The Quest for Surfing Safety

Description: Leo and I catch up on a busy week of security news, and then we follow my ongoing search for a low-hassle solution for safely browsing the danger-filled World Wide Web.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-520.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-520-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here as we wrap up 10 years of Security Now!, our 520th episode. And you know what, there's only more news, not less. There's Firefox updates and issues, Flash updates and issues. We will talk a little bit about his decision to stop using NoScript and start using Sandboxie, a lot of feedback on that. It's going to be a great jam-packed episode. Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 520, recorded Tuesday, August 11th, 2015: The Quest for Surfing Safety.

It's time for Security Now!, the show that protects you and your loved ones and their privacy online. And this is the man to do it, the man of the hour, the man of the day, the man of the week, the man of the year, Mr. Steven "Tiberius" Gibson.

Steve Gibson: And thankfully you cannot keep that up for the next two hours because...

Leo: It starts with a G, and it rhymes with V, and that stands for virus. He is the creator of SpinRite, world's best hard drive utility. But also, also the first guy to discover spyware, write an antispyware tool, and he's had his own run-ins with the bad guys, so he has a lot to say about security. And I think until you get the Steve Gibson seal of approval, there are many, many people in this world who say, you know, "I'm not going to trust that. I want to hear from Steve." And this is where you do it each and every week. You doing good?

Steve: So, yeah. So we've got some Black Hat follow-up, as expected. But there's just so much news that I thought, you know, this is another one of those where news has to take first priority. So fear not, everybody who's been submitting your questions to the mail bag. Hopefully next week we will do a Q&A. But I wanted to mostly do a news catch-

up this week, just because there's so much to talk about. But also, my declaration last week that I was considering, well, that I had, actually, disabled NoScript, oh, that caused quite a flurry of reaction...

Leo: Oh, I bet. Oh, I bet.

Steve: ...among our listeners. And then there was a rare but true zero-day flaw discovered in Firefox, whose nature demonstrated that even putting Firefox in a sandbox wasn't protection. So I've spent some time brainstorming, and I want to wrap up today's news catch-up with the title of the podcast, which is "The Quest for Surfing Safety."

Leo: Good.

Steve: Because I'm on the quest, and I have an idea. Essentially, I've arrived at where I think I have to be. I'll share a little bit about what my browsing around the 'Net has found so far, and maybe our listeners will be able to nail this down based on some of the constraints that I have and help me find the right solution. So I think a lot of fun for us.

So I want to talk about StageFright and how that's evolving, the Android exploit that we've now mentioned the last several weeks.

Leo: Yeah, and we talked on The New Screen Savers about it, which was great, on Saturday. And that was when the Zimperium tester had first come out. And you pointed out that Lookout has one, too. And I keep running it on all my phones, including a brand new Moto G that just came out. None of them are safe.

Steve: Right. Okay. So there are, since we last spoke, what I was hoping would happen has happened, and that is we have not one, but two free, freely downloadable testers for the StageFright problem, one from the guys Zimperium, who originally found these six exploitable zero-day, well, actually they weren't zero-day. They were - he did a static code analysis and found, I think it was more like 12 problems, six of which could be leveraged into remote code exploits. That was going to be, well, it was the topic for his presentation at Black Hat this past week. And what happened was they became zero-days technically because they were discovered in the wild. So their plan was initially to wait until August 24th to release the proof of concept unless they were discovered in the wild, which they have been, and have now been incorporated into some exploit kits.

So what we have now, as you mentioned, Leo, are two freely downloadable test apps. And I have a link in the show notes. It's odd that it's not easy to find on their site. I had to go back in my Firefox browsing history and search that in order to find the page where they have a zip file containing 10 MP4s. Which, if you send them to someone, I mean, like, or to yourself, will demonstrate - these are their proof-of-concept files. So it's an MP4, a multimedia file which demonstrates each of these various flaws.

So not only can you use the testers - and what's interesting is that initially there was some disagreement between the two. Zimperium's required three different types of privileges, which put some people off because the one from Lookout didn't require any privileges. Also, it seems that - I'm trying to remember which way it went. The Zimperium one was still complaining about vulnerabilities in patched phones when the

Lookout one said, oh, you're patched now. You don't have anything to worry about.

However, Zimperium rereleased theirs. And on my - I have a Samsung Galaxy 4, I think it is. Three out of the six tests show green for these are okay, whereas the other three show red. So that gives you a lot more granularity than the Lookout app, which just says, okay, everything's fine; but doesn't, like, demonstrate explicit responses. And as I understand it, Zimperium's test is actually sending your phone or maybe their app some specific tests, which it verifies. So theirs may, although it kind of got off to a rocky start, may ultimately be the better of the tests.

But we now have them. And as I said, that's what I was really hoping for. And we've also had some movement in the industry. Samsung has said that they're going to now be doing monthly patches. And I don't know exactly what that means. But that's really what we need. As I've been saying, the smartphone industry has been really neglecting the fact that these are computers.

Leo: Samsung, LG, and Google have all agreed to do monthly patches.

Steve: Great. And as I said to you, when we were talking about this on The New Screen Savers on Saturday, cars are in the same category now. I mean, essentially, what everyone has to get is that. whether it's a doorbell or a thermostat or a car or a smartphone, these are computers. They have computers in them. And no one thinks that they have any security vulnerabilities when they ship these, or they would fix them. But everyone is always finding security vulnerabilities. That's just, I mean, no one wants to believe it. But look at the history. You know, we're wrapping up 10 years of this podcast, and we've got so much to talk about that we don't have time for a Q&A because there's just so many problems.

Leo: Isn't it amazing? Yeah, yeah.

Steve: It is. It's not getting better, it's getting worse. So what I think, it just has to be recognized that there are going to be problems. People, anyone doing a computer-based device has to put their hubris to the side and say, you know, and build in from beginning the ability to fix this somehow on the fly. If the device is connected to the Internet, that puts it in danger, then it can be connected to some sort of an update facility to do secure fixes. Even if they don't think they're ever going to use it, I'll bet you they'll be glad they built that in.

Leo: Yeah. There is, you know, Google says we're using address space randomization, ASLR...

Steve: I'm glad you mentioned that.

Leo: ...in all versions of Android since I think 4.0.

Steve: Right.

Leo: What does that do?

Steve: The proper way to put that is it makes it more difficult. It does not end the problem because we are constantly seeing ASLR bypasses. So, yes, it makes the exploit more difficult, but far from impossible, unfortunately. So, yes, that's a good mitigation, but it's not something you can rely on. We really, you know, we want this - you don't want your phone to be able to crash if it receives an MMS message because, as we know, crashes evolve into exploits. And, boy, this is a target-rich environment when you talk about 950 million devices. And the problem is many of those are still never going to get fixed that are vulnerable, you know, the earlier versions of this. They're still in use.

Leo: Yeah. Yeah, that's a big issue because, I mean, there are just so many phones out there that are never going to be patched. And so ASLR only makes it harder. It doesn't mitigate it.

Steve: Well, yeah. It mitigates it. But it doesn't foreclose the possibility.

Leo: My understanding, you correct me, my understand would be because the address space is randomized, while you could probably crash a phone with 4.0 or later, it would be very hard to have code jump to memory and execute, which is what you'd want to do to exploit it; right?

Steve: I would say it's trickier.

Leo: Not impossible.

Steve: Not impossible because, again, like for the last several years, I mean, you know, Windows has had ASLR now for many, many versions.

Leo: But it's not on by default, and most people don't turn it on because it breaks code. A lot of people put code in data.

Steve: No, you're confusing that with DEP, Data Execution Prevention.

Leo: Oh, that's right, I am. Okay.

Steve: DEP is the one that they sort of turned on softly, or they only turn on for their own things that they are secure are DEP-compatible.

Leo: That's executable code in the data space.

Steve: Right.

Leo: This just randomizes memory locations.

Steve: Yeah. And maybe theirs is better. I have not looked at it closely. So maybe Android's ASLR is for some reason able to do - is, like, more difficult to defeat. But it's being defeated in Windows and Mac all the time.

Leo: Okay. Okay.

Steve: So it just - it ups the ante.

Leo: Good to know.

Steve: And essentially you just want these things fixed. You know, it's better to fix them, rather than hope that somebody won't guess right. Because typically, for example, they're able to randomize the boundaries of the libraries to, like, one of 256 positions. But not a million positions, just one of 256.

Leo: Right.

Steve: So, you know, if it crashes 255 out of 256 phones, but you've got a target space of 950 million, then you've still got several million phones that just by shooting them blind are going to take over some of them. So, yeah. So I would say yes, I mean, it's good that it makes it more difficult. But there are still ways to bypass ASLR. It's just more work.

So on the Windows 10 privacy side, we've got some updates. It was inevitable that somebody would create an app. And so there's an app for that. It's on GitHub, called Disable Win Tracking. And it does four different things. So it's open source, and it's been evolving. It's at version 1.5 now as it's being fine tuned. And the author states that he intends to add features to it, as I said, as we sort of learn more about Windows 10.

And so it's sort of a simple-to-run app. It's written in Python. And so you can, if you've already got Python installed, and one other requirement, there's an additional library it requires, then you could just run the Python script in order to make it go. But there is a self-contained executable that includes the Python runtime bound into the EXE. So that's probably easier for most people to use.

Anyway, so it allows you to turn off what they call their "telemetry services," which is sort of the background, how are you using Windows, what are you doing stuff. So that's one option. You can either delete or disable the tracking services. There's radio buttons where you can choose to delete or disable. And I would just be happy with disabling, for what it's worth. There are two services, one called DiagTrack Diagnostics Tracking Service, and one called dmwappushsvc, which is a WAP Push Messaging Routing Service.

And I don't know, I don't think it makes sense to delete them. I think disabling them,

you know, people are disabling Windows services all the time. They stay disabled. So I would just say people ought to disable them so they could, if there were some need to reenable them in future, you could. And you can disable logging that is enabled by certain trackers. Anyway, so a few features which are very simple to disable using checkboxes. And I imagine we'll see similar things like this in the future. So I did want to mention that.

Also, remember the people - there was a huge breach of I think it was 13 million-plus emails that escaped, and a service was produced called IsLeaked.com. And IsLeaked.com, you could put your email address in, and it would just do a search to tell you whether that email address was among those that were leaked in this massive breach. These guys did a very nice kind of walkthrough guide, and so their domain is fix10.isleaked.com, which is sort of a summary of what you and I walked through with our listeners last week, Leo. And by the way, we got a huge number of...

Leo: People loved that, yeah.

Steve: ...of very positive tweets, feedback from last week's walk through the privacy settings. And so fix10.isleaked.com is a...

Leo: This is screen shots of all the stuff we talked about, basically, yeah.

Steve: Yeah. Yeah. It's sort of the - it's basically screenshots and red circles around you want to set this switch to this and this switch to that. So, and maybe for people who want some additional information. Again, this is, unfortunately, there's enough issue with Windows 10 privacy that we're seeing lots of articles and sites and now apps that are suggesting here's how you can do this.

Also people were very concerned about whether TrueCrypt was cracked because the website that I love to hate, TheRegister.co.uk...

Leo: Yeah, you know, I saw that, and I thought, "Steve will want to talk about that."

Steve: Oh, yeah. Well, first of all, it is significant that not one other site, tech or otherwise, quote, "picked up on this story," unquote. And so TheRegister.co.uk's headline was, "Wait, what? TrueCrypt decrypted by FBI to nab doc-stealing sysadmin." And their subtitle was "Do the Feds know something we don't about crypto-tool? Or did bloke squeal his password?"

Leo: You get a choice there. One of them seems more likely than the other.

Steve: Yeah. So they say, you know, the first paragraph says: "Discontinued on-the-fly disk encryption utility TrueCrypt was unable to keep out the FBI in the case of a U.S. government techie who stole copies of classified military documents. How the Feds broke into the IT bod's encrypted TrueCrypt partition isn't clear. It raises questions about the somewhat sinister situation..."

Leo: Oh, please.

Steve: "...surrounding the software team's sudden decision to stop working on the popular project last May." Oh, lord. Well, okay. There's nothing sinister about it.

And one thing we know about TrueCrypt is that the sole vulnerability in this or any other password-based system is how good is your password. And if your password is not good, there are tools that can do brute-force decryption of a password. We know that. And as far as we know, TrueCrypt has survived two audits that have looked closely at the way it implemented its crypto, and nothing was found wanting. So this is either, yes, he gave away his password, or someone applied a brute-force cracking tool.

There's no doubt that the FBI has access to a brute-force cracking tool for TrueCrypt. And we have covered many stories where people used good passwords. And there was one that I remember, only because it involved several nations, it was like the intelligence or law enforcement in Brazil absolutely had to get data from a TrueCrypt-protected drive. And they sent it to the U.S. FBI after they failed to be able to get into it, and our FBI couldn't, either, because the person used a good password. That's all you have to do.

Leo: Doesn't even have to be, like, that good. Just pretty good.

Steve: Yeah.

Leo: And I would think it'd be enough.

Steve: Yeah. So, okay. I'll come back to this topic of Firefox vulnerability later. But I did want to make sure everyone knew that they needed to at least update to 39.0.3, with some level of urgency, but not house on fire. But 40 was just released this morning, version four zero, just released this morning. So first of all, the vulnerability that was fixed in 39.0.3, I think I was at 39 when this happened. And so they were doing some incrementals in the meantime.

Here's what happened, though, which is what is a little bit chilling. First of all, we almost never discover any actual vulnerabilities in Firefox. Typically it's in add-ons, you know, like Java or Flash that are running in Firefox. And of course we talk about add-ons for controlling those, most notably NoScript. So a security researcher, Cody Crews is his name, discovered that a malicious advertisement on a Russian news site was exploiting a previously unknown, so this was a zero-day vulnerability, in Firefox's built-in PDF viewer. It was able to essentially breach the same-origin policy.

The same-origin policy is another thing we've talked about a lot. That is probably the single most significant security firewall that all browsers enforce, which says, if a page comes from Amazon.com, script which is running in that origin, that is, the idea is that, if the HTML which contains JavaScript, that is, as we know, executable code. But that executable code is unable to access anything other than from Amazon.com. It specifically can't just go out and do other things because that would represent a serious security problem. So individual origins, the origin being like www.amazon.com, code loaded from Amazon.com is restrained, restricted to that same origin, to only reading and writing things within that same origin.

Well, it turns out that there was a way that code in Firefox's PDF viewer could allow an origin breach, a same-origin breach, which allowed some script which was provided by this ad to access the system's local file system. So essentially it got, in this case, the cross-origin wasn't to a different Internet site, it was to the local file system. That JavaScript then ran, you know, it searched through - I was going to say "ransacked" - the local site, looking for subversion files, s3browser files, FileZilla, and libpurple config files on Windows systems. And on Linux systems it looked for global configuration files in the /etc/ directory, as well as .bash history, .mysql history, .pgsql history, and .ssh files, any other files with "pass" or "access" in their names, and shell scripts. And, if found - and of course Linux systems would typically have lots of those, it would upload those to a server in Ukraine. So, oh, and it left no trace on the host or the target system, the vulnerable system, the victim, that this had happened.

So Mozilla responded immediately after Cody's report. This was not ever seen widespread. It was, you know, he discovered it. He reported it. They fixed it. So only people who happened to use a version of Firefox that had the PDF viewer in it were vulnerable. Oh, and only Windows and Linux were initially known to be vulnerable, although in a later update to their posting on the Mozilla blog they noted that they had seen Mac then being targeted. A version of the payload was also looking for those sorts of files with Mac filenames and also trying to export those.

So I guess really the only problem would be somebody who wasn't updating Firefox, who still had an older version of Firefox, and happened to go to a site that was serving an ad that was leveraging this problem. This was immediately fixed, like in hours. And I saw people congratulating Mozilla on the speed with which this was addressed. So you want to make sure you've got at least 39.0.3.

And what's interesting is that my proposed solution, that is, for running Firefox in a sandbox, would not have prevented this problem because Sandboxie prevents system alteration, but doesn't prevent the browser from reading from the system if it wants to. So that gave me some pause and sent me on a search for another solution, which is what we'll talk about later in this podcast.

Meanwhile, we've gone to v40 of Firefox, with not huge changes. Largely they're billing this as an appearance and usability improvement for Windows 10. So it more closely matches the Windows 10 UI. It's easier to use on a touchscreen. They, like, increased the size of the close buttons on the tabs to make them easier to touch. The address bar uses a larger font. They've improved some of the graphics for smoother scrolling and animation. Some small security improvements, but nothing huge. Better warnings for unsigned browser extensions. And it's worth noting that, when they go to the next major release, v41, they're going to completely block unsigned browser extensions. So I'm sure that browser extension authors are going to be - are already aware of the fact that their extensions will not be permitted to operate with the next major version of Firefox. So Firefox moves forward.

I was on with Father Robert before Black Hat, and he had Mark Smith on, who enjoys going to Black Hat and was talking about how great it is to be there. And I was sort of - and you and I have talked about this, too, that my sense is, yes, I mean, it's an experience. But everything that happens is posted online.

Leo: Right.

Steve: And you have the problem of being there, that many times four things that you want to see are all happening at the same time.

Leo: Yeah, Robert said the same thing, actually, yeah.

Steve: Right. And anyway, I got a kick out of this retweet. Our friend Matthew Green, the Johns Hopkins famous cryptographer, retweeted a tweet from Thomas Ptacek, who is one of the founders of Matasano Security we've spoken of often. And he tweeted, "There is nothing you will learn at Black Hat or DefCon that you won't learn quicker online."

Leo: Wow.

Steve: And I thought, well, there's two security gurus who are in agreement with the fact that, yeah, I mean, I get it that going to the conference, the whole experience is the whole hacker ethic sort of thing. And it's like, yeah, okay. I mean, I guess in the same way that I'm really not a gamer, but I like looking at the technology of the games, similarly I'm not into the hacker meme. I like the technology of the security issues. And so I think that really is my focus. So I'm happy that it's all online.

Leo: In fact, they came back with video, too, so we'll have a lot of coverage on our shows coming up.

Steve: Oh, good. Good, good.

Leo: Yeah.

Steve: So there was this one really interesting - I couldn't wait to see what was going to become of this - Black Hat presentation titled "The Memory Sinkhole." And the abstract wins the hyperbole award of the year. The title was "The Memory Sinkhole - Unleashing an x86 Design Flaw allowing Universal Privilege Escalation."

Leo: Oh.

Steve: Oh, I know.

Leo: Wait a minute, x86. That means everything.

Steve: Yeah. It's like what we've had since the mid-1990s.

Leo: What?

Steve: And they said: "In x86, beyond Ring 0 lie the more privileged realms of execution, where our code is invisible to AV, we have unfettered access to hardware, and can trivially preempt and modify the operating system."

Leo: Jiminy.

Steve: "The architecture has heaped layers upon layers of protections on these negative rings, but 40 years of x86 evolution have left a labyrinth of forgotten backdoors into the ultra-privileged modes. Lost in this byzantine maze of decades-old architecture improvements and patches, there lies a design flaw that's gone unnoticed for 20 years."

Leo: The way you're reading this makes me think, mm, no.

Steve: "In one of the most bizarre and" - I will say "bizarre and complex," he gets that right - "the most bizarre and complex vulnerabilities we've ever seen, we'll release proof-of-concept code exploiting the vast, unexplored wasteland of forgotten x86 features, to demonstrate how to jump malicious code from the paltry Ring 0 into the deepest, darkest realms of the processor."

Leo: Ooh. Ooh. Ooh.

Steve: "Best of all, we'll do it with an architectural zero-day" - yes, it's also a zero-day - "built into the silicon itself, directed against a uniquely vulnerable string of code running on every single system."

Leo: Bwa-ha-ha.

Steve: So it involves SMM. And you may think, what? Is that the scattering matrix method?

Leo: No.

Steve: Is it Scots Musical Museum?

Leo: You googled it.

Steve: Is it the Science Museum of Minnesota? Oh, no.

Leo: I'm thinking it's memory manage.

Steve: The Shanghai metals market? It wouldn't be that.

Leo: No, no, no.

Steve: How about single-molecule magnets? Nah, unlikely, no. Single monthly mortality, which actually turns out to be a measurement of prepayment of loans? No. Social media marketing, of course not. Soft magnetic materials? Nope. Solar maximum mission? Unh-unh. Stepwise mutation model? Well, no. How about storage modification machine? We're getting warm. Or could it be system management mode?

Leo: Oh.

Steve: Yes, indeed.

Leo: Next time, can you turn the lights off and hold a flashlight under your chin?

Steve: So it turns out, I mean, so I read this PDF, his whitepaper, which was blissfully short. The abstract took up about the first quarter of the first page. There's something that we've always had called the Programmable Interrupt Controller, the PIC.

Leo: Yeah.

Steve: We've always had that. That's, you know, that's how x86es do interrupts is that you have a whole bunch of wires, of interrupt signals. And everybody remembers the painful days, like the XT and the PC, where you had IRQ 8 and IRQ 9, and, what, COM port, COM0, or was it COM1, I don't remember what it was named. I think it was COM1 used IRQ 4.

Leo: Right.

Steve: And 2 used IRQ 3.

Leo: And COM3 used IRQ 4 again.

Steve: Exactly, yes, right. And I do remember that the disk drive used IRQ 10. Oh, no, no. Video was IRQ 10. No, I'm confused because INT 13, well, there was interrupts and IRQs. Anyway, it was a mess. And you'd sort of run out of them. And there were some, as you noted, that you could share, and some that you couldn't.

Leo: Interrupt conflicts were the bane of my existence for 10 whole years.

Steve: And, but, boy, you know, if you were doing The Tech Guy back then, Leo - actually you were.

Leo: I was. No, when I say "bane of my existence," I mean calls. "I am trying - I'm having trouble. My modem's on COM1, and I've got a mouse on COM3, and they don't seem to work." Well, you've got an interrupt conflict, my friend. "I move the mouse, and the phone goes out." [Groan]

Steve: Ohhh. So as we evolved, we got the APIC, the Advanced Programmable Interrupt Controller. And so the Intel architecture has both I/O ports and lots of memory, as we know. It turns out that reading and writing to I/O ports is inconvenient for the operating system. It's easier if it's just in memory. And so, for example, video, the original video display was memory mapped, meaning that the processor would write to memory addresses, and that was actually performing essentially an I/O operation to hardware on the display adapter where the video memory resided. Similarly, the advanced programmable interrupt controller. It had some I/O where you could use input and output instructions. But they're very crude on the Intel system.

So instead, they memory map this advanced programmable interrupt controller so the operating system can just see it as a range of memory. And it is a 4K page which can, for convenience sake, can be placed anywhere within the 32-bit address space, the 4.3 billion addresses. It needs to be aligned on a 4K boundary. So it's like, it's a 4K page that can exist on any 4K block of memory within the 4.3GB region.

So there's something else called System Management Mode. And it, too, is memory mapped. And it can only be accessed from Ring 0, that is, the most privileged ring of the operating system. And that's also true of the advanced programmable interrupt controller. Both of these things can only be accessed through Ring 0.

It turns out that it is possible until Sandy Bridge. So Sandy Bridge happened, that was the micro architecture we got starting in 2011. So four years ago Intel realized this was a problem, and they fixed it. So this is really no longer a problem for anyone. But in 2010 and earlier, from like 1997 to 2010, it was possible to deliberately program the advanced programmable interrupt controller so that its memory mapping overlaid the system management mode memory mapping. And if the two collided, reads that were being attempted to be made from system management mode would come back as zero, and writes would be intercepted by the APIC and would not penetrate, would not get through to system management mode.

So what this guy figured out was, if you, like, really had to, and you already had OS-level Ring 0 privilege, there was a way to exploit, until Sandy Bridge, the fact that these two things were memory mapped and that the APIC, the advanced programmable interrupt controller, took priority, that is, if these regions of memory were colliding, the system gave priority to the APIC. And so he figured out a way to use that interaction so that the system management mode would not receive the instructions it was supposed to. They would be blocked by the programmable interrupt controller. And if you also knew exactly what code was where in this particular motherboard's system management mode memory - and they all differ by motherboard version and make and model and update level, so it's completely impractical - then it was possible to, if you already had root privileges, to modify the firmware to install a rootkit in the system management mode memory. And that's what it was.

So it was like, okay. Well, fine. The sky is not falling. Life has not ended as we know it. You already have to have the kind of privilege that would allow you to install a rootkit in a system. But this allows you, if you know enough about the hardware you're trying to install it on, essentially to push a rootkit down into the firmware through this little hole,

if, like, everything lines up just exactly right.

And as I mentioned, we talked about this on The New Screen Savers on Saturday, and I referred to it as like, you know, if you were hopping on your right foot during a blue moon and everything lined up perfectly, then maybe this would work for you. But, like, okay, you know, a perfect example of somebody really, I mean, the description was more fun than anything else. So nothing to worry about here. Nothing to see.

Leo: It's almost as if they make money if more people show up for their talk. Like they're really selling the talk.

Steve: Well, and nobody disputes that this was a very clever hack. Nobody would have ever thought about it. But nobody ever has or actually could ever use it, either. So it's like, okay.

Leo: Yeah. Well, nice.

Steve: Now, I know you're a fan, Leo, of PushBullet.

Leo: Yeah, I use it.

Steve: And I haven't quite figured out how to use it, but I want to because I'm sort of straddling ecosystems. You know, I'm not completely in iOS, although I have an iOS phone and iPads. And it would be nice to get them to communicate into my Windows environment with my browsers. And as I understand it, it's able to do that.

Leo: PushBullet's available on iOS. I didn't know that.

Steve: Yeah, it is.

Leo: I thought it was just an Android-only.

Steve: Yup, it is on iOS. Anyway, the news is that they just added something that had been much requested. And so for any of our other PushBullet users, they now add end-to-end encryption.

Leo: Oh, nice.

Steve: So, yes, so that is nice. You need to give each of your PushBullet instances a password. And so, you know, the moment I heard "end-to-end encryption," it's like, whoops, wait a minute, if you don't also have authentication, then we know you could suffer a man-in-the-middle attack. So that really wouldn't buy you any security. The good news is they did it right. First of all, I did some follow-up research, and the guys at

PushBullet were saying, wait a minute, you know, if you trust us, then you don't need end-to-end encryption. And if you don't trust us, then you shouldn't be using this anyway because who's to say...

Leo: What we're doing, yeah.

Steve: ...we didn't bury something, yeah, exactly. Which of course is always the dilemma. But the people that wanted this, and there was some dialogue over on Reddit, explained that it's not PushBullet we don't trust, it's anybody else listening to the traffic. And that was the problem, that all of the PushBullet traffic, even though it - and the PushBullet traffic was over SSL, so it was secure anyway. But if you allow that SSL might be vulnerable, there was no inner encryption inside the SSL tunnel to protect the PushBullet traffic. Now there can be.

So anyone using PushBullet as of, I think, like this just happened, like today or yesterday - oh, actually it's dated. It was an article in Android Police dated today, 8/11, saying that PushBullet now supports end-to-end encryption, so for notification, mirroring, SMS, and copy-and-paste, and sync, so its various features. You need to give it a good, strong password. And it uses that to generate a symmetric 256-bit key to encrypt the tunnel. So all of your various instances at each end will need the same key. But when they have that, then you really do have strong encryption of the stuff that you're pushing back and forth. So I wanted to let everybody know that was now there, to update your copies of PushBullet.

Leo: Yes.

Steve: Okay. Now, this is bizarre. Or, in fact, it's so bizarre that security people didn't believe it. Oracle's Chief Security Officer, a woman named Mary Ann Davidson, posted something on her Oracle security blog which caused the security researcher whom we've spoken of often, Matt Blaze, to assume that Oracle had been hacked, and this was posted as a parody. He tweeted, he said, his first tweet was: "My first assumption after reading this was that Oracle's web server was hacked and this article was a parody." And then he said, he tweeted: "Oracle should get an honorary membership in the Locksmithing Guild for their heroic 'security by not looking' policy." And it has since been pulled by Oracle, and Oracle has apologized, believe it or not.

Leo: Yeah, yeah. Although you can see it on the Wayback Machine, if you should so desire, yes.

Steve: Yes, it has been captured. It's been archived. So essentially this Mary Ann Davidson, I don't, I mean, and reading this really is interesting because I don't know the person, and I don't want to make any disparaging assumptions. But, wow. It's difficult to understand what frame of mind Oracle's chief of security officer must have to have truly said these things. So, for example, essentially what this came down to was she was threatening anyone who did a security analysis of Oracle's intellectual property, Oracle's code, with breach of license, and complaining that people were doing a static code analysis and finding problems with Oracle's software and reporting it.

So among other things in her blog she said: "I want to reiterate that customers Should

Not and Must Not" - and those are in caps, capital S, Should, capital N, Not, and capital M, Must, and capital N, Not - "reverse-engineer our code. However," she writes, "if there is an actual security vulnerability, we will fix it. We may not like how it was found, but we aren't going to ignore a real problem. That would be a disservice to our customers. We will, however, fix it to protect all our customers, meaning everybody will get the fix at the same time. However, we will not give a customer reporting such an issue that they found through reverse engineering a special one-off patch for the problem. We will also not provide credit in any advisories we might issue. You can't really expect us to say, 'Thank you for breaking the license agreement.'"

And then she said elsewhere, but in the same blog post: "If we determine as part of our analysis that scan results could only have come from reverse engineering," or, she says, "in at least one case, because the report said, cleverly enough, 'static analysis of Oracle [blank]'" - whatever the product was - "we send a letter to the sinning" - and she used the word "sinning" - "to the sinning customer, and a different letter to the sinning consultant who was acting on behalf of customer, reminding them both of the terms of the Oracle license agreement that preclude any reverse engineering." And then she ends with, again all caps, "So Please Stop It Already." Unbelievable. So anyway - and I'll remind everybody that this is the company who brought us Java.

Leo: Yeah.

Steve: You know, the most ridiculously security vulnerability breach-filled nightmare on the web for a decade. Which is really only no longer a problem because kicking and screaming, they finally basically gave up. And there are still companies that absolutely require that Java be accessible in browsers. But fortunately, most people outside of a corporate environment are no longer running Java. And of course Brian Krebs's advice, and ours, for years has been you do not need it. You should not have it. Just, you know, don't patch it, remove it. And this company is saying that they're unhappy with the fact that people are now inspecting their code for problems and finding problems. They're saying, "You should not be looking at our code. That is a breach of our intellectual property." It's like, wow. Okay.

Leo: Amazing, yeah.

Steve: You know, this just says stop using Oracle stuff. And it generated such an uproar within the security community that they removed the blog posting. The link now gives you a 404 error. It was captured by several different archiving sites, just to say, whoa, okay, look. Consider what this, I mean, what must - this is not some random unofficial person. This is the head of security at Oracle posting this.

Leo: Amazing.

Steve: And saying, "Don't look at our code. Bad. And if you do, well, yes, we'll fix it, but then we're not giving you any patches or credit or anything." Wow.

Meanwhile, the EFF has released their v1.0 of a product I think, Leo, you'll be able to get behind.

Leo: I installed it.

Steve: Because this is not an ad blocker.

Leo: Yeah, I installed it immediately.

Steve: Good. This is Privacy Badger.

Leo: Although it could be kind of used as an ad blocker. It works as an ad blocker.

Steve: Well, okay, yeah. And we're going to go into this.

Leo: On our site it works.

Steve: I think this is, well, actually there's something you can do that I think you probably will do, Leo.

Leo: Okay.

Steve: And that is, they have, sort of in concert with this, the EFF and a bunch of others, let's see, Disconnect, Medium, MixPanel, AdBlock, and DuckDuckGo, all have produced a Do Not Track policy document. And you should, because you are not tracking, you know, TWiT.tv is not a tracking site, you can put this policy document on your server, and Privacy Badger will see it and then understand that you're not a tracking site. Anyway, so let me back up a little bit.

Leo: Well, okay. But let me show you the tracking that's on our site. Actually, well, I could - just in response to that, they identify four trackers. Google Analytics, they say, is a tracker. We of course use that to measure traffic. Ad Services is an ad - that's what serves the - there's two banner ads on our page. That's what serves them. Fonts are Google typefaces, and New Relic is a monitoring system that warns us if our server is down. So while Google Ad Services is certainly tracking, these others, I mean, are tracking, but only in a sort of noncommercial sense.

Steve: Right.

Leo: And that's the problem with a tool like this is it doesn't really distinguish between, I mean, in this, it does say all of these things are not tracking.

Steve: Let me explain what they say it is. And again, this is just released, so we need to get some experience with it. So they said: "EFF is excited to announce that today we are

releasing version 1.0 of Privacy Badger for Chrome and Firefox. Privacy Badger is a browser extension that automatically blocks hidden trackers that would otherwise spy on your browsing habits as you surf the Web.

"As you browse the Web, Privacy Badger looks at any third party domains that are loaded on a given site and determines whether or not they appear to be tracking you, in other words, setting cookies that could be used for tracking, or fingerprinting your browser. If the same third party domain appears to be tracking you on three or more different websites, Privacy Badger will conclude that the third-party domain is a tracker and block future connections to it.

"For certain websites, if Privacy Badger were to block an embedded domain entirely, it would break the site's core functionality. For example, if Privacy Badger were to block licensebuttons.net, Creative Commons buttons would no longer load. In these cases, Privacy Badger blocks the domain from setting or receiving cookies or 'referer' headers, but allows the embedded content to load." So it's not, they're saying, it's not just a black-and-white block everything indiscriminately tool.

And then they said: "To be clear, EFF isn't against websites seeking to build businesses around advertising. More business models means a more vibrant web. But advertising cannot come at the expense of user privacy and the inviolable principle of consent. Until the online tracking industry changes its ways, the only option for users is to protect themselves by installing tools such as Privacy Badger.

"Privacy Badger 1.0," they say, "works in tandem with the new Do Not Track policy, announced earlier this week by EFF, Disconnect, Medium, MixPanel, AdBlock, and DuckDuckGo. Installing Privacy Badger also enables the DNT flag as a clear signal to sites that the user wants to opt out of online tracking. Privacy Badger inspects third-party sites for a commitment to honor that request under the DNT Policy." And this is that document I mentioned a second ago. It says: "If it finds one, it will unblock that third party by default. That way, web services that do the right thing by users can continue to collect anonymous data or show anonymous ads, while those that don't will be foiled by Badger's protections."

So, okay. So what they're saying is they do have what essentially is a conduct pledge. And you can see it at EFF.org/dnt-policy. And that's a pledge which companies can put on their server which Privacy Badger will see and essentially negotiate, essentially say, okay, these people are pledging to honor user privacy. We're going to consider that they're not tracking, even though it does allow for - this pledge allows for the anonymous collection of data. And of course we know that the EFF tends to be very proactive about these sorts of things.

But for me, we know that I've been a proponent of the DNT header. And if users want to say "I don't want to be tracked," then the browser ought to be able to transmit that to sites, saying, you know, I am asserting that I don't want to be tracked. The problem, of course, is that it's just based on the honor system. There's no way to enforce not tracking unless we block content. So, I mean, I think we're in a situation where we need sort of a negotiated solution. We need something where users can feel that their privacy is being respected while a business model can continue to survive that generates ad revenue for showing people ads.

And then the question is, you know, I guess really the issue is the tracking side. I have no problem seeing ads, but there's the argument being made that databases are being aggregated in order to improve the quality of the ads being seen, and that makes some people uncomfortable. So anyway, I did want...

Leo: Then they should stop going to the sites, instead of telling me how to run my business. You don't know, no one knows whether...

Steve: I know.

Leo: By the way, just in case you're wondering, 41%, use of adblockers is up 41% in the last 12 months. And a study by PageFair and Adobe says that costs publishers $21.8 billion in ad revenue. That's all I'm going to say.

Steve: Wow, yeah.

Leo: Doesn't cost me anything. That's not where we make our money. But...

Steve: Well, and remember, too, that I - was it TechCrunch? I think it was at TechCrunch where I found the blog posting from the founder who said, you know, we're really being hurt by this, folks. We're offering free content.

Leo: Well, that's a mistake. They probably shouldn't say "free." No dollars are exchanged, but we're offering you content.

Steve: Or sponsored, yeah, sponsored content.

Leo: Ad-sponsored content. And, I mean, you know, you look at GigaOM going away. You look at the, you know, sites shutting down, it's just going to accelerate. It doesn't really affect our podcasts. But I just - I worry that there's not going to be any news sites to report.

Steve: Well, and don't you also feel it's a little bit like, I mean, that there is like, well, what we've gotten into is an arms race.

Leo: Yeah.

Steve: Because, for example, I go to sites, and a huge popup comes up that blocks the entire page. And it's like, okay, fine. And then I have - and if I'm on a pad, and I've already zoomed in, then I have to zoom out, find the close button, the close X in the upper right-hand corner.

Leo: No, I understand it's annoying. Then you shouldn't visit that site.

Steve: But I don't know that until I go there. I mean, you know, I'm following links.

Leo: Yeah, well, then, don't go back. I mean, the way to handle that is not to visit sites that offend you with their ad policies. The real - I understand the real issue is invisible tracking. You don't know what's going on. A big takeover ad, if you don't like that, just don't go to the site. But what we need things like Privacy Badger for is to let us know what's behind the scenes.

Steve: Do you think that a big takeover is effective? I mean, is anyone looking at that and going, oh, wow, I did need that? I mean, isn't it just pure annoyance?

Leo: I think probably it is. I don't think - I agree, but I don't think people would do it if it weren't effective. The real problem is nobody sees banner ads at all. You know, I tried to not have banner ads on our website. I didn't want to have that because I knew we'd have this conversation. And I know people, I got email from people who said I'm never going to listen to your podcasts again because you have trackers on your website. And so, but, you know, we use those banner ads as value-added to sell podcast ads. It's part of our business model. I feel like these are kind of - these systems are imposing an arbitrary business model on a site because you don't like the business model. But that's not kind of how it works. You don't go in a store, say you know what, you shouldn't be selling shoes because you're just - you just don't go to the store anymore.

Steve: Yeah. I mean, and from a technology standpoint, the trouble is users do have the ability...

Leo: Right.

Steve: ...with the technology to block the retrieval of third-party ads. The only way I can see this works is if these become first-party ads. That is, if the site itself serves them, rather than bouncing the user's browser...

Leo: Well, that's, again, your decision about how a business model should run. The reason it's not done that way is because advertisers want, at least, at the very least, they probably want information, but want to count it themselves.

Steve: Right.

Leo: I can say, hey, a million people saw your ad. They want to know. They want to count it. At the very least they want that.

Steve: Right.

Leo: And you can't be...

Steve: I think, I mean, we have a tension. There is tension here between what users want and this ad-based model. The problem is the ad-based model is not able to force people, due to the technology of the browser and server, is not able to force people to view content.

Leo: No, no.

Steve: I guess what'll have to happen...

Leo: You want to know why you see so much "Honey Boo Boo" on network television?

Steve: Yeah, well, I don't...

Leo: Do you want to know? And why "True Detective" is on pay television? Because the ads don't work. So they have to do cheaper and cheaper, crappier and crappier programming, aimed at people who can't figure out how to TiVo on fast-forward.

Steve: Yeah.

Leo: That's why you see more "American Idol" stuff, and why they really focus on live, because people are skipping ads.

Steve: Yeah.

Leo: I'm hoping, I figure I'll be out of this business before it really crashes. I'm hoping.

Steve: Yeah, I mean, it's going to be interesting to see how it evolves because, I mean, okay. So adblockers, if adblockers are being adopted by that percentage of people, I don't believe there's that percentage of people that even are aware of tracking at all. It must be annoyance. The ads got too abusive.

Leo: Probably, yeah.

Steve: And so people looked for a recourse.

Leo: I would guess that's most adblocking, yup.

Steve: Yeah.

Leo: It's annoyance.

Steve: And so, yeah. And so I would - and this is the problem, is that - and remember that phase when ads were flash-based, and stuff was jumping around? I mean, it was really...

Leo: I don't know if it's in your rundown, but Yahoo! was serving malware last week.

Steve: Yes.

Leo: On Flash-based.

Steve: You're right. I forgot to get that in the notes, yes.

Leo: Well, you know, it's so much now, it's like a hundred - according to this study - and I, you know, it's from Adobe, I don't know - 198 million monthly active users were running major browser adblocking extensions as of June. 198 million.

Steve: Yeah.

Leo: Mostly in Oregon. Forty-five million are in the U.S. Which, I mean, so it's a small number compared to the total number. But it's growing fast. That's the cause for concern. Oregon has the highest adblocking rate in the country, 16.4 percent, Washington, D.C. half that.

Steve: Interesting. I wonder, like, if the Oregon newspaper, if the Oregon web-based newspaper has particularly annoying ads.

Leo: Maybe, could be.

Steve: Something like that. But...

Leo: It's probably a higher percentage of technically sophisticated people, frankly.

Steve: Ah.

Leo: I would guess.

Steve: So my point is that we know that, like, the volume gets turned up on ads on

television, even though there's now legislation that says it's illegal to do that. I watch it happen. Ads are desperate to get your attention.

Leo: Mm-hmm.

Steve: Similarly, web-based ads, they're using animation and distraction, I mean, they want to distract us. And we don't want to be distracted. So, I mean, so really the way to solve the problem is for the advertisers to back off and recognize there is a limit to how distracting they can be before users are driven to find a solution. And the fact is, a solution exists called adblocking.

Leo: Right.

Steve: You know, your banner ads are not, I mean, I had to look for them, Leo. I couldn't even find them on your page.

Leo: I know. That was the compromise. I said, "I don't want banner ads." Lisa said, "We have to have banner ads. It's like a million dollars a year difference." Not from the ads alone, but the value that they provide to our ad sales. And I said, well - and we negotiated it down to two in a fairly unobtrusive ad. Actually, as it turns out, we can't sell the bottom ad. So that...

Steve: Yeah, I mean, I had to...

Leo: That we give people.

Steve: Yeah, I mean, I had to look for them.

Leo: I know.

Steve: Yeah, I mean, they're completely innocuous.

Leo: Yeah. Well, we all get hurt by the obnoxious ads. But, you know.

Steve: Right. I think that's exactly it.

Leo: Yeah. That's all right.

Steve: And in fact, that's sort of what Adblock Plus, that was what, remember, that was their deal. What they had, they had, if your ads are not annoying, then we're going to let them through. And otherwise, and then...

Leo: Well, they had an approved ad policy. But you could also pay them.

Steve: Right, and that's the problem is that they wanted to make some money, too. And so then it became racketeering.

Leo: You know, I would be just as happy to have everything we do behind a paywall. But it has some negative impacts. First, it's undemocratic. If you can't afford to watch our stuff because you don't have a buck a show or $5 a month, that leaves you out in the cold, and I want everybody to be able to see this. Also, I've found, in our early years, as you might remember, we asked for - you were the first show to have ads, by the way. We asked for donations. And it was fine, but it was enough for one person to kind of make a living.

Steve: Right. Right. And you know, one of the things that could work, a model that could work, if we ever get a micropayments system in place. Because, for example, sometimes I follow a link to The New York Times, which is behind a paywall. And sometimes it doesn't matter. Sometimes it does. And I remember, and this happened, like, last week, you've already seen five things this month.

Leo: Right.

Steve: So subscribe or die. And they showed me enough of the whatever it was that I want the whole thing. If there were a micropayment scheme, I would press a button for them to have a penny or whatever they want from me. I mean, they could show me. You know, if you want to read this, it'll cost you two cents. I think, yeah, I want it. It's worth two cents to me. Bang. Now I have access to it. You know? Maybe that's where we're going to have to end up getting is a system like that.

Leo: I don't know what the answer is.

Steve: I know. I don't know. So for anyone who is wondering if we've had enough time bashing all kinds of exploits so far, there is a new version today of Flash. I didn't even bother digging into what nightmares they had solved. But I was running an older one. It's now at v18.0.0.232. So 18.0.0.232. And you can go to Adobe.com/software/flash/about, and hopefully your browser won't run it. It'll require you to click on something to, like, permit that because the Flash, Adobe's version checker, their Flash About thing, is itself Flash, which a secure browser will not run by default. So hopefully.

And, let's see. Oh. A miscellaneous question. Many people have asked this, and I finally was moved to put it in the notes. A couple people tweeted, how do I clean my Contigo thermos? You know how everyone has seen my thermos that I use. And, boy, after a year or two it looks like - what's that stuff that's absolutely black?

Leo: Ebony? Ivory?

Steve: There's a term for, like, a matte black finish which is just like no light bounces from.

Leo: Ice?

Steve: Anyway, you look in it, and it's just black. Anyway, the point is there is an amazing cleaner, and it's called Puro Caff, P-U-R-O C-A-F-F, two words. Amazon carries it. I found it at Starbucks. It's what Starbucks uses for, like, dealing with their own coffee stuff.

Leo: You know, we are going to get our Temperfect Mug one day.

Steve: Yeah, I know, if they - I get email every month or two. We're still at it, you know. And it's like, okay, well, yeah. Anyway, so for what it's worth, if anybody is seeing that the inside their anything that contains coffee is really looking scary, Puro Caff, P-U-R-O C-A-F-F. It is freaky how well it works.

Leo: I'll have to get some of this.

Steve: You put like a tablespoon in the bottom and fill it up with hot water, and it fizzes. It's like kind of a Vesuvius sort of thing. And you just sort of leave, let it sit for a couple hours. And then when you pour the water out, it's all brown, and it leaves pure silver behind. It's just incredible how well it works.

Leo: Now, I had a Screen Savers mug that over 20 years or whatever got pretty well stained. It was a ceramic mug. And Lisa said, oh, no problem, just put some crushed ice and salt and water in there and leave it. And so, and I'm thinking - and it did, it worked. It's like it bleached it. So I'm wondering if maybe there's salt. So that does look good. I'm looking at Steve's perfect...

Steve: Yes, that is after I did it. And it was black. It looked like the heart of darkness, like, before. And it's like, zero effort. You just put a tablespoon of that stuff in there, and it's like shiny and brand new again.

Leo: And it's not poison.

Steve: No. And it can't be because, you know, it's meant to be going where...

Leo: Sure. It couldn't be poison.

Steve: Could not be poison. No. I mean, I would rinse it thoroughly because it's scary. It's scary stuff. But, boy, does it do the job.

Leo: I'm getting some right now. Couldn't be poison.

Steve: Unh-unh. And I did want to mention that next week is the finale of the first season of a show that has been renewed on AMC. I've referred to it a couple times. It has ended up being so good. It gets top recommendation, and that's "Humans" on...

Leo: "Humans."

Steve: "Humans." It's about things that are not. It's sort of a near future, that is, they're not trying to pretend that all of their cars are like whining with a turbo-sound engine, that they're just normal. It's a British production. But it's in a world where there are Synths, synthetic humans, which are produced as sort of helpers, you know, clerical workers and crossing guards and things. And they're not much fun at a party. Put it that way. They're kind of flat. They have no affect. Except there are some that are different.

Leo: Uh-oh.

Steve: And it has been a really good first season. So for anyone who is looking for a series that's worthwhile, I imagine you can find them online. Probably AMC will allow you to stream the back episodes. It's a slow burn. It's not like some ripping, roaring action. I wasn't sure if it was going to really evolve. But I've ended up being very satisfied with it, so I did want to give everybody a heads-up there, following sci-fi as I do.

Leo: I love our chatroom. "I prefer the original Swedish version, 'kta mnniskor,' from 2012. If you really like this, you should probably watch that."

Steve: Is it in English? Because I don't do subtitles.

Leo: No. "Humans." And then "Mr. Robot"? Did you decide? Good or bad?

Steve: Oh, I'm - it's interesting. Many people were put off by the druggy-ness of it.

Leo: Yeah.

Steve: I just think it's very good. I mean, I enjoy it.

Leo: "I finally started watching "Sense8," which you recommended.

Steve: Yes.

Leo: And that's really lush, gorgeously shot, just beautiful.

Steve: Yes. Yeah. And, I mean, you're right, the photography is incredible.

Leo: Major motion - it's a Netflix original. It's major motion picture quality. It's amazing.

Steve: Yup. And did get renewed.

Leo: Oh, good. I'm not through with it yet.

Steve: For what it's worth, this second season of "True Detective" was really something, too. Wow.

Leo: Did you like it?

Steve: Oh. Well, it was dark.

Leo: A lot of people hated it.

Steve: I just - I didn't mind it. I thought it was, I mean, I get why people wouldn't - I'm not recommending it for everyone. But I just, you know...

Leo: They liked the first season, beautifully shot, fascinating acting, you know, really something to watch.

Steve: Real characterization.

Leo: But the worst plotting I've ever seen.

Steve: Yeah.

Leo: And it's like Nick Pizzolatto, who writes these, I really feel like he needs some work.

Steve: Yeah.

Leo: Needs some plotting help.

Steve: Well, I got a nice note from Michael Coyne in Raleigh, Essex, U.K. Actually, this is not just a testimonial. This actually is a question sort of about SpinRite's ECC readout. He said: "Dear Steve. I am no propeller head, but I really stay up on the security side of things, and I'm so glad I stumbled upon Security Now! about five years ago.

"SpinRite increased the ECC maximum number on my old 250GB computer by about five points over several years. And it says on one of the GRC SpinRite pages a 100% healthy drive shows no red. My old computer was certainly healthy in that respect. My 2TB drive isn't quite as healthy, as I always see some red blocks on the ECC readout while SpinRite is working on the drive. And the seek error maximum has increased three points in the six months I've owned the computer. Best wishes, Michael Coyne."

So I just wanted to respond to Michael's statement, essentially, to explain that, well, unfortunately, as I've mentioned, drives are beginning to rely much more heavily on error correction than they were before. What is unique about SpinRite is that the SMART data only reflects trouble the drive is having. And it only is going to have trouble when you're asking it to do something. So there are, like, there are lots of SMART monitors around that will show you what the drive's parameters are.

But I would argue, and do, that only when you are putting the drive under stress and making it do lots of seeks and making it do lots of reads, that then the ECC reading actually has some meaning because SpinRite shows it to you in real time. Is the drive correcting more than it thinks it should? Those red dots are the health rating from the drive itself, which is different than the raw error correction count which SpinRite also shows.

But here, as those red blocks appear, this is the drive saying, ooh, I'm correcting more than I'm thinking I should be. So you don't normally ever see those when the drive is operating. It's only under SpinRite, where we're just saying read, read, read, read, read, read, read, and the drive is beginning to say, oh, I'm seeing a little more error correction than my own designers thought was healthy for me. So it is something worthwhile keeping an eye on over time because, as drives get older, they can have increasing problems reading back what they wrote before. And again, only SpinRite shows it to you.

Leo: It's a good point, that you've got to stress it before it'll show information about itself.

Steve: Right, right. Just like us.

Leo: Just like us.

Steve: Just like people.

Leo: Most of the time we sleep.

Steve: Okay. So finally, the quest for surfing safety. Everybody knows that I've been a huge proponent of just say no to script, to JavaScript, to scripting; that, unfortunately, scripting lies at the heart of both the functionality of the web we're using today and sort of the anti-functionality, the malicious functionality of all the vulnerabilities that we see. Even when the actual problem is in Flash or in Java, it's scripting which is invoking these problems, typically. Not necessarily, but it generally is the way it happens. Or it's scripting itself that is the problem. So the point is that disabling scripting can, well, has been demonstrated to be a far safer way to function.

As I mentioned last week, I've given up. I'm spending so much time permitting scripting of all the crazy number of assets that sites are beginning to require, it used to be one or two different places that a site would also need to have scripting enabled. Now it's 30. And, you know, and NoScript just shows this long list of things. And you enable a few things, and then the list grows because those things that you've enabled are now wanting to use their own, and technically it's fourth-party sources that they're relying on. So then the list expands. And finally it's like, okay, forget it. This is, I mean, I need a different solution.

So then I thought, okay, what about going back to Sandboxie? The problem is I was brought up a little bit short by this vulnerability that I talked about this week in Firefox because this was a script, okay, so it's true, if I were still using NoScript, and I was on that Russian site that was trying to serve me a Russian ad that was using scripting, NoScript would have protected me. But if scripting were enabled, Sandboxie would not, because what Sandboxie does is it allows reads from the system, assuming that those are safe. But it sandboxes writes, so that any writing, any modification to the OS will go into the sandbox, where they are subsequently read from, assuming that you need some interaction, but the writes do not get pushed out of the sandbox to the OS. Well, obviously, if you had some evil script that was exfiltrating files on your system, Sandboxie doesn't solve the problem.

So I think the solution for safe surfing is for us collectively, us the Security Now! community - and there is a community. We have a community of listeners. We've got people submitting all kinds of ideas and questions through the Security Now! mailbag. And of course I have an active Twitter following that is tweeting me lots of stuff going on. I think what we need to find is a small and lightweight OS that can run in a virtual machine where we can run the browser of our choice. I choose Firefox. Some people are choosing Chrome. And I think...

Leo: There is a Firefox OS; isn't there?

Steve: Yeah...

Leo: Like there's a Chrome OS?

Steve: Well, but those are standalone.

Leo: Ah.

Steve: Now, maybe the Firefox OS would run very small in a VM. That's certainly

something to look at. And I've invested no time in this because my focus is full. My full-time focus when I'm not producing the podcast is on SQRL, and we've had - I haven't talked about SQRL for a couple weeks, but there's been some really great forward motion. About several months ago I did mention that, during a bunch of brainstorming, I came up with a kernel of an idea that we collectively honed into something really workable in the GRC newsgroup and have come up with a solution that makes SQRL absolutely and completely unspoofable, which was one of the early problems that was recognized, was there was some concern about SQRL's spoofability. And we'll get into that sometime when we're talking about the technology of it.

But anyway, my focus is still there. But I found something called Tiny Core Linux, which is as small as about 15MB. And it looks like it runs Firefox. And so if I could be running my web browser in a VM, then I think that's the right way to go. Then it could be unfettered. I wouldn't worry about ads.

Leo: It's the ultimate sandbox.

Steve: Yeah. Oh, exactly, the ultimate sandbox. It can't see my OS. It can't write. It is, it's an OS running with full isolation.

Leo: And you refresh it each time; right? So it starts from scratch each time.

Steve: Correct. Correct.

Leo: Yeah. So that way all cookies are blasted. Supercookies are meaningless.

Steve: Yup. Yup. I think that's the way to go.

Leo: And then your IP address won't change.

Steve: No, but IP address is real soft tracking material. It's not, I mean, it's true that it is relatively static publicly. But, I mean, I'm more concerned - I'm much less concerned about tracking than I am about malicious content. So, I mean, that's why I was running with NoScript, was it was never ads I wanted to block.

Leo: Right.

Steve: It was scripting that was just too prone to problems. And I think that's what we need. So, you know, browsers...

Leo: Here's another something the chatroom's passed along, MenuetOS, M-E-N-U-E-T-O-S.

Steve: Yup. I know about Menuet.

Leo: It's not a Linux, but it is a UNIX - actually, no, it's not based on POSIX, it's its own thing.

Steve: Yeah. The question is, does it have a state-of-the-art browser?

Leo: Yeah.

Steve: We need it to be able to run, really, either Chrome or Firefox. I mean, I don't want to run IE, and there's no reason to when we have something that's much more feature-rich and plugin-able.

Leo: It's showing a browser on a screenshot, but who knows what it is. I'll have to...

Steve: Yeah, we need - so anyway, I'll put it out to my listeners because we've got super-capable listeners. Let's think about, sort of as a communal project, a small VM, because we don't want to burn up RAM, that all it has to do is run a browser, Firefox or Chrome. Probably, you know, I think it ought to be Linux because I think Linux is going to be smaller than anything else. And Chrome and Firefox are both available for Linux. So I think that's the right solution.

Leo: Here's another one. The chatroom's been great on this one. TAILS. It's called The Amnesiac Incognito Live System, T-A-I-L-S.

Steve: That all sounds good.

Leo: It's a live operating system, runs from DVD, USB, or SD. I'll have to see what - I don't know what it is. Oh, it's Debian, so you could run Chromium on it, then.

Steve: Yeah, now, there again, there you'd be rebooting your system to run it.

Leo: Right, because it runs through Tor.

Steve: Yeah. So we really want something that is small, that runs in a VM. That, like, will run in, like, VMware, or what's the free one? I'm blanking on the name.

Leo: Yeah, I always get it wrong.

Steve: Anyway, everybody knows.

Leo: I always say OneDrive. It's the one from, sad to say, Oracle.

Steve: Yeah, exactly.

Leo: Yeah.

Steve: Yeah. But hopefully they haven't messed it up. Well, we are at the end of our 520th episode.

Leo: Wow. Wowie, wowie, wowie.

Steve: Yeah, we have done 10 years. Our first episode, Episode 1, was - and I'll forever regret not numbering Episode 0. But anyway...

Leo: I'm sure we did an - there was an Episode 0 of TWiT, you know. So I'm sure there was an Episode 0 in some way. We probably recorded it on the set.

Steve: Well, it was August, yup, August 19th, 2005. And here we are on the 11th. So next week will be the 18th, meaning that this really is, this is the end of Year 10. And we'll be starting into Year 11 right on the episode number that we would expect, on 521. So I want to, again, thank you, Leo, for making this possible for me, and our listeners for creating just a great podcast. I love the interaction that we have now with the mailbag and with Twitter. I feel a real sense of community. And, wow, it's been 10 years.

Leo: We looked a lot younger back then.

Steve: Oh, my lord.

Leo: Actually, this is the old Screen Savers show. But same idea.

Steve: Yeah, yeah, yeah.

Leo: Ten great years, Steve. I really cherish our friendship and value the show. It's easily the geekiest show we do on TWiT.

Steve: Yeah.

Leo: Most hardcore. And, you know, not a week goes by where somebody, including me, doesn't say, got to find out what Steve Gibson thinks about that.

Steve: Well, and I do know from Jeff, Jeff during New Year's, I was chatting with him on the New Year's Eve show, and he said that for the last couple years at least we've seen constant growth.

Leo: Yeah.

Steve: That Security Now!'s been a steadily growing podcast.

Leo: It's done very, very well, yeah. I think it went up, what was it, 18% this year. And it's just, yeah, very steady growth. As it should be because the world's getting more dangerous, and we need you more.

Steve: Well, and it means that, while we are going to see some churn of our listeners, in general people are finding value here, and they're staying for more.

Leo: Oh, yeah. Oh, yeah.

Steve: So again, I really thank our listeners for their support and loyalty.

Leo: Here's to 10 more; okay?

Steve: Absolutely. I'm ready.

Leo: After that, all bets are off.

Steve: I don't think we're going to run out of things to talk about. There's no sign of that happening.

Leo: Great to have you. We do Security Now! Tuesdays. If you've got to know, what would Steve say about this, you've got to tune in Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC on TWiT.tv. You can also pick up episodes at GRC.com. That's Steve's website. That's where you'll find SpinRite, the world's finest hard drive maintenance and recovery utility. You'll also find lots of freebies, lots of information, 16Kb versions of the show if you're really bandwidth-impaired. Great transcripts, show notes, everything's there. And that's the place to go to ask questions because we might be able to get a question and an answer episode in next week. I don't know. Crossing our fingers.

Steve: I think. Let's hope.

Leo: Go to GRC.com/feedback, or you can query Steve on Twitter. His Twitter

handle is @SGgrc.

Steve: I noticed that my Twitter followers, I hadn't looked at my count for a while. It's getting close, it's at 49 something.

Leo: That's good.

Steve: So it's looking to be 50,000.

Leo: That's pretty significant, yeah.

Steve: So getting there, yeah.

Leo: In fact, I think we could give it a little boost. If everybody who listens to this show would now go follow @SGgrc, I think we can get it into the six figures here.

Steve: I meant to say that on Saturday, I forgot, on The New Screen Savers, because that's a somewhat different audience.

Leo: Yes.

Steve: And so that would be the key because I'm sure everyone who is listening to this who's also on Twitter is already following me.

Leo: I would hope so.

Steve: Yeah, well, I mean, why not?

Leo: Why not? All right, Steve. Have a great week. Congratulations on 10.

Steve: Thank you, my friend.

Leo: Ten more to come.

Steve: Hey, congratulations to us. You've built, you know, you took a dream, and you've built a network. I mean, something really substantial, Leo.

Leo: It's a lot of fun.

Steve: So good going.

Leo: Thanks, Steve.

Steve: Thanks, buddy.