SECURITY NOW!

Transcript of Episode #497

Vehicle Hacking

Description: Leo and I discuss the week's tamer-than-usual news; then we host a terrific interview of the team (recently featured on Sunday's "60 Minutes") who have been working with DARPA to address the challenge of hardening high-tech networked vehicles - autos and UAVs - against malicious hacking attacks.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-497.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-497-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here, and we have a great show planned for you, not only some security news, but a little later on in the show an interview with the two guys who hacked the car Lesley Stahl was driving in during "60 Minutes." We're going to talk about vehicle hacking and why it's going to be and may already be a huge problem. Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 497, recorded March 3rd, 2015: Hacking Vehicles.

It's time for Security Now!, the show where we protect you, your privacy, and your security online with this guy right here, Mr. Steve Gibson. Yes, sir. You going to do the "live long and prosper" for us in honor of...

Steve Gibson: Oh, I have a little mention down in our Miscellany today because I had an encounter with Gene Roddenberry.

Leo: We have an amazing show for you. Coming up a little later on, we're going to talk to two guys who are kind of heavy-duty researchers in what they call, what, physical security?

Steve: They called it "cyber-physical security."

Leo: Cyber-physical security.

Steve: That is, attacks against physical things that we care about, like driving down the freeway in our car. Yes, I mentioned a couple weeks ago when "60 Minutes" had that segment showing a wireless hack of a car that the producer, Lesley Stahl, was driving. And the person who was in the passenger seat was part of this team who said, "Okay, now, Lesley, pull up and stop in front of this line of orange cones." So Lesley pulled up to them, put her foot on the brake, and nothing happened. And I...

Leo: That's scary. She plowed right into the cones.

Steve: Yeah. And it's interesting because, as I mention later in the show - I don't know how I know that I mention it later in the show, but I'm psychic [laughter].

Leo: To pierce the veil of secrecy, we recorded the second half first because we wanted to let Lee and Pat, our guests, get going in their life.

Steve: I got out of my time machine and came back to meet with Leo.

Leo: We edited this to be in a different order than the actual recording order. In standard podcast order, right, with the top-of-the-show news and then our interview. But it's surprisingly jarring, when all our lives we've pushed down the brake pedal, and the car slows down and stops. to push down the brake pedal and have the car not stop. And it turns out that I was just naively assuming there would still be a physical backup; that we wouldn't entirely be relying on fly-by-wire technology. But these guys explain in the future that in fact what's happened is pressures for lightening the weight, lowering the cost, adding features. All of these things have sort of come together to turn these cars into rolling computer networks with upwards of 50 different so-called ECUs, the individual computers scattered around the car. So, for example, there'll be DC supplied to the whole array of lights along the back of the car. And the CAN bus runs over. And over the CAN bus comes the instructions for which lights to turn on and off. There isn't, you know, used to be there was a big wiring harness.

Leo: A physical wire that goes to the back, yeah, point to point.

Steve: Right, where the big bundle of cables went back to all the specific lights that you want. Now they just send it DC.

Leo: But that makes sense because we're used to that with networking. It makes sense. Have a central bus and send signals to different components.

Steve: Yup, and in the future, one of these guys will say, yes, the CAN bus is the best network that the '80s could ever come up with.

Leo: That's a pull quote from the future, ladies and gentlemen. I look forward to that. But we have some other security news to talk about, too, I know.

Steve: Yeah. So this week we're going to talk about another SOHO router backdoor; an interesting but not-yet-perfect web-based encryption service, which is a little bit of a lesson for our audience I thought they'd get a kick out of. And I just wanted to make a note of a blurb that came up during the week about creating an Internet service and having your life threatened. And then two little blurbs about the evolution of DNS and search, some miscellaneous stuff, and then our main feature of the week, which is this report on hacking vehicles from the guys who have done it. So otherwise there wasn't really much horrifying security news. So instead, the entire topic of the show, having your car hacked is somewhat horrifying.

Leo: Okay, Steve. Let's see. What have we got?

Steve: So this generated a lot of Twitter traffic because it seemed really severe. Upon closer inspection, it's real, but it's sort of a corner of the 'Net. Still I wanted to bring it to people's attention, if for no other reason than to reinforce the importance of something we've been long saying.

What was discovered by a couple of guys who reverse-engineered the firmware that they found on their own router - and these are security researchers who presented their findings to the International Conference on Cyber Security and Cyber Law a couple weeks ago, on February 21st, was that there was an undocumented backdoor because in their firmware the password "super," S-U-P-E-R, for both username and password was burned into, embedded and hidden in the firmware, alongside whatever username and password the user would assign. So that meant that this particular router they had would respond, with no other documentation about this anywhere, if you used "super" twice for username and password. And they then did a partial scan of the Internet and found more than 200,000 routers sitting on the 'Net into which they were able to log using "super" for the username and password.

Now, the good news is none of these are big brand-name routers. Well, although Realtek is a name we recognize, although they're big for network interface cards, and TRENDnet is a name I know. But Digicom, Alpha Network, Pro-Link, Planet Networks, Bless, SmartGate, and Blue Link are among the model numbers, makes and models of routers that they have found. And these are all - it looks like there's some sort of an underground router firmware that's being shared, like these are sort of off-brand routers. The hardware has become commoditized. So you build commodity hardware, stick your logo on the front, and then you get this not-sure-where-it-came-from firmware that happens to have "super" as the username and password. So maybe the bargain basement router is making you vulnerable.

The problem is that this particular router, as evidenced by the fact that at least 200,000 of them have been found, has the WAN admin enabled by default. Which means all of those routers they were able to log into. And right now today, those are all completely open to remote exploitation. So, again, you absolutely want to make sure, not only that your router doesn't respond to "super" as its username and password, but that you have explicitly disabled WAN interface. We know that that's not complete protection because we've been covering stories where routers will still respond to magic Ethernet packets or to funky packets on some strange port, because the designers of the router wanted to have admin access, sort of off the record. Which is just not a safe thing to do.

Leo: Did they say the brand?

Steve: Yeah, there's a bunch of these. It's on that next page of the show notes, Leo, Digicom, Alpha Network, and so forth. So they've identified...

Leo: Oh, yeah. Oh, yeah.

Steve: What they did was they scanned the Internet and found all of these routers. When they logged in, the routers said, yeah, I'm a Digicom.

Leo: Hey, I'm Super Super.

Steve: Yeah.

Leo: Hey, Super Super.

Steve: Exactly.

Leo: Well, if you turn off WAN administration, then they'd have to have physical access. So that at least - but there's no other way to mitigate that; right? I mean, that's just built into the firmware.

Steve: Correct. And so, again, the other lesson here, I think, is that what we're seeing is that router firmware is too much on the frontline to leave it up to the manufacturers. I'm here to say now in 2015, given that there is well-documented, well-scrutinized, third-party, open source replacement firmware, that's what you should be using. The Tomato firmware and the DD-WRT firmware, and there's another one, too. I can't think of it. Of course Astaro has their platform that you can load into a small, like a fanless PC form factor, and that would make sense, too.

Leo: And, you know, we should also give a pat on the back to ASUS because they actually make - they sell routers with DD-WRT in it.

Steve: Perfect, yes, yes.

Leo: So, I mean, the problem with DD-WRT is it supports a subset of all routers, so you've got to make sure you have a compatible router. Or, if you buy one of these ASUS routers, it's in there, and upgrades are not coming from ASUS, they come from DD-WRT.

Steve: And I like their ad. They said, "You spoke, we listened."

Leo: Yeah.

Steve: Which is to say, you know, we're going to sell you a router specifically so that you are in control of the firmware. And again, I think the router is our first line of defense on our networks. And we're seeing too many examples of both deliberate and inadvertent exploitation of the trust that we put in the firmware. It's time to revoke that trust, to say nope, I'm only going to run something that I know, essentially, as my interface to the Internet.

Leo: But this is another case of the race to the bottom. You're selling $30 hardware.

Steve: Yup.

Leo: It's commodity hardware. You're not going to put any effort into it. Nor, and this is even worse, are you going to patch it. You're just going to assume, eh, they'll buy a new one.

Steve: Right. So those 200,000 people are probably toast. They're people who did just buy some off-brand router at Staples that they had on sale because they were trying to move a bunch of them. They said, oh, this is probably as good as anything else. Well, no.

Leo: Do you like DD-WRT? Or do you have a preference?

Steve: I don't. I like, I actually like what Astaro has done, just because it's got, for our audience, for the more technical audience, it is a very powerful solution.

Leo: Yeah, that's a great way to go. But that requires having PC hardware and putting the operating - yeah, it's complicated.

Steve: Multiple NICs and so forth, right.

Leo: And there is OpenWRT. Maybe that's the other one you were...

Steve: Oh, I think that was the other one, yes, yes, right. And Tomato, a lot of people like the Tomato firmware.

Leo: I've used Tomato. I love Tomato. But I think that's an even smaller subset of supported routers.

Steve: Yeah.

Leo: Yeah.

Steve: So anyway, okay. Next up, if you can show the picture of the week, Leo, I got just the biggest kick out of this.

Leo: I love it.

Steve: This is actually a screenshot of Notepad after I decrypted a file under a new online service. This is the GRC CipherSuiteOrdering.txt. That's the file I've talked about which is just readable ASCII. It'll say, you know, ECDH, Elliptic Curve Diffie-Hellman underscore CBC, you know, so forth. I mean, it's legible, readable ASCII. Except not in this screenshot. So here's the problem. This is a neat service that needs a little more work. It's called InstantCryptor, I-N-S-T-A-N-T-C-R-Y-P-T-O-R, InstantCryptor.com. The guys look like they've got their hearts in the right place. It is a browser-based, on-the-fly, very simple to use, encryption/decryption service. It's all done in the browser, and in the FAQ they explain it, and it all makes sense. There's no reason why you cannot do this securely in today's JavaScript in the browser.

And in their documentation they explain: "The password will be hashed with the SHA-256 algorithm. The mode for encryption is 256-bit Rijndael/AES in CBC mode." All of our listeners know what all that means. "The file blob [meaning the resulting blob] will be read as an array buffer and fed into the encryption function. The result is then uploaded to the chosen cloud service. The uploaded files are displayed in the tool, and decryption works accordingly. The code of the main JavaScript file is unminified, and the interested developer can have a look at it." So I like that. They're saying, here it is. We're not obscuring the code. We know you're going to want to see this for yourself. So there it is. And then they conclude, saying: "The main action happens in the last two functions at the end of the file."

So what they have is this browser-based service, InstantCryptor.com, that runs under Firefox or Chrome, because you're going to have a little bit of browser centricity because it's using code for the specific JavaScript implementation where there may be differences between browsers. And, you know, Microsoft is always lagging, so it's not running on theirs yet. And it can target either Dropbox or Google Drive.

So I went there. I said I want to use Google Drive. And it took me to Google's authentication permission screen and said, "Do you want to allow this site to have access to your Google Drive?" And I said yes. So at that point it switched me back to there. And they said, "Enter a password you would like." So I typed in a password. And then it said, "And would you like to browse for files?" I clicked on the "browse," and I chose the CipherSuite text file that I just had sitting around, and I said, go, baby. And in no time at all, it had encrypted it and uploaded it.

I then went over to Google Drive, and they had created an InstantCryptor folder, where all of my encrypted files from this service would live. And it was absolute gibberish. And then I thought, huh. Okay, cool. And so I then reversed the process. But I was curious because nowhere did this mention authentication. And so I used a different password to decrypt the file, and the results are what you saw. It's a wonderful-looking unicode catastrophe.

So that's a problem because what that means is that they're doing encryption, but not authentication. And we've talked about why this is important for years. All good encryption not only encrypts, but authenticates, because although the nave user could be forgiven for thinking, well, if they don't know the password, then they're going to get gibberish. But the problem is this also doesn't detect any either accidental or malicious modification. And we know there have been attacks on SSL where authentication was not done correctly, where information leakage does occur. So it's theoretically possible for someone to mess around with an encrypted file that does not have an authentication wrapper around it.

The proper thing to do, what these guys need to add, and I'm hoping somehow this message will get to them, is that after they encrypt, then they authenticate. That is, they take the same password, or maybe a hash of that password, and they run it through, they use that to key an HMAC, which generates a message authentication code, and they add that to the end of the file. That's what they upload. Then when you go through decryption, the first thing they do before they decrypt is they authenticate. So they take the password you provide. They're already doing an SHA-256, but that's to get the encryption key. They hash out again in order to get the HMAC, the authentication key. And so they process the file, the encrypted file, as they did before uploading it. They do it again before decrypting it. And they verify that they get the same message authentication code as is tagged onto the end of it.

When that verifies, that tells them two things: There has been no modification of the file, either inadvertently in transmission or by any malicious agent. So it verifies the contents and that the password is correct. So what they should have told me when I typed in the wrong password is, uh, sorry, that password is not the one used to encrypt this file. And hopefully they're also doing some password hardening, that is, they didn't say so, they just said SHA-256. If they're simply doing a hash of my password, then that's a problem because it would be possible to start doing a brute-force attack if they're only doing one SHA-256, and just look at the beginning of the file until they guess something that makes the beginning of the file look correct.

So anyway, it's sort of a nice little object lesson. Simple, clean, cool service. I imagine people might like to use it because you could, if you shared that folder, you could then send your password or have that known through some other channel, get it to somebody, and say, hey, I just uploaded a file, I just encrypted it because this particular one we care about. You know the password. And then they could get it. Or you could take it and email it to them, whatever. Anyway, just sort of a cool service, but missing a few important things for a service like this.

I'm sure you saw the news, Leo, that ISI has now essentially made a death threat against Twitter founder Jack Dorsey. In an online post which has since been removed on the "Just Paste It" site they said: "Your virtual war on us will cause a real war on you." And then they had a doctored photo of Jack with gun sights centered on his face. They're unhappy, that is, ISIS is unhappy - ISIS is apparently behind this - because Twitter is being sociably responsible and taking down the terrorist Twitter accounts as quickly as they find them. And I guess Twitter has hundreds of people whose full-time job is responding to reports of this and verifying them and then taking fraudulent accounts down. And it's just sort of a sad wakeup call that that's, you know, on one hand it demonstrates, I guess, that the Internet is as real as everything else in the world today, but it's also unfortunate.

Leo: Also it demonstrates how media-savvy ISIS is.

Steve: Yes.

Leo: And that's really the real story here is that they have, unlike other terrorist groups, figured out exactly how to play the game.

Steve: Yeah, well, they have camera crews and lighting consultants. And it's just like, oh, lord. Yeah, wow. Now, this is interesting. This is not a security story. But we talk a lot about DNS. And I thought this was really just curious. And that is the news that Google has just paid $25 million for the entire .app top-level domain.

Leo: That's a valuable domain, if you think about it.

Steve: Oh, my goodness, *.app.

Leo: All you have to do is search for all the apps whose domain name is somethingapp.com to realize how valuable that is.

Steve: Right.

Leo: Almost every app on iOS and Android has a website that is somethingapp.com.

Steve: So then it turns out this is not that new because Amazon last year paid 5 million for .buy.

Leo: Oh, ho ho ho ho.

Steve: Also 2.2 million for .spot. And until Google's 25 million purchase - and these are auctions, by the way. So Google outbid everybody else who wanted *.app, essentially.

Leo: Who gets the money from that?

Steve: That goes to ICANN to support all of the things that they're doing.

Leo: They're doing good.

Steve: Yeah.

Leo: Wow.

Steve: And then Dot Tech, a company called Dot Tech paid 6.7 million for .tech, T-E-C-H.

Leo: We should mention that this all - ICANN created this policy maybe a year ago, something like that, where you could register a TLD, any arbitrary TLD. And if you had sufficient, I don't know what, funds, I guess, you could own it.

Steve: Well, yeah. And so this is what I want to talk about because, for example, Amazon has applied for a total of 76.

Leo: Wow.

Steve: They're called gTLDs, generic Top Level Domains. And Google has applied for a total of 101. So, for example, Google also wants .blog, and they want .cloud and .search.

Leo: Wow.

Steve: And so here's two interesting tidbits from Google's application to ICANN. For .blog they said: "Our application for the .blog TLD describes a new way of automatically linking new second-level domains to blogs on our Blogger platform. This approach eliminates the need for any technical configuration on the part of the user and thus makes the domain name more user-friendly." Okay. And I'll just read the second one, then we'll talk about this. They also want .dev. So for .dev they said: "Second-level domain names within the .dev proposed gTLD are intended for registration and use by Google only, and domain names under the new .dev gTLD will not be available to the general public for purchase, sale, or registration. As such, Google intends to apply for an exemption to the ICANN Registry Operator Code of Conduct as Google is intended to be the sole registrar and registrant."

Now, this represents a complete change in the way - as demonstrated by the fact that Google is going to need an exemption to the ICANN policies. Because Google is saying, now, traditionally what would have been done is it would have been dev.google.com. But Google has a lot of money, and ICANN has said we'll sell global top-level domains. So Google says, oh, good. We want .dev just for ourselves. We're going to use it internally; .dev will never be available for anyone else. That's Google internal top-level domain name. And they're saying this similarly about .blog. We're going to buy, we're going to outbid anybody else for .blog and then use our blogging platform to automatically hook blogs to the .blog TLD. So, for example, I have a WordPress blog, blog.grc.com. And I've got my DNS set up so that blog.grc.com redirects to the IP address at WordPress so people can get to the blog.grc.com with that domain name, but it actually goes to WordPress. And so what Google is saying is, eh, we'd just like to buy blog, please, and then it'll be Steve.blog will be the name for this.

So anyway, I just thought it was interesting. I mean, as you said, Leo, it's generating a lot of money. It's also taking potentially powerful global top-level domains off the market forever. Now, I don't know that it's a bad thing. They've never been on the market before. And the counterargument is will anything ever really replace .com? Consumers are so invested in .com being where you go that, you know, that's always the domain you want. And when I see things that are, like, .net or .org, I think, wow, I hope people remember that it's a different suffix on the end of that name.

Leo: Yeah. No, I think we've seen this before. Remember when Sex.com sold for some ungodly amount of money and was never really worth anything.

Steve: And XXX, triple-X, I think also did.

Leo: Well, because everybody thought, oh, the generic, you know, TV.com went for a lot of money because the generic .com name is going to be somehow hugely valuable. And then I think what happened is nobody really - who enters in dot anything?

Steve: Exactly.

Leo: If you want to go to a site, if I want to go to GRC, I'll just type GRC into Google, and it'll bring me there.

Steve: Exactly.

Leo: So I don't even know how relevant that all is anyway, anymore.

Steve: Agreed. And in fact, that takes us perfectly into the second story because Google has also announced a skunkworks project that is apparently not yet public. Well, I mean, I know it's not public, but it is private. And that is that they have been building something for some length of time called a "Knowledge Vault." And New Scientist has a story explaining that apparently all the big companies we know about - Apple and Microsoft and Google and, like, the main big movers on the Internet, they're all doing this. We see this ridiculous falling cost in the cost of storage and the ability now to send bots out and suck in the whole Internet, which of course is the way Google's index works.

They've announced something called KBT, knowledge-based trust. There's an interesting paper, a whitepaper that I've linked to in the show notes, which is full of summation symbols and differential calculus and, I mean, it is really toe-curling academic stuff. But what they're proposing to do, in the same way we've talked about before, for example, the famousness of Google, the thing they did that immediately took us all away from Alta Vista that we were using at the time, is to rate a site based on the quality and count of links linking to that site. That was a simple means for them to get a metric on the quality of a given page on a site.

Now Google is claiming that they've noticed that the Internet is full of misstated untruths and fallacies. And they're not wanting to rank as highly sites containing untruth as sites containing true things. And Google believes they have a way to mechanistically tell the difference.

Leo: Wow.

Steve: Which is an amazing claim. And so they're saying that they have this knowledge vault which contains 1.6 billion facts. This automated knowledge vault has 1.6 billion facts. And of those, 271 million are rated as "confident" facts. And that means that Google's model ascribes a more than 90% chance of that being true. And apparently it builds this huge cross-referenced fact base, knowledge-base structure. And in the same way that Google said a while ago, and we've talked about it recently, that they're going to add whether sites are using security or not as an additional hint to the ranking engine in Google searches, they are saying, this is what they are proposing to do, that they're going to start affecting ranking results based on this knowledge vault's determination of the factualness, the truthfulness of individual sites. And the world is changing.

Leo: Well, and it also raises some issues. I mean...

Steve: It does, yes.

Leo: I mean, facts is facts. But we may be not in agreement on whether something's a fact or not.

Steve: Really. Now, it's interesting because it used to be that you uprated yourself, if you were an SEO-crazed person. I've never just bothered. I just figured the only way to do SEO right is just have a site with high-quality content, and don't worry about it. But of course that's not why you have an SEO department whose whole goal is to give you a high search order ranging. But traditionally, you know, it was arranged to get inbound links and make your server run fast because it's also heard, for example, that Google is going to tend to increase the ranking of sites that are served faster because they want to encourage that, and they think that maybe the speed of the server is connected to the quality of a site. I mean, they're looking for "signals," as they call them.

And so I agree with you, Leo, the notion of Google, who, I mean, Google is now a verb. You don't go to a search engine to look something up, you "google" it. And Google is our portal. It's the way we view the Internet. Anything I want to find out, Google is my search engine. I put the term in. I put a sentence in. Sometimes I ask it a question. And there's, like, there's my answers, right there. So it's incredibly valuable. And now they're adding this new twist which is, eh, we're going to try to downgrade the sites that are BS.

Leo: Some stuff is verifiably factual. But there are people who say we didn't land on the moon. And...

Steve: I don't think we're going to be hearing from them anytime soon, Leo.

Leo: Well, but, I mean, I don't know.

Steve: They're going to disappear.

Leo: I'm sure they believe that what they're saying is factual. Is it up to Google to decide that?

Steve: And, exactly, isn't that interesting. Because the other controversial thing that has been reported is that when I pull up a Google search, and for example when Jenny pulls one up, we get different results.

Leo: Right. Because you're logged in, yeah.

Steve: Yes, we're logged in, and we have relationships with Google, and Google knows something about us and tries to give us the results we want. Well, the problem is that tends to increase the fragmentation of sort of the community because it's sort of like a self-fulfilling prophecy or self-reinforcing selection.

Leo: Yeah. I mean, what if they had done this in the time of Galileo? And, well, everybody knows that the Sun revolves around the Earth.

Steve: Yeah, yeah.

Leo: That's a fact, isn't it?

Steve: Epicycles. Epicycles, yeah.

Leo: Yeah. So that's a fact. So facts...

Steve: Yeah, who is this weird guy? Good thing that he doesn't show up in the...

Leo: Can't find him in the Google search.

Steve: Yeah.

Leo: I think that is a typical engineering point of view, that there are facts...

Steve: Boy, if you look at this paper, though, your eyes will cross.

Leo: I did, I did, I looked at the math. Holy cow.

Steve: Oh, whew, baby.

Leo: Yeah.

Steve: And I will say, independently, the concept of an automated knowledge vault is intriguing. I mean, we saw what IBM's Watson could do on "Jeopardy!." It's like, okay, I'm giving up and going home. Thanks anyway. It was just shocking to see what it could do. And so I think, with processor power becoming what it is, and storage becoming as big as it is, there's something, you know, we may still not know how the human brain works, but we may know how to model knowledge and regurgitate it and see how it referentially links to itself. Wow.

So, Spock.

Leo: Yes.

Steve: I'm sure everyone who is listening to the podcast knows that Leonard Nimoy died last week, 83 years old, COPD. And I think he was not misquoted as saying that he wished he hadn't smoked as much as he did. I think he gave it up, as I recall, but he spent a lot of time smoking. And so he was in the hospital at the time and had been battling COPD for some time, so it didn't come as a surprise to people that were close to him. And I heard you guys talking about it on TWiT. There was a question about whether Shatner was going to be able to be there. And first it was thought that he wouldn't be, but then I heard later that he was able to get on a plane and make it to the memorial.

Leo: I guess so, yeah, yeah. He had a charity event that he had to go to, but I think he managed to work it out, yeah.

Steve: Yeah. And everyone knows that I've always been very much of the Star Trek generation.

Leo: You can tell because those of us in the Star Trek generation practiced so that we could get our fingers to do that. I can only do it on one hand, but...

Steve: Yeah. And actually...

Leo: It doesn't come naturally, I don't think.

Steve: The best man at my wedding was unable to do that.

Leo: Oh, dear.

Steve: And I begged Gary, I said, because Gary had really a dark sense of humor, and I said, okay, now, you know, you're going to be asked to say something. Please don't embarrass me. Please. You know, I mean, because lord knows what he could say. And so he honored my wish. And he got up with the microphone, and he said, "Steve asked me not to embarrass him, so I'm not going to." And he held up his hand, and he had two rubber bands...

Leo: Oh, how funny.

Steve: ...around each pair of fingers because he wasn't able to do the "live long and prosper" sign...

Leo: Oh, that's funny.

Steve: ...without a little bit of help. But I did hear during the news coverage of this, some people got the story wrong about why "Star Trek" only had three seasons. They were complaining that it was - I think it was on at the wrong time. The person that I saw talking about it, I think it was George Takei, actually, because he was interviewed on one of the shows, said that it was on at, like, 10:00 o'clock on a Friday night.

Leo: Oh, that's terrible.

Steve: And, yeah, because, I mean, no one's watching TV then.

Leo: Yeah, right.

Steve: But I had an occasion, I was at Comdex one year, and that's the year that I bumped into Stew Alsop, who was at the time the editor-in-chief of InfoWorld. And he and I set up the deal for me to do the column, which initially was called "Behind the Screens," but there was a collision with CompuServe. They had a trademark on that. So then he said, "Steve, we can't call it 'Behind the Screens,'" so I renamed it "Tech Talk," which of course was the column that I did for eight years every week in InfoWorld. We are at a bar hanging out and having a couple of beers, and he said, "Hey, do you have any dinner plans?" And I said no. He said, "Well, I'm going to meet some people, and we're going to go to dinner. Why don't you tag along?" I said I'd be glad to.

So we go upstairs in the Las Vegas Hilton and knock on this random door, and someone opens it. And there's some geeky-looking kind of heavyset guys just sort of lounging around, apparently waiting for Stew. So we walk in, and they introduce themselves, and I don't remember anybody else's name in the room because one of the people there said "I'm Gene Roddenberry."

Leo: Wow.

Steve: And at that point it was like, oh, my god. Now, I'm not going to...

Leo: All conversation ended.

Steve: I'm not going to start drooling and be the crazy born-again fan boy, everything. But I did arrange to sit next to Gene at dinner. And so I just said to him, you know, I'm a fan. So I love sci-fi. What you did with "Star Trek" was, I said - in fact, I did tell him, I think, that my buddies and I made a "Star Trek" movie, and I talked about that in the podcast, where we scratched the emulsion in order to make phaser beams and had arranged to have people stand under a chandelier and beam out and all that. I mean, seriously geeky stuff. Oh, and I talked about how we made the audio track because Scott Wilson's sister had stuffed bunny rabbits, and so the aliens were the Bunnons. And we...

Leo: The Bunnons.

Steve: And we used a still-frame camera in order to march them down the hallway and attack us and so forth.

Leo: We are attacking you. Yeah.

Steve: Yeah. And that was where "yo-sha ba-di-dro, sna-na ba-na-ni, pa-shor-yor-nar-ros" came from.

Leo: Aha.

Steve: Because if you reverse that, it comes out "We are the Bunnons. Surrender your ship or be destroyed." So, oh, yes. Anyway, and so I said to Gene, "What happened? We only got three years." And he said, "Oh, there's an interesting story there that," he says, "not many people know." And he said this was before the time that Nielsen was doing demographics. Nielsen was in the ratings business, and they were just counting heads. They weren't counting whose heads. But they were collecting the data, they just weren't using it. So based on the numbers compared to other shows that "Star Trek" was up against during the original Kirk and Nimoy and Shatner and Spock years, "Star Trek" really didn't do that well, compared to - just in terms of raw numbers.

Years later, Nielsen introduced demographics where they realized it was important to care who was watching. Because five year olds weren't buying anything. Neither were 70 year olds. But yuppies in their 20s and 30s and 40s were buying cars and baby formula and homes, and it turned out that in the process of developing and tuning the Nielsen demographics, they reprocessed all of the data they had collected in the past, just because they had a database of it, in order to design their model. Never in the history of television had there been a show that more exactly perfectly targeted the demographic that advertisers wanted than "Star Trek." And no one knew it then. And had they known it, it'd probably still be going today because it was exactly the audience that advertisers wanted. And I just said, wow, that's a cool story. And that's straight from Gene himself.

Leo: Yeah. That makes sense. I mean, it's obvious in hindsight that the show was immensely influential.

Steve: Oh, look at the legs it's had. It's changed our culture. I mean, we have things like phaser beams and tractor beams and "Beam Me Up, Scotty." All of that came from there. I mean, none of that exists. We're amazingly creative little monkeys.

Okay. So you said you would see it last week. I know that you've seen it since last week. Our listeners haven't had your corroboration of my opinion of "Citizenfour."

Leo: Oh, it was really good. It was surprising because I'd put it off because I thought, I know the story, and it's going to be kind of grim, and it's just going to be slow, and I don't really need to spend two hours watching this. But, you know, Laura Poitras did a really good job of adding drama. And as you said last week, it's really interesting that she had the cameras right from the beginning. In fact, Lisa and I were watching, and Lisa loved it, too. And I said, "Oh, he's got the shirt on. He's about to do the interview" that was the first, you know, I recognized the shirt. And so, and that was, like, half an hour in. So you really got a sense of - and I was staring at Edward Snowden the whole time because you could - he didn't - outwardly it wasn't really obvious that he was nervous. But it was so clear that the guy was just really scared of what was about to happen.

Steve: Yeah.

Leo: And knew that the NSA would not be happy. And I feel like he was waiting for somebody to break down the door at any moment. It was a really - the tension...

Steve: Oh, when you saw him twitching.

Leo: He was very twitchy.

Steve: The alarm went off in the hotel a couple times.

Leo: Yeah, wasn't that a moment?

Steve: And they're, like, oh, my god, you know. And like someone knocked at the door.

Leo: They put the T-shirt over his head.

Steve: Yeah, I mean, yeah.

Leo: Wow.

Steve: Yeah. Anyway, so I wanted to make sure - it appeared briefly on - the Huffington Post published an article, I guess it was yesterday, with a link to the show, which I presumed was legitimate because it was the Huffington Post, you know, not some PasteIt.com or something. It turned out it was not legitimate. I got a tweet back from - because I said, hey, you know, here's a copy of "Citizenfour," anyone who wants it. It is on, you said, oh, yeah, it is on iTunes and HBO Go. So I just wanted to - I wanted your confirmation of my opinion because, as I said, I loved it. And I came out of there feeling something I really - with an appreciation I didn't have before, which was that even though - and I know you are of the same feeling. Even though we are not crazy concerned about privacy, I came away with a deeper respect for the rights of those who are. That is that, you know, privacy is a right.

Leo: Yeah.

Steve: And if that's how we're going to define it, then we ought to honor that definition, rather than say it's a right and then not have it treated like one.

Leo: It was also fun to see Ladar Levison in it, and Jacob Applebaum, although Applebaum I think has posted, not exactly a rebuttal, but, you know, and I didn't get a chance to watch it, so I shouldn't speak. But somebody sent me a link. The subhead was something like "Misrepresentations are justified for the greater good." I got the impression that Applebaum might have thought that he was a little bit misrepresented in there. But in any event, must watch. And if I can find the video, maybe the chatroom can pass it along. Just fascinating. And of course we haven't seen the end of it. I mean, I don't know much more material there is, but every week there's something new.

Steve: Yeah, yeah.

Leo: It's incredible. And here we are, 18 months in.

Steve: Okay. So two more things. I know also that you have also fallen in love with a previous show recommendation of mine. I once explained that I normally like shows involving titanium endoskeletons with plasma-driven robots and such. And so when somebody was recommending "The Good Wife" to me, that was about as far, as a title, off of, like, what kinds of stuff...

Leo: Me, too. Me, too. Yup.

Steve: ...would appeal to me. But it is fabulous. And I know you've discovered it, and you've been running forward through it.

Leo: Yeah, yeah. Lisa and I are really enjoying it, yeah.

Steve: Yeah. And so to that I wanted to add one more show, which is really coming on well. It's in its third season. So for those who like to binge, there's plenty there. And that's, on FX, "The Americans." Are you watching it?

Leo: Oh, yeah. I've seen a couple of seasons of it, yeah, yeah.

Steve: Okay. Because for those of our listeners who haven't been, it tells a really interesting story of some embedded Russian spies in Washington, D.C., back in the Reagan era, so it's set back in time, Cold War. And just, anyway, I'm loving it. So I wanted to put that out there as another recommendation from me of, you know, not technical shows, but just really good television.

Leo: Yes. I agree.

Steve: And today voting opens for the Podcast Awards. I don't know why the page says "Vote today, vote tomorrow, vote often."

Leo: Well, there you go. That explains a lot.

Steve: Maybe they want you to come back for advertising impressions. I don't know. I would love you to come back to vote for Security Now!. So it's PodcastAwards.com. I'm down at the very bottom in the Technology. The first item in the Technology section is Security Now!. I would love to win this year. I just - we haven't won for a number of years. I think it would be fun. So I'd just like to see if our audience has the power to make that happen. I imagine that they do.

Okay. And this is totally crazy, but apparently authentic. Some Cornell University researchers were wondering about epidemic outbreaks. And so they decided to model what would happen if zombies were real. So this is the "Zombie Outbreak" model. There's a beautiful animated, automated tool on GitHub. I've got the link to it in the show notes. And Vox.com has the original link about the "zombie apocalypse science." And this was legitimate mathematical modeling. And if you just want the short version, you want to head to the Northern Rockies. That, it turns out, in the event of the Zombie Apocalypse, is where you want to be.

Leo: Oh, good to know.

Steve: Yeah, good to know. But if you're curious, you might put in "zombies-usa," imagine that, into the universal Google search finder. That will probably take you to the GitHub page that has that as its name, Zombies-USA. And then it's just a beautiful - it's a map of the U.S. with bright spots showing population centers, and you can see what happens if the zombies decide to get restless.

And lastly, this I've been waiting on. I've been sitting on this since February 24th, just waiting for its turn. I got the biggest kick out of it because this was a Francisco Gomez, who wrote to Sue, who forwarded it to me on Tuesday, February 24th. The subject was "Oh, my god, you guys are incredible/SpinRite." And he said - this is not long. So he said, "I could write a looong [with a bunch of o's] email, telling you how great you are.

But after 12 hours other people spent fighting and failing to get the data restored, you guys made me look like a hero in front of my wife. Her IT department said she had lost all of her data. But after using SpinRite and running chkdsk, I was able to extract all of the data, even when the hard disk drive was encrypted. You guys are geniuses. Sent from my iPhone." So thank you. And that's why the email wasn't very long, because he was typing it on his touch screen. Anyway, thank you, Francisco. I appreciate that.

Leo: All right. We've been waiting for this for a long time, Steve. I'm very excited. You mentioned this last week, that you would have these guys on the show. And we even showed the video. Was it Andrea Mitchell who plowed into those cones?

Steve: No, it was Lesley Stahl.

Leo: Lesley Stahl, that's right, yes.

Steve: It was on "60 Minutes" a couple weeks ago. One of their segments was talking about the problem with the current lack of real security in automobiles. And I was just - I was floored by the idea that there was no longer a reliable connection between the brake pedal that the drive controls and the braking of the vehicle which the pedal is supposed to enact, essentially. I mean, the idea, I mean, it's one thing to say, oh, we've got software because we want to have USB and play our iPhones through our cars, and we want OnStar, and we want to be able to approach the car and have the doors magically unlock and all these fancy things. Oh, and I'd like to have the car know whose key is using it so that it automatically adjusts the seats so that it remembers the preferences of the current driver.

All of that is cool stuff. But I don't, in the process, want to sacrifice the fundamental imperatives of the way the car works, that is, that the thing you need is when you press on the brakes, you stop. And of course we know, what was it, about a year ago that Toyota went through the acceleration problems. We never really got a straight answer about that. There was some nonsense about, oh, the carpet was getting tangled up...

Leo: That was their story.

Steve: ...in the accelerator pedal. Okay. And then a couple years ago we covered on this podcast the UCSD researchers who were working with, I think, University of Washington. And that was sort of the first surfacing of our sort of low-concern that something was wrong with the way cars were being developed, that it just, like - and it's not surprising, either, because this is what we see time and time again is a system which is not computerized, then begins to get computers, but the focus is on making it work. And only after the fact, sometimes after a great deal of pain, is making it secure.

So by happy coincidence, I was going through the Security Now! mailbag a couple weeks ago, and I ran across a note from one of the guys at Galois, which is G-A-L-O-I-S dotcom, who were the people who are shown on the "60 Minutes" report who are recipients of funding from DARPA because, just in the same way that the Defense Advanced Research Project Agency was the originator of the Internet, now DARPA has something called HACMS, sort of an awkward acronym, but the best way to pronounce it is HACMS, where they're funding various organizations, whether commercial entities or educational or research or governmental, whatever. They don't really care who they're giving money to, but they've pulled together a project whose goal is to tackle the challenge of truly securing our cars.

And so for this week's podcast, thanks to their interest, I mean, they're Security Now! listeners, at least some of them, we have the guys who have successfully hacked, not only cars, but UAVs, to talk to us about the nature of the vulnerabilities, how acute the vulnerabilities are. And their focus has been on working with DARPA and the other team members on coming up with ultimately and someday a true solution to this problem.

Leo: Well, without further ado, let's say hello to Lee Pike and Pat Hickey from Galois.com.

PAT HICKEY: Hi.

Leo: Hi, Lee. Hi, Pat. Welcome.

PAT: Thank you.

Leo: Good to have you. So what was the brand of car Lesley was driving?

Steve: No, no, no. We can't...

PAT: We've been asked not to.

Steve: So here's the point, is...

Leo: That's all I want to know, so I don't buy it.

Steve: ...all cars have this problem.

Leo: Yeah.

Steve: And they increasingly have this problem. There are somewhere - as many as 50 so-called ECUs, sort of autonomous subassembly computer units, in a high-end car. And even economy cars will have maybe half that, in the low 20s. And so the reason the make and model wasn't shown was, first of all, they don't want to really upset anyone. And it's really not fair to discriminate because all cars today are like this.

Leo: Well, wait a minute. All cars are like this?

Steve: Take it away, guys.

Leo: I mean, you can hack my car?

PAT: Probably not in specific. What we've been part of is security researchers have found vulnerabilities. We reproduced some of those to help the "60 Minutes" folks demonstrate that.

Leo: So each model would have a different exploit.

LEE PIKE: That's right.

PAT: Yeah, or makes might have some components that are common across several model years.

LEE: And many are going to be shared because it's all coming from the same suppliers, the subcomponents.

Leo: Right. What is the typical avenue of exploitation?

LEE: Well, so it depends. I mean, there's two kind of classes. So one class of exploits is if you have physical access to the car. And this makes it much easier because you can basically hook right into the data bus.

Leo: Yeah, we've always said that. If somebody has your computer, you're...

LEE: Right.

Leo: You're pretty much screwed.

LEE: That's right. Well, it's even easier than that because, if they have your computer, and you've got an encrypted hard drive, it's password protected. There's some challenge to get into your computer. There's no such challenge to get into an automobile.

Leo: Interesting. And CAN bus is the bus on most of these?

LEE: That's right.

Leo: We use CAN bus for our audio. Can you hack my audio?

Steve: One of the problems, you could sort of divide attacks into two categories. There's like a targeted attack where someone specifically wants to attack someone else, like

Putin is unhappy with one of his adversaries, for example, and decides to go after them. And then the other would be sort of opportunistic attacks. And we see both classes of those attacks on the Internet. And it turns out that the same sort of models apply here. For example, the diagnostic computers in car dealerships are also connected to the Internet. And so when that computer connects to your car, that's a standardized interface, and it's a means for the diagnostic computer to access the networks in the car. And all of those little ECUs are firmware-reprogrammable, meaning that they can be attacked.

PAT: That's right.

LEE: Yeah, and furthermore, you know, you can, with a computer, you need to actually grab it. If you want to reprogram it, put a new program on it, you need to have access. But for convenience you don't want to have to pull out every small ECU in the vehicle. So you can do this all over the bus, which makes it even more problematic.

Steve: For example, in one example of an attack, a malicious CD was put into the CD player on a car. And believe it or not, cars have buffer overrun problems, just like we do in all of our regular software today.

Leo: Yeah, yeah. I guess the real concern, though, mostly, is what happened to Lesley Stahl where she was hacked remotely. I mean, if somebody's got access to my car, I'm going to presume they can do stuff to it. But the real scare is that I could be driving by somebody. How hard is that to do?

PAT: So pretty much all modern cars, the high-end ones, will have a telematics unit, which is a cell phone modem, a 2G or a 3G modem. And the attack that was demonstrated in the University of Washington/UCSD paper was that you could call that cell phone number. There was an in-band modem that had a buffer overflow. So if you played all the right tones to it, you could reprogram its firmware and [crosstalk]; right?

Leo: An audio hack, I like that.

PAT: The telematics unit is your gateway into the system. It's just as if you have physical access at that point. So all the other attacks then can be executed.

Leo: Wow.

Steve: Yeah, so basically you have a car connected to its own cell phone, and it's got problems in the software that can be overrun.

Leo: It's got to be pretty targeted. Is that right? I mean, it's not going to be something that somebody could just say, oh, I see a car over there. Or is it?

LEE: Well, I mean, once you find the vulnerability, it's not going to be automobile specific. So just like a computer virus or worm, right, once it finds a particular vulnerability in an operating system, any car with that telematics system is going to be vulnerable.

Leo: Right, right.

PAT: And the other strategy that these attackers showed was that the telematics units tend to all be in, like, specific phone number blocks.

LEE: That's right.

PAT: So the search space defined, oh, I know there's a car of this make and model parked in this parking lot. Let's just sit here for a couple of hours until we guess the right phone number. So it's doable. It's not trivial, but it can work.

Steve: So probably the way to think of it is a little bit the way we think now of Windows and Mac and Linux. Those three different operating systems all have vulnerabilities, but the same virus cannot attack all three of them. You need an OS-specific virus.

LEE: That's right, yeah. And in addition, like I was mentioning earlier, you've got, even in different makes and different models, the same suppliers oftentimes delivering that software to multiple different vehicles. So it could be a completely different vehicle and vulnerable to the same attack.

PAT: Right. And this isn't something that typically, like, consumers or even the car manufacturers themselves care that much about. They're just buying an off-the-shelf part.

Leo: Well, they will.

LEE: That's right.

PAT: I hope they will.

Leo: Are there attacks in the wild already? I mean, have you - or is this all theoretic?

PAT: There's none that we know of.

LEE: Yeah, other than key fob entry into vehicles.

Leo: Yeah, right.

LEE: That's been reported in the media, but...

PAT: They tend to focus on the most profitable attacks first.

LEE: Steal the car, in other words.

PAT: Yeah.

LEE: That's right.

PAT: Others in the media have shown you can go on the black market eBays, essentially, and find a kit that does the keyless entry, some kind of buffer overflow or who knows what with the keyless entry.

Leo: And the attacks you're talking about really would be more malicious or even murderous attacks.

PAT: Yeah, but it also could be used for this exact same...

Leo: To steal a car?

PAT: ...like, car theft thing. It's probably just that there are lower hanging fruit at the moment.

Leo: Right.

Steve: I guess, too, that if you have a given piece of hardware that can unlock the doors only of a certain make and model, that's fine. You just walk around the city looking for that particular car, and those are the ones you steal.

Leo: Or you have the hack for the cars you want the most.

Steve: Right.

PAT: Yeah.

LEE: Yeah.

Leo: We know the car thieves are looking for specific models, generally, yeah.

LEE: I think it's important to remember, too, it's not just about people wanting to steal vehicles. But as cars become more connected, you have your personal information on them. People might want to track you. Governments might want to track you. You're vulnerable to these invasions of privacy, vulnerable to having your personal information stolen through the vehicle.

PAT: Right. Another attack that was demonstrated by the same researchers is, using the microphone that's for the phone calls or other functions to listen to whoever's in the car without them ever knowing it. I think that one was demonstrated on "60 Minutes," as well.

Leo: Well, mostly they're just going to hear me singing, and swearing a little bit as people go by.

Steve: And I guess then we're, unfortunately, for the last year and a half, we've been talking about the consequences of well-financed nation states' interest. And this does unfortunately sound like the sort of thing that the high-end law enforcement organizations might task a division to go develop these capabilities for as broad a spectrum, as broad a class of vehicles as possible so that they have them available to them when they need them.

Leo: We're talking to Lee Pike and Pat Hickey. They're with Galois.com. And they were, well, their technologies were featured on a "60 Minutes" piece with Lesley Stahl showing how easy it was to hack an unknown American - is it mostly American cars? European cars have the same problem? Japanese cars?

LEE: They all have them.

PAT: Yeah, all of them have them.

Leo: It's all the same. There's no advantage.

PAT: Yeah. I mean, we've heard either anecdotally or in published work from every single make and model that you can think of.

LEE: If you get an F-150 from 1970, it might not be vulnerable.

Leo: The dumber the car, the better.

LEE: That's right.

PAT: Yeah.

Leo: Now, aren't auto - but I presume auto manufacturers are aware of this. But, as I mentioned, their timeline for updating models is a lot slower than, you know, computers.

LEE: That's right.

PAT: Right, but [crosstalk].

Leo: What are they doing to mitigate these problems?

PAT: And the other thing we found is that car manufacturers are very cost-sensitive. So if you have to add a TPM-like device, a Trusted Platform Module, which is used to secure some computer boot chains, if you had to add one of those to every ECU, it would increase the final sales value of the car by, like, thousands of dollars.

Leo: Right, right.

PAT: So the heavyweight solutions we're used to won't apply in this field, unfortunately.

Leo: I remember talking to the folks at Ford, and they said one solution would be to have two computers, have the telematics computers separate from the computer that runs the car. Is anybody doing that? That's certainly a costly solution.

PAT: I mean, that's actually the case now. There are separate computers for all these separate tasks. The problem is that they're all on a completely unprotected network. That network has no authentication or access control. So the telematics computer, once it's been reprogrammed through a buffer overflow somewhere, can start acting like it's the brake pedal sensor.

LEE: And, you know, there's also a - sometimes safety and security are in tension with each other. So you might think, oh, let me completely separate the cabin from any of the engine control. But if you're in, say, an accident, and you've got sensors that detect a crash in your engine or near the front of the car, you might want to tell the cabin, you know, unlock all the doors so you can get out. So it's not always obvious that you can make these security constraints.

PAT: Right. It can be difficult to completely isolate components. And right now there's not very many good mechanisms for doing fine-grained compartmentalization.

LEE: That's right.

PAT: Which is generally part of every security story is the principle of...

LEE: Least privilege.

PAT: ...least privilege.

Leo: Right. Well, so sharing a bus is inevitable because they have to talk to each other.

PAT: Right.

Leo: But you need to have some procedures in place to.

LEE: And one thing that's problematic is, you know, in particular you've mentioned the CAN bus already, and it's a broadcast bus. You know, there's no time division. And it's got a very small payload size. So it's really incompatible, it wasn't designed for having any kind of security. So it's very difficult with - and that's not going to go away any time soon. There's too much infrastructure around it. It's a cheap bus. People know how it works.

PAT: Right. I mean, there are upcoming buses that'll allow, like, upgrades to CAN that allow higher bitrates so you can actually start adding signatures on every packet. Whereas right now there's only an 8-byte payload. So...

Leo: Oh, wow.

PAT: ...even if you only - if you gave up just one byte for a signature, well, then it'd be easy to guess; right? And you'd have given up a quarter or an eighth of your space.

Leo: I didn't realize it was so unsophisticated. This ain't Ethernet we're talking here.

LEE: No.

PAT: This is the finest field bus 1980 could design.

Steve: So, okay. You guys' focus has been on fixing this. So talk to us a little bit about the notion of provably secure systems. And we ought to also mention that you have also been spending time on the whole UAV issue. And we were talking before the show that of course, oh, yeah, no one's ever hacked a UAV. They're completely secure.

LEE: Yeah. So our specialization at Galois in general is a branch of software engineering called "formal methods." And the idea is, rather than giving you incomplete evidence about the correctness of a system that you get through testing, the realization is that software fundamentally is a mathematical system. And we can prove properties about this. So rather than just trying to test and, you know, always missing bugs, let me give you a piece of software where I've proved the correctness once and for all that, no matter what inputs it gets, it computes the right values. And so using these techniques we've developed, and we've been applying these to the embedded system world in particular for this program, so that the kinds of things that we've talked about here just aren't possible, so things like buffer overflows.

Leo: That's interesting that you could prove correctness.

LEE: That's right. And this isn't a new idea. So this has been around for, well, nearly since the beginning of computer science. But so one of the things that makes it possible today is that, because of the power of computers, before you had to do these proofs by hand. So literally you're a mathematician. You're sitting down with a pencil and a piece of paper. And this works for a 10-line program, but it doesn't scale to a hundred-line or a thousand-line or a million-line program, especially if it's concurrent. But with more powerful processing, with better algorithms, we can actually have the computers do the proofs for us.

Leo: And these are not brute-force kind of testing every possibility.

LEE: No.

Leo: This actually mathematical proof.

LEE: That's right.

PAT: Right.

Leo: That's impressive, wow.

Steve: Algorithmic theorem proving sort of process.

Leo: Yeah, that's amazing, yeah.

LEE: That's right. And this has been taken up in the hardware industry. So, for example, Intel, after the FDIV bug, started doing this in designing hardware. And there the payoff is obvious; right? It's very hard to patch, well, you can't really patch hardware. So they have to get it correct. Part of the problem is that in the software industry we've had this kind of sense that, oh, you know, software, we can always fix it after the fact, or how hard is software to write? And so we just haven't been applying these same approaches.

Steve: And I think, again, where the industry is, what we've seen, as I mentioned, we're seeing it now with the Internet of Things. People are finally beginning to - people jumped on the concept of having their refrigerator on the Internet so they could talk to it at work for some reason. That was all good. But immediately those Internet of Things things started getting hacked. And really, as soon as you have telematics systems, where cars have cell phones, and they're able to use technology to call 911 if the car has some problem, or the OnStar system, or you need to ask to get your car unlocked because you locked your keys in somehow, all of that stuff, now our vehicles, our cars are Internet of Things vehicles that unfortunately move at very high speed and carry our bodies around in them.

Leo: Yeah. Kind of want to make sure that's reliable. It's not like a rocket to the moon; but, hey, you know.

LEE: Right. So, you know, in the academic world, people call these things cyber-physical systems (CPS). So the idea of we're having software-intensive computers meeting physical systems where they can kill you, they can destroy property. So this is, yeah, this is [crosstalk].

PAT: Nonmaterial harms like privacy invasion, as well.

Leo: We're talking to Lee Pike, who is the research lead at Galois for cyber-physical systems, and Pat Hickey, his colleague. They are the ones who demonstrated on "60 Minutes" hacking a car while Lesley Stahl was driving it, a very brave thing of her, as it turns out. Somebody in the chatroom asked, are more modern vehicles like, let's say, Tesla, less prone to this kind of thing? Or are they even more so because they're so reliant on electronics?

PAT: Yeah, the increased reliance on electronics is just - right now there's not the kind of culture where they focus on the safety first. They focus on the features first because that's what people buy; right? They can understand it from...

Leo: Oh, yeah. My Audi, my brand new Audi has lots of gizmos and gadgets in there.

PAT: Right.

Leo: That's not necessarily more secure. That's probably less so.

PAT: Oh, right. In fact, most of those are like new attack surfaces; right?

Leo: Great.

PAT: Every cell phone it supports connecting to is a fresh stack to look for a buffer overflow.

Leo: Right, right. I know you don't want to speak specifically about any brand, so we'll table the conversation about Tesla.

Steve: Well, and one question I have is have you seen any evidence that there is any specific manufacturer who is extra good about this? Or are they all just so hampered by the CAN bus and the legacy load that they're dragging forward that they're just more concerned about the shape of the exterior metal than they are the composition of the internal code?

LEE: So I don't know if I want to speak about any particular manufacturer. But I think there has been - the work has been a wakeup call to the industry in general. And there's committees looking at improving the security of automobiles, people taking this more seriously, even on the safety side, for example, with the Toyota case. So I'm somewhat optimistic.

PAT: Right. The real problem is that right now car manufacturers might leave out seatbelts if they weren't required to, you could cynically say, if they weren't incented to by regulations. And there are no incentives from, like, any kind of regulators about cybersecurity yet. And that's something that I can personally hope will happen in the near future because I think we do need it.

Leo: And we've always seen tradeoffs between security and convenience, as you talk about that all the time, Steve. And then somebody in the chatroom is pointing out buyers would probably rather see a car start up faster than have a slow check for authentication.

PAT: Right.

Leo: That would be seen as a detriment, even though it might keep them more safe.

Steve: There's a presentation by the gal who's been running this DARPA project, the HACMS project. And she referred to - so she is in the middle of this. She knows all about what they're doing. She's participated in the R&D. And she had an interesting observation to make which was - which essentially we saw on Lesley Stahl's face on "60 Minutes." And I imagine that this kind of publicity does more to incent manufacturers to try to be able to say "Our cars don't have this problem" than anything else. But Kathleen, who is with this DARPA project, explained how unnerving it was when she was driving a car and knew what was going to happen, she knew - I mean, she was in an area where it was safe, she wasn't actually going to run over anyone or damage the car - that when the car was hacked, and she hit the brakes, and nothing happened, I mean, it was viscerally jarring because from the first moment we get behind the wheel when we're 16, every single time we press the brake pedal, the car slows down.

And you can imagine the panic. In fact, we've seen it on Hollywood movies for years. Back then it's someone drains your brake lines overnight, and you live at the top of a steep winding road hill, and now you're in trouble because you're pumping on the brakes and nothing is happening. But that's the sort of thing which hacking vehicles today means. I'm just stunned that - and this is what surprised me was that the electronics is in even the braking loop, that there isn't still like a fallback. And I understand, you know, I guess regenerative braking is certainly one reason why you're actually not braking in a traditional fashion. In a hybrid vehicle, you're wanting to capture that energy and put it back in the battery. But on a car with disk brakes, I would like to know that there's an actual...

Leo: An override.

Steve: ...mechanical, yeah, mechanical linkage.

Leo: I presume handbrakes are, but they're disappearing. Modern cars have electronic brakes. I don't have a handbrake in my car.

PAT: Yeah.

Steve: And Kathleen mentioned on this video that the ECU, there's an ECU in charge of your seatbelt tensioning. Can you imagine if your car suddenly grabbed you?

Leo: My car grabs me all the time. Oh, but, you mean for other reasons, yeah. Unexpectedly, yeah. So this is a bad trend, really, the move towards fly-by-wire, or as somebody in the chatroom said, die-by-wire vehicles.

LEE: Well, it's, I mean, so I think the other side, to play devil's advocate, is that you can have more advanced, say, braking, or more advanced control. You can have a lighter, cheaper system that you build because you don't have to put in hydraulics; you don't have to put in physical connections. And so to the end consumer, I've got a more advanced vehicle with more features that's cheaper and more fuel efficient. That's as good as it gets, except for the security aspect.

PAT: I think, you know, there's a lot of smart automotive engineers that really know how to make good design tradeoffs when they're building cars. But the design tradeoff they don't have in their mind or maybe aren't weighing high enough right now is security. And that's something that I think is fixable. And there's just, you know, many steps are going to be required to fix it.

LEE: And we need to decide - go ahead.

Steve: No, no, I'm sorry. Go ahead, Lee.

LEE: We need to decide as a society, too, right, what the limits that we're willing to - what kind of costs we're willing to incur for extra security; right? So we could have a perfectly secure, basically everyone driving an Abrams tank. But no one wants to pay for that. You know, it's not fuel efficient. And so there's a tradeoff. And in commercial aviation there's legislation; right? It's not up to the individual manufacturer to decide what that kind of reliability is. And we don't quite have that yet in the commercial automotive world.

Steve: So, yeah, I was going to say that I guess that I was wondering whether the best we could expect is pretty much to follow the same course we've seen with consumer computers, where finally, after a long time of email scripting being enabled by default, that finally got turned off because it was causing so much havoc. And so incrementally, but really arguably very slowly, we're moving forward. And even today we're still getting, you know, CryptoWall is encrypting people's files, and we've got spyware now in the ads that we've been talking about. We were talking about Komodia just last week. And so on one hand, it would be sort of pessimistic, but maybe realistic, to think that we're going to have to go through the same process. On the other hand, it is, as you were just saying, Lee, the case that the automotive industry is subject to a great deal more functional regulation than somehow the PC software business has ever been put under.

LEE: Correct.

Steve: Maybe just because of history.

LEE: So, but, you know, legislation tends to just ignore - so, you know, we talk about boxes, right, like you deliver some sort of unit that may or may not have software in it. But the software is just treated like a black box. And you just assume that the software works correctly. And I think that kind of mindset we need to move away from. And so, you know, a lot of times the automotive manufacturers who are integrating the systems, they can't even see the software. It's proprietary. The suppliers provide it. And so there's no, you know, we just assume that it works correctly, but of course it's not the case.

DAN: I think the angle on how our work fits into that is, you know, right now I'm sure every one of those suppliers has a QA manager. And that QA manager goes to the sales team and says, here, here's what you can tell the manufacturers, that it must be safe because our team used a bug tracker and used source control and used some automated...

Leo: There you go.

LEE: ...you know, Coverity, maybe, it's like a code quality metric.

Leo: Right.

LEE: Just for instance. And that's wonderful. But you have to trust that the team did their jobs, and without very much visibility into that team, and without, you know, it's very hard for a third party to verify that a bunch of humans did an activity correctly. The idea with formal methods is you don't have to trust the process that created the artifact, but you could do a test on the artifact that would prove the properties about it.

Leo: Is this form of verification commonplace in the auto industry? I mean, are they aware of this, even?

LEE: There's some awareness. So as far as I know, the main application that, you know, across the software world in general is static analysis, so the idea that there's vendors who have tools that will run over your source code and try to find vulnerabilities. There's two problems with this in general, though. So they're great, and everyone should be doing this. The tools have really come along. But they give a large number of both false positives and false negatives. So, you know, you might see a whole bunch of reports, but the engineers, you start - your eyes glaze over because there's too many false positives.

Steve: Positives, yeah.

LEE: The other problem is that one might argue that this is at too low of a level for some of the properties that we care about. So, you know, all you can really - it's not going to tell you, does your software implement the function that you implement, because that's in your mind, or it might be in comments, or maybe there's a couple of assertions. So this is really just saying have you written your C or C++ correctly. And so some of the verification work, and the most important verification work, is at the architectural level, so looking at things like data flow, looking at things like the networking. And so we're doing work and research in that area, as well. So before you even write a line of code, is your design correct? There's a nice paper from Amazon, actually, recently about using - they've recently taken up formal verification in designing some of their highly distributed algorithms for the data centers.

PAT: Yeah, a lot of those are for databases where the algorithms are really just too complex for a human to do in their head.

LEE: That's right. And it's not even about the source code at that point. So they talk about the benefits of using formal methods tools to find out is this algorithm going to work? Are there bugs in it before they even write a line of code? Because once you've written the code, it's kind of game over. It's so hard to go back, try to fix bugs, if there's a fundamental design flaw upfront.

Leo: I wonder if heads of state in their big armored limousines, they must, I mean, there must be, somewhere, somebody who's really paying attention to this. They would be targets, I would imagine.

PAT: I don't know anything about that in specific.

Leo: I'm just thinking that it seems like a real great opportunity for bad guys. And I think you make an interesting point, that the methodologies that are used currently in software development for PCs just - it's just not going to be sufficient...

LEE: That's right.

Leo: ...for this kind of platform.

PAT: Right.

Steve: I saw in one of...

PAT: I guess the stance on security needs to take it from the beginning to produce a secure system. Security isn't something you can bolt on afterwards.

Leo: We've talked about that a lot, haven't we, Steve.

Steve: So, yeah. So, okay. In one of the papers I actually saw an acronym that chilled my blood. And I needed - I'm glad I have you guys on the line because there isn't actually an SQL, a SQL Server, in any cars today, is there?

LEE: Yeah.

PAT: You know, who knows? If you have an MP3 player in your car, that might be a Linux application that's backed by SQLite. I would implement it that way, probably.

Leo: And it's certainly in your phone, so...

PAT: Yeah.

LEE: Yeah.

PAT: So why not?

Steve: Yeah, yeah.

PAT: I mean, these are great tools.

Steve: There was a mention of SQL injection attacks. And I thought, oh, my lord, we really aren't learning anything.

Leo: You've got to cover everything. So I'm curious, as we start talking about autonomous vehicles, I mean, I don't think an autonomous vehicle is necessarily more vulnerable. If you could take over the system, you could take over the system. Doesn't matter if a human or a computer is driving. But I wonder if the move toward autonomous vehicles will support you guys in the sense that people are going to be more aware of the potential risks, and they're going to pay more attention to correct this and so forth.

PAT: Yeah. So certainly a vehicle being autonomous helps people think of that problem a lot more clearly, like, okay, I'm going to be giving it commands from my desk. But how do I know somebody else can't do that? It helps break away that abstraction of the human being in control by proximity. But as far as, like, certification goes, or the culture of security goes, I think that's just totally on a case-by-case basis, whether the people building that particular system cared about this enough, or whether they didn't.

Leo: Right.

LEE: But I would actually argue that, you know, autonomy, it introduces yet another vector of attack. So there's just a whole bunch of software there. And then, furthermore, now we're talking about a whole bunch of additional sensors that your vehicle, aircraft or automobile, is now dependent on. So, you know, if your GPS - your car might use GPS coordinates to help you navigate. But if your car is actually using GPS, the autopilot, to navigate, well, then, you know, we have to think about what happens when there's a denial of service, or someone's attacking some of the sensors.

Leo: And it's interesting, you also mention it's not merely to take over the car to steal it or to disable the brakes to kill the occupant. It could be a privacy - used to invade privacy, as well.

LEE: That's right. And there's even new standards that are being proposed by the

Department of Transportation, so vehicle-to-vehicle communication is what it's called, the idea that this allows, regardless of the make or manufacture, different vehicles to communicate with each other so that you can do collision avoidance is one of the main use cases.

Leo: Mmm, right.

LEE: But this is, you know, so there's a nice safety argument for this. On the other hand, you can use the same infrastructure, perhaps, if it's not implemented correctly, to track vehicles. So if every vehicle is broadcasting where it is, if everything's using a public key infrastructure, and it's not done correctly, you might be able to determine, you know, where's someone going, what are they doing, without having to, say, physically bug the vehicle.

PAT: Right. We've seen over the years lots of attacks that, without attacking the PKI specifically, there are various other ways that you could differentiate certain cars from each other, or, sorry, certain computers from each other and make very educated guesses about that being the same computer, as you saw before.

Leo: Well, this is exciting, isn't it, Steve. Isn't that special. Really, what I find interesting is Lee and Pat's background and kind of almost a higher level of academic research and proving software correct and so forth. And it sounds like we need to use a variety of disciplines going forward to protect ourselves. And, you know, it's frankly not just cars anymore. Everything. It could be elevators next.

LEE: Yeah.

Steve: Well, and they have a great paper, I linked to it in my Twitter feed, titled "Securing the Automobile: A Comprehensive Approach," which is, like, maybe, oh, it's 10 pages long, although the last page is all references. It looks like maybe the last two pages are all references. And it is exactly that. It's academic and scholarly, and it talks about the development of formal software proofs for the correctness of this. And I guess I like it because I'm glad that DARPA is once again spending taxpayer money to encourage individuals with the proper experience and talent sets and backgrounds to consider how we fix this in the future. Maybe not today. Maybe not the next model year's car. But at some point it'll probably be some catastrophe that occurs, as is so often the case, that finally is a wakeup call and makes Congress move and puts a mandate in. And we'll be annoyed that it'll have a 10-year implementation horizon, but the automakers will have argued that that's how long it takes to move the whole supply chain up. And a decade later, we may have the fruits of the labor that these guys are investing in right now.

Leo: Lee, are you any relation to Rob Pike?

LEE: Not that I know.

Leo: Okay. Just chatroom wanted to know. And, you know, there's an article in the Huffington Post. Richard Clark is a national security expert, advisor to many Presidents, speculating that a journalist who was killed in an early morning auto accident last week might have been killed, he said it was consistent with a car cyber attack. So, I mean, I think that may be a little bit sensationalistic. But I wonder how long before this kind of stuff starts hitting the...

Steve: So we're calling them cyber-physical, cyber-physical attacks.

Leo: Cyber-physical, yeah, yeah.

Steve: Wow, okay.

Leo: That's what these guys are experts in.

Steve: We will add that to our lexicon.

Leo: Hey, it's great to talk to you.

Steve: Thanks very much, guys.

Leo: Lee Pike, Pat Hickey from Galois, G-A-L-O-I-S dotcom. Really appreciate your taking the time to join us today.

PAT: Thanks for having us.

LEE: No problem, thank you.

Steve: Thanks, guys.

Leo: What an interesting subject. I just - I find that fascinating. And I really - we've kind of talked about this before, the idea of proof of correctness in software.

Steve: Yeah.

Leo: Which really does seem like the realm of the academic. But if it's possible.

Steve: Well, yeah. And I think Lee put it exactly right, and that is, it's a little bit like the puzzle I solved, the longest repeating strings problem, because some people tweeted back and said, oh, that's been solved, it's on Wikipedia. And it's like, yes. It's been solved for toy problem sets, like for very small corpuses. It's trivial to do. You build a substring tree, essentially, and then you just ask that tree where the longest repeating string is, and the tree expresses it.

The problem is, for a huge corpus, you can't build the tree because the size of the tree explodes. Similarly, you can do trivial software proving systems that can prove that where you express what an algorithm is going to do, the software is able to take that machine definition expression and analyze the algorithm and verify it. And exactly as Lee said, the problem is it doesn't scale. It just, I mean, or I should say until recently we haven't had the power of the computers and the RAM necessary and the formal attention.

There's, again in this DARPA presentation, it shows an exponential improvement in the strength of software-based correctness proving. And essentially in the last 10 years there's been a series of breakthroughs that have made it far more practical to begin to apply this kind of technique where you can finally say definitively, you have a definitive counterargument to the complaint that all software has bugs. It's like, well, we can actually mathematically prove that some doesn't.

Leo: Fascinating. Steve, what a great show. Thank you.

Steve: Absolutely.

Leo: Lots of fun.

Steve: I was so tickled to get the email from these guys saying, "Hey, we watch Security Now!, and we're the guys that were on '60 Minutes.'"

Leo: Cool.

Steve: So I'm glad we could have them, you know, put a face behind the story.

Leo: Galois.com. Steve Gibson is at GRC.com. That's where you'll find SpinRite, the world's best hard drive maintenance and recovery utility, all his great freebies. More work on SQRL coming down the pike. It's getting exciting now.

Steve: Yeah, we've got iOS clients and Mac clients and a whole bunch of different web server platforms. So we're getting there.

Leo: You can also leave questions for him next week if, the good lord willing and the creeks don't rise, we'll have a Q&A episode. Go to GRC.com/feedback to leave your question, or tweet Steve at @SGgrc, and we'll try to get your questions on. Steve has 16Kb audio versions of this show on his site, along with great transcriptions from Elaine Farris. We have full-quality MP3s and video and all of that stuff at TWiT.tv/SN. You can also subscribe to any of those formats on iTunes or whatever you use on your smartphone or your mobile device. But do subscribe because you don't want to miss an episode. And you don't want to be one of those people who says, "Steve, I've started listening at Episode 1." There's almost 500. It's going to take you a while. Subscribe now.

Steve: Yeah, there are, like when we did that set on how the Internet works and how CPUs...

Leo: Good stuff.

Steve: Remember the whole processor architecture stuff? There are some gems back there.

Leo: Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.