What Is RADIUS?

Description: After catching up on another whirlwind week of really interesting Internet security news, Steve and Leo provide a brief overview of "RADIUS" - the 22-year-old pervasive, but often unseen, protocol and system for providing wide area network user authentication and accounting.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-431.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-431-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson with a lot of security news. We'll talk about that Coin wallet idea. And he'll also explain what RADIUS is. A lot of people asking, how come proXPN only allows 12 characters on a password? And is that insecure? No. Turns out it's very common practice. Steve explains why it's not a problem, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 431, recorded November 20th, 2013: What Is RADIUS?

It's time for Security Now!, the show that protects you and your loved ones, your privacy, your security online. The Explainer in Chief, the King of Security Now!, Steve Gibson is here. He's actually in-studio with me. We just give him a little box to look through, a little window so it feels like he's onscreen. No, he joins us from his Fortress of Securitude in beautiful Irvine, California each week. Hello, Steve.

Steve Gibson: We've been having day-long scheduled power outages down here.

Leo: What? How dare they?

Steve: It's, well, you know me. I'm pretty much SOL. When there's no power, it's like, okay, well, thank goodness Kindles last a long time. So I've been doing some reading and then working on catch-up stuff. So we've got one more next Monday, and that ought to finish it.

Leo: What are they, power washing the nuclear plant or something? I mean, why

would you have day-long outages?

Steve: No, well, Irvine is all underground power, as all modern metropolises these days, metropoli. And I've been here since '84, and there's never been anything like this. And we just have old equipment. And so I think they're having to do huge upgrades. And they're also pulling, I'm seeing big truck-mounted reels of, you know, everything is 440 volts underneath the ground. And then they have local subterranean transformers that drop it to 220 and to 110. And so these massive cables are being pulled. So I think they're probably increasing capacity, maybe just rotating out old cable for new. I've been curious enough, and certainly I've had some time on my hands when all the lights are out to wander down and say, okay, so what is this? But it doesn't look like there's anyone really to talk to, so...

Leo: They don't know. I don't know. Yeah, just putting that line in there.

Steve: Yeah, just they said draw this over to here and then connect it to that, yeah. There were a couple guys out in front the other day, and they looked like that. They were listening to some radio with their feet up on the dashboard. I thought, okay, I'm not going to get the whole story from these guys, so. But the power came back on, and everything's fine.

Leo: What do they say? What is black and orange and sleeps eight? A CalTrans bus.

Steve: Yeah.

Leo: It's a bad joke.

Steve: So a number of people have complained, over the months that proXPN has been a sponsor, that proXPN's password length, their maximum password length is 12 characters.

Leo: Hmm. That's usually a bad sign, isn't it.

Steve: Well, but it's not in this case. And that's why I thought something we've never talked about is RADIUS. RADIUS is a 22-year-old technology that was first - there was an RFI that was put out, like almost pre-Internet, for developing distributed or wide area network authentication. And so the rules that apply to a RADIUS-based authentication are very different from everything we've been teaching everyone for the last couple years about a website should never be able to show you your password in the clear; and if it's a fixed length, that means they're not hashing it, you know, blah blah blah blah blah. And but proXPN allows you, when you stop to think about it for a second, you can connect to servers all over the world, wherever you choose to connect to, and somehow that server knows you have an account with proXPN. And our end-users, our listeners, will have encountered this. We've never talked about WPA2 Enterprise. What is that for WiFi? And that's WiFi where you don't use a preshared key, but where the router looks

up your credentials on the fly. And that uses RADIUS.

And so anyway, our topic today, if we get to it, because there's also - there's so much news and really interesting things that I want to talk about. Well, that's nominally the topic. But if we do it next week, well, so be it. But I wanted to explain to people why this wasn't a problem that proXPN - I mean, I'm not defending proXPN. They can take care of themselves. But there's a misconception that the only way to do passwords is the way we've been talking about. And actually, if you have very different needs, which is what a global network has, then the rules are a little bit different. So I wanted to cover that.

And the other thing that I thought was funny was, it's like, wait a minute. All a proXPN account is doing is authenticating you to access their servers. They're the ones who want it to be strong. We really don't care. I mean, it's like you could give your - you wouldn't want to give your credential to somebody else. But, I mean, you could. You would never do that with cloud storage, where someone would have access to your stuff. But here it's really reversed. It's proXPN that would want secure authentication. But you don't really care. I mean, who cares if, like, someone gets your credentials and is able to have access to their networks. They're the ones who want the protection, rather than the user. So it's sort of turned around anyway.

But anyway, we have a ton of stuff to get to. We had the 9th Annual Podcast Awards. I didn't find out about it until 24 hours before they were closing. I want to talk about that. GRC is now full Perfect Forward Secrecy operational. There's been news on the CryptoLocker front, of course. Bitcoin has had a wild ride. There was news that concerned a lot of people that came out last week about the OS underneath the smartphone OSes that apparently isn't very secure. Two really interesting new payment systems have surfaced. A bunch of miscellaneous tidbits. And, as I said, if we ever get to it, we'll talk about RADIUS.

Leo: Holy cow.

Steve: So Calomel SSL Validation. It's funny because I installed the shortcut after thinking that I would probably now get the big blue shield that we had not been receiving until now. And on the Welcome page in Firefox, I found us. It said "Special thanks to Steve Gibson and Leo Laporte for mentioning the Calomel SSL Validation Firefox add-on in Security Now! Episode 428 and also Security Now! Episode 430." And before long it'll probably be edited to say "and also in Episode 431," because that's this one. And sure enough, we are now the blue shield, which is as good as you can get.

Leo: Who are these guys?

Steve: So, okay. What happened was, the reason I started talking about them in Episode 428 is that, with the move to Firefox v24, which is already a version ago, they - I think it was for 24. Maybe it was 25. I think we're at 25 now. But an add-on had access to granular detail about the connection security. And so the Calomel, that's C-a-l-o-m-e-l, add-in, it installs a shield icon to the immediate left of the URL bar in Firefox. And it's blue or kind of orange or I'm not sure what colors. I've seen like Google's not looking so good. They were, when I went to my Google Drive to set up the document to send you, Leo, I got only like a kind of a pukey-looking yellow color. And I don't remember exactly why. But if you click on that, you get in-depth details. In fact, if you scroll down on my notes, Leo, I took a screenshot of GRC's current Calomel SSL validation.

Anyway, the point of this is it gives you very granular readout on the connectivity Firefox has with the server. And, for example, it shows that we're using an Extended Validation certificate. I kind of wish it gave me a little more credit for that because that's nonspoofable; whereas regular domain validation, DV certificates, can be spoofed with a man-in-the-middle attack. And that's one of the extra benefits of Extended Validation. But Calomel doesn't give us any credit for that. They just note it, that it is there. And so then it ranks the site. It shows the cipher suite you're actually using between the browser and the server, the type of key exchange. And you can see in GRC now we have ECDHE.

We talked about ephemeral Diffie-Hellman key exchange some time ago on this podcast [SN-328]. And that's the kind of key exchange where you're not using the server certificate both for authentication and to establish key agreement. You do the key agreement separately. You only use a certificate for authentication. And that's what allows - that's what prevents a future compromise of the certificate from being able to decrypt past encrypted conversations. So if you don't have Perfect Forward Secrecy (PFS), as a consequence of having an ephemeral key, that is, a key that's negotiated on the fly between the endpoints, if you don't have that, then captured traffic can later be decrypted if the certificate ever falls into the wrong hands, the [cough] NSA [cough] or anyone else.

Anyway, so we're now blue shield because we have Perfect Forward Secrecy, at the expense of being technically vulnerable to the BEAST attack. The reason I hadn't set us up this way a long time ago was that we were getting dinged for being vulnerable to the BEAST attack. But now that Safari has updated themselves, and in fact they're the last browser to do so, all the browsers now provide client-side protection against the BEAST attack. And that was an attack against the CBC. The CBC-style method of using a cipher was prone to some exploits if the browser didn't take measures. I think Opera was the first, and then Chrome and IE and Safari. Now all the major browsers are providing that themselves, which allows me then to put the CBC ciphers back at the top of the list and move RC4 all the way down to the bottom.

Now, in an interesting little side effect, I broke GRC's email. Neither myself nor Sue nor Greg are any longer able to establish an SSL connection to our own server. But that's because we're using an ancient version of Eudora still. And when I updated all the servers over the holidays last year, that was the first time we were able to establish SSL connections into email. It's not a big deal because we establish point-to-point connections to GRC. But if, like, Sue's out traveling and wants to use open WiFi, it'd be better for our email to be encrypted and not in the clear, as it otherwise would be. So it must be that the cipher suite that Eudora is looking for, I removed inadvertently. Oh, and my iOS devices won't connect, either. None of my iPads or my iPhone will now connect securely to GRC.

So anyway, I haven't figured that out. I just did this last night. And I immediately sent texts to Greg and Sue saying, uh, I can't send you email because I just broke email. So you guys are going to have to step back from using SSL briefly until I figure out in detail what I broke. So I'll have to add back a cipher suite that Eudora wants and then see what's going on with iOS. So I, you know, I'll figure that out. But that's always sort of the - it's a consequence of sometimes tightening up security is you break things, which is why we've talked often about various companies only inching forward slowly to make sure they don't break things.

And in fact, perfect case in point is our next topic, and that's Google, who has completed its intended and early announced migration to full 2048-bit SSL certificates. They were saying by the end of 2013. They decided to increase the priority of doing that and stepped up - they have now - they announced yesterday, I think it was, that they are

now 100% 2048-bit SSL and Perfect Forward Secrecy cipher suites throughout. Which we know matters because we know they're one of the targets of PRISM and other efforts by the NSA to perform data collection. So they are now working furiously to do the same thing, to bring up strong security among all of their inter-datacenter links, which they believed were secure and private, but found out that there have been taps installed. So that they're working on also.

And Marissa Mayer announced that Yahoo! will be following suit. And I did see some reaction to that announcement, essentially sort of yawns from the security community saying, oh, Yahoo!'s going to be making themselves more secure? Yeah, okay.

Leo: Well, that's good. Why is that, I mean...

Steve: I guess it's just that it's like...

Leo: They were being snarky.

Steve: Yeah, they were. It was like Swiss cheese deciding it wants to have fewer holes.

Leo: Oh, really? Yahoo! is not so secure?

Steve: Oh, my. Oh, my god, Leo. Oh.

Leo: Oh, okay. So SSL is the least of it.

Steve: Exactly. It's like, okay, well, that'll be nice. But that only leaves 12 other ways we can get in. So...

Leo: I see. Okay.

Steve: So, yeah. So Yahoo! said they, too, are going to encrypt all the data between their datacenters. And it's like, ah, okay, fine. And they're going to offer users an option to encrypt all of the data flow to and from Yahoo! by the end of the first quarter 2014. Which of course everybody else has been doing now for a year. They're like, oh, well, you know. Yeah, uh-huh, yeah, we, too. So that'll be nice.

CryptoLocker: The top of the news there is just sort of random. I don't know why this picked up so much traction. But a random police department in Massachusetts, in Swansea, S-w-a-n-s-e-a...

Leo: Swansea, yeah.

Steve: Yeah, Swansea, Massachusetts. They got had by CryptoLocker and paid the ransom. Which, based on the bitcoin value at the time, they had to pay two bitcoins for a total of $750. And but then in their press announcement, I'm not really sure what the sequence was or why they even mentioned it. Because, I mean, it really got a lot of traction, the idea that a police department got had by this. And they were saying, oh, but, you know, don't worry, we're completely secure, and there was no exposure of any of our data. To which I did see a comment from Brian saying, uh, yeah, yawn, sure, okay, yeah. But they've paid their ransom and apparently got their data back. And they must not have been backed up because of course having a current backup is a way to thumb your nose at these guys.

Leo: Well, also the police authorities are saying don't pay these guys because it just encourages them.

Steve: Well, yes. And...

Leo: But you see the problem.

Steve: Yes. Easy for them to say. And that's just it. When I see people saying this, it's like, yeah, okay, well, easy for you to say. But it doesn't always work out that way. I mean, I owe my existence, I mean, the reason I'm here able to do the podcast with you, Leo, I mean, to take the time to do that, is thanks to SpinRite, which is recovering people's data which they don't have backed up. So it's like...

Leo: Right. If people backed up, you'd be out of business. So...

Steve: Yeah.

Leo: Yeah. So stop it, people.

Steve: So we're beginning to see some sense of how pervasive CryptoLocker is. BitDefender Labs has reverse-engineered, as have many others now, the domain name generation code which is how the CryptoLocker virus or malware - I guess it's really not a virus because it doesn't spread itself so much as it does - you pick it up phishing, malicious websites or ads or whatever, or phishing email. We've talked about this before, how based on the date, a cryptographic algorithm is used to generate a large, I mean, many hundreds of random-looking domain names, and so there's like a spray. And the client will then start making DNS queries of all of these domain names, trying to find the one from among them that the bad guys have actually preregistered and set up a server on.

This is sort of - this is security through obscurity. It's not perfect, but it works. Because the only way to prevent this would be for some authority to go and register, preemptively register all of those hundreds of more domain names, which is a real problem. Also, because it's many different top-level domains. It's not just in dotcom, but it's many other dot prefixes, or suffixes, rather, that we have talked about in the past.

So what this means, though, is that CryptoLocker is trackable. You may not be able to intercept it, but you can track it because security researchers would merely have to register one or two domains in the future and then set up sinkholes to monitor the DNS queries to those domain names. That wouldn't allow them necessarily to block CryptoLocker because they can't respond to it affirmatively. But they can measure it. And that's what's been done.

So here's a metric from BitDefender Labs, who did this, and it is chilling. In less than a week, less than one week of monitoring, they detected 12,016 individual IP accesses, individual IP queries, to a DNS domain that they were monitoring. Now, you multiply that by two bitcoins, which is the ransom, and it's hard to say what the value was at the time. But say that it was $500 because it's certainly there now. That's $12 million in ransom in less than a week, Leo.

Leo: That's, like, hard to believe. That's why they do it.

Steve: And, well, unfortunately, it's why it will never go away. It's why, as I said when this first happened, this is really bad because this means they can make money. And if they can make money, this is all anyone will ever do again.

Leo: And that's why law enforcement says don't pay them.

Steve: Yes. But unfortunately...

Leo: We've said the same thing. Don't buy - you know, spam has a very low cost of entry. And it must work well enough to make it work because even though we say again and again, don't - what, are you crazy? Don't buy Viagra through an email. But it must work, and it's the same thing. You know? The cost of doing this is so low that...

Steve: Now, what's really interesting is that, since we have IP addresses, we can now determine where the infections are coming from. And Microsoft's TechNet blog, for some reason they wanted to give it their own name. They call it Crilock, C-r-i-l-o-c-k. But they've got a really interesting chart...

Leo: Looks like spray paint.

Steve: Yeah, in their blog, which I have here in the show notes if you want to bring it up, Leo, which demonstrates a phenomenal weighting of infection by country. That big blue pie is the U.S.

Leo: 79% U.S.

Steve: Yes, is leading everyone, is leading the researchers to believe that these are highly targeted attacks. For whatever reason...

Leo: Oh.

Steve: Yes, that's the point.

Leo: Interesting. So it's more like spearphishing.

Steve: Yes, exactly. And, I mean, I've been accumulating what I assume is CryptoLocker viruses. They're being sent in ZIP files. And I have a hex viewer that I can look at safely. And so I'll right-click on the file that is an attachment in email, like any ADP payroll, that's the one I seem to be seeing a lot. And if I right-click on it, I can see the first two bytes are PK, which is old Phil Katz, who designed the format. But then inside, in the clear, is the filename in the PK ZIP format, in the ZIP file, and I can see that it's .pdf.exe. So it's pretending to be a zipped PDF. It's actually executable. And I have these sitting in a folder, and I've been wondering if anyone would want to, like, deliberately infect themselves to have the experience or see it work or whatever.

Leo: See it work. No.

Steve: Yeah. It's a little bit dangerous. But...

Leo: So you've been getting it.

Steve: I'm getting it in one of my - because I also have honeypot email accounts that I've created over time.

Leo: So it's targeted in what sense? I mean, what is it about your honeypots that are attracting attention?

Steve: I mean, they've just been around for a while. They're longstanding...

Leo: So anybody could get this. It's not targeted in the sense that it's you.

Steve: Correct. And obviously police departments are getting it. All kinds of people are getting it. So it is email that is sending you this loader virus, or this loader malware, which then goes and picks up the rest and installs it and does its dastardly deeds. But clearly not just global surfers running across Internet pages. This is why it's 80% of the infections are U.S. They are highly targeted.

And I got a note in Twitter from someone named Abe, who said - he said, "@SGgrc Do you think the recent surge in the price of bitcoins has anything to do with CryptoLocker?" And he was one of many people who were speculating. And I don't think so. I think the reason we saw this bizarre, I mean, it jumped, bitcoin jumped up to $900 at one point, even north of $900.

Leo: What?

Steve: Yes. Earlier this week it was $900.

Leo: You should have sold. That's 45 grand you had there, Steve.

Steve: I know.

Leo: I should have sold. I got seven or eight bitcoins. Holy cow.

Steve: It was Congress. U.S. Congress was holding hearings about the nature of virtual cyber currencies. And the testimony was far more positive than people expected.

Leo: I see.

Steve: It was less kneejerk morons talking and more smart people saying, well, you know, yes, bad guys are using this. But there are some serious benefits to virtual currencies. And bitcoin just took off like a rocket, up to north of $900, and stayed there for a while. And then it's come back down. But still, it's in the 500s, I think, at this point.

Leo: $570, last I checked Mt. Gox. Is Mt. Gox the place you look for value? Because they say $700-some is the peak.

Steve: I found a neat app that I like called ZeroBlock. ZeroBlock is an iOS-only app, but it has a very nice screen that shows you what's going on. You're able to tap up at the top to change exchanges. You can pull down, and it will do a running computation of your number of bitcoins at the current price and show you how much you're worth. You can also slide to the side and look at year/month/day charts of various sorts. It's free. And if you want to - I think it's 99 cents, and then you can get the charts in color. So anyway, it's a very nice little app that I like, that I'm running.

Leo: I'm surprised you can still use it. Apple's been blocking bitcoin apps on the App Store for reasons nobody really...

Steve: Interesting, because I got a few of them, and then chose this one from all of them. So, yeah, $565 at the moment. And it was almost twice that not long ago. But it was a consequence of, I mean, I think people are nervous. I don't have a good feeling long term about the future of any virtual currency. I just think our government is going to be unable to resist stomping on it. Then it becomes a pirate currency, and that's going to hurt it some. And certainly all the attention that CryptoLocker is getting...

Leo: Yeah, well, it'll grow, too. So far, from the governmental point of view, the people who are using bitcoin are criminals.

Steve: Right.

Leo: At least there's two prominent examples.

Steve: Right, right. And there was also some news in this last week that assassinations can be purchased in bitcoin now.

Leo: Yeah. We talked about that on TWiT, the bitcoin assassin. Because didn't Ross Ulbricht or whatever his name is, the guy who is allegedly the mastermind behind Silk Road, apparently he tried to hire an assassin using bitcoin. With no result. I don't know what kind of assassin you'd get using bitcoin, but...

Steve: It must be that there are bitcoin miners who are pretty happy. I mean, I just...

Leo: You are.

Steve: I just got 50 back in the day, and it's like, wow. That's, like, you know. But, I mean, but seriously, there were a lot of people who invested a lot of money in hardware, and they were cranking out bitcoins when they could. And I'm sure there must be wallets that are stuffed with them right now.

Leo: Somewhere.

Steve: So that's very cool, yeah. So, okay. A second OS hiding within every mobile phone. This was another topic of great interest over the past week. Here's what's going on. We all focus on Android security and iOS security and passwords and encryption of the ROM. Do you use a four-digit code or a long password? How do you make sure that your memory in your iPhone is strongly encrypted when you're not using it, blah blah blah blah.

All of that is only one OS in a dual-OS architecture. There actually is a second operating system. And this is the so-called "baseband" operation, where the actual cellular protocol is managed. iOS has none of that. Android has none of that. They all just buy it from Broadcom or Qualcomm or one of these "com" companies. They're typically using an older ARM processor. Generally it's, like, v5. And, for example, in the case of, I think it's Qualcomm, there's 69 threads running on this ARM processor that actually manages the cellular dialogue, the cellular protocol, the connectivity to the cell tower, the handoff, the signal strength measurement, the whole underlying cellular tech.

Now, this is stuff from the '80s. And what's happened is it hasn't changed much. And basically, it's like, it's what's in the dumbest cellular phone you can get. So when you add brains to a dumb phone to turn it into a smartphone, what you're doing is you're just

putting this blob of UI and much more power on top of - you're sort of hooking it to the buttons that were on the dumb phone. But so there's still a dumb phone underneath all of our smartphones. And the point is, it is riddled with security problems. Back in 2010 a security researcher messed with his GSM phone, and he reverse-engineered the code in the so-called baseband processor and found all kinds of problems.

And in this last week, the reason this sort of came to everyone's attention is OSNews.com did a story about the so-called second operating system hiding in every phone. And the guy who wrote, Thom Holwerda, he wrote: "The insecurity of baseband software is not by error; it's by design. The standards that govern how these baseband processors and radios work were designed in the '80s, ending up with a complicated codebase written in the '90s - complete with a '90s attitude towards security." Enough said. I mean, we know what a '90s attitude toward security is. It's like, yeah, my password is "monkey" and so forth.

And he says: "For instance, there is barely any exploit mitigation, so exploits are free to run amok. What makes it even worse is that every baseband processor inherently trusts whatever data it receives from a base station - in other words, a cell tower. Nothing is checked. Everything is automatically trusted. And lastly, the baseband processor is usually the master processor, whereas the application processor, which runs the mobile operating system, is the slave. So we have a complete operating system," he writes, "running on an ARM processor, without any exploit mitigation, or only very little, if any, which automatically trusts every instruction, every piece of code and data it receives from the base station you're connected to. What could possibly go wrong?"

Leo: I mean, this is - I think this has been more widely known than you realize. A lot of times when you jailbreak a phone or modify a phone or hack a phone - the baseband software is the radio software. So basically you often have to modify the radio software. So, and I think that's one of the reasons you can do it is because hackers take advantage. Hacking and jailbreaking, rooting and jailbreaking, rooting for Android and jailbreaking for iOS, almost always take advantage of security flaws. That's what makes it easy to do. And I think often the security flaws in the baseband are where they attack. So while this is a revelation to some, I think this isn't such news.

Steve: Well, what I think we're going to see, I mean, so people were asking me, what does this mean? And my take, I mean, I understand what you just said, Leo, but we're now seeing enabling technologies we haven't had before. We've talked about software radios, which have really come up in capability and down in price. That will enable individuals to trivially set up malicious fake cell towers. And so it's been one thing for the software running on the phones to be insecure. But once what's running on the phones is understood, we're going to see next-generation exploits of evil cell towers, so-called base stations, set up. And people's phones will connect to them because they will look exactly like a Verizon or an AT&T or a Cingular or a Sprint or whatever, and the phone will connect to it. I mean, these things still had the Hayes instruction, the Hayes command set.

Leo: Yeah, ATDT, yeah, yeah.

Steve: It's like, oh, my god. That's all in there. And I think we're going to - we haven't, I mean, even though the vulnerability's been understood, as we know, ease of exploitation

matters, and cost to exploit matters. And when the software programmable radios mature, and people start writing exploit kits and starting posting them, we're going to start seeing problems that we haven't seen before. So my take is I don't think anyone's going to - maybe the people doing the radio software are now understanding they need to wake up. And I hope they are because, if they don't, this will be the next frontier. Even though it's an old one. And really, things that are being exploited are typically been around for a long time. But it's like, oh, look, we can do this now. And so people are going to start.

Leo: It's definitely in the interests of Qualcomm and these companies to secure this stuff.

Steve: Yeah. So HTTP/2 has also been in the news a little bit. We're now at HTTP/1.1. And HTTP/2, there is an organization, and if you want to just have your eyes cross, Leo, look at tools.ietf.org/wg/ - "wg" is for working group - /httpbis. That's the page where they're managing - yeah, there it is. And, I mean, it just - it's like, oh, my lord. This is what it takes in terms of committees and interacting and working groups and so forth to move something that is as big and as huge and powerful as HTTP to its next phase. And HTTP/2 is where we're wanting to go next. So one of the...

Leo: That's what "bis" means. It means the second.

Steve: Yes. So one of the things that is high on the agenda, more so now than ever, is essentially binding security much more tightly into HTTP than before. Everyone knows that we have HTTP and HTTPS, where the "S" means secure. With HTTP/2, the discussion is should there be nonsecure HTTP? That is, they're seriously looking at making HTTP/2 secure always. That is, if you are HTTP/2, that is, HTTP v2, you are secure, period. Now...

Leo: SSL style.

Steve: Exactly, SSL style, yes.

Leo: And we now have the processor power to do that, and that seems completely sensible.

Steve: Yes. Now, but the problem is how do we get there from here because, for example, lots of sites don't have and don't feel they need privacy, that is to say, encryption. They're just random pages. It's like Wikipedia hasn't had it until just recently. It's like, hey, this is just all public knowledge. This is just - we're just a big database serving pages. Why do we need security? Well, people want not to be eavesdropped on as they poke around Wikipedia because now we know everyone's being profiled. When you were talking about what ISPs are doing, ISPs can see, if you're not using a VPN, everywhere you go because you're technically using, typically, you're using their DNS servers to ask for the IPs of every website that you look up. So they know everything about you.

So one of the possibilities is known as "opportunistic encryption," or also known as "relaxed TLS." And the idea is that - remember that SSL provides two things. It provides authentication of the server, typically - you can do the client, but typically the server - and also privacy, thanks to encryption. But those two things are separable. They're technically not the same. In fact, we were just talking about how what Perfect Forward Secrecy does is it makes the key agreement separate from the authentication so that authentication doesn't govern, isn't used directly for controlling the key and thus allowing traffic capture to be later decrypted if the key was known. So authentication and privacy are separate.

So relaxed TLS says we can establish - and here terminology is very easy to get wrong, but important. We can establish a private encrypted connection without certificates. And that's the case because the certificate is really only asserting the identity of the server. And we could use Perfect Forward Secrecy without a certificate just by setting up, and we've talked about it, a Diffie-Hellman handshake does allow two endpoints to establish a secure - to agree upon a key which is then used for ciphering, where a passive man in the middle, completely observing their communication, is unable to determine the key that they each can. But it doesn't protect you from active man in the middle. That is, if you don't have authentication, then somebody could insert themselves actively in your connection, and you would establish a key with them, and then they would establish a key with the other endpoint and be able to decrypt your traffic.

So this is why there's been discussion in the working group. It's like, well, okay. So, yeah, it would be better to encrypt everything, except how do we explain this to people? Because, see, right now when we talk about a green title in the URL, meaning that you're encrypted, you're secure, that security assertion currently means we verified the server's identity. But if we're going to do opportunistic encryption, as it's called, or relaxed TLS, then we're not going to be verifying the server necessarily. Sort of there'll be like a second grade of security. It's like, well, it's not encrypted, but - oh, I mean, sorry. It is encrypted, but it's not authenticated, which means passive eavesdropping, like in a Starbucks open WiFi, I mean, that's useful to not have everything in cleartext.

But so anyway, so you can see that this is a very complex issue. What does this mean, essentially, if we - how do we convey this, and what do we want HTTP/2 to be? What they're suggesting is that v2 will be like HTTPS is now. And only if you're v2, you're secure, meaning you have privacy from encryption and you have strong authentication of the endpoint. And but then, of course, clients that didn't support HTTP/2, or servers that were not offering an SSL certificate, wouldn't be able then to claim, in their handshake, they would not say I support HTTP/2. They'd be supporting v1.1, which is what we have now, and offering no certificate.

So then the question is, well, okay, is there then a place for relaxed TLS? And part of this dialogue, this argument, is no, let's just - HTTP strict transport secrecy or security, remember we've talked about STS, Strict Transport Security, provides a lot of strength. And listeners will remember when I asked Google to, and Google agreed, to add GRC into Chrome so that Chrome will never accept a nonsecure connection to GRC. It is wired into the browser. So even the very first connection, which technically represents a tiny window of vulnerability, even that is protected from being eavesdropped on. And that would allow an attacker to get a wedge in, essentially, and cause other connections to the website not to be secure if the first one is allowed not to be.

Anyway, we've covered all this in the past in our podcast about HSTS, HTTP Strict Transport Security. So anyway, v2 of HTTP is coming along, and I'm glad. And I'm also glad that, again, this is another - more fallout from Snowden is a much heightened focus on privacy of HTTP web connections by default. We're not sure how we're going to get

that yet.

And Elaine sent me an interesting link. She ran across in her travels a court order which was using Freedom of Information to compel the government to tell us what it has in the way of an Internet kill switch. And the term "Internet kill switch" has come up from time to time. And it's like, okay, really? Is there such a thing? And apparently now it's unequivocal that something, a facility like that exists. I got a kick out of the fact that it's known as "Standard Operating Procedure 303," because the first thing that came into my mind was, which apparently results in the generation of 404s.

[popularresistance.org/court-demands-how-does-government-turn-off-the-Internet]

Leo: I don't think they were thinking about that.

Steve: I don't think so either. But as I understand it, and as the documents they're trying to get imply, our government, through Department of Homeland Security, does have the ability to shut down both public and private carriers of traffic on the Internet in the event of some emergency. And...

Leo: Hard to imagine what that emergency might be.

Steve: I just think that's a bad idea. I just, I mean, it's like, the more we depend upon the Internet, the less it's possible to understand. I mean, it makes us feel like a Third World country to have the ability to, like...

Leo: Yeah. I mean, first thing you think of is what happened in Egypt and Syria when the...

Steve: Precisely.

Leo: When people were rebelling, they immediately turned off the Internet. I think the theory is that one kind of cyberwarfare would involve a concerted attack, let's say on our electrical grid, and if it were felt that the only way to protect the electrical grid would be to turn off access. And it might be, I mean, who knows where that kill switch lies? It might be saying no access outside the U.S. Or it might be let's just shut down the Internet. I mean, it's not clear what that is. But let's say it says, look, nothing outside our shores can come in on the Internet. That might be a reasonable response to a cyber attack on our grid.

Steve: It absolutely must be, Leo, that there is the facility to sever the trunks.

Leo: Right.

Steve: The submarine cables, all those optical cables, and probably satellite backup. There's got to be the way, there's got to be the means to isolate the U.S. from the rest of

the globe. I'd be shocked if that didn't exist. I hope it doesn't extend to "intra," as opposed to "inter." That is, intra-U.S. I would sure hope stays connected. I mean, I think it would, really, I mean, I don't think the government even knows now what would happen if they tried to kill the Internet intra-U.S. Can you, I mean, can you imagine how dependent we are on our connectivity?

Leo: Yeah. I'm sure the argument goes something like, well, if they kill the grid, the Internet does down anyway. So let's protect our grid by shutting down the Internet temporarily. But what it does raise is the specter of these countries that decided that instead of allowing criticism and open conversation, they would just shut down the ability to social...

Steve: Well, yeah, and we can no longer question whether or not hostile, potentially hostile foreign states...

Leo: And we know that's going on, yeah.

Steve: ...have cyberwarfare initiatives, and we know we do. Which still seems like science fiction to me. It's like, okay.

Leo: Given the cozy relationship that the government has with Sprint, AT&T, T-Mobile, and Verizon, they're probably - the companies that run the backbones, chiefly Sprint, they probably have a - they can make a call and say, hey, could you just shut down the backbone for a little bit? We're having some trouble here. Right?

Steve: Yeah. No, I'm afraid that's true.

Leo: I'm sure that exists.

Steve: Yeah, yeah, it has to. Here's a little random tidbit that I got a kick out of. This was actually thanks to Simon Zerafa, a frequent contributor to my Twitter feed. It turns out that some researchers at MIT were curious about what the optimizer in the very popular GCC compiler - what's the GCC stand for? Probably GNU something compiler [GNU Compiler Collection]. Anyway, it's the compiler that everybody in the, I mean, that's the compiler that the whole open source community operates on. It turns out that the compiler that is currently operating everywhere, that everyone is using to compile open source code, has been discarding what it considers to be ambiguous or do-nothing code.

But it turns out those are deliberately created security-relevant checks, including those things we talk about here all the time, null pointer checks and pointer overflows. And you can imagine how this could happen because a sufficiently clever compiler stands back and looks, it's like it's processing the flow and, to what degree it can, understanding what's going on in the code. And so it sees something that is, like, checking to see if a pointer is zero or not. And but then nothing happens, like with the result. I mean, an exception gets raised, but it doesn't see that there's a clear effect. Or a pointer is being checked for bounds, but when in its less than infinite wisdom it looks to see what effect

that has downstream, it says, well, this has no downstream effect, so it removes it. So it turns out that the actual object code that has been generated for some period of time, I mean, substantial code, doesn't contain the security checks which the source code authors deliberately put in.

Leo: I'm not sure how that could happen because this kind of optimization, as you say, it takes out loops that do nothing. Well, I can't imagine any optimization that does nothing. Is it instrumentation that's not turned on that...

Steve: No, it's present. But, see, you might argue that this optimizer needs to be fixed, and one certainly would.

Leo: Or that the security code has to be written better.

Steve: It's looking at the downstream consequences to see whether the code has an effect on the outcome. And arguably, security checks don't. I mean, they're doing something. They're validating assertions. But they do not affect the outcome. And that's the point. And the optimizer says...

Leo: Yeah, I'm not sure I buy that. I mean, don't you put in test code all the time that doesn't affect the final outcome, but would throw a flag if there were an error?

Steve: Well, there is a really...

[Talking simultaneously]

Leo: Bad optimization is what it is.

Steve: There's a really nice PDF that I linked to in the show notes from the MIT guys. They've gone through and found thousands of packages in the current Debian repository...

[pdos.csail.mit.edu/~xi/papers/stack-sosp13.pdf]

Leo: It's ironic, isn't it.

Steve: ...that have all been ruined by this. So, yeah, I mean, it's definitely real. In fact, there's their comment in that SecurityCurrent page talking about it, with a link to the study.

[securitycurrent.com/en/research/ac_research/mot-researchers-uncover-security-flaws-in-c]

Leo: So they give as an example null pointer checks. But you don't want to optimize out null pointer checks. You always want to check for null pointers.

Steve: Yes.

Leo: That's not - that's a bad optimizer. Or pointer overflow checks.

Steve: But my point - es, yes, it is. But it is in the GCC compiler. It is doing this. So it's a good thing they found out that we need to remove that and then recompile everything.

Leo: Yeah. Undefined behavior and unstable code. That's just a - that's a bug. I think that's a bug in the optimizer. That's terrible.

Steve: Well, it was - someone wrote it in there and thought, oh, look, this will remove something that doesn't have - this doesn't have a consequence, yeah. Okay. Two really interesting new payment systems that I received a lot of information about. And that...

Leo: I know where you're going with one of them, and I think you might have fallen prey to a very good publicist. This Coin thing. I would love to hear about the security implications of it.

Steve: Well, okay. First of all...

Leo: Do it in order because, I'm sorry, I didn't mean to...

Steve: No, it's okay.

Leo: But I've never seen more publicity in one day. It all happened in one day. And people came into our chatroom and said, "What about Coin? What about Coin?" And I think there was a very good, concerted effort to raise attention and awareness to something that is a nonstarter. But go ahead. Let's talk about it.

Steve: Yeah. So here's what it is. And anyone who's interested, it's OnlyCoin.com, and - beautiful website, and a minute-and-46-second video which explains it. And it's intriguing. And I think I know how it works. That is, knowing how it would have to work.

Leo: The theory is that a credit card uses a mag stripe reader, and that you could have in one credit card a programmable mag stripe.

Steve: No.

Leo: No?

Steve: No.

Leo: Okay. Because, by the way, this is a nonstarter in every country but the U.S. where they use Chip and PIN.

Steve: Correct. So the way it looks like it would work, Leo, is what you said.

Leo: Right.

Steve: And what it actually is, I think, is very clever. Which is why I take my hat off to these guys. So what you get is you get this credit card that looks exactly like the - remember the VeriSign one-time password card? It was a credit card-size thing that had a button on it. And when you pressed the button, it would give you a six-digit output. So this credit card has - it's got an eInk display and a merged-in battery that lasts about two years. It runs Bluetooth 4 LE that we were talking about last week, the low-energy version, and it ties to your cell phone.

So in operation you have a credit card reader, very much like Square, the little fob that plugs into your earphone connector, or earphone and mic connector, on a smartphone. And then you scan your various cards one at a time through this reader gizmo into your smartphone. And then there's all your credit cards in your smart phone. And then through Bluetooth LE, your smartphone, they say, loads all of those credit cards, however many you've got, maybe 40 credit cards, into a single card. Now, you then - there's a button on this single card where you press it, and it cycles through which credit card you want this to be. And then you're able to swipe that stripe through a standard mag stripe reader, and that card gets billed.

Now, you would think, I mean, from what I just described, you would think this is some amazing card because, as you press the button, it's like reprogramming the mag stripe. Well, okay, that can't - we don't have that technology. That's nano stuff with Spock and Kirk. And it doesn't have to do that, which I think is why this is so brilliant. This is all sort of a user interface. This is just my theory. This is all just a very clever UI where the card appears as you press the button, and the eInk display changes to show you which card it is. This is a little sleight of hand. The card itself is statically programmed with their own credit card.

Leo: Ah, that's what it is. That's what it is, yeah.

Steve: And so they are a proxy for all these credit cards.

Leo: This is not new, by the way. There are companies that do this.

Steve: I think it's - oh, no, I did not know that. That's cool.

Leo: They don't do it with all the sleight of hand. They just have you have a credit card that you charge everything through, and then it splits it out into whatever you choose.

Steve: Ah. And so what these guys have done is they...

Leo: That makes sense.

Steve: Doesn't it? I think it's so cool. So as you press the button on the credit card, it cycles through showing what card it's going to be. It tells your phone using its Bluetooth link. Your phone tells them using your WiFi or cellular link up to their cloud-based service. Then, when this card gets swiped, the charge - and they've established themselves as a merchant, as a credit card company. So the charge goes to them. It sees what the current setting of the card is, and then it acts as a proxy, sends the charge off to BofA or Chase or whomever.

And all of this happens in real-time. I mean, remember, I wrote an eCommerce system for SpinRite processing, so I've done all of this. This happens in real-time. That comes back to them. They then bounce it back to the merchant and say accepted or declined or whatever, and charge the card. So anyway, I just think it's just - I assume that's what it's doing. It's not reprogramming the mag stripe on the fly. We don't have the ability to do that. That would just be crazy.

Leo: I didn't even think about that, but of course you're right. That's nuts.

Steve: Yeah. But it's just like - and I do think it's a little weird because, I mean, I love the idea that you press the card, and it shows the change. But then you hand it to the waiter to pay your bill. How do you know he's not going to press the button, too?

Leo: Right. And...

Steve: And they say, oh, we made it so it's not easy to press it by mistake. It's like, eh, okay.

Leo: And now you know the waiter is going to say, what the hell's this? Give me a credit card. And it's not going to work outside the U.S. because chip and PIN. And, I mean, the problem I have with this is that they're really lobbying people to give them money now. If you do it now, you'll get half off.

Steve: Yup. You have 23 days left of 50% off.

Leo: Which means giving them $55 for something that, if you ask me, will never see the light of day. So I just would warn people to be, I think, cautious.

Steve: Yeah, oh, yeah. I should say I'm not endorsing this except I just love the hack. This is a clever hack where...

Leo: Yeah, I like that part. I think your - and you're so good at reverse-engineering this stuff. That makes a lot of sense. And in fact that kind of thing exists. There are these special cards that are kind of, you know, allow you to have a variety of affinity cards. But you do it online. It's a little bit slower.

Steve: Yeah. And are there some, like, with an iOS interface, so you just set the card on...

Leo: No, no, no, no. They're all just plastic. You would then - I should find it because there's a couple of these out there. Somebody sent me a link. My real problem is this was - I smelled something funny because I've never seen such a successful one-day campaign to get attention. And everywhere, from the Christian Science Monitor, USA Today: "Coin reinvents the wallet." "Coin changes the wallet." And I see...

Steve: Well, and it's a slick video, Leo. You watch that video, you know, because it pulls the sleight of hand. It makes you think that you've got a programmable credit card. It's like, wow. That's not how they do it, but it's the way it looks.

Leo: And that bothers me, too. In fact, Kickstarter's banned these slick videos. They say you have to have an actual prototype in your video because - and this is not on Kickstarter. And they bill you immediately. This is their own website, and because they don't have the rules of Kickstarter, they don't have to tell you anything negative about it, like the fact that the credit card companies - Visa, MasterCard, et cetera - could immediately clobber this because it...

Steve: True.

Leo: Saying no, nice try, boys.

Steve: You're right. They could easily say we're not accepting charges from you, and it's game over.

Leo: Yeah. I think there are security implications, especially if you're sending this through Bluetooth LE.

Steve: No. I see no security problem at all.

Leo: Because it's in your phone. All the data's in your phone. It's never anywhere but your phone.

Steve: No, it is also in the cloud. All the cards need to be in their facility in order for them to be able to proxy those cards on behalf of their card.

Leo: Yeah. So if you trust Coin...

Steve: Yup.

Leo: Which is a big if, they're saying...

Steve: On the other hand, well, yes, your debit cards are at risk. Remember that your credit cards are always backed up by the credit card company.

Leo: And they've changed the law on the debit cards, too, so that there's a limit on how much you can lose.

Steve: Ah.

Leo: I think it's 50 bucks.

Steve: Good.

Leo: Nevertheless, I just think this is a nonstarter, and you're giving 55 bucks to a company that's going to disappear.

Steve: I wanted to talk about it because I thought it was clever...

Leo: And I love it that you figured out how they do it. And I think that's, you know, I'll tell you where I failed is I just read it and go, oh, yeah, reprogram the mag stripe. Of course.

Steve: Okay, that's No. 1. No. 2 is on Kickstarter. And it's called Loop. I have a problem with the fact that they acknowledge it only works 90% of the time.

Leo: Oh, boy.

Steve: So it's like, well, okay, but that's bad. But what it is, is again clever. So what we know about a mag stripe reader, and I'm sure all of our tech-y listeners know this, you look in that little slot, and you see a magnetic recording, or in fact reading...

Leo: Read head. A read head, yeah.

Steve: A read head, yeah. And you can see it. And so that, I mean, I often look in to make sure I've got the card oriented right because you want the stripe side of your card to be facing the read head. So what we know about - and people who remember that animation I did, that JavaScript, when I was playing with JavaScript animation, showing magnetic flux reversals and so forth, and obviously I've spent some time thinking about the way hard drives read and write. You have, on the stripe, you have reversals of magnetism on that stripe. And that induces a magnetic field in the windings of the read head. This read head is like - think of it as a very sensitive magnetic microphone. And so the completely passive, simply magnetized stripe is able to essentially send a signal into the read head, which it picks up. But so could an inductive coil. And that's what these guys have done.

Leo: Ho ho.

Steve: So they have either a back that you can add to an iPhone containing an inductive coil, or a fob that you plug on, again to the headphone/microphone connector, that contains the inductive coil. So the idea is you, again, you select a credit card. And so now you're at the supermarket, and you're standing in front of the little swipe-your-card terminal. Instead, you bring your phone within an inch of the slot and press a button, and it sends the magnetic signal that the card being swiped would send magnetically across the air gap to the head, which picks it up. So, very clever.

Leo: Yeah. That's neat.

Steve: Yeah. Now, the problem is...

Leo: Only works some of the time.

Steve: Yes, 90% of swipe pay terminals. So they say, oh, now you don't have to carry your credit cards with you. Well, yes, you do, for the 10% when it doesn't work.

Leo: Well, and the 50% of the time when the clerk says, "What are you doing? Get away from my terminal with your phone."

Steve: Well, and you can't give it to the waiter in the restaurant. You don't want to hand your phone to him. Oh, it's unlocked, and here you go. Go hold this near the terminal. It's like, uh...

Leo: The other problem I have with both of these is it's kind of solving a problem that doesn't exist. Is it really such a pain to have a credit card? I don't...

Steve: Well, that's just it, too. I have a main card and a couple backups. But that..

Leo: Yeah. You carry a wallet anyway; right? Just keep a credit card in your wallet. What are they solving? What massive problem - and I can't believe they raised $123,000 on this.

Steve: I know.

Leo: This is my big problem with Kickstarter.

Steve: They wanted a hundred grand. They got 123,788 when I checked.

Leo: Unbelievable.

Steve: Apparently it's two veteran charge card guys. One of the guys pioneered mag stripes. It's like, that's - maybe he's looking for...

Leo: Well, that's neat.

Steve: Yeah. I mean, so they've got their tech down. Anyway, it's cool on Kickstarter. It's Pay with Loop, for anyone who's interested in taking a look at it. And of course the other problem is that you need to then have this bulky back, you have like this bulky case for your iPhone which doubles as, like, a backup charger because they thought, well, let's give it some more functionality because otherwise we're asking a lot.

Leo: Which is worse, carrying a wallet with some credit cards or a bulky back on your iPhone?

Steve: I know.

Leo: I just am baffled by...

Steve: I know.

Leo: And this is my problem with Kickstarter is I think people are a little bit suckered sometimes by this stuff. But anyway, okay. Well, thank you for the analysis.

Steve: Yeah. So we've got two interesting...

[Talking simultaneously]

Leo: ...to know about it. Yeah, they're interesting.

Steve: Two interesting payment solutions. Oh, a bit of errata from last week. Elaine wrote: "Steve, you may have had feedback on this already, but you accidentally named PandoDaily as the fraudulent bitcoin exchange, rather than as the high-tech newsletter than ran a story about the fraudulent Chinese bitcoin exchange GBL."

Leo: Oh, golly. Sorry, PandoDaily.

Steve: Yes. So sorry, PandoDaily. She said, "I played with it in the transcript" - so thank you, Elaine, for correcting it preemptively. She said, "but you might need a verbal correction."

Leo: There you go.

Steve: So, done. She said, "Just mentioning it now because I'll have forgotten all about it by tonight when I send the transcript." So she sent me a little piece of email right when she encountered it and said, whoops, might want to fix that.

My iPad mini arrives in two days, Leo, on Friday.

Leo: Wow.

Steve: So I'm excited for that. I've not seen any mini retina, so I'm very anxious. I did go by my local Verizon store and look at the current mini, which, boy, does it look grainy. Oh. We're all - that's just like how did we ever look at this and think this was good? Wow.

Leo: I have two minis. I like them a lot. Although it's interesting, the guys at DisplayMate have done color accuracy analysis, and because Apple's decided to go with this indium gallium zinc oxide technology, IGZO, instead of a more traditional LCD technology, they say the color gamut's not so good. In fact, they're beaten pretty handily by the Nexus 7 and the Amazon Kindle Fire HDX.

Steve: I don't care about that at all.

Leo: I don't think you care about that.

Steve: No.

Leo: The crispness is very nice.

Steve: Oh, I can't wait. And, now, I also just yesterday picked up the rumor of next year a 12.9" display iPad. You've no doubt seen word of that?

Leo: Yeah, we've heard the rumor, the iPad Maxi, or the maxi pad. But, you know, I'm not sure I buy that.

Steve: Well, and for me, the use case would be, I mean, I've got one of my iPads sitting next to me, like where I watch TV. And I'm grabbing it all the time to look things up or to check things, or what's the weather, what's the movie schedule, blah blah blah. So I could see a larger pad that was deliberately non-portable. I mean, it almost doesn't need to have a battery in it.

Leo: Yeah, but you bought Kindle Fire DXes, too, which have since been discontinued because nobody else wanted them.

Steve: Oh, I love my DXes, Leo.

Leo: Yeah, but I don't think anybody bought them because Amazon's not making them. So, yeah, I think this is a rumor. But you know with Apple rumors one never knows.

Steve: So I wanted to also just mention that my recommendation of Incipio DualPro case got a huge amount of positive feedback. They began to arrive, apparently from my mention last week, people started getting them in the last couple days. And many people said thank you, thank you, thank you. One person, Rick, tweeted, and his Twitter handle is @slartibartphast, or so...

Leo: You know where that comes from; right?

Steve: We know where that comes from. And he said: "Steve, oh, Steve, iPhone since 3, never a case, never dropped." And then he said, "I also hate plastic on couches."

Leo: That's what I liken it to, but...

Steve: So, I mean, and I totally understand. The phone by itself is just exquisite. But I just - it is trying to leap from my hands. It's trying to increase its level of entropy. So I just don't want to let it. I did have some people saying, asking me, what do I like for the iPad? And something I've never mentioned before that I really like is something called a GelaSkin, G-e-l-a-S-k-i-n.

Leo: Oh, yeah. You sent me one.

Steve: Yes. I'm a huge fan, huge fan of the GelaSkin. It is an - I even - here's one on

my iPad 1 that I have.

Leo: Pretty. That's Keith Haring.

Steve: Yup, exactly. They have real artists, a huge selection of skins. And so what this is, for people who can't see, it's a sticky, but it's removable, a relatively thick and tacky backing. So I put it on the backs of the Pads, and I have it on my Kindles. You can get it for phones, Kindles, iPads, iMacs, all range of different devices. So it gives it a really interesting sort of - you can be making a statement with it. "Stay calm and carry on" is one that they've got, and many others. But it's grippy. So it gives you some sort of a tacky back. It's sticky enough that, for example, you could just hold - you just put your hand behind an iPad, and it'll just stick to your hand without slipping, which the metal backing wouldn't otherwise.

Anyway, so check out GelaSkins.com. They're not inexpensive, but they last forever, and you can peel them off and remove them. So it's not like political bumper stickers that you end up advertising the candidate who lost, to your embarrassment, for...

Leo: You sent me, yeah, you sent me these for my Kindle, way back when. But they didn't do cases back then.

Steve: Right. And so I like those guys a lot.

Leo: Yeah, neat.

Steve: And, oh. I also wanted to mention, just for people who don't follow me on Twitter, I often get questions from people that I'm happy to respond to except I have, I don't know, many thousands of listeners now, 30-something, and so I just can't do @ mentions back. I don't feel like I can do that, or my Twitter feed becomes conversations with people rather than sort of announcement stuff that I want to keep. So for what it's worth...

Leo: Little tip, though, Steve. Little tip. If you have at the very beginning of your tweet the @ sign, if that's the first character, then the only people who will see your @ reply are people who mutually subscribe to both of you, and of course the person you're responding to. It does not enter your general feed.

Steve: Oh, no kidding. So when people are sending me @SGgrc's, that's not going out on...

Leo: If it's the first character. If it's the first character, it's only seen by people who subscribe to the two of you mutually, and both parties. That's why sometimes you'll see a tweet with a dot @.

Steve: Yes, yes.

Leo: That's intended to make it public.

Steve: Oh, because then the @ sign is no longer first.

Leo: If the @ sign is first, that's what happens.

Steve: Okay. Well, thank you. Thank you, thank you.

Leo: Now, of course you can, if you - yeah. But that's, now, if you go direct...

Steve: So why have a DM, then, if you have that, which seems the same?

Leo: Well, DM is a step farther because DM's only seen by the two parties at either end.

Steve: Ah, purely, purely private. Okay. So people sometimes respond to me. I'm unable to DM you. And I'm thinking, well, yeah, sorry, but I'm not following you.

Leo: Because you have to follow them; right.

Steve: Okay, cool. Well...

Leo: You don't want DMs.

Steve: Now I'll be able to reply to everybody without worrying about it messing up my Twitter feed.

Leo: Yeah, it's kind of - that's what Twitter did kind of for exactly this reason.

Steve: Neat. I like that. So Mike Willis, tweeting as @mikewillis, or Wills, sorry, Mike Wills said: "Based on @SGgrc recommendation, I'm running SpinRite on my wife's laptop to get the HD in prime condition again. And I just - I thought this was a perfect segue because I did see, I think, an increase in SpinRite sales after talking last week about, first of all, sharing the testimonial where that hugely damaged drive, that laptop was recovered by SpinRite, and got back data, corporate data that was otherwise in great danger.

But this notion of preventative maintenance, I also, in the last week or two maybe, there was someone who was running SpinRite on a brand new drive. Oh, I remember, it was forwarded to me through Greg because he had some other detailed questions about SpinRite, so Greg forwarded it on to me. But he bought a brand new drive and didn't - he

hadn't run across the SMART page. In the last few months I wrote - well, more than that, but relatively recently, I did a couple pages that it clearly explains SpinRite's SMART monitor, the Self Monitoring Analysis and Reporting Technology, SMART, and how there are some bars which get pushed down to reveal red. And that's not good when that happens.

And so this guy had a brand new drive that he ran SpinRite on, and three of the bars got pushed down. And it seemed to be generating lots of seek errors and lots of corrections, more than he felt comfortable with. So he returned the drive. It was replaced with one. And when he did the same thing, the bars did not get pushed down. So this is one of the coolest things, I think, about SpinRite 6, is that it's nice to have the SMART stuff there. But it really only means something when the drive is under load, when the drive is being asked to do work. That's when the drive notices that it has problems that it wouldn't otherwise notice.

And so what you want is you want this nice synergistic combination of SpinRite putting the drive under load, constantly reading the drive's SMART feedback, and showing it to the user. So the drive is seeing that it's having problems because SpinRite's asking a lot from it. And the drive is - these bars that are being pushed down are the SMART parameters which are being suppressed, or depressed, by the work.

So if people already own SpinRite, and you haven't run it for a while, I mean, it is tremendous preventative maintenance. But while it's running, or maybe after it's been going for a while, switch to the SMART screen and make sure that you've got green, or I think it's maybe cyan all the way across, and no red showing, because you want to get the red out. You shouldn't have red there. And if you're interested for more, I do have some documentation on GRC.com that shows a screenshot of this that I've highlighted sort of with, like, callouts, showing where all these things are and what they mean.

But anyway, this guy replaced his drive. And when he did the same thing, the second drive did not have its bars depressed because that drive's SMART system was not being surprised by the drive's brand new condition being untrustworthy. So many people say they run SpinRite on brand new drives before they deploy them, which this is exactly why you would want to do that. So it certainly does make sense. And we hear from people all the time, SpinRite owners, of course you can't prove a negative, but they're saying people's hard drives all around them are dying, but they run SpinRite preventatively, and they've never had a hard drive die on them. So we don't know that it's SpinRite, but it seems pretty likely that it is.

Leo: All right, Steve. RADIUS. This came from a question about another one of our sponsors, proXPN.

Steve: Well, actually - yes, yeah. Many, many - I'm sorry. I didn't mean to cut you off.

Leo: Go ahead. Go ahead.

Steve: Many people have - I've seen tweets where I've been mentioned in Twitter where they've been responding to proXPN saying, wait a minute, 12 characters? What kind of security is that? Come on, get real. I'm using proXPN for security, and you guys apparently have lousy security. And proXPN responds and says, well, no, we're using RADIUS. So that's a limitation of the RADIUS setup, but that's not bad. And so finally I

thought, okay, look, I'm just - let's talk about this because it's something we've never discussed on the podcast.

To give you an idea of how venerable RADIUS is, it is an acronym. RADIUS stands for Remote Authentication Dial-In User Service. So this dates back 22 years, to 1991. And probably the best way to think of a RADIUS server is sort of similar to DNS, in the same way that DNS is used by clients to look up the IPs of websites using their domain name. And in that sense the DNS service is centralized, that is, people all over the place can use that server to perform their lookups.

RADIUS provides an authentication service, a username and password authentication service. And again, it's centralized. It uses UDP protocol by default, just like DNS does. And there is a packet-level specification. And the one place users may have seen it is in their own routers because, for example, in a corporate environment, a corporation might have WiFi routers scattered all over the facility, many of them, on all the different floors, in order to get blanket WiFi coverage.

Well, it's obvious - well, several things are obvious. First of all, in that setting, you would not want to have a single username and password or a shared key because then all the employees would have to be using the same shared key, preshared key. And changing it would be horrendous, if you ever needed to change it for some reason. And when an employee left the company, they would leave with the knowledge of what the preshared key was that they had to put into their laptop once, and that's a problem.

So there's a different model, for example, in this mode of how to authenticate users to WiFi. And that is, every single one of those individual WiFi routers, running just like the WiFi routers we all have at home, instead of using a preshared key, you use RADIUS. Sometimes it'll say RADIUS and give you a field to put in the RADIUS IP. And this is exactly like DNS, where you put in the DNS server IP. Or maybe it'll say WPA Enterprise, or WPA2 Enterprise. And what that Enterprise typically is, is code for a centralized authentication service.

And so what happens in this mode, in this multi-floor corporate environment, is somebody walks in with their laptop, and they want to associate with the router, the WiFi router nearest to them. The router doesn't contain, itself, a database of every username and password within the whole system. And so obviously, if we're going to move away from everybody having the same preshared key, what we do is we switch over to a model where everyone has their own username and password for logging into the corporate wireless network. But now you have another problem. If every user has their own username and password, and a given user is at a single access point, what is it going to have? The database of the entire company directory? No.

And first of all, these are little tiny underpowered sort of embedded devices. They don't have the local storage to store the whole database in order to provide local authentication of every corporate customer who might come to them. Instead they pass that on. In the same way that they don't have all of DNS, they don't have any, in fact, of the accounts for the corporate environment. They use RADIUS. RADIUS is the standard in the industry. Routers understand RADIUS, switches, port forwarders, all kinds of equipment knows RADIUS.

There is something called FreeRADIUS - it's probably .org, but if you just Google "FreeRADIUS" you'll find it - which is a terrific open source implementation, and free, obviously, of this RADIUS service, which has grown over the last 22 years. And, I mean, it's RFC-crazy, with a whole series of RFCs that have been updated over time as the technology has evolved from the original dial-in approach to VPNs and networks and

tunnels and crypto and everything that we've got.

But so, and back to our corporate model, somewhere IT manages a single RADIUS server which is on the front of a database. And this could be a SQL database. It could be LDAP. It could be a flat file, any kind of database that RADIUS understands. And today's RADIUS servers understand how to talk to all of these. And so the user wants to authenticate with their own unique username and password. That's given to the local WiFi router or hotspot. It then knows the corporate IP of the corporate RADIUS server. It encrypts the username and password because it has a preshared key with the RADIUS server. So it encrypts that over the network. The RADIUS server receives it. It turns around, does a database query to say is this user authenticated, and there's lots of granularity.

And this thing is, like any spec that's been around for 22 years, it does all kinds of other things, too, that the IT may want. It also does accounting, is another facility, not just authentication and authorization, but also with accounting of how much bandwidth or how much time and so forth, depending on what the company wants. So the RADIUS server looks it up, turns around and sends another packet back to the endpoint saying, yes, this user is authenticated, or no, they're not. And the connection goes from there.

So now the same model, exactly that model, similarly makes sense in a global setting, where any entity like proXPN that has servers deliberately scattered around the world has a single customer base, that is, a single set of authorized users who are able to log into any of those VPN endpoints. Identical model. So proXPN manages this single database, which is their customer or account database, that has a RADIUS server on it. And similarly, all of the access points spread around the globe, when an individual connects to it, they authenticate with their credentials, their username and password, for proXPN. The radius client that's running in that VPN endpoint sends a UDP packet, after encrypting it, down to the centrally located RADIUS server which looks the user up and then decides whether this is somebody they know, sends it back to the RADIUS client, that then authenticates the user and allows them to establish a VPN tunnel to the endpoint.

So with a 12-character maximum length - which I won't defend the fact that that's not long. We all know that's not long. But the nature of the lookup process means that anyone trying to attack this is necessarily, unless they were to hack proXPN's backend database, is inherently doing an online attack. So you'd have to guess the password, and then that goes to the VPN endpoint that sends the UDP packet down to the server, which makes a database query, looks it up, says nope, wrong, bad guess, sends that information back to the VPN endpoint that then says, sorry, we can't authenticate you. There went one guess.

Well, so that just makes brute-forcing RADIUS-based passwords virtually infeasible because it's got to be an online attack. They allow upper- and lowercase of alpha AZ. So that gives you 52 characters, plus 09, so that's 10 more. So now we're at 62, plus !, @, #, and $. So that's four more. So we're now at 66-character alphabet. So a 12-character password with a 66-character alphabet, assuming good entropy, that is, that you arrived at it randomly, gives you 6.8 thousand billion billion passwords. Which is a lot of passwords. That's 72.53 bits of entropy, which is not 128, but 72. And for online attacks, that's way more than enough. It's certainly the case that they need to have good security and protect their backend and their database.

But finally, I'll just remind everyone, as I did mention at the top of the show, that this is more to protect them. Users are not storing anything in a proXPN account. This is them, this is the user gaining access to the account. And if the worst happened, and someone

was abusing a password that escaped from them, well, they could change their password, or they could contact proXPN and say I think my account's been hacked, set me up with another one. So the whole notion of logging in with an authenticated account is to protect proXPN from abuse of their global network, rather than the user. Again, it's not like the model where you're wanting protection against cloud access to all of your cloud-stored files. All you're doing is you're saying this is really me, somebody who has an account. So let me connect to your server.

But anyway, so I wanted to clarify that, to respond to all the tweets that I've seen from people who are annoyed by the fact that there's a 12-character limit. I don't know any details about the backend database, the details of RADIUS, why they chose a 12-character length. Probably it's just that they recognize an online attack is infeasible. They've got sufficient security for their own needs on the back end. And using RADIUS is what's going on. It's very different from a single website which is, as we know, storing characters, or could be storing characters in an insecure fashion, and damage could be done to us if our account were hacked on that one web server. Instead, RADIUS is used to distribute authentication. And we're using it to gain access to their VPN network. So it's not a problem, although it's something we've never talked about before. So I just felt like it was a loose end. And that's how RADIUS works.

Leo: Is it widely used?

Steve: Oh, my god, Leo, it is everywhere. I mean, it's funny, too, because we see DNS because it impacts end-users. I think we've never talked about RADIUS because it's just never had an impact on most of our customers. And in fact that's why I was moved to talk about it was because it's RADIUS, remote authentication, or distributed global authentication, which is like poking itself out into the user's experience through proXPN's use of it. But otherwise, yeah, we just never encounter it. I'm sure that lots of our listeners who are in IT go, oh, yeah, we got RADIUS. That's the way everything authenticates within our entire corporate network. Probably every corporation, if they're not using some proprietary Microsoft active directory solution, RADIUS may well be what they're using. And because, I mean, it is the standard.

Leo: It's a Cisco solution.

Steve: Cisco adopts it. Nobody owns it. So Cisco's just one of many people who have it. In fact, I have a note here in my notes where I was looking up, just sort of getting a sense for it. Cisco says: "DHCP Server RADIUS Proxy." They said: "The Dynamic Host Configuration Protocol (DHCP) Server RADIUS Proxy is a RADIUS-based address assignment mechanism in which a DHCP server authorizes remote clients and allocates addresses based on replies from a RADIUS server."

So there's an example where RADIUS is the backend, and the DHCP server is making RADIUS queries to a RADIUS server. And so essentially RADIUS is the database that DHCP is drawing on, rather than DHCP server having the database itself. I mean, so this is just pervasive within enterprise-class IT. RADIUS is what everyone is using.

Leo: And is it the case that a bank wouldn't want to use it, or somewhere where you are trying to protect something behind that password?

Steve: Oh, yeah. I would think banks probably are using it to authenticate all their own employees. But it wouldn't make sense to expose it to the world.

Leo: On the web, yeah.

Steve: Exactly. Exactly.

Leo: So this is a specific reason, is because you're logging into an account at proXPN. You're actually getting on their network, in effect.

Steve: Yes.

Leo: So that's a logical thing for them to use.

Steve: Well, and if they didn't do this, then they would have to have some database replication technology.

Leo: Which could be, as we have learned from Google, more of a problem.

Steve: Exactly. Where every one of their globally located VPN endpoints would have to have the entire database of all their customers, and then they'd have to be keeping that current all the time. Whereas, by using RADIUS, their database is in one location. And when, as customers come in and go out, they're immediately authenticated or deauthenticated across the entire network. It's just - it's very cool. And that's - it's been going on silently in the background of the Internet for 22 years.

Leo: Wow. Steve Gibson explains all. He knows all. And you can follow him at GRC.com, that's his website, where SpinRite lives, the world's finest hard drive maintenance and recovery utility, and all the freebies he gives away all the time. You can also follow him on the Twitter, @SGgrc. You can also ask questions at GRC.com/feedback. Next week, if all goes well, a Q&A episode.

Steve: Unless something catastrophic happens.

Leo: Unless the world ends. Or something less than that. What else? Oh, yeah, if you go to GRC.com, Steve has 16Kb audio of the show, the smallest version of audio for the show, as well as transcripts, which are the smallest version of all, for your perusal. We have higher quality audio and video available at our site, TWiT.tv/sn. You can watch us do the show live, every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, that is 19:00 UTC. And it will be changing, as I mentioned. January 8th we will shift to a new time. Tuesdays at 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 21:00 UTC will be our new time starting January 8th. Just a little program note. What are we doing for the holidays? You said you had something planned, I think.

Last year we did the Portable Dog Killer or something. No, you played a tape, an old tape.

Steve: Yup.

Leo: So start thinking about it.

Steve: Yeah, I will. I have all of the original appearances that I made on The Screen Savers.

Leo: That would be fun.

Steve: You and Kate Botello and...

Leo: Yeah, Trouble in Paradise.

Steve: Kevin was about nine years old.

Leo: Click of Death.

Steve: Uh-huh. Yeah. So I thought maybe I'd put those together, and that would be fun. Another blast from the past.

Leo: We're trying to get to know our users. Don't forget our survey, TWiT.tv/survey. And if you've taken it already, our hearty thanks to you. And we are looking for people for New Year's Eve. We're going to do a 24-hour of New Year's. Steve will be with us in-studio for that, at TWiT.tv/nye, if you'd like to participate. We want countdowns all over the world for 24 hours. Thank you so much, Steve. Always a pleasure. We'll see you next week on Security Now!.

Steve: Thanks, Leo.