BitLocker: Chipped or Cracked?

Description: What's the story behind the massive incredible three million toothbrush takeover attack? How many honeypots are out there on the Internet? What's the best technology to use to access your home network while traveling? Exactly why is password security all just an illusion? Does detecting and reporting previously used passwords create a security weakness? Will Apple's opening of iOS in the EU drive a browser monoculture? Can anything be done to secure our router's UPnP? Has anyone encountered the "Unintended Consequences" we theorized last week? Are running personal email servers no longer practical? And what's up with the recently reported vulnerability in many TPM-protected BitLocker systems?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-961.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-961-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. Yeah, that three million toothbrush DDoS attack thing, maybe that wasn't exactly how it happened. Steve has the details on that. Why is password security really just security theater? Yikes. And then we're going to talk about what probably many of you heard about, the BitLocker hack. Is it something you should worry about, and what can you do about it? Steve's got a very simple fix. You'll want to listen to this episode for sure. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 961, recorded Tuesday, February 13th, 2024: "BitLocker: Chipped or Cracked?"

It's time for Security Now!. Yes, you wait all week long for this moment. I know you do. Steve Gibson is here, the man in charge, with the latest security news. Hello, Steve.

Steve Gibson: Hello, Leo. Great to be with you. This is the 13th, which is regarded as a sort of an unlucky number, at least in the West. I know that China's got a whole bunch of numbers.

Leo: Oh, yeah. Eight is lucky. I can't remember what the unlucky ones were. But yeah, yeah.

Steve: Yeah.

Leo: Thirteen, though, this is the day before Valentine's Day, so it's only unlucky if you haven't bought Lorrie a gift yet.

Steve: Oh, the best thing about my having - actually, she chose me more than I chose her - is that she could care less.

Leo: Oh, good.

Steve: Absolutely. I have to, like, you know, by the way, honey, it's your birthday. What? Oh.

Leo: Love that.

Steve: So it's wonderful.

Leo: Love that.

Steve: Yes. I've been - in earlier years of my life this has been that Valentine's Day was my most hated day.

Leo: It's terrifying.

Steve: Oh, and because, you know, girlfriends were comparing what their friends' boyfriends or husbands did. And it's like, well, he did more than you did. It's like, oh, god, just shoot me now.

Leo: You can't win. You can't win. You can't win. So what's in the docket for today's show?

Steve: We have a mostly listener-driven show because, as I was going through the incoming from our listeners, they expanded into some really interesting discussions. So we do of course have what's the story behind the massive incredible three million toothbrush takeover attack.

Leo: Oh, I'm so sorry I brought that up last week.

Steve: No, no, you were right where the rest of the Internet was.

Leo: I got suckered. I got suckered with everyone else.

Steve: We were on the leading edge of a fiasco.

Leo: Although if I had just used my mind, my noggin, I would have realized how hard to believe it was. But anyway, you'll get to that.

Steve: Well, okay. So there's some interesting stuff that went on with that.

Leo: Oh, good. Okay.

Steve: Also we're going to look at how many honeypots are out there on the Internet? It's more than you might think. Also, what's the best technology to use to access our home networks while we're traveling? Exactly why - get this - is password security all just an illusion?

Leo: Oh, no.

Steve: Oh, yes. Does detecting and reporting previously used passwords create a security weakness? Will Apple's opening of iOS in the EU drive a browser monoculture? Can anything be done to secure our routers' really problematic UPnP, you know, the Universal Plug and Play?

Leo: Oh, yeah, I just turned that off. You told me to turn it off, so I turned it off.

Steve: Yay. Yes. Has anyone encountered the "Unintended Consequences" we theorized last week? The answer, uh-huh. And even I have. Are running personal email servers no longer practical? And finally, what's up with the recently reported vulnerability that afflicts, affects many TPM, you know, Trusted Platform Module-protected BitLocker systems?

Leo: Oh, boy. Yeah, that's a big one, yeah.

Steve: So today's topic or today's podcast titled "BitLocker: Chipped or Cracked?" So I think we have another great podcast for our listeners.

Leo: Well, this would be a good time maybe to mention that you should use a password manager. I don't know what Steve's going to say, but passwords aren't completely an illusion. The Picture of the Week. I am ready to scroll it up now, though, if you're ready.

Steve: Yes. This one, I gave this one the title "Your municipal tax dollars, hard at work."

Leo: Uh-oh. Now, John, I can't show this at this point because you haven't given me a switch for the computer screen. So I guess, people, you're just going to have to look at your show notes. Here it is, "Your municipal tax dollars, hard at work." Oh, I wish you could see this. Maybe, Steve, you're just going to have to describe it for us.

Steve: Okay. So, well, I always do for our listeners who are driving and commuting and jogging or whatever it is they're doing. They're listening rather than viewing. This is just another one of those insane, like, what are they thinking? So we have a street corner. We're zoomed in on just one corner, like where you would have sidewalks. And it looks like a rural community. We see something in the background with a couple trailer homes and some parked cars and some, you know, screens and things. It looks like rural U.S.

And, okay, now we have it on the screen, for those who are watching the video. And there's like a patch of sidewalk concrete, and the curb is dropped down to street level so that if you're rolling up with a wheelchair, you'll not have to go over the bump, or you're using roller skates or whatever. There's even that special textured, in this case it's bright pink and kind of nubbly rubber on the leading edge so that I guess if you're not sighted, or maybe in a wheelchair, you're able to sense that you're on the edge of the sidewalk. The problem is, this sidewalk extends maybe a yard? Maybe three feet? And then there's this big sign sticking up that says "End of Sidewalk." Because, well, I mean, it's correct.

Leo: At least you got up the curb okay.

Steve: But I just, you know, Leo, you look at this. And, I mean, the pictures we've been showing recently, there just has to be a story behind them.

Leo: I'll tell you the story. This is malicious compliance.

Steve: Yeah.

Leo: This is complying with ADA regulations. But the problem is, and this is the same way in Petaluma, I don't know what it's like where you are, but developers don't want to put in sidewalks. So if they're not absolutely required by city regulations to put in sidewalks, they won't. And this is an example of they had an ADA compliance feature, which is a curb cut. But they didn't have to put in a sidewalk, so they didn't.

Steve: So it's not a sidewalk, it's a sidestep.

Leo: Yeah, there's no point.

Steve: Because basically you take one step, and then you're done.

Leo: It's ridiculous.

Steve: And I was thinking, maybe the corner is there to join another sidewalk running in the other direction, but it looks like - but there's grass along there, and it looks like this is aimed in only one direction of entry from the street. Anyway...

Leo: It's the least they can do, literally.

Steve: And in case you weren't sure, Leo.

Leo: Yes?

Steve: The sidewalk has ended because there's grass. So there's a "End of Sidewalk" sign posted.

Leo: I like that. That's a really useful piece of information, yes.

Steve: Yeah, because you wouldn't know otherwise.

Leo: You would if you were in a chair. You would immediately sense the change in terrain. Good lord. Oy.

Steve: Wow. Wow. Okay. So just as we were recording, and I did give this one the title "Brushing Up on the Facts," just as we were recording last Tuesday's podcast, news was breaking across the Internet that somewhere around three million electric toothbrushes had all been compromised and had been enslaved into a massive global botnet. And more, that actually it had been used to attack a Swiss firm, blasting them off the Internet completely.

Leo: And I owe you such an apology for breaking into the show, breathlessly relaying this story. I should have known better.

Steve: Why isn't that in your notes, Steve? Well, gee.

Leo: I should have known better.

Steve: So The Independent's - but Leo, really, you were in good company. The Independent's headline was "Millions of hacked toothbrushes used in Swiss cyberattack, report says." Fudzilla, well, that's well named, their headline was "Hackers turn toothbrushes into cyber weapons."

Leo: Oh, boy.

Steve: Boing Boing headlined: "Millions of smart toothbrushes used in botnet attack on company." Even ZDNet, and they actually made it even worse with the way they ended their headline, ZDNet's headline: "Three million smart toothbrushes were just used in a DDoS attack. Really." Well, not really. Even Tom's Hardware: "Three million malware-infected smart toothbrushes used in Swiss DDoS attacks. Botnet causes millions of euros in damages." We even know, Leo, what it cost...

Leo: We don't. It was made up.

Steve: ...the company that was attacked.

Leo: Oh, god.

Steve: And finally The Sun, The Sun reports: "Over three million toothbrushes 'hacked' and 'turned into secret army for criminals,' experts claim."

Leo: Now, in my defense, this news headline came over the wire as you were doing the show.

Steve: Yes.

Leo: But I should have used some critical thinking because I have one of those toothbrushes. They're not connected to the Internet.

Steve: Right. They have Bluetooth. They do not have WiFi.

Leo: They do connect, I mean, I guess you could in theory hack them because they do connect to your phone. So if you had a malicious app...

Steve: So it could jump from the phone into the toothbrush.

Leo: Yeah, and then you had a malicious app that then...

Steve: And this really brings a whole new meaning to the notion of disinfecting your toothbrush.

Leo: Yes, yes. But there's not enough power in there. There's not enough memory. And most importantly, there's no WiFi.

Steve: Well, there were many, many similar reports, hundreds, yet none of them of course were true. Highly respected news outlets repeated the story because, well, talk about clickbait. Oh, goodness. So how exactly this massive reporting screw-up came to pass even today remains a little unclear. I should note that all of the responsible reporting, for example Tom's Hardware I think has had three updates since then and like really been diligent in rolling this thing back and correcting their own record. So everybody who did this, you know, said whoops, and fixed it.

But I think that, well, part of the problem is, in following up and following back this trail, the parties who were directly involved still to this day disagree about who said what to whom. This occurred during an interview with, well, I've got the details here, so I'll explain it without quoting myself, or misquoting myself. But what was originally published was certainly hair curling, if not teeth straightening.

So here's what the world read. This was in the original article: "She's at home in the bathroom. She's part of a large-scale cyberattack. The electric toothbrush is programmed with Java, and unnoticed criminals have installed malware on it and approximately three million similar toothbrushes. One command is enough, and the remote-controlled toothbrushes simultaneously access the website of a Swiss company. The site collapses and is paralyzed for hours, resulting in millions of dollars in damage."

Leo: I'm so embarrassed.

Steve: "This example, which seems like a Hollywood scenario, actually happened."

Leo: No, it didn't.

Steve: "It shows how versatile digital attacks have become." Yes, even your toothbrush is not safe. Stefan Zger, head of the Switzerland offshoot of the cybersecurity specialist firm Fortinet, said: "Each device connected to the Internet is a potential goal, or can be misused for an attack. Whether baby monitor, web camera, or the electric toothbrush, the attackers do not care."

So the day after hundreds of media outlets worldwide repeated the false claim that a botnet of three million toothbrushes had attacked a Swiss company, Fortinet, the now quite embarrassed cybersecurity firm which was at the center of the story issued a statement. They said: "To clarify" - yeah, let's get a little clarification. "To clarify, the topic of toothbrushes being used for DDoS attacks was presented during an interview as an illustration of a given type of attack, and it is not based on research from Fortinet or FortiGuard Labs. It appears that due to translations, the narrative on this topic has been stretched to the point where hypothetical and actual scenarios are blurred." Wow. Give that PR person a raise. That's just, well, the hypothetical and the actual met in the middle, and we're not sure where one ended and the other one started. And after all, it was lost in translation. Right.

So Fortinet went on to say that its experts have "not observed Mirai or other IoT botnets targeting toothbrushes or similar embedded devices." Now, Graham Cluley, who's been following this whole mess, the day after that day after, on Thursday the 8th, Graham wrote: "I can imagine how a Fortinet researcher might have regaled a journalist with tales of how IoT devices like webcams could be hijacked into botnets for DDoS attacks. After all, this has happened. However, giving the journalist a juicy hypothetical example of millions of smart toothbrushes taking down a Swiss company is playing a dangerous game."

He says: "I'm not surprised that journalists seized the story; and as we've seen, then other news outlets repeated it without double-checking its truth. A more experienced spokesperson would have gone to pains to make it clear that the toothbrush DDoS attack example was hypothetical and had not actually happened. Failing that, since the original article was published" - get this - "on January 30th, Fortinet had plenty of time to contact the Swiss newspaper and correct the report, or post a clarification on social media debunking the story as the hysteria spread in the press.

"But Fortinet did not do that until skeptical voices in the cybersecurity community questioned the story. Ironically, Fortinet's researchers have published some genuinely interesting proof-of-concept research in the past on the toothbrush topic, albeit hacking Bluetooth-enabled toothbrushes to mess with brushing time rather than knock a company's website offline." So anyway, many of the various publications that were

forced to update, amend, and retract what turned out to be an erroneous story, took the time to add that, you know, like trying to cover themselves a little bit while, yes, whoops, this didn't actually happen, it was still an entirely possible and even likely scenario. Except of course on toothbrushes that only had Bluetooth it actually wasn't.

And of course that may also account, you know, the fact that we're prepared for this, that could account for the fact that everyone rushed to submit the story. You know, even though it was not true, it carried the ring of truth for any tech publication since, as everyone listening to this podcast knows, routers and security cameras and IoT devices of all makes, models, and functions are indeed being compromised and enlisted in botnets daily. It's not science fiction, even though this particularly intriguing story was pure fiction. So anyway, Leo, again, no harm, no foul. And I may have picked up on it if my newsgathering had been a little later in the day than it turned out to be. Because this, as you said, just happened as we were beginning the podcast. So there.

Okay. So I got a kick out of the blog post headline posted at the VulnCheck website. It read "There Are Too Many Damn Honeypots!" So here's what the VulnCheck guys explained. They wrote: "Determining the number of Internet-facing hosts affected by a new vulnerability is a key factor in determining if it will become a widespread or emergent threat. If there are a lot of hosts affected, there's a pretty good possibility things are about to pop off," as they put it. "But if only a few hosts are available for exploitation, that's much less likely. But actually counting those hosts, turns out, has become quite a bit more challenging."

They said: "For example, take CVE-2023-22527." So that's last year. "This affected the Atlassian Confluence servers." They said: "At the time of writing, Confluence has appeared on CISA's KEV" - K-E-V, the Commonly Exploited Vulnerabilities list - "nine [yes, nine] times." They wrote: "That's a level of exploitation that should encourage everyone to get their Confluence servers off the Internet. But let's look for ourselves. There are a number of generic Confluence Shodan queries floating around, but X-Confluence-Request-Time might be the most well known. This simply checks for an HTTP response header being returned."

In other words, okay, so breaking from them for a second, as we know, the Shodan Internet search scanner is constantly scanning the 'Net and aggregating the presence of hosts on the Internet. Who's listening to what port on what IP? And in the same way that Google indexes the Internet so that it's easy to find a site by search terms, Shodan indexes the Internet so that you're able to find vulnerable or at least present services by IP and type of service. So it's a search engine for stuff that's listening on ports.

So Shodan can make an HTTP query to Confluence's service port; and if the reply coming back from that port contains the reply header "X-Confluence-Request-Time," that strongly suggests that there's a running Confluence server answering queries at that IP and port. So the VulnCheck guys then show a Shodan screen capture showing, get this, 241,702 occurrences of that reply header being returned from queries across the Internet. Then they point out one particular thing. They say: "241,000" - it's a little more than that - "hosts," they said, "is a great target base for an emergent threat. But on closer examination, there's something off about the listed hosts. For example, this one" - and they select one - "has the Confluence X-Confluence-Request-Time header, but it also has an F5 favicon," you know, as in the well-known security firm F5 Systems. Uh-huh.

Leo: Uh-huh.

Steve: And, they say: "It also claims to be a QNAP TS-128A." You know, a NAS device. They say: "This is a honeypot," you know, because it's arranging to look like a bunch of things in order to attract flies.

Leo: I've got to tell you this is something that our sponsor would never have done. They have so much, so accurate, and they don't put their little logo in it, and they don't impersonate more than one device. So this is not the Canary, obviously. This is some other...

Steve: Right, right. Well, and I was thinking about this, too. Canaries are not meant to be publicly exposed.

Leo: That's right.

Steve: They're for your LAN in order to detect intrusion.

Leo: That's what you want. You don't - yeah, exactly, yeah.

Steve: Yeah. There was no reason you would stick it out there just to take incoming from the Internet.

Leo: We know there's bad guys out there. We don't have to attest to that.

Steve: Yeah. What we want is to find out if any of them get inside.

Leo: Mm-hmm.

Steve: So the VulnCheck guys say: "Whoever created this honeypot was somewhat clever. They mashed together the popular Shodan queries for Confluence, F5 devices, and QNAP systems, to create" what they described as "an abomination that would show up in all three queries. To avoid throwing exploits all over the Internet, and thus getting quickly caught, some attackers use Shodan or similar to curate their target lists. This honeypot is optimized for this use case."

Leo: Oh, interesting.

Steve: "Which is neat. But it blocks our view of what is real."

Leo: Right.

Steve: "Can we filter them out of our search?" They say: "At this point, it's probably useful to look at what a real Confluence server HTTP response looks like. The server has a number of other useful headers to key off of, but we'll try to filter by adding in Set-

Cookie: JSESSIONID=. That update brings the host count down." Okay, so now they're saying - so they modify their Shodan query so that they want it to have both that very popular X-Confluence-Request-Time header, and to be setting a cookie named JSESSIONID=. So they're doing an AND on those two requirements. And they write: "That update brings the host count down from 241,702 to just 37,964, so just shy of 38. And they call that 'probably-actual Confluence servers publicly exposed to the Internet.'"

But is that number real? They say: "It still seems high because most of those do not respond with an actual Confluence landing page. A simple way to capitalize on that is to also search for a snippet from the Confluence login page in our search criteria." So they add another term to the Shodan query looking for the returned HTML to contain the phrase "confluence-base-URL." And they say: "Ah, now we're down to 20,584," a little over half as many as before they added that additional term. And they write: "This knocks off 17,000 hosts, and things are looking more Confluency. But there seems to be a whole bunch of entries without favicons. Let's drill down into that one and see."

So they do that, looking for the presence or lack of any favicon for the site. And at one point it occurs to them to examine the value being returned in the Confluence JSESSIONID cookie settings reply header. And what do you know. A great many of those across the Internet have identical values. Meaning they're not being generated dynamically. They're part of some fixed Confluence simulating honeypot, and the simulation took some shortcuts, that is, the simulation of the honeypot took some shortcuts, for example, randomizing the JSESSIONID which gives it away when it's examined closely enough.

By applying this spoofed JSESSIONID filter, the number now drops to 4,187 probably authentic publicly exposed Confluence servers. So again they write and conclude. They said: "A quick investigation suggests that this could be the complete set of real Confluence hosts, or just very, very good honeypots." They say: "That's a reduction from around 240,000 hosts all the way down to just 4,200. That means there are approximately 236,000 Confluence honeypots on the Internet, or more than 50 times the actual number..."

Leo: Of Confluence users.

Steve: "...of real Confluence servers."

Leo: I'm thinking, well, that's interesting. Why do people want to do public honeypots? I don't get that.

Steve: Right. Just, you know, just probably to see. Anyway, they say: "A vulnerability that only impacts 4,000 hosts is much less concerning than a vulnerability that impacts 240,000. Understanding the scale of an issue, and therefore being precise about the number of potentially impacted hosts is important, too. Those who copy overinflated statistics or haven't done their due diligence are making vulnerabilities appear more impactful than they truly are." Uh, three million toothbrushes, anyone?

"Anyway, while we focused on Confluence," they said, "this particular problem has been repeated across many different targets. Honeypots are a net good for the security community. But their expanding popularity does make understanding real-world attack surfaces much more difficult for defenders, not just attackers." And Leo, I really think you raise a good point. You know, we're talking a quarter of a million.

Leo: That's a lot of them.

Steve: Bogus Confluence servers. What? You know, you're right, that's - I don't know that there are that many bad Russians.

Leo: Not as much fun to be a hacker as it used to be. I just...

Steve: So anyway, this will be a very good rule of thumb for us to keep in mind moving forward. Academically, it's interesting that the explosion in honeypot use and population is this large. I mean, it's like, what? Who are all these people? You know, that's sort of astonishing. But this means that the tendency to immediately rely upon and believe the results of a simple, not-very-critical Shodan search for a given open port - assuming that that means there's a truly vulnerable service running there - needs to be significantly tempered. And it also suggests that future Internet vulnerability scanners will themselves need to do a better job of filtering out the honeypots since the problem has obviously become nothing less than massive.

Leo: And it might be worse even than that because these were not well configured honeypots. I mean, any hacker worth his salt would have immediately noticed the Fortinet or the F5 icon and the fact that it was both a QNAP and, I mean, that's a little bit, you know, the whole thing doesn't ring true. And I would think most bad guys, except for script kiddies, would be sensitive to that and watching out for that. There are probably many, many, many more that they can't see because they're well configured. They look just like a real Confluence server.

Steve: Yup. Yup. Leo, let's take our next break, and then we're going to plow into some user-driven, really interesting discussions.

Leo: And again, I'm sorry about the toothbrushes. They're just Bluetooth devices. They can't...

Steve: Leo, I'm glad to know you're taking good care of your teeth. Teeth are very important.

Leo: Yes, they are. I should have just paid more attention.

Steve: You've got a hi-tech toothbrush. Hopefully it hasn't been hacked to have its running time reduced because...

Leo: We will get to him. Someday we will kill him with tooth decay. It may take a few years. And now on with the show, Steve.

Steve: So Dextra tweeted: "Hello, Steve. Thank you for introducing me 13-plus years ago to the world of being security-minded from a tech perspective. I travel a lot, and over the years I've been working on trying to come up with a solution where I can appear on my home network so I can access and watch content on my cable provider's app while

being secured with the least amount of possibility of opening my home router up to external threats. I've a Synology RT2600ac router at home. I've recently started to travel with a Beryl travel router. I do have an extra Synology RT2600ac router that I've traveled with in the past. Do you have any suggestions on how to go about appearing to be on my home network in a secure manner so I can access my cable provider's catalog and live TV? Signed, RM."

Okay. So this has changed over the years. Ten years ago the standard generic answer would have been to arrange to set up a VPN server at home, and then VPN into your home network from afar. That's no longer the optimal solution. Among other things, it's often more easily said than done, and it requires opening a static port through your home router, which is then visible to anyone on the Internet, like Shodan, not just you. While there are ways to do this safely, it's no longer necessary thanks to the widespread availability of many free and terrific overlay networks.

The very early such network we talked about many years ago was Hamachi. It was originally free, then it went paid, and then it was purchased by LogMeIn. It's still possible to use LogMeIn's Hamachi for $50 per year. But many free solutions exist, and they're just as good: Nebula, which was done by the Stack people; Tailscale; and ZeroTier are three of the very popular ones. Since I didn't know anything about Synology's RT2600ac router, I went over to Michael Horowitz's astoundingly useful and comprehensive "RouterSecurity.org" site. It's, you know, as the name sounds, RouterSecurity is one word, dot org. There's just so much stuff there. His site allowed me to quickly learn that the router has some possible use as a VPN client, it builds in a VPN client, but it doesn't appear to be general purpose enough to host - the router itself doesn't appear to be general purpose enough to host an overlay network, which any Raspberry Pi can do, for example.

So this would mean that, when traveling, some machine inside your home network would need to be left running. But, as I said, that could just be a Raspberry Pi serving as a quiet, always-on, fanless network node to anchor the overlay network. Then you'd run another node on your laptop, and all of these things are multiplatform, so whatever OS you're carrying it'll be compatible. And then you'd be all set. Essentially, your laptop and your cable provider's catalog and video streaming would see that you were connected to them from home, and this is just trouble-free.

Now, as I've mentioned before when I've talked about overlay networks and these various ones, I get people saying, okay, well, which one do you recommend? I can't recommend one because I have not had the need nor the chance to do this myself since I've not been traveling. But the next time I'm going to be out and about, I will make time to check out the various overlay network solutions. I can say, however, that the response from our listeners who have bitten the bullet and set up overlay networks has been, like, gob-smacked positive.

I mean, they can't believe, they just, you know, they can't believe that it is that simple to obtain world-class security in cross-device networking through the public Internet, which is anything but secure. So, you know, the day has truly arrived when it no longer needs to be difficult in order to do that. You just have to poke around. There's, you know, YouTube is full of how-to videos on overlay networks. Again, Nebula, Tailscale, and ZeroTier are top of the list.

Leo: Seems like Tailscale is very popular.

Steve: Yes. Yes.

Leo: So that would be my first guess.

Steve: So Evan, wow, I can't pronounce his name, Phyillaier - sorry, Evan. Anyway, he said: "Hey, Steve. Love the show. I run an ecommerce site, and my customers have been asking for an easier way to log in. I was wondering if there are any security considerations for going passwordless via email only. The system I would like to set up is registration and login via email, i.e., customer just enters their email and then receives a six-digit code in their email to authenticate and log in. Is this just as secure as email plus password authentication? Thanks." So I thought that was a really interesting and intriguing question.

Leo: Yeah.

Steve: So let's answer the last question first: "Is this just as secure as email plus password authentication?" At first we might be tempted to answer, "No, it cannot be as secure since we've eliminated the 'something you know'" factor from the login. But of course that's a red herring; right? Since, as I've often noted, every login everywhere on the planet always and without fail has the obligatory "I forgot my password" link.

Leo: Right, right.

Steve: And, sadly, we're now also seeing "I can't use my authenticator right now." Like, oh, my god, that annoys me. It's like, what? So you don't need that really yet, either. It's kind of like, well, yeah, how about if you have it? Wow. And, you know, and I've even made this notion of the ever-present email link into a joke. You know, where someone explains that they don't need no stinking password manager while they're creating an account by just mashing on their keyboard to fill-in their password field. And when they're asked, "But, but, but how do you login again later?" they glibly explain that they just click on the "I forgot my password" link, then click on the link in their email that they receive, and they're logged in.

The point, of course, is that so long as all username and password logins include the "I forgot what I was supposed to remember" get out of jail free link, our ownership over and control of our email is the only actual security we have.

Leo: Sad to say, yeah.

Steve: The rest is just "feel good" security illusion. This in turn means that the service the password and the password manager are actually performing is only "login acceleration." If your password manager is able to supply the password quickly and painlessly, then the much slower "I forgot my password" login process, which is always available using an email loop, can be bypassed. So it's login acceleration, which is good. As Bruce Schneier would probably describe it: "The password is just security theater."

Leo: Oh, god.

Steve: So calling passwords a "login accelerant" is the perfect context to put them in.

Leo: This is so important. Please, everybody, clip that paragraph, that previous paragraph, and send it to everybody. Because we've said this many times. The weakest link is always the real determinant of how much security you have. And if there is a "forgot my password," that's the weakest link. That's the security you've got.

Steve: Yup. So let's return to Evan's question: Is emailing a one-time passcode to someone who wishes to login just as secure as using a password? It should be clear that the correct and defensible answer is yes.

Leo: It's identical, actually.

Steve: Yes. If the users of his ecommerce site do not wish to be hassled for a password, there is no reduction in security to eliminating passwords entirely and just using an email loop. However, there's also no need for even a six-digit code, since that does not provide any additional security, and it's more hassle which Evan and his users are wishing to avoid. What Evan wants to verify is that someone who is wishing to login at this moment is in control of their previously registered email account. Remember, that's the same fallback test that's being used by every login challenge in the world. This means that all Evan needs to do is email this user a direct login link which contains a one-time passcode as a parameter. And since the user no longer needs to transcribe it, the passcode can be as long as Evan wishes. 32 digits? No problem.

The only requirement for security is that the code must be unpredictable and only valid the first time it is used. Okay. So how do we do that? Let's design the system for Evan. We'll start with a monotonically increasing 32-bit counter. That'll be good for 4.3 billion logins before it wraps around. Now, you can make it 64 bits if you like, so that the most significant 32-bit counter is incremented if the lower 32-bits should ever overflow, even though that would seem to be quite unlikely. And actually, since we're going to put a timestamp in this design also, even if you did have 4.3 billion, and it finally came around to the same, you would not have a valid timestamp in any event.

Okay. So we have a binary value which will never repeat since it's a simple counter that only ever counts upward. And it's stored non-volatile by the server so that it takes like in the registry or in a file so that it writes it back and always starts, even after a reboot, with the next count from where it had left off.

Okay. So we can do several things with that, always incrementing binary value. It could be fed into the AES Rijndael Cipher which is keyed with a random secret and unchanging key. That secret's known only to the server. It's also, you know, it might be coded into it or also written somewhere so that it's non-volatile, it never changes. Then the Rijndael is a 128-bit block. So the 128 bits that comes out of the cipher, basically we have a random secret key which is going to encrypt our 32-bit counter into a 128-bit result. That you run through a Base64 converter, those are available in every language, which produces 22 ASCII text characters.

Since the encryption key will never be changed, and the input to the cipher is an upward-counting counter, the output will never repeat, and it will be cryptographically unpredictable. So we've met several of our conditions. Unpredictable. It never occurs again.

So, just to explore the territory, you could take a salted hash with a secret salt. The counter value would be hashed, and then the hash's output would be similarly converted into text using Base64. Now, it's true that there's an infinitesimally small chance of a

hash collision where two different counter values might produce the same output, but any good hash will be cryptographically secure. And remember that any single bit which changes in the hash's input will, on average, change half of its output bits. So collisions there would really not be a problem. But no reason not to use Rijndael. That's kind of cool anyway.

Okay. So now we have a 22-character one-time token. Evan's ecommerce system should append that token to the link that's sent in the email to the individual who has just asked to log into his system. The instructions in the email are to simply click the button in the email. They do that. This confirms that someone who provided the email address is receiving email at that address, and they are instantly logged in.

At Evan's end, when the token is obtained and the email is sent, those two items along with a timestamp are added to a "Pending Login" list, a list in the sense of a linked list in programming terms. Anytime someone clicks a link, the list is scanned searching for a matching token. The objects on this "Pending Logins" list should use a timestamp so that they are self-expiring. And the way I've organized this on my own expiring lists, of which I have many over in GRC's server, managing all the DNS stuff and ShieldsUP! and everything, and of course this is technically called a queue, is that as I'm traversing the list from its start, I'm also checking the timestamps for every object that I encounter, whether or not they match the one I'm looking for.

If that object's timestamp has expired, I delete it from the list right then so that the list is self-pruning. When I get to the object whose token matches, and if its timestamp has not expired, this confirms the login. I accept the inbound link and log this person in and remove that little object from the list. It would remove itself after it timed out anyway, but might as well, you know, keep it clean.

So anyway, this simple system gives us everything we want. We have unpredictable self-expiring single-use tokens - oh, and that's the other reason to remove it from the list. As you're honoring it and the login, you delete it from the list so that anyone who might capture it somehow is unable to log in again using that token, which is meant to be single use. Evan's users no longer need to mess with a password. They simply go to a login page, enter their previously registered email address, click the "email me" button, open the email that they received, click the button, and they're in. No passwords to worry about, and every bit as secure, actually, as if a password were being used.

If you have a password manager, then you have - you're able to use, on sites that support passwords, you're able to use that as an accelerant to logging in. But it doesn't make you any more secure. And you could argue, if it's a poor password, it could make you even less secure. And that's the danger; right? Passwords that are bad allow bad guys to brute force. If you don't have a password, there's nothing to brute force. So you can make the argument that a passwordless login is even more secure than a system that did have passwords. Yikes. But a really great question, I thought.

Leo: Yeah. I mean, yeah, really you've got me thinking. A lot of...

Steve: It's counterintuitive; isn't it.

Leo: Yeah, Medium uses that. They don't have passwords. It's sort of annoying because it means I have to go to my email every time I want to log in.

Steve: Exactly. And Evan is suggesting that his users would rather do that than have to remember a password.

Leo: So I'm seeing that more and more often on sites like Medium where you just don't set up a password. You just say "email me." Micro.blog does that, too.

Steve: Well, and we're about to encounter that because what you're describing is the unintended consequence that was last week's topic of sites asking for your email because they want to replace first-person tracking because third-person tracking is going away.

Leo: Right, right, right, right.

Steve: Anyway, we'll get there in a second. Margrave said: "Hey, Steve. I've been a loyal listener since the early days. And though I'm not a security expert, I work in Software Quality Automation and have found the Security Now! podcast incredibly helpful several times. I recently created a LinkedIn article and was given the option to share it on social media. When I chose Facebook, I encountered an interesting situation. I remembered changing my password, but it struck me as odd that Facebook would notify me about it." And in his message to me he included a screenshot of what he encountered where it's a Facebook popup, says log into your Facebook account to share. And then it says "You entered an old password. Your password was changed about two weeks ago. If you don't remember making this change, click here." And then it prompts him for his current password.

So then he continues: "I'm not entirely sure if this is a positive or a negative feature for Facebook. Sure, Facebook is often filled with a lot of random stuff like pictures of cats in sunglasses, chickens wearing hats, breathtaking sunsets from someone's backyard, and other equally ridiculous images. But this made me ponder the implications of such notifications. I'm curious to hear your thoughts, as well as those of other listeners, on this feature Facebook is offering.

"I'm also eagerly awaiting SpinRite 6.1. It's been a fantastic tool, and I appreciate all the other facets of your podcast, including your involvement with Vitamin D3. Best regards, Tom."

Leo: Part of this is because Facebook, which was originally for college kids - exactly 20 years ago, by the way, it launched - is now primarily for old folks, people like you and me, who forget our passwords, who change our passwords and forget we changed our passwords, things like that. And who are often, often hacked. I think Facebook accounts are most often hacked. I mean, very, very common.

Steve: So I don't see, to answer Tom's question, any downside to this. And given that Facebook, exactly as you said, Leo, caters to the people who are taking and posting those images which do not impress Tom, I can see the merit in reminding someone when their password was changed, and then for whatever reason they entered their earlier password.

Leo: I think more sites are going to be like this, to be honest with you, as we age.

Steve: Yeah. Yeah. And in fact, you know, to demonstrate that, in Tom's case this was useful to him. He did recall having changed his password several weeks before, but for whatever reason he entered his earlier password. The alternative to having Facebook helpfully saying, hey, you entered an old password, would be "Sorry, that password is incorrect." This would be more confusing than having Facebook recognize and helpfully report that the password was the user's attempted use of an earlier password. And I don't know whether multiple people in a household routinely share a single Facebook account.

Leo: Oh, that's a good point, yeah.

Steve: But if so, one of them might have changed their shared password and failed to inform the others. So this would be a huge help in that instance. The only problem I can see would arise if Facebook were to honor Tom's use of his retired password, but that would obviously or hopefully never happen. So I don't see any downside. And we know that those really annoying systems require their users to periodically change their passwords for no reason, and then also refuse to allow any recently used password to be reused. And, you know, they are - so that means they are similarly storing previous password hashes. So the practice of remembering previous password hashes is not new. I think this amounts to a useful and user-friendly feature.

Leo: And secure, which is what he absolutely worried about.

Steve: Yes, exactly. I don't see any problem for security.

Leo: Good.

Steve: Gimix3, he says: "Hey, Steve. I've been thinking about this thing that now we'll be able to choose our browser in iOS. And whilst I'm excited to be able to run Firefox in my iPhone, I'm feeling a bit uneasy. Safari, by being imposed on iOS and the default on macOS, has gained popularity over the years, and has been 'too big to ignore' until now. Are we going back to the days of the hegemony of Chrome, and websites that can only be used on Chrome?"

So I thought about this for a while, and I would say that it's really up to the other browsers. All of the standards that Chrome, obviously and currently the global dominant browser for here and for the foreseeable future, the standards that Chrome is using are open, open source, and available for adoption by anyone. It may indeed be that if they wish to retain what market share they can, they will need to adopt the same set of open standards that Chrome has. These next few years are going to be really interesting. The only place where Apple is being forced to allow third-party browser engine cores is the EU. And we know that Apple is infuriated by this interference with their sovereignty over their own platform. So it seems unlikely that Apple will similarly be opening their devices to other browsers elsewhere.

Also, the Internet as a whole appears to finally be maturing, waking up and sobering up a bit. We're seeing things tightening up everywhere. Advertising is pulling back. Sites that never had a clear and justifiable reason for their own existence, yet were carrying a huge overhead with a plan to, well, make it up in volume, they're disappearing. What a shock. So in today's climate, I cannot see anyone willfully turning away visitors who come surfing in from any platform. Perhaps internal corporate sites might force their

employees to use some specific browser in order to run their poorly designed software that will only run on a specific browser.

But that's their fault. That is never going to happen as a general rule. No matter what happens on the platform side, especially with the web standardization process so well established today, I doubt we're ever going to see any public sites, certainly none that plan to survive, telling their users that they must go get another browser. That's - I think those days are over. And really those were written by, or those days were largely back when browsers were incapable of doing everything. And so it was go get Flash, download Flash if you want to use this site. And as we know, entire sites were once written in Flash, which, you know, was crazy. So any browser that wouldn't run Flash, wouldn't be able to run that site.

Barbara says: "It occurs to me that the third CISA recommendation might address Universal Plug and Play issues. If UPnP is on, and malware tries to open ports, the user would be notified; right?"

Okay. So Barbara's referring to CISA's third recommendation which we discussed last week, about configuration changes to the router or network device requiring a manual - oh, changes which affect the security requiring a manual intervention by the user of some kind, like them going over and pressing a button saying, you know, enable me to make changes to this router. And she raises a very good point about UPnP, which we know is a real security problem. But I'm afraid that's not what CISA was referring to, and there's really no good way to deal with that particular problem. UPnP is so ubiquitous that all of today's routers enable it by default out of the box. Otherwise things break. And since it's not the router's fault when UPnP is abused, there's no downside for the router to default to having it enabled, as they all do.

The last thing any router manufacturer wants is for some online reviewer to write up that they swapped in this router, and a bunch of things that were working before broke. You know, the fact that it broke because the router's more secure will be lost on the audience. So the value of UPnP for providing hands-free connectivity, which is what it does, is that it needs no management interface. That, you know, it's just magic. Unfortunately, its magic is black, and it is certainly prone to abuse because it allows anything on the internal network, without any authentication barriers of any kind, to create static incoming port mappings to whatever devices are chosen. Because of UPnP's totally freewheeling nature by design, there's no way to require any sort of manual intervention. Today's networked devices just expect UPnP to be there, and for their network traffic to be able to come and go as they please. And unfortunately, secure it is not.

Guillermo Garca said: "Hey, Steve. Listening to SN-960 and your explanation on the reaction and workaround to Google's Protected Audience solution. I have two comments. If this registration requirement is widely adopted, I'm wondering how that will affect the indexing spiders that index the web for us. And then I wonder what kind of password reuse nightmare will emerge if a login is required for every website on the web."

I thought those were two good points. And the second of those two questions occurred to me last week. If we're being asked to create what are essentially throwaway accounts just for the privilege of visiting websites, then why not use a throwaway password? Come up with something that probably meets modern password requirements and reuse it for sites that just don't matter. The problem, of course, is that there will probably be some tendency to keep using that password even on sites that are not throwaway. So this reuse for convenience is instilling a very bad habit, which, you know, we spent the last decade training everyone out of. And this would also render our password manager's "web checkup" features useless since they would be freaking out over all of our deliberate password reuse.

As for spiders, I hadn't considered that. And I wonder how that works today, since news sites behind paywalls appear to be indexed. One thought would be that the user-agent header which identifies a spider might be checked by the site. But of course that would be easy for anyone to spoof in order to get past the paywall, just like the spider does. I suppose that the IP address blocks from which spiders crawl are likely well known and fixed. Or you could do reverse DNS on the IP to see if like it's coming in from Google, from a Google.com property. And of course IP addresses cannot be spoofed.

So it would be possible to admit incoming requests from a set of previously well-known IP ranges without requiring a gratuitous login first. But having said that, there's really no reason why spiders could not just log in like everyone else. I'm sure that, assuming this comes to pass, the problem of keeping the web indexed will be solved somehow. And what we're about to learn is that it turns out no password required. And that's the solution, just like you mentioned, Leo, for Medium.

Earl Rodd tweeted: "Regarding 'Unintended Consequences' and websites requiring an account to view their content. I first encountered this a few weeks ago and wondered why. It clearly was not a paywall. Now I understand why." Referring to last week's podcast. He said: "In fact, no password is needed since it's not an account, but merely a way to track me. They did verify that my email was a real one. So the friction for a user is minimal. Really nothing to remember except my junk email which I have for such purposes." He said: "P.S.: The site was foxnews.com," he says, "one of the several entertainment sites I look at to see the going narratives related to the news."

Okay. So first of all, it's very interesting that no password is needed. And Earl is correct, the only thing they really want and need is our email address. That's what they're trying to get. I went over to Fox News and poked around a bit, and I was not initially prompted for anything. I noticed in the URL bar that Firefox was saying that I had given the Fox site some special permissions of some sort. It turned out that I had disabled autoplay, and audio was blocked on that site. So I cleared any cookies that Firefox might have been carrying.

And then, sure enough, I got the same thing Earl reported. I grabbed a picture of it by myself for the show notes. And it's a box that says "Join Fox News for access to this content." It says: "Plus get unlimited access to thousands of articles, videos and more with your free account." Then there's a form to fill in, just a one-liner, "Enter your email," and then a Continue button. And then in the fine print below it says "By entering your email, you're agreeing to Fox News Terms of Service and Privacy Policy, which includes our Notice of Financial Incentive." That's bold. "To access the content, check your email and follow the instructions provided."

Okay. So they do that page fade effect where you can see the top of the story, but it fades to white and so that it becomes unreadable while this box appears. I was curious about the notice. So in other words, you can't really continue reading the story until you've entered your email, clicked a button, gone to your email, clicked the link there to confirm your email address. All of that gives your browser a cookie which is now tied to your email address. So every time you come back in the future they know who you are. So I was curious about this "Notice of Financial Incentive" they referred to.

Leo: Yeah, what could that be?

Steve: So I followed the link which brought me to the following disclosure. Under Notice of Financial Incentive it says: "This notice applies to our offers or programs (each an 'Incentive Program') that link to this section of our privacy policy." Okay. And of course the page blocking that email that brought me here linked to this, so it applies to what we

just did; right? "And which California may consider to be a financial incentive. You can opt-in to participate in an Incentive Program by providing your email address or other personal information. In exchange for providing your personal information and depending on the incentive program in which you participate, you may be able to access certain content, features, or events, receive a discounted price on an applicable subscription, or receive special news alerts or other entitlements. We will, in turn, use your personal information for the purposes set forth in this privacy policy, such as sending you alerts and marketing messages and personalizing your experience, including providing advertising you may find more relevant and interesting.

"To the extent we can estimate the value of your personal information to us, we consider the value of the offer such as special content or features, the cost to us associated with providing the offer" - in other words, right, it's a net zero, it's a net equal - "and the potential benefit to us in the form of additional advertising or other revenue we may receive as a result of you using our services. The value to us, if any, will depend on the extent to which you engage with our services." Which, boy, you know, some attorneys made a bunch of money putting those couple paragraphs together. Basically what this amounts to is you've given us your email address, which we're going to use to enrich ourselves. And the more time you spend here, the richer we get.

Leo: Yes.

Steve: And, you know, all the sites that are doing this say the same thing. It's very clear that this is exactly what Earl, who first encountered this, suggested it was. I don't visit the Fox News site often enough to have appreciated this as a change of behavior recently, but apparently Earl does, and it changed for him. This is new.

Leo: Yeah, it's probably the - the CCPA I bet is that financial, the California Privacy Act is the financial one.

Steve: Right.

Leo: But yeah, it makes sense. You know, we don't need a pass - you don't need no stinkin' password.

Steve: Nope.

Leo: Just give us your email. That's all we ask. I see that a lot, by the way.

Steve: Yup. As he put it, "I first encountered this a few weeks ago and wondered why." You know, it's not obnoxious, and the lack of any request for a password makes it much less obnoxious. So it looks like we have a perfect example of last week's topic, the unintended consequences of trying to take tracking away from an industry that does not want to let it go.

Leo: Exactly.

Steve: Everyone who fills out these "Join our site" online forms, aside from subjecting themselves to an ever-increasing torrent of spam, will be receiving a completely legal and legitimate first-party browser cookie to uniquely identify them to the site and tie it to their email address. So long as their browser returns that cookie during that and subsequent visits, they will be seen as a "member" of the site so they won't be bothered again. This is a one-time deal. However, yes, the site - with members come advantages; right. The site will in turn forward the visitor's email address to all partners, including all advertisers on that site...

Leo: You bet, you bet, yup.

Steve: ...who will effectively be paying them, be paying the site for that information. Before I had switched away from the site, by the way, uBlock Origin's blocked access attempt count was up to 98 different domains.

Leo: Oh, my god. That has to be a record.

Steve: 98.

Leo: Holy cow.

Steve: 98.

Leo: Oh, boy. You're going to start getting - there's more ads that'll sit in your email, I think.

Steve: Yup. There's more evidence of this. As I was researching the title story for today, the Trusted Platform Module BitLocker decryption story, I scrolled down on the PCGamer.com site, and I encountered exactly the same thing: "PC Gamer Newsletter. Sign up to get the best content of the week, and great gaming deals as picked by the editors." And there it is. And then there's two checkboxes that were not default checked, which I at least appreciated. That was "Contact me with news and offers from other future brands" and "Receive email from us on behalf of our trusted partners or sponsors." Yikes.

And then same fine print: "By submitting your information, you agree to the Terms and Conditions and Privacy Policy and are aged 16 or over." So one thing I didn't mention last week during our discussion of this is that, if anyone doesn't yet have a throwaway junk email account, now would certainly be a good time to establish one. No site to whom we provide this email address will be respecting our privacy. That's the entire point of obtaining our email address. It's so that our privacy can be more explicitly ignored than ever before. And note that we are also implicitly agreeing now to every such site's Privacy Policy, which should be renamed their "Lack of Privacy Policy."

Leo: Yeah, and you know, even if you use a burner email, they don't care. It's a fingerprint. So, you know [crosstalk].

Steve: Yup, exactly. It all ties back to you.

Leo: It's marginally better, I guess. Now, I have to ask you one thing. In the screen shots, I see a LastPass icon. Are you still using LastPass? I thought you stopped using LastPass.

Steve: That's a good point. On this computer I think I must not have done it.

Leo: You forgot to install it. Okay.

Steve: Yeah.

Leo: All right. Oh, that's your screenshot computer. It's probably not your main machine. Okay.

Steve: Right, right, right, it's off the 'Net.

Leo: On we go with TPM, Mr. Gibson.

Steve: Okay. So Tom Walker tweeted: "Hi, Steve. Years ago you mentioned that you leave your phone plugged in all the time. Do you still do that? Just curious if, in your experience, that has kept the phone battery healthy?"

Leo: Ah.

Steve: Now, I do keep my phones charged up all the time. I have an iPhone X that is stuck to an electromagnetic charging stand at either my day or evening location. Otherwise it's in my pocket when I'm out or between locations. But the moment I return home I walk right to the charger, and it docks. Separately, I also keep an older iPhone 7 that my wife retired right here next to me as my desk phone. And as we can see in the video, it is never unplugged. It is essentially a corded phone. And I have three iPads which I use daily. Each, similarly, is always plugged in.

Now, I can't claim to have any clear experimental evidence that this helps the batteries to live longer. The science all says that today's lithium-cycle batteries do not like to be deep discharged. But neither do they like to be overcharged. No, they don't like that at all. They much prefer to be kept nearer to their fully charged state. And I assume that Apple understands all of this and is doubtless very careful not to overcharge their devices, so leaving them connected is safe.

One thing I can say is that my devices always outlive their batteries. That is, I never have batteries die on any of my things. So there's one data point. Another is that I have a friend of many years who used to allow his Apple devices to discharge fully before plugging them in. He was remembering the previous Nickel-Cadmium battery admonishments to always deep-discharge that type of battery chemistry, NiCad, to avoid the famous "memory effect" where NiCads that were only partially discharged a little bit

before they were recharged would start thinking that they were empty at that point where they had been recharged.

Anyway, he killed one Apple device after another until I noticed - I mean, and he complained to me. He says, "These darn Apple - these batteries are no good." And I noticed that his battery symbol was red and, like, screaming for his attention. It was in pain.

Leo: Oh.

Steve: And I explained that plugging them in at every opportunity is the way to treat Lithium-cycle batteries.

Leo: I think also you can trust companies these days to manage their batteries very well, especially Apple. Because Apple actually has traditionally smaller batteries than some of the other companies in their phones. And so they're very - they're constantly tweaking everything to make sure they get max [crosstalk].

Steve: Well, yes. But the problem is, if you refuse to plug it in, and you insist on using it, there's nothing Apple can do.

Leo: There's no way around that. Nothing they can do about that, if you're going to discharge all the way, yeah. I guess what I'm saying is my advice is generally to people just let the - don't worry about it. Let the phone do its thing. Devices these days are pretty good about all that stuff.

Steve: I would say, I mean, it took a long time to train my wife to plug her phone in if there was no reason not to.

Leo: Right.

Steve: That is, it's better to have it on power than not because, you know, if you don't have the habit, it's easy to leave it lying around. Then you grab it when you're running out the door, and it's already low, and it's not going to last long. And it's just not good for it. So anyway, for what it's worth, I just, you know, I keep everything charged up.

Mark Jones said: "Hey, Steve. I know you get no spam, but would like your advice on email deliverability. I too am an old-timer and maintained websites with email for decades. You have not commented on how hard email deliverability is in the age of SPF, DKIM, and DMARC. You also haven't offered advice about maintaining your own email server. February 2024" - that's now - "marks changes for both how Google and Yahoo regard appropriate settings. What's your take? Costs continue to escalate for services that interpret delivery failure events. EasyDMARC was free for multiple sites at one point, now only free for one. And paid plan is more than I pay per month for shared hosting. Is it time to give up running email off my own domains?"

Okay. So I've not commented upon the difficulty of email delivery in the age of SPF, DKIM, and DMARC because I've not yet tried ramping up GRC's rate of mail delivery. I do run my own email server, and it fully supports all three sender verification standards.

They are all configured and running, and GRC has been trickling email in and out of its domain for years with never a hint of any trouble. So it occurred to me there's some chance that I may have already established more of a positive reputation than I was worried might be needed. You know, it's not as if the email that will begin coming from me will emerge from some never-before-seen domain, and suddenly bulk email starts going out. So anyway, we'll see how it goes. And I will absolutely 100% share everything I encounter along the way.

But Mark concluded his note with the question, "Is it time to give up running email off my own domains?" And I think that's a question only he can answer. But from what he mentioned of escalating costs for something called EasyDMARC, for example, it doesn't sound as though he's running his own email server.

Leo: Yeah.

Steve: So he's incurring additional service costs. I am running my own email server, so I have zero cost associated with hosting email environments.

Leo: Well, and I'm going to, you know, our sponsor Fastmail will do all DKIM, DMARC, and SPF for you on your domains, by the way. I have my own - all my email comes to my own personal domain. I don't have Gmail or Yahoo or Outlook, anything like that. It's all Leoville or whatever.

Steve: Right.

Leo: And I do all the - the MX records are all through Fastmail. They do all of the authentication.

Steve: Nice.

Leo: I don't see any reason - I think if it's asking the question should I run my own server, only if you're Steve Gibson. You've got to be nuts to run your own email server. That's just, for one thing, you don't use - you're not using consumer IP addresses. Anybody who has an ISP-based IP address, forget your email ever getting through.

Steve: Yeah.

Leo: You lease Level 3 addresses; right?

Steve: Right.

Leo: You have commercial addresses. And you've been using them...

Steve: I have a block of 24.

Leo: And you've been using them for so long that they've never been used for spam.

Steve: Right.

Leo: So you're not on any blacklist or, I mean, this is - Steve is an unusual case. Very few people should be running their own servers. Domain's different. Servers, don't do that. Don't do that. That's a great [crosstalk].

Steve: I like it.

Leo: That's great. Well, you're fine now; right? Because for whatever reason, you know, those addresses are safe. And you're doing all the right authentication, so you're fine.

Steve: Right, right. Max Feinleib said: "Thank you so much for sharing @abrenty's tip about checking iOS app sizes." He says: "I just deleted over 10GB off my phone..."

Leo: Yeah, these sizes are giant.

Steve: "...in what seems to be nothing but cruft."

Leo: It's terrible.

Steve: So for anyone who's interested, remember, you know, go through, look at the sizes of the data. And if it doesn't make sense to you, delete the app and reinstall it. And none of that crap will come back. Really, Leo, it really is wrong. I'm surprised that Apple doesn't have like a space cleaner. On the other hand, they don't mind selling you larger memory for more money.

Leo: Yeah, I mean, a lot of the other stuff is stuff like attachments in your messages and things. And those get big. And, you know, they're not going to delete those willy-nilly. They're going to, you know, presume that you want them until you delete them. And I wish they did have a way of doing it, but they don't, yeah.

Steve: Yeah. Andre Arroyo said: "@SGgrc" - this was a public tweet. He said: "SpinRite 6.1 Release Candidate 6 running directly on my old iMac and booting off USB." He said: "I couldn't do this before. Now it's easy. Thanks for SpinRite and Security Now." And I put a big picture in the show notes just because it was very cool to see SpinRite sitting there proudly on his iMac screen.

Leo: Now, how does he do that? That's an advanced tip.

Steve: Yeah, you know, if you're able to boot from USB, that's all it takes.

Leo: Okay. And run - this is obviously an Intel iMac. It's not...

Steve: It's got to be an Intel iMac, yes, yup.

Leo: Because you have to be able to run this - you're still using FreeDOS right now; right?

Steve: Yup. Yup.

Leo: But next time it'll be this other DOS that you own, practically.

Steve: The RTOS.

Leo: As the last user.

Steve: I bought it as the sink was shipping. As the ship was sinking.

Leo: As the ship was sinking. Hey, I'll take it off your hands.

Steve: Wait a minute. I'll buy it. I'll buy it. So speaking of SpinRite, I am, as I had hoped, at work on SpinRite's documentation now while SpinRite's paint continues to dry. For example, one user in GRC's forums who had a dying SDHC SD card with a large non-critical file, wanted to experiment with its recovery. So here's what he wrote. He said: "Hi, Steve. I'd like to know if there is a way to have SpinRite perform an operation like a Level 2 scan multiple times. The reason I ask is that I have a Samsung 32GB SDHC card that has a couple of spots it cannot read or write to. I was able to copy all the files except one large one off it." He says: "(An MP4 phone video I took that's not important). And I've decided to play with it to see if it's recoverable. The card passes the SpinRite Level 2 test, but does not pass Level 3 in two areas where I get a 'Device Fault' error.

"The really interesting thing about this is that in running Level 2 a number of times, I've been able to 'heal' some of the bad spots and increase the amount of the file being copied using Windows from 60% to 86%. My thought is, if I was able to have SpinRite do the Level 2 scan overnight multiple times, it might just heal any remaining bad spots."

Okay. So the first thing I explained in my reply to him was that SpinRite can now be completely controlled from its command line. So it would be possible to start it with a command that will bypass any user interaction, select the proper drive and processing level, run SpinRite over the drive, then exit back to DOS once that's done. At that point it's a simple matter to create a DOS command script, which of course DOS refers to as BATCH files, that jumps back up to loop to repeat the command over and over until it's interrupted. So it would just be running a Level 2 over and over and over, which, you know, is apparently good for that drive.

The reason I'm mentioning this is that SpinRite's user can interrupt anything SpinRite is doing at any time. But if the user then manually exited to DOS in this situation, the batch file will still be in control and would immediately restart SpinRite. It would be possible to

exit SpinRite, then frantically hit CTRL-C over and over and over to attempt to get DOS's attention and break out of the loop. But that's certainly inelegant.

So when programs exit, this is all programs everywhere, a nearly universal convention is that they will return an "exit code" to whatever invoked them. This code can signify whatever the program wishes, which is typically the program's sort of generic success or failure. Today, SpinRite exits with a "0" exit code unless it's unable to parse its command line, in which case it exits with a "1." So what occurred to me while answering his question is that when SpinRite is exiting automatically due to the "exit" verb on its command line, and not because of a manual intervention, it could exit with an error code of "2." This would allow for much more graceful "infinite loop" termination by using the DOS line "if errorlevel 2 goto scan" at the bottom of the batch file. Anyway, that way it would loop. And when you used the ESCAPE key to get out of SpinRite, it would drop back out and break out of the loop elegantly.

So anyway, at some point when my eyes are crossing from writing documentation all day, I'll take a break from that to add this tiny little additional convenience feature. And this is the great advantage of having some time to let the paint dry. SpinRite is done. It's working perfectly. No one is encountering any new errors. And again, it's like, it's done. But there's still time for some minor touch ups, and history has shown that once I finally do release it as SpinRite v6.1 and have started working on its successor, I'm going to be extremely reluctant to mess with it any further. So now is the perfect time for those last little tweaks, while I'm working on the documentation and getting it ready for the world.

Leo: Nice. How exciting.

Steve: That is, really. Okay. So, BitLocker: Chipped or Cracked? The number one most sent to me news item of the past week - wow, it was like everybody, seen this, seen this, seen this? Oh, yeah - was the revelation that PCs whose secret key storage Trusted Platform Module functions are provided by a separate TPM chip outside of the main CPU are vulnerable to compromise by someone with physical access to the machine.

This came as a surprise to many people who assumed that this would not be the case, and that their mass storage systems were more protected than they'd turn out to be by Microsoft's BitLocker. During system boot up, the small unencrypted startup portion of Windows sees that BitLocker is enabled and configured on the machine, and that the system has a TPM chip which contains the decryption key. So that pre-boot code says to the TPM chip, "Hey there, I need the super-secret encryption key that you're holding." And the TPM chip replies, "Yeah, got it right here. Here it comes, no problem," and then sends it over to the processor.

The only glitch here is that anyone with a hardware probe is able to connect the probe to the communicating chips of the processor or the TPM chip, or perhaps even to the traces on the printed circuit board which interconnect those two, if those traces happen to lie on the surface. Once connected, the computer can be booted, and that entire happy conversation can be passively monitored. Neither end of the conversation will be any the wiser, and the probe is able to intercept and capture the TPM chip's reply to the processor's request for the BitLocker decryption key.

These are the sorts of tricks that the NSA not only knows about, but has doubtless taken advantage of, who knows how many times. But it's not made more widely obvious until a clever hacker like this "StackSmashing" guy, that was his handle, comes along and shines a very bright light on it. So it's a wonderful thing that he did. And I should note that this is not the first time this has come to light. It happened a few years ago and a

few years before that. So it's the kind of thing that surfaces every so often, and people go, "What? Oh my god."

Okay. The fundamental weakness in the design is that the TPM's key storage and the consumer of that stored key are located in separate components whose communication pins are readily accessible. And the obvious solution to this dilemma is to integrate the TPM's storage functions into the system's processor so that their highly sensitive communication remains internal and inaccessible to casual eavesdropping. And as it turns out, that's exactly what more recent Intel and AMD processors have done.

So this inherent vulnerability to physical attack occupies a window in time where discrete TPM modules exist and are being maybe overly depended upon for their security, and before their functions had been integrated into the CPU. It's also unclear, like just broadly, whether all future CPUs will always include a fully integrated TPM, or whether Intel and AMD will only do this for some higher-end models. Or perversely, it turns out, some lower-end models.

Anyway, all of this created such a stir in the industry that yesterday, on Monday the 12th, Ars Technica posted a very nice piece about the whole issue. And under the subhead "What PCs are affected?" the Ars guy wrote: "BitLocker is a form of full-disk encryption that exists mostly to prevent someone who steals your laptop from taking the drive out, sticking it into another system, and accessing your data without requiring your account password." In other words, they're unable to start up your laptop, so they just take the hard drive out and stick it in a different machine which they know how to start up.

"Many modern Windows 10 and 11 systems," they write, "use BitLocker by default. When you sign into a Microsoft account in Windows 11 Home or Pro on a system with a TPM, your drive is typically encrypted automatically, and a recovery key is uploaded to your Microsoft account. In a Windows 11 Pro system, you can turn on BitLocker manually whether you use a Microsoft account or not, backing up the recovery key any way you see fit." They say: "Regardless, a potential BitLocker exploit could affect the personal data on millions of machines. So how big of a deal is this new example of an old attack? For many individuals, the answer is probably 'not very.'

"One barrier to entry for attackers is technical. Many modern systems use firmware TPM modules, or fTPMs, that are built directly into most processors."

Leo: I think all AMD systems do that; right?

Steve: Right.

Leo: Yeah.

Steve: "In cheaper machines," he writes, "this can be a way to save on manufacturing. Why buy a separate chip if you can just use a feature of the CPU you're already paying for? In other systems, including those that advertise compatibility with Microsoft's Pluton security processors, it's marketed as a security feature that specifically integrates these kinds of so-called 'sniffing' attacks. That's because there is no external communication bus to sniff for an fTPM. It's integrated into the processor, so any communication between the TPM and the rest of the system also happens inside the processor. Virtually all self-built Windows 11-compatible desktops will use fTPMs, as will modern budget desktops and laptops. We checked four recent sub-$500 Intel and AMD laptops from Acer

and Lenovo. All used firmware TPMs. Ditto for four self-built desktops with motherboards from Asus, Gigabyte, and ASRock.

"Ironically, if you're using a high-end Windows laptop, your laptop is slightly more likely to be using a dedicated external TPM chip, which means you might be vulnerable. The easiest way to tell what type of TPM you have is to go into the Windows Security Center, go to the Device Security screen, and click Security Processor Details. If your TPM's manufacturer is listed as Intel (for Intel systems) or AMD (for AMD systems), you're most likely using your system's fTPM, and this exploit won't work on your system. The same goes for anything with Microsoft listed as the TPM manufacturer, which generally means the computer uses Pluton.

"But if you see another manufacturer listed, that is, not Intel, AMD, or Microsoft, you're probably using a dedicated external TPM." He said: "I saw STMicroelectronics TPMs" - that's a very popular one - "in a recent high-end Asus Zenbook, Dell XPS 13, and midrange Lenovo ThinkPad. StackSmashing" - the guy who publicized this again, you know, reminded everybody of this, "also posted photos of a ThinkPad X1 Carbon Gen 11 with a hardware TPM and all the pins someone would need to try to nab the encryption key, as evidence that not all modern systems have switched over to fTPMs - admittedly something I had initially assumed," he wrote. "Laptops made before 2015 or 2016 are all virtually guaranteed to be using external hardware TPMs when they have any.

"That's not to say fTPMs are completely infallible. Some security researchers have been able to defeat the firmware TPMs in some of AMD's processors with 'two to three hours of physical access to the target device.' Firmware TPMs just aren't susceptible to the kind of physical, Raspberry Pi-based attack that StackSmashing demonstrated."

Okay. So there is some good news here, at least in the form of what you can do if you really need and want the best possible protection. It's possible to add a PIN to the boot-up process even now so that the additional factor of "something you know" can be used to strongly resist TPM-only attacks. Microsoft provides a couple of very good and extensive pages which focus upon hardening BitLocker against attacks. I've included links to those articles in the show notes. But to give you a sense for the process of adding a PIN to your system right now, Ars explains under their subhead: "So what can you do about it?"

They say: "Most individual users don't need to worry about this kind of attack. Many consumer systems don't use dedicated TPM chips at all, and accessing your data requires a fairly skilled attacker who is very interested in pulling the data off your specific PC rather than wiping it and reselling or stripping it for parts." He says: "This is not true of business users who deal with confidential information on their work laptops, but their IT departments hopefully do not need to tell anyone to do that."

Okay. "So if you want to give yourself an extra layer of protection, Microsoft recommends setting up an enhanced PIN that is required at startup, in addition to the theoretically sniffable key that the TPM provides. IT admins can enable this remotely via Group Policy. To enable it on your own system, open the Local Group Policy Editor, using Windows+R to open the run, and then type gpedit.msc, hit ENTER. Then navigate to Computer Configuration > Administrative Templates > Windows Components > BitLocker Driver Encryption > Operating System Drives.

"Enable both the 'require additional authentication at startup' and 'allow enhanced PINs for startup.' Then open a Command Prompt window as an admin and type manage-bde-protectors -add c: -TPMAndPIN. That command" - and this is all in the show notes, of course - "that command will immediately prompt you to set a PIN for the drive." I would think of it as a password. Anyway, he says: "Once you've done this, the next time you boot, the system will ask for a PIN before it boots into Windows." He says: "An attacker

with physical access to your system and a sufficient amount of time may be able to gain access by brute-forcing this PIN, so it's important to make it complex, like any good password." And again, I would make it really good if you're taking the time to do it at all. Why not?

He finishes: "A highly motivated, technically skilled attacker with extended physical access to your device may still be able to find a way around these safeguards. Regardless, having disk encryption enabled keeps your data safer than it would be with no encryption at all, and that will be enough to deter lesser-skilled, casual attackers from being able to get at your stuff."

So, ultimately, we're faced with the same tradeoff as always: convenience versus security. In the absence of a very strong PIN password, anyone using a system that is in any way able to decrypt itself without their assistance should recognize the inherent danger of that. If the system escapes their control, bad guys might be able to arrange to have the system do the same thing for them. That is, decrypt without anything that they don't know. Requiring "something you know" is the only true protection. Maybe something else that you have, if that could be arranged. That's what I did when I did my little European trip to introduce SQRL is I had my laptop linked to my phone, and my iPhone had to be present. At the same time, BitLocking, or BitLockering, a drive is certainly useful since it will very strongly prevent anyone who separates the drive from the machine from obtaining anything that's protected in any way.

So BitLocker: Yes. PIN: Yes. And as we've seen, it's possible to add a PIN after the fact. And if your PIN is weak, you can still strengthen it, and you should consider doing so.

Leo: Do we still like VeraCrypt? Would you prefer VeraCrypt to BitLocker? BitLocker's so convenient.

Steve: It's convenient. And VeraCrypt is 100% strong. I was thinking the same thing. BitLocker suffers a little bit from the monoculture effect of everybody having it, and it just being built in. On the other hand, its convenience means that it won't get in anyone's way.

Leo: Right. Yeah, you just log into the computer as normal.

Steve: Yeah.

Leo: Yeah. But if you wanted really better security, I think VeraCrypt is - we still - that's still is our choice, now that - what was it, its predecessor? I've forgotten now.

Steve: TrueCrypt. TrueCrypt.

Leo: TrueCrypt. TrueCrypt is gone. Yeah, yeah, yeah.

Steve: Yup.

Leo: All right. If your PIN is weak, you can still straighten it. The motto of the day.

Steve: If it is weak, you can still straighten it. I like it, Leo.

Leo: That's Calia, who is a textile worker. Thank you very much, Steve Gibson. You are the best. Steve lives at GRC.com. That's where SpinRite 6 also lives, soon to be 6.1. Like a butterfly, it's coming out of the chrysalis and emerging into the...

Steve: There's movement, folks. There's movement.

Leo: There's movement. The wings are fluttering. If you get 6.0 now, you get 6.1 automatically free. Well, not completely automatically. You have to download it. But it's worth getting 6.0 now so you can have it, and 6.1 the minute it's available. You can also get the beta version now if you are an owner. You can also get this show at the website, GRC.com. And that's free. He has two unique versions, a 16Kb version for the bandwidth-impaired, and the very well done transcripts by Elaine Farris so you can read along as you listen, or search, or that kind of thing. All that's at GRC.com, along with SpinRite and ShieldsUP! and ValiDrive and all the great stuff Steve does in assembly language in the middle of the night. What are your coding hours? You're not a late-night coder, I don't think.

Steve: No. I'm 68. I'm not a late-night coder. When I remodeled my home, and I was 38, it had blackout drapes installed in the master bedroom. You know, I have normal cloth drapes, and then behind it is opaque, like, thick, I don't know, vinyl so that - because I would be coding, and I'd be looking out the window and noticing the sky was getting lighter.

Leo: Yeah.

Steve: It's like, "Oh, no."

Leo: Yeah. Oh, that feeling.

Steve: And I always, afterwards, I chastised myself. I never wanted to stop. But I was useless the next day. I mean, it just screwed everything up.

Leo: For at least a day, yeah.

Steve: So what you needed to have is the self-control back then to make yourself go to sleep. Now I don't need self-control because I'm tired. And so I'm, like, looking forward to hitting the sack and being fresh in the morning.

Leo: Well, it is fresh, almost 6.1, almost fully cooked. We have the show at our website. What? What?

Steve: Lorrie does comment, when I mention that, she says, "Well, yes, you're tired. You just coded for 18 hours straight."

Leo: It's amazing.

Steve: So there is that.

Leo: 6:00 a.m. till 10:00 p.m. or something like that; right?

Steve: Yup. Yup.

Leo: Yeah, very nice. He's a hard-working guy.

Steve: I love to code.

Leo: Yeah. I mean, it's fun, isn't it.

Steve: Yeah.

Leo: I am completely stuck.

Steve: Better than anything I've ever found. Well, except one thing, but you can't do that all the time.

Leo: Second best thing, yeah.

Steve: That's right.

Leo: A lot of endorphins, though. Very good for the endorphins. Thank you, Steve.

Steve: I will be back in a week. Bye-bye.