securing, and ultimately, deploying Intelligent Transportation Systems (ITS) will grow disproportionately and potentially choke-off innovation.

Consumer and enterprise IT companies develop software under overly aggressive time frames to establish "first-to-market" competitive advantage, often with a "fix-it later mentality" with respect to reliability and security. Most innovations that appear in IT typically take five years or longer to appear in transportation, as new practices and technologies mature. This paper suggests that the slowness is not because of any lack of innovation or sophistication on the part of vehicle manufacturers or road operators, but a more measured and realistic assessment of the risks of adding connected services. This lag in innovation takes into account that the fact that automotive software components and advanced traffic management systems are much more complex to design, and costly to update. Manufacturer responsibility for reliability and liability for defects is much more concrete and transparent than in the case of IT products.

Conventional security management in automotive has been focused on theft prevention; however, as vehicles become "always online" with advanced telematics, security assurances will become more demanding and the potential attack surface will grow. Next-generation vehicles will also likely feature Dedicated Short Range Communications based Vehicle-to-X cooperative crash avoidance applications ("X" being vehicle, infrastructure, and mobile device) as envisioned in the US Department of Transportation's Connected Vehicle program. Vehicle-to-X communications will greatly increase the scope and reliability of future crash avoidance and driving automation systems. Connected Vehicle program includes a "security credential management system" designed to guarantee vehicle-to-vehicle/infrastructure crash avoidance applications are able to maintain the integrity and anonymity of safety data being broadcast from other equipped vehicles or traffic signals. The National Highway Traffic Safety Administration intends to make a decision about the deployment of Vehicle-to-Vehicle communications base crash avoidance application in 2014.

Current trends suggest that future transportation products will use more software and will be more complex and connected than they are today. Because of this complexity and connectivity, and the continual need to expand functionality and maintain reliability and security, products will begin more and more to resemble services. As mobile device manufacturers have begun to provide ongoing software maintenance and security to increase the reliability of their products, smart cars and smart infrastructure, if security is to become a priority, will need guarantee that software and new technologies systems are maintained to ensure security.

Cybersecurity in intelligent transportation requires the same conventional "defense-in-depth" strategy that IT systems must employ to maintain security. A three-fold approach employs *constructive*, *operational*, and *reactive* cyber-defense strategies. *Constructive Strategies* require a lifecycle approach to software development that seeks to reduce likelihood of vulnerabilities being introduced in product development phase, before systems are deployed to end users. *Operational Strategies* focus on continual *remediation* -- patching vulnerabilities quickly and implementing and maintaining strong authentication and access control, while applying *mitigation* -- proactively shrinking systems' total attack surface by reducing exposure of critical elements. Finally, *Reactive Strategies* assume the inevitability of compromise and instead seek to actively out-smart and shut-out attacking adversaries through real time operational intelligence and analytics. The combination of *constructive*, *operative* and *reactive* strategies over the long term can create "defense in depth," where overall security can be achieved in scales greater than the sum of their parts.

Landscape of Current Security Vulnerabilities and Threats

Long term vision requires a comprehensive and sober assessment of the threats to safety and mobility systems in transportation. Although there are no instances in highway transportation where cyberattacks resulted in a system failure that lead to a death or injury, the specter of such outcomes is frightening. The potential objectives that may drive attackers to exploit vulnerabilities and compromise transportation systems may vary widely, but the overall outlines for attack are conceivable in most transportation products and services. These so-called "threat models" or threat assumptions can be distinguished by two broad categories in any risk assessment – whether a system is *non-safety critical* (which is primarily mobility applications) or *safety and operational critical*.

Generally speaking there are three types of attacks on non-safety-critical categories: *Financial, Privacy*, and *Operational* that cover both products and services. In order of relative severity, they include but are not limited to compromise of products or services/operations such as 1) initiate related unauthorized financial transactions (e.g. transportation fees, tolling) manipulation of driver/vehicle/freight telemetry measurements (e.g., asset tampering and misappropriation , cargo theft, or fraud where transactions are based on telemetry such as tachometer) ; or 2) compromise privacy through surveillance and 3) manipulation and disruption of logistics and mobility operations (e.g. unauthorized tampering with a fleet dispatch system, changing a of Dynamic Message Signs, or activation of emergency vehicle traffic signal pre-emption etc..) Non-safety critical systems threat models based on adversary motives to attack primarily for criminal gain or some other reward.

Safety and operational critical systems have a different threat model focused less on motive and more on the size of impact and immediacy of potential successful exploition. Compromise of transportation products include 4) immobilization, theft, damage to transportation assets such as traffic control devices, vehicles or freight, especially in critical moments, 5) manipulation or disruption of ancillary safety features (e.g. disconnecting a vehicle occupant protection or crash avoidance features), 6) manipulation or disruption to systems may reduce driver awareness or controllability and increases crash risks to drivers, passengers, or other road users (e.g. sabotaging vehicle braking or allowing a traffic signal assign right-of ways to conflicting flows of intersection traffic).

There are three kinds of technology systems: *Information Technology* systems, *Operation Technology* systems, and *Networking and Communications* systems. IT security focuses primarily on information services. IT security is primarily information assurance and is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording, or destruction. IT security ensures the confidentiality, integrity, and availability of information, and is rooted in a given organization's operational critical assurance objectives, management rules, and security policy. "Confidentiality" refers to preventing the disclosure of information to unauthorized individuals or systems, while "data integrity" means maintaining and assuring the accuracy and consistency of data over its entire lifecycle. Availability means for any information system to serve its purpose, the information must be available when it is needed. IT systems typically emphasize confidentiality and integrity over availability.

IT services typically rely upon commercial "off-the-shelf" or "open source" hardware and software, and industry *de facto* standards for computing and networking. Off-the-shelf software does not provide immediate value out-of-the-box, but must be properly implemented and tailored to fit business or organizational processes.

Implementation of security is costly and time consuming component of implementing IT systems. IT are designed to be extensible through add-ons and application programming interfaces have larger attack surface out-of-the-box and must be painstakingly configured and maintained to reduce remote access of critical resources and data from adversaries. Software development lifecycle for these systems are typically very short, as the lifetime of a system lasts about three to five years. The architecture often features massively networked, distributed and virtualized computing assets, with clients (e.g. PCs and mobile devices) and host applications both storing data and supporting processing.

"Operational Technology" is more single product- or system-oriented, is more expensive to engineer, and has a longer lifespan, typically beyond a decade. Programmable control systems were designed in the 1970s, and distributed programmable systems evolved in the 1980s. Examples of OT systems in transportation include automotive electronics and traffic management systems such as traffic signal controllers, whose legacies are rooted in 1980's architectures. Although computing assets are distributed (e.g. programmable logic controllers and sensors) they are often self-contained within a single product or system and have no, or very few, interfaces to other OT, IT or networking and telecommunications systems.

Next generation OT systems, describe often as the "Internet of Things" or Machine-to-Machine (M2M) applications, however, are a remake of legacy self-contained OT architecture to take advantage of outside computing resources in the "cloud." Developers of M2M applications will aspire to use standardized interfaces, communications protocols and service oriented architectures to connect traditional consumer and business appliances and industrial sensors to "cloud" computing platforms to support online services and remote device maintenance. Vehicles with next generation telematics and advanced traffic signal controllers suggest that closed, legacy OT systems are progressively welded together with newer open M2M/cloud-based services.

Whereas most IT systems are "best effort" in terms of availability, OT systems that manage physical processes must have computing resources dedicated for real-time processing. For example, cyber-physical processes include braking of a vehicle or the changing of a traffic signal phase, which cannot wait for computing resources to become available. OT systems security focuses therefore on maintaining availability and integrity, and these system protection measures are emphasized over confidentiality. OT systems feature real-time operating systems (ROTS) that are not optimized to share computing resources with a large number of applications, as conventional IT platforms do. They are instead designed to execute time-sensitive tasks under strict deadlines.

OT architectures can be centralized like IT systems (e.g. servers with distributed clients), but more often rely upon a local internal network of peered distributed Programmable Logic Controllers (PLCs) with customized networking protocols and software (i.e. non-commercially available software, or "firmware") that serve only one purpose. (For example, windshield wipers are controlled by a single PLC module, also known as an Electronic Control Unit, rather than a centralized general computer.) Reliability and safety requirements are typically much higher for OT than information services. Whereas information services might fail to process a particular transaction and can recover from such a failure by retrying (e.g. a payment) or rolling back a transaction (e.g. from a backup or cache), OT systems can neither retry nor recover a process (e.g. a vehicle braking command) from a failure. Most OT in transportation is safety and operational critical, where failures cannot be tolerated.

Networking and communications systems include conventional wireline or wireless networks built upon open standard internet protocols, or may include so-called closed/legacy systems such as Global Standard for Mobile

(GSM) communications. The integrity, availability, and confidentiality of data, however, still must be ensured while data is in motion over networks.

Much security in these systems is built to support application level connectivity. For host computers, authentication and access control are often the center of gravity for security assurance, and are required to support inter-networking. Hosts need to authenticate the identity of the other party before transmitting data to prevent the disclosure of data, or the inadvertent provisioning of secured computing resources to a potential adversary. Authentication is typically supported by building a "web of trust" between computer hosts, or a hierarchical public-key infrastructure with digital certificates that allow hosts to authenticate the identity of partners with which it wishes to communicate through a trusted third party. Authentication, data confidentiality, and data integrity (Data-in-motion encryption) strategies prevent eavesdropping or alteration of data when in transit between hosts. There are a number of session level and higher layer protocols (e.g. TLS, IPsec) that are designed to support web access, virtual private networking, file transfer, and remote host management.

Information Technology Vulnerabilities

Most risk to security is associated with commercially available third party IT software designed for enterprises and consumers. According to IBM, there were 8,168 publicly disclosed vulnerabilities in 2012 in IT systems, 42% of which (or 3,346) were exploited. Of the publicly disclosed vulnerabilities, only 350 were in mobile software (e.g. Android, iOS, etc..) Most of the popular concern related to cybersecurity today is related to the never ending discovery of vulnerabilities in commercially available mainstream IT products and systems, and the disruption attacks exploiting these vulnerabilities cause to commerce and privacy. Most non-safety critical transportation services, mobility applications in particular, rely upon third party "off-the-shelf" or "open source."

In general, the "attack surface" of a IT services is the aggregate of all known and latent vulnerabilities taking into the consideration all of the security controls already put in place across all subsystems and networks. Security officers often seek to reduce the attack surface by either reducing vulnerabilities through software maintenance (i.e. updates, patches) or configuration (turning off insecure features), or alternatively by adding additional security controls. (e.g. encrypting data or adding redundant access controls).

To exploit these vulnerabilities, there are several attack vector categories. The "vector" of an attack is the main means by which the adversary reaches its target. Security incidents often combine vectors to undermine commonly deployed hardware (devices), software (operating systems and applications), and networks. Broad categories of attack vectors vary, but the three broad categories are *Unauthorized Access*, *Malicious Code*, and *Reconnaissance and Networking-based service* attacks. These attack vectors can target applications, host and client operating systems, and even networking equipment.

Unauthorized Access attack vectors weaken security protections such as access control or manipulate authentication between communication partners to undermine confidentiality. Penetration of systems to achieve privileged access to computing resources and data is typically the goal. *Malicious code* vectors vary based on the target, delivery, and propagation mechanisms, and include code injection, Trojan programs, viruses, and worms

and other techniques to compromise the confidentiality or integrity of data or to establish a toehold into systems. *Reconnaissance and Networking-based service* vectors seek to use network facilities to discover exploitable software vulnerabilities for other attack vectors or to reduce the availability of host computing or networking resources through denial of service attacks.

Combinations of vectors are often employed to achieve an adversary's goal of compromising a given system. Attacks can be any combination of direct and indirect approaches. Direct attacks attempt to undermine security policies such as access control (i.e. the compromise of security protocol through an implementation attack). Indirect attacks attempt to undermine a system beyond the scope of a security system by tricking legitimate users, either into revealing security credentials such as passwords, or into downloading and executing malicious code. (also known as a social engineering).

Publically unknown vulnerabilities, and the corresponding attack vectors that are hoarded by the most sophisticated adversaries, are held as privileged information, and they are used sparingly to penetrate networks and applications. These are known as *zero-day exploits*, with "zero" equaling the number of days of awareness of the threat. Zero-day exploits are "open wounds" that cannot be closed quickly or easily, and are often therefore persistent. .. These advanced threats are "persistent" in that the average vulnerability window of a *zero-day* exploit is ten months. (A vulnerability window is the time between when an exploit is discovered and the time a remediation or mitigation strategy is developed and finally applied.) Even if the initial zero-day exploits are found in targeted attacks and security patches are developed to plug critical vulnerabilities, the exploits are eventually incorporated into hacker tool kits, which are sold in underground markets to other hackers or cybercriminals. Adversaries using these tool kits then conduct reconnaissance for systems that have not yet been patched, and then conduct attacks against remaining insecure systems.

Client and host computer operating systems and applications have large numbers of vulnerabilities. For the most part, however, not all software vulnerabilities are created equal, and they do not all pose the same risk. Based on data from 40 million security scans, the cloud security company Qualys found that just ten percent of vulnerabilities are responsible for 90% of all cybersecurity exposures. Most vulnerabilities are low-level and entirely preventable bugs, such as the "buffer overflow error," where user input is not validated and system administrator-level commands can be surreptitiously inserted (i.e. command/code injection) into the memory buffer and run, allowing privileged access to computing resources.

Messaging and other onlineapplications have also improved the reach of potential adversaries. Distribution and installation of malware on these devices has been greatly enabled by the use of web browser exploit kits (mostly targeting cross-device/operating system Java Virtual Machine software vulnerabilities), which are designed to infect the highest number of systems possible, not necessarily target individual users. Hosted web services present vulnerabilities, especially through cross-site scripting (i.e. injecting applets into a third party's website that redirects the website visitors' submitted data to the attacker) or through injection of code into databases (SQL injection to remotely modify data in an online database without authorization).

The root cause of the most egregious IT vulnerabilities is usually poor security choices (tradeoffs between security and functionality) or software development processes (e.g. requirements, design, coding and testing) that introduce bugs that may be exploitable. End users of third party products are typically left unaware of the problems until they surface in a publicized attack or patched in a maintenance update. Failing to apply patches and

maintain IT software by organizations deploying affected software are typically the greatest contributing factors to most cyber-attacks.

IT software security is slowly improving, but progress is uneven. Personal computing operating systems are generally the most common and vulnerable targets and have not benefited as quickly from better software development practices, such as the *Secure Software Development Lifecycle Process* (Secure SDLC). Secure SDLC, encompasses all of the steps that an organization follows when it develops software tools or applications. Vendors that incorporate security in the software development lifecycle benefit from products and applications that are "secure-by-design," which is relatively effective in eliminating the more common and preventable vulnerabilities (Examples include misconfiguration, kernel flaws, buffer overflows, insufficient input validation, symbolic links/file descriptor, and other types of low level bugs). Even though personal computers have fallen behind, mobile device operating systems that support mobile phones or tablets, however, generally have benefited from Secure SDLC.

There was a 32% increase in the number of documented vulnerabilities for mobile device operating systems such as Apple iOS and Google Android in 2012 and, not surprisingly, a 58% increase in mobile malware installed by end users, according the most recent annual Internet Security Threat Report from Symantec. However, IBM has suggested that most of the vulnerabilities in mobile device operating systems may be far more limited than once thought, and that mobile devices would fare better than traditional computing devices, such as PC operating systems, that suffer from dependence on legacy design and software code.

Time will likely tell whether mobile operating systems may be more secure, especially for mobile devices. Applications, in particular for portable mobile devices, may represent unique and growing risks in IT. First mobile devices (much like vehicles) are not under constant physical lock-and-key and can be easily lost or stolen. Stolen devices may enable attackers physical access to compromise systems that otherwise would have required more difficult efforts to establish remote access to exploit. (Physical access also provides adversaries with another critical resource, which is time to complete their attack before it is discovered and remediated.) Mobile devices are special cases and by virtue of their portability can be compromised "network" attack vectors themselves. Most portable mobile devices are often connect to untrusted networks, bypassing perimeter defenses such as firewalls that are often set up by enterprises and other organizations to monitor and mediate remote access. A compromised mobile device can use wireline connection via Universal Serial Bus (USB), or wireless local area network connection to deliver malicious code to a host.

Second, sensitive organizational or enterprise data is either stored or is otherwise accessible from mobile devices. Mobile applications often log location data (using GPS, Wifi, or cell tower positioning) which may be stored insecurely. Freeware or "grayware" that monitors user location does not aim to harm users, and often provides real functionality and considerable value to the users. For example, traffic information and navigation services are often provided at no charge or minimal charge, in exchange for collecting location, speed, and other information that is used to improve the quality and coverage of the service provided. However, if the freeware or grayware's has no security policy or security controls, then sensitive road user location and other data may be compromised, or the online service itself may be manipulated (e.g. showing traffic congestion where none may exist). If data is transferred to a third party host that implements less stringent operational security controls, then the data may be further vulnerable to exfiltration or unintentional disclosure.

Most IT software vendors do not comprehensively employ constructive security strategies such as secure SDLC, which tends to shift burden of security to the end user. Commercially available IT security products and services such as anti-virus help the end user bare this burden; typically focused along a single and difficult to sustain line of defense—through *operational* controls focused on *remediation* or *mitigation*. *Remediation* focuses on patching vulnerabilities in systems, if software updates are made available. Where remediation is not possible, *mitigation* removes a vulnerable system or service from operation, whitelisting or enforcing a list of permissible connections, or other actions to reduce the attack surface. Securing large numbers of applications, host/client systems and services, local area networks and network perimeters has been for the last several decades one of the most complicated and costly operational challenges in IT.

Networking and Communications System Vulnerabilities

Beyond the network perimeter of most organizations is the public switch network and other telecommunications infrastructure. While most cyberattacks are against IT systems and focus on device, host and application level security, networks and communications systems can also be vulnerable. The security objectives of an application in reference to connectivity usually focus on authentication of communication partners, and ensuring the confidentiality and integrity of data payloads as they traverse over multiple unknown and untrusted networks.

Attacks can occur at a number of different layers, from physical layer (wireless and wireline links) on up to the individual communication sessions, and often defenses are must also layered to ensure the integrity, availability, and confidentiality of data being transmitted over a network or multiple heterogeneous interconnected networks. Some common issues include attacks on security protocols, inadequate *authentication mechanisms, telecommunications carrier "insider" threats,* and *denial of service*.

Authentication is important because parties to communication must also trust that they are speaking to a legitimate counterparty, rather than a third party that may be masquerading as the communication partner. Data transmitted can be captured by a third party that situates itself between two parties in the communications media (e.g. wiretap). Data therefore must be encrypted, and the security protocols implementing encryption must be robust to attack. Furthermore, special attention must be paid attention to wireless networks, as they are especially accessible to adversaries and therefore vulnerable.

Wireless communications create a number of problems, and vulnerabilities that exist in these systems are difficult to address. Fixed "wireline" networks benefit from the fact that to intercept a transmission, an adversary needs physical access to the fixed network, which is often difficult to achieve. With regards to wireless networks, an adversary only needs to be in radio range of the wireless network to launch an attack targeting it. "War-driving" is a form of reconnaissance where adversaries drive a car through wireless coverage areas with a laptop and antenna on-board to detect Wi-Fi wireless access points or cell towers and to collect configuration information in order to search for vulnerabilities. Network or protocol analyzers' packet "sniffing" and capturing utilities are common tools for such attack pretexts. Competitions at past Black Hat security conferences have shown that some war-driving exploits can detect wireless data up to 45 miles away.

The security of wireless systems has been compromised and frequently vulnerable systems still continue to be part of the telecommunications infrastructure. For example, wireless systems, such as Wi-Fi using *Wired Equivalent Privacy* (WEP) and *Second Generation (2G) GSM cellular* were both compromised because the encryption protocol was either cracked, as was the case with Wi-Fi WEP, or because an adversary took advantage of an authentication security design flaw in GSM. Generally, vulnerabilities in telecommunications infrastructure are persistent long after they have been discovered. Vulnerabilities in wireless wide-area communications (cellular) networks cannot be easily "patched," unlike application-level vulnerabilities, and often require extensive modifications to infrastructure, usually at significant cost.

Local and personal area wireless systems, such as Wi-Fi or Bluetooth, demonstrate the persistence of vulnerabilities in communications technologies. Where flaws such as WEP encryption are discovered, users may not be aware of the need to update their equipment to repair, or if they are aware, they may not know how to configure updates. Further complicating matters, occasionally security updates are not "backwards compatible" with older hardware or equipment. WEP was discovered to be vulnerable in 2001, and by mid-2000s, at least one retailer had almost a million credit and debit card numbers stolen and sold to fraudsters through a war-driving WEP attack at a remote retail store location.

The authentication flaw in 2G was discovered more than a decade ago, yet most of the United States is still covered by 2G and users of cellular networks are still vulnerable. The design flaw in GSM 2G networks is that the system was designed to authenticate users to the network, but does not required network to authenticate its identity to the mobile user, thus allowing an attacker to "impersonate" a cellular carrier. The design flaw allows an attacker to establish a "rogue" 2G base station and posed as a wireless mobile network operator in a classic "man-in-the-middle" or "network impersonation" attack. (Incidentally, because the 2G network has the largest coverage footprint, nearly all telematics systems or machine-to-machine devices employ 2G modems, mostly because of its coverage in both urban and rural areas.) Most mobile network operators in the U.S. have suggested sun-setting 2G and replacing it with more secure 3G and 4G coverage by 2017 or some time thereafter. However, 2G networks, and their inherent security vulnerabilities, will persist for possibly a decade until wireless carriers complete all upgrades nationwide to their cellular infrastructure.

Rarer threats, but still significant, are telecommunications *insiders*. Cellular networks connect 2.6 billion users globally, which is more than double the number of wireline internet users, and which represents a target-rich environment for hackers. Cellular systems have a lot of "stateful" equipment that exposes information about user profiles, locations, and applications used mostly for the purposes of billing and network management. Employees of mobile network operators have sold subscriber information, such as *International mobile Subscriber Identity* (IMSI) numbers, to target individuals for espionage and surveillance (e.g. discovering privileged information or monitoring location in process of committing fraud or other unethical behavior). "Insider threats," such as unscrupulous foreign country roaming partners of US mobile network operators, have been able compromise the privacy of US users by giving up US subscriber information needed to support international cellular roaming.

Denial of Service (DOS) attacks is another persistent, structural vulnerability. Cyber criminals are becoming organized and profit driven; however, some adversaries and "hacktivists" seek to use networks to disrupt applications for political purposes. Denial of Service (DoS) attackers primarily attempt to make systems unavailable, often by tying up vital communications, networking, or computing resources. DoS is a form of

"internet street protest" that takes up the capacity of the network in the same way a mob might congest streets in a city. Although jamming wireless communications at the physical layer (radio spectrum) is difficult, attacks utilizing internet networking protocols (e.g. protocol floods using forged sender addresses) are less risky because of anonymity the internet affords and the difficulty to attribute attacks to specific systems or adversaries. DoS attackers, who often hijack other host computers (i.e. "Zombies" or "Botnets") to conduct distributed denial of service, which make it extremely difficult to identify adversaries. (Bots have even been found in embedded systems, and an attack in early 2014 showed that a 100,000 devices were infected and commandeered as "ThingBots.") Denial of service may overwhelm servers and shut down applications, but the additional traffic traversing critical communication links may deny access other hosted IT services. According to IBM, the risk of DoS attacks is an average of 12 hours per year, with extreme outages at nearly 24 hours. DoS attacks exceeding 20G bps, which will overwhelm almost any online application service's bandwidth, more than quadrupled in 2013, according to Arbor networks.

Denial of service is also a risk with cellular systems, smart mobile phones and M2M. Smartphone and M2M terminal "apps" make continuous queries to the network to access cloud applications and require constant synchronization with the network. More terminals have led to a significant increase in "signaling" traffic in both 3G and 4G networks. Signaling traffic sets up communications sessions, authtenticates users and devices and manages mobility across coverage areas. Signaling is significant, as a million connected smart phone users can generate 31,000 transactions per second during peak hours on a cellular network. Unexpected signaling spikes, often initiated by poorly configured, high bandwith applications, have been known to overload networks elements to cause large-scale network outages. Hackers have been able to manipulate networks using malware to create such DOS "signaling storms."

Furthermore, DoS has been combined with traditional attack vectors. DoS attacks are often preludes for other attacks. The prelude DoS strategy has been used against high profile institutions such as banks, so that defenders are less vigilant during the main attack, which seeks to establish toeholds inside a critical system. Even wireless denial of service attacks have been combined with other exploits. For example, most cellular users connect to 2G networks, or a combination of 2G and 3G networks, where both generations of technologies might be housed together in a single base station. One common attack is to set up a rogue 2G base station and electronically jam 3G networks, taking advantage of the fact that 3G phones will revert to 2G when 3G networks are unavailable. The adversary then impersonates the telecommunications network operator acting as the "man-in-the-middle."

Since the security of telecommunications infrastructure is uneven and not all network equipment can be trusted, the solution is often to use so-called higher layer solutions, often tailored to the application category, to protect data in transit and ensure authentication during a communications session. The most common strategies are "tunneling" using IPsec or adopting "application level" solutions. Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of a session, and for negotiating which cryptographic keys that will be used during the session. IPsec is used to establish a "virtual private network" that allow a group of hosts widely scattered to act as if they were communicating over a single secured local area network. Where application developers, such as those developing web services or remote access administration tools, want finer control over how security is managed, they often

implement Secure Sockets Layer/Transport Layer Security (SSL/TLS), Secure Shell (SSH), or other higher-level protocols.

Use VPN and higher level security protocols must be managed to ensure security. Over time, as many communications security protocols like earlier versions of SSL have been cracked, organizations have needed to update or reconfigure these cryptographic elements. For instance, it is estimated that 22% of websites still use insecure versions of SSL/TLS. However, update of these protocols is less burdensome than upgrading lower layer networking and telecommunications security elements. Overall, the flaws in networking and communications systems do represent risk to applications, but for the most part, these systems represent less risk in aggregate than host and application vulnerabilities. It is estimated that nearly three-quarters of cyber-attacks occur at the host and application layer.

Security of Road User Mobility Applications

A number of transportation systems rely upon traditional IT systems, and as road users become more dependent upon them, there are greater risks that attacks may disrupt driver, passenger and freight mobility services. Road users include freight carriers, logistics firms, large passenger carriers (public transit operators, urban rail and intercity bus operators), small passenger carriers (taxi and livery fleet operators), rental fleet operators, as well as the general driving public.

Road users are greatly dependent on mobility services such as navigation, traffic, weather, tolling, and parking. Mobility services typically give road users the tools they need to arrive at a given destination, generally with the additional goal of measuring and meeting the demand for trips, frequently in the face of limited transportation network capacity (e.g. congestion) or under energy efficiency or environmental constraints.

For drivers, mobility information services reduce the cost of searching for information and reduce the uncertainty regarding to trip routes, travel times, costs, payment options, and other needs. Mobility services may also include many other broad operational goals such as improving safety or reducing the environmental impact of travel over time. (For example, applications such as eco-routing may use personal navigation devices to route drivers not based upon shortest route or time saved, but optimized for trip fuel savings and/or reduced environmental footprint.) Some information services match underutilized assets, such as unoccupied parking spots, idle taxies or rental vehicles, or empty passenger seats, with consumers demand for trips or destination amenities. Vehicle oriented services often are designed to preserve and maintain vehicle assets from wear and damage through remote monitoring, such as vehicle diagnostics, fleet management, and pay-as-you drive insurance.

For passenger and freight fleet operators, mobility services utilize automated vehicle location (AVL), asset or passenger tracking, computer-aided dispatch (CAD), typically integrated within a Fleet Management Systems (FMS). First responders such as Fire, Police, Emergency Medical, Transportation, and Public Works departments also manage large fleets and rely on FMS's. Some advanced fleet management systems measure vehicle utilization and analyze on-board diagnostics information to support preventative maintenance (e.g. to minimize fleet downtime and increase asset productivity). Fleet carriers measure usage in categories like freight delivered,

vehicle access and rentals processed, taxi passenger/transit trips taken, toll facilities and parking utilized, 911 calls responded to, etc., and must process financial transactions or operational activity logs finely tuned to this usage.

For road user mobility, security requirements emphasize the integrity, availability, or confidentiality/anonymity of services for road users. There are four broad categories of security: Road user *integrity*, *availability* and *authentication*. *Road user integrity* focuses on the integrity of telemetry data being stored in vehicles or other systems, such as odometer readings. *Road user availability* policies ensure that services will be available when needed, such as when making transportation payments (e.g. a transit, parking tolling), and may provide alternatives when critical services are not available, such as cash payment. *Road user authentication* controls are designed to verify an identity to process ticketing, tolling and transit payments, or to assert identity, such as assuring that a bona-fide emergency vehicle is seeking to access to execute a signal priority function at a road intersection. Some systems seek not only to assert or verify an identity, but to ensure that personally identifiable information is secure from third parties such as partner organizations or prying hackers. *Road user privacy* policies seek to de-identify or obfuscate data to preserve the confidentiality of driver data that may be used for other useful purposes in highway traffic or parking applications. Such private data may include origin and destination location, route, speed, and other data generated by vehicle- or infrastructure-based sensors.

Road user security requirements for freight mobility often focus on integrity and availability of data from vehicles, but also integrate logistics data. Freight businesses often operate their own fleet delivery vehicles, depots, and posting stations, covering international service footprints. Large freight and courier services, such as UPS, FedEx, and the U.S. Postal Service rely upon supply chain management systems that manage cargo from local pick-up through delivery. Others are small operators (sometime single owner/operators) that support freight forwarding, but also rely upon online tools to interact with customers. Lost or stolen cargo is a significant risk for both large and small and occurs in a range of freight-forwarding and storage operations. Exposure is greatest when vehicles are in the process of being loaded or unloaded. The now-widespread use of sealed containers and “track and trace” systems has helped to reduce petty pilferage. However, track-and-trace also allows for large-scale cargo theft if these systems are compromised, making them lucrative targets for more sophisticated criminal hackers. The integrity and availability of these logistics systems are critical to prevent disruption of freight commerce.

The mobility information services chain for road users is long and includes data production, data processing, and information dissemination. For example, transportation infrastructure operators such as a highway operating or tolling authority may collect traffic, weather, tolling, parking, and intermodal connections data, and may (often through a third party application provider) further processes and package pre-trip or in-transit information (e.g. alerts, routing etc.) for dissemination to various communication channels (e.g. roadside dynamic message sign, 511, radio, TV, web service to mobile device etc..). Security controls must be applied to ensure the integrity, availability of information at each stage of the mobility information services chain.

Some traditional services, such as navigation, are changing as the sources of data become more diverse. Mobile devices have become sources and stores of data related to road infrastructure, and road use. Companies like TomTom, Waze and others organizations provide information about turn-by-turn navigation, weather, traffic, and amenities (e.g. fueling stations, businesses, etc..) to road users in exchange for location, speed, and other information collected (or “crowdsourced”) from their subscribers’ devices. Data is also collected, bartered, and/or sold by organizations such as commercial freight fleets that collect traffic information from their trucks.

Organizations such as Open Street Map, Google, and Here even crowdsource road maps to include points of interest, GPS traces of road geometry, and user diaries documenting the placement and condition of buildings, signals, signs, and other roadside infrastructure assets. The crowdsourcing of such data reduces the cost of data collection and may even improve the coverage and quality of mobility information in some cases.

Cloud-based "new mobility" services are recent mass-market phenomena that have garnered a lot of adherents among road users. Whereas conventional mobility services are often provided by local transportation authorities (e.g. transit agencies, individual parking operators) or regional organizations (e.g. E-ZPass Interagency Group), "new mobility" services are typically designed for scalable nationwide service, covering millions of road user accounts. Mobile device apps are the "authentication" gateways that allow road users to purchase shared mobility and other services. Such services include rental/car sharing (e.g. *Zipcar* or *Car2Go*), bike sharing (*Alta*), peer-to-peer car sharing *(Relay Rides*) and ride sharing or slugging (e.g. *Lyft*), all part of the much-hyped "sharing economy." Other new mobility companies provide livery and taxi dispatch services (*Uber)* and parking (*ParkMobile*, *Streetline, Parkme*, among others) at large scale, and have greatly augmented or even disrupted traditional, locally provided markets for these services. and travel metasearch engines such as *Ride Scout,* aggregate data from all of these services in a way that is similar to airline fare aggregators such as Kayak.

Because of their size and scale, these services are often centrally hosted, so availability is important. Hackers seeking to disrupt these services will have a greater impact in the future when road users become more dependent upon them. Shared mobility systems are also a potential target for criminals seeking to steal, disable, or track vehicles, so the confidentiality, integrity, and availability of these systems must be maintained through security controls. Furthermore, road user privacy can also be compromised as GPS location information is crowdsourced from devices, so confidentiality controls must be assuredly applied.

There have been attacks on these mobility and safety information services that were carried out in order to disrupt, defraud, or just experiment. The attack on the San Francisco Bay Area Rapid Transit (BART) system carried out by the hacktivist collective Anonymous in 2012 is one example. Others are penetration attacks on Traffic Management Centers (TMCs), usually with the goal to manipulate roadside Dynamic Message Signs. However, there have been a number of attacks on Radio Frequency Identification (RFID) transponders (also known as Near Field Communications or NFC) that are embedded in wallet-sized cards, and increasingly in mobile devices, to authenticate payments at point-of-sale terminals (POS). POS terminals in transportation are typically walk-up/drive-up parking gates and transit turnstiles, but some are also tolling gantries where vehicles are moving at high speeds. Researchers who have hacked such systems were able to reset and reuse -- free of charge -- transit access cards in San Francisco's MUNI, New Jersey's PATH, and Boston's "T" transit networks. Researchers have also cloned FasTrak tolling transponders to allow a hacker to claim someone else's account ID and charge tolled trips to them.

Security of Transportation Safety-Critical Systems

Transportation safety systems include most of the control functions of vehicle across several domains (powertrain, control, body, and telematics). Critical safety systems are also integrated into highway and rail infrastructure, and include traffic control devices that automate signaling and assign right-of-way. Intersection traffic signals, railroad crossing gates, and other critical roadside devices require high levels of availability and integrity. These systems have undergone extensive safety engineering, to ensure that they remain dependable in the face of random errors and faults conditions.

Automotive systems for decades improved in terms of safety and reliability, and security engineering has recently become another element of focus. Automotive systems have become more dependent on software driven innovations in the last decade and a half. The University of Washington and the University of California-San Diego (UW-UCSD) Center for Automotive Embedded Systems Security (CAESS) conducted an experimental security analysis of automotive systems in 2010. The researchers were able demonstrate, in the 2010 work, the ability to control in an adversarial fashion a vehicle's control functions in a way that completely ignored driver input. The team did this by reverse-engineering benched vehicles, extracting software from the electronic control units (ECUs) that control the engine, transmission, braking, body, telematics, and ignition.

On a benched vehicle, the UW-UCSD team injected packets into the car's controller area networks (CAN) to manipulate ECUs in such a way as to change vehicle states, producing effects such as random braking. (The experimenters also acknowledged that vehicle systems respond appropriately when internally networked components are prevented from communicating—but tolerance to adverse manipulation is not part of the same critical assurance criteria.) They recognized that networked vehicle ECUs either had no authentication and access controls, or that those controls were inherently weak. The UW-UCSD team was able to bypass controls and was able to update ECUs, or re-flash them, while the vehicle was running. Researchers speculated that they could reflash the ECUs with malicious code if they had desired to do so

In its 2010 report, the UW-USCD research team only suggested what the impact might be *if* an adversary were to maliciously compromise a car's internal communications network, not *whether* an adversary might be able to do so, beyond prior physical access to the vehicle. Since a potential adversary's direct physical access to a vehicle, rather than remote access, represents no greater threat than the risk of conventional sabotage or theft, the researchers conducted follow-on experimental analysis of the external attack surfaces of a car in 2011. The researchers concluded that indirect physical access could make a difference. "Indirect" physical access would require an attacker compromise an appliance (mobile phone or dealer diagnostic reader) that would be attached to an On-Board Diagnostics (ODB-II) port, a USB port, or a Bluetooth radio connection to the vehicle. "Indirect" access could be used to pass through an attack payload (e.g. Trojan horse) to infect the telematics unit. The attack payload could then could be activated remotely at a particular time, or be triggered the attacker at any time upon transmission of a unique specified sequence of common commands to the telematics unit.

The UW-USCD team did find one method that enabled remote access that did not require the physical presence of an attacker. The team found a method to bypass the authentication mechanism of a commonly-used, proprietary cellular based data link protocol. They did this by discovering and exploiting the commonly-encountered buffer

overflow bug, enabling the team to dial the car's telematics unit and force it to download and execute malicious code. The payload could then exfiltrate data found in the telematics unit such as GPS coordinates, and turn on the in-cabin microphone for surveillance. A toehold could then be established or mount additional attacks from the telematics unit onto other networked parts of the vehicle, such as the theft prevention module.

These experimental analyses revealed possible internal (direct and indirect physical access to vehicle) vulnerabilities and external (direct remote access) attack vectors. To date there have been no publically-disclosed vehicle system exploits yet discovered "in the wild." Understanding and gaining access to internal vehicle network components is still likely beyond the capacity of all but the most sophisticated potential attackers. Furthermore, the UW-UCSD experiments still require a considerable effort, especially for remote compromises of multiple vehicles, and would-be attackers are more likely to seek the path of least resistant by finding easier, more conventional methods of sabotage to compromise vehicle safety. Nonetheless, their work was thought-provoking and suggests that if attackers in the future become more sophisticated, such remote cyber-sabotage might be move beyond the realm of the theoretical.

Unlike in conventional IT, there is still a great deal of heterogeneity of architecture, components, and software between makes of vehicles. This heterogeneity makes large-scale vulnerabilities difficult to discover, assuming potential adversaries are somewhat resource-constrained and seeking path of least resistance to collect the greatest number of exploits. Nevertheless, the attack surfaces of vehicles are changing by adopting common standards: intra-vehicle indirect physical access (ODB-II), wireless Tire Pressure Monitoring system, multi-media device (USB, Bluetooth, ZigBee); and remote direct access telematics (via cellular 2G). Next generation vehicles will likely include Wifi, 4G cellular for mobile broadband, and vehicle Dedicated Short Range Communications (DSRC) for crash avoidance. There are likely more than 6 million light vehicles currently with 2G access via telematics surfaces, and it is estimated that number will grow considerably.

One risk that must be addressed is mobile device integration into vehicles. Smart phones and other personal navigation devices have long been brought into the car by drivers for telephony and traveler information. Some speculate that these devices will be integrated into vehicles by pulling telemetry and providing services similar to conventional built-in telematics, but this may be done so primarily to reduce the growing problem of driver distraction (e.g. drivers attempting to use mobile devices while operating their vehicles) . NHTSA estimates that nearly 6,000 persons per year die in car crashes that result from driver distraction. As automakers seek to integrate appliances into vehicle cockpit dashboards, head units and the ODB-II port to support driver, passenger, and freight mobility services, mobile devices may be used as stepping stones to compromise homogenous large production vehicles or fleets.

Automotive manufacturers must manage the risks associated with the coming complete integration of mobile broadband and the vehicle. Embedded cellular telematics systems are gaining traction. GSMA predicts that embedded telematics (and incidentally also tethered telematics) annual growth will reach nearly 11 million by 2020, which would encompass nearly every new vehicle that rolls off US assembly lines. Embedded telematics is becoming more sophisticated, function rich, and extensible. OnStar's Advanced Telematics Operating System (ATOM), the car industry's largest cloud-based automotive platform, has offered a restricted or "closed" Application Programing Interface (API) where select third parties may offer services such as peer-to-peer car sharing. (Drivers can remotely unlock and start their vehicles for friends, family, or renters.) Such high profile

telematics systems may be tempting targets, especially for car thieves, so service providers will likely be placing enormous emphasis on operational and reactive security protection strategies to deter would-be attackers.

Embedded telematics has also advanced in not only providing road user mobility applications such as car sharing, but also safety and vehicle maintenance. Tesla rolled out an over-the-air update to ECUs that control their "smart air" suspension in their Model S vehicle after a high profile crash. The crash was the result a vehicle running over a large metal object that punctured the Patharmored plate on the Model S underside that protected the battery, causing a fire. The remote reflash of the ECU reconfigured the suspension to reset the ground clearance of the Model S at highway speeds to reduce risk of underbody impact damage. The Tesla Model S air suspension update is a window into a potential future – a future where car maintenance and software maintenance merge and the auto original equipment manufacturers takes a bigger servicing role through telematics.

Not only must mobile broadband to the vehicle be carefully monitored and managed, but also connectivity to road infrastructure. For safety critical traffic management systems, security must be ensured as these systems are becoming ever more adaptive and interconnected, typically remotely managed by road operators' Traffic Management Centers. Some elements of critical assurance have been well understood and have been implemented for decades in roadside traffic control devices, such as traffic signals and ramp meters. "Conflict monitors," for example, are designed to assure that for a traffic signal, a "green" phase is never allowed for two opposite traffic flows for any given simultaneous period. The conflict monitor shuts down the traffic controller, putting signals in all directions in "flash" mode (e.g. where all directions receive a blinking red signal).

Traffic signals phases (combinations of "reds" and "greens" in opposite directions) are also frequently actuated based upon revised timing plans or even in real time based on traffic flow that is detected by sensors. Signal phase and timing are often tweaked periodically (or in real time for adaptive systems) to provide optimized traffic flow. Although arbitrary changes in signal phase and timing may create some disruption by delaying traffic, as long as conflict monitors are still functioning, there is minimal risk to safety. Efforts to have traffic signals transmit their signal phase and timing through various channels (for example using DSRC at the intersection, or other telecommunications media) have been underway allowing next generation vehicles to harmonize speed and braking with changes in traffic signal phase. The US Department of Transportation is currently conducting research within their "Connected Vehicle" program to look at the operational issues to ensure safety, security and reliability of transmitting signal phase and timing data.

Next Generation Security and Defense-in-Depth

Why are systems that are trusted with critical tasks not always "trustworthy"? The answer is that the demand for complex software-based systems has increased more rapidly than the ability of programmers to design, build, test, implement, and securely maintain them over their product lifecycle. On another level, the demand to provide connectivity and/or add new functionality to existing products have on occasions out stripped the capacity for security engineers to design and properly implement security protection features such as authentication and access control. Often, the assumptions made when security features were designed are dramatically changed by events - new threat models emerge as constraints holding back attackers fall, such as computing resources or knowledge.

There are several culprits:

- Lack of investment in upstream activities such as requirements definition, design, and architecture development;
- Lack of time, resources, expertise, techniques, or tools to reduce errors in coding phase or later in testing and quality assurance
- Changes of fundamental assumptions later in the product lifecycle, such as dramatic changes from the previous design or in the threat model (e.g. adding features that expands attack surface, or dramatic changes in adversaries incentives and resources).
- Different models and vintages of applications, devices and networks that are interconnected, each with unique security controls and vulnerabilities.

Given this state of affairs, the burden falls on the end users of software of ensure security, where failures are frequently attributed to:

- Very limited ability to attribute attacks to individuals and therefore to use law enforcement to deter attacks, or otherwise significantly increase the potential costs and risks to cyber-criminality).
- Lack of time, resources, expertise, techniques, and tools to remediate and mitigate impacts of software errors and security vulnerabilities in implemented products
- Limited ability to collect and analyze data to identify attack patterns and disrupt attacks in real time, before attacks progress far enough to result in significant losses.

Continued innovations in transportation are becoming ever more dependent on device connectivity and mobile broadband, but also on ever more sophisticated software. The costs for software in automotive electronics are estimated to approach the 50% margin in car manufacturing in 2015, and that nearly 90% of vehicle innovations are centered on software. In modern automobiles, there are over 100 million software lines of code (SLOC) across 70 to 100 microcontrollers. (By comparison, a modern military aircraft has only 20 million SLOC). The amount of software will increase as new technologies that support vehicle crash avoidance, driving automation, and advanced highway traffic control progress in the marketplace.

Road Infrastructure active traffic management systems depend greatly on software. There are around 300,000 roadway traffic signal controllers in the country, around half of which utilize some form of software operating system with modular hardware or software components. Nearly 8,000 are so called "modern" Advanced Transportation Controller traffic controllers that include the well know real-time Linux. Linux is an open source kernel and operating system package available in several variants that support nearly up to 80% of all network servers and almost 100% of all high performance computers. As of 2013, the Linux 3.10 release had 15 million SLOC.

Software bugs average is about 15-50 errors per 1,000 lines of delivered code, and this fraction may vary by industry and application. However, this fraction is usually representative of code that has some level of structured programming.. This results in a high level of maintenance required in the form of software updates and patches. Software and hardware defects in vehicles are surprisingly common, with nearly 50% of the warranty costs related to flaws found in automotive electronics. In general, the problem is not specific to transportation, as multiple studies estimate that maintenance costs are at least 50%, and sometimes more than 90%, of total costs associated with even simple software systems.

Unfortunately "ship and maintenance patch" cycle cannot be easily fixed by testing. Discovering vulnerabilities through testing is an incredible challenge and cannot alone establish software assurance or security. One strategy is to increase testing, which raises the question of just how much testing can be done given limits on time and other resources most organizations face. The possible combinations of system configurations, interacting job threads, data errors, user errors, and hardware malfunctions even for simple systems can be astronomical. Most organizations do not have enough time or resources to test all functions for complex systems.

In fact, a large-system "testing footprint" likely cannot cover even 1% of all the possible combinations of system states, which explains why modern IT systems in particular have a seemingly endless number of bugs and security vulnerabilities. Even after years of software patches, defect-discovery rates generally have not decline for IT systems. The reason is that these systems generally enter test from software build with many thousands of defects, and testing footprint typically finds less than half of them. The rest are found by users whenever they use the system in ways that were not anticipated and tested.

Testing also fails to find must vulnerabilities because of extensibility. System developers constantly seek to expand the scope and lifecycle of their products. Modular or "extensible" systems are architected to allow components to be integrated and stitched together on an as needed basis to expand feature sets. However, extensible systems often fail in way not anticipated by designers and testers when implemented, where security controls often fail at the seams between components.

The long-term improvements in cybersecurity require innovations in *constructive*, *operative*, and *reactive* security. *Constructive* security shutters potential vulnerabilities before they surface in final products -- in the requirements, design, coding and testing phases of the software development lifecycle. *Operative* security focuses on *remediation* -- maintaining software products, by patching potential vulnerabilities found after implementation. *Operative* security also includes *mitigation* -- limiting exposure of critical functions through configuration and management. Security administrators must be able to employ *reactive* security that can detect adversary

surveillance and casing, and can respond to potential intrusions while in progress. *Reactive* security is often the more complicated to deploy as it requires growing expertise in analytics and "operational" intelligence.

Constructive Defense and Software Assurance

In general, ever greater demands from business, consumers and governments for more functionality have accentuated the need to add tremendous size and complexity to software. Well over half of errors are introduced in software begin early in the software development lifecycle, specifically during the requirements and design phase. Coding also adds significant errors, while testing often finds only a minuscule fraction of bugs generated in the earlier phase. Not every software error may be critical and lead to a failure in functionality. Nor does every software error represent a security vulnerability. However, overall aggregate reduction of all errors will improve reliability and reduce opportunities for adversarial exploitation. Over the long term, constructive security requires considerable innovation in software assurance and implementation "upstream" in product development -- so-called Secure Software Development Lifecycle practices.

The reality is that *no software product of nontrivial size and complexity can be assumed free of error or security weaknesses.* Therefore, even the best designed "secure" systems that addresses all critical assurance requirements can never achieve a complete state of unassailability. Software assurance is the level of confidence that a given software element functions in the intended manner. The U.S. Department of Homeland Security (DHS), defines assurance as the measure of confidence that software can be "predictably executed" assuming no exploitable vulnerabilities exist. Assurance is difficult to measure precisely, but it is well understood by the software development community that good requirements, design, and architecture -- traditional upstream product development, increase predictability of functions and security of those functions.

Given our ever increasing dependence on software, improvements in reliability and security must be rooted in reducing complexity. NASA for example had been known for decades for developing reliable safety critical software for cutting edge aerospace and advanced robotic systems. But as their systems became more complex, software reliability and began to wane. In a report issued in 2010 after a series of mission failures, NASA recommended that systems use vetted requirements to avoid unnecessary features and what it termed "incidental complexity." Good software architecture is the most important defense against incidental complexity in software designs, but good design skills are rare. NASA's study made three recommendations: (1) allocate a larger percentage of project funds to up-front architectural analysis in order to save in downstream efforts; (2) create a professional architecture review board to provide constructive early feedback to projects; and (3) increase the ranks of software architects and put them in positions of authority. NASA problems reflect those in information technology in general -- our ability to design and construct software systems with assurance is being outstripped by organization's ever expanding expectations of what computing systems should accomplish. If so-called "automated vehicles," such as the Google Self-Driving car, are commercialized, then expectations and dependence on software systems in highway transportation will grow by leaps.

Efforts have been underway in the automotive industry to improve upstream software development practices to improve the integrity of safety critical systems. Such standards, such as the *International Standards Organization*

(ISO) draft 26262 Road Vehicles - Functional Safety, which has been developed to deal with ever more complex software requirements and designs in functional safety systems for road vehicles. ISO 26262 is similar in function to the Federal Aviation Administration (FAA) Directive Order (DO)-178B, *Software Considerations in Airborne Systems and Equipment Certification,* which was implemented since 1992 with the purpose of providing a framework for establishing assurance in aviation software. Systems potentially covered by ISO 26262 include passive and active occupant protection systems, crash avoidance/advance driver assistance systems, drive-by-wire systems, and electronic stability control, among others.

ISO 26262 describes an automotive lifecycle (management, development, production, operation, service, and decommissioning) for safety, and supports an automotive-specific, risk-based approach for determining risk classes similar to those described in the FAA's DO-178B. ISO 26262 specifies *Automotive Safety Integrity Levels* or *ASILs*. IS0 2626 uses ASILs for specifying the necessary safety requirements needed for achieving an acceptable level of risk. A number of experts have suggested incorporating malicious threat assumptions and models to *ASIL* determination to inspire requirements for critical assurance related to security, not just safety. Ultimately, the hope of automotive electronics engineers is that both safety and security requirements defined through the ISO 26262 process may support validation of the level of safety and security that is being achieved in later stages of product development.

But the role of the road user, particularly the driver, and road operator is crucial. The principle for design of secure systems is "easy to use securely, hard to use insecurely," which is similar to a principle used in safety engineering, which is focused on making it as easy as possible for the driver to operate the vehicle in a safe manner (e.g. any given driver is accustomed to the position of the brake pedal and how to use it). ISO 26262 *ASIL* in particular examines the role of the driver to respond to a potential unintended failure of a system, known as "controllability." An example of redundant "controllability" is that if brakes fail, a driver can exert braking control using the parking brake. Each *ASIL* ask the question, "If a failure arises, what will happen to the driver and associated road users?" and measures probability of exposure to danger, vehicle controllability and severity of failure to determine the integrity level that must be achieved in the requirements process. In use cases where the driver cannot mitigate loss of control in the event of a catastrophic system failure, the integrity requirement is more stringent and more attention, time and energy need to be put into software development lifecycle process to support the function.

Furthermore, coding and testing expertise, techniques, models, training, processes and tools are also important for software assurance and constructive defense. When requirements are translated into software designs and lines of code, errors are likely to be introduced, and software development techniques vary. Techniques range from "Code-and-Fix" or "Cowboy Coding," where programming is creative and free-form, to time consuming, highly specialized and structured techniques such as "Cleanroom," where code build emphasis is on defect prevention with the goal of producing "clean" software with a "certifiable" level of reliability. Process improvement is also important. Process improvement models, such as the Carnegie Mellons' Capability Maturity Model (CMM), offer another technique which uses maturity ratings to encourage organizations to evaluate and improve techniques and practices, by increasing predictably, effectiveness, and control of software development through optimization and continuous improvement.

The use of structured coding techniques to find and fixing bugs in the coding phase pays off. The costs of fixing bugs rises the later in the lifecycle that the errors are caught. Catching and fixing errors in the later testing and

maintenance phases costs, on average, $9,000-$15,000 per defect, as opposed to nearly zero in the upstream coding, design, and requirements phases. Larger investments of time and the use of best practices in upstream activities not only pay off in terms of code quality and cost, but also in terms of software reliability and security.

Coding and testing often require close coordination in order to catch errors introduced in requirements and design phase. “Cleanroom” and other structured techniques, for example, rely upon statistical quality control to determine assurance in upstream design and coding phases, rather than waiting for final product or unit testing. In such structured software development techniques, a suite of test cases is created to match the probability distribution of the projected product use patterns, based upon requirements and designs. The testing process includes defining the frequency distribution of inputs to the system, the frequency distribution of different system states, and the expanding range of developed system capabilities. In the testing phase, independent testers record observed failures and determine the statistically significant, or “certified,” measure of software reliability.

Coding tools and guidelines are also important. Motor Industry Software Reliability Association’s (MISRA) issues "Guidelines for the Use of the *C* Language in Vehicle-Based Software," also known as "MISRA C." These guidelines, describe a workable subset of *C*, the most widely used higher level cross platform programing languages, that avoids many of the languages well-known problems. *C* code that claims conformance to MISRA C must comply with 93 required rules. Conforming code must adhere to these 93 rules, as well as another body of advisory rules where practical.

Research and practical efforts to improve software quality through Software Development Lifecycle processes matter. Microsoft cofounded the $30 million Sustainable Computing Consortium-based at Carnegie Mellon with NASA and many other firms to promote standardized ways to measure and improve software dependability. Quality control efforts are effective, and examples cited include Lockheed Martin software in its C-130J “Super” Hercules turboprop transport aircraft used such methods to cut development costs by 80% while producing software that passed stringent Federal Aviation Administration certification with “very few errors.”

In addition, structured processes allow independent testers to submit a limited number of test cases to ensure correct system operation for situations in which a software failure may result in a catastrophic failures. NASA and the FAA, which develop and certify aviation and space systems that often cannot be patched easily or inexpensively, have typically emphasized the development of certifiably reliable software. As cars begin to add more automation, either through crash avoidance (e.g. Forward Crash Prevention, Vehicle-to-Vehicle/Infrastructure Safety Communications) or driving automation (e.g. Adaptive Cruise Control or full range “self-driving”), one of the key challenges will be to certify that such vehicles are safe given the wide range of potential driving scenarios, system inputs, and system states that may occur.

For the most part, ISO 26262, MISRA, structured coding techniques and other lifecycle development frameworks, methodologies, and tools represent the technical state of the art -- the highest level of development of a device or process at this particular time. Despite current state of the art, software assurance is still difficult to achieve given limited time and resources to conduct testing. Some long range technologies and innovations, however, may push software assurance beyond the current practices and technologies.

One new technology area may be automated testing techniques and high performance computing. A typical test footprint of a complex system may cover a miniscule portion of all combinations of system state. Having a high-

performance computer run automated test routines of a vehicle's functional safety features through the simulation of every possible input combination (e.g. acceleration, steering, etc..) and system state (e.g. yaw rate, speed, detected obstacle distance etc.) could potentially take years even for a simple crash avoidance feature, given the complexity of modern automotive electronics and the number and combination of conditions under which a driver and a vehicle must perform. Even with automated testing, test cases must be limited to a few that are determined (and perhaps measured statistically) by designers to be critical in terms of their potential impact to safety and operations.

Another long range approach is to improve the performance of automated testing tools using next generation high performance computing technologies, such as quantum computers. Quantum computing is new and rare technology that has the potential to step-wise improve software assurance, among many other innovations. Conventional computers process zeros and ones serially, but quantum computers can use quantum-mechanical phenomena to perform operations on data. At the quantum level, a computer may able to program atoms to represent all possible input-output combinations, and to do so simultaneously, so that, for example, a test of an algorithm see all input combinations at once, rather than one-at-a-time. A conventional computer would need to serially cycle through every possible input combination to arrive at a testing result, meaning it would take longer than the age of the universe to complete some of the most demanding calculations within test scenarios.

One of the first potential applications of quantum computing may be automated testing. Validating the performance of software is a time-consuming and expensive part of the software development process, even when sound software development lifecycle process and structured programing techniques have been utilized. A quantum computer could automate testing by scanning through the switches and combinations in the software code and makes sure that an algorithm, a software model or even perhaps an entire system put together may be performing in a way that the designers expected it to. The speed at which quantum computing can test features means more features may be tested in the same production window. This will expand the testing footprint of systems in development and potentially reduce the number of bugs that might reduce reliability or expose potential vulnerabilities in final software products.

Quantum computers could also solve other transportation problems beyond testing and software assurance, such as calculating the shortest route or optimal schedules to speed passenger and freight mobility and improve asset productivity (e.g. the computationally hard "traveling salesman" problem). Yet much time is needed before high performance computing can make an impact in software assurances to support functional safety and security, or to improve transportation mobility and logistics. In recent tests of the commercially available D-Wave quantum computer showed it to be 3,600 times as fast as a conventional current high performance computer. Currently NASA, Google and Lockeed Martin have purchased versions of D-Wave (128 and 512 qbits) quantum computers, however, there is still much work to be done in understanding the performance, constraints and potential of this new technology. With higher levels of driving automation (e.g. driver assist systems and self-driving cars) and greater reliance on software in vehicles, there is great potential and hope that high performance computing may tackle software assurance and lower the cost and complexity of development of certifiably safe intelligent transportation systems. Improvements in crash avoidance driving automation require engineers reign in complexity.

A recent comprehensive forensic examination of automotive engine-control code from one automotive manufacturer showed OEMs may be struggling keeps software simple. The cyclomatic-complexity scale is tool used to rate software complexity, with a 10 rating is considered normal and a 15 rating being the maximum complexity limit. Within the examined engine control unit several functions rated higher than 50, with some over 100. Forensic examination showed the engine control system had more than 11,000 global variables. Proponents of structured methods in computer programming generally consider the use of "global variables" as very poor engineering practice because global variables can potentially be modified from any part of the program, and any part of the program may depend on it, which makes entire code too complex to for programmers and testers to understand. A global variable therefore has an unlimited potential for creating mutual dependencies, and adding mutual dependencies increases complexity and make software product untestable and unmaintainable.

Even if software assurance improves through incremental and continuous improvements in software development practices or potentially revolutionary innovations in high performance computing, security will still be a concern. Security systems still need to be designed to remain robust against a large number of potential threats that are difficult to contemplate, model and test, especially over a long software product lifecycle. A so-called "Secure Software Development Lifecycle" (SSDL) practice therefore differs in important ways from other software lifecycle and assurance processes used to validate safety and reliability, though improvements in one area will lead to improvements in the other.

Software Maintenance and Operational Strategies

The requirements for authentication and access controls are typically established in the very beginning in the software development lifecycle, and these controls must be configured once a given system is fielded and maintained until the system is decommissioned. Often layered on top of authentication and access control is network perimeter controls designed to restrict remote access to systems and services, usually through the use of firewalls. Such preventative controls seek to reduce exposing vulnerabilities that are either latent or known to security administrators but cannot be quickly or cost effectively patched. Authentication, access control and network perimeter controls are required to enforce not only security policies focused on information assurance, but also management and secure control of software maintenance.

The center of gravity in security is authentication and access control. Authentication verifies identity of those individuals (or other systems) requesting access to computing resources. Access control enforces security policy, which specifies what "level of access" can be given to those persons or systems that have been authenticated, to ensure confidentiality, integrity and availability of data. Security policies are established by system owners to reflect trust, with the highest level of access (or privilege) given to persons or systems that are deemed most trustworthy. The owner of a system typically holds the authority to determine trust and assign corresponding level of system privilege often based upon various factors, such as relationship to the system owner (e.g. employee, business partner, customer, etc..). Operational security includes the maintenance of security controls, such as authentication and access control processes. It also focuses on creating a secure environment where software can maintained and security vulnerabilities patched.

Hackers undermine access controls by seeking to acquire access privileges beyond those allowed by security policy. They do this either by directly undermining or circumventing access and authentication controls (e.g. by guessing passwords, exploiting buffer overflows, or discovering backdoors etc..). They also may undermine access controls indirectly by manipulating and deceiving trusted parties who already possess privileged access (e.g. inserting malicious code through scamming/social engineering etc..). Perimeter defenses are then added on top, designed to filter out traffic on services where other controls may be weak or are not able to cover the entire surface of a system.

Operational security must also maintain the integrity of the software maintenance process. Software engineers and system administrators constantly become aware of shortcomings in their software designs or configuration, and they incrementally fix these shortcomings through maintenance patches which typically require a distribution channel to push out updates. As time passes, new susceptibilities to technology protocols are discovered, and system administrators must be prepared to change policy and to push out software updates, configuration, and user guidance to ensure its implementation. For example, operational security focuses on implementing and enforcing a policy directive that forbids the use of technologies or protocols with known susceptibilities to eavesdropping (e.g. such as outdated versions of SSL).

The lack of a process,or the use of a process that is poorly designed and vulnerable to manipulation, in verifying the source and integrity of software maintenance updates is one major vulnerability. This patch distribution, often done over the internet, adds expense and even introduces new vulnerabilities as the process for updating software may be manipulated (e.g. through the introduction of malware). Patches and updates typically require a separate and very costly security infrastructure that can authenticate the origin and integrity of software updates.

This security and software maintenance process in particular is critical in a number of intelligent transportation systems that use networked programmable logic controllers. For embedded or “machine-to-machine” devices, controller area networks and other operational technology, there are three operational protection measures: *Secure booting*, *updates and patches*, *access control* and *device authentication*. When a device is turned on, it boots or loads software, the hardware must “trust” that the software being loaded during booting is legitimate and authorized to run on the device. Once a device is in operation, it will often receive updates and patches that must also be authorized and legitimate, and must be applied in a way that does not risk the operations of the devices and the functional safety of the system. Both booting and patching require cryptographic signatures to verify the legitimacy of the software.

Machine-to-Machine devices are different than computers in that they do not have a user that can validate himself by keying in log in name and password. M2M systems instead have credentials stored in a tamper-proof secure storage area. Access control and device authentication is required if the device is plugged into a network and must communicate that it is a legitimate “trusted” device. The device must also confirm that it is connected to a legitimate network, not under the control of an adversary. Networks or communications sessions in which the adversary is in the middle of the transaction, privy to traffic, is known as a “network impersonation” if at the network level, and a “man-in-the-middle” attack if at the session level.

Two other operational controls encompass perimeter defense - *firewalling* and *intrusion detection*. Even if software developer or system administrator finds a vulnerability, then there is no guarantee that it may be patch-able in a timely fashion. Even with IT systems, the window between when a potential exploitable software error is

uncovered surreptitiously by a potential adversary and the date is it discovered (either because of a successful attack, or through so-called "white hat" penetration testing and/or "bug" bounties) by the software developer or system administrators in the field can be considerable. Even after that time period, the date patches can be developed by the software developer and then extensively distributed and applied by system administrators can be take additional time, sometimes months. Between the time an exploit is discovered and a patch is developed and applied, perimeter defensesare typically the only countermeasures remaining in that period that might protect an unsecure system. Perimeter defenses reduce exposure of crippled systems and services from threats emanating from the outside world.

Traffic from the internet destined for a particular computer host or electronic control unit must be inspected to ensure it is not attempting any execute any unauthorized run time processes or attempting to illegitimately root, reboot or patch devices. Host based *firewalls* and *intrusion detection* systems inspect traffic at the packet level. Most embedded program logic controllers utilize unique communication protocols (for example, automotive uses protocols such as CAN or Flexray for internal networking of PLCs) that are translated to and from conventional protocols (such as the ubiquitous Transmission Control Protocol/Internet Protocol, or TCP-IP) between network domains. Protocol filtering and deep pack inspection at the gateway to these domains can be used to filter out traffic destined to devices that is not explicitly authorized.

Because program logic controllers used in automotive and other OT systems are small and do not typically have the computing resources to support strong authentication, more needs to be done to protect them at the network perimeter. Beyond a *firewall* and *intrusion detection* is the establishment of systems that restrict logical access to the OT network. A *demilitarized zone* (DMZ) network architecture use firewalls to prevent network traffic from passing directly between the publicly switched internet, corporate IT networks and a protected OT network. The purpose of the DMZ is to clearly separate, winnow and filter traffic in the zone so that it can be more easily scrutinized by "guards" such as intrusion detection devices. DMZ may also include additional layers of authentication and access control for users that seek to pass through the DMZ, in the form separate credentials, so undermining access controls and gaining access to an adjacent corporate IT network does not result in guarantee compromise of the OT network.

It is recommended that control systems should also use a network topology that has multiple layers, with the most critical communications occurring in the most secure and reliable layer. Some of these strategies have likely already been applied in the automotive telematics systems, though their implementation has not been ideal. Telematics systems like Onstar or others likely employ a DMZ to mediate traffic between vehicles and their telematics "cloud" and call centers. (Some Traffic Management Centers may also establish DMZs between their center and field traffic control devices). Even within vehicles, often separate controller area networks are established to support safety critical, low latency, applications and are separated from less critical network controlling non-safety functions. As the UW/USCD group discovered, however, there are some potential unmediated connections between these safety-critical and non-safety-critical networks that can enable an attacker to jump from one network to the other.

Operational strategies to isolate OT networks are critical, and failure has in the past result in some unfortunate events that compromised signaling systems in transportation. In August 2003, the Sobig computer virus was blamed for shutting down railroad signaling systems throughout the east coast of the U.S. The Sobig virus installed

a "back door" that let the hacker gain access without detection. Even though the signaling system was not the intended target of the virus, the signaling system components likely had an unauthenticated, unmediated connection to the rail carriers corporate IT networks, which fell prey to the virus.

Reactive Defense and the Rise of "Big Data"

Beyond operational security, reactive strategies arrest attackers that have not already been successfully repulsed by other operational countermeasures. Reactive controls look especially at host and application activity logs and seek to identify adversaries that have already established a "toehold" in a system. Reactive defense strategies assume the inevitability of a security breach, and controls monitor traffic both at the network perimeter, but also inside organization's local networks. Reactive security is the last line of defense and presumes other security controls have been bypassed or have failed.

The protocol for most cyber-attacks is composed of seven stages: *reconnaissance, probe and attack, gaining a toehold, advancement, stealth, listening post,* and *takeover*. Reactive security attempts to detect incidents at these different stages, and send alerts to system administrators that an attack may be in progress, especially in the early stages where toeholds might be prevented. For example, reconnaissance is the unauthorized discovery and mapping of systems, services, or vulnerabilities. Ping sweeps and port queries typically precede an actual access or Denial of Service (DoS) attack, or a targeted attack

Once a toehold is established, where an adversary finds a security weakness to gain entry into a system (such as a buffer overflow or command shell injection), an attacker then may advance to "rooting" the host system (advance from a unprivileged account to a privileged one that has access to all files and computing resources) and then to installing a backdoor that hides the attackers tracks. The attacker then "listens" to logs and other system activity, attempts to access other systems that have a trust relationship, before completing its goal to either steal data or disrupt system operations.

A network-based intrusion detection system (NIDS) involves the deployment of probing devices or sensors throughout the network which capture and analyze the traffic as it traverses the network. An NIDS evaluates a suspected intrusion once it has taken place and signals an alarm. An NIDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system, and is another form of an "application layer" firewall.

Beyond real-time reactive security are "big data" and predictive analytics. A "big data" approach mines data to identify unique patterns, filters out not just "signatures" but also "precurors," Both signatures and precursors are catalogued across an entire an organization, or even an entire industry where many organizations that have the same "risk profile" may choose to cooperate and share information about security incidents. Precursors are different from signatures in that they may capture external and internal reconnaissance, or attempts to escalate

access privileges and other activities that may appear to be innocuous but statistically outside of the scope of normal operations.

To determine what constitutes so-called "normal" operations, organizations would need to compile, audit, and analyze massive amounts of operational logs and security incidents reports. A big data approach would aggregate and correlate a diverse set of host event data and networking logs into a forensics database – one that can be leveraged to identify patterns of attack and anomalies on networks, as well as to identify access and usage patterns of applications and confidential data on host machines. With normalization, predefined and customization correlation rules, and policy and compliance driven searches, an organization can resolve security threats faster.

A big data and predictive analytics approach is key for organizations in the transportation sector where maintaining both connectivity and mission critical or safety critical services are the norm. Organizations that provide passenger and freight mobility services whose business models cannot let security constrain connectivity to its customers or business partners, for example would likely need to deploy some form of security predictive analytics. (For example, graylists of IP addresses may be generated and scrutinized more fully to ensure they are legitimate customers or supply chain customers.)

Entities in the transportation sector that are at higher-than-average risk to be targeted are firms where potential service disruptions or compromises would either be severe and direct -- danger to safety or result in a disruptive high visibility event from which recovery is time and resource consuming and difficult. Such categories of entities might include automotive manufacturers, services providers and suppliers, or fleet operators, that deploy and operate and maintain systems to reduce vehicle theft or eliminate unauthorized use (e.g vehicle alarms/immobilizers and remote start), guarantee performance of vehicle safety systems. (e.g. crash avoidance or occupant protection systems), or track critical freight vehicles/cargo (e.g. freight and especially hazardous cargo vehicles). Infrastructure entities may include road agencies or other infrastructure operators (e.g. State, Local and multimodal/regional transportation authorities etc..) that operate traffic control systems. Within the next several years, new vehicle-to-vehicle/infrastructure crash avoidance systems utilizing DSRC will require a *Connected Vehicle Security Credential Management* entity that will ensure the security of new "cooperative" vehicle safety systems that will be deployed in the near future.

Instilling Discipline and Advancing the State-of-the-Art

Many security failures are not the result of lack of best practices, guidance, tools and security controls and protection measures available, but often in lack of organizational policy establishing those measures, or where policy does exist, or lack of effective implementation. As defenders must be strong everywhere, strategy, governance and controls are critical to ensuring no gaps in defense. With the explosion in cybercrime and cyber-espionage, and rising fears of cyberterrorism, attention has converged on the vulnerabilities lurking in legacy code and less-than-secure IT and OT systems. Organizations must establish risk management strategies, rules, norms and culture that maintain continuous defense-in-depth posture --- that *constructive*, *operational* and *reactive* defenses are established and monitored.

The role of institutions and governance is critical to cybersecurity. Identifying the sources of risk and best mitigation strategies is hard. It is even harder when an organization frequently changes its operations in response to response market pressures or regulatory demands. Some business or operational requirements, such as adding or connecting a system or service to improve functionality, may incur risk (ie connectivity grows attack surface), changing the organizations "risk profile." Decisions to centralize information systems also change the threat landscape. For example, an organization that makes the decision to begin collection and storage sensitive data (e.g. financial, location data, etc..) to take advantage of a business opportunity or achieve operational efficiencies must be prepared to re-evaluate their assumptions about threats. In other words, such sensitive data stored in enterprise systems risks creating a more valuable asset from the perspective of potential attackers, often while simultaneously giving more individuals connections to it.

At an organizational level, industry and public sector practice often focus on a process to budget resources to reduce the vulnerabilities related to a risky activity. Such activities may include developing a complex software system, offering a particular new online information service, or inter-connecting an existing field system to a publically switched network to support remote monitoring. Alternatively organizations can make a strategic decisions to eliminate the potential risky activity altogether. They may also shift risk related to a given activity or risk management onto insurers or business or operational partners through contractual arrangements.

Organizations often fail to apply enough resources to reduce risks. Enterprises and organizations who purchase software likely underinvest in downstream operational security activities that could compensate for poor software development practices and shoddy security in commercially available products. According to a 2012 survey of technology managers conducted by the US Ponemon Institute and Bloomberg, organizations that want to achieve the highest possible level of IT security—capable of repelling 95% of attacks—would have to boost spending from the current combined $5.3 billion to $46.6 billion, a nearly ninefold increase. Even to be able to stop just 84% of attacks, these organizations would have to approximately double their investments.

From an investment standpoint, some research models suggest that optimal spending on cybersecurity operations protection for IT systems for a given enterprise should be between a quarter and half the expected pecuniary losses suffered in the event of a successful high impact attack. When cost of cybersecurity vulnerability to the entire economy is studied, a conservative estimate is that cybercrime (or other less pecuniary motivated forms of

mischief, such as hacktivism) represents nearly $100 billion annually in losses to the US economy. This represents about 1% of US GDP and may be growing, according to the research group the *Standish Group,* the security firm *McAffee* and the *Center for Strategic International Studies (CSIS)*.

Beyond budgeting resources, translating risk profile into tangible risk management objectives is difficult exercise for Chief Risk Management Officers and Chief Information Officers, and is more difficult with changing organizational goals and new technology. Once an entity baselines operations, assets and establishes critical assurance requirements and risk management goals, management objectives are established to target specific risks for mitigation, establishing internal controls and security protection requirements throughout an entity. There are two kinds of requirements, *principle* based and *rules* based.

Principal-based requirements are less technically prescriptive and internally weather time better than rules-based requirements, but are harder to articulate and implement. *Rules*-based requirements are more concrete and easier to apply and audit for compliance. *Federal Information Processing Standards* (FIPS) and other practices spelled out by the National Institute for Standards and Technology (NIST) or the *Payment Card Industry Data Security Standard (PCI DSS)* group are examples of rule-based guidance. Generic frameworks also include ISO 2700 series provides best practice recommendations on information security management, risks and controls within the context of an overall information security management system (ISMS). The ISO 2700 series is similar in design to management systems for quality assurance (the ISO 9000 series) and environmental protection (the ISO 14000 series).

NIST released the first version of the *Framework for Improving Critical Infrastructure Cybersecurity* in early 2014. NIST's *Framework*, created through collaboration between industry and government, consists of standards, guidelines, and practices to promote the protection of critical infrastructure. NIST's *Framework* is used to "identify and prioritize actions fo reducing cybersecurity risk, ans is a tool for aligning policy, business and technological approaches managing that risk." Core *functions* enumerated include tasks to *identify* risk, *protect* services and assets, *detect* security events, *respond* to those events, and *recover* post event, often with rules-based standards often identified in each category. Furthermore, framework functions implementation can be graded as *partial*, *risk-informed*, *repeatable* or *adaptive* – with the goal to create an operational culture that address dynamic risks and threats. Organizations are encouraged to create "profiles" that measure and reveal an organization's current tolerance for cyber risk as a starting point for evaluating potential future desired states. NISTs framework is similar to protocols the insurance industry is beginning to use to evaluate cybersecurity risk to rate potential customers.

Guidance and governance not only effects operations, but also suppliers of software, especially safety critical software. With potential to have billions of interconnected embedded devices in everything from heart pacemakers, to nuclear reactor cooling systems, to vehicle braking electronic control units, security is of great concern in safety certification processes. Transportation, health care and energy sectors are especially stringent, occasionally addressing conformance – the planned and systematic set of multi-disciplinary activities that ensure software processes and products conform to requirements, standards/ procedures. The *Federal Aviation Administration*, *U.S. Food and Drug Administration* and the *U.S. Nuclear Regulatory Commission* all require review of software development practices to understand how software assurance and *constructive* security strategies have been incorporated into safety critical systems. For *operational* security, the *U.S. Department of Homeland Security* sponsors critical infrastructure protection *Sector Coordinating Councils* (SCC) that promote cybersecurity

guidance within the sectors. DHS' *Science and Technology Directorate* sponsors research into new technologies designed to help industry cope with the new challenge of securing advanced systems.

The National Highway Traffic Safety Administration (NHTSA) does not review software development practices, but automotive manufacturers still must meet the requirements set in the *Federal Motor Vehicle Safety Standards*. The *NHTSA Electronics Systems Safety Research Division* and the *U.S. DOT's ITS Joint Program Office* have encouraged the development of *ISO 26262* and other constructive security practices designed improve software quality in automotive electronics, improve reliability, safety and reduce cyber-vulnerability. Furthermore, industry is working together through the Southwest Research Institute (SwRI)'s *Automotive Consortium for Embedded Security* (ACES) to conduct high-risk/high-reward cooperative research and to possibly serve as an independent verification and validation entity for the auto industry security systems. This consortium should begin work in 2014.

U.S. DOT's *Connected Vehicle* program seeks to ensure that vehicle-to-vehicle/infrastructure communications crash avoidance applications are secure-by-design, to eliminate the risk that adversaries can manipulate safety and mobility features of vehicles or traffic signals. In the future, there will be an ever wider variety of communication and sensing technologies embedded in vehicles, from AM/FM/HD radio, radio-frequency tolling tags, to satellite, Bluetooth, and cellular. A relatively new technology is Vehicle-to-*X* (X is for *vehicle*, *infrastructure*, or *device*) safety communications utilizing DSRC technology to warn drivers of dangerous vehicle conflicts or potentially hazardous roadway conditions. Vehicle crash avoidance applications which utilize DSRC are currently being designed to securely and anonymously exchange GPS coordinates and other critical safety data with relevant neighboring vehicles, with the goal of reducing the risk of vehicle-vehicle collision. Under the US DOT's Connected Vehicle program, the federal government hopes to kick-start the deployment of DSRC in light-duty vehicles sometime beyond 2014.

Research into traffic management and transit system security is also a priority. The USDOT's *Volpe National Transportations Systems Center* had conducted research on behalf of the USDOT and DHS in setting forth roadmaps, guidance, requirements, tools and techniques, and standards for building a variety of secure systems. The Federal Highway Administration has worked closely with *Idaho National Labs* SCADA Test Bed in evaluating and monitoring risks to highway traffic management systems. The *National Academies of Engineering, Transportation Research Board* and the *National Highway Cooperative Research Program* (NCHRP) are also conducting research in transit security.

Although research, standards and best practices are important, investment in security requires long term planning and resources. In the past, vehicles were final products with very little after-purchase maintenance required from manufacturers, beyond warrantied service. With more connected vehicles, *operational* and *reactive* defenses will need to be maintained and supported over the lifespan of connected vehicles. In the same way, most technology deployed by road operators is designed to minimize ongoing operational costs, as road agency operations and maintenance are often severely budget constrained between major capital investment cycles. Advance traffic management systems and Vehicle-to-infrastructure applications will require a new approach to supporting operations and maintenance of security as an ongoing service to the driving public.

Conclusion

Applications, devices and networks are increasingly interconnected, and each must be secure for all to be secure. "Defense in depth" security requires *operational* and *reactive* approaches, as well as *constructive* long term efforts to improve the security and reliability of application, device, and network software beginning in design and development. *Constructive* security protection strategies focus on improving planning, implementation and testing of software to reduce errors that, if exposed, can lead to a security breach. *Operational* security focuses on software deployment – specifically configuration and maintenance to ensure that vulnerabilities are patched or services that are known to be susceptible to manipulation are shuttered. *Operational* security also establishes firewalls to reduce adversary access to vulnerabilities that have not, or cannot, be patched. Lastly, *reactive* security assumes that attackers are scanning for access or have already penetrated operational defenses. *Reactive security* attempts to disrupt attacks in progress and those about to occur by predicting attack sequences.

Defects in software are often hidden costs to system buyers, whether they are enterprises, governments or consumers, and the costs are enormous. Since few organizations have the resources or expertise to develop their own software, they therefore cannot rely on *constructive* strategies to improve their security through improved *Secure Software Development Lifecycle* engineering. Instead they must rely upon third party software developer's diligence, and where diligence is lacking, introduce their own *operational* and *reactive* controls to patch and secure their systems. Since IT systems are both so widely standardized, commercialized, and configurable, they have larger attack surfaces out-of-the-box and must be painstakingly configured and maintained to reduce unauthorized remote access of critical resources and data. Much of cybersecurity guidance from standards bodies' authorities are devoted *operational* strategies such as information assurance, and patch and configuration management.

Improvements in techniques for writing software have failed to keep up with the explosive increase in the size and complexity of functional requirements. Security authentication, access and perimeter controls must cover a larger and larger number of functions with successive generations of systems and applications. Incredibly, software development teams often spend more than half of their budgets to repair flaws that they themselves produced -a figure that does not include the even more costly process of software maintenance – the furnishing product support and developing patches for software problems found after release.

Attackers typically start with the path of least resistance by finding vulnerabilities in systems that provide the greatest number of potential targets, and most of these are in widely commercialized and deployed IT platforms. In transportation, most mobility application supporting road users rely on IT security to support transportation payments, maintain location privacy, and prevent manipulation of telemetry for the purposes of fraud or theft. Where services are centralized and cloud-based, the aggregation of data in one place makes these services a bigger, more attractive target to potential attackers. For car-sharing, just-in-time logistics, tolling, smart parking and other mobility services, it is vitally important to prevent disruption to new services as businesses and consumers become more dependent upon them over the next several years.

Researchers have voiced concerns that past and current generations safety automotive electronics and traffic control devices may vulnerable to attack. These embedded legacy systems are brittle, often do not have the computational resources to support strong authentication and access control, and cannot bend under the weight a

determined and well-resourced adversary. Where legacy systems are connected, exceptional efforts and additional perimeter controls must be added to these systems to ensure security. Next generation of vehicles and even intelligent traffic control systems will feature multiple broadband connections such as cellular, WiFi, Bluetooth, and DSRC either directly or through tethered mobile devices. The attack surface for next generation intelligent transportation technologies will grow considerably and therefore so must the time and resources devoted to securing safety-critical systems.

Robotics and computing technologies are finding their way into road vehicles, as witnessed by the efforts of the auto manufacturers and Google to create "self-driving" automobiles. New technologies focused on "connected" crash avoidance and driving automation require enormous computing power to sense and perceive all manner of obstacles and driving conditions; and to plan, and execute safe maneuvers automatically with minimal driver intervention. As most crashes are the result of driver error, self-driving vehicles will likely decrease the numbers of fatalities and injuries on US roads by the thousands and millions, respectively. Robotics and connected vehicle technologies will also inspire new security management practices and technologies to assure the public of not just the improved safety, but also the greater security of these next generations of high-tech cars. To that end, the US Department of Transportation has sponsored academic and industry applied research to design security and privacy controls into of next generation cooperative crash avoidance technology utilizing vehicle-to-vehicle and vehicle-to-infrastructure Dedicated Short Range Communications.

The *National Academy of Sciences Committee on Electronic Vehicle Controls and Unintended Acceleration*, in their 2012 report, recognized that even current automotive electronic systems have already become overwhelmingly complex. The long term hope is that "roadware" will improve in quality as more resources are devoted upstream in development activities, especially those that improve security design and architecture, so that security will be "baked-in" from the beginning. As mobile devices have become more secure than traditional PCs, there is hope that these strategies may improve cybersecurity over-time in successive generations of systems. New computing technologies, techniques and even innovations in high performance computing may also allow more complex systems to be developed and most importantly, tested to provide assurance of reliability, safety and security.

Select Bibliography

Anderson, Ross. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Indianapolis, IN: Wiley Pub., 2008. Print.

Boudriga, Noureddine. *Security of Mobile Communications*. Boca Raton: CRC, 2010. Print.

Center for the Protection of National Infrastructure. *Cyber Security Assessments of Industrial Control Systems*. Rep. Department of Homeland Security, Nov. 2010. Web. 14 Nov. 2013.

Checkoway, Stephen, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." *Center for Automotive Embeded Systems Security*. University of California San Diego and University of Washington, 12 Aug. 2011. Web. 6 Jan. 2014. <http://www.autosec.org/pubs/cars-usenixsec2011.pdf>.

Chong, Jane. "Bad Code: Should Software Makers Pay? (Part 1)." *The New Republic*. The New Republic, 03 Oct. 2013. Web. 15 Oct. 2013. <http://www.newrepublic.com/node/114973/print>.

Chong, Jane. "Why Is Our Cybersecurity So Insecure?" *The New Republic*. The New Republic, 11 Oct. 2013. Web. 15 Oct. 2013. <http://www.newrepublic.com/node/115145/print>.

The Economic Impact of Cybercrime and Cyber Espionage. Rep. McAfee, July 2012. Web. 01 Oct. 2013. <http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime.pdf>.

"Executive Order 13636: Cybersecurity Framework." *Executive Order 13636: Cybersecurity Framework*. National Institute for Standards and Technology (NIST), 12 Feb. 2014. Web. 31 Mar. 2014.

Graham-Rowe, Duncan. "Road Tolls Hacked." *MIT Technology Review*. Massachusetts Institute of Technology, 25 Aug. 2008. Web. 14 Nov. 2013.

Koscher, Karl, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. "Experimental Security Analysis of a Modern Automobile." *Center for Automotive Embedded Systems Security*. University of Washington and University of California at San Diego, 19 May 2010. Web. 06 Jan. 2014. <http://www.autosec.org/pubs/cars-oakland2010.pdf>.

Langner, Ralph. *Robust Control System Networks: How to Achieve Reliable Control after Stuxnet*. New York: Momentum, 2012. Print.

Naone, Erica. "How (Not) to Fix a Flaw." *MIT Technology Review*. Massachusetts Institute of Technology, 14 Aug. 2008. Web. 14 Nov. 2013.

National Research Council. *The Safety Challenge and Promise of Automotive Electronics: Insights from Unintended Acceleration*. Publication no. TRB Special Report 308. Washington, DC: National Academies, 2012. Print.

Qualys. *The Laws of Vulnerabilities: Six Axioms for Understanding Risk*. Rep. Qualys, 2006. Web. 15 Oct. 2013. <http://www.qualys.com/docs/Laws-Report.pdf>.

Roadmap to Secure Control Systems in the Transportation Sector. Working paper. N.p.: n.p., n.d. Print. Prepared by: The Roadmap to Secure Control Systems in the Transportation Sector Working Group August 2012 Facilitated by: U.S. Department of Homeland Security's (DHS's) National Cybersecurity Division (NCSD), Control Systems Security Program (CSSP)

SBD. "2025 Every Car Connected: Forecasting the Growth and Opportunity." *GSMA*. GSMA, Feb. 2012. Web. 05 Jan. 2014. <http://www.gsma.com/connectedliving/wp-content/uploads/2012/03/gsma2025everycarconnected.pdf>.

United States. NASA. Office of the Chief Engineer. *NASA Study on Flight Software Complexity*. Ed. Daniel L. Dvorak. NASA, 05 Mar. 2009. Web. 17 Oct. 2013. <http://www.nasa.gov/pdf/418878main_FSWC_Final_Report.pdf>.

United States. National Institute of Standards and Technology. *Guide to Industrial Control Systems (ICS) Security*. By Keith Stouffer, Joe Falco, and Karen Scarfone. National Institute of Standards and Technology, June 2011. Web. 25 Dec. 2013. <http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf>.

United States. U.S. Department of Transportation. Research and Innovative Technology Administration. *An Approach to Communications Security for a Communications Data Delivery System for V2V/V2I Safety*. Research and Innovative Technology Administration, Nov. 2011. Web. 15 Oct. 2013. <http://ntl.bts.gov/lib/43000/43500/43513/FHWA-JPO-11-130_FINAL_Comm_Security_Approach_11_07_11.pdf>.

Weiss, Joseph. *Protecting Industrial Control Systems from Electronic Threats*. New York: Momentum, 2010. Print.

"What Is the ISO 26262 Functional Safety Standard?" *Ni.com*. National Instruments, 23 Feb. 2012. Web. 02 Jan. 2014. <http://www.ni.com/white-paper/13647/en/>.

Wolf, Marko. *Security Engineering for Vehicular IT Systems: Improving the Trustworthiness and Dependability of Automotive IT Applications*. Wiesbaden: Vieweg Teubner Research, 2009. Print.

About the *Connected Vehicle Technology Scan* Series

Under sponsorship from the *US Department of Transportation* (US DOT) Intelligent Transportation Systems Joint Program Office (ITS-JPO), the *Intelligent Transportation Society of America* (ITS America) is conducting the *Connected Vehicle Technology Scan and Assessment* project.

This two year scanning series of *Connected Vehicle Insight* reports will assess emerging, converging, and enabling technologies outside the domain of mainstream transportation research. ITS America seeks technologies that will potentially impact state-of-the-art or state-of-the-practice in ITS deployment over the next decade, with an emphasis on the "connected vehicle."

The *Technology Scan Series* notes trends, technologies, and innovations that could influence, or be leveraged as part of, next-generation intelligent transportation systems within the next five to seven years. The series' focus is on developments in applied science and engineering and innovation in data acquisition, dissemination, processing, and management technologies and techniques that can potentially support transportation.

To learn more about the Technology Scan Series, and to see more *Connected Vehicle Insight* reports, please visit http://www.itsa.org/knowledgecenter/technologyscan.

To learn more about US DOT's research in Intelligent Transportation Systems, please visit http://www.its.dot.gov.