IoT Isolation Strategies

Description: This week we look at another device to receive DoH privacy, a browser to block drive-by downloads, my favorite messaging solution going open source, a new and trivial attack against hundreds of thousands of WordPress sites, Facebook's new vulnerability disclosure policy and their publication of WhatsApp security advisories, forthcoming security researcher policies for U.S. government properties, a new Tor Project membership program, Intel's latest microcode patches, the result of a small but significant double-blind controlled trial related to COVID outcomes, a SpinRite update, and a discussion of the need and means of enforcing strict IoT network isolation.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-783.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-783-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. For a long time he's talked about the three-router solution to isolating IoT devices. Well, for the first time ever, Steve has finally got some IoT devices. He's been spending some time thinking about it, and he's got several really useful and fairly easy techniques for protecting yourself against those little servers all over your house. Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 783, recorded Tuesday, September 8th, 2020: IoT Isolation Strategies.

It's time for Security Now!, the show where we cover your security and privacy and all that jazz online from the red sun headquarters up here in blighted Petaluma. I'm saying hello to Steve Gibson. It's a little clearer down there, I know.

Steve Gibson: Little clearer. It's been hot and, boy, was it humid. Or I guess it is today, actually. I've not been outside.

Leo: Muggy? Yeah.

Steve: Yeah.

Leo: We have a lot of smoke still from fires. I know you have fires down where you are, too.

Steve: Yeah.

Leo: Hope you stay safe.

Steve: California's been crazy.

Leo: It's been bad, yeah.

Steve: And actually it's the heat and the humidity that is actually responsible, oddly, for the topic this week.

Leo: Oh, really.

Steve: Because, yes, I have bit the bullet finally and stepped into the IoT world.

Leo: What?

Steve: Because I wanted some of the functionality that IoT provides. And of course in order to do that safely you can't just stick these things on your network. And so I thought it would be useful to revisit IoT isolation strategies, which is our topic for the week, because I just implemented two different ones for my two locations.

Leo: Oh, good, yeah. Because people ask me about your three-router solution all the time. That has become legend.

Steve: Right. Well, it ends up being a good solution, even only if you use two routers, which we're going to talk about. But we had a bunch of stuff going on. We're going to take a look at another device to receive DoH privacy protections, a browser to block drive-by downloads, or I should say another browser which will begin blocking drive-by downloads. And we're going to sort of remind ourselves what those are and what that's about. And of course I heard you already talking about my favorite messaging solution. I think they were listening to you, Leo, or somebody's ears were burning. I'm not sure how that works. Because my favorite messaging solution has announced that it will be going open source.

Leo: Yeah, this is huge, huge. Yeah, yeah.

Steve: And of course you know what that is. Yeah, huge. We've got a new and distressingly trivial attack against hundreds of thousands of WordPress sites.

Leo: Oh, boy.

Steve: And we just seem to be talking about this now all the time. Once upon a time it was Flash exploits. Now it's WordPress. Also Facebook has a formal vulnerability disclosure policy. And also they're going to be publishing WhatsApp security advisories. And there's some interesting back story for both of those. We've got forthcoming security researcher policies for U.S. government properties which is really welcome because everyone's a little twitchy about, well, if I find a vulnerability on a government site, I don't want to get sued for poking at it or talking about it or even informing them.

We also have a new Tor Project membership program. Intel's latest microcode patches and a couple takes on that. Also in one piece of miscellany the result of a small, but surprisingly significant, that is, a P of less than 0.001, double-blind controlled trial which was related to COVID outcomes. A quick SpinRite update, and then we're going to dig into the need to isolate your IoT devices and my thoughts about how to do that. So I think a great podcast for our listeners.

Leo: Some meaty material to sink our teeth into. And I'm looking forward to hearing about IoT solutions because I know that's a big one. Steve?

Steve: So our Picture of the Week is just something, it's kind of something I've always been bemused by and just sort of sad about, which is what has happened to our URLs. I love this because this was something someone tweeted to me, email that he received at his Gmail account. And it reads: "We have reset your EA account password. We know this is an inconvenience and hope you recognize that we take the safety of your personal information seriously. What happened? After detecting potentially suspicious activity associated with your EA account, we reset the password to protect your personal information. At this time we do not think the suspicious activity is the result of unauthorized access to EA databases. We think the activity may be related to issues like phishing, weak user passwords, logging in from shared components," blah blah blah.

Anyway, "What do I need to do?" it asks the user. And it says: "Use this URL to change your password." Now, in this email the URL, I mean, it didn't even begin to fit on one line. It's one, two, three, four, five, six, seven, it's eight lines long of just random wrapping gibberish. And, okay, so yeah, you could click on it, and clearly that uniquely identifies the recipient of the email to signin.ea.com.

But what I loved was down at the bottom, and this is - we've talked about this before - good security advice. It says: "Tip: We suggest manually typing the URL into a browser to ensure that you open a secure page. Secure page URLs use the https prefix." And of course good luck typing this URL into your browser. You'd way be better off just starting over with EA. Anyway, I just got a kick out of that. It's like this maybe once upon a time they were using URLs that you had some snowflake's chance in hell of manually entering into your browser. But those days are long gone. This thing is a nightmare. So just click it and hope for the best because you're not going to be able to type it in.

DoH is coming to Chrome for Android. Although the desktop releases of Chrome have offered the enhanced security and privacy that many of us love, offered by DNS-over-HTTPS, which appeared in Chrome 83, two major versions ago. It wasn't added to the iOS or Android editions back then. But last Wednesday's Chromium blog was titled "A Safer and More Private Browsing Experience on Android with Secure DNS." And I think I would argue that that's a good place to have DNS-over-HTTPS.

Paraphrasing from Google's posting for the sake of brevity, and to skip a bunch of stuff that we all already know, they said: With Chrome 85, we're extending support of Secure DNS in Chrome to Android. As we did for the launch of DoH on Chrome for desktop platforms, we'll progressively roll out DoH on Chrome for Android to ensure the feature's

stability and performance, as well as - and I think this was significant, too - help DoH providers scale their services accordingly.

You know, we've talked about the fact that arguably setting up persistent HTTPS connections to DNS servers is a significantly more burdensome process for the server than just fielding UDP packets. DNS, traditionally over UDP, is in comes a packet and out goes a reply. HTTPS involves the whole TLS handshake to set it up. And then you're maintaining a persistent session. You don't want to have to do that every time you make a DNS query or it'll be way slower, and nobody wants a performance hit when you get authentication and privacy. But still, you have to maintain state over at the server end in order to offer this service. And that can require the providers to scale in order to make that happen. So presumably they're going to be keeping an eye on how well this does. And in fact they talk about that a little bit, about the possible need to fall back.

They said: "Android Chrome will automatically switch to DNS-over-HTTPS if your current DNS provider is known to support it." And I just hit the spacebar by mistake and scrolled. "This also applies to your current Android Private DNS [DNS-over-TLS] if you have configured one." They said: "This approach means that we can preserve any extra services provided by your configured DNS service provider, such as family-safe filtering, and therefore avoid breaking the user's expectations," you know, making it transparent.

They said: "In this automatic mode, Chrome will also fall back to the regular DNS service of the user's current provider, for example, if you had configured DNS-over-TLS, in order to avoid any disruption, while periodically retrying to secure DNS communication." So presumably, as I said, if the provider were to become overloaded or stop offering the services for a while, it'll just smoothly drop back to traditional DNS.

They said: "In case this default behavior isn't suitable to your needs, Chrome also provides manual configuration options, allowing you to use a specific provider without fallback, as well as the ability to completely disable the feature." And they said: "If you're an IT administrator, Chrome will disable Secure DNS if it detects a managed environment via the presence of one or more enterprise policies. We've also added new DNS-over-HTTPS enterprise policies to allow for a managed configuration of Secure DNS and encourage IT administrators to look into deploying DNS-over-HTTPS for their users."

So anyway, basically it's what we've seen before, offered by other, well, by Chrome and pretty much now all of the browsers who are wanting to move their clients and users from DNS over UDP, traditional DNS, over to HTTPS. So it's just going to happen seamlessly. Chrome will start supporting it. If they see that you're using - and it comes with like a list of known providers. If they see that you've already switched your DNS there, then they will just upgrade you to that provider, if that provider is known to offer the service. And because it'll be an automatic upgrade, if there's a problem they'll back you out of it. And then in addition, once that's provided under Chrome for Android, you can also switch it to sort of a non-automatic mode, provide the IP address and other URL requirements for DNS-over-HTTPS and start using it.

So again, to me I think that makes a lot of sense, especially when a mobile device is out roaming around a lot more. You're away from home where your DNS is only going, essentially never even going public at all. It's just to your own local provider in order to get resolution. So good to see this happening. And of course we've talked about how this is also being rolled out for Windows that'll be supporting DoH natively for the whole OS.

We're currently today at Firefox 80, eight zero, .0.1. And that was since last Tuesday, which quickly followed the move to Firefox 80 when it fixed a couple of regression and other problems. 80 had a problem which was fixed in 80.0.1. They fixed a performance regression when encountering new intermediate certificate authority certificates. They fixed crashes possibly related to GPU resets. They fixed rendering on some sites which

were using WebGL. I think we talked about that last week, actually. There was actually a security problem with that. Fixed the zoom-in keyboard shortcut on Japanese language builds. So, okay, a little obscure there, but that's fixed. And also download issues related to extensions and cookies.

So we're now at 80.0.1. It's 82 coming next month that I want to talk about because two editions from now, point releases notwithstanding, Mozilla will finally be catching up with a useful security feature which Chrome has offered since March of last year, 2019. And that's the feature of blocking so-called "drive-by downloads" from sandboxed iframes. And really, for any browser these days I would be asking what took them so long.

It does break something that has otherwise been working, so I guess I can understand. Obviously, there's always a huge reluctance on the part of any browser to change behavior in a way that's more restrictive. You don't want a browser update to suddenly cause things that used to work to stop working. And so it's like, hey, before I updated my browser, this was working. Now it's not. Of course the browser gets blamed for that, rather than perhaps a site doing something which has now become against best practices.

So we talked about this recently, that is, the idea of so-called "drive-by downloads." It's possible for JavaScript running on a page to all by itself, autonomously, initiate a download of a file by that browser. And you often see that in some of those a little more sketchy download repository sites where you go, you click on a download because you want something that you haven't been able to find anywhere else. And it says, "Your download should start in 12 seconds," or some length of time. And then it sits there counting down. And it's like, okay, what am I waiting for? Why didn't it just start now? Well, it's because they've got advertisements all over the page. And they're hoping that, if they just hold you there, if they just stall the download that you want, maybe you'll see something that interests you, or who knows. Maybe the advertisement knows how long it's got your eyeballs, and they're actually getting paid more if the ad stays visible longer. That's certainly possible, and that's been done in the past.

In any event, the point is this is script which is deliberately not triggering the download in your browser until it decides to. Then it does. So that's all being scripted. And of course you'll also notice that sometimes it'll say, "If your download does not start in [some length of time], click here," which is the non-scripted, normal, click to start the browser URL in order to manually trigger the download. So the point is script is able to do this. And when I talked about it before, I mentioned that, oh, the place I see it is on SourceForge. Often there's something that I'm wanting to get from SourceForge, and for whatever reason they do the same thing. You click on the download link which actually takes you to a different page where it says, oh, your download will start in X amount of time. And it's like, okay, fine.

When that finally happens, what I was noticing was that, in Chrome, I would wait to see something happen, not noticing that it had opened a little download tab thing, a little box down in the lower left of the browser window. So unless you're looking for it, it just sort of appears down there. That's if you're still using Chrome's default, which is not to ask you for the download location every time. And this is what we were talking about before is, because I realized I could change this, I did. I went in, I changed the default to prompt for where it wants to go, and now I get a big pop-up browsable dialog that allows me to specify where I want the download to go. So it's no longer transparent and invisible.

And of course bad guys can use this. And this is the point of the whole drive-by downloads. Bad guys can use this behavior on web pages to quietly cause your browser to download something that you may not notice. And so some time later you look down, and you go, oh, there's something that was downloaded. And it's also, of course, on your

computer. So perhaps at some later date you look in your downloads folder, and you see something that you don't remember getting, don't know what it is, and so you click on it because it's there, and you think, well, I must have downloaded it on purpose by mistake previously, but no. Malware did this.

So the good news is that's going to be changing. We've established that a website's JavaScript is able to trigger a download which might go unnoticed by a user, who might later then make the mistake of clicking it. It could be malware, and then they're in trouble. So that functionality for so-called "sandboxed iframes" started being disabled by Chrome a year ago last March, March of 2019. It could be reenabled with a flag, if that caused a problem. They were kind of hedging their bets. But then that got removed, I think it was in May of this month that they finally took that away.

So this happens with so-called "sandboxed iframes." A sandboxed iframe is one that includes the tag "sandboxed." And what sandboxing in iframe does is immediately restrict what any content in the iframe is able to do. It's a good thing to do. And we talked about this some years ago, the idea of just adding a "sandbox" tag to all of the iframes that you have any control over. Or at the time, ask your webmaster to do that. The point being that very often iframes are typically providing content from offsite servers, in other words, often advertising servers, and you don't want the content that gets loaded by the iframe to have the total run of your web page. You don't want it to be able to do anything that the hosting page is able to do. It needs to be restricted. Sandboxing does that.

One of the things that was still allowed in a sandboxed iframe was, if scripting was allowed - which almost always has to be allowed for advertising scripting. So for example, if you're a site that wants advertising, you set aside some space in an iframe where you then reference an advertising server. Part of their terms are going to be, well, we need to run script in our iframe in order to do ad rotation and to monitor ad hits and clicks and eyeballs and so forth. Until recently, that script was able to do a drive-by download. Chrome backed it off. Now, in October, next month, Firefox will also do so. If for some reason you need an iframe to be able to do so-called "drive-by downloads," or to allow scripting to trigger a download for some reason, you are able to override that in the iframe by adding an "allow download" tag to the "sandbox" tag. So you're able to still back that behavior out. But by default that will be shut down.

And so once again, as ever, we are inching forward bit by bit, gradually undoing what was now seen as or is seen now as too permissive security mistakes that we collectively made as an industry back when we were all still amazed that all of this stuff worked at all. Now we know it works, and we're wishing that it was a lot more secure than the way it was originally created. Of course it's always difficult to take away functionality because that risks breaking things. So we're just doing it bit by bit, carefully.

So the news that I'm excited about - and Leo, I know you are excited about it - it was by far the most tweeted bit of news I was receiving last week is the news that Threema has decided to go open source, and also to welcome a new partner. They announced it, I think it was last Thursday. They announced in a blog posting that they were going to be teaming up with an investment company, a German-Swiss investor, I guess it's Afinum, A-F-I-N-U-M, Afinum Management AG. And it looks to me, since the original Threema founders are staying in place and are retaining, they made a point of saying, a significant ownership interest, what I think must have happened is they had managed to build Threema up to the point where it was valuable enough that they could peel off some equity in order to raise money for themselves, in order to do things that they want to do moving forward. And one of the things they had wanted to do, which they now feel they can, is to take Threema open source.

And of course we talked about Threema just recently because they had just added, it was in the context of them adding video calling. And in order for it to be done as well as they wanted to do it, that is to say, as truly privately as they wanted to do it, they had to modify the WebRTC standard, which was not protecting some of the metadata, signaling information that was used to set up the WebRTC call. They modified the WebRTC standard. They're going to be working to have that standard essentially backported into WebRTC, and then they were able to add it to Threema. So all of our listeners know that I love Threema because their system is the easiest, well, I'm sorry, their system is the most secure because you could argue that it's the least easy to use. On the other hand, it's not that difficult. You were just demonstrating it on iOS Today, I think.

Leo: Yeah. And I've shown my QR code and everything.

Steve: Yeah.

Leo: It does, it is kind of - prefers in-person meetings. The reason I like Threema better is, unlike Signal, it doesn't require you to give a phone number or any personal identifying information. So it really is private. Which Signal isn't, unfortunately.

Steve: Correct, yeah. Well, and I say it's the least easy to use because that doesn't mean it's difficult. But it just means that its key management is not 100% transparent and automatic. And so my point is, as I made it before, if you actually do care about security, then managing your most important secret...

Leo: You want that to be hard.

Steve: ...which is your keys - yeah, exactly.

Leo: You want that to be sophisticated. But you saw how easy, if you watched iOS today, how easily Mikah shared his QR code with me, and I shared mine with his, and now we have three dots, each of us, because we verified in person, in effect. And people were saying, well, why are you showing your QR code on the air, Leo? Well, if you presume that you're seeing me, and this is really me, with that QR code...

Steve: I'm sure it's you, Leo.

Leo: You don't need to do it in person. It's just that in a way that you're comfortable that that is me, and it hasn't been hijacked, the QR code hasn't been somehow replaced in transmission to you.

Steve: Correct.

Leo: And I think that's a fairly safe bet.

Steve: And we know what it is that you're showing is not a secret. It is your public key.

Leo: That's my public key; right.

Steve: So, yes. So by definition it can be public. It's just like a PGP public key.

Leo: Right.

Steve: And in this case it's a way of visually exchanging it between people. You're exchanging your public key. So what that means is several things. It means that anything they create and encrypt with your public key can only be decrypted by your private key. So they know that nobody but you can read it. And similarly, they have also encrypted it with their private key. And so when you receive something from them, which you decrypt first with your private key and then with their public key, you also know it absolutely came from them. So it prevents spoofability.

So, I mean, it is the way to do truly secure TNO, Trust No One, messaging. And it's a point I've always been making about iMessage is, yeah, I'm sure it's encrypted. I mean, I'm sure it's end-to-end encrypted. But from a purely anal retentive security standpoint, if I'm not myself managing keys, then somebody else is. I've delegated key management. Which is, I'm not saying it's bad. It just means you trust Apple. And I know we do trust Apple. But you've delegated your key management to a third party, which now it's not TNO, it's Trust Only Apple, TOA. And that's probably fine.

Leo: And I always averred that if it's not open source, it's not TNO because you don't know what's going on in the binary blob. But once it's open source - and actually I think Threema had this interim thing we talked about last week where they had a provable that you would encrypt a file and run it against open source and prove that they were using that bit of it was open source. But this is [crosstalk].

Steve: And it was being periodically audited by outside entities, so they had that. What they said is, they said: "Security and privacy protection are deeply ingrained in Threema's DNA, which is why our code gets reviewed externally on a regular basis. Within the next months, the Threema apps will become fully open source, supporting reproducible builds."

Leo: That's key also.

Steve: "This is to say" - yup - "that anyone will be able to independently review Threema's security and verify that the published source code corresponds to the downloaded app." And it's going to be on GitHub.

Leo: I think that's [crosstalk].

Steve: And they also said: "In the future it will be possible to use multiple devices in parallel, thanks to an innovative multi-device solution."

Leo: That's one problem is you can't use it on your phone and your iPad. You can use one or the other.

Steve: Exactly. They said: "In contrast to other approaches, no trace of personal data will be left behind on a server. Thanks to this technology, Threema can be used on a PC without a smartphone."

Leo: That's awesome. So right now you run the website, show it the special QR code from your Threema app on your phone, and you can do it on the desktop. But it's a temporary. You have to keep pairing it and things like that. And I have to say that I've made Threema now my default instead of Signal for that simple reason. I think this is - now that they're open source, I agree with you. I mean, I trust Moxie Marlinspike, et cetera, et cetera. And by the way, the other thing that Threema allows you to do is share your contact list so that you can see if anybody else you know is on Threema. I would not do that, and it is not required.

Steve: Right, right.

Leo: So please don't share my information with anybody else. Let them find me [crosstalk].

Steve: Well, Leo, and in this day and age of Zoom conference calling and the ability to, for example, use iMessage to exchange the Threema QR code, you can put it on a screen, snap a picture of it, send it to somebody. So, I mean, you can create this secure sharing easy enough in order to know that you've swapped public keys.

Leo: Yeah. I put my ID on my website. Mostly because if somebody wanted to communicate with me privately, I've really come to the opinion that, even though I use PGP, and I love it, it's not inherently secure. It's too hard for most people to set up. Secure messaging is certainly the preferred way, if you want to have a secure conversation.

Steve: Well, it's like - I just love the original comment. What's the best camera? The answer is, well, the one in your pocket.

Leo: Yeah, right, right.

Steve: Or the one you have with you when something happens that you want to take a picture of.

Leo: Right.

Steve: Similarly, what's the best encryption? It's like, well, there may be really good encryption which is really difficult to use. So how about really good encryption that's simple to use?

Leo: That you've got, that works, yeah.

Steve: Exactly. So WordPress once again. It's not good when a zero-day flaw is discovered being actively exploited in an extremely popular plugin for WordPress. And it's also somewhat jarring that we keep covering exactly this news. In the latest of a continuing series of such WordPress vulnerabilities, the so-called "WordPress File Manager" plugin is currently being, and I say "currently" because there are still hundreds of thousands of vulnerable WordPress installations, actively exploited to permit full website hijacking. That'll ruin the day of anybody who has a big investment in their WordPress site.

The Sucuri WordPress security team said that the vulnerability was introduced into the May 5th v6.4 of WordPress File Manager. However, other reporting suggests it's been there since 6.0, so substantially longer than May 5th. It may have only been seen for the first time there. And so File Manager, this WordPress File Manager, is used as an alternative to FTP for managing file transfers, copying, deleting, and uploading files. And when I said "popular," I wasn't kidding. It's used by more than 700,000 active WordPress installations.

The mistake that was made, everybody's going to be able to understand. It was minor. Well, okay, dumb, but had major consequences. One of the plugin's files was renamed by the development team while they were developing it for testing purposes from its safe inactive form to its dangerous active form. And the project with that renamed file was then distributed. The file was "connector-minimal.php-dist," but it was mistakenly left renamed to "connector-minimal.php."

So as anyone who's done any web work knows, a file ending in .php-dist would not invoke the PHP interpreter to parse and process its PHP script. If the web server happened to have a MIME type associated with that file extension, that is, php-dist, you know, it might download it to you, if you were to query for it. But it would not execute it as a script. But any remotely accessible file ending in .php would be executed because the web server would invoke the system's registered PHP interpreter. So the only thing that any attacker needed was to invoke that script remotely and have at it. Which is exactly what then started to happen. What that errant file permitted was bad, or as the Sucuri team said: "Leaving such a script intentionally designed to not check access permissions in a public build causes [what they described as] a 'catastrophic vulnerability' if this file is left as-is in the deployment." And that's what happened.

They said: "This change allowed any unauthenticated user to directly access this file and execute arbitrary commands to the library, including uploading and modifying files, ultimately leaving the website vulnerable to a complete takeover." And that's 700,000. Wait, no, I'm sorry, it was 53% of the 700,000 WordPress sites using this File Manager were vulnerable. So more than 350,000 WordPress sites at the time of its discovery. The solution, which appeared in the replacement v6.9 distribution, was simple. They simply deleted the file and any other unused .php-dist files.

However, a week before the file was removed, a simple proof of concept was posted on GitHub, which led to a rapidly mounting wave of attacks against websites until those sites were updated to v6.9. Sucuri says the exploit rapidly gained traction. The first attack was spotted last Monday, August 31st, the day before a fixed version of the File Manager was released. This ramped up to 1,500 attacks per hour, and a day later this had increased to an average of 2,500 attacks per hour across the global Internet of WordPress sites. And by the next day, September 2nd, that is, when we were doing this podcast last week, Sucuri was clocking around 10,000 attacks per hour. Sucuri said that they had tracked hundreds of thousands of requests from malicious actors attempting to exploit this trivial-to-exploit vulnerability.

Later analysis showed that the flaw was in, as I mentioned before, File Manager from v6.0 to 6.8. Statistics from WordPress show that currently 52% of installations are vulnerable, so a little over half of the 700,000 known to be vulnerable. And as of last Thursday, the next day, September 3rd, only 6.8% of those 52% of the total 700,000 originally vulnerable WordPress websites had updated. So again, the fact that a patch is made available doesn't mean that everybody instantly has it. So this has just been a catastrophe.

And as I mentioned before, the last thing you want is to have a PHP-based system accessible to the rest of your Internet. We're going to be talking at the end of the podcast about sequestering an IoT network from the rest of your network. I've talked about before, and I'll just say again in this context, you should also sequester a server where you've got PHP social media stuff, whether it's an online forum or it's a WordPress site where you added plugins and such, which are hard to avoid adding because they do create additional value. Similarly, that server should be sequestered from the rest of your corporate or personal or local network because this kind of thing can happen just too easily. Maybe only the website would be defaced. But if you're a known enterprise with a WordPress-based presence, bad guys might not stop at just defacing WordPress. They could use this as the beachhead through which to launch an attack to move laterally in your network. So the idea of creating lots of compartmentalized websites really does make sense.

So Facebook published a new, for them, Vulnerability Disclosure Policy. The many Facebook platforms run on a bunch of code, and much of the code is theirs. But as is increasingly the case as the industry matures, code pulled from many third-party libraries is also often used. So the question becomes what should Facebook do when it finds a problem in some third party's code? This is sort of like a Facebook commercial version of Google's Project Zero, where although in their case they're not deliberately going out and trying to find problems, they're needing to make sure that their amalgamated result functions, and they're needing to use third-party libraries.

So they formalized and published their policy, which is I think well thought out and reasonable. It's similar to Google's Project Zero, with some inevitable differences. In explaining this, they explained what they were doing. They said: "Facebook may occasionally find critical security bugs and vulnerability in third-party code and systems, including open source software. When that happens, our priority is to see these issues promptly fixed, while making sure that people impacted are informed so that they can protect themselves by deploying a patch or updating their systems. That sounds simple and clear cut. However, vulnerability disclosure is anything but simple. Here is what motivated our policy."

They said: "First, not all bugs are equally sensitive. A high-impact security issue requires much more care before it is publicly disclosed. The policy outlined below explains how we handle vulnerability disclosure. And, two, fixing an issue requires close collaboration between researchers at Facebook reporting the issue and the third party responsible for fixing it. With this policy, we want to clearly and unambiguously explain our expectations when we report issues we find in third-party code and systems. We also make sure when Facebook will disclose these issues."

So for their policy they said: "In a nutshell, Facebook will contact the appropriate responsible party and inform them as quickly as reasonably possible with a security vulnerability we've found. We expect the third party to respond within 21 days to let us know how the issue is being mitigated to protect the impacted people. If we don't hear back within 21 days after reporting, Facebook reserves the right to disclose the vulnerability. If within 90 days after reporting there is no fix or update indicating the issue is being addressed in a reasonable manner, Facebook will disclose the vulnerability." Okay. So three weeks to hear something saying okay, we've established

communications, we're on it; and three months for, very much like Google's Project Zero, a drop dead for this is going to be, you know, Facebook's going to disclose.

So they said: "That said, we will adhere to the vulnerability disclosure steps and the proposed timelines whenever reasonably possible, but we can envision scenarios where there might be deviations. If Facebook determines that disclosing a security vulnerability in third-party code or systems sooner serves to benefit the public or the potentially impacted people, we reserve the right to do so." And they said: "Here are some details."

For reporting they said: "Facebook will make a reasonable effort to find the right contact for reporting a vulnerability, such as an open source project maintainer. We will take reasonable steps to find the right way to get in touch with them securely. For example, we'll use contact methods including, but not limited to, emailing security reporting emails like security@ or secure@, filing bugs without confidential details in bug trackers, or filing support tickets. The contact should acknowledge the report as soon as reasonably possible. The contact should confirm whether we've provided sufficient information to understand the reported problem.

"In its report, Facebook will include a description of the issue found, a statement of Facebook's vulnerability disclosure policy, and the expected next steps. If needed, Facebook will provide additional information to the contact to aid in reproducing the issue. If we do not receive a response within 21 days from a contact acknowledging the report of a vulnerability, we will assume that no action will be taken. We then reserve the right to disclose the issue. For purposes of the disclosure timeframe, Facebook's sending the report constitutes the start of the process. Facebook will generally decline to sign non-disclosure agreements specific to an individual security issue that we have reported."

Under mitigation and timeline they said: "Whenever appropriate, Facebook will work with the responsible contact to establish the nature of the issue and potential fixes. We will share relevant technical details to help expedite the fix. The contact should be as transparent as possible about the mitigation progress. They are expected to make reasonable effort to fix the reported issue within 90 days. Facebook will coordinate the disclosure with the availability or rollout of the fix. If no fix is forthcoming at the 90-day mark, we will notify the contact of our intent to disclose the reported issue. If there are no mitigating circumstances, we will disclose the issue as soon as we are reasonably able to do so."

And finally, under disclosure: "Depending on the nature of the problem, there may be a number of disclosure paths. One, we may disclose the vulnerability publicly; two, we may disclose it directly to the people using the project; or, three, we may issue a limited disclosure first, followed by a full public disclosure. Facebook will work with the contact to determine which approach is most appropriate in each case." They said: "Our intent is to disclose vulnerabilities in a way that is most helpful to the community. For example, we may include guidance on workarounds, methods for validating patches are in place, and other material that helps people contain or remediate the issue. We may choose to include a timeline to document communication and remediation actions taken by both Facebook and the third party. Where reasonable, our disclosure will include suggested steps for mitigating actions. We will include a CVE when available and, if necessary, issue an appropriate CVE."

And then they said, under additional disclosure considerations, they said: "Here are some potential scenarios when Facebook may deviate from our 90-day requirement: if the bug is actively being exploited, and disclosing would help people protect themselves more than not disclosing the issue. If a fix is ready and has been validated, but the project owner unnecessarily delays rolling out the fix," they said, "we might initiate the disclosure prior to the 90-day deadline when the delay might adversely impact the public.

And finally, if a project's release cycle dictates a longer window, we might agree to delay disclosure beyond the initial 90-day window where reasonable."

They said: "Facebook will evaluate each issue on a case-by-case basis based on our interpretation of the risk to people. We will strive to be as consistent as possible in our application of this policy. Nothing in this policy is intended to supersede other agreements that may be in place between Facebook and the third party, such as our Facebook Platform policies or contractual obligations." And they said: "Finally, this policy refers to what Facebook does when we find an issue in third-party code. If you believe you have found a security vulnerability on Facebook or other member of the Facebook family of apps, we encourage you to report it to our Bug Bounty Program."

So anyway, they've done a formal disclosure. I think all of that, this is sort of what the industry is coming to accept as the proper way this is being done. Anybody who has got code out in public of this sort, who's got libraries which are in use by others, should make sure that they have some means of clearly being contacted if somebody finds a problem so that they're not surprised by the news that a problem in their code has just been made public because it wasn't easy for others who were finding problems to notify them privately, allowing them time to make the fix public and coordinate the disclosure. So I think that's just all for the best. It's the way we're seeing it being done these days.

And they also produced a WhatsApp Security Advisories page. It's whatsapp.com/security/advisories. They wanted to do this, and they explained what was behind this. They said: "Due to the policies and practices of app stores, we [Facebook or WhatsApp] cannot always list security advisories within app release notes. This advisory page provides a comprehensive list of WhatsApp security updates and associated CVEs," you know, the common vulnerabilities and exposures. "Please note that the details included in CVE descriptions are meant to help researchers understand technical scenarios and does not imply users were impacted in this manner."

So essentially what they're saying is app stores, where new versions of WhatsApp would be made available, may not have the freedom, as a consequence of the terms and conditions of the app store, to say as much as they want to. So they wanted to explicitly create a page where they'd be able to do that. And the current page is gratifyingly short at the moment. There were six CVEs disclosed for 2020 to date, and one of those was in 2019. So unlike Microsoft's, where we're getting more than 120 problems every single month, here we've got a gratifyingly short policy, or list, rather, of CVEs for all of 2020.

And along the lines of vulnerability disclosures, it turns out that some work that was initiated last year has come to fruition for U.S. agencies, the result of which is that U.S. agencies must adopt and publish vulnerability disclosure policies for researchers finding problems on U.S. government sites by March of next year. Bryan Ware, who is the Assistant Director of CISA, the Cybersecurity Infrastructure and Security Agency with the DHS in the U.S., blogged about the formalization of the U.S. government's policy. In his posting, he explained that last November CISA asked for feedback on a draft of the binding operational directive which would be requiring most executive branch agencies to create this formal vulnerability disclosure policy, or VDP, which would inform those who discover flaws in a U.S. agency's digital infrastructure, where to send a report, what types of testing are authorized for which systems, and what sort of communication to expect in response.

He said: "We'd never done a public comment round on a directive before; but since the subject matter was coordination with the public," he said, "this one merited it. And even though the comment round spanned every holiday," he said, "from late November to early January" - so I guess, what, Thanksgiving through Christmas and New Year's - he said, "the quantity and quality of feedback was stellar." He said: "We received over 200 recommendations from more than 40 unique sources: individual security researchers,

academics, federal agencies, technology companies, civil society, and even members of Congress. Each one made the directive draft, its implementation guidance, and this final VDP template better."

He further explained that several of the submissions asked whether the mobile apps that agencies offer to the public would be in scope of these agency VDPs, which was something they hadn't considered before and agreed that should be. A few comments suggested several ways of thinking about the problems that would remove ambiguity around the scope, but including vulnerabilities and misconfigurations, which was something that had not been in scope before. They also talked about reporting requirements which stop when everything is in scope, and how to respond to anonymous vulnerability reports. And of course there's no response possible because they're anonymous.

He said that a number of comments discussed their use of target timelines concerned that the directive not mandate specific deadlines for remediation. Fixing a vulnerability is not always a push-button, and requiring deadlines might create perverse incentives where, for example, a lower severity but older vulnerability might take organizational precedence over a newer but more critical bug. He said that imposed deadlines might also cause rushed fixes. So he said the final directive makes clear that the goal of setting target timelines in vulnerability disclosure handling procedures is meant to help organizations set and track performance metrics. They are not mandatory remediation dates.

So anyway, I've got, for anyone who's interested, links both to a PDF of the detailed recommendations and also sort of a web-friendly version. Essentially what it means is we will have a unified set of guidelines by the end of the first quarter of next year, which will serve to pull the U.S. government agencies together and make it very clear how researchers can interface with different agencies, how to communicate with the agencies, what is and is not fair game for vulnerability disclosure, and also to make it explicitly safe for the information security community to poke at sites when it's clear that the poking is meant to be in favor of the security of the site and not to take advantage of it. In other words, white hat versus black hat. Or I guess now we say good hat versus bad hat.

So anyway, all of this is addressed in this what they called the "Binding Operational Directive" for U.S. agencies, each of which will be putting out their own version of that tweaked in order to fit their specific requirements. But everybody's going to have to have one in place by the end of the first quarter next year. So I think that sounds like a great plan.

Tor has launched what they called a "membership program," starting or effective, it was announced, last Monday. They announced it as a new means for nonprofit and private sector organizations to financially support the Tor Project's work. And we've talked about it a little bit in the past. Everybody knows what the Tor Project is. It's been one of our really cool technology topics that we've talked about on the podcast through the years. As we know, the Internet was designed to work, and somewhat audaciously and incredibly at the time, it was designed to robustly interconnect up to 4.3 billion endpoints all at once. And that was only under IPv4. Of course now under IPv6 that number is just ridiculously large. But it was never designed, from the beginning, to incorporate authentication or privacy. We added that stuff later. But its fundamental nature has always been hostile to the provision of anonymity, which is the high bar that the Tor Project has set for itself.

In their announcement they wrote: "For a while, we've been thinking about how to continue to increase the diversity of funds in the Tor Project's budget; and, more importantly, how to increase unrestricted funds. The latest is a type of funding that allows us to be more agile with software development of Tor and other tools. We decided

to create a program inspired by what Tor is based on, community. Our goal is to build a supportive relationship between our nonprofit and private sector organizations that use our technology or want to support our mission."

So the five founding members of this new initiative are Avast, DuckDuckGo, Insurgo, Mullvad VPN, and Team Cymru. And perhaps that number will grow over time. As we know, the Tor Project is unique, and it clearly has a place in our global Internet ecosystem. I think there was a round of layoffs some time ago. There was some problem with funds. They get a lot of their funding by first generating a proposal and submitting it to various organizations, which then have to look at the proposal, put it through their funding cycles. As I've researched a little more deeply into this, that often has something like a six-month turnaround.

So what that does is it makes it very difficult for them. They have to decide what they want to do, generate a proposal, submit it into the budget cycle of the entities from which they get funding, and then sit back and wait six months. Which means it's very difficult for them to do things in a much speedier fashion. The idea of having a bunch of organizations that are interested in supporting the Tor project and are not tied into specific development budget cycles will allow them to operate much more quickly. And again, as I said, I just think Tor, you know, it's trying to do something very difficult to do on the Internet. We've talked about how hard it is to do this. But it's got lots of good applications. Also, unfortunately, some sketchy ones, too. But still, anonymity has a place. And I think this will make the project a lot more nimble going forward.

In the middle of July this year, Intel released updated microcode for a dauntingly long list of processors. It was updated to incorporate the latest round of fixes for the most recent results of academic researchers further impacting Intel from a security standpoint. That microcode, which can currently be loaded by Linux whenever it boots, is posted on GitHub, and I have a link in the show notes. There's also a separate link for the even newer 10th-generation Ice Lake microcode. And I have that link also in the show notes. Microsoft has packaged and last week just released updated microcode for Intel processors, though not yet for Ice Lake.

Since these will not be included in Windows 10's monthly updates, anyone feeling that they need to have them for enhanced security will need to go get them deliberately. And as we know, given that there has never been even one proven successful exploit of these edge case flaws outside of academic research, and the fact that they measurably and in many cases significantly reduce our processor's performance by disabling previous performance optimizations, and given that none of these are code execution flaws, and none of them certainly remotely available, they are all only a possibility for information leakage at the margins. I am certainly not going to bother. Probably ever. Okay, "ever" is a long time. We don't know what the future holds.

But aside from being great to discuss here on this podcast from a theoretical computing security standpoint - and it has been wonderful to talk about these things because they're interesting from a technological theoretical level - there's absolutely no reason in my opinion why any end user running a personal workstation, even in an enterprise environment, would ever have any need for any of this. And as I said the last time we talked about this, I really could see Intel eventually offering two separate families of processors. You know, just coming out with their highest performance family, which offers all of these performance benefits, with an understanding that there's going to be, in order to get the performance, there's going to be some potential for cross-process information leakage. But in return, you're going to get all the performance you could ever ask for.

And then have a separate lower performance zero-tolerance family which could be used for those far less common situations where untrusted processes might be sharing

common hardware, and where absolutely no possibility of cross-process leakage can be tolerated. Maybe they'll do that, or maybe it'll be a simple switch that you can throw. We know that there are means in the private registers of the chip for turning on and off some of these mitigations, so there is already some of that.

The affected processor families are Amber Lake, Avoton, Broadwell, Cascade Lake, Coffee Lake, Comet Lake, Haswell, Kaby Lake, Skylake, Valleyview, and Whiskey Lake. But again, not Ice Lake, the 10th-generation processors. And across that set of processor families there is a separate Windows 10 update for each of the eight different versions of Windows 10. And I'll comment again I thought there was only going to be one version of Windows now. Apparently I was quite wrong about that. I've got a link to the Intel announcement with all of the processor guidance specifics for this incredibly long list of processor variations underneath each of those families.

Lawrence Abrams over at BleepingComputer has compiled all of the various links for all of the various Windows 10 releases, including the server versions, which is maybe where you might feel you need it. But even then, only if you're hosting VMs with untrusted code executing. I just, you know, this just seems like so much on the margins to me that it really seems like where we are now, several years downstream of the first of these edge cases that you and I, Leo, started talking about in 2019, and nothing still has continued to happen. I just can't see any strong basis for people doing this.

Leo: Yeah, yeah. It's too hard.

Steve: Yeah, just doesn't make sense. And it's interesting to me that Microsoft's not automatically pumping all these out because they know it's going to impact performance, and nobody wants that.

Okay. So that's all of the security news of the week. There was, I wanted to mention - this was often tweeted to me, as well, although just recently. There was conducted in a hospital in Spain, just published last Thursday in JAMA - the Journal of the American Medical Association, so one of the top most respected journals. This was the result of a very small, only 76 people, but it was an extremely well-conducted, randomized, double-blind study, so neither the patients nor the physicians or hospital staff knew who was getting what. It was a study giving Vitamin D to these 76 patients in a 2:1 ratio, meaning that 50 of them received Vitamin D in addition to all of the hospital's standard COVID-19 best care practices; 25 were not given Vitamin D, but they also still received all of the hospital's standard best practices.

I've got a link to Chris Masterjohn's blog posting about this which just came out. There's also a YouTube video. Actually, I also made it the grc.sc Shortcut of the Week, so grc.sc/783. This is a well-known Dr. John Campbell talking about the study. He's an M.D., and he has 740,000 YouTube subscribers. Anyway, he's just jumping up and down about this. The study was so significant, despite the fact that only 76 people were involved. Of the 50 patients who did receive Vitamin D, zero were put in ICU. I'm sorry. One of the 50 patients on Vitamin D ended up in ICU, zero deaths. Of the 26 who did not receive Vitamin D but otherwise received all of the identical care, 13 of the 26 went into ICU, 11 of whom died.

So, I mean, this is - we've talked about Vitamin D before. We saw an earlier chart that was posted. This keeps being in the news. We keep seeing studies. And in fact this doctor is really furious that there isn't more attention being given to this because it looks like it's really significant. So again, I just wanted to mention all the links are here for people to check out for themselves. Vitamin D status looks like it is very, very significant for the outcome of coronavirus.

Leo: I take my 5,000 IU every morning.

Steve: Good, ditto. And everybody I know.

Leo: Thank goodness. Yeah. Thanks to you. Thanks to you, Steve, really. I mean, you get a lot of credit for that.

Steve: Well, I read about it, as our listeners know, years ago, and it was just so clearly important.

SpinRite's work is progressing nicely. The update to the earlier previous work from 2013 is now finished. The benchmark is running for that. It has been well tested for the last four or five days and got great results. So the next thing up will be my amalgamation of the new support for the older IDE/ATA mode controllers with the newer support for the AHCI controllers. Then I'll get it all packaged up for its broader first testing by everyone here. Maybe by next podcast, that might be pushing it a little bit, but certainly by the one after that. And I'm very excited to have our listeners able to play with this stuff, benchmark their mass storage drives, all mass storage drives.

In the process they will be verifying that SpinRite, the next SpinRite, 6.1, will work on all of their hardware and drives because that's the point of this early benchmark test is to really just pound the crap out of this code that I've just written because it will then be moving into SpinRite, where it will become the new core of SpinRite's direct hardware access technology. And then 6.1 will happen. So anyway, exciting times. We're getting close.

Leo: Woohoo.

Steve: Yeah. So as I mentioned at the top of the show, California's recent heat and humidity wave...

Leo: Ugh. Ucky.

Steve: Yeah. It's been brutal, Leo.

Leo: It really has.

Steve: But you know we shouldn't complain because most of the time we live in paradise.

Leo: Oh, I know. We pay the price, though, don't we.

Steve: Yeah, yeah. Amber waves of grain and [crosstalk].

Leo: Amber waves of fire. You should see, I mean, it's what, it's 4:00 in the afternoon. It looks like the twilight of the gods out there. It's gloomy, it's gray...

Steve: It's weird. All day the sun has just...

Leo: It's weird. The sun is red.

Steve: The color of the sky has just been - yeah.

Leo: It's not good.

Steve: So I decided that I wanted to obtain remote control for my workplace's air conditioning so that, for example, if the day was going to be extremely hot, and I was going to be arriving at my workplace early, I could have the AC kick on at 5:00 a.m.

Leo: Smart, yeah.

Steve: To begin cooling the place off so it's habitable by the time I get here. And, I mean, it's like 90 degrees in my office when I walk in.

Leo: Ooh, no, no.

Steve: It's like, okay, I've got to fix that. And I also wanted to have 24/7 temperature and humidity monitoring. I mean, like why not see what the temperature is? That's just kind of cool. And I guess I was just sort of dispositionally ready to start playing with this stuff a little bit. So I purchased an Emerson Electric WiFi cloud-based thermostat and a continuous monitoring and logging thermometer-hygrometer. Emerson Electric is a high-end commercial equipment provider that I had known of, and they also have a consumer branch. And I'm very pleased with my little $91 thermostat purchase. I don't need a touchscreen or high-resolution color display. I just want minimal remote control and monitoring with some state-of-the-art scheduling features.

The logging thermometer brand is Govee, that I've never heard of before. It's from the Chinese company Shenzhen Intellirocks Tech. Co., Ltd.

Leo: Okay, mm-hmm.

Steve: I know, right. Uh-huh. At my other location, Lorrie and I were annoyed by the old-school mechanical timer we had turning some little popcorn lights we have on a tree on and off at dusk, like on at dusk, off after we went to sleep. So I found just a cool little four-pack of WiFi outlets for $23 from Hong Kong-based Gosund. Once again, who? Yes, that's right, Gosund. All of these IoT technologies tie, of course, into my networks at both locations through standard 2.4 GHz WiFi. The very nice $39 Govee logging thermometer uses a little plugged-in base station which links to its indoor or outdoor remote sensor with a low power 433 MHz coded broadcast.

And while all this was great, I was quite conscious of the fact that, if I'm able to control the lights in my home while I'm out roaming around, and to similarly check the temperature and humidity at my office, I'm doing so by contacting servers in Shenzhen and Hong Kong. Which have in turn been contacted by the WiFi-connected IoT appliances which are now beginning to populate my two locations. And I do not mean to in any way besmirch these Chinese devices or their China-based services. These are amazingly inexpensive and capable devices backed by free services. And I don't care, frankly, if they go out of business in five years.

Starbucks charges more than $7 for coffee. I paid $6 for a miraculous WiFi-connected AC plug that can entertain any sort of complex schedule I can imagine with manual override. It's an incredible value. But it's not so incredible if it leads to intrusions into my home or work networks and gets me hacked.

So here we are, 15 years into this podcast. And we've recently covered critical vulnerabilities in the third-party TCP/IP stacks used by billions of similar embedded devices. I have no idea what's inside these little white plastic miracle pods. I have no idea how anyone can sell me one for $6. And I know I'm not alone. In fact, I'm probably nearly the last person to add some of this automation to my home or office. I've been playing with it now for a week. It's really cool. It all works.

But think about this. Right now at this very moment my world has three new persistently established outbound TCP links to external servers to which I am completely blind and over which I have no control. So imagine what a graphic of the United States would look like, showing probably tens, if not hundreds of millions of persistently connected IoT devices linking across the continents to their home-hosted servers and services. And now try to convince yourself that this hasn't occurred to some foreign power who may have agencies that are feeling a little defensive toward the U.S. at the moment.

Now, it's true that my various PCs and Apple devices have something similar. But we know that a huge amount of work is continually going into the maintenance of the security of those devices - our PCs, our Macs, our Apple devices. They're getting patches all the time. All the bits are being scrutinized. Does anyone imagine that my $6 AC plug is ever going to see a firmware update? Not a snowball's chance in hell. And that's fine, too, because it's working perfectly. But modest as it is, it is, though it's hard to believe, it's a computer. It's in my home. It's on my network. And it is always connected to China.

Which brings us to the title of today's podcast: "IoT Isolation Strategies." Because the one thing I didn't say is that, despite all of these multifarious potential threats, both my home and workplace networks are today completely safe from external intrusion and attack through any defects or flaws that those incredibly cost-effective, affordable, and feature-packed devices might have. And the question is, is everybody else's network who's listening to this similarly safe? We've spoken about this before, but I thought that it would be useful to do a refresher.

My own actual use of these technologies sort of like brought this home to me. I ended up having to use different strategies in my two locations because I just had different hardware available to me. Obviously, all these IoT devices connect via WiFi. So the easiest and most straightforward solution is to use or obtain a WiFi router which offers a guest WiFi feature. In my home location we have an ASUS RT-AC68U, a nice router, a few years old now. I recently updated its firmware. So ASUS is keeping it current. And it offers a guest WiFi feature. The guest WiFi has a different SSID, so it comes up in the list, and you can see it. It has its own password and, crucially, no access to anything other than the Internet. That's what the guest WiFi feature on one of these later model routers offers.

It may, however, have access to Universal Plug and Play, UPnP. That I'm not sure of. And of course any UPnP might be programmable to allow incoming unsolicited traffic to enter the non-guest network. I don't know one way or the other about that. That's something I haven't tested because, for me, UPnP is the first thing I disable when setting up a network. Of course all of our listeners know that it's by design an inherently and entirely unauthenticated protocol. So any device on the interior LAN is able to use UPnP to establish connectivity to it at its whim. It's important to disable it, if you can, or to really restrict its control using some additional firewall rules if you need to.

So you'll want to make certain that guest WiFi also has no access to the router's management web interface. It shouldn't, but be sure by trying to access the router's web management and make sure that you just don't get any connection at all. And really, at this point, as you're connecting things to your network, it's really crucial that you use an impossible-to-guess username and password for the management interface of your router. There's just no reason not to. You don't need to get into it very often, and you definitely don't want foreigners to get into it ever. Your browser will remember its wacky login username and password.

So just make one of them that's impossible to guess, absolutely crazy impossible to enter even by hand, and leave it up to your password manager to fill that in. And by all means take advantage of the fact that you have two crazy fields, both the admin and the password. Make them both insane. Why not?

So at home, when I was setting up that little $6 AC plug, I switched my iPhone over to the guest network, so that was the network that the plug would see when it paired up with my phone, and I gave it my guest login username and password. So that's what the little plug is using. It is an untrusted visitor in my home forever, and it should be in everyone's. Okay? But let's suppose that your WiFi router doesn't have a guest mode option. There's really no safe way to share a WiFi access point with IoT devices where everyone is on the same wired and wireless network.

I did give it a bit of thought. For example, an IoT device will almost certainly have a fixed MAC address, and it will always be configured via DHCP. Probably you don't have any choice. It's just, you know, oops, look, it's on the network, yay. Right, easy. Okay. But the fact that it has a fixed MAC address means that, if your router allows for MAC-based DHCP assignment of fixed IPs, you could arrange to group all of your IoT devices within a fixed IP block within your network. Then use router firewall rules to block that region's access to the rest of your network.

But that's a lot of work, and it's still not a great protection since the devices would still be on the same Ethernet broadcast domain, and they would have access to your router's management interface, which just makes me nervous. A better solution is to pull a retired low band, that is, you know, a 6.4 GHz-only WiFi router or access point out of the garage, you probably have one that you retired laying around somewhere, and set it up as an isolated guest IoT network. And that's what I did here at my workspace.

So I have two access points whose network traffic is isolated from each other at the router. We'll talk about that in a second. But because I'm also a bit of a belt-and-suspenders guy, and because it was possible, I also disabled the 2.4 GHz radio of my primary WiFi access point. All I need there is 5 GHz. And without exception, all of my IoT devices, the three that I have so far, are 2.4 GHz only. And that's probably not going to change anytime soon because 2.4 GHz is less expensive. Going dual band would be more expensive.

Leo: Well, and most IoT devices don't have 5 GHz.

Steve: Exactly. There's no way they're going to do 5 for a $6 power plug. So all IoT is going to be 2.4. There's no way that any IoT thing could even see the high band, the 5 GHz. So anyway, I've got my - just because I could, I've got my IoT devices on a different, completely different frequency than my iPads and iPhone and laptops that are all able to use the faster 5 GHz band. As we know, 5 GHz has a different penetration and reflection characteristic, so maybe it might be a problem, but you could try it in your home. Turn off the 2.4 GHz radio and see if you still have good connectivity for all your stuff wherever you are. In which case you could just dedicate that to IoT.

As for separating the IoT access point from the rest of your network, there are two solutions. If you have a pfSense-based router, as I do, my little - I've talked about the SG-1100 from Netgate. And I have an older, can't remember the supplier, but it's pfSense based. Actually, I have them in both locations. Or if you've got one of those little nifty Ubiquiti Edge Router X's that we were talking about years ago, either of those use an actual individual NIC, a network interface, for each physical port.

That allows you to strongly isolate the IoT network by placing on its own subnet. You could leave your home on 192.168 dot whatever dot whatever, and set up the IoT network on a different private subnet, like 172.16 dot whatever dot whatever. That way they will be on entirely separate Ethernet broadcast domains, no way for them to see each other. And the IoT network will have no way to see anything else. It'll be able to freely access the Internet, hook into cloud services anywhere. But there's no way that anything that goes wrong there would be able to infect the rest of your network. What?

Leo: Forgive me, though, because this has always bugged me, even with the more complicated three-router solution. One of the features of IoT devices, maybe you don't care, but is that I can sit at home on my network with my phone and access that device from my phone.

Steve: So I'm able to do that.

Leo: You can do that across the other network?

Steve: Yes.

Leo: Okay.

Steve: Yes, because they are each going out to the remote service and bridging that [crosstalk].

Leo: So they're NATing it that way, NAT traversal that way, rather than going direct.

Steve: Well, they're actually not...

Leo: If I connect, if I do my Hue lights, you're saying I'm going to the Hue server and then coming back in, as opposed to just across my network. Oh, okay.

Steve: I think that you'll find that that's what happening, yes.

Leo: Oh, okay, all right. That explains it. Yeah. So as long as they're network addressable from the Internet...

Steve: Correct. Now, I should mention that I'm sort of in crawl before you walk mode. I don't have a home hub or anything else. All of these things are hubless devices.

Leo: Right.

Steve: But they're working great for me with iPads and iPhones from all my locations.

Leo: That makes sense.

Steve: Because they're all reaching out to cloud services.

Leo: Right. You're going to a third party. You're doing kind of like a NAT traversal kind of thing.

Steve: Exactly.

Leo: Okay.

Steve: Correct. And of course therein lies the reason you want the isolation because I'm going out to a third party.

Leo: Right. Well, I understand. Yes, I understand. But you don't want somebody to come in from the outside world onto your network with your devices via this Internet device, so it's important that this Internet device have a barrier between you and your devices. And I always thought that would hamper operation. But if you're saying I'm going out to public Internet every time, then it wouldn't, obviously, yeah.

Steve: Right, right. All these things now are cloud based. So anyway, the first idea is to use a router, if you have one, where you've got individual NICs that allow you to set up separate networks per port. Then you would take your older low-band 2.4 GHz-only router, use it as an access point, plug it into the port to which you have assigned a different subnet, and it's on the 'Net. It's able to do everything it wants to. But there's absolutely no way that it's connecting to the rest of your network. So that's the solution, if you have one of those routers.

If you don't, then we talk about the cool old solution that we talked about that you mentioned, Leo, years ago. You don't really need a three-router Y. A second router is really all you need. You would put the insecure one on your WAN, and then your secure router would plug into it. And it would use the one-wayness of your secure router to protect your secure network and your 5 GHz-band radio from everything on its WAN side

which would include the insecure network and then your connection out to the public Internet.

So the idea is it's a two NAT router solution where the inner NAT router is your secure one. The outer NAT router is the insecure one. Your Internet work, where you care about security, is protected by its router in the same way that it would if it were right on the WAN. If it were right on the public Internet, everything upstream of it would be the Internet. Now you have your less secure network is part of what is upstream of your main secure router, and no way for anything to crawl back into your secure router that is out on the WAN interface, as is the IoT network. I think that's, you know, it's quick and easy, simple. It does require a little additional hardware.

But if you've got a second WiFi router around that you're not using, make it the router that connects to your home's WAN, and then plug your secure router into it. You've then got the double NAT solution protecting your home from anything that might happen on the insecure network. And then I would say, you know, go to town with IoT, and security will not be a big concern.

Leo: Very cool. Very cool. Yeah, you solved the one conundrum that I've had all along, and I should have asked you years ago, which is how do you talk to these things. I wonder, though, if there are some IoT devices that you address directly via their local IP address. I just don't know. I just don't know.

Steve: I wonder how you would know what it was. I guess if you were to broadcast an Ethernet, if you did an Ethernet broadcast you could find it. In the same way, for example, that computers now find HP printers.

Leo: It's Rendezvous, used to be called Rendezvous. It's Bonjour now or whatever. There's an Apple technology for discovery, zero-config discovery. I mean, these things happen. I just, you're right, it would make a lot more sense just to connect to a third-party server from both sides. You don't have to find it.

Steve: Well, and all of these servers are using cloud functionality.

Leo: Yeah.

Steve: For example, the thermostat, or the thermometer- hygrometer from Gosund, it's logging. It's doing two years of cloud logging. So anytime I want to I'm able to bring up an up-to-the-moment log of past temperature and humidity...

Leo: That's cool.

Steve: ...which my phone doesn't store. It doesn't store. It's sending it off to Hong Kong, where the service exists. And the Gosund app then reaches out to the cloud service in order to update its log. So, I mean, and similarly...

Leo: Completely connecting to their server, yeah.

Steve: ...these little $6 plugs, they don't have the schedule in them.

Leo: Right.

Steve: They have, yeah, exactly, they have a persistent connection, and the service maintains the schedule and turns these things on and off for me. So for $6.

Leo: You know what, it's amazing because these things are so fast, I'm thinking of my Nest cameras or my Nest Hello doorbell. You can quickly see them. But there is a little bit of a lag as you probably are, you're going to the cloud. They're going to the cloud, and you're getting the picture back from them. That does make a lot of sense. And they don't operate if you don't have Internet access. They don't just operate in the network without Internet access; right?

Steve: Ah, right. So it's not just LAN.

Leo: Yeah, it can't just be LAN, yeah. Well, that makes a lot of sense. I just unfortunately, I could have done this, but I just unfortunately got the whole house wired, and it's got a central router that everything's connected to. So most everything's Ethernet. There's still WiFi. I guess what I could do, the WiFi is - I'll figure that out. I could guess I'd put a second WiFi network in there that's only 2.4 GHz.

Steve: Exactly.

Leo: You don't want multiple WiFi networks though; right? Aren't they going to collide with each other? I guess they...

Steve: Well, and actually that is the other advantage of using 2.4 and 5. They won't collide with each other. And so you're giving your IoT stuff its own territory to have. And then all of your laptops and iDevices, they're all able to run at 5.

Leo: So I could get - I have plenty of cheap - I have lots of leftover WiFi routers. I have the Edge Router X. I could put the Edge Router X, plug that into the Ethernet, and set up a new 172-dot subnet.

Steve: Exactly.

Leo: With that WiFi router. And then just have that be the 2.4 GHz. That's a good idea. That doesn't seem too hard to do.

Steve: And Leo, I tell you, you know, I mean, if you can imagine a globe, we've all seen those globes of the missile traces in war games or something. Can you imagine a picture of the connections that are from even just the U.S.'s IoT devices; you know? Lord knows how many you have.

Leo: Oh, my god, yeah.

Steve: I've got three. I mean, there's just hundreds of millions of connections.

Leo: I have three per room, yeah.

Steve: Yeah.

Leo: No, every room has at least an Amazon Echo and a Google Assistant in it. Obviously those all go to the central cloud.

Steve: Yup.

Leo: I can control my lights, my Hue lights from either. But obviously that's going to the cloud and then coming back down to the hub and doing it. So yeah, yeah, I could do this. It's going to be fun. But, yeah, interesting. I bet no one does this, though. Just, you know, a handful of paranoid people.

Steve: And this was my point, Leo. There will be, mark my words, there will be an exploit of some sort of widely dispersed IoT device where there's 100 million individual homes that are all linked to some service, or 10 million or something, where a vulnerability is found that allows bad guys to get into, like, all of those internal networks. It is a way in, a way past everybody's NAT router. We've been talking about NAT routers as a firewall forever. These things, this little $6 power plug is sitting inside my network calling China.

Leo: Right.

Steve: With a persistent connection.

Leo: Yeah. Wow. Yeah. That's scary. Even if you're buying big name brands, and they're constantly being updated, firmware updated and stuff, it's still scary. Yeah, you're putting a server in your house. Inside your house you're putting a server.

Steve: Mistakes happen. Mistakes happen. And they're keeping this podcast in business.

Leo: Yeah, we've always said, as soon as you put a server inside your network that is publicly accessible, which these have to be, you're at risk.

All right. Thank you so much. Always a pleasure, Steve Gibson. Let's hear it for him. Little golf clap. Little round of applause. Steve Gibson. GRC.com is his website. That's where you can go to get copies of the show. He's got 16Kb versions of it for the bandwidth impaired. He's got the normal 64Kb audio. He's also got

transcriptions. Those are all available, thanks to Steve and Elaine Farris, who writes everything down. And GRC.com's the place.

While you're there, pick up SpinRite. Why not? You're already there. It's Steve's bread and butter. It's the world's finest hard drive maintenance and recovery utility. You buy it today, 6.0, you'll be getting 6.1 the minute it's available. Plus you can kind of participate in the development of 6.1, I know. That's kind of part of the privilege of being an owner.

Steve: It's true. Yeah.

Leo: Yeah. GRC.com. Lots of other stuff there. If you're interested in Vitamin D, read up on that and all of Steve's health research. That's all also there. It's one of those sites you get to, and you go, oh, that's interesting. Then find another thing, another thing. Plan a couple of hours at least. All afternoon. In fact, plan a week. What else are you going to do?

We also have versions of the show at our website, TWiT.tv/sn. There's audio and video there. You can watch us do the show live. We try to be about 1:30 Pacific, 4:30 Eastern, 20:30 UTC of a Tuesday afternoon or evening. But sometimes it runs a little late, depending on the morning shows. Easiest thing to do, go to the website, TWiT.tv/live. It's live all the time. There's always something going on, and you can watch the behind-the-scenes, watch other shows, and on a Tuesday afternoon watch Security Now! with Steve Gibson.

The best thing to do, though, is subscribe. Find a podcast application, search for Security Now!, subscribe to it. We're everywhere. Been around 15 years, you better believe we're everywhere. And that way you'll never miss another episode. And I know you don't want to miss an episode. Thank you, Steve. We'll see you next week on Security Now!.

Steve: Thanks, buddy.