SECURITY NOW!

Transcript of Episode #699

Browser Extension Security

Description: This week we look at the expressive power of the social media friends we keep, the persistent DNS hijacking campaign which has the U.S. government quite concerned, last week's iOS and macOS updates (and doubtless another one very soon!), a valiant effort to take down malware distribution domains, Chrome catching up to IE and Firefox with drive-by file downloads, two particularly worrisome vulnerabilities in two Cisco router models publicly disclosed last Friday, some interesting miscellany, a particularly poignant SpinRite data recovery testimonial, and then some close looks at the state of the industry and the consequences of extensions to our web browsers.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-699.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-699-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about this week. Of course, iOS flaws, including the FaceTime flaw. He's also going to talk about DNS hijacking, why so many sites host malware, and he has a little bit to say about an interesting study that shows how your online friends say a lot about you. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 699, recorded Tuesday, January 29th, 2019: Browser Extension Security.

It's time for Security Now!. Here he is, ladies and gentlemen, the man you've all been waiting for for a whole seven days, Steve Gibson, the man of the hour. Let me get this nonoperational Facebook portal out of your way here. I don't know what I'm going to do with this thing. I don't even have a Facebook account anymore. And the last thing I want to do is give Mark Zuckerberg a 4K camera into our house. So you want one?

Steve Gibson: No.

Leo: Okay.

Steve: Thank you.

Leo: I didn't think you would.

Steve: Thank you so much.

Leo: Hi, Steve.

Steve: Yo, Leo. So we are one episode away from 700.

Leo: Woohoo!

Steve: It's funny when I talk to people about - like people who don't know what I'm into. I met with some friends late last week, and I mentioned, yeah, I've been, you know. Like whatever happened with that podcast? I said, oh, not whatever happened.

Leo: Whatever happened to that podcast.

Steve: I said it didn't go anywhere. It's still right here.

Leo: Whatever happened to it.

Steve: How did that turn out? Oh, well, it's still in the works. And I said, oh, yeah, we're in year 13. They go, what? And I said, yeah, we just recorded 698. They go, episodes? I said yeah. And, you know, problem is I just have to figure out what to put in each one because there's just so much competition for stuff to talk about.

Leo: It's not that you have too little, it's you have too much. There are so many things, yeah.

Steve: Yes. And in fact on Sunday, two days ago, I presented SQRL for the first time in its finished form to this LETHAL, they call themselves LETHAL, Los Angeles Ethical Hackers and Leets group.

Leo: Oh, interesting.

Steve: And a lot of them were Security Now! Listeners, so they're smiling right now because they go, yeah, we saw him. And several of them reminded me of, oh, yeah, remember when those podcasts were, like, 18 minutes? It's like, no, I don't - can't really say that I - I can't say. And remember the honey monkeys? It's like, oh, yes, we all remember the honey monkeys.

So anyway, there were several pieces of news, a really interesting security analysis of browser extensions, and then some rumblings about Chrome's movement from a version 2 to a version 3 of the browser extension API that made headlines. So I thought, let's call this one "Browser Extension Security," since it's all kind of about that. So that's our main focus for the day.

Also there was an interesting article, a study that was done - it was published in something in a branch of Nature magazine, but gosh, I can't remember now the name, but a reputable magazine. It was an analysis of the expressive power of the social media friends we keep and what an analysis of who we friend or who friends us can say about us, even when we don't say anything about ourselves. And the strength of what they found was what was interesting, so I thought I would just touch on that.

We've got a persistent DNS hijacking campaign which has really got the U.S. government upset, so much so that they imposed some 10-day deadlines on some performance. And the good news is the government is up and running again, so there's a chance that those deadlines can be met. So we'll talk about that. We have also last week's rather significant iOS and macOS updates. And of course, as we all know, we'll have another one very soon now as a consequence of the Facebook bug that was found.

Leo: FaceTime.

Steve: FaceTime, right, sorry, FaceTime bug that was found, which we'll talk about. Also a Swiss site has made a valiant effort since I think it was March of 2018 to take down malware distribution domains. We got some interesting stats from that which just sort of focus on, I mean, really put a point on how big a problem this is.

We've got Chrome catching up to IE and Firefox with its handling of and suppression of drive-by file downloads. Two worrisome vulnerabilities in two Cisco VPN routers that are so bad that, if anyone has them, they should just stop listening to the podcast and...

Leo: Oh, wow.

Steve: I mean, it's, you know...

Leo: Go fix.

Steve: Continue later because there's still more stuff to talk about that's good. But, oh, goodness, I mean, this is bad. Cisco released the patch on Wednesday, and then the exploit was published on GitHub two days later on Friday, and this attacking and scanning immediately began. So this is the world we live in now.

We've also got some miscellany. Leo, you're going to want to have a hanky handy.

Leo: Oh, I do.

Steve: I'm not kidding. A particularly poignant SpinRite data recovery testimonial that - yes. And then we're going to take, as I said, a close look at the state of the industry and consequences of the extensions we're making to our browsers.

Leo: Nice.

Steve: And so it's just sort of a fun Picture of the Week. It was in my archive of things to get to that I just got a kick out of. So I think another great podcast for our listeners as we approach number 700 with a podcast that isn't yet history.

Leo: Somebody in the chatroom is already saying, "Please don't quit after 1000. Please, please."

Steve: That would be after 999.

Leo: 999, yeah, because he apparently can't handle four digits.

Steve: No, that'll be my - I wrote it in assembly language, and that's all there is to it.

Leo: I only have four registers. Or one 32-bit register or whatever it is.

Steve: I have this...

Leo: What?

Steve: Our Picture of the Week.

Leo: I have it. Oh, it's so funny.

Steve: It's just good. It's showing a big, sturdy-looking brick building with a big stop sign on the side and the guard with the little gate down, and it's labeled "U.S. Cybersecurity." We've got even razor wire spiraled along the top. And then we also see a side view of the building where the Windows logo with its four panes is shown with the blue one cracked through, with a ladder propped against the building. So it's like, yeah. And I put my own caption on it, "The Weakest Link?" So, yeah.

Leo: Mm-hmm, very good.

Steve: And I've said before, I guess we've talked about there being some migration away from Windows. I think it was China, was it, that they're working on a Linux-based...

Leo: They have a Red Linux that's their own distribution, yeah.

Steve: Right. And I just - I'm stunned that anyone who isn't a clear ally of the U.S. would just pervasively use a closed source operation system from a U.S. company. It's just - I just, you know, there are just too many opportunities for, I mean, even if there weren't any deliberate backdoors, and as far as we know there are none, we know Patch Tuesday, it's happening every Tuesday just like the podcast.

Leo: Yeah, every month.

Steve: Well, every month.

Leo: And we do talk a lot about our fear that Huawei and other Chinese companies are watching us. But remember there are a lot of countries around the world that don't trust our software and our hardware for the same reason. And we know we engage in this. That's what we learned with Vault 7 and Edward Snowden's revelations and all of that.

Steve: Yeah, yeah. So this interesting piece of work, I titled it "The Acquaintances We Keep" because the research is titled "Information flow reveals prediction limits in online social activity."

Leo: Oh, well, now that you put it that way.

Steve: Much simpler, yeah. And in fact I am going to also paraphrase what they said in the abstract after I read it. But I thought it'd be an interesting little tidbit to tuck away that our listeners could bring out at the proper time, perhaps when standing among a group of non-techie people, like with a drink in your hand or something.

The abstract of the paper says: "Modern society depends on the flow of information over online social networks, and users of popular platforms generate substantial behavioral data about themselves and their social ties. However, it remains unclear what fundamental limits exist when using these data to predict the activities and interests of individuals, and to what accuracy such predictions can be made using an individual's social ties.

"Here" - and that is in this research - "we show that 95% of the potential predictive accuracy for an individual is achievable using their social ties only, without requiring that individual's data. We used information theoretic tools to estimate the predictive information in the writings of Twitter users, providing an upper bound on the available predictive information that holds for any predictive or machine learning methods. As few as eight to nine of an individual's contacts are sufficient to obtain predictability comparable to that of the individual alone."

They said: "Distinct temporal and social effects are visible by measuring information flow among social ties, allowing us to better study the dynamics of online activity. Our results have distinct privacy implications. Information is so strongly embedded within a social network that, in principle, one can profile an individual from their available social ties, even when the individual themselves forgoes the platform completely."

In other words, what they're saying is that the people that we are linked to has such strong predictive value that just knowing eight or nine of them and knowing about them is as predictive as knowing everything about us. Which I just thought was an interesting data point, not something that I would have imagined possible. But in this day and age with unlimited computing power and data storage, these are the kinds of mischief that academics are getting up to. So not a big deal. I just thought it was sort of interesting that, yeah, if we know who eight or nine of your contacts are, knowing about - well, and the other thing, too, Leo, is that it strongly suggests the nature of how we choose our friends is they're people who tend to align with us in various ways.

Leo: Yeah, yeah.

Steve: So there has been, boy, I don't know if I have the date where it began, where it was first seen. But like a year? An ongoing - I've seen references to it as I've been dipping around the security world. But nothing really - it didn't really come up to the level of needing to say something because there wasn't anything definitive. It's been a DNS hijacking campaign which has sort of been in the background. It isn't widespread. It's targeted. But it's interesting.

What got it onto my radar was where I saw how the U.S. had reacted to US-CERT's warning about it, finally warning about it last week. So we have the National Cybersecurity and Communications Integration Center, the NCCIC, which is part of the Cybersecurity and Infrastructure Security Agency, the CISA, which is aware of a global Domain Name System, DNS - so we have acronym soup here - infrastructure hijacking campaign. Using compromised credentials - and that's the key, like how they get in.

Using compromised credentials, an attacker can modify the location to which an organization's domain name resources resolve. In other words, they hack into your DNS provider. Which is why, for example, I use Hover, and Hover uses an authenticator. And as annoying as it sometimes is to go, okay, fine, and put in the six-digit code that's only good...

Leo: Oh, I do that. Oh.

Steve: You betcha. It's like, this is probably more than anywhere else. Well, LastPass does it. Hover does it. Google, you know...

Leo: I have Gmail. Whatever your email provider, that's the other one. Yeah, those are the big three, aren't they.

Steve: Yeah. So there are places where it's like, yes, this is annoying, but it's better. It creates a dynamic code that makes it much more difficult to attack. So anyway, but there are lots of DNS providers who don't do that, don't offer that. For example, Network Solutions, no. So thank you very much.

Leo: Among other reasons.

Steve: Yeah, well, exactly. I think I have, like, one, maybe two domains there still. I completely moved off of them, where I was from day one, over to Hover. But the only ones that are there are the ones that only they provide. I just can't get them anywhere else. So it's like [grumbling sounds]. So by compromising someone's DNS login, the attacker points their DNS, their authoritative real DNS to a different IP away from their servers, typically to where they've got a spoofed website already set up in advance.

So this of course enables the attacker to redirect all of the traffic subject to the global DNS caches expiring, as they will over the course of, depending upon how long the records have been allowed to be cached, for maybe eight hours, maybe 24. But traffic immediately begins to flow. I've seen that myself because back when GRC was having to change its IP in order to avoid long-term persistent DDoS attacks, we would change the IP and then be like, nobody there. It's like, okay, hello. We're here. Then, of course, change the DNS record, and then gradually over the course of some number of hours the traffic would begin to pick up as people were like, oh, we don't know where GRC was, but they're back. It's like, well, actually, no, you kept trying to go to the old IP because we changed to a different one, and your DNS only now caught up as a consequence of the fact that DNS is caching.

Anyway, so this is bad. You don't want your DNS to get hijacked for many reasons. But this is one of the things that's been going on. It allows the attacker to redirect the traffic that would normally go to the true servers, to an attacker-controlled infrastructure; and, what's interesting, to obtain valid encryption certificates for an organization's domain names, which then enables man-in-the-middle attacks. And that's one of the things that has sort of raised the level of concern and is where I'm going to focus some of our attention as we talk about this more.

So this NCCIC in their coverage encourages administrators to review FireEye and Cisco Talos Intelligence blogs on global DNS infrastructure hijacking for more information. Additionally, NCCIC recommends the following best practices to help safeguard networks against the threat. Well, I'm skipping that because we're going to talk about what the CISA, which is the subpart of the NCCIC, said, which is where these, like, respond within 10 days or else.

Anyway, I had the links to FireEye's report, which I will share because they've got some detail. They said: "FireEye's Mandiant Incident Response and Intelligence teams have identified a wave of DNS hijacking that has affected dozens of domains belonging to government, telecommunications, and Internet infrastructure entities across the Middle East and North Africa, Europe and North America. While we do not currently link this activity to any tracked group, initial research suggests the actor or actors responsible have a nexus to Iran.

"This campaign has targeted victims across the globe on an almost unprecedented scale, with a high degree of success. We have been tracking this activity for several months, mapping and understanding the innovative tactics, techniques, and procedures" - and now we have a new acronym, the TTPs, the Tactics, Techniques, and Procedures, okay - "deployed by the attacker. We have also worked closely with victims, security organizations, and law enforcement agencies where possible to reduce the impact of the attacks and/or prevent further compromises.

"While this campaign employs some additional tactics, it is differentiated from other Iranian activity we have seen by leveraging DNS hijacking at scale. The attacker uses this technique for their initial foothold, which can then be exploited in a variety of ways. In this blog post we detail the three different ways we have seen DNS records being manipulated to enable victim compromises. Technique one, involving the creation of a Let's Encrypt certificate and changing the 'A' record, was previously documented by Cisco's Talos team. The activity described in their blog post is a subset of the activity we have observed."

Okay. So as I said, let's think about what this means. If a site's DNS record can be changed, then subject to DNS caching expiring and needing to be renewed, all traffic to the domains controlled by the altered DNS record will be redirected to an attacker-controlled IP address. And since that redirection includes the lookups being performed by the Let's Encrypt services, the attacker is able to immediately auto-obtain and auto-install a valid certificate for their fraudulent site at its fraudulent IP. So this, of course, further defrauds every visitor to that site who will see a fully correct https://. Did I spell the domain correctly? Yes, everything is correct. And look, it's HTTPS. It's valid.

So we've often talked about the need to have secure domain name lookup. And as we can see, this was made even more important with the advent of Let's Encrypt because incorrect DNS is Let's Encrypt's greatest weakness. We've raved about the idea of having Let's Encrypt and being able to sort of have the equivalent of opportunistic security for connections, even though Let's Encrypt has weakened the identity guarantee, or I guess I should say it's transferred it entirely to DNS.

Once upon a time, where you had a human in the loop - and you still do, of course, with other types of certificates that provide a much stronger assurance about the identity of the entity to whom you are connecting. What Let's Encrypt has done by completely automating the lookup or the certificate issuance process is it has - essentially it's transferred all of the veracity of its identity assertion over to DNS. And unfortunately, DNS is not up to the task yet. Someday, when DNSSEC is fully deployed, then we'll have secure domains, and this kind of fraud will be dramatically mitigated, if not completely eliminated. We're not there yet.

So we're sort of in this awkward stage where everyone's rushing to HTTPS. Browsers are becoming really militant about not having an HTTPS connection, where developers are being inconvenienced because they can't see into their traffic any longer, which used to be really convenient. So we're having to set up our own proxies in order to intercept traffic, or we're able to typically look at the web browser developer mode in order to see what's going on with our browser. But it is difficult to look at it now on the wire because it's all being encrypted.

So the predictable effect this has had, that is, that Let's Encrypt's automated certificate issuance has had is that what once was relatively useful identity assertion value of an HTTPS certificate has been reduced significantly. And it is interesting, I mean, that these bad guys have taken the time to obtain certificate certs for their DNS hijack domains, which suggests that, as we have seen, HTTPS really has become the de facto must-have property for websites. And certainly the high-value sites they're apparently going after are HTTPS sites. They may be pinned as HTTPS in the web browser using HSTS so you no longer have the option of trying to do, for example, what we've talked about before, an HTTPS to HTTP downgrade attack, where you just switch all the URLs down to HTTP and assume that no one will notice. Browsers just won't do this.

And of course the problem is that Let's Encrypt is now really being strongly compromised. We've talked about this before. Back in March of 2017 we covered the news of an analysis that showed that Let's Encrypt had issued, get this, 15,270 certs containing the string "PayPal."

Leo: Oy.

Steve: Yeah. So that just sort of shows you that...

Leo: No good deed goes unpunished is what that shows you.

Steve: Exactly. Exactly.

Leo: Geez. But it doesn't mean if you use Let's Encrypt that you're vulnerable. It just means people are misusing it.

Steve: Well, yes. It means that because it took the man out of the loop, it took all human verification out of the certificate issuance process and automated, that's both its benefit and its liability. What it means is that we can't really any longer rely on automated certificates to assure us that we're connecting to the authentic domain that we believe we are.

And in fact that takes me to one of the conclusions of this is I'm wondering if web browsers, well, if the certificate standard should not be augmented with a flag that is set, if the certificate was issued automatically, that is, certs that do not have a human intervention that Let's Encrypt, and presumably in the future there will be other automated certificate issuers, if automated certs shouldn't be required to set a flag in the certificate properties indicating that it was issued through automation. And then that would allow browsers to provide some indication of some sort, optionally, that yes, you're secure. But just FYI, this certificate was issued automatically, not with a human oversight. Just, again, as sort of a beacon, not saying that that's bad, but it is subject to abuse, and this is what's happened.

Leo: Isn't that what EV certs, though, are? I mean...

Steve: Yes, yes.

Leo: And they're trusted at a higher level because, I mean, I know DigiCert for our EV certs calls me and asks, I mean, there's a lot of verification.

Steve: Yeah, yeah.

Leo: But even with non-EV there's not that, I mean, they might send an email to check your email; right?

Steve: So there's DV, which is domain validation, which is what this is. Essentially, we always had domain validation certs. But even then you had to go through some hoops. They weren't automated. You had to, like, they would email you a blob to put on your website's home directory, and then you'd say it's there, and then they would go retrieve it from the root of that domain, which would serve to prove that you, the recipient of that email, were in control of that domain. And they said, well, if they're able to put a blob of text we provided them on the domain, then they must be okay to give the cert.

Leo: Let's Encrypt gives us something like - they've tried to do something like that. They have kind of some sort of domain control verification process, automated.

Steve: Oh, yeah. Well, it's automated, and that's the problem is that it's all you need is DNS. If you have DNS - and of course the whole point of this is that DNS is not yet robust. Well, and we could argue that even the system is stronger than the management because this is the attack of the management interface, which is why we were talking about Hover and time-based tokens and how very important it is for people to maintain good control over their DNS records because it's the way these exploits happen.

So in their background that they wrote: "In coordination with government and industry partners, the Department of Homeland Security Cybersecurity and Infrastructure Security Agency (CISA) is tracking a series of incidents involving Domain Name System infrastructure tampering. CISA is aware of multiple executive branch agency domains that were impacted by the tampering campaign and has notified the agencies that maintain them." I mean, these are dot gov. "Using the following techniques, attackers have redirected and intercepted web and mail traffic, and could do so for other networked services.

"One. The attacker begins by compromising user credentials, or obtaining them through alternate means, of an account that can make changes to DNS records. Two. Next, the attacker alters DNS records, like Address (A), Mail Exchange (MX), or Name Server (NS) records, replacing the legitimate address of a service with an address the attacker controls." Actually, changing the name server records, that's diabolical because that would allow them to sort of set up an unobserved persistent ability because those are the records to which the ultimate reference is made. So they could change those then later, at a time of their choosing. So that's kind of tricky.

"This enables them to direct user traffic to their own infrastructure for manipulation or inspection before passing it on to the legitimate service, should they choose." So it allows them to sort of basically set up a clean man-in-the-middle attack. This allows a risk that persists beyond the period of traffic redirection.

"Three. Because the attacker can set DNS record values, they can also obtain valid encryption certificates for an organization's domain names. This allows the redirected traffic to be decrypted, exposing any user-submitted data. Since the certificate is valid for the domain, end users receive no error warnings. To address the significant and imminent risks to agency information and information systems presented by this activity, this emergency directive" - that's what this was, an emergency directive - "requires the following near-term actions to mitigate risks from undiscovered tampering, enabling agencies to prevent illegitimate DNS activity for their domains and detect unauthorized certificates."

So this is required actions. "Within 10 business days for all dot gov or other agency-maintained domains, audit public DNS records on all authoritative and secondary DNS servers to verify they resolve to the intended location. If any do not, report them to CISA. CISA recommends agencies prioritize NS records and those associated with key agency services offered to organizational users and the public. For example, websites that are central to the agency's mission, MX records, and other services with high utilization."

Action two: "Change DNS Account Passwords. Within 10 business days, update the passwords for all accounts on systems that can make changes to your agency's DNS records. CISA recommends the use of password managers to facilitate complex and unique passwords."

Action three: "Add Multi-Factor Authentication to DNS Accounts. Within 10 business days, implement multi-factor authentication for all accounts on systems that can make changes to your agency's DNS records. If MFA cannot be enabled, provide CISA with the names of systems, why it cannot be enabled within the required timeline, and when it can be enabled. CISA recommends using additional factors that are resilient to phishing. Consistent with NIST SP 800-63B, Short Message Service SMS-based multifactor authentication is not recommended."

Action four: "Monitor Certificate Transparency Logs. Within 10 business days, CISA will begin regular delivery of newly added certificates to Certificate Transparency logs for agency domains via the Cyber Hygiene service. Upon receipt, agencies shall immediately begin monitoring CT log data for certificates issued that they did not request. If an agency confirms that a certificate was unauthorized, it must report the certificate to the issuing certificate authority and to CISA."

And then the CISA said: "CISA will provide technical assistance to agencies that report anomalous DNS records. We will review submissions from agencies that cannot implement multifactor authentication on DNS accounts within the timeline and contact agencies as needed. We'll provide regular delivery of newly added certificates to CT logs for agency domains, and we'll provide additional guidance to agencies through an emergency directive coordination call following the issuance of this directive, as well as through individual engagements upon request."

So the government is taking it seriously and basically is taking clearly some affirmative action here to remediate any yet-undiscovered but pending attacks as a consequence of previous or getting underway attacks against the dot gov and other important infrastructure. So it's nice to see something like you've got 10 days to do this, to get this done.

But also interesting that what we're seeing is attacks becoming more sophisticated because they have to be. Basically, coming up with a way to get admin control over DNS, then using that to set up an alternative infrastructure to obtain Let's Encrypt certs through automation and then leverage that in order to create a man in the middle and gain access to data for as long as it can be had.

And of course, you know, lots of communications is trusted. And you can imagine that that is, if you're able to establish yourself in a man-in-the-middle position, for as long as you can keep that you have a pipeline for all of the sensitive information that flows over that. So it's not just the badness of getting yourself into that position, but the huge vulnerability that is created when you're able to see all the data that flows over that link during that time.

Leo: Makes sense, yeah.

Steve: Yeah. So I don't normally talk about iOS and macOS security updates. I have a few times in the past. This one caught my attention just because, when I searched on the word "arbitrary," the page lit up.

Leo: Arbitrary.

Steve: Yes. Because that's the phrase - Apple uses the phrase "arbitrary code execution."

Leo: Ooh. That does not sound good.

Steve: So I believe I heard Rene say last week that he was surprised that the update was to 12.1.3 because he was expecting, I think, it to go to 12.2. He's the Mac guru follower genius guy, so I don't know what that's about. But anyway, what we got was 12.1.3. Presumably, maybe Apple already has other plans for 12.2, and that hasn't happened yet. So these things apply to iPhone 5s and later, iPad Air and later, the iPod Touch sixth generation. And that caught my eye, as I said, because when I searched the security news details page - I have the link here in the show notes for anyone who's interested - for the word "arbitrary," I got...

Leo: There's a lot.

Steve: ...Bluetooth. Yeah. I got Bluetooth: An attacker in a privileged network position may be able to execute arbitrary code. And they describe an out-of-bounds read was addressed with improved input validation. But until then you've got an over-the-air remote code execution vulnerability. In FaceTime a remote attacker may be able to initiate a FaceTime call using arbitrary code execution. That doesn't seem that bad. But still you don't want that. A bunch of kernel impacts. A kernel arbitrary code execution. Those are never good.

There was, in Apple's libxpc, which is part of the iOS process management system, there was an arbitrary code execution. Also another arbitrary code execution in SQLite. WebKit had a bunch, and those are not good because of course that has a lot - WebKit is Internet-facing. So there was paliciously - paliciously.

Leo: Palicious and delicious both.

Steve: Yeah, exactly. Processing maliciously crafted web content may lead to arbitrary code execution. Actually all three of them say that. So there was a memory corruption issue was addressed with improved memory handling. A type confusion issue was addressed with improved memory handling, and multiple memory corruption issues were addressed with improved memory handling. As we know, Apple doesn't ever give us any details. It's just be happy these are no longer going to bite you. And Fluoroacetate was involved. And I remember we talked about him before, or she or whoever, working with Trend Micro's Zero Day Initiative. Fluoroacetate reports to Trend, who then reports to Apple. Also WebRTC, again potentially high impact because that tends to be Internet facing. So there was an arbitrary code execution vulnerability there.

Leo: And for "arbitrary" read "malicious"; right?

Steve: Oh, yeah, yeah. We provide the code that we're going to stuff down your throat whether you like it or not.

Leo: "Arbitrary" sounds so harmless.

Steve: Exactly.

Leo: But it isn't completely arbitrary. It's mostly whatever the bad guy wants to execute.

Steve: Exactly.

Leo: Yeah.

Steve: Exactly. So as we know, iOS has historically been less prone to reverse engineering attacks than Windows. Yet a lot of these seem not good. So I don't know why it is, but my systems update lazily. It'll be like a week will go by, and then something will begin to say, yeah, you know, we'd like to reboot your iPad or your phone or something. And I'll go, oh. Anyway, so this time I went looking, and I was asked if I wanted to download an update. And I said yes, thank you. So I would just suggest to our listeners, I mean, again, probably targeted attacks. As far as we know, well, we don't know whether they are in the wild or not. They weren't disclosed as zero-days, so we can presume they're not. But it would be good to update.

And consequently, or as a consequence of there being a lot of code overlap, an increasing amount of code overlap, and a common code base between iOS and macOS, the macOS update that was Mojave 10.14.3, which was also known as, for High Sierra, they simply called it the "Security Update 2019-001" and the same thing for Sierra, most of those things were also affecting macOS.

Leo: That's a good point. All of us - you're right. Of late you get updates on both iOS and macOS at the same time.

Steve: Yeah, and...

Leo: That's a unified code base. That's interesting, yeah.

Steve: Right. And so the good news is...

Leo: WebKit, of course, WebKit's the same on both, yeah.

Steve: Yes. And the good news is, when you fix it on one, you fix it on the other. The bad news is you've got vulnerabilities on both. So that's not so great.

Leo: Interesting observation, yeah.

Steve: Yeah. There were some additional, well, so first of all, all of the same ones were present in the macOS: Bluetooth, FaceTime, WebRTC, SQLite, the IOKit, and those affecting the kernel. And then additionally, specific only to macOS, Sierra, High Sierra, there was a remote code execution vulnerability in the Intel graphics driver and a privilege elevation issue in the OS's hypervisor. And then all three macOS versions that were patched were affected by an out-of-bounds flaw in the Quartz Core that could allow an attacker to read unrestricted memory. So update now and update often.

And then of course we should mention while we're here that this FaceTime bug which has been much in the news, I mean, the headlines were all screaming with their hair on fire. As I understand it - you guys talked about it on MacBreak just before this podcast - it would be possible for someone to call someone, to use FaceTime to call someone else, then add themselves to the FaceTime group, and as a consequence of a bug in the indexing of the identities in the group management, somehow the recipient of the call, the FaceTime running in the recipient's phone would misunderstand the addition of a redundant group membership to be answering. And as a consequence, without the recipient of the FaceTime call answering, the call would be answered at their end, thus enabling the microphone and camera. Which is a mistake.

So as I said, I expect we will have that fixed within the next few days, I mean, because, again, it got a lot of attention in the press. I had some friends say, hey, you going to be talking about that on the podcast today? It was like, yeah, we will mention that. So not much more to say about it. Were there any specific mitigations, Leo, that Rene and...

Leo: Well, they've already done it. Apple's turned off the Group FaceTime feature. So you can't...

Steve: Okay.

Leo: Yeah, the mitigation is done.

Steve: Okay.

Leo: And, yeah, you could turn off FaceTime if you were really concerned. But I think because Apple's turned off the server, I think you're all right.

Steve: Yeah, yeah, good. So the group service died.

Leo: That's right.

Steve: And as soon as they get the patch pushed out...

Leo: That's exactly right.

Steve: ...then they will turn it back on again. Cool.

Leo: It's, you know, I mean, it's not a good bug. It allows somebody to snoop on you. But it's not a terrible bug. You'll hear the ring. They can only snoop on you for a while until it stops trying to get through.

Steve: And, you know, I have found, Leo, I don't know if there's been any detailed research on this, but the inside of someone's pocket...

Leo: Not a good place.

Steve: It's very dark. It's very dark in there.

Leo: And muffled.

Steve: You may see, depending upon the knit of the fabric, you may see some little bits of light shining through. But probably not, you know, not a bad bug.

Leo: And if you've ever been butt dialed, you know that the contents of that audio isn't usually very useful.

Steve: No, a lot of sort of scratching around and rumbling and, yeah, exactly.

Leo: I mean, it's a bad bug, but they're going to fix it.

Steve: It is, yeah. So at the end of March 2018, a Swiss site, abuse.ch, initiated its most recent project which they call URLhaus, H-A-U-S. I've got a link in the show notes for anyone who wants more detail. And I sort of rewrote their statement to kind of bring it current. It was written sort of in the past tense, and it was written in English probably by non-native English speakers, so their English is way better than my Swiss.

They wrote: "At the end of March 2018, abuse.ch initiated its most recent project called 'URLhaus.' The goal of URLhaus was to collect and share URLs that are being used for distributing malware." And wait till you hear how many. "The project was a huge success. With the help of 265" - and Leo, I had to force myself not to write 256 because my fingers, they just automatically...

Leo: There's a groove in your brain.

Steve: So I don't even, yeah, I was like, okay, wait a minute.

Leo: 2^8. It's obvious.

Steve: Yeah. Not 256, "...265 security researchers spread across the globe, URLhaus was able to coordinate the takedown of almost 100,000 malware distribution sites..."

Leo: Holy cow.

Steve: "...in 10 months. During that time, these 265 researchers identified and submitted" - get this - "in average 300 malware sites to URLhaus each day in order to help others to protect their network and users from malware campaigns. Working with the security community, URLhaus managed to get the attention of many hosting providers, helping them to identify and remediate compromised websites hosted in their network."

They say: "This was not a simple task, specially for large hosting providers who have tens of thousands of customers, and consequently a great many hijacked websites within their network that are being abused by cybercriminals to distribute malware. Nevertheless," they write, "URLhaus in average counts between 4,000 and 5,000 active malware distribution sites every day, which is," they say, "way too much. The following chart shows the number of active malware distribution sites tracked since the launch of URLhaus." And it's sort of sad. The blue line indicates the number of abuse reports sent out to the corresponding hosting providers and network owners.

So what's interesting here is, first of all, for those who are listening and cannot see, the graph spans June 12 of 2018 through January 15 of this year. And it shows in red the active malware sites creeping upwards from at the beginning around 2,500, up to a max of it just barely crosses 7,000. Now, that's not because that many new sites were created. That's the awareness of them that was growing during the project as the 265 globally distributed researchers were feeding 200 to 300 new URL reports per day into URLhaus, who was aggregating them and consolidating them and counting them. So the awareness of these sites kept growing.

Meanwhile, there was an ongoing background effort shown by the blue line to thwart this. And what's interesting is there is, looks like maybe in September, on September 2nd, maybe between the 2nd and the 11th somewhere, that's where the graph is marked, there's a big blue spike up to 3,000 where 3,000 malicious sites were reported during a small period of time. And that sort of seemed to tip the scales. And we then see over the course of September, October, November, the number of sites dwindling, and then another little uptick. And then there's some more reports, and they get pushed down again.

So the point is, this graph shows a real cat-and-mouse ongoing battle between malicious sites being created and then being taken down, then being found, then being reported to URLhaus. Then URLhaus reporting them out to the hosting providers, saying, hey, you probably are not aware of this, but this domain name has got bad stuff on it. You should take it down. It's hurting people.

So then they also have, and I have it copied in the show notes, a table which they say: "The table below shows the top malware hosting networks" - now, again, this is not the network's fault, necessarily, although you could draw your own conclusions - "hosting active malware content, counting online malware distribution sites only as of January 20th, 2019." Okay, so that's very recent. They say: "As you can easily spot, two out of three of the top malware hosting sites are hosted either in the U.S. or China."

And what was interesting, and a little sad, is the length of time it takes for reports to be acted upon. The worst were Chinese sites. What they said is what is also an eye-catcher on this table is the takedown time of malware sites hosted in China. The three top Chinese malware hosting networks have an average abuse desk reaction time of more than a month. So from time that it's reported, four weeks go by before any action is taken. Which is unfortunate.

Although in general it's not good, if I look here for the shortest one, there was actually one of the Chinese networks was three days, 11 hours, and 50 minutes. The worst was one month, 23 days. Okay, so that's way over a month, almost two months. And they had 163 malware URLs. The next biggest was 256 malware URLs. That was a Chinese host that reacted after one month and nine days. On the other hand, the number one hosting site provider was Digital Ocean in the U.S. And they had 307 malware URLs, so more than any other provider. And their reaction time was six days, 12 hours, and 56 minutes. So certainly...

Leo: We should mention, by the way, they're a sponsor, as you know.

Steve: Okay.

Leo: And one of the reasons people use them is it's so easy to spin up a site; right?

Steve: Right.

Leo: It's a simple thing to do.

Steve: Well, yes, exactly. And also I was just going to say that these guys have to be responsible because they don't want to take down, they shouldn't take down a site based on a report without verifying it. So otherwise you've got script kiddies maliciously reporting good sites that they don't like as being malicious, and getting them booted for no good reason.

Leo: Very good point, yeah.

Steve: So when you have a huge number of sites, there's a lot of remediation work and burden that it has that goes along with it. So anyway, so they went on to talk about what malware was found there. And the number one malware by a long shot was something called Emotet, which is a very capable and increasingly flexible trojan which is sort of multipurpose. It gets in, and then it's polymorphic. It changes shape. It's very hard to deal with. And of course the bad guys are constantly churning out new domains to host this stuff and then spew out links in social networks and on download sites and in ads and wherever they can to get people to click on them to download the malware and then go from there. So, boy, unfortunately that's the world that we live in today. Crazy.

Chrome will be playing catch-up to IE and Firefox when it comes to mitigating drive-by downloads from iframes. Web browser iframes, and we've talked about them, have always been frightening from a security standpoint. We often talk about the classic tradeoff between security and flexibility. Nothing could be a better example of that than the iframe. iFrame, as we know, is short for inline frame. It allows the designer of a web page to set aside a rectangular region, a frame, whose contents will be filled in by the result of an iframe URL fetch.

So the origin web page specifies the URL. Then the browser goes to fetch it and to render it sort of as a mini web page unto itself. And they are, iframes are what have enabled the entire web browser advertising industry, since they conveniently allow web pages to monetize themselves by agreeing to set aside space, these physical frames, which will be filled in by advertising aggregators and for which the originating websites then are paid, based upon the number of times those iframe URLs are pulled and displayed.

The danger is that, unless controlled and restricted, what are essentially mini web pages are full browser citizens capable of loading anything they choose and, as we've often talked about, running JavaScript, whatever JavaScript they may wish. So way back in Internet time, on October 6th of 2015, the Chromium bugs list contained the observation - and I've got the link for anyone who's interested. This is the Chromium bugs list.

"IE and Firefox do not allow download from a sandboxed iframe." And then the posting included with this sample shows a simple little, I mean, like, minimal HTML. You know, it shows a <!DOCTYPE>, then <html> opened, the <head> open, the </head> closed, the <body> tag opened, and then <iframe sandbox src> and then a URL, where the action is equal download and then close the </iframe>, close the </body>, close the </html>. Basically just an iframe. And back then, IE and Firefox were already not downloading content from a sandbox tagged iframe. That was flagged as a bug in Chromium on October 6th of 2015, and it has remained outstanding ever since.

The good news is, although Chrome still has not, that does appear finally to be changing. Various tech news outlets are reporting that Google's developers of Chrome have finally started working on adding drive-by download protection to Chromium. The new feature is already active in the current Chrome Canary edition build and is scheduled to land in the stable version with Chrome 73 sometime in March or April. Analysis has shown that when downloads are triggered in a web page's iframe element, hidden in its code, those downloads are almost always malicious. Yeah, no kidding. I mean, like, why are you downloading a file in an iframe? That really should be restricted. And this is the number one way that malvertising is still crawling into people's machines.

So I guess it was out of an abundance of, well, who knows. Maybe Google is using this extensively in their own web apps. It wouldn't surprise me. Certainly there are use cases for it. But the typical use, unfortunately, has been malicious. So the Chromium developers stated: "We plan to prevent downloads in sandboxed iframes that lack a user gesture, and this restriction could be lifted via an 'allow-downloads-without-user-activation' keyword, if present in the sandbox attribute list."

So basically the idea would be they're going to finally block by default, but allow permission if the designer of the web page intends for the iframe, the sandboxed iframe, to allow un-user-initiated downloads. So that solution makes sense since you can imagine there might be instances where a web page might wish to allow content within an iframe to have download privileges. So looks like we will be getting that at long last in Chrome, which is all for the better.

Now, this is where anyone who is responsible for a Cisco WAN VPN router, an RV320 or an RV325, pauses the podcast, takes those routers offline, and updates their firmware. You can come back to the podcast; but you are absolutely, please, you are absolutely permitted to go unplug your router. And if you're streaming the podcast, well, then, yes, you'll get disconnected. But believe me, you'll be glad.

Two days after Cisco released patches, last Wednesday, security researcher David Davidson published proof of concept exploit demo code on GitHub. I have the link to his publication in the show notes. His GitHub posting was titled "CVE-2019-1652 and -1653 Exploits for Dumping Cisco RV320 Configurations and Debugging Data and Remote Root Exploit!" So attacks against these routers began shortly after David's code went public. Okay. So first of all, the two vulnerabilities are horrendous. This 1653 allows a remote attacker to get sensitive device configuration details without a password. How convenient. 1652 allows a remote attacker to inject and run admin commands on the device without a password.

David's GitHub posting could not have been more seductive and damaging. I mean, he, like, left nothing to the imagination. He called it "Cisco RV320 Dump." He cited the CVEs and said "Exploits for Dumping Cisco RV320 Configurations and getting RCE." In other words, remote code execution. And he said: "Implementations of the" - and he quoted the two CVEs - "exploits disclosed by Red Team Pentesting."

He says: "I only tested these on an RV320; but according to the Cisco advisory, the RV325 is also vulnerable. The following Shodan queries appear to find them. If you're curious about how many are out there, there seems to be quite a few." Then he gives four links to the Shodan queries to find the routers. And so he's using ssl:RV320, so that's going to be port 443 of course. So RV320 and RV325. And then also port 161, that's the SNMP port. So that RV325 and 320.

He says: "The vulnerabilities allow for the following: Dumping in plaintext the configuration file, including hashing for the web UI. Dumping encrypted diagnostic/debug files, including config, and the /etc and /var directories. Decrypting the encrypted diagnostic/debug files." And he says, "Yes, you get /etc/shadow. And post-authentication remote command injection as root via the exposed web UI." And then he says: "As an aside, the default creds are cisco:cisco." In other words, username and password, cisco:cisco.

So Troy Mursch at Bad Packets Report did some white hat scanning and put up some more details last Saturday, the day after David's disclosure. He posted on January 26: "Over 9,000 Cisco RV320/RV325 routers are vulnerable to CVE-2019-1653." He wrote: "On Friday, January 25, 2019, our honeypots detected opportunistic scanning activity from multiple hosts targeting Cisco Small Business RV320 and RV325 routers. A vulnerability exists in these routers that allow remote unauthenticated information disclosure via 1653, leading to remote code execution via 1652." Yeah.

"Using data provided by BinaryEdge, we've scanned 15,309 unique IPv4 hosts and determined 9,657 Cisco RV [and we know the number] routers are vulnerable to 1653. That's the information disclosure. 6,249 out of the 9,852 Cisco RV320 routers scanned are vulnerable." And he said in parens: "(1,650 are not vulnerable, and 1,955 did not respond to our scans). Of the RV320, 16,247 are vulnerable, RV320 routers are vulnerable. 3,410 out of 5,457 RV325 routers scanned are vulnerable." So this is a disaster. They are sitting there, wide open. Cisco released patches two days before they got scanned, the scanning began. So we know thousands of them, if not almost all of them, are going to be subject to attack and compromise. They are very popular among enterprise and ISP providers.

So as I said, if you have any RV320 or 325 routers within your purview, you're responsible for them, just go unplug them. Update them and get them plugged in again. I hope that the owners of those were on a mailing list, took this very seriously if they received email from Cisco, and didn't say, okay, yeah, we'll get around to that next week because it took two days. And again, this is the world we live in now.

Okay. Some miscellany. Last Sunday's first presentation of the completed and finished SQRL system went well. I had previously given presentations of SQRL at DigiCert's security summit and then to Stina and her crypto colleagues at Yubico. Each of those points in time caught SQRL where it was then. Today, as I've been saying recently, it's finished. So the meet-up of what is known as LETHAL, the L.A. Ethical Hackers and Leets, was my first opportunity to publicly present the entire system from soup to nuts.

There is a video that was made, and it was sort of a best effort. We turned the room lighting down so that everyone in the room could see the screen, but that left me pretty much in the dark. And I think the battery died a couple times. I mean, it was just an amateur video. I have a link to it through Filemail. I shared it over in the newsgroup this morning, and I got a couple comments back saying, yeah, you know, you were great, but I guess there are two big audio dropouts and so forth.

So anyway, I'm sure I will eventually speak in front of a group that records video all the time, and we will end up with a good presentation. This one was - it's not what I want to do with you, Leo, and with Jason and with Mike Elgan. I want to do a user-facing presentation, like what does the user see, and then also answer all of your "but what about" questions.

What I did with this group, because they were a bunch of techies, was to do - it was a technical presentation. But to that end, during the presentation I kept referring to GRC's online documentation, but also needing to continually apologize that those pages are now more than five years old, and as a consequence much has changed since then. So my own next top priority, like what I'm going to be doing tonight, is to begin the process of rereading those, revising them, and synchronizing with the way it actually turned out so that those are caught up, and so that from now on, when I'm referring people, like oh, yeah, all the documentation for this is online, it actually will be, instead of being, yeah, well, you know, the way I thought it was going to be five years ago because it's radically improved and tightened up since then.

And in the meantime, our wonderful XenForo developer sent me yesterday his version 1.0.0 of the SQRL integration for XenForo using the new SQRL service provider API. It is a formal XenForo add-on. XenForo is the web forum software I chose which has a mature add-on architecture. And so, for example, this would allow anyone who drops this onto their XenForo forums to add SQRL authentication, just like it would take 10 minutes. You would need an SSP API server because, you know, the SQRL Service Provider. Right now I'm the only one that exists, and mine is for Windows. Probably runs 32-bit Apache, too, because I wrote it as an ISAPI, you know, the IIS API, which IIS and Apache both support - although mine is 32 bits, and IIS allows you to mix bitness, 32- or 64-bit, but Apache doesn't. You have to have a 32-bit Apache to run 32-bit modules. And of course I wrote mine in assembly language. We've got a guy in the newsgroup who is in the process of recoding it in Open Portable C. So as soon as we have that, then we'll have something that we can compile under multiple platforms to really make adding SQRL very, very simple. So every day we get closer.

Leo: Woohoo.

Steve: Tonight I will drop SQRL into our forums, and we will begin the process with the guys in the newsgroups checking it out. And then I hope to very soon be able to tell our podcast listeners about it; and, Leo, to schedule my visit to your studios to have a professional recording of how SQRL works.

Leo: I promise our battery will not die.

Steve: And you can actually see me. Oh.

Leo: And there'll be light on your face.

Steve: So, okay. This caught me by surprise. And Joey, his name is Joey Kelley. He's a listener. He said: "Hello, Steve and the rest of the GRC team." He said: "I purchased SpinRite some time ago and have had few occasions, thankfully, to use it." He says: "Most of my customers," he says, "I own a small consulting firm on the side, have been convinced of the need to back up their data; and, even when a hard drive fails, it is usually an inconvenience, not a total disaster.

"Earlier today, I tweeted you a quick picture entitled 'Dynastat Engaged!' showing SpinRite going to work on a drive that I am glad to say was successfully helped by your product. Since the story behind this is interesting, I thought I would relay it to you." And so, yeah, you just showed the picture on the video, Leo. It just shows - and I went back and captured it, just so I could put his picture in.

He said: "In the past couple of years I have become quite good friends with a couple and their family that run a store and lunch counter near where my parents live. I swing in often, and I've tweaked their computers here and there as asked. About two weeks ago they mentioned they had an old computer in the back that they would like to have the data from. That old story, repeated countless times between you and Leo on Security Now!, began to play in my head, ending with that well remembered line: '...and there is no backup.'

"Knowing that these people have had a lot of knocks in their lives, I thought I'd help them out and wound up taking the old computer home. I set it up through a USB-to-IDE adapter and set SpinRite to going. It found the typical one unrecoverable sector in the early sectors of the drive, did some recovery, and then finished. However, it would still not come up in either Windows or Linux as a valid drive. Figuring I had nothing to lose, I reran SpinRite on it with a Level 4 scan, and only four hours later the scan completed. I was able to pull the drive up in both Linux and Windows, and was able to copy all of their data off the drive. Success.

"Then something caught my eye, the name of one of the User folders. Almost 15 years ago, these folks lost their 14-year-old son in an accident. This was the family's computer at the time, and it contains photos that they have nowhere else of their son, which they now have back, thanks to SpinRite." He said: "Thank you for giving this family their memories back."

Leo: Very nice.

Steve: So, Joey, thank you for sharing that. His URL is JoeyFixesComputers.com. And he said: "You have my permission to use this email as you see fit."

Leo: Oh. That is a great story. That must make you feel very good.

Steve: It very much does, yeah. Yeah. We have someone, Tony West, who's watching. He tweeted: "Watching Security Now! with @SGgrc for the first time. Is that the biggest coffee cup ever, or a camera illusion?"

Leo: Well, show us your coffee cup, Steve. It's no illusion, my friend.

Steve: And of course, yes, hello.

Leo: That's the 10-inch coffee cup.

Steve: That's right. It's that the lens of course is so close.

Leo: But wait a minute, though. Pull it back because it's still as big as your head.

Steve: Yeah.

Leo: If you're drinking coffee out of a cup as big as your head, it's a big cup.

Steve: Leo, it's a two-hour podcast. What am I going to do? I didn't want to run out.

Leo: I love those.

Steve: So, browser extension security. We have the persistent tradeoff between capability, flexibility, and security. And it's been a theme of mine that I've noted to call this a "persistent" tradeoff, rather than "inevitable," because I think that using today's computer technology and today's computer architecture, we're pretty much stuck with trading off one for the other, you know, security versus freedom. But I think that's only due to the way we're currently solving these problems. And, okay. Yours is bigger than mine, Leo.

Leo: It's a TWiT mug.

Steve: Oh, my goodness.

Leo: Sorry, I didn't mean to interrupt.

Steve: That's all right. The only problem is you pour hot coffee in that, and it turns into cold coffee.

Leo: Pretty quick. Surface area.

Steve: So much volume to heat up, yeah. Anyway, so I believe that it's only due to the way that we're currently solving these problems that we are in the dilemma we are. I think it's inertia holding us back and that we will eventually have, I don't know what it's going to be, but hopefully we'll see it in our lifetime - I think we probably will - a rethinking of the way we do this because, basically, we're not doing anything differently than we were back in the mainframe days when machines with lights like that were blinking. So but for now, here and today, we clearly do face a tradeoff. And this tradeoff, which Google and Chrome, which of course is now the number one web browser worldwide, that's what the tradeoff is continually struggling with.

They're currently working to evolve the interface, essentially the API, offered to third-party browser extensions. And that interface determines, obviously, what the hooks are that the extension has into the browser, what it's able to do. We've been at what they call Manifest v2 because an extension contains a manifest where it declares what features and services it needs, and then the browser interacts with it based on its declaration of stuff that it wants to be able to do. The Chromium team is currently headed toward Manifest v3. But in this struggle, this tradeoff of security versus freedom, it's predictably ruffling some feathers.

ZDNet noted in their coverage, they had an article titled "Chrome API update will kill a bunch of other extensions, not just ad blockers. Chrome extensions for antivirus products, parental control enforcement, and various privacy-enhancing services will also be affected." And Bleeping Computer had two separate postings, noting that our podcast favorite, uBlock Origin, may die. And I think we must have referred to that in the last couple of weeks, Leo, because that sounds familiar to me, that we were talking about uBlock Origin being endangered.

Leo: Yeah, well, it doesn't work on Safari unless you go to the GitHub page and so forth. And I guess this is maybe an issue there. But I hope, and I presume, Gorhill is going to do something about that. I mean, that's a big, big deal.

Steve: Yeah. Well, and the other thing that may die is Tampermonkey, which I thought, what? Apparently it's as popular as uBlock Origin. Well, so both of Raymond Hill's, a.k.a. Gorhill, as you referred to him, extremely popular Chrome extensions, uBlock Origin and uMatrix, would, as he understands it, die. If the changes happen...

Leo: He can't fix it?

Steve: Unh-unh.

Leo: Whoa.

Steve: Well, and we don't know how grumbly he's being. But he said if the...

Leo: Yeah, he's pretty grumbly.

Steve: Yes, if the changes happen as they're currently defined in the next version draft of this v3 Manifest. So in the forthcoming v3, the Chrome developers have stated their intention to limit the blocking capabilities of something known as the webRequest API, which Raymond's extensions require. The current proposal reads: "In Manifest v3 we will strive to limit the blocking version of webRequest, potentially removing blocking options from most events, making them observational only. Content blockers should instead use something known as 'declarativeNetRequest.'"

Raymond said that phasing out 'webRequest' API in favor of the 'declarativeNetRequest' API would mean the death of uBlock Origin, which is used by over 10 million users on Chrome. He wrote to the bug tracking page where this Manifest V3 work is being discussed. He said: "If this (quite limited) declarativeNetRequest API ends up being the only way content blockers can accomplish their duty, this essentially means that the two content blockers I have maintained for years, uBlock Origin," and he says, parens, "(uBO) and uMatrix can no longer exist." In his posting, he explained that his extensions are incompatible with the proposed declarativeNetRequest API.

Leo: Oh.

Steve: Uh-huh. Because it allows for only one specific filtering engine, whereas uBlock Origin and uMatrix rely on various filtering designs to do their job properly. The proposed modification is more oriented toward more limited fixed static filtering capabilities such as those provided by Adblock Plus. But the redesign would also limit the number of filters to 30,000, which while that sounds like a lot, is insufficient even for Adblock Plus. Ray uses the example of the EasyList filters with rules for removing unwanted web content, which is currently larger than 30,000 entries and is not sufficient for, he says, a modern user's filtering needs. The EasyList rule set is used by both Adblock Plus and uBlock Origin and is much larger than the limit imposed by this declarativeNetRequest.

So the good news is, and I'm so happy that we have this, I mean, there is a forum that exists where this stuff is being seen. I mean, you know, there's interaction between the devs and the counter devs, those who are pushing back against. And so, as we'll see in the second big topic here for the end of the podcast, there are security consequences to this. But, and I get it that Google isn't, we know, removing things without cause. But there is a tradeoff in our current model, the way things work today, of security and freedom.

So what about Tampermonkey? This was the first I had encountered the term. And I thought, what the heck is Tampermonkey? Well, first of all, it is nothing like Honey Monkey, so disabuse yourself of any belief. But I have heard of Greasemonkey, and it is like Greasemonkey. So although I've been unaware of it forever, over 10 million users are making use of this. And I don't know who they are, but they are pretty much super techies. Okay. So what all of these things are - Greasemonkey, Tampermonkey, there's also one called Violentmonkey, I don't know why, but anyway. [Crosstalk]. It does. So those are the three big things. They're called userscripts managers.

Leo: Yeah. Greasemonkey's been around for years.

Steve: Yeah. Yeah, yeah, yeah. In fact, Greasemonkey predates Chrome. It was available on Firefox in the old days.

Leo: Right.

Steve: So what these all are, and it's like, okay, I thought uBlock Origin gave me all the control I needed. No. These things allow scripts to be added, injected, userscripts to be injected into websites, the pages of websites, in order to alter web browser behavior, to tweak it for whatever goal or benefit the user has. Apparently Tampermonkey is currently now at the top of the heap, the most popular userscript manager. Chrome supports Tampermonkey or Violentmonkey. Apparently not Greasemonkey. Firefox supports all three - Greasemonkey, Tampermonkey, Violentmonkey. Safari does support Tampermonkey. Microsoft Edge, Tampermonkey. Opera, Tampermonkey or Violentmonkey. Maxthon, only Violentmonkey. Dolphin, only Tampermonkey. And so on. So anyway, Tampermonkey pretty much has it.

So the way Tampermonkey describes itself, it says: "Tampermonkey makes it very easy to manage your userscripts" - I didn't know my userscripts needed management, but I'm learning - "and provides features like a clear overview over the running scripts" - which you would of course want if you wanted your userscripts to be managed - "a built-in editor, ZIP-based import and export, automatic update checks, and browser and cloud storage-based synchronization." All the things you would want from a good Tampermonkey.

However, userscripts, they say - oh, no. This is me talking because I did some browsing around, and I found a horrible downside. Userscripts are very powerful. And, of course, the script repositories have been overrun with malicious scripts hoping to get themselves injected into an unwitting user's browser. So it should not be surprising that Google devs are wondering whether this whole thing is a good idea. In his Google group's posting, Jan Biniok, Tampermonkey's creator, writes, he says: "Hi, Chromium developers." And "Hi, Devlin," who communicated with him via email. He says: "I'm the Tampermonkey developer, and I have not studied all the planned changes in detail yet. But this is the one that worries me most."

And then he quotes, saying: "Beginning with Manifest v3, we will disallow extensions from remotely hosted code. This will require that" - which, okay, sounds good to me. But Google says: "This will require that all code executed by the extension be present in the extension's package uploaded to the web store. Server communication, potentially changing extension behavior, will still be allowed. This will help us better review the extensions uploaded and keep our users safe. We will leverage a minimum required CSP [Content Security Policy] to help enforce this, though it will not be 100% unpreventable, and we will require policy and manual review enforcement, as well."

So, okay. Just stepping back for a second, basically they're saying we're not going to - we're considering in v3 disallowing an extension's freedom to just reach out and download code and run it at will. That's like, yeah. Anyway, so then of course Jan, who is upset about this, says: "While the text above might be interpreted in a way that an extension like Tampermonkey can continue to exist, I got the following explanation from Devlin in an email."

And Devlin writes, and apparently he's with Google because he says "we": "Note that we will be limiting remotely hosted/arbitrary code execution" - there's that phrase again, Leo - "in all contexts. The goal is that we should be able to perform an in-depth security review of an extension and be confident in what it does and whether it poses a security or privacy risk to users, which is possible through web page contexts, as well. But let's move this conversation to another thread," says Devlin.

So then Jan says: "I understand the need for security, but this means that v3 P1, in the way it's currently planned, will stop Tampermonkey from working entirely because arbitrary code execution is Tampermonkey's main functionality. Every little userscript would then have to become an own extension." Or he probably means its own extension. "Anyone who wants to do that has to pay $5 to be able to publish an extension. There are so many use cases" - well, and god, that would bury Google under random arbitrary extension storm - "many use cases for userscripts, so I hope that this planned change is reconsidered. One possibility would be, for example, a new permission that relaxes this constraint and allows remote code execution again. All extensions with this permission could then be provided with a special warning and be examined more intensively. What do you think?"

He finishes: "I've been working on Tampermonkey since Chrome v4 or 5, and I could not live without it anymore." And I'll just comment that somehow I have been living without it. But I certainly understand, I mean, once upon a time I was living without JavaScript, too, with NoScript. But those days are gone.

And to Jan's posting, somebody replied: "Jan, this new remotely hosted code restriction also affects my project, a process automation/RPA system. I asked the team about it at the Chrome Dev Summit last November, bringing up Tampermonkey as an example of a productive and popular use of remotely hosted code, arguing that a permission showing a big scary warning on install should be adequate. Their response was that remote code is too big a threat vector. Extensions can do too much harm, and even an extension that starts out benevolent might be later compromised. They seem pretty committed to the decision."

He said: "My solution is to output a browser extension for each customer, one containing all the automations/userscripts they use and want. If you're interested in this kind of approach for Tampermonkey, I'd be happy to collaborate. It avoids the new restriction, but does complicate the script update process, requiring the user to rebuild periodically. I have some ideas for reducing rebuild frequency," blah blah blah blah blah.

So those are perfect examples of where, I mean, do we want a web extension to explicitly open our browser to remote code execution, which Google has seen hugely abused and wants to eliminate. I don't know.

Leo: Hmm. You have to do what's secure. I mean...

Steve: Yeah, yeah. Google is, I mean, like, maybe stronger sandboxing so that you can't hurt yourself? But as we'll see, even that doesn't solve the problem because it turns out there are very - there's a lot of power that extensions can use within the browser, which takes me to the final piece of this.

The paper was shortened EmPoWeb, short for Empowering Web Applications with Browser Extensions. A French security researcher - boy, and there's no way I'm going to pronounce his name, Doliere Francis Some - took a long look at the security implications, I guess his team because he says "we," maybe, of the extreme powers which are currently given to browser extensions.

His abstract reads - I think it's a 19-page paper: "Browser extensions are third-party programs, tightly integrated into browsers, where they execute with elevated privileges in order to provide users with additional functionalities. Unlike web applications" - okay, and he uses that term throughout, so let's make sure we understand it. Web applications are web pages. They're JavaScript apps running on a page, like, you know, editing documents, spreadsheets, Google docs sorts of stuff. That's a web app.

So "Unlike web apps, extensions are not subject to the Same Origin Policy and therefore can read and write user data on any web application. They also have access to sensitive user information including browser history, bookmarks, credentials (cookies), and list of installed extensions. They have access to a permanent storage in which they can store data as long as they're installed in the user's browser. They can trigger the download of arbitrary files and save them on the user's device.

"For security reasons, browser extensions and web applications are executed in separate contexts." Okay. For security reasons. So there was an attempt at isolation. He says: "Nonetheless, in all major browsers, extensions and web applications can interact by exchanging messages." And I would argue they have to. I mean, otherwise there's no point in having them. "Through these communication channels, a web application can exploit extension privileged capabilities and thereby access and exfiltrate sensitive user information.

"In this work we analyzed the communication interfaces exposed to web applications by Chrome, Firefox, and Opera browser extensions. As a result, we identified many extensions that web applications can exploit to access privileged capabilities. Through extensions' APIs, web applications can bypass Same Origin Policy and access user data on any other web application, access user credentials (cookies), browsing history, bookmarks, list of installed extensions, extensions storage, and download and save arbitrary files in the user's device.

"Our results demonstrate that the communications between browser extensions and web applications pose serious security and privacy threats to browsers, web applications, and more importantly to users. We discuss countermeasures and proposals, and believe that our study and in particular the tool we used to detect and exploit these threats can be used as part of extensions review process by browser vendors to help them identify and fix the aforementioned problems in extensions."

So I'm going to skip this - he provided a wonderfully detailed 19-page PDF. I have the links here. So I have just two last paragraphs. They're long, but - well, they're not, they're medium-length - but that I snipped from toward the end of this after all of the underlying foundation is laid down. But these essentially are what he found.

He says: "We built a static analyzer and applied it to the message passing interfaces exposed by Google Chrome, Firefox, and Opera extensions to web apps. When the tool found that a privileged extension capability could potentially be exploited by web applications, the extension was flagged as suspicious. By manually reviewing the code of flagged suspicious extensions, we found that 197 of them, mostly on Chrome, can be exploited by web applications" - meaning just arbitrary websites - "to access elevated browser features and APIs and sensitive user information."

Okay. So just to pause for a minute, that means, if you have any of these 197 different extensions installed and visit a site that is aware of these, the site can leverage the extension behind your back to attack you.

Okay. So they continue: "The extensions we have found have vulnerabilities that can be exploited by web applications [i.e., web pages] to, one, break the privilege separation between extensions and web applications and execute arbitrary code in the extensions context," which of course is privileged. "Two, bypass the Same Origin Policy and access user data on other applications," that is, other web pages. "Three, read user cookies and use them to mount session hijacking attacks." In other words, the return of Fire - what was that thing, Firesheep? Basically, if you went to a malicious site, it could get your Google session cookie, your Amazon session cookie. The idea that we're all being statically logged on is convenient for us, but it represents a liability if anything else can get those cookies. And this allows that.

"Read user cookies and use them to mount session hijacking attacks. Access data such as user browsing history, bookmarks, list of installed extensions that besides violating user privacy can be used for tracking purposes. Five, store and retrieve data from extensions persistent storage for tracking; trigger the download of malicious software on the user's device whose execution can then damage user data." So that's what they found.

In their conclusions they mentioned their disclosure to vendors, and they said: "We have disclosed the list of extensions to Chrome, Firefox, and Opera. All vendors acknowledge the issues. Firefox has removed all the reported extensions. Opera has also removed all the extensions but two, which can be exploited to trigger downloads. The reason given by Opera is that the downloads can only be triggered from specific websites. However, we made them observe that those websites include third-party scripts that can also trigger arbitrary downloads. So discussion still continues with Opera on the two remaining extensions, in particular to ensure that users are aware of the downloads. Chrome also acknowledged the problem in the reported extensions. We are still discussing with them on potential actions to take, either remove or fix the extensions."

So a beautiful piece of research, and important. So what this says is that there are nearly 200 extensions, most on Chrome, where the capabilities snuck past the extension reviewers. And if a page wished to, it could leverage the presence of the extension in order to do some or all of these things, depending upon what the extension is doing. So it is the case that we have an important tradeoff here between capability and security. We've talked before about how incredibly useful some of our browser extensions are. We don't want to lose them.

And Leo, I was about to mention when we were talking about the first portion of this that maybe Chrome could have like an expert user version, or expert user mode. Maybe Chrome as the mainstream browser has to eliminate some of the flexibility if it cannot provide it securely. But we could still have a browser like Firefox that would have a different user profile, that would allow some of these things moving forward.

I don't know how this is going to sort out. It would be nice if we could have all of this, have this kind of capability and power and security. The fact that only 197 extensions are problematic suggests that it's possible to do what these things do without them being exploitable. So maybe that's the solution is just focus, you know, because god knows there's a bazillion extensions for Chrome. And if these guys, out of that, only found 197 that were causing this kind of trouble, one would think those could be fixed.

So anyway, it does suggest that we've got a ways to go. I don't know what v3 will mean. I would hate to lose uBlock Origin over on Chrome. I would hate for Firefox to follow because it's where I mostly depend upon uBlock Origin. And, boy, the web is no fun without something like this to tame the pages to some degree.

Leo: I think increasingly people are - this doesn't solve it on mobile, but at home are using router-based ad blocking, like the Pi-Hole or Eero or Plume. Both of my routers have ad blocking in the router.

Steve: I don't know how you could do that over HTTPS.

Leo: Yeah, I guess you couldn't, could you, because you can't see into it, yeah.

Steve: Yeah, right.

Leo: So that's not going to be a long-term solution.

Steve: Although they could do DNS blocking. So if they refuse...

Leo: Yeah, they block sites, yeah, the sources.

Steve: Exactly.

Leo: And think that's mostly what they do. So you can't go to ads.syndication.google or whatever.

Steve: The problem, of course, then is you get all these little broken things all over the page.

Leo: Yeah, holes. I know. You get a lot of holes. That's, yeah, that's how Pi-Hole works.

Steve: Okay.

Leo: Are there browsers that - do you think at some point there'll be - I don't think Google's going to do an inexpert version of Chrome. But at some point Firefox or some other browser might be the expert's browser.

Steve: Yes. I mean, I think that. I could see that being the way this evolves is that, you know, Firefox wants to hold onto a niche, and that would be a beautiful niche for them to have.

Leo: Yeah.

Steve: And Lord knows, if there are 10 million people that have downloaded honky-tonk monkey, what was that thing called?

Leo: You'd better not mess with me.

Steve: Tampermonkey? I don't know who, but 10 million people are injecting scripts into other - okay, well, you know, you guys really have your propellers wound pretty tight, I guess.

Leo: At this point, you know, it's really Chromium versus the rest of the world. And so, you know, in fact one Microsoft developer said Firefox should just stop, just give up and join the rest of the world so there'll be one web standard, Chromium, now that Edge is going with Chromium. I completely disagree. And this is an opportunity for Firefox to find a niche.

Steve: And the Mozilla team is staying, I mean, at par. They're keeping up in performance and speed, and they're staying with the standards. They still have the best side tabs. I can't, you know, I mean, there's nothing better than that. So, yeah. I'm glad they're hanging in there.

Leo: Although, if people say it's less secure, that's not going to be good.

Steve: What's less secure?

Leo: Well, because they support these extensions, that they're more vulnerable. That wouldn't be good.

Steve: Yeah.

Leo: So I don't know.

Steve: Everybody I know uses Chrome. That's just what they use.

Leo: I think you said this already, but I just want to make it clear. There's no implication that Google's doing this because they don't want ad blockers. This is a security thing.

Steve: No, no. In fact, Adblock Plus, because it's a simple monotonic filter, could continue to be used. But apparently they're trying to limit the number of filter rules that can be imposed. So if that happens, then we'll have to choose the most important 30,000, believe it or not, and maybe lose some of the obscure ones. But no, this would allow ad blocking, just not the...

Leo: Not Gorhill style.

Steve: Exactly, where it's doing all - it's like a multifactor filter where it's hooked in all over the place and doing all kinds of - allowing you to have a multimodal filter of different kinds of things.

Leo: Yeah. Well, this will be interesting to watch as it develops. Steve, as always...

Steve: I'm just glad we have a dialogue. As long as there's discussion, there's hope.

Leo: Yeah, yeah. And this is a proposal from Google. This isn't written in stone yet.

Steve: Yup, yup, and they're discussing it.

Leo: Yeah. And they're hearing, I hope.

Steve: And we're done discussing it, Leo.

Leo: We're done. We're going to go home now. Steve's home is GRC.com. That's where you'll find SpinRite, the world's best hard drive maintenance and recovery utility, saving family photos all over the world. You'll also find this podcast. He's got audio and transcripts there at GRC.com, and all his freebies, like SQRL. GRC.com. You can tweet at him at @SGgrc. In fact, he'll take DMs from anybody, crazy man. So if you've got a tip or a question, do it there or at GRC.com/feedback.

You can get audio and video versions of the show at our website, TWiT.tv/sn for Security Now!. Or subscribe on your favorite podcast application. That's the best way to do it. That way you'll get every minute. And you can complete your set, have all 700 next week, 700 Security Now! episodes.

We do the show every Tuesday, 1:30 Pacific, 4:30 Eastern, 21:30 UTC. Come by and watch it live at TWiT.tv/live, join us in the chatroom at irc.twit.tv, or download at your convenience. We don't care how you listen, as long as you listen. We'll see you next time on Security Now!. Bye-bye, Steve.

Leo: Thanks, Leo.