Proactive Privacy, Really!

Description: This week Steve and Leo discuss Symantec finding 40 past attacks explained by the Vault 7 document leaks, an incremental improvement coming to CA certificate issuance, and Microsoft's patching of a zero-day Office vulnerability that was being exploited in the wild. They ask, "What's a Brickerbot?" They cover why you need a secure DNS registrar, This Week in IoT Tantrums, a headshaker from our "You really can't make this stuff up" department, the present danger of fake VPN services, and an older edition of Windows reaching end of patch life. They continue with some "closing the loop" feedback from their listeners and a bit of miscellany, then close with a comprehensive survey of privacy-encroaching technologies and what can be done to limit their grasp.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-607.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-607-lq.mp3

SHOW TEASE: It's time for Security Now!. My goodness, we've got a lot to talk about, including Steve's new 15-volume sci-fi opus; lots of security news, including an IoT company that's completely out of control. And then at the end, I promise, and I know this because I'm speaking to you from the future, Steve will cover protecting your privacy as you surf online. A really great how-to, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 607, recorded Tuesday, April 11th, 2017: Proactive Privacy, Really!

It's time for, you got it, Security - well, because you downloaded it - Security Now!, the show where we talk about security, privacy, protecting yourself online. And we're going to do that this week, I promise you, with Security Now!'s host, Steve Gibson, the host with the most security.

Steve Gibson: Well, we tried to get to our main topic last week, proactive privacy. And as everyone knows, we spent two hours getting right up to the line. But I didn't want to shortchange the topic because I know it will be of interest to our listeners. In fact, there's already been some industry events that have followed on from what Congress has done. So anyway, today's title is "Proactive Privacy, Really."

Leo: Honest.

Steve: We're actually going to get to it today.

Leo: We promise.

Steve: And while I was putting the show together, it was funny, too, because people were still sending me topics and ideas and "Oh, Steve, did you see this?" And they were saying, "Now, I know you're trying not to have too much to cover this week so you could get to the topic, but..."

Leo: Oh, that's cute. They know.

Steve: Yes. But we do have a bunch of fun stuff to talk about. Symantec has found 40 past attacks which are explained by the Vault 7 document leaks. We're facing an incremental improvement in forthcoming CA, that is, Certificate Authority certificate issuance integrity, which is hopeful. Today is Patch Tuesday of April, and Microsoft patched in today's patch a very worrisome zero-day vulnerability in Office that was being exploited in the wild. We'll talk about that. There's a new bot in town that has been named Brickerbot. We're going to address the question of why you really need to secure your DNS registrar and how a Brazilian bank found out what happens if you don't.

We have This Week in IoT Tantrums, and a headshaker from our "You Really Can't Make This Stuff Up" department. The present danger of fake VPN services. An older edition of Windows today hit its end of patch life. We've got some closing the loop feedback from our listeners; a little bit of miscellany; and then, as promised last week and delivered this week, a comprehensive survey of privacy-encroaching technologies and what we can do to limit their grasp. So I think another great podcast.

Leo: And of course today is the day that Microsoft starts putting out the Creators Update for a lot of people. So it's not only a Patch Tuesday, but some people will start getting the Creators Update, and over the next few days you'll be getting a Windows 10 Creators Update. I know you won't be, but...

Steve: I don't even know what that is.

Leo: Well, it's just, you know, I warn you because probably next week and the week after and the week after we might have some other things to talk about having to do with Windows 10.

Steve: Indeed. And if there is, we certainly will.

Leo: Yes. All right, Steve.

Steve: So we have a really great Picture of the Week. I just got the biggest kick out of this. This shows two tweets. Tavis Ormandy first tweets: "I'm in Miami for @InfiltrateCon. Let me know if you want to catch up." And then the response from someone whose handle is "the grugq" says: "Where specifically will you be alone and unguarded? Asking for @LastPass."

Leo: That's funny. That's funny. Although I doubt that that's actually how LastPass feels. I'm sure they're grateful to Tavis.

Steve: Oh, of course not. Oh, absolutely, absolutely.

Leo: That's funny.

Steve: So Reuters picked up the news that Symantec had said to them, essentially in a press release, although they weren't naming the CIA as a function of their corporate policy not to do so, they had, after Symantec looked over the Vault 7 document leaks, pieces clicked into place, and they were able to go back and look at data that they had captured of previous attacks where they hadn't known exactly what was going on. And in 40 different instances, that is, cyberattacks against at least 40 organizations around the world, they found that the tools referenced in the leaked WikiLeaks documents that are ascribed, believed to be legitimately from a CIA document trove, matched perfectly the attacks that they had no attribution for until now.

So it shouldn't come as a surprise. And these 40 corporations that were attacked by these were spread out around 16 countries. So just sort of an interesting data point, that this is what you would expect. And props to Symantec for being in the business, having their feelers out, collecting this sort of data, and then saying, you know, we ought to take a look at the stuff we've collected in the past and see if any of this now makes more sense. And they found out, yes, apparently, indeed it does.

Leo: Now, were they widespread like hack attacks? Or were they targeted?

Steve: These were targeted against specific organizations.

Leo: Okay, good, good.

Steve: So Symantec probably has contractual security relationships with companies all over the place? And so they've got their monitors and probes in those companies' networks.

Leo: Got it, got it.

Steve: And so they're collecting data and archiving it in order to understand what's going on.

Leo: You have to wonder if they went to those companies and said, hey, by the way, you know that attack? That was the CIA.

Steve: So we've talked endlessly about certificate authorities, about how we've got kind of this creaky system that's the best that we know how to put together at this point, given the technologies and tools that we have, that essentially allow two parties that have never met before, meaning a server and a client, to arrange at least a one-way trust relationship - that is, for the client to be able to know that it is actually connecting to the server it believes it is thanks to a third party, the third party being the certificate authority, where the server has proved its identity to the certificate authority and the certificate authority has given the server a certificate, essentially an identity assertion certificate, which it gives to the browser. The browser then is able to verify the authenticity of the certificate by checking its signature, which can only be created, thanks to the magic of crypto, by that certificate authority, which thus proves through this chain that at one point the certificate authority was convinced that the server was who they said they were.

So that's the system. Unfortunately, there are all kinds of ways this can break. And we've talked about many of them. Our old-time, long-time listeners will remember the podcast where I had a meltdown when I looked at the size of the certificate authority root store in Windows because I remember when it was 11 trusted CAs, and it was like 400. It was like, what has happened? Because the nature of the system I just described means that anyone can sign a certificate for any server. And if you trust the signer, then you trust the server. And so that does create a problem with abuse.

And, well, for example, we have seen situations where a fourth certificate authority - that is, not a first, second, or third party, but a fourth party, someone completely unrelated - issued a certificate for, for example, Google.com. And we trust it because we trust all the certificates that party issues. So the problem that any of the hundreds of certificate authorities we trust can sign a certificate for any domain, that's an aspect of frailty in our system. Just last week an RFC that's been in the process for four years was formalized, and the CAB, the Certificate Authority Browser forum, or consortium, has formally required that, within six months, by September of 2017, all certificate authorities must honor a new record type for DNS.

We've talked about DNS. Our listeners know how excited I am about the idea that someday DNSSEC will happen, sort of in the way that someday IPv6 will happen. Eventually people will just give up or run out, and they'll have no choice. And so someday we'll have DNSSEC, which provides end-to-end security for DNS. We don't have that now. So that represents a weak link in our existing system. But the reason I'm so excited about DNS is that it is, once it's secured, it's this otherwise very well-designed hierarchical caching directory. And you can put all kinds of stuff in it, not just IP addresses.

But, for example, we're already storing text records to use for helping to diminish spoofing, where for example the DNS says for an email server that valid email from, for example, GRC.com can only come from this domain or this IP. And so somebody receiving email that wants to make sure it's not spoofed can make a DNS text record query for the SPF record and get what I am publishing as the only valid source for email from GRC.com. So that's an example of how we've already extended DNS beyond IP addresses to other stuff.

Well, what is coming is known - and so the typical IP is an "A" record, an address record.

Or if it's IPv6, it's an "AAAA" record. We also have NS records, name server records; and TXT, text records, and so forth. Well, we're going to get a CAA record which stands for Certificate Authority Authorization. And what this is, it's very much like the antispoofing for email, where the domain owner, like GRC.com, the domain owner is able to assert in the CAA record who is able to issue certificates for that domain. So of course my CAA record will say Digicert.com.

And so what this does is it publishes the name of my authorized certificate authority, who is the signer for my certificates. There's no enforcement here, but what this does is the CAB Forum is saying that within six months all certificate authorities must query a domain that they are being requested to issue a certificate for, for the CAA record. And if that specifies a certificate authority other than them, they must decline the certificate. So essentially it's a way of authorizing who you want to be able to generate certificates for your domain.

To the degree that that authorization is honored by other CAs, it will prevent a class of problems that we've had. It's not strong protection, and no way is it cryptographically amazing. It's like a hint or a clue. It's, you know, this is who my CA is. And you can do a comma-separated list. You can also have a null list. You can do double quote, semicolon, double quote [";"] which means nobody is authorized to issue certificates for the moment. In which case, after September, or actually at any time that you're publishing a CAA record moving forward, if you yourself want to ask your CA to give you a new certificate, you'll have to put them in that record so that they can see they're authorized. Once you've got your new certificate, you could change it back to a null list, which locks down any subsequent certificate issuance - but, again, for those authorities that follow it.

So this won't do anything to prevent deliberate malicious issuance of certificates. But again, it's something that is easy to be retrofitted in, which we normally have a problem with. It's very hard to - like, for example, DNSSEC. It's having a hard time because we already have DNS. IPv6 is having a hard time because we already have IPv4. Here we can just easily layer this on. Everybody's got six months. Now, newer editions of BIND, the old-school DNS server, like from 9.8 on, I think, support it. There's an RFC that explains it. There's a bunch of existing services that either have it already supported, or it's coming online.

And it is possible, for example, if you have a retrograde version of BIND - I'm still running 9.2, I think, for my DNS - you can explicitly specify the record number. If your DNS server doesn't know what a CAA record is, it's Type 257. So there is a way to sort of override its lack of knowledge, sort of put in, like manually create a record entry, if your server doesn't support it. But I will be updating myself quickly, or soon, because it's just - it's time to. In fact, I think my other Unix server is running a newer, non-BIND DNS. I don't remember now what it was that I chose.

But anyway, so just nice, backward-compatible, it's not going to end all the problems. But for well-meaning certificate authorities that don't know they should not issue a certificate for a domain, starting in September of this year, about six months from now, they will be forced to verify that they're not excluded from issuing a certificate based on the policy being published by that domain. So I think it's a nice step forward. Just, you know, more good things. And, boy, it took a long time to get it through the RFC process and through ratification. But that has now finally happened.

FireEye first detected a new way of executing malicious code through a Microsoft Office OLE2Link object. OLE2 is Object Linking and Embedding. It's technology that has been around for about a decade, where Microsoft was sort of doing this whole document unification technology back in the earlier days of Windows. They kept it quiet. They

notified Microsoft. And Microsoft had plenty of time to fix it, which is the only reason we got a fix in today's Patch Tuesday for this. However, prior to this morning, last week the news got out from other sources who detected this. FireEye kept it quiet. Microsoft certainly kept it quiet.

But the news got out. It was a classic phishing email attack where a Word document would be sent to somebody containing an embedded OLE2Link object. When the user opened the document, Office Word, still called winword.exe, would execute and issue an HTTP request to a remote server to retrieve a malicious .hta file. HTA is a Microsoft abbreviation for HTML Application. That appears as a fake RTF, Rich Text Format file. And so by taking advantage of - once again, here's an interpreter which is being abused. The RTF file format interpreter had a flaw that allowed a maliciously formatted RTF document to get itself execution. So the ATA application loads and executes a malicious script.

There were several different documents that were observed. It would terminate the Windows Word process, download one or more additional payloads, then download a decoy document for the user to see, while installing malware in the background. And a fully patched right up to yesterday system would bypass all existing mitigations and manage to execute bad stuff on the user's machine.

Now, there is a registry tweak which can be employed to shut this down in the short term. And we got news of that last week as soon as FireEye went public with this. But now we have a patch, that's really what everybody should do at this point. And of course it generated lots of coverage because it was an easy exploit. Word is widely deployed, and there was no existing mitigations that were able to catch this. So anyway, it's been mitigated now, and the vulnerability is gone from Windows.

And I think I heard you talking about this over the weekend, Leo, this Brickerbot?

Leo: Yeah. It's a new acronym, PDoS. I like it.

Steve: Yes, the PDoS. So Radware was the company that found this. And I edited down some of their coverage. They said: "Imagine a fast-moving bot attack designed to render the victim's hardware nonfunctional." Okay, so this isn't installing a bot in your camera or your DVR. This is deliberately, permanently breaking it, or bricking it, thus the name Brickerbot. This is a bot that bricks your exposed Internet of Things devices. And PDoS, as you noted, Leo, is the new term for Permanent Denial of Service. It's becoming increasingly popular in 2017 as more incidents involving this sort of hardware-damaging assault occur. It's also known as "phlashing," P-H-L-A-S-H-I-N-G, phlashing. Much like phishing, this is phlashing in some circles.

The PDoS is an attack that damages a system so badly that it requires replacement or, if possible, reinstallation of the hardware. It exploits security flaws or misconfigurations, like we've been talking about, and can permanently destroy the firmware or basic functions of a system.

In Radware's case, they set up a honeypot which, over a four-day period, recorded 1,895 PDoS attack attempts performed from several locations around the world. The sole purpose was to compromise the IoT device for the purpose of corrupting its storage. They've seen two different ones. There was one they called Brickerbot.1. It first appeared, but was rather short-lived. And then Brickerbot.2 appeared, which was clearly from the same authors, basically doing the same thing, but it was hiding its servers behind TOR. So all of the IPs that Brickerbot.2 appeared to be coming from was quickly

identified as Tor egress nodes, exit nodes. So it was not possible to backtrack any further because Tor was doing what Tor does, which is hiding the identity of the person making the queries.

This is a brute-force telnet attack, which of course is the same exploit vector we've talked about that was used by Mirai recently to attack CCTVs and expose DVRs and things that had a public telnet port on the Internet. It does not try to load a binary. It does have a list of brute-force attacks it goes after. The consistent thing it first tries is the admin username of "root" and the password of "vizxv," which is a well-known password for the Dahua brand DVR, DH-DVR3104H. And if you google "vizxv," you'll find a bunch of references to it.

So it is a well-known telnet administrator username and password that gets you in, or at least it did. I think, depending upon how prolific Brickerbot has been, those machines may no longer be responding to telnet or anything else. Which of course brings them to the attention of their owners, who are scratching their heads, thinking, wow, I wonder why my...

Leo: Send it back.

Steve: Yeah, my CCTV just doesn't work anymore.

Leo: Send it back. You know, this is a longstanding tradition of creating malware for good. It's still illegal.

Steve: Yup.

Leo: It's still malware. But I can't - I don't completely blame the motivations.

Steve: No, you're right. You could argue that, if they don't do this, then bad guys will...

Leo: Mirai will come along; right.

Steve: Exactly.

Leo: They're trying to get there before Mirai does.

Steve: Exactly. Yeah. So one disturbing thing is that, in that first attack where they were able to find the IP addresses, they were spread around the world. The devices doing the attacking were exposing port 22, which is the SSH port. And they found that they were running an older version of the Dropbear SSH server. As a consequence of that, using Shodan, they were able to identify these as Ubiquiti network devices. Among them were Access Points and Bridges with beam directivity.

And we talked about this particular subset of the Ubiquiti AirOS a couple weeks ago. This

does not affect our cute little Ubiquiti EdgeRouter X devices, or the EdgeRouter Lites. But it does affect, apparently it's still some vulnerabilities in this AirOS that, you know, Ubiquiti said, well, we've got patches out for all this. They did, remember, they dragged their heels. But the problem is the fact that they're just making patches available is different from the Ubiquiti owners knowing the patches are available or maybe even caring. Once all of these IoT things get deployed, they just pretty much get forgotten about until they stop working. Or in this case they've been commandeered and turned into Brickerbot scanners which are looking then for other IoT devices that they can take over.

So anyway, yet another new - oh, I forgot to mention that the last thing that's done after these things crack in is, when they're all through figuring out where they are and running their commands, they proactively attempt to shut the device down. They remove the default Internet gateway. They then issue an "rm -rf." "Rm" stands for remove in Unix parlance; "-r" means recursive. And then a /*, which means everything you can from the root outward, basically wipe the file system. It also then tweaks kernel settings to limit the maximum number of kernel threads to one in order to essentially clamp down on what the device can do. It wipes the IP tables, firewall, and NAT rules, flushing them, and then adds a rule to drop all outgoing packets.

So, I mean, it really does everything it can to destroy the device and take it offline. So maybe it'll come to the user's attention, and they'll fix it. Who knows? Maybe update their firmware or have somebody who knows something about Internet security to shut this thing down so it doesn't have, I mean, we're talking about an exposed telnet server. I mean, it's hard to say the devices that do that don't deserve a little of their own medicine. I mean, it's either, if they're not going to be fixed, then they're going to be commandeered into a botnet. And that's not good for everybody else.

Leo: Right. Yup. It's the Internet, self-healing.

Steve: That's right. Never a pretty process, but someone's got to do it.

Leo: Right, right. Just don't get caught; okay?

Steve: So on Saturday at 1:00 p.m. last October the 22nd, hackers changed the DNS registration of all 36 of a very large, well-known Brazilian bank's online properties, commandeering their desktop and mobile web domains, taking all visitors to the attackers' perfectly constructed fake spoofed site, where the bank's victim customers proceeded to dutifully hand over all of their account information, believing that they were at the bank. This is the largest, longest DNS registration attack that we've seen.

Kaspersky was the group that found it and reported it. They also believe that the hackers also were able to redirect all transactions at ATMs, that is, even automated connections, not user browser connections, at ATMs and point-of-sale systems to their own servers, thus collecting the credit card details of anyone who used their card for about five to six hours, which is how long it took for the bank to get control back of their DNS that afternoon. A Kaspersky spokesman said: "Absolutely all of the bank's online operations were under the attackers' control for five to six hours." They watched malware infecting customers. So it's not enough that they spoofed the banking site and got their, you know, "Hi, please log on," but they also gave them some malware for their trouble.

Kaspersky has not released the name of the bank that was targeted in the DNS redirection attack. But the firm says it's a major Brazilian financial company with hundreds of branches, operations in the U.S. and the Cayman Islands, 5 million customers, and more than $27 billion in assets. So not a small operation. Although Kaspersky says it doesn't know the full extent of the damage caused by the takeover, it should serve as a warning to banks everywhere - well, and, frankly, for everyone everywhere who has an important website - to consider how the insecurity of their DNS might enable a nightmarish loss of control over their core digital assets.

Now, we know that the servers where the traffic was redirected to was Google's Cloud Platform. Google was in no way complicit. They were just offering Cloud web services like anyone does these days. And so what happened was that somebody broke into, either hacked their password for their DNS registrar or broke into the registrar. It's probably the former. It's probably that they were subject to a brute-force attack, and the attackers were able to log into this Brazilian bank's registrar. When you do that, part of the registration records for a domain, you know, the registrar is sort of the ultimate root of everything. So the registry record says, what are the two name servers for this domain? That is, essentially, for example, if it's a dot com, then this is the DNS servers that the dot com domain points to for the IP addresses for that domain.

So, for example, in my case, I'm using Hover now as my domain registrar. And our listeners will remember that, when I switched - and this was the first instance where I talked about why it was so clear to me that using a time-based one-time password was better than a text record. I immediately set up strong two-factor authentication so that I know what my key is; they know what my key is. And the only way to log in, even if you did manage to brute-force my username and password, and I don't know what my password is because you know where it's stored, and you know how long it is, it's complete pseudorandom gibberish. So good luck brute-forcing it. And even if you did, then you'd still be asked what's your current six-digit, time-based, one-time password.

Clearly, this Brazilian bank didn't have that level of concern. So somebody managed to brute-force their login, almost certainly. And what's interesting is that they also had HTTPS certificates. That is, people logged into the actual domain name and got the green "go" sign, got the padlock saying this is a secure connection because it was. Because this wasn't a spoofed domain. This was the real domain. In fact, the customer shortcuts would have worked. They didn't even have to type it in. They just clicked the shortcut to their bank that they've been using for years, and it took them to their bank, except it wasn't. And unfortunately our listeners can probably guess where the HTTPS, that is, the TLS certificates came from. They were minted by Let's Encrypt six months before.

Leo: Oh.

Steve: So what probably happened...

Leo: But don't Let's Encrypt certificates expire in three months?

Steve: I don't know.

Leo: I thought they were three-month certificates.

Steve: In this case they were issued six months earlier. That much I do know. It may be something that you have control over. But in this case what happened was they probably briefly switched the domains they wanted HTTPS certificates for, immediately used Let's Encrypt to authorize the domain, and then switched it back. So there was probably - so the point is that any registrar would have made the same mistake, except that Let's Encrypt being automated allowed them to set up a whole staging and get ready. So they switched over, got the certificate, probably switched back so that nobody would realize what had happened, and then continued to get ready.

You know, the fact that they also did ATMs and point-of-sale devices, I mean, this was a big, well-orchestrated setup. And of course they didn't know how long they were going to have. As it is, they got five to six hours before the bank said, okay, we've got to get control of our DNS back. So I don't blame Let's Encrypt except that, in this instance, the fact that it was automated would have allowed a brief change in DNS to slip through the automated check. And then they could have put it back, and nobody would have been the wiser. And notice that, in this instance, the CAA would not have prevented it. They would have switched DNS to their DNS server, which could have authorized Let's Encrypt to issue the certificate.

So even if you had a CAA record, in this kind of attack, well, I mean, the problem is, if you, again, as Kaspersky says, and as we know, if you don't have your DNS pointing at you, you're really in a world of hurt. So everyone who really cares about their DNS should think about how secure their logon to their registrar is. It really, in many ways, it is the ultimately critical attack point for everything else downstream because all of your authentication just goes out the window if you don't have strong DNS. Which, again, is why DANE, the DNS-based security, looks good. But we still need DNSSEC in order to protect it.

Leo: Steve Gibson, Leo Laporte. We're talking security. And as Paul Harvey would say, "Page Three."

Steve: So that break gave me an opportunity to dig into the question you raised, which is exactly right. Because we talked about Let's Encrypt and the fact that the certificates would have a short duration.

Leo: That's right.

Steve: But since they were automated...

Leo: Big deal; right.

Steve: ...that didn't matter.

Leo: Right.

Steve: Exactly. And of course it caused a lot of controversy because people are used to multiyear certificates. So back around the issuance date, I mean, as this was all

beginning to come online, the executive director of ISRG that is the parent of Let's Encrypt said, "We're sometimes asked why we only offer certificates with 90-day lifetimes. People who ask this are usually concerned that 90 days is too short and wish we would offer certificates lasting a year or more, like some other CAs do."

Now, this next paragraph I'm wondering about. He says: "Ninety days is nothing new on the web. According to Firefox telemetry, 29% of TLS transactions use 90-day certificates."

Leo: Huh.

Steve: What? Okay. And he says: "That's more than any other lifetime."

Leo: What? Oh, you know why? Steve, Google.

Steve: Ahh.

Leo: Google does short-term certs; right?

Steve: Okay. Yup. Yup. Because they're issuing their own, so, yes. Exactly. "From our perspective," he writes, "there are two primary advantages to such short certificate lifetimes. They limit damage from key compromise and misissuance."

Leo: Since we have no revocation possibilities; right?

Steve: Yes, exactly, exactly. "Stolen keys and misissued certificates are valid for a shorter period of time. And they encourage automation, which is absolutely essential for ease of use. If we're going to move the entire web to HTTPS, we can't continue to expect system administrators to manually handle renewals. Once issuance and renewal are automated, shorter lifetimes won't be any less convenient than longer ones. For these reasons," he writes, "we do not offer certificates with lifetimes longer than 90 days. We realize that our service is young and that automation is new to many subscribers. So we chose a lifetime that allows plenty of time for manual renewal, if necessary. We recommend renewal every 60 days. Once automated renewal tools are widely deployed and working well, we may consider even shorter lifetimes."

So who knows. Maybe there was a typo in Kaspersky's note, or maybe they saw a certificate that was older on a server in the same domain.

Leo: More scary, maybe these guys, the bad guys figured out a way to kind of forge these certs and extend the date; right?

Steve: Yeah. Don't know how you could do that.

Leo: That's be hard to do, though, huh.

Steve: Yeah.

Leo: That's built into the key, I presume.

Steve: Yeah. And it wouldn't be their clock or time of day.

Leo: Oh, okay.

Steve: It would be the time at the signer's end. So I don't know how you would do that. And again, if you could, people would because it's like, hey, why not get a longer cert? I'm 100% behind shorter duration with automated renewal. I think that's just...

Leo: We've given up on the notion that you'll ever have a revocation system that works, so that's the next best thing.

Steve: Yes, yes. This week's IoT tantrum. And I know you talked about this over the weekend. A maker of smart garage door openers...

Leo: No, we didn't talk about this. I love this story.

Steve: Oh, responded to a bad Amazon review by remotely disabling the customer's purchased and paid for and operating device. The customer had left a comment, first on the support forum for this company - and it's Garadget, G-A-R-A-D-G-E-T, Garadget, so it's just kind of a cute play on "garage" and "gadget" - had left a comment on the support forum complaining about technical issues. And in order to be safe for work, I changed the noun so that I could say it on the air: "Wondering what kind of piece of" - and I changed it to crap - "I just purchased here." So this customer was...

Leo: That's it. That was it. That was the bad review.

Steve: Yes.

Leo: Yeah.

Steve: No, no, no, no. He says it on the forum. Then they followed up with a negative Amazon review saying, quote, "Junk." All caps, "DO NOT WASTE YOUR MONEY. iPhone app is a piece of junk, crashes constantly, startup company that obviously has not performed proper quality assurance tests on their products." And, frankly, that was then supported by the company's reaction, who responded online and posted their response,

saying: "Martin. The abusive language here and in your negative Amazon review, submitted minutes after experiencing a technical difficulty, only demonstrates your poor impulse control."

Leo: Oh, lord.

Steve: "I'm," writes this support person for Garadget, "I'm happy to provide the technical support to the customers on my Saturday night" - ever heard of online? Come on - "but I'm not going to tolerate any tantrums. At this time your only option is return Garadget to Amazon for refund. Your unit ID, 2f0036, will be denied server connection." So this company...

Leo: Wow.

Steve: Yeah.

Leo: It's like the Soup Nazi.

Steve: Exactly.

Leo: No garage door for you.

Steve: No, you lift the garage yourself. Oh, wow.

Leo: No soup for you. Oh, my. And by the way, I think the support person was the founder of the company. I think it's a one-man - it was a small company.

Steve: Yes, yes. And, you know, so this guy, live and learn. Garadget then defended itself in a subsequent post, saying it took action to "distance from the toxic individual." Which also suggests maybe that not everybody should be in business for themselves. And so this person then posts: "Okay, calm down everybody." Because I should mention there was this tremendous backlash throughout, both on their forum and on Amazon, with everybody taking umbrage at the fact that the company would respond to a negative review by blacklisting and denying service, denying the use of their product to this person.

So this guy says: "Okay, calm down everybody. Save your pitchforks and torches for your elected representatives. This only lacks the death threats now." He continues, "The firing of the customer," which is what he's calling it. We fired our customer. "The firing of the customer was never about the Amazon review." Uh-huh. "Just wanted to distance from the toxic individual ASAP. Admittedly not a slickest PR move on my part. Access restored, note taken." So anyway, yeah.

Leo: Wow.

Steve: Not the way you make friends and influence people in a positive fashion. You know. Anyway. Okay. Now, from our "You really can't make this stuff up" department, we have the newest entry in the ransomware division, called Rensenware. It is a new twist on ransomware. And I'm not kidding about this. This is not an April Fool's joke. This actually exists. Instead of requiring an infected user who's just had all of their machine's data encrypted to pay a sum of money, typically in bitcoin, as we know, to regain access to all of their locked files, Rensenware actually requires them to reach a high score of 200 million points on the anime bullet hell shooter known as "TH12: Undefined Fantastic Object." But it must be played on the "lunatic" difficulty level. In other words, either invite your grandson over, or beg Paul Thurrott to come unlock your machine.

Now, this was apparently a joke gone wrong. The creator of Rensenware has apologized for the software. He says: "I made it joke, just laughing with people who like Touhou Project Series," says Tvple Eraser, who then released a tool to bypass the lock on the files for anyone who may have downloaded the original version by mistake. He has also replaced the Rensenware version with a safer cut version that doesn't lock your files by forcibly encrypting them. So it apparently it was a joke gone wrong. He wasn't, you know, obviously he wasn't making any money on it. He just thought, ha ha ha, this'll be funny.

The problem is apparently it's like virtually impossible. I don't even know what an anime bullet hell shooter is, but it sounds pretty bad, especially when you have to play it on a "lunatic" difficulty level. So the good news is, if you did happen to get caught by this - hopefully it didn't exist long enough for that to happen with much probability - there is an unlocker for it.

Nicholas Deleon is a writer for Motherboard. And he wrote a really fun three-article series. He didn't know it was going to be three articles when he started, when he wrote the first one. But he knows what's going on in the industry. He's active on motherboard.vice.com site. So he was suspicious when, over the course of a couple days, he received different phishing emails from "MySafeVPN" with what would be convincing to anyone else technical details. But Nicholas smelled a rat, and a story, and he got one.

The phishing emails referred, in different cases, in one case to Plex and another case to Boxee. And he was familiar with both because he's deep in the industry. And he remembered that both of their online forums had been hacked years before, and they had lost control of all their usernames and email addresses, which would make them perfect sources for phishing bait. So somebody apparently obtained those lost databases and was scamming people who are concerned about what is now being called "America's War on Privacy" as a consequence of the decisions, the new legislation which the Congress generated and approved, and then Donald Trump signed into law, which prevented the legislation that would have gone into effect that required ISPs to proactively ask for permission to monitor their users, prevented that from happening, as we've been discussing. And thus the incentive for the topic of this week's "Proactive Privacy, Really!" podcast.

Anyway, I'm not going to go into any further detail. I did tweet the links for anybody who's interested, and they're in the show notes. It's a very well-written, fun story. He had email exchanges. He actually spoke to these people on the phone and describes the conversation in detail. He used Google's street view in order to find, apparently, a car rental, something called Fox Car Rentals somewhere, which is apparently where this

place is located. The upshot is three really fun stories and, of course, a takeaway for our listeners because we'll be talking about VPN systems and services here toward the end of this podcast.

And it's not surprising that fake VPNs or VPNs are going to be surfacing in order to target consumers' concerns because the most obvious thing one would do, if your connectivity provider - if you're worried about them snooping on you, is to wrap your traffic in a VPN so that it's encrypted as it passes through and out of your local ISP's control. So again, we'll be talking a lot about VPN services. But I did want to just note, because it won't be talking about fraudulent or spoofed VPN services, but that's a thing. And also you certainly want one with as much integrity as you can find. And I will mention later - I know, Leo, you're a fan of Sonic, and in doing a little bit of digging into this that Sonic offers a VPN, a free VPN service to all of their subscribers as just part of the service.

Leo: I didn't know that. I'll have to add that to the ad. You know, though, I mean, okay. So if you're trying to avoid snooping by your ISP, using your ISP's VPN might not be a good solution. On the other hand, Sonic is one of the few ISPs that has pledged not to snoop on you, not to intercept traffic. They fight federal law enforcement orders. I mean, you know, they're [crosstalk]. They're one of the good ones in the EFF. I'm only sad that everybody can't get it. I mean, it's kind of geographically limited.

Steve: I can't.

Leo: I know. You have to be in Northern California, pretty much. They were at one time thinking about expanding beyond this area. And maybe they are now, too.

Steve: So I was about to say that there's been some news that maybe Google Fiber will be coming to Southern California. But then, you know, Google. It's like, okay.

Leo: All the fibers [crosstalk].

Steve: You know, if ever there was a connectivity provider that is all about knowing who you are and leveraging that, when they bought, well, I mean, they're all about advertising.

And in the "just have to say it because why not," this Patch Tuesday, today, marks the first month that Windows Vista will not and will no longer receive any further updates. 2017, April 2017, is end of support life for Vista. And I put in my notes, "Does anyone care?"

Leo: Oh, lots of people care. You still use XP, Steve. Right?

Steve: Yes, but, I mean, is anyone still using Vista? Vista was...

Leo: Well, okay. Yeah, that's - yeah.

Steve: You know, as we remember, it was troubled from the start. Remember WinFS? It was supposed to have this super fancy amazing file system. Microsoft was going to change everything. And it's like, uh, okay. And Vista was really late. I mean, I guess this was in early Ballmer days because of course Steve Ballmer famously launched XP and then turned his sites on, you know, let's do the next thing. And that was going to be Windows Vista. So the Win file system got aborted. There were all sorts of fabulous features that we were promised that were all rolled back. Remember that it also had a far, far too aggressive and intrusive UAC. It was like that thing was popping up constantly and driving people crazy.

And so one of the nice things that they fixed in 7 was they toned the UAC down, still giving you basically a compromise so that it wasn't in your face nearly as much as it was in Vista. Windows 7 did end up inheriting many of Vista's innovations. But basically it was, okay, let's get away from this thing as quickly as we can. So anyway, for what it's worth, 2017 was that. Now, we will be discussing the same date in five years. 2020 is end of life - wait, no. Yes. No, three years. 2020, April 2020 is the similar date for Windows 7.

Leo: Yeah.

Steve: So three years from now they'll be no longer issuing monthly patches for Win7.

Okay. To close the loop with some of our listeners, just some fun things here. I did get a tweet from someone whose handle is ReliefTwitcher. He said: "@SGgrc Have had media.autoplay off in Firefox about a year." Remember we talked about that recently, that you could flip that off to kill autoplay. But then I had a problem that, in at least one case, I didn't mess with it too much, that, for example, I couldn't get a YouTube video to play, even when I clicked on it and pounded my fist on the browser and said, come on, play. It just wouldn't do it.

So he said: "Found the same as you with YouTube and other video sites. Easy solution: Click a tiny bit into the progress bar, then click play." And he says: "'autoplay off' is great for sites where you don't want to block all ads totally."

Leo: Oh, clever.

Steve: So nice workaround, @ReliefTwitcher. Thank you for that. And then a number of our listeners got a big kick out of last week's rant, mine, about JavaScript. Rasmus Beck said: "Watching Security Now! Episode 606. Getting the feeling @SGgrc doesn't quite like JavaScript." And Kyle said: "@SGgrc But tell us how you really feel about JavaScript." You know, when I was describing it as a heinous abomination and so forth.

Leo: Oh, that, yeah, yeah.

Steve: So I did want to just say to our listeners, don't get me wrong. I think that

JavaScript is an abomination, while at the same time understanding why it is so, and managing to write my own beautiful code in JavaScript. Password Haystacks functions entirely thanks to JavaScript being able to perform, keystroke by keystroke on the fly, brute-force search depth analysis. And by the way, nearly 4,000 uses of that page a day, 3,844 times that page is brought up every day. And remember the Off The Grid Latin Square Solver.

Leo: Right.

Steve: That I wrote in JavaScript. And because there were so many more Latin Squares possible than just 2^256, any normal pseudorandom number generator doesn't have enough entropy. So I created the Ultra High Entropy, that is, the UHEPRNG. And there's a page for that, Ultra High Entropy Pseudo-Random Number Generator, specifically to create enough possible Latin Squares that I felt like, okay, it's worthy of the Latin Square safety. Also remember I did that animation, GRC.com/animation.htm is that really cool demonstration of how data is stored on a magnetic platter and reread and reconstructed. And then nobody knows about it, but there's even a breathing pacer. GRC.com/breathe.htm runs a nice, simple little JavaScript app that just helps you breathe more slowly and deeply.

Leo: This is something I never knew about.

Steve: Because that triggers a strong parasympathetic reaction and relaxes you. I mean, that's the whole meditation and yoga and all that. It turns out there's something known as stretch receptors. And so breathing deeply and with your lower lungs, so called "belly breathing," pushing your lung out rather than your chest out, is very good for that. So anyway...

Leo: Someone needs to make a Steve Gibson Swiss Army Knife iOS app that is just your website, all the stuff in there.

Steve: Well, and in fact I did the breathing pacer like this in JavaScript because now it's multiplatform. And I specifically did it so that it would run on an iPad or on an iPhone. So my point is, yes, I have all those feelings about JavaScript. And as a language, I understand why it is the way it is. But it is an abomination. Yet it's also what we have, if we want to do client-side browser scripting, run code in the browser. You have to type the URL in, Leo. There's not even a...

Leo: There's no menu for your breather, breather?

Steve: No. GRC.com/breathe.htm. And that'll take you to it. And it'll just...

Leo: Steve's the last guy in the world, by the way, still using .htm. It's okay, Steve. You're an old-school guy. That goes back to the days when Windows only could have a three-letter document extension.

Steve: Yeah.

Leo: So what do I do? I should just follow the breathing here?

Steve: Yeah, if you can. You hold your breathe at that point, and then you do a very slow, 10-second exhale. So this gives you three breaths per minute, or a 20-second breathing cycle. And the idea is you want a faster inhale and then a hold and then a slower exhale. And over time it triggers a very strong parasympathetic reaction, if you don't pass out.

Leo: You must have amazing lungs. I cannot come anywhere close to this. You can change the parameters, though.

Steve: Yes, you're able to tweak them to suit your purpose.

Leo: That's really cool. I didn't know you did this. Wow.

Steve: Yeah. So anyway, so I know JavaScript. I've used it for many interesting projects. It has its place. But, yes, it's just - they're in the process of trying to fix it because remember that this came from Netscape back in the day because they just said, let's do some simple scripting tool for nonprogrammers. Okay? Scripting tool for nonprogrammers. You know you're destined for trouble.

Leo: That's a bad start. Yeah, that's a bad start.

Steve: If that's your definition, if that's your goal...

Leo: Oh, and then let's build it into every browser. Why not?

Steve: Oh, yeah.

Leo: I should point out that Steve so eschewed JavaScript for so long that he did his whole menuing system in CSS at GRC.com.

Steve: Yes. JavaScript-free.

Leo: Yeah. But you finally, at some point, said, well, let me do some JavaScript using the best parts, the good parts.

Steve: Yes. The things I do in JavaScript are those you cannot do on the server side.

Leo: Right.

Steve: So it makes sense for that.

Leo: So if you're really interested, kids, might be good to just check, you know, look at his source.

Steve: Yeah, just look.

Leo: Learn a thing or two, kids.

Steve: I didn't obfuscate any of that, so you certainly can look at it.

Leo: Learn how Steve does this, yeah, yeah. Nice. I like it.

Steve: So Manuel Cheta said: "SGgrc Will these tools actually help against ransomware?" And then he cites The Hacker News saying: "No More Ransom - 15 New Ransomware Decryption Tools Available for Free." And the bad news is, well, only maybe.

Leo: You know, I saw that headline, I thought, well, that's dopey.

Steve: Yes. There were some well publicized mistakes.

Leo: Poorly written ransomware.

Steve: Exactly. In a few instances where the encryption was done wrong, the key was left behind, or the key was static, and once it was figured out it applied to all of them, you know. The point was ransomware not done right, which is what we could all wish for more of; but unfortunately, in general, no. CryptoLocker hits you, it's game over. It's get out your bitcoin wallet or go find some bitcoins and so forth.

Leo: The unfortunate thing about articles like this, I got a call on the radio show from a guy saying, oh, I don't have to worry about ransomware. No, you really do. This doesn't work for most ransomware.

Steve: Yes, yes. This is a little bit like, okay, I'll buy SpinRite once my drive crashes. Well, okay. But you could also prevent it from crashing.

Leo: Did I tell you, I was having dinner with a guy who is on the board of a - I don't

want to say the name, but it's a federal banking thing.

Steve: Okay.

Leo: And so he's at the board meeting, and they tell him, yeah, we're stockpiling bitcoins in case we get bit by ransomware.

Steve: [Choking]

Leo: I know; right? I know. Well, I'm glad they're doing that. That's good. They're really prepared.

Steve: Yeah.

Leo: Geez.

Steve: Well, although I have to say, it also, I mean, although that's fatalistic, and it's not proactively secure...

Leo: Well, I hope they're doing other things, too; right.

Steve: Yes, yes. But as we've said, if your organization is not completely in your control, if you're Sony Pictures, you know, and there's a zero-day flaw, and any one of your employees in the organization clicking on the link in email is all it takes, well, yeah, having some bitcoins handy may not be a bad idea.

Leo: As Plan Z.

Steve: Yeah.

Leo: And, by the way, don't count on it working.

Steve: No, exactly. Although - yes. You can't count on it. And so...

Leo: There are some honorable ransomware creators out there.

Steve: So my argument about hard drive failure is, well, better to prevent your drive from dying. And in the ransomware case, better to have a current backup so that you're able to fall back to a snapshot that is useful for you.

And finally, Gengar said: "@SGgrc If I fail school because of this game, can I come work for you?"

Leo: Uh-oh.

Steve: Then he sent me a screenshot. He's on Level 4000. That's the next page of the show notes, showing Level 4000. And I wanted to take the opportunity to remind our listeners: Squareit.io. It is really charming. It is getting half a million downloads in one month. As you can see from Level 4000, it doesn't end after Level 17 like some of the annoying things we've, I mean, fun, but unfortunately not deep enough puzzles we've talked about before. Both iTunes and Android. And it's simple, but it's just right.

Now, I mean, I see a lot of these things. Our listeners know I love puzzles. So I'm actively vetting, like, more of stuff than you can believe. And few things make it through. You know, Blockwick was a win. We've talked about various ones. This one is charming. It's free. Squareit.io. And so I'll just recommend it again. Once you get the hang of it, this guy probably - he's obviously become an expert. But it's not super difficult, but it's, again, there's an art to making these things that aren't just sort of brute-force puzzles. They're elegant. And Squareit.io is really elegant. And in fact I think that's also the website; right? Yeah, Squareit.io is the website. And you can find links there to the app on the various mobile platforms.

Michael Surette asked: "What's up with ShieldsUP? My open ports 22 and" - wait, 22? Okay, well, SSH, I hope you're secure. And 25, he's got an email server for some reason. "My open ports 22 and 25 show as stealth unless I do a comma-separated custom probe." So he's wondering why, when he does like an all ports probe, they're coming up stealth, but he knows they're open. And if he only asks ShieldsUP! for those ports, it shows them correctly. This happens so often, I wanted to explain. It's his router. Increasingly, routers have sort of lame, but it's like, eh, okay, scanning protection. So if the router sees a bunch of incoming TCP SYN packets scanning for ports from a fixed IP, from a given IP, it will put up a time-based port block on that IP. And since that's not GRC, that's shieldsup.grc.com, it doesn't affect your connectivity to GRC. I scan from a dedicated IP that identifies itself as what it is in its reverse DNS.

But the point is, if you believe you have open ports, and you do the full port scan, as soon as the scan gets going, the router says whoa, and just puts up a complete shutdown on the ShieldsUP! IP so that all of your ports appear stealth, although it's actually your scanner working on your behalf. That's always a feature you can turn off. And so if you turn that off in your router's configuration, then do a full port probe, you'll see them. Or you can selectively probe a few, which won't trigger the router's protection. And that's why the full port scan shows stealth, whereas a selected port shows the truth because a few ports isn't enough to upset the router and cause it to preemptively block anything else incoming.

And then our last bit of feedback from our listeners. Tom Corwine brought up a great point in two tweets. He said: "Re all this talk about ISPs forcing us to install their own CA cert. How would our IoT devices use secure connections?" And then his second tweet was a follow-up saying, "Come to think of it, those IoT devices don't even need to use a public CA. Their manufacturers can mint their own certs."

And I replied, I said: "Tom, that's exactly correct. We need a public CA system only so that clients and servers don't have to have any foreknowledge of each other. They both

rely upon a mutually known third party. But the IoT model is different. The devices can embed a cert for only the manufacturer's servers." And I'll note that, if they were to do that, they would probably then also not be connecting over port 443. They'd probably just choose like 4343 or 4433 or whatever they want to. The point being that that would eschew the ISP's filter that is only going to be filtering 80 and 443, and allow them to connect and encrypt their connection using a certificate that they have from their own server. And in that case, it can be an extremely long-lived certificate.

It's only the CA guidelines that restrict to two or three years for certs, as we were talking about before. I, for example, I have a certificate, www.steve, that I use here just so that I'm able to do my own testing of security stuff. And I have my own DNS that points that to the www.steve machine. That's obviously completely invalid. There is no Steve TLD. But I made a cert that was good, I think it was for 50 years. So I just don't have to ever bother with it again. It's completely useless for any other purpose, and I have been using it for years.

So the point is you can certainly make a certificate with a super long lifetime. And were I in the IoT business, I think that, I mean, I don't even see a downside to it. I would embed that certificate and then protect it because you don't have the advantage of expiration. On the other hand, an IoT device isn't going to be receiving updates on a dynamic basis, so it wants to have something very long-lived.

So, Tom, thanks for giving me an opportunity to note that. I'll bet that's - it'll be interesting to see if IoT devices are using security to the degree they are, and whether they are connecting to remote servers over 443, standard HTTPS, or are they making just some other non-HTTPS port? If that's the case, then we're back to ISPs being able to tell us we need to put their certificate in our devices if we're going to use their service because IoT won't cause that to break down.

And we have some errata. Fun errata. Our listeners will remember that my head exploded last week.

Leo: It's come together nicely, though, afterwards, so that's good.

Steve: Yes, thank you. I read that news that the Random.org site was being incorporated into libsodium, the crypto library I love that SQRL uses, that more and more people are using because everything except this was being done right. And the news was that they were going - it would be the default soon for their cryptographically secure pseudorandom number generator. And I just put my head in my hands; and I said, oh, my lord. And our listeners will remember that I was, like, trying to think, okay, what possible, possible utility could this have because the only way I could possibly see it of use is if you were on some sort of platform where there was just absolutely no available source of local entropy that some sort of isolated stranded device could come up with, then, okay, kinda maybe. I mean, just you could use packet-timing. Use anything. But don't go to Random.org, which is an otherwise nice site, but don't build it into your library.

Um, I didn't look at the date of that news. And what was 10 days ago?

Leo: Yeah? April Fool's Day, maybe?

Steve: Uh-huh.

Leo: Whoops. Goddamn April Fool's Day. Goddamn it to hell.

Steve: So it got me. Yup.

Leo: Son of a bitch.

Steve: Thank god. I mean, good. I have no problem being wrong on this one. Please, thank god.

Leo: Oh, I should have caught that. I apologize. I should have caught that for you.

Steve: I mean, you know, I mean, I was, like, stretching. It's like, okay, really?

Leo: It makes no sense, it makes no sense.

Steve: Well, how could you - how, what, how, what?

Leo: It's a pretty subtle April Fool's joke, to be honest. It's a very subtle April Fool's joke. Maybe a little too subtle.

Steve: Yeah. So I was awake till 2:25 a.m. yesterday morning.

Leo: Uh-oh. Why?

Steve: Finishing the 15th book in the series.

Leo: Honor Harrington?

Steve: No. That was another one, one with a marathon.

Leo: Yeah.

Steve: This I can recommend without reservation.

Leo: But you didn't read all 15 in the last week, did you?

Steve: No, no. I found out about it from my sister's husband, my brother-in-law, at Christmas. I was telling him about the Tanis Richards series by M.D. Cooper. I said it's really fun military sci-fi, four books. And he said, okay. He said, "All I read is military sci-fi." He said, "You've got to read the Frontier Saga." And so I came home, I finished whatever I was reading at the time, and I thought, okay. Leo and listeners, oh, my goodness. First of all, yes, I read 15 books in the old-school style of visually scanning printed words on a page.

Leo: Good lord.

Steve: I know.

Leo: Are you insane, man?

Steve: I write in assembly language, and I read the way my grandfather did.

Leo: Read dead trees.

Steve: It is a fantastic series. I would say I like it better than any of the other two, Honor Harrington or Tanis. They're both - the Tanis Richards M.D. Cooper author, the Intrepid Saga.

Leo: But wait, Steve. You've only read Part 1.

Steve: I know. It is 15 books in a series, with a planned five-series set. So 75 books in total.

Leo: Who is this Ryk Brown? And what drugs is he on?

Steve: It's R-Y-K Brown. If you are a Kindle Unlimited subscriber, as I am, they're all free. That is to say, they're all in that program.

Leo: So you didn't do dead trees. You did eInk.

Steve: Oh, oh, yeah, yeah, yeah. I don't do dead trees anymore. I do eInk. And I love my Kindle. Anyway, I just - everything about this, I don't want to go on for too long, but everything about this is good. And this is not a spoiler because everyone knows I don't do spoilers. You learn instantly, probably from the back cover of the book, that this is in the future, and that something known as the Bio-Digital Plague has ravaged the Earth and our colonies and knocked us back to the Stone Age. We're in the process of recovering, having lost everything, and we discover a data ark hidden in the Swiss Alps, which is the sum of all human knowledge prior to this plague, which helps to bootstrap us back. Unfortunately, there's a spinoff of humanity, our bad guys, who have decided to

survive by taking what they need from others.

So at the beginning of the book, a bad accident puts a very green crew in charge of one of Earth's very few FTL-capable ships. And this first 15 books is just - the Honor Harrington stuff, there are fabulous good parts, but like five in the 20 books of the Honor Harrington series. And Honor Harrington is being made into a series of movies. I so wish that the Frontier Saga was doing it in its place. This is so much better.

Anyway, for people who like military sci-fi, the people are not all super-enhanced. It's different than some of the other sci-fi where you've got implants and telepathy and all kinds of other crazy stuff. These are just regular old people in the future. But again, the way the plots are woven together, there's nothing stretched-out feeling. Lots of action. Great characterization. You care about the people. Anyway, enough said. TheFrontierSaga.com. The author is Ryk, R-Y-K, Brown. And if you are a Kindle Unlimited subscriber, you can read them all as part of your subscription.

And, finally, this week's Pithy Slogan: "When in doubt, encrypt. When not in doubt, be in doubt." So I thought that was good. When in doubt, encrypt. When not in doubt, you should be in doubt.

And I got a classic nice note from a customer, Brett Parks, whose subject was "SPINRITE [in all caps] saves the day yet again." He said: "I've been using SpinRite for, what, 20 years now or more. When all else fails, it will =at least=" - he brackets it in equal signs, sort of chevrons - "it will at least bring a hard disk back to where it will at least be bootable and readable. And it just did that again with the 1TB hard drive on my primary machine, which went belly up and took my accounting, taxes, email, client lists, yada yada yada, along with it. It took a couple of runs, but SpinRite got it back to where I could," he says, "I could get EVERYTHING [all caps] off it. Not bad for old school, eh? Thanks yet again. Brett Parks, Lexington, South Carolina." And Brett, thanks for sharing your experience.

Oh, and I did want to note, remember last week I said that I was going to come up with some way to provide some sort of benefit to the listeners of this podcast who have been supporting me by purchasing SpinRite, and in many cases anxiously waiting for 6.1. And I said I'm going to figure out some way to thank people for that. And I know what I'll do. I am, as a consequence of having spent so much time with SQRL - and by the way, I'm just having a ball working on SQRL. I had to tear myself away from it yesterday in order to pull this podcast together, but making lots of very fast progress. I'm working now on the self-uninstaller. I don't want to have a separate uninstaller because, well, I maybe don't need one. But the trick is how to get an executable file which is running to delete itself. It turns out that's not easy. I've got it working so that you would just be able to say, "Go away," and it will do that for you.

But anyway, what I'm going to do is, to compensate for how long it's taking me to get going, I'm going to get 6.1 out sooner. But I know that I will be at a point that I will consider it safe, but at prerelease. And at that point it will be available to our listeners exclusively. So I'll work out the details when we get to it. But I just know that, I mean, essentially, the work I was doing on 6.1 before, I had a whole group of people who were testing it as we were going along. That's how, for example, that I know that it is able to run at half a terabyte per hour. It's already done that.

So I'm going to - my priority is going to immediately get a 6.1 out the door, even before it's over on the Mac platform. Just get it running on the existing PC platform because I can do that sooner. But then I will shortly do a 6.2 which adds native Mac support. And with .3 and .4 and so forth, I'll add native USB and so forth rolling forward. But that's

what I'm going to do. I will - it's always the case that something is workable and usable, even if it doesn't have all of the finish and polish on it. I mean, for example, we've been using SQRL for a year, but not in a way that I felt it was ready to get the full seal of approval.

So SpinRite 6.1 won't take nearly that long. I mean, as I said, when I suspended it to go to SQRL, it was coming along. I was much more ambitious about what I was going to do. I'm going to dial that back now in order to get something out fast so that people who have 6 will be able to get this tremendous benefit of its compatibility, freedom from the BIOS, able to operate at literally the absolute maximum speed the drive is able to transfer at. That is, that's where I am because this thing allocates a 32MB buffer, whereas drives are typically limited to a 64K buffer. So this thing is a vastly larger buffer that allows streaming transfers that can also be overlapped. So anyway, no stone unturned in the performance of this thing.

And Leo, we're going to discuss Proactive Privacy and do the Proactive Privacy Roundup.

Leo: We made it.

Steve: Yes. We have time.

Leo: We have time. We have a whole half hour. I hope that's enough.

Steve: Yup.

Leo: All right, Steve. We've been waiting two weeks for this.

Steve: Okay. So our ideal, the ideal situation would be for a user to have an entirely separate one-to-one relationship with every entity that they transact with on the Internet, so that there is no cross-entity leakage, so that where they go only knows about them. And there's no concept of tracking. The word was never minted because it was never possible. We know that's not the world we live in. So what are, first of all, what are the tracking hooks? What are the things, the technologies that start us being able to be tracked?

Well, the first one is the great-granddaddy of them all, which is our IP address. As we know, in order for Internet traffic to get back to us, when it's outbound it must carry essentially a return address so that the server that we're connecting to or the service that we're reaching or email, you know, whatever it is needs to have the IP, the public IP address for us.

So the first and most obvious thing is our IP address. Now, it's true that those are not typically static over the long term. Back in the old days, where you were dialing in with a modem into a modem pool for CompuServe or AOL or EarthLink or whatever, you got an IP address literally assigned for that connection. It was only valid for that connection. When you disconnect and reconnect later in the day to check your email, you're going to have a different IP address.

Now, IP addresses technically, as we know, are "leased," is the term, through DHCP. And

there's, in part of the DHCP protocol, the lessee says, "I need to renew my lease. Here's the IP I currently have." And so there's actually in the protocol is an attempt to remain static, that is, to keep the same IP. So, for example, in my case, where I've got a cable modem, and I have a router which is independent, and it's never powered off unless something sort of catastrophic happens, I mean, typically almost never. The last time it was, was when we had that weird connectivity problem, and I was trying everything to see, to verify it wasn't my problem, it was something upstream at the cable provider. Well, I did, I power cycled it and so forth. But even there, when I was completely off the 'Net for half an hour and got back on, I was able to recapture my same IP. DSL connections, which are up statically, will tend to have a static IP. So in general, your IP won't change for a connectivity session. It's not guaranteed not so, but it generally doesn't.

Now, it is the case that, where a household is behind a single NAT router, anyone who's interested in who you are, if they only have your IP, then that reduces them to household-level granularity. That is, the IP address wouldn't say whom within the house, if that's all they had. But that's certainly better than nothing. And as we'll see by the end of this, all of the evidence that we know of demonstrates that there is a huge industry that surrounds tracking. There is a lot of money involved. There's huge incentives. And while there are smart people who are working on blocking tracking, the people who are working who are being the developers, the programmers, the coders, whose paycheck is dependent upon them figuring out clever new ways to track us, that is, not to get shaken, they're doing everything that they can, and they're just as good.

So the point is the only rational position to take is, if you care about this whole issue of tracking, you must assume that everything that we can think of, we who want to resist being tracked, everything we can think of, the other people, the trackers have thought of, too. And so, for example, I'll make the case that IP address, low granularity as it is, transient as it is, it is still a powerful signal. And, for example, if you suddenly became unrecognizable, that is, you changed OSes, you changed browsers, you used a freshly wiped and scrubbed system, you were in incognito mode, but you didn't change your IP, you're not fooling anyone because even though, you know, somebody looking would see, would know the IP you had five minutes ago and would lock onto you with at least knowing you are the same IP, even though you are otherwise completely unknown. And all of your attempt not to be connected would be lost because you hadn't also changed, arranged somehow to change your IP at the same time you had done everything else.

Okay. So that's IP. Cookies, of course, is the second oldest and most mixed-blessing tracking system. Cookies, as we know, were originally conceived by Netscape as a means for maintaining our session state. And they're still used today for that. Session state acknowledges the fact that our browsers do not have, typically do not have a persistent connection, that is, it's not like an old-school Lear Siegler or Hazeltine terminal, wired like with a modem to a remote mainframe, like back in the CompuServe days, where you're typing essentially on a console of the mainframe which has been remoted to you. Instead, a browser makes a query, it receives a response, and it disconnects. And then the user looks at some things, clicks a link, that makes another query. And it receives a response, and it disconnects.

So the question is how are those two separate events, those two queries, associated with being the same person? Of course first-party cookies is the way we do that. When the server responded to the first query, it gave the browser a cookie tagged with its domain and various other requirements. For example, if this is a session cookie, we hope that it has a secure tag on it so that, if ever an HTTP query was made to that domain, that cookie would not be transmitted. Instead, it would only be sent back by the browser with subsequent queries over a secure connection. And there's expirations and persistent

versus session and other sort of variations. But that's the concept.

I would argue that a lack of foresight, probably - well, maybe not, maybe just the wrong decision, or just the way things evolved, but what we ended up with was no explicit consideration for sourcing content from third-party sites. That is, a website originally hosted all of its own content because there just wasn't anybody else you could go to to say, here, have some images for me. And they go, what? Yeah, we don't do that. So everybody was hosting all their own stuff in the beginning.

Then, especially with the advent of advertising, where a site could generate revenue by hosting somebody else's commercial content on their own page, suddenly now the web browser is going to a third party. Well, third parties, unless explicitly prevented, because your web browser is asking for content from that site, that site, that is, an advertising domain, a web beacon domain, a Facebook Like button, a Google Analytics bit of script, whatever that content is, it's making a query to that other domain because it's not the primary, the first-party domain. That's what makes it the third party. But within the constraint of that domain, cookies are allowed unless disabled.

Now, what's interesting is, I mean, and I've been aware of this concern about third-party cookies forever, GRC has another non-advertised, not linked to GRC's main menu, deep bunch of technology known as Cookie Forensics. At GRC.com/cookies is a set of pages which have been in place for, I don't know, like a decade or so. So, for example, as a consequence of that, I know, because I looked this morning, that of GRC's 43,576 unique visitors last week, and this is updated at midnight on Sunday night every week, a snapshot is made and updated from the stats that were accumulated. So 43,576 unique visitors last week. 79.21% of GRC's visitors, so just shy of 80%, had third-party cookies enabled.

But among those visitors who were using Safari, only 18.83% of Safari visitors had third-party cookies enabled. Why? The tyranny of the default. Apple's Safari browsers are alone in the industry in disabling third-party cookies by default. And in fact five years ago Google agreed to pay a $2.5 million civil penalty to the FTC after they were caught deliberately using some JavaScript trickery to leverage a small flaw in Safari's iFrame handling at the time in order specifically to place tracking cookies into Safari browsers that were configured not to allow it.

And even today there is discussion among developers who are being thwarted by Safari. These are the people I talked about who are on the payroll of advertisers and tracking companies, annoyed that Safari is causing them so much trouble. I say bravo to Apple. And I wish, I mean, and we've talked about first- and third-party cookies a lot on the podcast because it's always been the primary tracking technology built into browsers.

Anyway, GRC.com/cookies for anyone who is interested. And I even have a cool page that shows a graph of the percentage of which types of cookies are enabled and disabled by browser vendor. And back at the time, many browsers had cookie-handling flaws which that page was demonstrating, that is, the Cookie Forensics. You're able to run, I should mention also, there is a forensics page. You can perform a test yourself on your own browser.

And, for example, I had third-party cookies enabled in Firefox. I'd forgotten about that. Because remember when they removed it from the UI, where it used to just be a checkbox you could turn off, and they buried it under, like, almost hard to find. You have to go to Tools, Options, then Privacy. And then you look at that page, and it's not apparent. And then under the History dropdown you have to change it to "Use custom settings for history." Only then does it reveal a bunch of settings, and among them are

"accept cookies from sites." And then, under that, "accept third-party cookies," you can set it to "never."

So before I did that, and I used GRC's cookie forensics, it showed me that in the second half of that page was all red showing that, like yours shows, Leo, that you are accepting third-party cookies on your browser currently. If you turn that off and rerun the test, it'll show you that they are being blocked, and you don't have a problem. And again, cookies are not the only way people track us, but it is certainly a way for them to grab hold and lock onto us.

So even today, problems remain. I found an interesting link I won't go into in too much detail because it's JavaScript trickery. But the NCC group that we've talked about in the past, a great security company, they have a post about setting cookies for third-party websites in different browsers. And their TL;DR reads: "This post discusses the results from our research into the ability of third-party websites to set cookies for third-party websites across different web browsers. The ability to set cookies in this manner not only facilitates tracking, but also opens up other opportunities and avenues of attack."

And the short version is they have a really nice chart where not one single browser was immune to all variations of the attack - not Chrome, not Safari, not IE11, not Firefox. I didn't see Edge. I don't think I remember seeing Edge there. But the point is that even today there are still problems with the way cookies are handled because, like so many things, it wasn't designed to be bulletproof from the start.

And here's the problem. After all, I mean, first of all, turning off third-party cookies is simple. I would argue that since Apple and Safari do it, and a huge chunk of the web is not broken for them, and the tracking people are pulling their hair out with frustration, the first thing anyone should do, it's trivial, turn off third-party cookies. Thank you very much. And then verify that they're off. One of the problems we used to see is that - and the forensics page, the Cookie Forensics page at GRC, you're also able to just, by the way, to say Google, GRC Cookie Forensics, and it'll take you to that page because Google has found it long ago. You can, if you turn off cookies, some of the browsers would not renew their cookies, but they would still issue stale cookies. And the forensics page shows you the age...

Leo: Eww.

Steve: ...of the cookies.

Leo: Who wants stale cookies?

Steve: Nobody wants to have a stale cookie. So, yes, you could get stale cookies. Now, with all that said, we still have a problem, Houston, because HTTP redirection chains, which are becoming much more prevalent, completely thwart third-party blocking by allowing first-party queries. You may have noticed, for example, that the links on Google no longer take you, as they once did, actually to the thing Google has indexed. Those are all Google links. And so you're clicking on a Google link whose tail contains the URL of the site you're going to go to, which is Google's way of acquiring information about you because, I mean, they already got information about you when you brought the Google page up. But now they know what you clicked on.

And you may have noticed that sometimes you click on a link, and your URL flutters a few times. It goes [vocalizing], and then you kind of get somewhere, it's like, what the hell just happened? Well, what just happened was you went through a chain of maybe five or six other sites that have chained themselves together only for the purpose of tracking you. What happened was you went to a site not where you thought you were going, a third-party site. But because your browser went there, it's briefly first-party. It got a cookie from you. And then it forwarded you to the next site in the chain, which got its cookie from you, or gave you its cookie, and then sent you to the next site in the chain, and so forth.

So this is the argument that some people have made about why bother blocking third-party cookies because HTTP redirection and JavaScript tricks still allow sites you are not directly visiting to get first-party access to you. And this is one of the ways that happens. I counter by saying, if third-party cookies are still today making tracking developers pull their hair out, then that's every bit of reason to disable them because obviously everything still works. And so just don't make it easier for the bad guys.

And finally, plugins have historically been another source of tracking problem. Plugins, I mean, we've talked about flash cookies for a long time. There were instances where you could wipe out all of your browser cookies, but if you still had Flash, the site could use a Flash cookie in order to create that persistent link to you in order to repopulate your first-party cookies. And there were even services that actually sold that facility. They said, "We have technology that prevents us from losing track of people. Even if they delete all of their browser cookies, we're still locked onto them." So browser plugins was one other way that there was a problem.

Okay. Next in line is things your browser sends in its metadata. That is, other stuff than just the query or the cookie. And I had to look at the date on this because 16 years ago, on March 17, 2001, the people in one of GRC's newsgroups stumbled on the fact that the EarthLink custom browser was apparently embedding a unique token into every query. I created a page about it. It generated a huge amount of brouhaha on the Internet. And EarthLink quickly produced a statement and demonstrated that what they were doing was generating - they had created this weird-looking serial number thing which was static for individual users. And when we compared notes, those who had an EarthLink browser, everybody's seemed to be different. So it looked exactly like something meant to fingerprint us.

And it turned out that we got a breakdown of how it was composed, and it was a whole bunch of characteristics about the machine it was running on, things like screen width and screen height and so forth. Still, it's the kind of identification information that people didn't like having their browser transmit. And EarthLink removed it shortly thereafter because I think it was the kind of thing that was there, and they weren't even using it. And after everybody screamed about it, they said, okay, fine, bad idea. The point is, none of this is new. Sixteen years ago GRC had a page, GRC.com/su - that's for ShieldsUP!. So GRC.com/su/earthlink.htm. And that thing is 16 years ago, where we were looking at the privacy consequences of our browsers embedding stuff that we're not happy with.

Of course, we've talked about something similar to that with Panopticlick. Panopticlick is the EFF's project. And I ran it just before, actually while putting the show notes together on an otherwise rather locked-down Firefox browser, mine. And so the test results came back, is your browser blocking tracking ads? Yes. Is your browser blocking invisible trackers? Yes. Does your browser unblock third-parties that promise to honor Do Not Track? No. Does your browser - which is, you know, good. Does your browser protect from fingerprinting? No.

And, boy, I then click on "show full results for fingerprinting." This thing has matured more. My browser, unfortunately, has a unique fingerprint. And if you look at the details of what the fingerprinting returns, now it's gone from a few things to an encyclopedia that apparently is unique about my machine. So that's a thing, too. That is, metadata that our browser can - I'm sure they're running scripting. So metadata that the browser or JavaScript is able to use in order, again, to come up with something unique about us.

And lastly, DNS queries. Remember, as we've mentioned, that even an ISP that cannot see the content of your HTTPS connections, they do know the IPs to which your traffic is bound. And unless you do something to hide your DNS from them, they also know every domain you and your browser look up. And believe me, as we know, that's no longer just where you go, it's everything that your browser sucks in from the web. The browser needs the IP of all of those domains. And so those are outbound DNS queries. And even if you don't use the ISP's server, that is, if you're using OpenDNS, for example, doesn't matter. DNS is not by default encrypted. Even if your other traffic is, DNS is UDP with no encryption.

So one thing to consider is using DNSCrypt, D-N-S-C-R-Y-P-T dot org. It's becoming increasingly mature. It's open source, both clients and servers. There are clients for Windows, macOS, Linux, Android, iOS, and even routers. So you could put DNSCrypt in a router which is able to host it. And your systems all just use your router's DNS. It encrypts your queries out to a public DNSCrypt resolver. And there are many now public DNSCrypt resolvers around the world. And that prevents anybody other than the DNS server from knowing who you're asking the IP addresses for.

So those are the various technologies, both intended and unintended, and metadata for compromising us using the technology. Of course the other point, and this is a point, Leo, you have correctly made frequently, is what about the major web players? We've talked about Google and Google Analytics. Google knows who we are, if we have an account at Gmail or Groups or an account with any Google property. Our system contains their cookie, and we want the convenience. I don't even think you could use Google without at least having a first-party session cookie. And if you only did that, you'd have to log in again every time you reopened your browser. So it makes more sense to use persistent cookies to cut down on that. At least Google has gone HTTPS.

But Facebook also knows who you are, if you are a Facebook user. And everywhere over the 'Net they plant their little Like buttons. Those Like buttons are browser queries back to Facebook that are cookie-enabled. And unless you have blocked them, they're a beacon telling Facebook not only who you are, but where you are right now and what you're doing. And the fact is any major player on the web, a Google, a Facebook, a Microsoft, a whomever, Bing and so forth. They are giving your browser cookies. And anytime your browser ever fetches anything from them, no matter where you go, they know who you are, and they know where you are.

And so this is one of the reasons to help people who are concerned about their ISPs, like what this legislation means. It's why it's like, okay, well, calm down because arguably, as we're moving more towards HTTPS and TLS connections, in the same way that we're going dark to the NSA, who can no longer see our traffic, we're also going dark to our own ISP, who cannot see our traffic. They can see our DNS. So a simple thing to do would be to disable third-party cookies because we know the tracker people are still being driven crazy by that, and configure yourself to use DNSCrypt. And then most of the majority of your activity and the DNS you look up is being blinded. But remember, the ISP still sees the IPs you go to. So for that, we need to talk about a VPN.

It's worth also mentioning first the option of using incognito browsing modes. That's not

a complete solution, but it's part of an us going dark toolkit, that is, the idea of wiping our history. We don't want to have visited links show up because we know there have been hacks that show they're able to probe the visited link cache in our browser by noting what color they are because browsers display visited links in different colors, and that's something that somebody can remotely obtain. And we also want to not transmit any cookies from our non-incognito mode. So it's sort of a way of just looking like somebody unrecognizable. But if you've got the same browser plugins, the browser plugin could be deanonymizing you. And if you're using the same IP, we know that that can transiently allow someone to make the bridge from incognito to non-incognito and back and forth.

Okay. So a VPN. The good news is VPN crypto is rock-solid and bulletproof these days. I would argue that OpenVPN is the leading contender as a protocol. It is both an implementation, but it is also a protocol. There is a very nice VPN known as SoftEther.org which offers a bunch of clients and servers, and it offers OpenVPN as a protocol. The advantage of OpenVPN is it's rock solid. Many VPN providers, like proXPN, for example, offer VPN as a solution, that is, as their primary protocol. So the advantage of that is there are OpenVPN clients everywhere. It is rock solid. There are point-and-click installers that are easy to use. What it does is, as we know, it encapsulates all of your traffic out to the VPN server.

Now, it's worth making sure, though, that your instance of a VPN is encapsulating DNS because there are some that don't. There are some that are protocol specific; and, for example, they'll only encapsulate your HTTP and HTTPS traffic, maybe your email. That is, they're selective, rather than encapsulating everything. So if you thought you had full encapsulation, but your DNS was not being encapsulated, you'd want to know that. You can use DNS Spoofability Test to find out. If you run the DNS Spoofability Test at GRC, among many other things, it shows you which DNS servers are resolving your current DNS queries. When you bring up a VPN, you hope that changes. If the same DNS servers are resolving your DNS queries with your VPN up, and you use the DNS Spoofability Test, then that means your VPN is not a full-isolation VPN. And then the question is what other than DNS, if it's not encrypting and concealing your DNS queries, what else is it not encrypting and concealing?

Now, the bad news about VPNs is they don't proactively do anything more than that. They are super useful for protecting your local traffic from local snooping. Your ISP, the hotel you're in, the caf you're in, over open WiFi, the airport you're in, et cetera. So that there are certainly use cases for it. And it also changes your IP, which is crucial if you want to immediately go dark, along with doing a lot of other things. But VPNs don't otherwise proactively do more than that.

And the other problem is that there is the concern of the emerging traffic on the Internet being at well-known VPN server, so-called, you know, you could call them exit nodes. Not quite the same as Tor because, of course, Tor is largely used by people who are trying to evade, I mean, like proactively putting up with much slower Internet use in order to be anonymous, to have their IP be anonymous. So Tor exit nodes tend to attract much more negative or law enforcement-style intelligence attention than just random VPN servers on the 'Net. But there is still some concern about the inherent traffic concentration that VPN exit nodes create.

And I have in the show notes a bunch of notes about choosing a VPN provider. There are a number of VPN ranking sites. I don't have any preferred one. Of course, you know, proXPN has been a long-time sponsor of the TWiT Network, and we know that they don't log. But there are many sites that rank and rate and have, like, large spreadsheets. I would imagine there's probably even a Wikipedia page that has all kinds of characteristics

of VPNs. So many are good. You do want to be careful that you've got one with a reputation. I forgot that one of the things that Nicholas did at Motherboard when he was looking at MySafeVPN was he did a WHOIS query, and he noted that their registration for MySafeVPN was dated March 30th. So they have not been around a long time.

Leo: My brand new SafeVPN.

Steve: Exactly. So checking the reputation of any of these places is something that you would want to do. And of course seeing how long the WHOIS record has been there is one way to do it. I remember noting that mine was only six months younger than Microsoft.com. So, you know, we both got into this game in the early days. So as for Tor, eh, I was hoping that the Tor browser would make a more proactive effort at protecting privacy. It has a few mild tweaks to it. It has NoScript installed. It has HTTPS installed. And some default settings have changed. But it's not like they built in a really good query filter that absolutely strips the browser headers to the bare minimum.

That's what I would do. I would have, you know, a bogus User-Agent and an Accepts header for what content you can accept with no extraneous information in it. I mean, I would really strip it to the bone. They don't. So I don't think Tor buys you that much, if what you're really concerned about is just, like, depending upon how much time you intend to be in private mode. And it's not easy to stay there because you're going to have problems with services being as sticky as you want them to be.

One last thing I'll note is that in the last couple weeks I saw an Indiegogo project called Filter, which was yet another bogus filter router project where they were claiming to do lots of inspection of your traffic in order to protect you from malware and other problems. Well, we know that just as the ISP cannot see into encrypted traffic, neither can your router. So unfortunately they are dramatically overselling what they're able to do.

So with all of that foundation, the last page of the notes here is strategies and tactics for maximizing your online privacy. As I said at the top, due to the tremendous pressure to track and profile, and the massive amounts of money behind that, we must assume that everything that can be done to track and profile is being done. And I will remind us of this week's slogan: When not in doubt, be in doubt.

Leo: I love that slogan.

Steve: Isn't that great? Yes. You must change your public IP because, that is, so think about, like, you're about to embark on something where you need to go dark. You need to emerge on the Internet anonymously. So you must change your public IP because, again, if it can be done, we must assume it is being done. And if any of these people on the Internet have been seeing a lot of traffic from this IP, and suddenly a browser query comes out with no headers and no cookies and looks completely anonymous, but from the same IP that something was just coming from two minutes ago, okay, suddenly that new traffic that just blew its anonymity has been associated with the immediate previous traffic.

So changing your IP has to be part of it. Go to another location. Shut down your cable modem or DSL long enough to obtain a new IP. Maybe if your router has a release-and-renew DHCP, you might be able to force a change of IP. And of course now you can just

put "my IP" into Google, and they'll tell you what your public IP is. Or use a VPN. So go to a different location or pretty much use a VPN. That will immediately change your public presence to the IP of the VPN server or wherever you've gone. You must use incognito mode, that is, switch to that mode where there is no browsing history to be brought forward.

And one thing to consider, if this was something you wanted to do from time to time, would be to wipe and freshly set up an Ubuntu Linux system on a retired laptop, just for browsing via a VPN. That is, make it your black ops laptop. Start it fresh. Set the desktop to bright red to remind you, never log in with any of your accounts. You use it for anonymously poking around or doing whatever you want to do. Or maybe use a VMware VM where you're able to use its snapshot feature so you're always able to wipe any changes and roll back to a pre-surf point. But when you switch into anonymity, you've got to change your IP.

I would also argue, why not change your browser vendor? Use a different browser when you're in that mode. And by all means you want to be encrypting your DNS with DNSCrypt. I think if you want to switch into that mode, change your IP, change your browser, use incognito mode, don't log in with any normal presence, and use DNSCrypt, you're good to go. You're about as proactive as you can be. Oh, and by all means turn off third-party cookie tracking because even to this day it's making the developers pull their hair out.

Leo: And we're done?

Steve: We are.

Leo: That's everything I need to know? The complete set?

Steve: That's the whole tune-up.

Leo: The whole kit and caboodle? You know, we've got to put this section out as a special for everybody who wants to go through your step-by-steps and do that. That's great.

Steve: I think we end up with a nice set of guidelines.

Leo: Yeah, very good. Wow, thank you, Steve. I really appreciate that. And as some people, whoever's watching the video stream might have seen, I'm slowly adopting your tips as you go.

Steve: Cool.

Leo: Yeah. We do Security Now! every Tuesday, about 1:30 Pacific, 4:30 Eastern, 20:30 UTC. If you want to tune in live, you can. I'm watching the chatroom. Steve is

not. He's busy. He's doing a show. But I am. We also invite you to download it and watch it after the fact. Steve's got copies on his website, GRC.com. He also has something that's unique to his website, which is human-created transcriptions. Elaine Farris does such a good job of getting all the geekery in, properly spelled and all that.

Steve: She really does.

Leo: Yeah, GRC.com.

Steve: I get into a SQRL mode, and the rest of the world just - I worked 14 hours day before yesterday on SQRL.

Leo: Oh, Steve, wow. You love what you're doing, though; right?

Steve: I had a ball. I love what I'm doing.

Leo: Are you doing - your side of SQRL is all assembly? Or are you doing some JavaScript and stuff, too?

Steve: All assembly.

Leo: All assembly.

Steve: Actually, there's a little JavaScript on the demo page because it pings the server to see whether a SQRL, like a mobile SQRL has authenticated, in which case the page magically updates, and you're logged in.

Leo: Ah, neat.

Steve: So, yeah. Cool technology. I will be doing a - I'm making great progress. So we'll be having a full SQRL demo and coming out party here before long.

Leo: We'll name that show "SQRL Mode."

Steve: Squirrely mode. It's finally here.

Leo: Squirrely mode. GRC.com has lots of other stuff that Steve has worked on over the years. You heard us talk about it, including, of course, SpinRite, the world's best

hard drive and recovery and maintenance utility. Got to have that if you've got a hard drive. We also host Security Now! audio and video on our website, TWiT.tv/sn for Security Now!. And you can subscribe everywhere. You know, we've been around long enough that it's in every podcatcher and so forth. So just subscribe, and that way you'll get it every week.

And I think there are a lot of people who not only listen every week, but save every episode because it's like having the six-foot shelf on how computers work and security and all that. So I encourage you to do that: GRC.com, TWiT.tv/sn, or your favorite podcatcher. I guess that's it, Steve. I hate to say it, but I'll see you next week.

Steve: Will do, my friend. Till next week. Bye.