Extrinsic Password Managers

Description: This week I want to start off with a calm rant to summarize why today's computer security is so atrocious. I think it's worth a bit of a reality check on that. Then we're going to look at a new feature in Firefox and at Firefox's apparent jump in performance. We'll touch on three new ransomware victims, look at what's been learned about how Colonial Pipeline was breached, and at the curious news that the FBI somehow managed to snatch all of DarkSide's bitcoins. We'll look at the latest good and bad news regarding WordPress, and at GitHub's updated policy regarding posting proofs of concept for ongoing attacks. I've finished Project Hail Mary, so I have a comment to make there, and I want to address the surprisingly controversial question of NAT versus IPv6. Then we'll wrap up by examining the question of whether password managers should be intrinsic to our browsers or extrinsic. I think we're going to have some fun.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-822.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-822-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with the great cybersecurity awakening of 2021. He says solving ransomware, it's as easy as fixing your software. We'll also talk about what happened to the bitcoin that was headed for the Colonial Pipeline hackers, and why Firefox is so much faster. Plus a look at Tavis Ormandy of Google's suggestion that you stop using a password manager. Really? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 822, recorded Tuesday, June 8th, 2021: Extrinsic Password Managers.

It's time for Security Now!, the show where we cover your security online, your privacy, your safety, and a little bit about how things work with this guy here who knows, Mr. Steve Gibson. Hi, Steve.

Steve Gibson: Just a little bit about how things work. Often how they don't work, actually.

Leo: Well, yeah, half the time.

Steve: On this podcast.

Leo: You know, I noted that this is the 42nd anniversary of the release of the 8086. And I thought I'd just mention that to you.

Steve: Ah, yeah.

Leo: The beginning of the x86 architecture, which it turns out the designer never thought would last. He said, "Had I known we'd still be using it 42 years later, I wouldn't have done segmented memory. I wouldn't have done a lot of things."

Steve: It was, for its time, I think it was the right chip. I mean, segmented memory, I'm in the middle of it now because everyone knows I'm in the process of working on the next SpinRite, and it's always used a segmented memory model. And it is a pain from a - well, of course I'm writing it all in assembler, too, which doesn't help because I have to do all this by hand.

Leo: You have to manage it, yeah.

Steve: Yeah. But, I mean, it's a pain to deal with it. But in terms of efficiency, if you assign objects to segments, then the segments become pointers to objects, and all of the pointers within the object are zero-based instead of being some random offset.

Leo: Oh, so that's kind of cool.

Steve: It is. It is. There's a lot to be said for it, and a lot to be said actually for the x86, once you get to know it. Someone posted somewhere, I saw it, like the 10 weirdest x86 instructions. And, oh, there are some strange things. I mean, where you've got to scratch your head and think, okay, if I had to use it, that instruction, what could I possible use it for? I mean, they're a little bizarre.

But anyway, this is Episode 822 for June 8th. And as you immediately grokked when you saw the title of this episode, "Extrinsic Password Managers," a whole bunch of our listeners regaled me with tweets and emails saying, "Hey, Tavis" - of course famous Tavis Ormandy of Google - "just talked about password managers, and he thinks they're bad. So what do you think?" And it's like, okay, we've got to talk about that.

But I want to start off with what I'm going to try to keep as a calm rant.

Leo: Oh, boy.

Steve: I'm just going to try, I'm going to try to not let myself get worked up, although it really does kind of wind my spring, summarizing why today's computer security is so atrocious. I think it's worth a bit of a reality check on that because it's so easy just to kind of take it for granted and go, oh, that's the way things are. So I just, as a consequence of the past week, where I've been having to listen to all of the popular press talk about Colonial Pipeline and the meatpacking system and ransomware and, you know, how we're going to fix this, I just thought, okay, timeout. So we're going to do that first. And then I think we're going to take our second break.

Then we're going to look at a new feature in Firefox, at Firefox's apparent jump in performance, which I can't believe I'm experiencing to the degree I am. We'll touch on three new ransomware victims, just naming them, I'm not going to spend time there; look at what's been learned about how Colonial Pipeline was breached; and at the curious news that the FBI somehow managed to snatch all of DarkSide's bitcoins. I mean, there's been a lot of strange things going on the last week.

We'll look at the latest good and bad news regarding WordPress and at GitHub's updated policy regarding posting proofs of concept for ongoing attacks. I finished the "Project Hail Mary" book, so I have a comment to make about it there. Of course no spoilers. And I want to address the surprisingly controversial question, which I stepped into unwittingly, of NAT versus IPv6. Then we'll wrap up by examining the question of whether password managers should be intrinsic to our browsers or extrinsic. I think we're going to have a lot of fun this week. And of course we've got a great picture.

Leo: Great topic, a lot of them. And I have some thoughts on the password manager thing, as well, because I think Tavis is a security guru and so forth; but I think he's also, as is often the case with these people, a little bit of a purist in his point of view. So I'm curious to see what you have to say about it, as well.

Steve: So our Picture of the Week is just - it's one from my archive that I get from our listeners. It's six frames of a cartoon. I got a kick out the fact that written vertically off down to the lower left it says "Giffs not jiffs."

Leo: I'm with you on that. I'm with you.

Steve: Okay. So the first frame we've got this guy sitting in front of his laptop, and he's thinking to himself, "Hmm, this article looks like it might be interesting." So in the next frame he's showing a little bit of surprise. He suddenly has eyebrows that are lifted, and clearly the screen is showing him, "Accept cookies?" And so, yeah, okay, fine. And then the next frame he's a little concerned. One eyebrow has dropped, and he's looking at "Disable your adblocker," which he's being told to do. Then the next frame, he's not looking happy now. He's kind of got - both eyebrows are down, and it says, "Sign up for our newsletter!" in another dialog box. And finally, in the second to the last frame he's like squinting and wincing because he's now been confronted with "Allow notifications from this site." And in the final frame he's turned around, and he's walking away thinking, "*Pft!* Not THAT interesting."

Leo: Yeah, I've been there, done that.

Steve: Oh, my god, what we're all being confronted with these days is just, you know. It was inevitable, Leo, the commercialization of what was once a fun playground for the techie nerds. Now it's big business.

Okay. So listening to the last week of the press talking about, I mean, like with no understanding of the ransomware problem, I just thought, okay. I'm going to try to lump all of the problems into one relatively short presentation. I titled it "The Great Cybersecurity Awakening of 2021."

Leo: Yeah.

Steve: Because, you know, okay, it's good, right, that the rest of the world is where we've been. We've been talking about this problem specifically, ransomware, for so long that I have repeatedly attempted to promise that I would stop talking about it. But it just keeps coming up.

Leo: Well, you can't now. I mean, it's become the cause clbre in security.

Steve: Right.

Leo: It's all anybody's talking about.

Steve: Well, and in fact last Monday Lorrie and I attended a small dinner gathering. And as the resident computer security guy, I found myself attempting to explain why most of what we were hearing about how this or that group was going to be appointed or created to get to the bottom of this was wrong and impossible, but understandable because no one wants to hear or believe that there is no simple fix for this - or, more truthfully, no fix at all, simple or otherwise. 87.8% - this is where I'm trying to keep myself calm - 87.8% of the world's desktop and laptop machines are running an operating system which is so riddled with bugs that they needed to stop releasing them as they became ready, or no one would have been able to get any work done.

So now they're clumping them up into monthly releases of between 50 and 150, where every month a handful are rated critical, and patch first with highest priority, because those are known to be currently in use, compromising unwitting people every month, every single month, month after month, with no diminishment or apparent end. And as we all know, Microsoft was informed of a horribly serious vulnerability in their Exchange Server product, in all of them forever through time, late last year. Yet it took until March for them to produce a patch, which they then did a week before they were planning to as an emergency because they believed news of this oh so juicy latest in a never-ending line of vulnerabilities had somehow gotten loose.

And of course it's not just Microsoft. Many of the largest players have now synchronized their monthly vulnerability patch release cycles to Microsoft's. So a monthly patch fest has evolved because we are apparently unable to produce software without serious exploitable flaws. And just last week, we noted how the latest ransomware, which was mostly just a handful of PowerShell scripts, was getting into people's computers when they clicked on a link in a perfectly authentic-looking and specifically targeted email that loaded and rendered an HTML page which contained and ran some JavaScript which secretly downloaded some malware and then politely explained to its user that they needed to click once more to open the document. Whose fault is that, exactly?

Our email clients finally, but only after a long siege of exploitation, have started to actively refuse to have anything to do with executable content. Thank goodness for that. But of course there's a workaround. Since we want the power and flexibility of having web apps, and since a web app might need to save something to our local computer, we've given JavaScript the ability to do so. Repeat after me: What could possibly go wrong?

So now the bad guys bootstrap. A benign-looking email loads up an HTML page containing a little hidden benign-looking bit of JavaScript. And when given permission by

its unwitting user, it downloads the executable file on behalf of the email that was originally received. How do we robustly solve that problem within the framework of computing that we've been making up as we go along? I have no idea because though no one wants to hear it, all of our systems are fundamentally and deeply broken. Why? Because the economics are all wrong. And those economics create perverse incentives. We reward new features because those can be seen. But security is invisible. No one gets credit for making something more secure because you can't prove a negative.

Before the release of Windows XP, Microsoft's Steve Ballmer famously jumped around onstage declaring that XP would be the most secure operating system they had ever created. History shows that it was the worst by far. But Ballmer's assertion was unchallengeable at the time because, as we've often observed, true security is only proven over time when, as, and if that unprovable negative is gradually proven by never happening. So how the heck do you reward that?

So we have some problems which are due to errors appearing in systems that are so complex that we've become tangled up in our own code. All we can do there is frantically patch and patch and patch all of the mistakes that we've made as we find them. And separately, we have problems like the email to JavaScript malware download bypass which arise from the abuse of the deliberate design of our systems. It's not a bug, it's a feature. Yet it's costing companies many millions of dollars in lost revenue, lost reputation, and ransom payments. Everyone wants to assign blame somewhere. Out of sheer desperation, enterprises are now sending their own employees baited emails, trying to trick them into clicking on a link that they shouldn't in order to pounce on them. Aha, your prior training has apparently worn off. Back to the reeducation camp with you.

What is wrong with us that we are now blaming the user for behaving in a perfectly normal and understandable fashion by responding to an email that appears to be in every way perfectly legitimate? We would like to shoot the messenger, but we can't find them. So we're going to shoot the recipient instead. We have to shoot somebody, apparently.

As for features versus bugs, we were once assured that Windows 10 would be the last Windows ever. Someone somewhere decided that change was bad for security. And they were right, of course. Now we hear rumors of Windows 11. Oh, joy. That's what we need. Apparently Windows is going to get lovely rounded corners for its rectangles to distract us from the minor detail that all of our desktop's icons have just disappeared. We're all so terrified now to click a link in an email that perhaps not having those sharp pointy corners will calm us down a bit.

During that dinner party, where everyone lost their appetite and stopped eating, I pointed at an AC wall plug that's controlled by a cloud-based service. I explained that the plug was connected back to servers in China, and that the software running inside that itty-bitty computer contained in the plug was known to have a handful of remotely exploitable vulnerabilities such that, if at any point someone in China wished to infiltrate the house's network to snoop around, it could be done.

And I noted that the exploitation of a vulnerability in the plug's firmware would only be necessary if the plug had not come preloaded with a deliberate backdoor, which would simply open when asked to allow foreign access. How would we know? There's no certification process. There's no qualification process. We click "Buy Now" on Amazon, and that little miracle is on our front doorstep the next day. It may have a UL Seal of Approval to attest that we won't be electrocuted when we plug it in, but that does nothing to regulate the foreign packets that flow right back from our household's internal networks to China, a country with whom the U.S. has a very complex relationship. Today, we are frantically deploying millions, if not billions, of Internet of Things devices throughout our lives because they are shiny, incredibly inexpensive, and do neat stuff.

But there's zero oversight anywhere in the design, implementation, and delivery of these devices. They're cute little time bombs just waiting to go off.

Accountability is another problem in the computer industry. If your car's brakes fail unexpectedly, or its wheels fall off when you take a corner, there are consequences for its manufacturer. They're accountable. Software companies are not. There are no consequences for Microsoft when they wait three months before patching a horrific vulnerability that had been demonstrated to them, the exploitation of which has without any doubt been incredibly expensive for their users. But not for Microsoft. Microsoft requires everyone touching its obviously defective software to explicitly waive all expectations of their software's performance or fitness. It's in every license agreement. They say it may work; it may not. We don't know, either. But either way, the risk is all yours because we did our best, and everyone knows that software, despite its name, is hard.

So what have the natural consequences of all this wrought? We have insecure hardware, processors, and memory running insecurely designed software, written in insecure languages, implementing insecure protocols and APIs, by people who require no formal training or certifications of any kind to create any of this. It's a black box in a black box in a black box. But it's also one other thing that we apparently prize more than anything and everything else. It is astonishingly inexpensive. That Chinese wall plug has been working flawlessly for a year, and it cost $5. I love it.

You can now purchase a breathtakingly powerful and useful laptop with a free Windows operating system, soon to have rounded corners, for a few hundred dollars. And storage, oh my god. And what operating system am I sitting in front of right now as I write this? Yes, the same, 87.8% of the world, Windows. I'm completely satisfied with it. It works great. Overall, what it manages to do is a miracle. Okay, so yes, last week when I fired it up, actually the Win10 box, to record this podcast, all of the icons had disappeared from the desktop. I waited a while to see whether they would reappear by themselves, and when they didn't, I rebooted. Now they're back.

Since most people have a very limited understanding of how their own bodies work, we rely upon highly trained medical professionals who have obtained extra education and certification. The system was designed to remove all possible sources of error. So the resulting medical care is astonishing. It can also be astonishingly expensive. And no one who has not been trained in the law should attempt to write a complex legally binding contract. But sometimes we need one. So we train up attorneys in the law. We require them to prove that they know how to write properly complex contracts by passing the bar. But now that contract will really cost you. But software, which may have been written by someone in his mother's basement, hey, it's free. And it's worth what you pay for it.

So throughout this little rant, I've attempted to touch on a number of points. Looping back to the original issue, all of a sudden people are saying that now they want security. But that ship has sailed, and it sunk. Much as we might wish we could, we can't just dust off Steve Ballmer and have him jump around onstage to declare today's problem solved. You can't slap a fresh coat of paint on top of the rickety and flaky computer systems and technology we have collectively and deliberately built, placing features before security, and expect to suddenly have any actual security. And more importantly, no one, no one would actually be willing to pay the cost to obtain true security, even if we knew how. And there's no reason to believe we do. The systems we have are not secure, and at this rate they're never going to be. But my god, are they inexpensive. And soon they're going to have really nice rounded corners.

Leo: I wonder if it's possible, though, I mean, if you started from scratch, to design a secure operating system. Aren't hackers so determined, and there's so much money to be made in exploiting them, is it possible to have secure software?

Steve: I would say it's possible to have secure software that doesn't do much.

Leo: Okay, that's fair. Yeah, yeah. But a general purpose operating system, could you do that?

Steve: Yes. Well, for example, the original iPhone, the problem was...

Leo: Didn't do much.

Steve: ...it didn't do much.

Leo: Didn't even have cut and paste.

Steve: No. You had - the home screen wasn't even full. It looked like there were things missing because there were only icons kind of in the upper half and half of the last line. And you were looking at it thinking, okay. I thought Jobs was a real pain to work with. How did he let this out of the barn? And so, but damn, it was secure because it was all from one place, and it didn't do much. My example of the malware downloaded when you click email, the problem is we want web apps which are able to save files on our computer. It wouldn't be much use if you could look at it, but you couldn't get at it. It's like, wow, this document looks great. Wish I could do something with it.

Leo: Right.

Steve: And that's the problem is that we breach the sandbox of the web browser deliberately because we want the feature of allowing a document to be downloaded to our desktop. And so a properly, a cleverly designed bit of JavaScript convinces a perfectly normal, sane, you know, this poor employee has been through several rounds of security awareness training, and they get an email from their boss from Panama, who happens to be in Panama right now, saying, "Hey, Sally, following up on what I asked you to do, here's the report you've been waiting for." Which is probably exactly what Sally's been waiting for. So she says, "Oh, good," and clicks the button. And now she's just infected her entire company with laterally spreading ransomware and the screen goes black. We see it in movies all the time. Unfortunately, these chickens have come home to roost.

And so the real thing that has happened is that, as you say, Leo, because of the incentive, which crypto currency leveraged with ransomware has created, I've long been talking about security as porous; right? It isn't perfect. The harder you press on it, if you press hard enough, you will squeeze some data through a surface that's trying to resist that. It's the reality of what we have today. And so, yeah, if we were happy with the original iPhone and its half screen full of icons, that's a secure thing. But you can't do much with it. You can't do what you want to. Same thing for web apps. If they can't touch the computer, then they're not that useful.

Leo: So, I mean, somebody asked on the radio show this weekend, well, what do we do about all this? And I pointed out that security is layered. There's no one layer that's going to fix all of this. Certainly we could make more secure operating systems. I don't think we'd ever make a perfect operating system. If you're going to connect to the Internet, you're going to have an attack surface.

Steve: And Leo, remember that Windows was never meant for the Internet.

Leo: Right.

Steve: It was meant for a modem.

Leo: Why is why, by the way, our mobile operating systems are more secure, because they knew those would be under attack. So they are marginally more secure. In fact, I think actually you can make a pretty good case that iOS is - there's malware out there, but it's fairly secure.

Steve: Yes.

Leo: So I guess it's possible to make something more secure if you really give your mind to it. And then of course companies need to do a better job of training employees. They need to do a better job of defending their own perimeter. There's a lot of things companies can do. Governments need to do a better job of putting pressure on rogue states to knock it off. It's kind of a mess. But I think that it's going to take a lot of different efforts in a lot of different arenas to fix this. It's no one thing. You're right, I mean, Windows is horrific.

Steve: Well, and the other gotcha with security is it is the classic chain; right? The weakest link in the chain.

Leo: Right, that's right.

Steve: To have security, every single link in a chain of 100 links - which 100 different groups and organizations and people all contributed their own link to, saying, oh, yeah, yeah, my link is really secure, don't worry about it - every single link has to be secure because, as you said, there's now a tremendous incentive to find the weak one. And you just pull hard enough on both ends, and that chain will find the weak link for you.

Leo: Yeah. So is there any hope? Or is it just going to continue to get worse and worse and worse?

Steve: I really do think that three digits is not enough numbering for this podcast.

Leo: I've been saying that for a while, Steve. I just want you to know. You can't stop.

Steve: We're at 822, and time is running out.

Leo: Yeah. I don't think we're going to fix anything by 999, that's for sure.

Steve: I don't think so. I'll do a little bit more news before we take our second break. Firefox will soon auto-update on Windows, even when it's not running, which I thought was interesting. This new feature will be in, well, it is in the Firefox 90 beta. Everyone is now running on 89. So the next major release should bring, when 90 goes to mainstream, we ought to all have that. Kirk Steuber, who's Mozilla's Platform Engineer, said: "Until now, Firefox has only downloaded and installed updates when the user runs it." And in my experience, not even then. You've got to go poke it. You've got to go do About Firefox, and it goes, oh, hold on a second, and then spins its little widget for a while, and then lets you restart it.

He says: "This means that users who only use Firefox infrequently may well be out-of-date. It also means that if they open Firefox again in response to a Firefox marketing campaign" - like to see some new feature - "they may not immediately get the features which have been advertised." So he says: "Background Update aims to address this problem by allowing updates to be downloaded and installed, even when the user is not running Firefox."

So by default there will be a ping which looks for an update every seven hours, when the browser's not running, to check for new updates. I thought that was interesting. Had it been eight, then it would fall on the same three hours every 24-hour cycle. But seven being odd and prime and not a multiple of 24, it means that it's going to kind of be roaming around, and probably everybody's browser will be doing it at different times, so that's kind of cool. Anyway, so every seven hours it'll check to see if there's anything new. And if so, it will get it. It also installs a little service, I think I had it here somewhere - oh, the Mozilla Update Service - to bypass the Windows User Account Control since services are able to run with system privileges. And this allows it to get the data from Mozilla in the background to update itself.

So anyone who wants to disable this, if you don't like it, you can go to about:preferences, look for updates, and then fuss with the UI which is presented there. And for the time being this only applies to Windows. They've announced no intention to do this either for macOS or Linux. So Firefox will be getting background updates by default with its next major release. And as I mentioned before, speaking of Firefox, my podcast prep workflow is to assemble a bunch of - basically assemble a collection of many stories of the week from a number of different sources. And I build them in Firefox using its left-hand vertical column of tabs which, you know, it doesn't have natively, but I use tree-style tabs. Then I open Chrome to edit the show notes in a Chrome Doc, and that's what produces the PDF every week. I also open the little desktop outliner that I've been using ever since the Palm Pilot because it had a desktop version where I do most of the writing.

So I'm bouncing around among all three tools. And yesterday evening, after updating the instance of Firefox to 89, which as I said, it didn't do by itself, but I looked, and it says, oh, yeah, hold on a sec. And this is the one that we talked about last week, and you got it while we were on the air, Leo, where the UI got kind of cleaned up and polished.

Leo: Mm-hmm, mm-hmm.

Steve: I am consistently noting that it is running far faster than it ever has. I mean, I wasn't expecting it to. I didn't think about it. But I'm so used to this workflow that as I was just, like, clicking on tabs and deleting them as I was covering topics, it was just like, wow. This is a lot faster. And we know that what it did was it broke everything down in individual processes, so that may be part of it. But I'm just saying, for what it's worth, if you haven't looked at Firefox in a while, after you start it, let it settle down, then go into About Firefox to give it a little kick in the butt to make sure that you get 89, and see what you think because, I mean, I'm very impressed with how much, I mean, it's viscerally faster.

Again, I wasn't looking for a speed increase. And it wasn't even the first time I noticed it. It wasn't until the third or fourth time that it just did something instantly that I was used to, like, waiting for. And it's like, wow, this is getting faster. So, very cool.

Also, Edge is taking its own approach to HTTPS switching. As we know, Google's Chrome now finally defaults to HTTPS when given a so-called "schemeless" URL, like TWiT.tv, GRC.com. And Firefox has also added an HTTPS-only mode designed to secure web browsing by rewriting URLs to use the HTTPS protocol. At the moment, in the case of Firefox, it's still disabled by default. Google had been performing experiments before they went with it, and they now have. But hopefully Mozilla will catch up. Until then, for Firefox, it can be enabled from the browser settings. But of course it's not our listeners who would be enabling it who are the most endangered by it not being on yet by default. It's everybody else.

In the case of Edge, here's what Microsoft has done. They've kind of gone their own way. They say: "Starting with Microsoft Edge 92, users can preview the Automatic HTTPS" - which is their name for it, Automatic with a capital A, HTTPS - "feature, which automatically switches your connections to websites from HTTP to HTTPS." They said: "When sites are loaded over HTTP" - and this we all know, but there's a little bit in here, that's good, to set the context - "attackers can view or change page content in transit, or redirect you to a different location than you expected.

"Most websites now support HTTPS, which can help protect against these man-in-the-middle attacks. However, too many of these sites aren't configured to require HTTPS, leaving open a short window of opportunity for attackers before the site can redirect to the more secure protocol. Some sites may not redirect visits from HTTP to HTTPS at all, leaving some visitors with a less secure connection. To help protect your information as you browse, we are introducing a feature called Automatic HTTPS" - oh my god, yes, you're the last to do it, but good - "now available for preview in Canary and Developer channels with Microsoft Edge 92.

"Automatic HTTPS switches" - here's the curious part - "switches your connections to websites from HTTP to HTTPS on sites that are highly likely" - what? - "to support the more secure protocol. The list of HTTPS-capable websites is based on Microsoft's analysis of the web, and helps enable a more secure connection on hundreds of thousands of top domains. Automatic HTTPS upgrades" - thank god they didn't try to put a trademark on that - "upgrades your connection only on HTTPS-capable domains by default in order to prevent connection errors and potential performance issues."

So, okay. Now, as we know, many sites like GRC and TWiT have been publishing a strict transport security header for some time. I've got a max age of one year. It's 31536000, probably seconds, which I'm saying after a browser encounters that from my server it can sticky remember that for one year. TWiT's max age is better, Leo. Yours is 604800000...

Leo: How long is that?

Steve: ...which is almost two years.

Leo: Wow.

Steve: And that's actually - that means that you did it more recently. It used to be that one year was the maximum, back in the beginning when I implemented it. Your guys waited longer and so are more current because it now should be longer. So I'm going to have to update mine. I learned all this this morning. It's like, hey, TWiT's got it right.

Leo: Well, we switched later than most because we switched only when we had to.

Steve: Right.

Leo: But credit Patrick Delahanty for doing it right, and maybe Russell had something to do with it, too. But, yeah, they know what they're doing. They're good. We have good engineering.

Steve: So Microsoft says: "This protocol switch process happens automatically and without intrusive notifications, so that you don't have to think about your connection to websites. Simply browse as usual. If you'd like even tighter security and don't mind potentially encountering connection errors more frequently, you can in Edge opt to switch all navigations from HTTP to HTTPS using the toggle at edge://settings/privacy." And they said: "If you want to test it right now, you have to open edge://settings/privacy and turn on 'Automatically switch to more secure connections with Automatic HTTPS.'" And so forth.

So anyway, they're catching up. In a couple releases, as soon as it gets out to the main channel, it'll be available. They've sort of taken the cautious approach of doing something, I mean, probably they've, you know, they've got Bing; right? So they're spidering the 'Net. And from all of the servers like mine and TWiT's, they're seeing our formal declaration that, yes, we will absolutely be supporting TLS from now until a year or two from when you ask. So that's probably what they've done is they've just basically incorporated that information. Maybe they've done offline tests of both to see, although that could be a little sketchy because it's not necessary for a site to be completely orthogonal in its support for HTTP and HTTPS.

Once upon a time we were all only selectively supporting HTTPS during, like, secure events, and then dropping our users back to HTTP because that's what everybody did. And you were supposed to be able to do that. Well, fortunately we know those days were not really as secure as we were hoping they were, and they are long gone. So anyway, Edge is joining the rest of the group, a little bit with their own flavor, but they'll be getting it right.

I promise to keep the enumeration of new ransomware victims short. So I'll just note that Fujifilm acknowledged last Wednesday that it had been hit by a ransomware attack. It's believed to have been launched by the QBot Trojan, who have recently teamed up with the REvil group. So the Qbot Trojan gets in and basically hands over access to the REvil group to perform their ransomware attack. The Massachusetts Steamship Authority, which is Massachusetts' largest ferry service, was hit by a ransomware attack, also last Wednesday, which led to ticketing and reservation disruptions.

And the University of Florida Health, also known as UF Health, is a healthcare network of hospitals and physician practices that provide care to counties throughout Florida. They suffered a ransomware attack that forced two of their hospitals to shut down portions of their IT network. So they're back to using pen or pencil and paper until their systems are restored. So yes, amid all of the fur flying and outrage about the Colonial Pipeline attack and the fact that we can't get hamburgers anymore, the ransomware attacks just keep on happening.

We believe that we know how Colonial Pipeline was breached. Bloomberg reported on Friday some of the findings by Mandiant, which is the group within FireEye who has been working with Colonial Pipeline to figure out how this happened. As always, attribution and post-attack forensics is difficult. But there's very strong evidence to support the theory that the attackers used a compromised VPN account password. Yup.

Leo: It was just like the Florida water plant. They were all using the same password.

Steve: Yup.

Leo: And the account was no longer active, but it was still usable.

Steve: Right.

Leo: Ridiculous.

Steve: I know.

Leo: No two-factor. Ridiculous.

Steve: The VPN login in question, which lacked any multifactor authentication protection, was not in use, but it had been left active, and it was at the time of the attack. The account's password was discovered inside a batch of leaked passwords on the dark web.

Leo: And you know why that was. Oh, go ahead, I'll let you finish the sentence, yup.

Steve: This suggests that an employee of the company may have reused that same password on another account that was previously breached.

Leo: So Colonial was not requiring password managers. They were letting employees set their own passwords, monkey123. Oh, I use it on everything. It's easy to remember. Unbelievable. Unbelievable.

Steve: So the takeaways are a little late. And it's always easy to admonish with "I told you so" after the fact. But unused accounts should always be disabled. Authentication should require multiple factors. And I suppose that, while I've never been a fan of forced

password changing, so long as the new passwords are unique and not shared, forcing a change might have prevented this entire mess.

I was recently informed that my logon to the management portal for Level 3 would be expiring, since I had not logged into it in six months. Okay, that's annoying, but it's good policy. And for things that are mission critical, like remote VPN access into a corporate network, the pain is clearly worth the gain. So, yeah. The one good thing that will arise from this attention, this is not unwanted attention. This is good attention that the world is now paying to this because, as we've often talked about, the CIOs, the Chief Information Officers, have been running around the C-suite executives, screaming about needing more, more, more. We need more budget. We need more closet space. We need, you know, whatever it is. They are resource constrained. We need to replace this crap which is 20 years old because we can't - and the bosses, "Well, it works, don't it?"

Leo: It works. Till it doesn't.

Steve: Exactly. It works until nothing suddenly does.

Leo: And this was the IT department that was hacked. So surely they should have done better.

Steve: Again, the beauty of the press is that when the executives go home, their wives are now asking them, "Honey?" Things that never occurred to them to ask. You know, "What's the budget for your company's security? Because, you know, I would really hate if Mabel at the club was able to scold me for your company being attacked."

Leo: Just put plastic bags of gasoline in their trunk.

Steve: Exactly.

Leo: Which should never have to happen.

Steve: Okay. So we got the word also that the FBI has struck back, which begs the question, "How, exactly?"

Leo: Yeah, no kidding. I would love to know how.

Steve: Yesterday's press from the U.S. Department of Justice was victoriously titled "Department of Justice Seizes $2.3 Million in Cryptocurrency Paid to the Ransomware Extortionists DarkSide." Now, okay. I've excerpted the interesting new parts, removing the remedial "What is ransomware?" bits. But here's what they said: "WASHINGTON - The Department of Justice today announced that it has seized 63.7 bitcoins currently valued at approximately $2.3 million. These funds allegedly represent the proceeds of a May 8th ransom payment to individuals in a group known as DarkSide, which had targeted Colonial Pipeline, resulting in critical infrastructure being taken out of operation. The seizure warrant was authorized earlier today by the Honorable Laurel Beeler, U.S. Magistrate Judge for the Northern District of California."

Okay, so now everybody's got to get their two cents' worth in. So we have Deputy Attorney General Lisa O. Monaco for the U.S. Department of Justice said: "Following the money remains one of the most basic, yet powerful tools we have. Ransom payments are the fuel that propels the digital extortion engine, and today's announcement demonstrates that the United States will use all available tools to make these attacks more costly and less profitable for criminal enterprises. We will continue to target the entire ransomware ecosystem to disrupt and deter these attacks. Today's announcements also demonstrate the value of early notification to law enforcement." Let us know. They said: "We thank Colonial Pipeline for quickly notifying the FBI when they learned that they were targeted by DarkSide."

So now we have the FBI's Deputy Director Paul Abbate added: "There is no place beyond the reach of the FBI..."

Leo: Oh, please.

Steve: I know, I know, you can run, but you can't hide - "...to conceal illicit funds that will prevent us from imposing risk and consequences upon malicious cyber actors. We will continue to use all of our available resources and leverage our domestic and" - I can hear the music in the background - "our domestic and international partnerships..."

Leo: [Vocalizing]. Oh, no, that's "Get Smart," never mind.

Steve: Yeah, good, "...to disrupt ransomware attacks and protect our private sector partners and the American children." Oh, sorry, the American public. I got confused.

Leo: Well, the children, too.

Steve: That was a different FBI speech.

Leo: Yes, that's right.

Steve: Acting U.S. Attorney for the Northern District of California Stephanie Hinds chimed in, saying: "Cybercriminals are employing ever more elaborate schemes to convert technology into tools of digital extortion. We need to continue improving the cyber resiliency of our critical infrastructure across the nation, including the Northern District of California." Okay, so just to name one.

Leo: Well, there's a reason, because the warrant was a subpoena to a bitcoin exchange in Northern California.

Steve: Ah.

Leo: And so the thinking is they didn't crack the password or anything like that. There was a custodial wallet. The hackers didn't do a good job of securing their wallet. There was a custodial wallet which the FBI got into, probably Coinbase.

Steve: So they said: "We will also continue developing advanced methods to improve our ability to track and recover digital ransom payments." And then, finally, and this is just more nonsense.

Leo: You've got to do these press-releases, though. It's in the great tradition of J. Edgar Hoover. You've just got to do this, yeah.

Steve: It absolutely is. So they said: "As alleged in the supporting affidavit, by reviewing the bitcoin public ledger, law enforcement was able to track multiple transfers of bitcoin and identify that approximately 63.7 bitcoins, representing the proceeds of the victim's ransom payment, had been transferred to a specific address, for which the FBI has the private key, or the rough equivalent of a password needed to access assets accessible from the specific bitcoin address. This bitcoin represents proceeds traceable to a computer intrusion and property involved in money laundering and may be seized pursuant to criminal and civil forfeiture statutes." And then it continues with a bunch of self-congratulatory paragraphs which we will skip.

So your notion of there being a vulnerable public exchange is certainly, I mean, that makes a lot of sense, Leo, because I was explaining to Lorrie the other day, actually it was at a dinner party, that while the cryptocurrency itself may be absolutely impenetrable, it's only valuable, it's only useful if you're able to move those coins, those virtual math coins in and out of specific currencies, fiat currencies that exist and are useful in the real world. So that transaction to and from the real world creates a point of vulnerability.

Leo: Often these exchanges have what they call a "custodial wallet," where in effect they have the keys to your wallet so that they can move stuff in and out for you.

Steve: Right.

Leo: And if that were the case, and a warrant were presented to that exchange, they would say, "Oh, yeah, here."

Steve: Yeah.

Leo: So it doesn't mean the FBI cracked the wallet. It means that the hackers voluntarily handed the keys over to a exchange in order to turn it into money, probably.

Steve: Right. Right. So the other possibility that occurred to me, without those facts, is that they asked for, the hackers asked for payment in Monero, which cannot be traced. Unlike bitcoin's quite visible public transaction ledger, Monero was designed to be untraceable. But the DarkSide group indicated, oh, also remember that the DarkSide group - oh, no. They indicated that ransom could also be paid via bitcoin for an additional 10%. So they wanted Monero.

Leo: Right.

Steve: But if you can't, we'll take bitcoin with a 10% extra vulnerability fee. So we also know that Colonial did not bother using the decryptor provided by DarkSide, which ended up being given to them for free, because it was too slow and buggy. But Colonial nevertheless did pay the ransom. So some have speculated that the ransom was paid, along with the 10% extra for paying via bitcoin, specifically so that the FBI could track the coin flow, which bitcoin permits.

Leo: Ah, interesting.

Steve: What's also odd is that, as we covered at the time, the Eclipse folks who analyzed the bitcoin transaction ledger and were the first to identify the wallet being used by DarkSide claimed that the ransom paid to DarkSide had been immediately transferred out of the group's wallet. So maybe it went to this exchange that you're talking about.

Leo: Had you and I used custodial wallets for our bitcoin - we were too secure. I looked at Coinbase, said I'm not giving you my wallet.

Steve: Right, I'm not, exactly.

Leo: We might still have our bitcoin, had we done that. But okay.

Steve: Yeah. Well, but on the other hand, how many times between then and now have we covered stories of exchanges being compromised.

Leo: Yeah, yeah, exactly, yeah.

Steve: So technically it would mean, if in fact Eclipse was right, and the transaction went in and out of the wallet they identified as belonging to DarkSide, that would mean - and besides, bitcoins are fungible. It's like saying "These are the bitcoins." Well, no. I own the number two; you know? You can't have number two. I have two. And it's like, well, wait a minute, I just wrote it down over here. No, no. That makes no sense. So anyway.

There are people who know exactly what transpired here, and they're not talking. But I'm extremely skeptical that this was, as you said, Leo, an actual breach of DarkSide's wallet. As we know, a bitcoin wallet is an abstraction. It's simply a public key to which bitcoins have been virtually transferred. And this virtual wallet's owner holds the matching secret private key. And a private key is an easy secret to keep. So if they had kept their wallet private key secret, no technical kung fu will somehow magically liberate the private key from its holder.

Leo: Sometimes I wish it would, to be honest with you.

Steve: Yeah. Yeah. So I'd be willing to bet my dinner that social and political pressure was brought to bear directly on DarkSide and that they were instructed in this scenario by the sorts of Russians who you don't say 'no' to, to immediately transfer the entire content of their bitcoin wallet to the following bitcoin address, which was a wallet created

by and under the control of the U.S. Federal Bureau of Investigation. And you know, I mean, that is an absolutely plausible scenario. If someone knocks on the door, as we said from the beginning, the Russian intelligence services, they know exactly who the bitcoin guys are. And so if they come knocking and say you just stepped in a big pile of you know what over there in the U.S., and you need to send your money to this address, and we're going to watch the ledger to verify that you do. It could have happened that way. So not a technical win. Pure and simple behind-the-scenes political leverage applied.

Leo: This was, by the way, that new Ransomware and Digital Extortion Task Force at the DOJ that did that.

Steve: Yeah.

Leo: So I'm glad they're taking this stuff seriously.

Steve: Yeah. And again, when you take meat away from Americans, and when the Eastern Seaboard runs out of gas...

Leo: Yeah, it gets a little serious.

Steve: That gets some attention.

Leo: Yeah, yeah.

Steve: So this is not some geek who's like, oh, well, sorry you got hacked, reload Windows. This is getting to be serious. And I'm glad for that.

Leo: Yeah.

Steve: Okay. WordPress is force-installing Jetpack to five million sites. Last Tuesday the Jetpack folks, who create and maintain one of the most popular plugins for WordPress, and I will tell you that that's the one that I was using...

Leo: Oh, yeah, I use it, too, yeah.

Steve: Yeah. It is the right one for managing a WordPress site. They acknowledged that they had quietly fixed and pushed out an update to resolve a privately reported vulnerability that, had it been exploited in the wild, would have been not good. Their disclosure for Jetpack v9.8 said: "We found a vulnerability in the Carousel feature and its option to display comments for each image." They said: "Thank you to nguyenhg_vcs for disclosing this issue to us in a responsible manner. We have no evidence that this vulnerability has been exploited in the wild. However, now that the update has been released, it is only a matter of time before someone tries to take advantage of the vulnerability."

And of course it's in PHP. And in the coverage of this they showed, like, the code that was changed; right? They said: "We consequently invite you to update your version of Jetpack as soon as possible. To help you in this process, we worked with WordPress.org Security Team to release patched versions of every version of Jetpack since 2.0. Most websites have been or will soon be automatically updated to a secured version. And in fact I think by the time this actually was released, it had been. And the show notes, it's page 9 of the show notes, at the top of them, shows the versions released include, and I can't even read them, I mean, there's one, two, three, four, five, six, seven, eight, nine, 10, 11, 12, 13, 14 per line, and seven lines. So, yeah.

Leo: All the versions.

Steve: All the versions. And then they said: "If you are running any of these versions, your website is not vulnerable to this issue," only because we just patched it for you. So and in fact in their coverage, BleepingComputer provided the current download stats, which are available on the WordPress plugin site, but you need to parse some JSON in order to get it. So it showed that today there had been 3,454,856 downloads. Yesterday, 632,530. Over the last seven days, 5,250,265. So, and all time, get this, 231,457,948. So yes, Jetpack, super popular, you and I are both using it, Leo, and in the last week the current 5,250,265 all got themselves updated before news of this went out in order to keep everybody secure.

And, you know, I understand that the idea of updates being force-installed makes many people queasy. I received some well-considered blowback from my suggestion a few weeks ago that allowing our routers to auto-update should be welcomed with open arms and celebrated. A number of people felt that this would also open us to supply chain attacks, if malicious firmware was ever allowed to get into all of some manufacturer's routers via their update servers. Okay, point taken. But I think it's a matter of the least bad of two options. Allow routers to continue using known defective and vulnerable firmware, thus exposing their unwitting users to significant danger which will never be cured, or allow improvements to that firmware to be pushed when needed. So on balance, I suspect that we're heading toward a more "pushy" future.

Another piece of WordPress news, the plugin is named Fancy Product Designer, but perhaps a more fitting name would be Fancy Site Destroyer. The Wordfence security guys have observed active scanning to exploit a critical zero-day remote code execution flaw which allows for complete WordPress and WooCommerce site takeover. So this is not the happy ending that the Jetpack guys had. A problem was found, it would have been bad if it got loose, if news of it got loose, because the vulnerable audience was five million. The good news is the person who found the problem responsibly reported it. It got patched and fixed, and nobody had to do anything. So five million remote takeovers were avoided. In this case we have 17,000 that are not going to be avoided because this thing has no update built in.

The Wordfence security guys have observed, as I said, active scanning to exploit a critical zero-day remote code execution flaw which allows for complete WordPress and WooCommerce site takeover. The problem lies in a plugin known as Fancy Product Designer. Anybody listening who has that, fix it now. It is a visual product configurator plugin for WordPress, WooCommerce, and Spotify. It allows its users to customize products using their own graphics and content. Sales statistics show that Fancy Product Designer has been sold and installed on more than 17,000 WordPress websites.

Wordfence's Ram Gall said: "The WordPress version of the plugin is the one used in WooCommerce installations as well, and is also vulnerable." The Shopify version of the plugin mitigates and likely blocks the vulnerability because Shopify uses stricter access

controls for sites hosted and running on its platform. However, on WordPress and WooCommerce, successful exploitation of the Fancy Product Designer flaw allows attackers to bypass built-in checks which would otherwise prevent uploading of malicious executable PHP files. And as we know, once you can get a malicious PHP file placed into a directory from which an HTML query will invoke it, it's game over. In short, the flaw allows attackers to completely take over vulnerable sites.

And as the WordFence folks noted, the hunt for those 17,000 sites hosting that plugin is underway now. Ram Gall added that: "This attacker appears to be targeting ecommerce sites and attempting to extract order information from site databases." And, he added: "Due to this vulnerability being actively attacked, we are publicly disclosing minimal details to alert the community to take precautions to keep their sites protected and to update." But again, it's just PHP, folks. So you look at an old version and the current version, and you can see what got changed and fixed. Attacks targeting the thousands of sites running the Fancy Product Designer plugin started more than four months ago, first being seen on January 30, 2021.

So I would say what that says is, if you were using Fancy Product Designer, updating it is probably not sufficient. You need to really closely examine your site because good guys, I mean, well, yeah, good guys can close it, but only after, in this case, the bad guys may have gotten in. Since the vulnerability is under active exploitation and was justifiably rated as critical severity, customers are advised to immediately install the Fancy Product Designer 4.6.9 patched version which was released last Wednesday on June 2nd. And as I mentioned, Wordfence will be holding off on releasing additional details about the vulnerability until a greater percentage of sites running Fancy Product Designer are updated to the latest version.

And since there is no notification system or built-in auto update, no way for the Fancy Designer people to push this out to those sites, hopefully they have some way of contacting the site's users and saying there's a critical vulnerability. You need to really take this seriously and update now. And I hope they do the responsible thing and say, since attacks began late January of this year, and sites have been vulnerable for four months, and it's not difficult to determine whether they are vulnerable remotely, you need to assume, unfortunately, we're sorry to have report this, that since you've been using this thing that you purchased from us for fancy designing, your site may have been taken over or compromised.

GitHub responded as I think we would have expected. We talked about how they somewhat controversially removed a proof of concept from their site because it was demonstrating how to exploit whatever it was, I think it was an Exchange Server exploit at the time. It's a good thing that Wordfence is withholding details. But details need to be withheld longer than they are being. Given current best practices in the security industry, I would argue those are no longer the best. So they have officially announced a series of updates to the site's policies to address the recently controversial questions surrounding how it will handle malware and proof-of-concept exploit code that's uploaded to GitHub.

Under Section 2 of GitHub's Acceptable Use Policies, they now state: "Under no circumstances will users upload, post, host, execute, or transmit any content that directly supports unlawful active attack or malware campaigns that are causing technical harms such as using our platform to deliver malicious executables or as attack infrastructure, for example by organizing denial-of-service attacks or managing command-and-control servers, with no implicit or explicit dual-use purpose prior to the abuse occurring."

In other words, that's the first clause of what I read covers the idea of posting a proof-of-concept exploit for an attack for which there is no current mitigation, or even after updates have been offered, but systems are still vulnerable, which was the instance in which GitHub had pulled the previous proof of concept. And I've got in the show notes a

much more lengthy description of the change of policy, if anyone is interested, and a link to it from GitHub. But basically it talks about that in greater detail.

The point being, yes, we're going to change our terms to be more in line what we think is reasonable, which is why should we be hosting proof-of-concept code which allows bad guys to grab that and immediately leverage attacks in the wild against servers or anything which is now vulnerable. So I think they pretty much had to make it clear that that's what they were going to do. And yes, people should not be posting widely available, public proof-of-concept code which can immediately be leveraged and taken advantage of.

Okay. So a bit of miscellany. I inadvertently set off a bit of a firestorm with my admittedly strongly worded statements two weeks ago about NAT. Okay. And I have in the notes IPv6. I didn't even refer to IPv6. One person tweeted via DM. He said: "You're completely conflating NAT and firewalls. You can certainly have a firewall without NAT," he says, parens, "(if you're not out of addresses) and have an easier to understand environment." Okay. And he didn't mention IPv6, and I hadn't either. But over in GRC's newsgroups someone posted: "Instead of being scared of IPv6, why not learn a bit about it and how security is provided for IPv6-connected devices?"

Then this person quoted a line from Episode 850, I guess it was, no, 822 is - anyway, he's got the wrong episode number. But he quoted me saying: "The nutty IP purists, with their heads well positioned far up their you-know-whats where the sun don't shine, have always decried the use of NAT." And so this guy continues: "They're right." Meaning the IP purists. He says: "NAT is an annoyance that provides nothing other than reducing the use of valuable IPv4 addresses." He continues: "You do not need NAT to provide security. All you need is a stateful firewall that only allows packets in that are part of or are related to an outgoing connection. You" - and here we have all caps - "DO NOT need to be translating between different IP addresses to achieve that. This provides EXACTLY [all caps] the same security that NAT does."

Okay. First of all, duh. For the record, I never suggested otherwise. I never, in any way, intimated that NAT was the only way to obtain the equivalent of a stateful firewall. In fact, my chosen personal firewall for the PC, which I endorsed 21 years ago in 2000, was ZoneAlarm. A bunch of embarrassingly old pages are still up on GRC talking about it. Zone Alarm's claim to fame, along with being application-centric, was that it was inherently stateful. It tracked connections and only allowed incoming packets for connections that had been previously established. There was no NAT involved anywhere.

But the entire world desperately needed the security provided by stateful packet-inspecting firewalls back then, 21 years ago, while personal home networks were growing like weeds and needing many, many of their own IP addresses on the LAN; when ISPs only had the one to give. Where was IPv6? Nowhere. And where is it today? Still mostly nowhere. Oh, sure, some ISPs are now, 21 years later, finally beginning to deploy IPv6. Was the entire world supposed to wait for IPv6 before we could have more than one device attached to our ISP's single-IP DSL or cable modem? NAT was a godsend, and it still is today. Someday we won't need NAT, but I'll wager that we're still going to be using it. And in any event, most users still have no choice because today, in 2021, IPv6 is still not ready for primetime.

And one last point. The guy who posted over in the GRC Security Now! newsgroup wrote, in caps, about IPv6: "This provides EXACTLY the same security that NAT does," which is also incorrect. "Anyone who understands security" - as you referred to earlier, Leo - "appreciates the concept of multiple layers of protection." Having a stateful firewall provides one good layer, yes, whether it's translating IP addresses or not. But having a local network of private and non-publicly routable IP addresses is another massively useful layer of security. Everything that happens inside our local networks is local and off

the 'Net, until and unless something wants to reach out onto the public Internet through our NAT router.

Getting everything to work through NAT has been a pain in the butt. We've had to develop robust STUN and TURN protocols and develop publicly accessible rendezvous servers to directly interconnect two devices which wish to peer when both are sequestered behind their respective NAT routers. But that's all been figured out now, and it works. And that sequestration also helps to keep us safe the rest of the time.

So anyway, I got a bunch of confused responses, and some obviously incorrect. So I just wanted to go on the record. I'm not anti-IPv6. I'm still waiting for it. It's still trying to happen. And I get it. It is, but we couldn't wait for it. We got NAT, and it gave us the security that we needed.

Leo: Yeah. In fact, unfortunately, Carrier NAT has put off IPv6.

Steve: Yes, right.

Leo: By solving the IP address scarcity problem.

Steve: Right.

Leo: I know, I can't believe we're still talking about this after all these years.

Steve: And Leo, this speaks to the inertia that we have.

Leo: Yeah. Why change it if it's not broken?

Steve: Right. And let's make the tiniest change we possibly can. All of our equipment is IPv4, so let's just stick another big Carrier Grade NAT in front of it, and then everything can stay IPv4, and we don't have to replace any of our equipment.

Leo: Yeah. That's the real reason, isn't it. It's down to money again.

Steve: Yeah. And expediency.

Leo: Yeah.

Steve: Okay. "Project Hail Mary." I finished reading the book this weekend, and I very much enjoyed it. One thing I can say without any doubt or question is that any possible movie that is shorter than about eight hours will necessarily utterly fail to do the book justice.

Leo: Well, I should tell you the same guy who wrote "The Martian," who I thought did a very good job with "The Martian," is writing the script for "Project Hail Mary."

Steve: Good luck.

Leo: Yeah, it's going to be challenging. There's a lot of...

Steve: The way I would put it, Leo, is that the book will be influential to the movie.

Leo: Andy said he was very happy with the script. He's seen it, and he was very happy with it.

Steve: Well, okay. It's a shame that you could not make a faithful movie from this book.

Leo: It'd be a better series. Make it a series.

Steve: Leo, it would be, yes, it would be a miniseries. I actually put that in the show notes here because the only way you could do it. On the other hand, it would be a miniseries for us. The general public would probably fall asleep.

Leo: Too much physics. I really don't want any more physics problems, thank you very much.

Steve: Right, exactly. It was a fun read. So here's what I would say. If you are going to see the movie at some point, if you imagine that the movie would interest you, and I think it should, I think our listeners should definitely do it, please allow me to urge you to read the book or have Audible read the book to you before you see the movie.

Leo: Oh, yes. I always say that, though.

Steve: Yes. I mean, but more for this than anything I can ever imagine because, Leo, I cannot imagine how you could do this in two hours. I mean, there were eight hours of content in just the problem solving, let alone setting up the problem and all of the politics and all the committees and all of the other stuff that was happening elsewhere, away from our main character, or that the character was involved, I mean, I just, again, I read "Jurassic Park," and I was annoyed when they left out some things that I thought were really wonderful. They're just going to have to leave everything out in order to make a movie of this. So really, really, really, it was a fun read. I really think our listeners would get a kick out of it.

Leo: Totally would. If you like this show, you will love, I think, "Project Hail Mary." It has a very nice ending, as well, which is good.

Steve: It did, yup, it did have a happy ending.

Leo: Yeah.

Steve: Okay. Extrinsic Password Managers. We're going to talk about that next.

Leo: Very interesting. I'm so glad. The other thing that you didn't address, but it's a rapidly developing story, is the use of the FBI and the Australian federal police of what was billed as an encrypted messaging service, ANOM.

Steve: Ah, yes, where they snuck themselves in.

Leo: They were in it for three years. They took it over and ran it for three years and apparently caught a bunch of people. But I want to see what the prosecutions show because I also worry that this is a pretext for saying, see, it's so great when we can see all the messages. We need a backdoor into all of them. In fact, they've already proposed that in Australia on the heels of the ANOM revelation.

Steve: And somebody tweeted to me with this news. The way I learned about this was him saying, one of our listeners saying, "Steve, this is what you've been saying is the problem with iMessage all along." Which, yes, if you're not managing your own keys, somebody else is.

Leo: Yeah, somebody else is. All right, Steve. Let's talk about this blog post.

Steve: So the name Tavis Ormandy is one we've often mentioned on this podcast because Tavis is a prolific security researcher at Google. He's constantly finding problems in this industry's security designs and implementations, posting to Google's zero-day project and starting timers to require companies to fix their stuff or else. So when Tavis posts on the topic of password managers to his own informal blog, it comes to the attention of many who are interested in topics of security. And many of those people listen to this podcast and wonder what I think of what Tavis says.

Now, you know that Tavis is on the techie side because his domain where this blog is located...

Leo: The URL, what is the point of that?

Steve: ...is, okay, so that's an instruction for the x86. It's compare and exchange eight bytes. The domain is cmpxchg8b. And in a multiprocessor environment, you need to give it the lock prefix so that you lock the execution against other processors doing it at the same time. So this domain is lock.cmpxchg8b.com. And I was talking earlier about the crazy instruction sets, and we were talking about 8086. Get this. The compare and exchange eight-byte instruction. So there's 32-bit registers. In this case EDX is one 32-bit register and EAX is another. So the description of what this one instruction does is compares the 64-bit value in the register pair, each 32-bits, EDX concatenated to EAX, with the operand, which is the destination. If the values are equal, the 64-bit value in ECX/EBX, a different pair of 32-bit registers, is stored in the destination operand. Otherwise, the value in the destination operand is loaded into EDX/EAX.

The destination operand is an eight-byte memory location. For the EDX/EAX and ECX/EBX register pairs, EDX and ECX contain the high-order 32 bits, and EAX/EBX contain the low-order 32 bits of a 64-bit value. So yes, that gives you a sense for the world that Tavis and I live in, programming in assembler and the x86 instruction set from time to time. That's actually - a variation of that is actually rather handy for implementing semaphores, but that's a topic for a different time.

Anyway, we've established that Tavis is a techie. So he was not overly wordy in what he wrote. So as I looked at this, I thought for the podcast any attempt I might make to summarize would likely be longer than what he explained. So I've removed his examples and his pointers to specifics. But I think that the best way for me to get to this is just to share what he wrote. And then I'll respond coherently. You and I, Leo, will talk about it.

So he wrote, Tavis wrote: "I've spent a lot of time trying to understand the attack surface of popular password managers. I think I've spent more time analyzing them than practically anybody else, and I think that qualifies me to have an opinion. First, let's get a few things out of the way. For some reason, few subjects can get heated faster than passwords." He says: "Maybe politics and religion, but that's about it. It's okay if you don't like my opinion. Second, everyone needs to be using unique passwords. I don't have to use a password manager to do that. Whatever system works for you is fine. If you want to use a notebook in a desk drawer, that's totally acceptable.

"Okay, let's begin." He says: "Conceptually, what could be simpler than a password manager? It's just a trivial key-value store. In fact, the simplest implementations are usually great. Good examples of simple and safe password managers are KeePass and KeePassX, or even Pass if you're a nerd. Things start to go wrong when you want integration with other applications, or when you want data synchronized by an untrusted intermediary. There are safe ways to do this, but the allure of recurring subscription fees has attracted businesses to this space with varying degrees of competence. I'm generally skeptical of these online subscription password managers, and that's going to be the focus of the rest of this article.

"I often say that 'Use a password manager' is bad advice. That's because it's difficult to tell the difference between a competent implementation and a naive one. The tech press can review usability and onboarding experience, but can't realistically evaluate any security claims. So how do you propose users tell the difference? For that reason, I think 'Use a password manager' is so vague that it's dangerous. A good analogy is telling someone with a headache to pop any pills they find in the medicine cabinet. Maybe they'll get lucky and find an aspirin; or maybe they won't, and you'll be making a call to poison control.

"Advice on this topic needs to be specific." Well, and ours has been, always, of course. He says: "It's better to recommend implementations that are well designed, rather than general product categories. This position is surprisingly contentious. Many people argue any password manager is acceptable, and that I'm sowing fear by actually evaluating vendor claims." He says: "I remain unconvinced." He says: "My primary area of interest is how remote attackers can interact with your password manager." Okay, that's certainly reasonable.

He says: "I'm not interested in things like testing how resistant encrypted blobs are to offline cracking. This might be a valid concern for some. But in most cases, if an attacker is in a position to access or tamper with encrypted state, then you were in trouble whether you used a password manager or not. There are two common issues I run into," he says. "The first is that trusted user interface elements are injected into potentially hostile websites. The second is that different components IPC" - and that's inter-process communicate - "over web-accessible channels, for example, WebSockets, postMessage, et cetera, without adequate mutual authentication."

He says: "Let's discuss user interface elements first. Most online password managers use content scripts, Javascript that is inserted into every website you visit. It's really easy to write content scripts, but really tough to make them tamper-resistant. That's kind of a problem, because they're going to be hosted in hostile environments. How isolated worlds interact is complicated enough, but password managers make matters even worse by blurring the distinction between user interface and content. We've already established that one component of online password managers must be injected into potentially hostile environments. How can those components communicate with other components?

"One naive solution would be to just use XHR or WebSockets to a local HTTP endpoint. This sounds appealing to developers. They're the native way to communicate on the web. The problem with this solution is that it's very difficult to differentiate between your content script and a hostile script running on the same page, but in a different world. Essentially, every implementation I've looked at has got this wrong, resulting in critical game-over vulnerabilities." He says: "Vendors come up with all kinds of hacky solutions to this, often involving inherently racy background scripts that try to verify a tab's origin.

"Another gripe I have with online managers is that they render browser sandboxes less effective. Modern browsers use a sandbox architecture to isolate components that can go wrong. The problem is that online password managers effectively inject privileged components into these sandbox processes with extensions. The purpose of sandboxing is to isolate potentially compromised components from each other. But if you stuff all your valuable secrets inside the sandbox, then what's the point? I worry that people don't understand the tradeoff they're making here.

"Despite what your vendor says, if their network is compromised, the attacker can read your passwords. Here are some selected marketing claims from password manager vendors: 'No one apart from you, not even us, has access to your passwords.' Or 'We keep your information private, secure, and hidden, even from us.' And finally, 'Your data is secured in a way that only you can view it and manage it. Our employees can't,' and so forth."

He says: "These claims are nonsense. An attacker or malicious insider in control of the vendor's network can change the code that is served to your browser, and that code can obviously access your passwords. This isn't farfetched. Altering the content of websites (i.e., defacement) is so common that it's practically a sport. The reality is that you have to trust your vendor to maintain their infrastructure and keep it safe. The existence of encryption - bank grade, military grade, or not - does not alter this." He says: "Perhaps you think this isn't a big deal. You already trusted them when you installed their software. Fine. But these claims are front and center in all marketing, so vendors must believe their customers care about it. I think these claims are bending the truth to assuage legitimate concerns.

"It's easy to poke holes in marketing stuff, but there are some other fun ones I noticed from real password manager vendors. 'Keystroke encryption protects everything you type from being read by cybercriminals.'" He says: "Oh, okay. Or 'Many of the .NET assemblies are obfuscated. So even using a disassembler, users are unable to view critical areas of methods, functions, and classes.'" And he says: "Well, I certainly feel safer." Et cetera, et cetera.

So he finishes: "If you want to use an online password manager, I would recommend using the one already built into your browser. They provide the same functionality" - okay, we'll talk about that in a moment - "and can sidestep these fundamental problems with extensions." He says: "I use Chrome." Yeah, no kidding. "But the other major browsers like Edge or Firefox are fine, too. They can isolate their trusted UI from websites. They don't break the sandbox security model. They have world-class security teams, and they couldn't be easier to use." I'll just interject here that they also share the

common networking problem, which he blasted the other ones for. "No doubt there will be many people reading this who don't like this advice. All I can say is I've heard all the arguments and stand by my conclusions."

Okay. So we obviously have a Google-centric person, and we have a Chrome-centric person, and we have an extremely smart and security-conscious person. I find no fault with anything Tavis has said. And yet I'm proudly and happily using not an intrinsic password manager, but an extrinsic add-on password manager. Why? Because I'm polyamorous. I move among multiple browsers to suit my various needs. I use Safari on iOS. And Leo, I heard you mention yesterday...

Leo: I do, too.

Steve: ...that you do, too.

Leo: Yes. And on macOS, actually.

Steve: Yeah. Right. I use both Firefox and Chrome, bouncing back and forth as needed. And I am often depending upon a single bridge between them. What we might term "intrinsic password manager lock-in" is a thing. I don't want to be prevented from moving to another browser if I choose. I'm not only polyamorous, I'm also fickle. And I want the freedom to use whichever browser I want. Also, I have not kept up with all of the many additional features offered by the best add-on solutions, but I know they offer a whole host of extras. I do take advantage of the Secure Enclave-style synchronized storage for random notes which allow non-password things to be kept safe somewhere and kept synchronized.

So I guess my conclusion here would be that, if someone has no need to ever bridge browser families, nor any use for all of the many extra goodies that are offered by third-party add-on extrinsic password managers, then yeah, such a person's need would be amply met just by sticking with the password managers which have finally been added to our browsers. But then you can't change your mind. You can't go anywhere. You're locked into that browser. And I cannot speak to the theoretical loss of security which Tavis argues will necessarily accompany the use of any third-party tool. I have no doubt whatsoever that there are many horrifically insecure and poorly designed password managers, just as with anything else. But what I do know is that we are not seeing any evidence that the most carefully and well-designed third-party password managers are introducing exploitable vulnerabilities. It would be such a disaster if they were that I'm sure we would know. And we don't.

Leo: Yeah, and - okay. So obviously I don't completely disagree. I understand what he's saying. I think that he forgets that we live in the real world.

Steve: Yes.

Leo: And by the way, I would like to point out it was only recently that Chrome's password manager did not expose your passwords in plaintext to anybody who had access to your computer.

Steve: I told the story a while ago of how, like, Lorrie couldn't remember some password. I said, oh, let's go look. And I showed her. She says, like, what? Yeah.

Leo: And Chrome's explanation was, well, if they have access to your computer, you're screwed. Well, okay, Tavis. Tell me more. You know? So that's problem number one is he asserts that Chrome and Firefox and Safari store passwords securely. I'll accept that, you know. Certainly nobody has done more to make Chrome and others more secure than Tavis Ormandy. But as you point out, it's not the ultimate solution because there's lots of other reasons we use password managers. And everybody's cross-platform, I would think. I mean...

Steve: This day and age? Windows and iOS. Windows and Android.

Leo: Yeah. If I only used Apple stuff, I think Apple does a very good job with their Keychain. It's very secure. It has a feature that Tavis kind of glosses over. How am I supposed to generate these long strong random passwords? He says use a notebook? I don't think he means that. You need something to do that. Now, maybe you've got a tool that does that, or you've got a bag of dice or something. I don't know. But that to me seems like a potential flaw in this. You do want some software to generate these passwords.

Steve: Yes. I go to a random generator all the time just to grab mine, without a second thought.

Leo: Yeah. And then, finally, I wonder if maybe his objections could be solved if you were using a password manager - I understand what he's saying about breaking the browser's security model. That's a fundamental thing.

Steve: Yes. As a purist, the sandbox, you want to have the sandbox be...

Leo: That I understand.

Steve: ...absolutely immutable.

Leo: So what if I use - and most password managers, certainly Bitwarden and LastPass and 1Password, allow you to run, in fact have a standalone app. It's not in the browser. If you don't, in fact, for a long time I used it this way. I would not install the browser extension because I was concerned about that. I would merely open the app, search for the password, copy it and paste it in. Would that be okay?

Steve: And now you've got the problem of that password existing on the clipboard. And the beauty of having it automatically inserted into the form for you by a password manager is that it doesn't touch your external computer. And the external computer is where malware lives.

Leo: So I'd like to know more about how Tavis proposes doing this. Does Chrome generate passwords?

Steve: I was just wondering that when you were talking about that.

Leo: I don't think so. Safari does.

Steve: I think Chrome does. I'm having a problem now with multiple things offering me passwords when I'm logging into a site for the first time.

Leo: So that's problem number one is we need a way to generate a unique password every time. It has to be easy and fast. And truly random, by the way, not a pseudorandom number generator. So we've got a truly random password generated by something. And he's proposing that we just let the browser remember it. And I guess then I could open the browser if I wanted to log into an app and copy it out of the browser. Because remember, people use password managers for apps and accounts of all kinds, not just in the browser.

Steve: Yes, that is the other point. I forgot to mention that also. I do exactly the same thing.

Leo: So I'm unclear what his - I'd like to know what his workflow is that he's so happy about. It strikes me it's onerous and prone to flaws and errors. And I think he's not living in the real world particularly.

Steve: And I think as he explicitly said, he's annoyed when he hears someone say "Use a password manager, period."

Leo: That I understand.

Steve: Not saying use this or that known to be reputable, high end, time and experience tested password manager.

Leo: Right. And that's why when you vetted and approved of LastPass, I was thrilled because that told me this is reliable. And one of the reasons I like Bitwarden is because it's open source and constantly being audited, has been audited multiple times.

Steve: And in fact, because LastPass wasn't open source, the only way Joe had of allowing me to verify what he had done was to allow me to play with a page which ran the same crypto that I was able to look at.

Leo: Right. And we don't know what's happened since then, either, I could point out.

Steve: No. No idea.

Leo: So that's my big problem with closed source for security in general is just lack of accountability. And I don't know, is the Chrome password manager part of Chromium? Is it in the open source? Or is it part of the stuff that's hidden by Google? I would guess it's the latter.

Steve: Ah, interesting. I don't know.

Leo: And there would probably be good reasons for that. But again, that's another issue. Of course Tavis Ormandy trusts Chrome. If I were a Google employee, I might, too. But, you know, that's his company, not mine. But I think the real takeaway, and I wish he'd just really left it at this, is the problem with password managers as an extension in your browser is it breaks the browser's security model. Had he said merely that, I think that's a point very well taken. Right?

Steve: Yeah, yeah.

Leo: And that's something we don't really consider. I actually have in the past because there have been JavaScript issues with all of these in the past.

Steve: Yeah. And one story that I looked at but didn't get to, just because here we are at two hours, was there was some mention, and I saved all the tabs so I can catch up next week if it ends up being useful...

Leo: That's another problem for another day.

Steve: ...is that there was just some mention about a consortium of the browser vendors getting together to look at improving extension security. So we may, you know, like Tavis may be having to give up on the idea of leaving the browser extension-free because any extension is going to be doing this.

Leo: Well, that's right.

Steve: Not just a password manager.

Leo: What about ad blockers?

Steve: Right.

Leo: We all use those. I think as a net gain running UBlock Origin is a security positive. But it does add potentially a problem because you're inserting, injecting JavaScript in. I think his point is well taken, but I'm not sure I agree with the final conclusion. Anyway, I'm going to keep using a password manager.

Steve: Amen. I'm not going anywhere. And I know our listeners aren't, either.

Leo: No, yeah. It's a really interesting post. But this is what I've always thought about Tavis. He's very, you know, he's a great engineer, brilliant guy, very black-and-white. Right? It's either good or not. That's typical engineer. One or zero. And the world is sometimes just a little more gray than that.

Steve: And don't forget the lock prefix when you use the compare and exchange eight bytes instruction.

Leo: You know, I thought this is ridiculously cryptic. And then at the top of the post he has what I presume is an ID to verify its authenticity, I would guess.

Steve: I looked at that, it's like, what the heck is that?

Leo: That looks like a hash. But I don't know. I mean, is it an MDA hash that you could say, oh, yeah, this has not been modified?

Steve: Well, of course if you modified it, then you just modified the hash.

Leo: Yeah, I just changed the hash. So I don't know.

Steve: So, yeah, I don't know what he's doing.

Leo: He's a kook. He's a character.

Steve: Maybe he wants to have a unique search term on the global 'Net so that you can always find his posts by ID.

Leo: Well, that's the good news. If that's your TLD, no one's, you know, that wasn't hard to register. I doubt there was a lot of competition for that. Have you ever - you've never met him; right? We should - I'd love to meet him sometime.

Steve: Yeah, be neat.

Leo: Find a chance to have a drink at RSA or something. Steve Gibson, you're meeting him every week. You'd better be here. You can't miss this show. Every Tuesday at 1:30 Pacific, that's 4:30 Eastern, 20:30 UTC. We gather to talk about security in the most in-depth, intelligent way that exists on the Internet. And we're so glad Steve does this, and we pray that he will continue past 999.

If you want to visit his site, GRC.com is the place to go. You'll find of course his bread and butter there, SpinRite, the world's best mass storage maintenance and recovery.

Steve: Thank you.

Leo: I'll get it one of these days. Mass storage maintenance and recovery utilities. I say that because it used to be hard drives, and now it works for SSDs, too. 6.0 the current version. 6.1 is coming. Buy it now, you'll get 6.1 free. And of course you'll also get to participate in the development of 6.1, which is in very active development. There's forums for that, and all his other forums. The forums are very active at GRC.com.

You'll also find the show, 16Kb versions. That's unique. No one else makes a 16Kb version of the show. He also makes a really nicely done, and on his own dime, I might add, transcript of the show so that you can read along as you listen, or you can use it to search. All of that you'll find at GRC.com, along with a 64Kb audio version.

We have the 64Kb audio and video, as well, at TWiT.tv/sn. There's a YouTube channel with all the shows, all 822. Well, no, actually, because we only started doing video after a few hundred shows. So the most recent 600 or so in video. You'll also find links to your favorite podcast players and an RSS link if you want to add it manually to your podcast player. That way you can get it the minute it's available of a Tuesday afternoon. And if you use a podcast client, please, if you have a chance, leave us a five-star review. Let others know how great Security Now! is.

Steve, have a great week, and I'll see you next week on Security Now!.

Steve: Thanks, buddy. Bye.