
# The 25 Most Attacked Vulnerabilities

**Description:** This week we examine a recently patched zero-day in Chrome and a nice new feature in that browser. We look at the site isolation coming soon to Firefox, and Microsoft's announcement of Edge for Linux. We have some movement in the further deprecation of Internet Explorer, and a potentially massive SQL injection attack that was recently dodged by more than one million WordPress sites, despite the fact that some admins complained. Then we have a bit of miscellany, closing-the-loop feedback, and an update on my work on SpinRite. We end by looking at the NSA's recently published list of the top 25 network vulnerabilities being used by malicious Chinese state actors to attack U.S. assets.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-790.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-790-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. This week on Security Now!, a zero-day in Chrome. Better get your patch running. Firefox and some new security measures to sandbox, to isolate processes. We'll talk about the top 25 Chinese attacks on the United States according to the NSA. Another flaw in WordPress you'll want to patch immediately, if it hasn't already been patched for you. And a whole lot more. It's coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 790, recorded Tuesday, October 27th, 2020: The Top 25 vulnerabilities.

It's time for Security Now!, the show where we cover your privacy, your security, your safety online with this guy right here, Steve Gibson of the Gibson Research Corporation. Hello, Steve.

**Steve Gibson:** Yo, Leo. Great to back with you again for Episode 790, our last episode of October 2020. And those who are watching the video will notice that I'm coming to everyone from our alternate location. I was working on the podcast, like a couple hours to go, and all of my connectivity disappeared. Later, when I got over here, I logged onto Cox, which is my cable modem provider, and I got a kick out of their notice.

They said: "We've noticed a temporary outage related to your Internet, TV, and phone service." So first of all, they noticed a temporary outage? They noticed an outage which they hope is temporary. And of course I'm hoping that, too. And they said the estimated time to repair, 4:13 p.m. So of course we record this between 1:30 and 2:00 typically. We're pretty much wrapping up around that time. So I thought, I think maybe I need to set up the alternate location. So I did so.

**Leo:** That's twice now in like a month or something; right?

**Steve:** Yeah, yeah. And of course, thanks to the nature of broadband in the U.S., I have no alternative provider. I mean, they're nominally pretty good. But again, as you said, Leo, it's getting to be like, okay, wait a minute. On the other hand, we did go for 15 years without any trouble, although remember the old days with two T1s?

**Leo:** Oh, yeah, yeah, yeah.

**Steve:** Those were the days.

**Leo:** Two T1s that gave you a fraction of the speed you're getting with a single cable connection.

**Steve:** Yes, exactly, exactly. The only thing it really had going for it was it had a lot of reliability.

**Leo:** It was reliable, that's right.

**Steve:** Because, yeah, back then it was twisted pair, and they were so expensive that it was a high priority to keep them up and running.

**Leo:** Yeah, 1.44 megabits.

**Steve:** Uh-huh.

**Leo:** Ridiculous. Why did we think that was a lot?

**Steve:** Yeah. And I remember, too, when I initially, like back in the day when I was getting it, I would mention that to people. And this is when of course people had dial-up, and they were like, you have a T1? Oh. It's like, okay, yeah, yeah. But fortunately technology moves on.

We've got a bunch of stuff, speaking of technology moving on, to talk about. There's a recently patched zero-day which hit Chrome, only the third in a year. Fortunately Chrome doesn't have many. But when it does, they tend to be significant because of course, as we know, it's the majority browser now on the Internet. There's also a nice new feature that Chrome is offering we'll talk about. We're going to take a look at site isolation, which is coming soon to Firefox, and Microsoft's announcement of Edge for Linux, which, okay, we'll talk about that.

We've got some movement in the further deprecation of IE, of course, Internet Explorer; and a potentially massive SQL injection attack that was recently dodged by more than one million WordPress sites. So here again, WordPress, we just keep talking about this. And as everyone knows, I shut down mine because it's like, no. I'm just crazy to do that.

And despite the fact that this was automatically remediated, some admins of their own sites complained about the fact that they were just saved. Anyway, we'll talk about that.

We've got some miscellany, a bit of closing-the-loop feedback, and an update on where I stand with SpinRite. And then we're going to end by looking at the NSA's recently published list of the top 25 network vulnerabilities currently in use by malicious Chinese state actors to attack assets in the U.S.

**Leo:** How many of them were leaked NSA vulnerabilities?

**Steve:** Was BlueKeep a...

**Leo:** Yeah, yeah.

**Steve:** Yeah, one of them is BlueKeep. And so I did see that.

**Leo:** I bet they didn't put that in the footnotes: "Oh, by the way, we wrote that one, mm-hmm."

**Steve:** And we do have a fun Picture of the Week that we will get to.

**Leo:** All right, Steve. Let's get to the Picture of the Day. I see it here.

**Steve:** So it's six frames, and it just caught my funny bone.

**Leo:** It's a conversation I've had many times, actually.

**Steve:** So we've got two cartoon characters standing in front of a table at the beginning. The first one approaches, and the second one is busy writing something. First one says, "Are you doing programming for fun?" And the other character says, "Yeah, I love being mentally challenged." To which the first one replies, "Well, I'm glad you've come to terms with it." And the first one says, "Thanks." And then we have him thinking about that for a minute, and then he scowls. It's like, wait a minute. What do you mean I've come to terms with being mentally challenged?

**Leo:** Yes. I like being mentally challenged. I do, yeah, mm-hmm. Oh, boy.

**Steve:** So anyway, speaking of being challenged, the Hacker News summed this up by writing: "Attention readers. If you are using Google Chrome browser on your Windows, Mac, or Linux computers, you need to update your web browsing software immediately to the latest version Google released earlier today." And this was last week. So even though that was last Tuesday, even my own always sort of sluggish Chrome had already updated. But this one, our listeners may just want to make sure that they are now running 86.0.4240.111. However, there's much more to last week's emergency update than what drove it. But we'll start with that.

So last Tuesday's release closed five vulnerabilities. Four were rated high severity; one was medium. And one of those four high-severity vulnerabilities was what we were just talking about, was a zero-day that was seen exploited in the wild, being exploited by attackers who were using it to hijack targeted computers. So that nasty one, it was numbered CVE-2020-15999. And what's significant is that it's a heap buffer overflow in FreeType, which is the widely used, open source, font rendering library which is part of Chrome, but many other things. Various bounty payouts were or will be made for the other four vulnerabilities, but this biggie was discovered in-house by Google's Project Zero researcher Sergei Glazunov.

But even so, even though it was found in-house, it was subjected to an accelerated seven-day public disclosure release deadline because the flaw was under active exploitation. And that's the Project Zero guidelines. You get 30 days for things that, like, yeah, you've got to get these things fixed. But if it's being used, if it is a zero-day, you get a week. As it happens, this only took one day for Google to begin pushing the update, which they did on the 20th. It was discovered on the 19th. They were pushing the fix one day later. Which is interesting because it wasn't even really their problem. It was in the FreeType library, not in Chrome.

Sergei immediately notified the FreeType developers, who also developed an emergency patch to address the issue and had it available the next day, on October 20th. And so that's FreeType 2.10.4. This is significant because FreeType is everywhere. Without revealing details of the vulnerability, Ben Hawkes, who is Project Zero's technical lead, warned via Twitter that while the team has only spotted an exploit targeting Chrome users, it's absolutely possible that other projects that use any earlier versions of the FreeType library, and there will be roughly a gazillion, might also be vulnerable and are advised to deploy the fix included with FreeType v2.10.4.

He tweeted: "While we only saw an exploit for Chrome, other users of FreeType should adopt the fix discussed here." And then he provided a link in his tweet. I've got the link here in the show notes. And it is part of the stable release of FreeType, again, 2.10.4. So what we do know, thanks to what Sergei has shared, is that the vulnerability exists in FreeType's function "Load_SBit_Png." So it's Load_SBit_Png. Which processes PNG images embedded into fonts. It can be exploited by attackers to execute arbitrary code just by using specially crafted fonts with embedded PNG images, which turns out to be something that FreeType supports. So not just curved glyphs, but you can embed images.

And since web fonts can be specified by a web page, and since the browser will go download the font and then render glyphs from those fonts, turning a theoretical FreeType flaw into an active exploit would not be difficult. Our listeners may remember that way back when I created - remember the Off the Grid cipher, which was based upon a Latin Square? I wanted to allow the user to choose from among a library of highly recognizable and blocky fonts. So I purchased a bunch of fonts, web fonts, for the purpose. And I used the web fonts system to render them in the user's browser. Meaning that I had all those fonts online. The style sheet on the page specified where the browser should go download that font, which it then did, and rendered on the page. Meaning that none of this is difficult to do.

Sergei added, he said: "The issue" - and get this, Leo. I mean, this is just so classic. "The issue is that libpng uses the original 32-bit values, which are saved in 'png_struct.' Therefore, if the original width and/or height are greater than 65535..."

**Leo:** They have overflow.

**Steve:** Uh-huh. That's of course the maximum value you can store in 16 bits is 65535, "...the allocated buffer won't be able to fit the bitmap." Sergei has also published a font file with a proof-of-concept exploit. All of this increases the urgency of updating anything that uses the FreeType font renderer in a way that would allow an attacker to provide their own malicious font.

In this "Load_SBit_Png" function, it obtains the image width and height from the image header as 32-bit integers because that's the size of height and width in the PNG image header. It then truncates, unfortunately, the obtained values to 16 bits in order to store them in the TrueType SBit Metrics structure, which only has room for 16 bits. Therein lies the problem. It then uses the stored truncated values to calculate the bitmap size, which of course will no longer be correct because they got chopped down to 16 bits. It then allocates memory of that size for the image backing store into which it will load the image as it renders it. And then it passes the "png_struct" and the backing store handle to the libpng function. Meaning that, yup, not that difficult to exploit.

We also know that this bug was introduced in the June 9th, 2015 release of FreeType v2.6. So it's been in every subsequent FreeType release for the past more than five years. So that's a big deal. Compared to that beauty, the other four things that were fixed in Chrome are relative yawners. Three of them are also ranked as severe. One's an implementation bug in Blink. Another is a use-after-free bug in Chrome's media handling. There's also a use-after-free bug in PDFium, which is part of Chrome. The remaining medium severity flaw is another use-after-free issue in Chrome's printing subsystem.

And so as I mentioned, this is a zero-day. This got fixed last Tuesday. Everybody wants to make sure they're running the latest Chrome, and I would imagine, I mean, clearly the fact that I already had it and my Chrome never seems to be in a hurry to update, means that Google turned the intensity up on this one because, again, this one was being abused in the wild and was seen there. And I mentioned Google has had three. Almost a year ago there was the critical remote code execution vulnerability patched last Halloween night. And the other was a memory confusion type bug fixed in February. So these things are few and far between. And of course thanks to Chrome's mature self-maintenance system, the browser, which would normally be the target for malicious abuse, it fixed it itself.

But, and this is the important takeaway, that may not be the case for every other use of FreeType. Anywhere an attacker can access a FreeType library built after June 2015 and not updated last week, who can arrange to render their own font glyphs under FreeType, is potentially exploitable with powerful consequence. FreeType is the font renderer in Android, in iOS and macOS. Java uses FreeType, as does the Sony PlayStation. Many open desktop operating systems and videogames use it. I mean, it is the font renderer of choice, and it's been vulnerable to this for five years.

So keep an eye out for FreeType updates. I imagine that the various Linux distros, because they're all FreeType based, that they will be pushing or probably already have because, I mean, this thing is like a fix-it-now level problem. Again, if there's no way for a bad guy to get your system to render a custom font, you don't have a problem because that's the bar that has to be passed. But many systems like just any web browser will do so. So it's an important thing to fix. And Chrome is sort of the canary in the coalmine. It's good that it was found because that system was able to get itself fixed immediately, thanks to Chrome being in constant contact with the Internet. And we'll be talking about this a little bit in a minute about WordPress because there's another serious problem that happened there.

Also in this Chrome, which is 86, it's now started to block I guess what I would call "slippery notifications." I've mentioned before that I was taken aback when some site I visited prompted me to - it said "enable and allow site-based notifications." Or that's

what I knew it was asking me to do. But in this case it was before it would allow me to proceed into the site it showed like on the page an arrow pointing up to where the permission to enable site-wide notifications appeared, and it said something like, "Please click 'Allow' to enable this website and proceed."

**Leo:** No, unh-unh. Bye-bye, website.

**Steve:** Yeah, exactly. I hit my back button on the browser and chose another link from the search engine that brought me to this misbegotten site. But of course we know what we're doing. Many users are going to get caught out by this. So the problem is that most users don't realize that this is entirely bogus and unnecessary and that what they're doing is giving that site permission to harass them with notifications which appear to be originating from their operating system, since the browser, whether they are on that site still or not, forms a conduit for subsequent messaging spam.

So fortunately Google has noticed this, too. And what's more, they've observed sites using notifications for active malicious purposes, including sending malware to or mimicking system messages attempting to obtain user login credentials. So, I mean, again, this just seems like a bad idea, the whole thing, the whole idea of like this background messaging thing. But maybe that's just me. Google's Web Platform Project Manager PJ McLachlan said: "Abusive notification prompts are one of the top user complaints we receive about Chrome."

So, starting with v86, Chrome will be automatically suppressing website notification spam on all sites which have shown a pattern of sending abusive notification content to their visitors. PJ said: "Our goal with these changes is to improve the experience for Chrome users and to reduce the incentive for abusing sites to misuse the web notifications feature." And this has been on Google's radar since v80 of Chrome. So they've been working on it kind of quietly in the background to refine its operation. And they've been a bit crafty.

Google's web spiders will subscribe to push notifications if push permissions are requested when the spider visits the site. That allows the spidering to predetect websites which may be misusing these notifications for the purpose of subsequently spamming their previous visitors to the site. And if such behavior is noticed, the spiders will use Google's safe browsing blacklist service to evaluate received notifications and automatically flag the site as abusing notifications for unwanted purposes, at which point that message goes out Chrome-wide, and notifications will just shut down from that site.

So as I said, I've never been a big fan of offsite browser push notifications. I suppose there are valid use cases for them. And as we become more browser-centric, more of our life gets spent doing browser-based apps that I guess receiving asynchronous notifications might make sense. But mostly I find I want to interact with a site by going there. And then when I close that tab, it's like okay, fine, I'm done with this now. Again, maybe I'm just a cranky old guy. But I've not seen a big use for them. What about you, Leo? Has that...

**Leo:** There are some. If you use Google Calendar, the web version can then notify you...

**Steve:** Like a scheduled event coming up.

**Leo:** A schedule. So there are some. If you, for instance, want breaking news, you know, you probably have it on your phone turned on on CNN or Google. You can also have it on your desktop. It's similar to that, but most sites, you're right, I don't want notifications. I really don't. If you use Gmail as your email, and you'd like to be notified when there's a new inbox message, that kind of thing. They're really limited applications. It's annoying as heck. And I agree with you. What I do is go in the browser.

**Steve:** Globally?

**Leo:** Firefox browsers and say, just don't ask me.

**Steve:** Yes, yes.

**Leo:** I don't want it, yeah.

**Steve:** Yes. So speaking of Firefox, site isolation is - we've touched on it. It's a state-of-the-art browser security feature which offers enhanced protection against some forms of security flaws by reducing each browser tab's attack surface. So that's always a good thing. And in practice it reduces the likelihood that malicious code on one site might be able to access the resources of another.

Now, of course we know that browsers, by enforcing the same-origin policy, already prohibit websites from accessing each other's data. But mistakes happen. Security bugs occasionally arise, or sometimes new and creative ways are found to bypass these permissions. This is one of the consequences is that they just keep adding more and more features to these browsers. Bad guys look at that and go, oh, you know, we can leverage this in some way that wasn't foreseen by the people who said oh, yeah, we want to compete with an operating system desktop, so let's just do everything.

So site isolation provides another perimeter of defense to make these sorts of attacks much less likely to succeed by placing web pages from different web domains into different operating system processes. The browser is able to leverage the underlying OS's native and time-hardened inter-process isolation. We already rely on our operating systems to keep processes contained and not misbehaving. So if you break the browser's tabs into processes, then on one hand you get a lot more processes. On the other hand there's already hundreds running in a modern operating system doing all kinds of stuff. But you get more isolation.

We talked about this initially two years ago when Google added site isolation to Chrome in the middle of 2018. That was Chrome 67. And after seeing that feature's success in Chrome, the following February 2019 Mozilla announced their own plans to bring site isolation to Firefox. I've been waiting for it ever since. They named their internal re-architecting project Fission, since it splits Firefox apart into separate processes. But doing this is much easier said than done when you're starting from a model where all the tabs live in a single browser process. So it took both companies, both Google and Firefox, or Mozilla, about two years in a time-consuming rewrite of large portions of each browser's internal architecture.

But Firefox is finally emerging from the other side of that effort. According to an update to the project's Fission wiki page, site isolation can now be enabled in versions of Firefox Nightly for those who like to live on the bleeding edge of browser development. I don't.

I'm happy to wait for it to come to the release channel. But you go to the about:config page. If you have Firefox Nightly, if you then search for "Fission," then you'll find fission.autostart. You set that to true. If you also then search for gfx.webrender, set webrender.all to true, don't change anything else, just those guys.

Restart the browser. And then if you open a few tabs and hover your mouse pointer over the tab, what you normally get is a pop-up description, like a long-form description of the title of the page. What you'll then find is at the end is the system's process ID, the operating system process ID for the process that that page is running in, as the tool tip pop-up text over the form.

So anyway, I'm glad to have Firefox moving forward. Clearly this sort of per-process isolation, you could imagine, if it took two years to re-architect the browser, this is the kind of thing they really wish that they had done, like, from the get-go because it's not an easy thing to change, like on the fly, downstream. But nice to have for us users. And I imagine it'll be - I'm sure I'll mention it as soon as it hits the standard stable distribution channel.

One last piece of news. I have a couple more, actually. But as we know, Microsoft's Chromium-based Edge browser, now available on Windows and Mac, iOS and Android. The only major platform that it was missing from has been Linux. The dev channel build of Edge for Linux is now available for download and installation. And it's running all of Edge's new features, including smooth scrolling, Google's extensions, themes, and so forth. So I just sort of wanted to give people a quick heads-up about that. Since Linux can already run Chrome or Firefox, and I always seem to be running Firefox on Linux, I'm unsure why anyone would want to run Edge there. But maybe developers, for example, might need to verify their compatibility with their stuff under Edge for Linux. So in any event, it's there for anyone who wants it. And I'm sure it'll be also emerging from the dev channel in due course.

One last piece of browser news. Inertia, as we know, in the computer world is confounding. Despite years of Microsoft pushing their users away from the increasingly antiquated Internet Explorer. IE still commands, believe it or not, a 5%, like one in 20, market share. It's like, wow. And the need to move away is not just for security, although it is. We keep running across weird VBScript problems that only IE invokes. Anyway, security should be enough reason. It turns out that IE has petrified. The rest of the web has moved on. Microsoft now maintains a list of 1,156 web domains - you might want to bring this up, Leo, I've got the link here in the show notes - 1,156 web domains which no longer work under Internet Explorer. And this list includes...

**Leo:** This is clearly wrong. I don't know what happened.

**Steve:** Oh, yes. When I brought mine up, I had one per line.

**Leo:** It may not work in Firefox. I'll try it in a different browser.

**Steve:** I think it might have been up because I compose in Google Docs. So some of the things that I bring up are going to be in Google Docs. But anyway, this list of 1,156 web domains which no longer work with IE includes Twitter, Facebook, Instagram, GoDaddy, Google Drive, Google Earth.

**Leo:** What? Geez, everything.

**Steve:** Microsoft Teams, yes, ESPN, Yahoo Mail, and a great many more mainstream and some obscure sites. There were some, like, how did you even know this didn't work?

**Leo:** Oh, this is an XML file, Steve.

**Steve:** Yes.

**Leo:** Oh, I get it, okay.

**Steve:** Yeah, because that file is being pulled by a Browser Helper Object. We've not spoken about BHOs, Browser Helper Objects, for quite a while. Microsoft will be installing a Browser Helper Object into any remaining and still surviving instances of IE.

**Leo:** Geez, there's a lot. I'm just amazed, yeah.

**Steve:** Yeah. And when one of these BHO-enhanced instances of Internet Explorer attempts to bring up any of those incompatible sites, Edge will be launched instead, along with a not-very-subtle banner recommending that the user switch to making Edge their default web browser. So this helps Microsoft. And Microsoft can't be blamed for the fact that some people just won't give up IE. It's a little difficult to understand now why someone would be feeling that way. If you're on Windows, I guess it was on XP, you weren't able to go beyond IE9, which was really old. But then so is XP. IE11 is where it left off. But again, it just won't run a lot of things anymore.

So I thought it was sort of nice that Microsoft is saying, okay, you could still launch IE. Instead of things just not working, like crashing, we'll see if it's a domain that we know about. If so, nothing we can do but move you over to Edge, which they will do, and then suggest that the user just use Edge from now on. So I thought that was kind of cool.

**Leo:** That's amazing. Wow, look at that.

**Steve:** Yeah. I mean, as I said, IE has clearly fossilized. It's just no longer functioning, no longer able to do what the web wants to. Of course it runs GRC.com just fine because my site is also petrified.

Anyway, WordPress once again, and not in a good way, in the news. In this week's installment of why you never want to host your own WordPress site, and also why you should really try to only run with the barest minimum of plugin add-ons, we now have the Loginizer add-on, which is installed in more than one million WordPress sites, which the WordPress security team took the rare step last week of forcing an update to, using a nearly unknown internal capability of WordPress.

It turns out that WordPress sites can be forcibly updated without their administrator's permission. And as a result, sites running the Loginizer plugin were forcibly updated to v1.6.4. Earlier versions of Loginizer contained a SQL injection bug that could have allowed hackers to completely take over any of those more than one million WordPress sites. So why would more than one million WordPress instances choose to install Loginizer? Because it promised...

**Leo:** Good question.

**Steve:** Yes, exactly. Because it promised to enhance the security of WordPress sites...

**Leo:** There you go.

**Steve:** Yeah, by providing IP address, black or whitelists for accessing WordPress's login page. Among other login-related features as, you know, Loginizer sounds like it's close to Agonizer, as it turns out, but in this case provides support for two-factor authentication or simple CAPTCHAs to block login automation. I've got a link in the show notes to the Loginizer page at WordPress.org. And it does, yes, it offers other things. A lot of them are rather obscure. My feeling is those things, most of those seem like features that should be built in natively so that users are not forced to download and obtain them through what prove to be insecure add-ons. I'm 100% certain that the authors of these plugins are well-meaning. But all of the evidence we've seen informs us that writing web-facing browser add-ons is not simple. It requires extreme awareness of security, and it should not be left to random, even well-intentioned, authors.

So I get it that WordPress wants to have an active and vital add-on ecosystem to enhance their platform's offering. But leaving popular features missing, such as at least some of those provided by Loginizer, hugely increases the footprint of exactly this sort of devastating event. Many of those more than one million potential attack targets could have been completely sidestepped from the start if WordPress natively offered the more obvious popular and missing benefits of that add-on. So it ought to be in the box.

While I was hosting my own WordPress site, I also employed my own IP-based filter for exactly that inherently vulnerable and abuse-prone login page. When I was setting up WordPress, I was somewhat stunned that here was just like this login page, and you were depending upon a username and password to get access into the internals of your site. But there was no other protection for it. So of course, because I'm running on top of IIS, I added some rules to a web.config file which is the equivalent of the .htaccess file typically supported by Unix servers. Easily done if you're the admin of the server, not if you don't have access down at that level.

But yes, my point is that ought to be built into WordPress. You should not - no one should need to depend upon lower level plumbing like I was able to do, or the need to go get an add-on, no matter how popular it is, which then exposes more than a million sites to a SQL injection attack. And what's even more weird is that it turns out this Internet-wide forced update was not without controversy. WordPress users have unfortunately been given the impression, well, okay, because there's a setting, that they control what's done when on their sites. There's a permission setting for allowing auto-updates. So of course I immediately turned that on when I was setting up my WordPress installation more than a year ago.

But many old-school WordPress admins enjoy the illusion that, if they have more control, things will go better for them. They are of course wrong. Ryan Dewhurst is the founder and CEO of WPScan. We've mentioned him in the past because he just sort of pops up whenever one of these horrible WordPress things happen. He was interviewed about this. He noted that the flaw "...allows anyone with some basic command-line skills to completely compromise a WordPress website." And he pointed out that the flaw's discoverer had also provided a simple proof-of-concept script in a detailed write-up which was also recently published. This was responsibly posted five days after the Internet-wide update was pushed to all known sites. So telling the world what he had found, what

the researcher who found this had found was fine because it had already been remediated.

But imagine allowing a known and powerful SQL injection attack to linger on more than one million WordPress sites. That bug is one of the worst security issues discovered in WordPress plugins in recent years, which probably explains why the WordPress team decided to forcibly push the patch to all affected sites. Had I been using that plugin, had I not already had "Yes, please update me all the time" turned on, and were I still using WordPress, which is not the case any longer, as I've said, then I would want that thing pushed out to me before it went public.

Ryan told ZDNet in their interview of him, he said this forced plugin update feature has been present in the WordPress codebase since v3.7, which was released seven years ago, in 2013. He believed that it's been rarely used. But that's not entirely clear.

He said: "A vulnerability I myself discovered in the popular Yoast SEO WordPress plugin back in 2015 was forcibly updated." He said: "Although the one I discovered was not nearly as dangerous as the one discovered within the Loginizer WordPress plugin." He said: "I'm not aware of any other cases of forced plugin updates, but it's very likely that there have been others." And confirming that, WordPress's core developer Samuel Wood said that the feature has been used many times, but declined to provide additional details about other instances. So maybe on a less widespread basis, not a million-plus instances, which of course really brought it to the attention of the foreground.

As I said, not everyone was happy with WordPress's unilateral move. Not long after the Loginizer 1.6.4 patch started reaching WordPress sites, users began complaining on the plugin's forum at WordPress.org. One disgruntled user posted: "Loginizer has been updated from 1.6.3 to 1.6.4 automatically, although I had NOT" - he had it all caps - "activated this new WordPress option. How is it possible?" Another added: "I have the same question. It has happened on three websites I look after, of which none of them have been set to auto-update." Okay, so first of all, why do you have websites with auto-update turned off? What is wrong with you? Okay, that was rhetorical, of course.

And in a follow-up, Loginizer's developer said that the security patch had reached 89% of all sites, thanks to the forced update. So oops, there are still, what, 11% unpatched? I hope they get patched soon. There's 100,000-plus WordPress websites with a well-known, now there's a proof of concept out, SQL injection vulnerability just hanging out in the breeze. I imagine those sites will be found and compromised. They must somehow be sites of which WordPress.org was not aware.

All of this podcast's listeners know that anything that's connected to the Internet needs to have some highly reliable means of being updated by its originator, by its source, when important problems are inevitably found. As we know, there is now a growing body, for example, of IoT devices that have been stranded. They've got known vulnerabilities, increasing numbers of them over time. And none of them will ever be updated. That just can't be the way things happen moving forward.

So my feeling is that WordPress should commandeer the best ideas from their add-on authors and implement them safely and securely. There ought to be a built-in IP-based whitelist and blacklist. There ought to be built-in multifactor authentication and CAPTCHAs. You shouldn't have to download an add-on in order to have those basic security features. I get it that they don't want to hurt their own developer ecosystem. But their reputation is going to get more tarnished if they don't fix these problems. And how can they because we've just got amateur, well-intended web script, PHP and JavaScript authors, doing the best they can; but that's not good enough, as we continually see, because we're seeing lots of WordPress problems.

And WordPress should firmly disabuse its platform's administrators of the idea that they can prevent the automatic updating of anything when WordPress feels that it is sufficiently important for things that are vulnerable on their WordPress installation to be updated. The guy who complained that three of his vulnerable sites were updated without his permission to remove a critical complete site takeover should pull his head out of you know where, and WordPress ought to also make their ability to do this part of their formal policy. They ought to have, like, maybe change the auto-update switch to auto-update critical security vulnerabilities or auto-update all, accept all updates, or only critical security vulnerabilities, rather than giving anyone the illusion that it can be completely turned off. Clearly, for seven years it cannot be completely turned off. Updates are going to be done.

I put this last piece under Miscellany because it's less directly related to security than it is to masochism. The news is that the refusal of a system to perform a Windows 10 feature update may now be bypassed. Thus, yes. You know, if things have been going well lately, and your systems have been running without problems, and maybe you're a bit bored, then forcing Windows 10 to update against its will may be just the thing to return some excitement to your life.

Yes, it's true. Microsoft has added a new Group Policy option that allows users to bypass the "safeguard holds," as they're called, which are placed on devices due to known conflicts with hardware or software on that machine. Because, you know, maybe Microsoft is being overprotective, and you'll get lucky. It could happen. This past month's October Patch Tuesday added a new Group Policy titled "Disable Safeguards for Feature Updates." Which is where in a chorus we all say: "What could possibly go wrong?"

**Leo:** Well, because it's Group Policy, it's probably for business users, for an IT guy who knows, has tested and says, no, no, this isn't a problem or something like that.

**Steve:** Yes, because we really need that - and I heard you and Mary Jo and Paul talking and saying that there really wasn't much that happened in H2.

**Leo:** That's a good thing in H2.

**Steve:** Yes.

**Leo:** Because it was really a bug fix for 04, which was a nasty, nasty update. So that's a good thing in this case.

**Steve:** Yeah.

**Leo:** Less is better.

**Steve:** So for those who are interested, it's located under Computer Configuration > Administrative Templates > Windows Components > Windows Update and, Leo, as you predicted, Windows Update for Business. So it's there.

**Leo:** Don't do it at home. Most people are not using Group Policy Edit at home.

**Steve:** Well, actually everybody has it. So, for example, any Windows 10 Pro.

**Leo:** The Pro versions have it, yeah.

**Steve:** Yeah, yeah. So you are able to go to gpedit.msc, and that'll bring it up. And there are some things that I've done in there where I wanted to tweak the way Windows, you know...

**Leo:** No, it's a useful tool, yeah.

**Steve:** Yeah, yeah.

**Leo:** But don't force the update, please.

**Steve:** No. Again, I just couldn't resist. It's like, really? You want to allow people to do this? Because, yeah, what could possibly happen? So anyway, I know that some of our listeners like to walk on the wild side. Here's your ticket, if things have been going too well for you lately.

I had a nice bit of feedback for Closing the Loop from a DavidMD. He said: "Hi, Steve. A little late here, but AdFind, typically pronounced A-D-Find, as in Active Directory Find" - and remember we talked about this last week in the context of Ryuk because that was one of the tools that the forensics examination found being used. It had been downloaded from that site that was pretty much full of Active Directory stuff.

Anyway, David provided a little background. He said: "It's an awesome admin tool for those of us in corporate Windows environments." He said: "I actually know Joe. We used to work together at HP, and he's been a Microsoft MVP for Active Directory for over a decade. He knows his structured directories very well. AdFind can even be used to query OpenLDAP directories." He said: "It's unfortunate that his tools have been used by malicious parties, but that seems to be more a function of the utility of his tools rather than a reflection on his motives." And of course I completely agree. We never meant to imply that Joe had any but the best of uses in mind. Of course, you know, anything can be turned for a malign purpose.

Anyway, he said: "I just wanted to speak up and defend Joe, though I think he'd be the first to agree with you on his website design chops." He said: "Probably why his tools are all CLI. Love the podcast. Keep up the great work."

**Leo:** Command Line Interface, yeah.

**Steve:** That's right. So thank you, David, for providing that bit of background.

**Leo:** It's a lot easier to write command line tools than have GUIs, I have to admit.

**Steve:** It is indeed. So a little quickie on SpinRite. The work on SpinRite is progressing very well. We're getting tantalizingly close to the first release candidate of the ReadSpeed Benchmark. During this past week I significantly rewrote a chunk of the benchmark that will be important for SpinRite. As I mentioned before, I'm implementing as much of the technology that SpinRite will need as I can now, rather than later, so that more of it can be tested sort of in vivo as part of the ReadSpeed Benchmark. The new maximum speed hardware drivers can enumerate and talk to all of a system's drives and controllers, but there will still be some drives that this new code doesn't yet handle, such as USB-connected devices. That'll be added immediately following the addition of UEFI booting and operation for SpinRite.

But I wanted to get directly connected parallel IDE and all serial SATA drives running at lightning speed first. And that means synchronizing the BIOS's view of the system's drives with the new view that this code provides from the PCI bus. And all of that work now appears to be working much better than I had hoped and expected as a result of the testing that we did in the last few days. So I'm working with a couple of testers to eliminate some of the one-off side effects of that rewrite. Then we'll move to the first release candidate of ReadSpeed.

Also GRC's web forums are bubbling along nicely. We've got more than 3,700 registered users, and generally around 80 or so visitors just poking around and looking, you know, reading the threads for which you don't have to register. Although I originally created the forums as a place for GRC project and product support, there was so much interest in just hanging out and chatting that I ended up creating an array of topic-centric groups to better organize the community that has been forming. So now we have a dedicated Security Now! podcast forum, and also forums for discussing security, networking, operating systems, software, hardware, coding, health, nerd recreation, sci-fi, and one titled "None of the Above."

So anyway, that's also turned into a fun place to hang. And once the ReadSpeed Benchmark is ready, that'll be the place to go to report and discuss your results and to tell me about any problems that you might encounter. The whole point of this is to essentially perform an early verification of SpinRite's forthcoming technology. If ReadSpeed works on your systems, from booting through obtaining results, then so will the next release of SpinRite. So we're getting close. And I'm being eliminated by the glow of my screen.

**Leo:** Yeah, it's very bright in there. Can you turn the brightness down? Are you in a nuclear reactor? I don't know what's going on. No, that's all right.

**Steve:** It's whitening my teeth.

**Leo:** Yeah, look at that. Yeah, they're going to look beautiful.

**Steve:** So last week the U.S. National Security Agency, our NSA, published a detailed report enumerating the top 25 well-known security vulnerabilities that are currently being consistently scanned, targeted, and exploited by Chinese state-sponsored hacking groups. And I don't know why they picked on China because we know that China's not alone.

**Leo:** I know why. Don't you?

**Steve:** I do, too. But we'll just move past that.

**Leo:** It's pronounced Chi-na.

**Steve:** So anyone who follows this podcast will already know or presume it is today's security reality, which might have been a good name for this podcast, actually, that all 25 of these vulnerabilities are well-known and have had patches available from their vendors typically for months or years. There's one that is still being attacked that is Symantec's we'll get to. It was fixed in 2017. Yet it's on the top 25 list, amazingly. The development of exploits for many of these is of course hugely aided by the nature of vulnerability disclosure these days; right? The vulnerability is found. It's responsibly disclosed, often.

For example, Zerologon, which is this horrible problem that we're suffering through right now, was found responsibly disclosed. Microsoft patched it. The group that found it, I can't remember their name, but they waited six weeks. Two Patch Tuesdays went by before they finally talked about it. That wasn't long enough because they then provided enough information that bad guys said, oh, thank you very much, we're just going to attack all those systems that have not been updated in the last two months.

**Leo:** That's the problem. You can give the company time to fix it, to offer a patch. But people have to apply the patch.

**Steve:** Yes.

**Leo:** And they don't.

**Steve:** Yes, which is exactly why in my previous WordPress rant I was saying, yes, WordPress should be able to push out something like this, and no one should complain. People should be saying "Thank you, WordPress, for protecting me," rather than "How dare you touch my site?"

**Leo:** We only recommend buying IoT devices that automatically update for that very reason; right?

**Steve:** Yup. So of course then proofs of concept are published, making it trivial to create working exploits. And it's not just state-sponsored Chinese hackers who are making hay. We've seen many of these flaws incorporated into the attack kits of ransomware gangs, malware groups, and nation-state actors from other countries, including Russia and Iran.

Okay. So introducing this six-page list, the NSA said: "One of the greatest threats to U.S. National Security Systems, the U.S. Defense Industrial Base, and Department of Defense information networks is Chinese state-sponsored malicious cyber activity." Okay. One of. Yeah, one of many. Because of course everybody knows about these. These are not just secret things. Notice that they all have CVEs. The top 25 are all known. It's not like they're secret things that China figured out that we don't have a number for. No, they're all patched, except not.

Anyway, they said: "These networks often undergo a full array of tactics and techniques used by" - again, uh-huh - "Chinese state-sponsored cyber actors to exploit computer networks of interest that hold sensitive intellectual property, economic, political, and military information. Since these techniques include exploitation of publicly known vulnerabilities, it is critical that network defenders prioritize patching and mitigation efforts."

They said: "The same process for planning the exploitation of a computer network by any sophisticated cyber actor is used by Chinese state-sponsored hackers." Boy, they really do want to focus on China. "They often first identify a target, gather technical information on the target, identify any vulnerabilities associated with the target, develop or re-use an exploit for those vulnerabilities, and then launch their exploitation operation."

And they finish, saying: "This advisory provides Common Vulnerabilities and Exposures (CVEs) known to be recently leveraged, or scanned-for, by Chinese state-sponsored cyber actors to enable successful hacking operations against a multitude of victim networks. Most of the vulnerabilities listed below can be exploited to gain initial access to victim networks using products that are directly accessible from the Internet and act as gateways to internal networks." And, finally, "The majority of the vulnerable products are either used for remote access or for external web services, and should be prioritized for immediate patching."

And note that the idea of providing a whitelist of IPs which can log into a WordPress site, for example, is a perfect instance. You're trying to protect your login page, which doesn't offer any stronger protection options. The fact is that, for example, users like me who have cable modems, my IP basically never changes. I mean, if the power is out for a day, then yes, when I come back on, that IP that I was using will have been grabbed by somebody else, and so I'll get a new one. But I'll go years with the same IP, making it entirely feasible to simply have the incoming connection IP checked against a short list, like there were two, my IP here, and my IP in my other work location. If neither IP matched, the page is unavailable. It just comes up 404. Sorry, nothing here. But in either of my locations, I bring up the page normally.

I mean, so there's an instance of an Internet-exposed, fundamentally inherently vulnerable page which just a little bit of attention can completely lock down so that nobody anywhere else, with any other of the 4.3 billion IPv4 addresses or any IPv6s, for that matter, are able to get to it. It just makes sense.

Okay. So remember that it was recently necessary for the U.S. Department of Homeland Security to demand that all government agencies immediately update their Windows deployments to remove the Zerologon vulnerability, giving them a deadline. And also with predictable consequence, even after that, some still had not. So it just is the case that these things are not being fixed.

Anyway, in light of this list that the NSA published, I thought it would be interesting for us to take sort of a stroll through a single coherent summary of what the NSA is currently actively seeing exploited online. And without stepping on the lead, I'll note that some of these 25 are surprisingly obscure. They don't tend to make big waves or headlines in every case, but they are nevertheless powerful network vulnerabilities. And it's interesting that some of these more obscure vulnerabilities are due to errors in deserialization. I ran across that word a number of times. And I thought, okay. We've talked about this before, but not for a while. It's most often seen with Java.

The idea is that an object, like a Java object, has a schema, which is a structured layout of data. Typically when it's in use it's occupying RAM. The structure's definition often grows over time as all sorts of optional things that weren't considered in the beginning are hung onto it. So they tend to be less often used. But the scheme has to have room

for all of that. That means that any particular instance is often only sparsely filled in. So in order to store it efficiently, when it's time to save it, like onto mass storage, or maybe transmit it over a communications channel, the object goes through a process known as "serialization." And that's not as in assigning it a serial number, but as in converting it from parallel to serial, or sequential data.

But the trouble arises at the receiving or the deserializing end, when software wishes to restore the received or the stored data back into its parallel or usable form. The same programmers who wrote the serializer typically also write the deserializer; right? I mean, that's just like the way the project's going to go. They wrote the serializer, or they wrote a spec, then they wrote the serializer, then they wrote the deserializer, both to the same spec. So they naturally assume that the data their deserializer is deserializing was properly serialized by their own serializer.

The trouble arises when that's not the case. And what is the inherent nature of any deserializer? It's an interpreter. And one of this podcast's fundamental precepts is that interpreters are surprising difficult to make bulletproof. They're one of secure software's greatest challenges and banes.

So what are remote threat actors actively scanning for and exploiting wherever possible? Number one, Pulse Secure VPN servers. That's got a CVE number. I'm not going to go over all the CVE numbers because they're like star dates. CVE-2019-11510. Okay, fine. Pulse Secure VPN servers, an unauthenticated remote attacker can send a specially crafted URI - there's that term we talked about last week, Leo - to perform an arbitrary file read vulnerability. This may lead to the exposure of keywords or keys or passwords.

So in other words, on unpatched Pulse VPN servers, I mean, they're not actually vulnerable now except they're vulnerable on the 'Net because people didn't patch them. You can just ask for keys and passwords using a specially crafted URI on those vulnerable servers. Whoops. Again, we know how to fix it. People haven't.

On F5 BIG-IP proxies and load balancers, the Traffic Management User Interface, also referred to as a configuration utility, is vulnerable to a remote code execution that can allow remote attackers to take over the entire BIG-IP device. Again, we talked about this when it happened. Yes, it's no longer in the news, but bad guys have added that to their permanent bag of top 25 tricks.

We also have Citrix Application Delivery Controllers and Gateway systems which are vulnerable to a directory traversal bug which could lead to remote code execution without the attacker having to possess valid credentials for the device. And there are two issues that could be chained to take over Citrix systems.

Then there's a different set of Citrix Application Delivery Controller and Gateway bugs which also impact their SDWAN WAN-OP systems. In this case, three bugs allow unauthenticated access to certain URL endpoints and information disclosure to low-privileged users. So it doesn't seem like that bad a problem, but we've already seen how these things can be chained together in order to create a much bigger problem. And yes, we also have this one, CVE-2019-0708. Those are numbers I used a lot last year. That was the BlueKeep vulnerability. A remote code execution vulnerability exists within Remote Desktop Services on Windows OSes. And of course remember that Microsoft used the euphemism "authentication bypass." Yeah, I would say that is an authentication bypass. That's been a huge headache for admins. And anybody who hasn't patched is still vulnerable.

**Leo:** I don't think that was an NSA vulnerability. I was confusing that with EternalBlue.

**Steve:** Correct. Right, right, right. I think you're right, Leo. Number eight, a remote code execution vulnerability in the MobileIron Mobile Device Management, that's MDM software that allows remote attackers to execute arbitrary code and take over remote company servers. Whoops. Now, there's an example. Okay. MobileIron Mobile Device Management. That one slipped under our radar. We didn't talk about it. Didn't make big headlines. It's not splashy. But if somebody was using it, hadn't updated, again, arbitrary remote code and remote takeover of company servers who are using that. Again, these things need to be able to patch themselves.

We also have SIGRed. A remote execution vulnerability exists in the Windows domain system, the DNS servers, when they fail to properly handle requests. We talked about that earlier this year. There's Netlogon, an elevation of privilege vulnerability when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller using the Netlogon Remote Protocol. We've been talking about that recently, also.

There's a tampering vulnerability in Microsoft Windows when a man-in-the-middle attacker is able to successfully bypass the NTLM, that's of course the NT LAN Manager, MIC, the Message Integrity Check protection. Not good when that happens. We have sending a handcrafted message to an Exim mail transfer agent. We talked about that not too long ago. But that causes a buffer overflow, and that's been there, Exim's had this problem since 2018. And still, again, if it's not causing you problems, you don't think about it. But it's just sitting there allowing the remote code execution and the takeover of servers running that MTA, that Mail Transfer Agent.

There's a remote code execution vulnerability in Microsoft Exchange software when it fails to properly handle objects in memory. We've got some Adobe Cold Fusion versions having an exploitable, and here's the first one of many, deserialization of untrusted data vulnerability.

**Leo:** It's such an interesting vulnerability, the serialization/deserialization. I love that, actually.

**Steve:** Yeah, it's exactly right, Leo, it's very cool from a sort of a theoretical standpoint.

**Leo:** Right. It's kind of related to not sanitizing your inputs. You know, you just assume, well, this was serialized properly, so I'm just going to treat it as if it was, and boom.

**Steve:** Exactly. Exactly. And of course it also can allow arbitrary code execution. There's the WLS Security component in Oracle WebLogic Server. That's what WLS stands for, WebLogic Server 15, which allows remote attackers to execute arbitrary commands via a crafted serialized Java object. So although they didn't say deserialization, that's the same problem there. And a vulnerability exists in the Oracle Coherence product of Oracle Fusion Middleware. They said this easily exploitable vulnerability allows unauthenticated attackers with network access via T3 to compromise Oracle Coherence systems. So a little special case. But the bad guys are scanning for these problems; and they'll find them, if you haven't patched them.

We've got the Widget Connector macro in Atlassian Confluence 17 Server, which allows remote attackers to achieve path traversal and remote code execution on a Confluence Server or Data Center instance via server-side template injection. And that's a 2019 CVE,

so it's been around for a while, too. Another 2019er is attackers who can send requests to an Atlassian Crowd or Crowd Data Center instance can exploit this vulnerability to install arbitrary plugins, which permits remote code execution. Again, not Atlassian's fault, by any means. They fixed this a year ago. But unless people are staying current with updates, you're going to remain vulnerable.

So, I mean, it's like this last mile problem. We just have to fix this. Zoho Manage Engine Desktop Central allows remote code execution because of - wait for it - deserialization of untrusted data. Last year: Progress Telerik UI for ASP.NET AJAX contains a .NET deserialization vulnerability. Exploitation can result in remote code execution. And we've got CurveBall, which was a name we enjoyed talking about earlier this year. That's a spoofing vulnerability which exists in the way Windows CryptoAPI, that's the Crypt32.dll, validates Elliptic Curve Crypto - that's ECC - certificates. An attack could exploit the vulnerability by using a spoofed code signing cert to sign a malicious executable, making it appear that the file was from a trusted, legitimate source.

So that's not a huge problem. But again, if they detect that you have somehow not fixed this, it makes it very easy to slip something past AV, which is checking the signature on certs. If it's checking it against a defective CryptoAPI, it won't see that it was not actually from a trusted source. From last year, an elevation of privilege vulnerability exists in Windows when the Win32k component fails to properly handle objects in memory.

Now, remember around the middle of last year there was that whole spate of Win32k things? Well, that was thanks to SandboxEscaper, who was going through her tough time and was upset at the world and releasing zero-days right and left. Anyway, those have stopped because Microsoft in their infinite wisdom hired her. Good. Thank you. But apparently it's still out there floating around in the ether. There are still systems vulnerable among the top 25. Twenty-three is, as I mentioned earlier, from 2017, a problem in the Symantec Messaging Gateway. They fixed it three years ago, but it's still out there on the 'Net giving some people some pain.

There's a vulnerability in the Cisco Discovery Protocol implementation for Cisco IOS XR software, which could allow an unauthenticated adjacent attacker to execute arbitrary code or cause a reload of an affected device. So that sounds like something, since it's adjacent, you need to somehow be close to it in the network in order to take advantage of that, since it's using the Cisco Discovery Protocol. And finally, DrayTek Vigor devices allow remote code execution as root, without authentication, via shell metacharacters. And again, that didn't make anybody's headline anywhere. But apparently, whatever DrayTek Vigor devices are...

> **Leo:** Sounds like a vampire or something. I am DrayTek Vigor. I have come to suck your blood. I think it's "vigor." But anyway, it must be some sort of, yeah, some sort of device.

**Steve:** Yeah. So that's the list. Even a three-year-old vulnerability like that flaw in that Symantec messaging gateway remains on the hit list of well-financed state-level attackers. And you know, Leo, as I was looking through this, thinking about the NSA, here in the U.S. we have a naturally U.S.-centric viewpoint. So we don't see stories about the NSA or the CIA hackers doing the same thing to China, Russia, Iran, or North Korea. And I really wish that none of this was happening on either side because it just seems so destructive and wrong. But from a patriotic standpoint, I cannot help hoping that we're giving as good as we're getting because we certainly do appear to be getting a lot.

Once upon a time, you know, the ongoing level of network intrusions was mostly off the radar. It was embarrassing for any company to admit when they discovered that their

own network security perimeter had been breached. So whenever possible, it would be kept strictly on the down-low and dealt with as quietly as they could. Maybe they'd have to disclose it in an annual notice to their stockholders if they were publicly traded. But certainly, if they were not a public company, no, they would just like fix it and, shhh, let's not, you know, we'll just hope nobody notices.

But today, as we detailed last week with Ryuk, however you pronounce it, we have ransomware which proactively makes any such network intrusion embarrassingly public and impossible to hide. And look, it's happening everywhere. So the only sane appraisal would be that non-ransomware attacks are also happening everywhere. They're just continuing to go unreported, like once upon a time in the old days, before ransomware was outing all of the companies that were being attacked. We have to assume that I would say an equal measure as we're seeing publicly disclosed thanks to ransomware not giving people a choice are still happening.

So, I mean, unfortunately, our networks are sieves. I mean, just like, as I've said, I've talked about security being a sponge. The harder you press, the more gets through. It's a mess. And here's a list of 25 things, all which have been fixed, in some cases for three years.

**Leo:** Now, do they say these are the most prevalent attacks? Or just here's an assortment of 25 attacks?

**Steve:** Yeah, it was...

**Leo:** These are the most common?

**Steve:** Yes. The most common that they are seeing.

**Leo:** Because I get attacked all the time.

**Steve:** And the most easily remediated. I mean, just update.

**Leo:** Right. Yeah, I get attacked frequently because now with my Ubiquiti I can see the alerts to all the various blocked attacks. And it's all over the place, all kinds of ports. They're probing all the time.

**Steve:** Well, and remember I coined the term years ago, IBR, Internet Background Radiation.

**Leo:** It's only gotten worse.

**Steve:** Yeah, I mean, there's still Code Red out there scanning for Windows NT, hoping to get lucky.

**Leo:** I think a lot of people, script kiddies especially, are just using scanners, or Shodan-type searches, just looking, you know, I know there's a vulnerability. I'll just scan millions of IP addresses a second and find which one has it. Because I get scanned hundreds of times a day, easily, for stuff.

Hey, I thought I'd share this with you before we wrap it up. This is from Harry McCracken's Twitter feed. And it made me think of you. This was 20 years ago during the 2000 election, the mobile edition of Gore 2000 for your Palm Pilot. Remember the program AvantGo? If you use AvantGo's software, you can download - and they suggest you sync at least twice a week to get the latest information on the election on your Palm Pilot. I thought about you when I saw that. We've come a long way.

**Steve:** I have one of those.

**Leo:** You have several, as I remember.

**Steve:** Uh-huh.

**Leo:** That's Steve Gibson, everybody. He is the man of the hour, the day, the week, the year, the last 790 episodes of Security Now!. If you want to keep up on what's going on in security, this is the place to be every Tuesday, right about 1:30 Pacific, 4:30 Eastern. Normally it has been 20:30 during our summertime, but we are going to, this coming Sunday, shift back to standard time, which means our time will no longer, what did I say, 21:00 UTC. It'll be 20:00 UTC. So we're falling back. So just a little word of warning. You may have already changed, a lot of Europe changed to standard time this past weekend. But ours comes on the day after Halloween this year. So just make a note of that, if you like to watch live.

The live streams are at TWiT.tv/live. Audio and video is there. Chat in the chatroom if you're watching live. They're live, too, most of them, irc.twit.tv. You can also get on-demand versions of the show from Steve's site, GRC.com. He has 16Kb audio for the bandwidth-impaired, 64Kb audio. He's also the only place you can get hand-coded, human-written transcripts of the entire show, so if you like to read along. It also makes it easy to search because you can search those transcripts and find the part of the show you're most interested in. That's GRC.com.

While you're there, pick up a copy of SpinRite, the world's best hard drive recovery and maintenance utility, version 6 current, 6.1 on its way. You can participate in that and get a free upgrade if you buy right now at GRC.com. There's lots of free stuff there, too, including ShieldsUP! and his Perfect Paper Passwords, all the information about SQRL. His new forums, they're up and running. Is that forums.grc.com?

**Steve:** Yes, yes.

**Leo:** Okay, good. You can leave a question there for him, GRC.com/feedback. But he's also on Twitter. I always put up your Twitter handle because he accepts DMs from anybody: @SGgrc. If you've got a tip or you just want to converse with the great Steve Gibson: @SGgrc. We have all the shows at our site, TWiT.tv/sn. There's a YouTube channel. Honestly, the best thing you could do, pick up a copy of Pocket Casts or Overcast or Shoutcast or whatever cast, all the podcast applications, and

subscribe. That way you'll get it automatically, the minute it's available of a Tuesday evening. I think we're done here, Steve. Thank you so much.

**Steve:** And Leo, we will be back together on the U.S. Presidential Election Day.

**Leo:** Now, did you early vote? Or are you going to be voting on Tuesday?

**Steve:** Oh, months ago.

**Leo:** Yeah. Me, too. I voted the minute I got...

**Steve:** Vote early. Vote often.

**Leo:** So we won't take the day off on Tuesday. But folks, if you're listening, and you want to take the day off so you can vote, Steve and I give you permission. In fact, we encourage you. We encourage you to do that because we're going to be here no matter what. Thanks, Steve. We'll have a great time next Tuesday. We'll see you then.

**Steve:** Thanks, buddy. Bye.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/