Terroristas poderiam usar Teslas para nos matar

Zach Aysan, cientista especialista em cibersegurança dedicado à defesa de melhor qualidade de vida nas cidades; membro da iniciativa comunitária CityAction em Toronto, Canadá.

Data da publicação:

Maio de 2018

Uma guerra no campo da cibersegurança dos carros autônomos está chegando. O que pode ser feito para ganhá-la.[1]

Em uma calma manhã de sábado em agosto do ano que vem, subitamente, em todo o país, 12.000 sedãs Tesla Modelo S ligam ao mesmo tempo. Com o piloto automático ativado, saem para a estrada. Alguns deles chegam aos postos de gasolina locais. Outros dirigem-se a subestações elétricas. Chegam a seus objetivos em velocidade máxima. As baterias do Modelo S explodem espetacularmente com as colisões, incendiando tudo na área. A rede elétrica de Los Angeles é derrubada quase imediatamente. Surgem centenas de incêndios. A América está sob ataque.

Isso parece ficção científica. Mas não é.

Com a segurança cibernética, a primeira coisa a entender é que a Internet é ingovernável porque a localização é irrelevante e a identidade é ocultável. Isso ocorre por padrão – é a Internet, onde supostamente todos podemos conversar com todos, e não foi projetada no nível do protocolo para exigir pagamento ou identificação.

Os cibercriminosos cometem erros e são presos ocasionalmente, mas os ataques podem se vir de nações que não cooperam com instituições internacionais ou de governos estrangeiros. E fora do mundo desenvolvido, há sempre uma distinção ainda menor entre indivíduos privados e atores estatais: um especialista em segurança de software pode trabalhar para enriquecer uma empresa criminosa ou a si mesmo e também trabalhar para o serviço de inteligência de seu governo. Isso torna a cooperação internacional ainda mais difícil.

A segunda coisa a reconhecer sobre segurança cibernética é que o ataque é muito mais fácil que a defesa. Os invasores podem sondar vários pontos, como computadores ou servidores previamente invadidos, alugados com informações de cartão de crédito roubadas. Eles podem pacientemente tentar estratégias diferentes até conseguirem. Invasores talentosos podem primeiro inventar um novo método de ataque e então escrever software para sondar servidores ou o tráfego de Internet, para criar uma lista priorizada de organizações potencialmente vulneráveis e só então começar a violá-las sistematicamente.

Um defensor, por outro lado, é um alvo à espera. Seu aplicativo é voltado para o público, com URLs,[2] domínios e centros de dados que qualquer pessoa pode investigar. Ele possui um conjunto consistente e detectável de sistemas operacionais, linguagens de software e bibliotecas que são tão bem compreendidas pelos potenciais adversários quanto pela equipe interna responsável por administrá-las.

E um defensor só tem que cometer um erro: um único ponto de entrada incorretamente protegido permitirá o acesso aos atacantes. Defender todos os pontos de entrada e mantê-los perpetuamente defendidos, apesar das mudanças nos requisitos organizacionais, de pessoal e de um fluxo interminável de atualizações de vulnerabilidades para bibliotecas de software, é praticamente impossível. E mesmo as organizações que têm a competência técnica e os recursos para se defender contra ataques persistentes, como a NSA, por exemplo, correm o risco de uma violação ou vazamento crítico por alguém de dentro. A superfície de ataque é enorme e a ameaça é persistente. Se você não puder prender os invasores e os invasores tiverem o tempo que precisarem

para encontrar vulnerabilidades, a pergunta não é se o sistema pode ser violado – é quando o sistema será penetrado. E o que os invasores farão quando o penetrarem.

Quando um sistema é penetrado, um invasor suficientemente preparado pode usar software pronto para atingir objetivos prioritários. Por exemplo, um malware que ataca uma plataforma de varejo pode inicialmente extrair listas de senhas e chaves criptográficas primeiro e só depois coletar informações, como o histórico de compras. Quando um invasor começa a suspeitar que suas atividades foram percebidas, ele pode criptografar os discos rígidos do servidor comprometido antes de devolvê-los a seus proprietários. Bem-vindo ao futuro cyberpunk: é mais estranho e menos sombrio do que se poderia esperar, mas os hackers têm organizações reféns em pânico lutando para pagar resgates em criptomoedas. Eu sei. Sou um dos que ajudam essas organizações durante as crises.

Nos anos 90, como um jovem irresponsável, eu hackeei computadores ilegalmente por diversão, antes de começar a escrever softwares comerciais críticos para a segurança de empresas de telefonia. Minha experiência inicial com criptomoedas e o trabalho como whitehat[3] atraiu empresas vítimas de ransomware.[4] Eu os ajudo a entender suas opções e os passos que eles precisam dar para colocar seus sistemas em funcionamento, com estratégias de mitigação para a próxima vez. (Se você está lendo isso e precisa de ajuda, desculpe, eu só presto serviços através de contatos pessoais para evitar ser alvo de cibercriminosos. Também não tenho mais criptomoedas pelas mesmas razões.)

Mesmo os defensores mais dedicados têm algumas vulnerabilidades. Os tipos mais perigosos são conhecidos como “vulnerabilidades de dia-zero” porque são fragilidades básicas no software que ninguém – nem mesmo os programadores originais – sabia que existiam. O dia-zero mais famoso é provavelmente o Heartbleed, uma falha de segurança em uma biblioteca amplamente usada chamada OpenSSL, que depois de descoberta causou pânico em várias organizações enquanto os administradores de sistemas procuravam corrigir a falha.[5] Como exemplo da gravidade desse risco, hackers utilizando as brechas do Heartbleed roubaram as chaves de segurança de um grande sistema hospitalar dos EUA, colocando em risco a privacidade de 4,5 milhões de registros de pacientes.

A terceira coisa a ser entendida sobre segurança cibernética é que certas classes de ataques cibernéticos, incluindo a maioria das vulnerabilidades de dia zero, podem quebrar todas as instâncias do mesmo sistema ao mesmo tempo. Por exemplo, enquanto seriam necessários dois mísseis separados para destruir dois drones predadores separados, uma vulnerabilidade de software pode ser explorada por um vírus que desative ambos os mísseis simultaneamente. Foi assim que o vírus do tipo ransomware WannaCry conseguiu infectar centenas de milhares de computadores, incluindo máquinas críticas usadas em hospitais no Reino Unido.[6]

E uma vez que os atacantes controlam um sistema, pode ser muito difícil recuperá-lo. Os dados viajam muito rapidamente. Um servidor em Austin, Texas, leva cerca de 140 milissegundos para enviar dados para um servidor em Tóquio. O que isto significa é que, se você confiar no julgamento humano durante um ataque para proteger um sistema desconectando-o, talvez seja tarde demais porque os dispositivos comprometidos podem desvincular-se do controle remoto. Um telefone hackeado, por exemplo, pode ter seu programa de controle de rede modificado para passar todas as informações por meio de uma VPN[7] controlada por um atacante. E sem contramedidas previamente configuradas pelo fabricante do telefone, as instruções para atualizar o software vulnerável do telefone podem ser automaticamente bloqueadas, resultando em um dispositivo permanentemente comprometido.

Para reiterar, aqui estão nossos três preceitos:

1) A Internet é anárquica. É difícil atribuir ataques e, mesmo quando possível, a divulgação pública desses ataques revela fontes e métodos.

2) A defesa cibernética é extremamente difícil, especialmente com o tempo, à medida que as organizações mudam.

3) Algumas classes de ataques cibernéticos permitem o controle de todas as instâncias de um dispositivo e, com o planejamento correto, podem impedir o acesso ao proprietário do dispositivo comprometido.

O que nos leva à interseção entre os computadores e o mundo real.

Em 2010, uma equipe de pesquisadores descobriu o Stuxnet, um vírus escrito em uma colaboração conjunta entre Israel e os Estados Unidos para interromper o programa de armas nucleares iranianas.[8] Embora os iranianos

tenham tomado medidas para garantir que seu equipamento de processamento de material nuclear não estivesse conectado à Internet, o vírus foi levado à instalação em um pendrive. Uma vez que o Stuxnet assumiu o controle, alterou sutilmente a operação das centrífugas da instalação para que elas se destruíssem lentamente, de forma aparentemente inexplicável, sem revelar a presença do malware.

O Stuxnet ensinou ao mundo da segurança cibernética duas coisas: primeiro, os vírus não são apenas ferramentas de inteligência – são ferramentas de guerra. Segundo, se um computador está ou não conectado à Internet é mais um fator do que algo decisivo. O programa de armas iraniano pode ser bem secreto, mas é vulnerável a um pendrive; um perfil no Twitter pode ser muito ativo, mas ainda assim pode ficar offline ocasionalmente. E os dados podem ser extraidos através de uma infinidade de métodos.

Com os avanços no aprendizado por máquina, os dados podem ser analisados e filtrados localmente, de modo que uma conexão com baixa velocidade ou alta latência deixe de ser uma barreira. Por exemplo, um atacante pode usar técnicas de conversão de voz para texto nos vídeos internos de uma empresa e apenas extrair as frases que mencionem palavras-chaves críticas para sua pesquisa.

E para mostrar como é complicada é essa “floresta de espelhos”, considere que o Kaspersky Lab, o grupo de pesquisa que descobriu o Stuxnet, foi recentemente classificado pelo Departamento de Segurança Interna dos Estados Unidos como tendo ligações com o FSB russo, o substituto pós-soviético da KGB. Então, talvez a descoberta do Stuxnet pelo Kaspersky não tenha sido apenas um presente acidental para os iranianos, afinal.

Nos últimos 20 anos, uma infinidade de dispositivos de nosso dia-a-dia tornaram-se computadores. As geladeiras são agora computadores. Os relógios são agora computadores. Até mesmo coisas como sensores de uso único empregados para garantir o endurecimento correto do concreto são agora computadores.

Carros também são agora computadores.

E eles são tão seguros quanto computadores em qualquer outro lugar. Em 2015, o jipe foi hackeado e a montadora teve que enviar milhões de pendrives para reparar o software automotivo.[9] Mas por que apenas o jipe? Se hackear um jipe é tão simples quanto hackear um servidor, e os servidores são rotineiramente violados, então onde estão todos os carros hackeados? É algo um pouco parecido com o Paradoxo de Fermi.[10]

A explicação é provavelmente uma mistura de fatores. Pode ser que os pesquisadores de segurança não estejam prestando atenção suficiente aos carros. Pode ser que os blackhats não estejam motivados para atacar veículos porque o retorno do investimento não valha a pena. Pode ser que hackers tenham problemas para adaptar técnicas que funcionam em servidores e computadores pessoais para carros, porque a área de ataque é menor. Ou a resposta poderia ser mais sinistra. O Uber escondeu uma violação de dados de 57 milhões de usuários, pagando os hackers.[11] Talvez as empresas automotivas tenham feito o mesmo em silêncio.

Os engenheiros estruturais definem o limite de flexão[12] que uma viga ou ponte pode sofrer durante a carga esperada, não porque os próprios desvios sejam necessariamente inseguros, mas porque esperam que as pessoas relatem de forma confiável quando as coisas parecem erradas – e se flexões grandes, mas seguras, são rotineiras; flexões grandes, mas inseguras, podem não ser reportadas. Os engenheiros estruturais também fazem cálculos estruturais mais conservadores quando os sistemas podem não exibir fraqueza potencial antes da falha.

Você deve pensar nos computadores da mesma maneira.

Computadores atingidos por malwares sofisticados não mostram sinais de infecção. Mesmo que um ataque requeira vários estágios ou computadores intermediários, como o Stuxnet, os vírus bem programados são invisíveis. Todo software, incluindo um vírus, é apenas código, código significa dados, e dados não mudam a forma como percebemos o computador em que ele reside, a menos que o software no computador esteja preparado para perceber a mudança.

E agora vamos às coisas assustadoras. Lembre-se do nosso terceiro preceito: que algumas classes de ataques cibernéticos comprometem todas as instâncias de um dispositivo.

Pensamos em Teslas como carros, assim como pensamos em um iPhone como telefone, mas uma explicação mais precisa da realidade é que eles são apenas computadores. Um leva você pelos caminhos enquanto o outro fica no seu bolso, mas isso é basicamente a soma da diferença. Não importa o quão estranho possa parecer para

um leigo, para um desenvolvedor de software a semelhança entre os dois é tão óbvia que nem vale a pena mencionar: eles são apenas sistemas operacionais em uma peça de hardware.

O que significa que algo como o WannaCry é tão possível para o Teslas quanto para os hospitais. Ambos são hackeáveis, e em escala.

Há outra diferença, é claro: o seu iPhone não pode mover-se por si mesmo. Mas um veículo autônomo, como um Tesla, que bate em uma fábrica de produtos químicos, em um subsistema elétrico, um duto de petróleo ou posto de gasolina correndo a 200 km/h pode causar muitos danos.

Agora vamos combinar esses dois pensamentos. O que aconteceria se alguém hackeasse milhares de carros autônomos de uma só vez e os transformasse em armas?

Nada de bom mesmo.

Um dos problemas que tive no último ano e meio é como comunicar essa ideia sem soar como um maluco ou péssimo ator.

Quando percebi isso pela primeira vez há um ano e meio, informei o Ministério de Segurança Pública do Canadá. Um ano depois, encontrei-me com o parlamentar federal de meu distrito[13] para descobrir que esforços estávamos propondo para mitigar a ameaça em potencial. Descobri que não havia apenas regulamentações sobre veículos autônomos, mas também não havia planos de criar regulamentos.[14]

Depois de conversar com a maioria dos principais fabricantes de carros autônomos, incluindo Tesla, BMW, GM e Toyota, percebi que os tomadores de decisão em grandes empresas automotivas, tal como os desenvolvedores de start-ups de software, não têm uma solução mágica.

Eles sabem que precisam de tecnologia de condução autônoma para competir no mercado. Eles também sabem que estão expostos. No momento, sua principal defesa é a obscuridade de sua plataforma, o que significa que, quanto mais bem-sucedidos eles se tornam, mais vulneráveis eles estarão. Não é uma posição confortável.

Ademais, a maioria dos desenvolvedores de software realmente não pensou simultaneamente em dirigir carros e na segurança cibernética, e ele pouco sabem sobre as interfaces vulneráveis dos sistemas elétricos automotivos.

O que isto significa é que temos tempo. Essas explorações não são difíceis para organizações como a NSA, mas não são algo que o ISIS ou a Coreia do Norte sejam capazes de dominar facilmente. Estamos expostos, sim, mas o céu não está caindo. Temos tempo para criar os regulamentos e acordos internacionais corretos – se pudermos incentivar a vontade política e agir.

Existem várias maneiras de abordar a segurança de veículos autônomos, mas vamos começar com uma avaliação franca do que não funcionará:

1) Confiar em software antivírus do mercado. O único antivírus que deve ser confiável é aquele que vem com o sistema operacional.

2) Utilizar dispositivos autônomos. Vírus Bluetooth que viajam através de lâmpadas inteligentes, dispositivos de depuração em sua oficina automotiva local, ou simplesmente erros antigos (como as vulnerabilidades em rádios definidos por software), são lacunas via rádio que não podem ser garantidas. As instalações nucleares iranianas estavam isoladas e isso não impediu o Stuxnet e a CIA, então não vamos achar que estamos protegido contra ataques da RPDC.

3) Revisão de código. Quando os governos ocidentais constroem seus sistemas de segurança, eles geralmente dependem de peças de hardware construídas na China. A “segurança” desses componentes é certificada pela inspeção estatística do código embarcado em um punhado de amostras.

Mas enquanto o governo britânico pode revisar o código de equipamentos de rede chineses em Banbury, Oxfordshire, se houver uma guerra total com os chineses eles provavelmente terão que substituir tudo isso por material americano. Porque a menos que você inspecione cada componente individual que você recebe de um fornecedor, e ele tenha um código fisicamente inalterável, você não tem ideia de qual código está realmente sendo

executado em seu sistema.

O principal impedimento contra um fornecedor chinês, como o fato de a Huawei inserir malware em seus produtos, é a perda de prestígio comercial - considerações de mercado que são discutíveis quando se trata de estabelecer prioridades estratégicas na guerra.

4) Confiar nas empresas automotivas. Equifax e Ashley Madison estavam em segurança. Até que deixaram de estar. A segurança nacional não é algo a ser confiado a corporações e certamente não a corporações de países com histórico ruim de segurança cibernética, como a China. O capitalismo recompensa invenção e risco, não mitigação de risco de longo prazo.

5) Certificação de componentes individuais ou veículos. A regulamentação detalhada e prescritiva e a certificação individual são muito lentas em relação ao ritmo acelerado do desenvolvimento moderno de software. Nossas corporações mais seguras atualizam seu código várias vezes por dia. Este não é apenas um artifício correlativo de empresas de tecnologia bem administradas – é causal. O primeiro agente a encontrar uma vulnerabilidade geralmente é a organização responsável pelo serviço ou dispositivo e ele trata de corrigir o problema o mais rapidamente possível.

Em vez disso, os regulamentos devem ser funcionais. Por exemplo, uma máxima como “dados nunca devem ser lidos por um dispositivo intermediário de rede” ou “nenhuma ação executada no computador de acesso deve alterar o estado do computador de controle” de modo que multas e recompensas de segurança não sejam arbitrárias, mas empresas automotivas possam ainda competir na velocidade de seu avanço tecnológico.

6) Permitir que o mercado leve em conta ataques cibernéticos em larga escala como parte do cálculo do seguro automotivo existente. Com todo o respeito, as seguradoras não têm nem os dados nem a experiência para efetivamente estimar esses riscos. A distribuição de Poisson[15] é maravilhosa, mas os vírus de computador invalidam todas as classes do mesmo sistema ao mesmo tempo – portanto, essa técnica estatística não deveria ser usada como base para avaliar ou prever ataques. Sem independência estatística, as vulnerabilidades dessa escala não podem ser precificadas com precisão porque é impossível obter probabilidades precisas da ocorrência de um evento-surpresa. Engenheiros civis projetam para uma tempestade destrutiva em 100 anos. Como seria um ataque cibernético destrutivo em 100 anos? Ninguém sabe.

7) Esperar que veículos autônomos sejam usados em ataques de pequena escala antes de elaborar a legislação. Se esperarmos por tal ocorrência, a legislação resultante provavelmente será direcionada para ataques de pequena escala e não focada no risco maior. Nossa primeira preocupação deve ser sobretudo a segurança nacional (hacks em larga escala de frotas inteiras), não apenas alvos específicos (hacks de carros individuais).

Então, o que funcionaria? Políticas eficazes devem começar com o reconhecimento de que os governos não conseguirão regular de maneira inteligente o problema. Um esforço bem financiado e de código aberto com recomendações claras será a maneira mais eficaz de proteger o veículo sem motorista.

1) Os profissionais de software devem educar e encorajar os engenheiros elétricos e mecânicos a apresentar propostas que ajudem as empresas de veículos autônomos e os governos a proteger o público.

2) As comunidades de inteligência e de controle de armas devem ajudar na elaboração de acordos internacionais para tornar ilegal o ataque cibernético de sistemas civis durante a guerra sob leis internacionais - e precisamos de especialistas jurídicos para elaborar regulamentos de referência que países menos avançados tecnicamente possam usar como base.

3) Nossos acordos comerciais devem refletir a natureza mutável de nossa interdependência. A China anunciou recentemente que empresas automotivas estrangeiras, como a Waymo, do Google, não podem fotografar cada centímetro quadrado de suas estradas devido a preocupações com a segurança nacional. Mas os veículos autônomos exigem câmeras e uma conexão à Internet para operar, portanto este regulamento terá o efeito de manter veículos autônomos de origem estrangeira fora das estradas chinesas.

Os chineses entendem a ameaça que os veículos autônomos representam e querem limitar sua exposição – ou estão usando as preocupações de segurança nacional para mascarar uma tentativa de incubar sua própria indústria de veículos autônomos.

De qualquer forma, os chineses entendem claramente algo que já não é percebido por muitos ocidentais: o comércio liberalizado é muito bom, mas a segurança nacional é mais importante. Sem a cooperação internacional sobre a regulamentação de veículos autônomos e acordos comerciais simétricos com disposições severas contra violação, não devemos permitir o acesso de estados não-amigáveis a nossos mercados de veículos autônomos. (Também não devemos permitir componentes desses países.) Os chineses entendem isso. Nós deveríamos também.

4) Qualquer dispositivo permanente, não militar, que possa voar, dirigir, andar, disparar ou nadar autonomamente deve conter um módulo de segurança padronizado. O poder do mecanismo de propulsão, assim como os computadores e sensores que comandam o dispositivo autônomo, devem conectar-se através deste módulo de segurança. E se isso não for possível devido à natureza do sistema de propulsão (por exemplo, dispositivos com foguetes químicos), um sistema de desativação de emergência deverá estar presente.

O dispositivo não deve ser acionável ou operável a menos que o módulo de segurança esteja presente e o dispositivo não consiga acessar seu próprio módulo de segurança de forma alguma. (Dispositivos militares não devem estar sujeitos a estes regulamentos.)

Os países devem ser capazes de especificar quais módulos de segurança são aceitáveis em seus domínios, e você esperaria que a maioria dos países desenvolvessem seu próprio módulo ou confiassem apenas em dispositivos com módulos de segurança de seus aliados mais próximos. Mas os dispositivos podem ser projetados para aceitar vários módulos, e qualquer um deles pode iniciar os procedimentos de segurança. Dessa forma, o movimento autônomo não precisa cessar ao cruzar uma fronteira.

(Embora seja necessário ter cuidado para garantir que os dispositivos sejam realmente independentes. Qualquer módulo de segurança deve ser capaz de desligar o dispositivo ou colocá-lo offline, mesmo se outros módulos tiverem comandos para agir de maneira mal-intencionada. A segurança deve ser aditiva, não multiplicativa.)

Os módulos de segurança devem poder comunicar-se através de múltiplos canais; incluindo satélite, rádio, LTE e até luz pulsante via câmera embarcada. Dessa forma, utilizando chaves e certificados criptográficos, os governos poderiam ordenar comandos de emergência de dispositivos autônomos (através do módulo de segurança), como “desligamento em 5 segundos” ou “cessar atualizações de software até instruções adicionais”. O módulo de segurança deve ser capaz de desligar a energia e o computador de controle em situações de emergência.

E, finalmente, para garantir a integridade do próprio módulo, independentemente do código presente no computador do dispositivo autônomo, o dispositivo não deve interferir no módulo de segurança ou no módulo de segurança de outros dispositivos autônomos.

(A maneira mais direta de proteger o módulo de segurança seria empregar a arquitetura tradicional do computador com uma conexão unidirecional para o computador principal e um retorno a um chip ASIC imutável com seu próprio par de chaves criptográficas e uma conexão direta com a fonte de energia.)

5) Sistemas redundantes padronizados são outra salvaguarda. Se a energia for removida do computador principal de um dispositivo autônomo, o veículo ainda poderá pousar ou estacionar com segurança. Desligamentos totais devem estar sempre disponíveis, mas não devem ser o primeiro recurso durante um ataque cibernético.

6) Para que um veículo autônomo vá mais rápido que um limite de velocidade predefinido, ele deve solicitar permissão para isso ao módulo. Dessa forma, você poderia chegar ao hospital rapidamente durante uma emergência, mas os governos poderiam limitar quantos veículos velozes são permitidos de uma só vez. Por que isso é importante? Porque um carro que anda com metade da velocidade tem um quarto da energia cinética – reduzindo a possibilidade de uma explosão de bateria em uma colisão.

7) Isolar o computador de controle. A maioria das unidades de controle eletrônico automotivo se comunica usando a rede local do controlador (barramento CAN) - um barramento central não criptografado e não autenticado. Em todo o mundo, os desenvolvedores de software cospem seu café ao ler a sentença anterior. Não permita que computadores de controle leiam dados diretamente do barramento CAN. E não conecte o computador de acesso do veículo ao computador de controle.

Se precisamos obter o estado do barramento CAN, ele deve ser feito através de um módulo intermediário que converte os sinais em um de uma série finita de estados enumerados. (E enquanto estamos nisso, devemos criar

um acordo internacional para abolir o barramento CAN e substituí-lo por algo mais seguro.)

8) Tome nota da Apple e use um Enclave Seguro dedicado a tarefas críticas de segurança, como a atualização do software do computador de controle. Assim como no módulo de segurança, o enclave seguro deve ter métodos de comunicação de reserva para desativar o carro com segurança durante uma vulnerabilidade crítica. Idealmente, o enclave seguro deve ser projetado com vários conjuntos de chips de diferentes fabricantes, a fim de mitigar a espionagem industrial ou vulnerabilidades como o Meltdown da Intel.[16]

9) Não confie na rede. Não confie em cadeias de DNS ou certificados. Empregue pinagem de IP e fixação de certificado com estratégias de reserva. Não confie apenas em HTTPS. Protocolos e cifras não são perfeitos e ataques de degradação de protocolo são muito fáceis. Use criptografia do lado do cliente, além de HTTPS, e use chaves realmente grandes.

Envie cada carro com sua própria chave One Time Pad (OTP).[17] E crie a OTP com várias fontes aleatórias seguras em computadores nunca conectados à Internet em local seguro e protegido usado para assinatura de código. Não deve ser fisicamente possível ler o mesmo bit duas vezes a partir da OTP. A revisão final do código deve estar em computadores nunca conectados à Internet. E nunca, nunca, permita o acesso SSH a qualquer veículo autônomo – mesmo aqueles em desenvolvimento.

10) Empregue criptografia e assinatura de código e dados em tudo que é possível, incluindo dados em memória volátil. Todas as atualizações no computador de controle devem ser criptografadas, assinadas e duplamente verificadas. (O checksum deve ser compartilhado com os governos e transmitido para o módulo de segurança.) Se o computador de controle não puder verificar a assinatura ou a soma de verificação da atualização de software com o módulo de segurança, o computador de controle deve desligar o veículo com segurança.

Há outras maneiras de os governos colaborarem para mitigar parte do problema: os governos devem se unir para criar um sistema de recompensas, a fim de incentivar os pesquisadores de segurança. (Recompensas pelo controle remoto real poderiam variar entre US$100 e US$10.000 por dispositivo, dependendo de fatores como a velocidade máxima atingível.) Os governos também devem concordar em impor multas severas e sentenças de prisão por fabricação, venda ou fornecimento de produtos eletrônicos automotivos falsificados.

E nos países avançados deve-se aumentar dramaticamente o financiamento para as unidades de guerra cibernética e encontrar uma maneira de expandir as reservas cibernéticas para envolver especialistas em computação no setor privado. Para aqueles que não puderam obter uma autorização de segurança, encaminhá-los para iniciativas de código aberto e think-tanks.

Finalmente, devem aumentar o financiamento para pesquisas em chips especializados em segurança, não em desempenho (para que vulnerabilidades como Meltdown e Spectre sejam menos prováveis) e criar regulamentos que incentivem linguagens de programação mais seguras, como Rust, em detrimento das inseguras ou de comportamento indefinido.

Há uma linha de esperança em todo o trabalho que temos que fazer: a natureza da ameaça do veículo autônomo pode finalmente trazer a vontade política, os incentivos econômicos e as ideias de que precisamos para proteger nossos sistemas de computadores do mundo real. E com um pouco de sorte, poderíamos acordar décadas a partir de agora e falar com espanto sobre os eventos cibernéticos do início dos anos 2000, como fazemos com os incêndios químicos nos rios em meados do século XX.

=-=-=-=-=

1Publicado originalmente em *http://www.weeklystandard.com/terrorists-could-use-teslas-to-kill-us/article/2011171*. Reproduzido com permissão. Notas de rodapé são da editoria da poliTICs.

2Sigla de “Uniform Resource Locator” - designa endereços mnemônicos de serviços Web na Internet.

3Expressão que denota um hacker bem-intencionado.

4Ataque a computadores ou redes que bloqueia os serviços ou sistemas até que um resgate seja pago ao atacante.

5Ver, por exemplo, https://en.wikipedia.org/wiki/Heartbleed

6Ver https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

7Sigla de “Virtual Private Network”, conexão ponto-a-ponto criptografada entre dispositivos de rede.

8Ver *https://pt.wikipedia.org/wiki/Stuxnet*

9Ver *https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway*

10Ver *https://en.wikipedia.org/wiki/Fermi_paradox*

11Ver *https://www.wired.com/story/uber-paid-off-hackers-to-hide-a-57-million-user-data-breach*

12Ver *https://en.wikipedia.org/wiki/Deflection_(engineering)*

13No regime parlamentarista do Canadá o voto é distrital.

14Ver *http://www.cbc.ca/news/business/autonomous-vehicles-self-driving-cars-uber-google-general-motors-1.4287591*

15Ver *https://en.wikipedia.org/wiki/Poisson_distribution*

16Ver *https://en.wikipedia.org/wiki/Meltdown_(security_vulnerability)*

17Ver *https://pt.wikipedia.org/wiki/One-time_pad*

Categoria:

- poliTICs 27