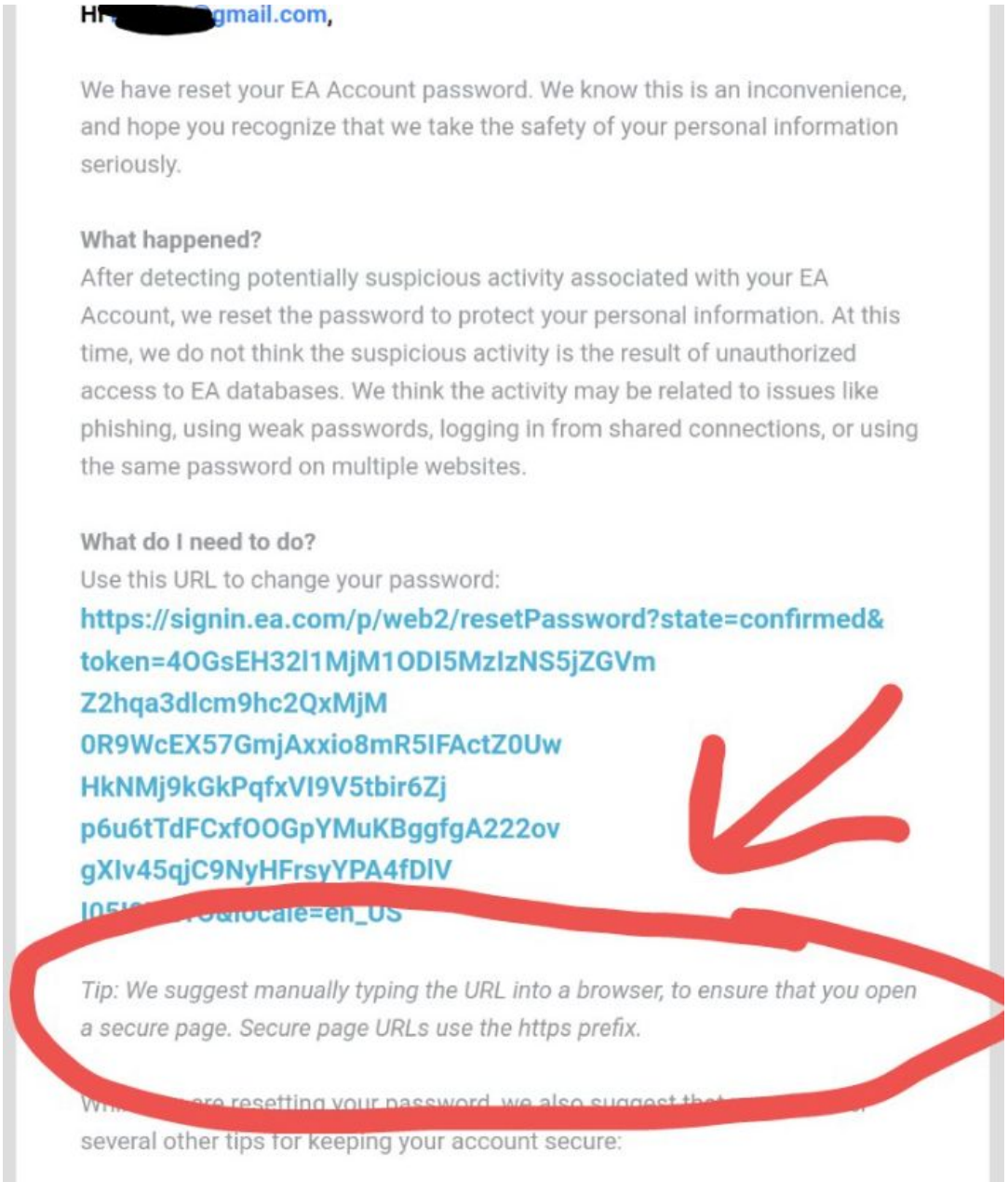
Security Now! #783 - 09-08-20

IoT Isolation Strategies

This week on Security Now!

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.



Browser News

DoH coming to Chrome for Android

Although the desktop releases of Chrome have offered the enhanced security and privacy of DNS-over-HTTPS since Chrome 83, it wasn't added to the iOS or Android editions. But last Wednesday's Chromium Blog was titled: "A safer and more private browsing experience on Android with Secure DNS". Paraphrasing from Google's posting for brevity and to skip a bunch of boilerplate stuff that we already know, they said:

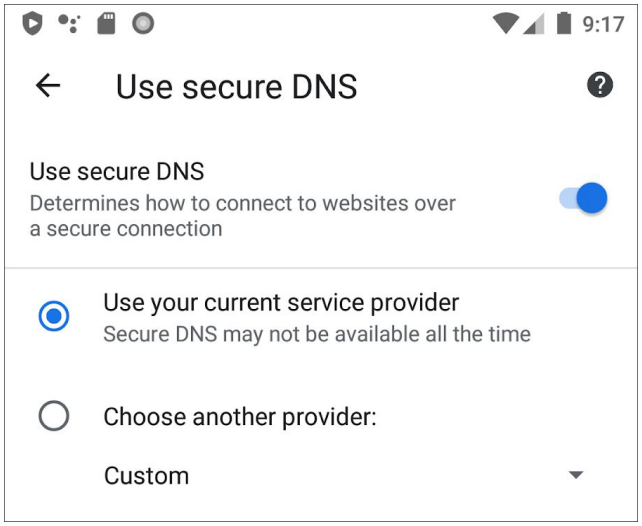
With Chrome 85, we're extending support of Secure DNS in Chrome to Android. As we did for the launch of DoH on Chrome for desktop platforms, we will progressively roll out DoH on Chrome for Android to ensure the feature's stability and performance, as well as help DoH providers scale their service accordingly.

- Android Chrome will automatically switch to DNS-over-HTTPS if your current DNS provider is known to support it. This also applies to your current Android Private DNS (DNS-over-TLS) if you have configured one. This approach means that we can preserve any extra services offered by your DNS service provider, such as family-safe filtering, and therefore avoid breaking user expectations. In this automatic mode, Chrome will also fall back to the regular DNS service of the user's current provider (including DNS-over-TLS if configured), in order to avoid any disruption, while periodically retrying to secure the DNS communication.

  [I presume what they're referring to here is that if a particular DoH provider becomes overwhelmed and unable to service the request load, that will be recognized and Chrome will fall back to whatever it was using previously.]

- 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.

- If you are 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.



While this milestone represents significant progress toward making browsing the web safer and more private, it’s still early days for DNS-over-HTTPS. As such, we remain open to feedback and collaboration with interested parties such as mobile operators and other ISPs, DNS service providers, and Online Child Safety advocates to make further progress in securing DNS.

When it becomes available the so-called “Secure DNS” will be enabled by default for all users, and once turned on, Chrome will attempt to make its DNS queries via DoH, where supported, and use traditional plaintext, in the clear, UDP DNS as its fallback.

Under the hood, Google said the feature works the same way they’ve implemented their desktop versions of Chrome so that users won't need to tinker with the Android's OS’s DNS settings. Chrome will maintain an internal list of DoH-capable DNS servers, and if the user has one configured as the OS-wide DNS setting, Chrome will use that server's DoH interface instead of the default one, and replace plaintext DNS queries with encrypted DoH queries on the fly.

And in situations where users don't want to change their Android device's system-wide DNS server to one that supports DoH, Google allows users to customize Chrome's DoH server just for their browser alone. In the screenshot above, the “Choose another provider” option allows users to provide the IP addresses of the DNS servers they wish to use only for browsing.

To operate seamlessly within the managed environments of corporate networks, Chrome for Android will automatically disable DoH if it finds itself operating in such an environment since IT staff typically deploy enterprise-wide policies to control the operation of the company's smartphone fleet for enhanced security.

Bye Bye Drive-By

We’re currently at Firefox 80.0.1 since last Tuesday which quickly followed 80 to offer a couple of simple bug fixes. Specifically, 80.0.1 ...

- Fixed a performance regression when encountering new intermediate CA certificates
- Fixed crashes possibly related to GPU resets
- Fixed rendering on some sites using WebGL
- Fixed the zoom-in keyboard shortcut on Japanese language builds
- Fixed download issues related to extensions and cookies

But it's 82 that I wanted to talk about...

Because two editions from now, point releases notwithstanding, Mozilla will finally be catching up with a useful security feature Chrome has sported since Chrome 73 in March of last year, 2019... And that feature is blocking so-called drive-by downloads from iFrames.

Frankly, for any browser, I'd be asking “what the heck took you so long?” The answer, of course, is the nearly terminal fear of breaking expected website behavior. If a browser update causes some site's features to stop working, it's not the site that gets the blame. “Hey! This worked before I updated by browser to its latest version” is not something that any browser developer wants to hear.

We talked about this recently. It's possible to Javascript running on a page to, itself, autonomously, initiate a download of a file by a browser. Any of those sketchy download repository sites which give us that "Your download should start in 10 seconds..." message is doing exactly that. They are deliberately holding us on an advertisement-laden page to give those ads more eyeball time, before finally deigning to trigger the download we're looking for. Sourceforge is the less-sketchy site where I most often encounter this behavior. For Chrome, in its default behavior of not asking for the download destination, the download can easily go unnoticed. It's just a little box down at the bottom left of the browser's window. And the last time I talked about this behavior I said that I was going to change Chrome's default so that I would be asked every time. I did that and now I am.

Okay, so we've established that a website's Javascript is able to trigger a download which, in the default case might go unnoticed by its user. Later the user might wonder what that that file is and click it to find out. Not good.

The functionality that Chrome provisionally first disabled last March of 2019, but which could be restored by setting a flag, was blocking code in sandboxed iFrames from being able to script a file download. As I said... "What the heck took so long?" And, then, finally just this past May Chrome made it permanent by also removing the option to re-enable it.

Our browsers are generally erecting more and more complete iFrame sandboxing. Always with the intent of not breaking anything that's reasonable to expect. And in next month's October release of Firefox 82, Firefox will finally get drive-by download blocking for iFrames.

We discussed sandboxing iFrames many years ago when it was first introduced. Chrome 4 and Firefox 17 were the first versions of each browser to support it. The idea is that by adding the "sandboxed" attribute to an iFrame definition a large set of dangerous behavior could be disabled. The intent was to recognize that iFramed content — often and typically sourced by an untrusted advertising server — should not be treated like a first class citizen, but should be regarded with skepticism. Since the hosting page defined the frame, it's able to decide what that frame's loaded content is able to do.

By default, the appearance of the "sandboxed" tag shuts down a large number of potentially dangerous behaviors. But because it's concealable that a website may wish to explicitly allow content being served from within an iFrame to trigger a download, permission to do so can be preserved by adding an "allow-download" term to the sandboxed tag.

As ever, we are inching forward bit by bit, undoing the too-permissive security mistakes we collectively made as an industry back when we were still amazed that all of this stuff worked at all. Now we know it does and we're wishing that it was also a lot more secure in the way it works.

Security News

By far the most tweeted-to-me news takes the top slot in this week's general news. And, Leo... You're getting your wish. They must have been listening to you and many others in the community, because last Thursday's blog posting was titled:

"Threema Goes Open Source, Welcomes New Partner"
They wrote:

> A new chapter is added to Threema's success story.
>
> **Strengthened Through Partnership**
> After an intense startup phase, Threema lays the foundation for continuity, further growth, and an acceleration of the product development thanks to the entry of the German-Swiss investment company Afinum Management AG.
>
> Afinum fully shares our values regarding security and privacy protection. The additional resources gained through this partnership enable Threema to grow beyond the German-speaking part of Europe, and we can use our energy for visionary new ideas and projects. That said, Threema's founders – Manuel Kasper, Silvan Engeler, and Martin Blatter, all software developers – will continue to lead the company and still retain a significant ownership interest.
>
> **Open Source and Multi Device**
> 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. This is to say that anyone will be able to independently review Threema's security and verify that the published source code corresponds to the downloaded app.
>
> In the future, it will be possible to use multiple devices in parallel thanks to an innovative multi-device solution. 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.
>
> In conclusion, Threema will become even more trustworthy and even more convenient to use.

As our listeners all know, I love Threema because their system is the least easy to use. By that I mean that all key management is up to the end user. That doesn't mean that it's difficult. That just means that it's not transparent and automatic. And my point is, if you actually do care about security then managing your most important secret — yours keys — is just not something that you can delegate. If you want the appearance of very strong security, then by all means leave your key management up to the system's provider. Problem solved. But then that means you are implicitly trusting them. And that's fine if you do trust them — today and tomorrow and whatever might happen. But "TNO" stands for "Trust No One" and that's a pretty binary thing.

Explaining in a communication with ZDNet, a Threema spokesperson said: "Being advocates of the Open Source initiative (one of our founders created the m0n0wall project that went on to become the basis for many security and firewall products, both commercial and non-commercial), we have been thinking about this step for a long time. And of course the users have often asked for it, too. Now we are in a position that allows us to go Open Source without endangering our business model and our source of income."

So it sounds as though they built Threema up to the point that it was valuable enough so that they could peel off some of the ownership equity to raise a pile of cash to fund their further ambitions.

This makes Threema the 3rd end-to-end instant messaging services, behind Signal and Wickr, to open their source for inspection and, presumably, community contribution.

Having always been a fan of Threema's approach, I'm delighted by this news since it's certain to give Threema a well deserved boost in popularity. And, unlike Signal, Threema has never tied its users' account creation to their mobile phone number, so it also has that going for it. And we were just talking about Threema because they added video calling through the WebRTC standard — and had to fix the standard along the way to fully protect their users' calling metadata which was otherwise being exposed and reducing privacy.

WordPress File Manager

It's not good when a 0-day flaw is discovered being actively exploited in an extremely popular plug-in for Wordpress. And it's also somewhat jarring that we keep covering exactly such news. In the latest of a continuing series of such vulnerabilities, the WordPress File Manager plugin is being actively-exploited to permit full website hijacking. That'll ruin your day.

The Sucuri WordPress security team said that the vulnerability was introduced into the May 5th version 6.4 of Wordpress File Manager, which 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 in use by more than 700,000 active Wordpress installations.

The mistake was minor but it had major consequences. One of the plug-ins files was renamed during development for testing purposes from its safe inactive form to its dangerous active form. And then the project, with that renamed file, was distributed. The file was “connector-minimal. php-dist” but it was mistakenly renamed to “connector-minimal.php”. 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 it might download it to you if you were to query for it. But it wouldn't 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 any attacker needed was to invoke that script remotely and have at it. Which is exactly what then happened.

Because what that errant file permitted was bad. Or, as the Securi team said: “leaving such a script — intentionally designed to not check access permissions — in a public build causes a "catastrophic vulnerability if this file is left as-is in the deployment.” 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.”

The solution which appeared in the replacement v6.9 distribution was simple enough: simply delete the file and any other unused .php-dist files. However, a week before the file was removed a simple working proof-of-concept appeared on GitHub, which led to a mounting wave of attacks against websites until they were updated to version 6.9.

Sucuri says the exploit rapidly gained traction. The first attack was spotted last Monday, August 31, 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 increased to an average of 2,500 attacks per hour. And by the next day, September 2nd, Securi 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 it."

Later analysis showed that the flaw is in File Manager v6.0 to v6.8. Statistics from WordPress show that currently about 52 percent of installations are vulnerable. With more than half of File Manager's installed base of 700,000 sites vulnerable and as of last Thursday, September 3rd, only 6.8% of those 52% of the total 700,000 vulnerable WordPress websites had updated to the new, patched version of the plugin.

Facebook's new VDP — Vulnerability Disclosure Policy

The many Facebook platforms run on a bunch of code. Much of the code is theirs, but as is increasingly the case as the industry matures, code pulled from many 3rd-party libraries is also often used. So the question becomes, what should Facebook do when it finds a problem in some 3rd-party's code? This is sort of like a commercial version of Google's Project Zero.

Facebook has formalized and published their policy which is, I think, well thought out and reasonable. It is quite similar to Google's Project Zero, though with some inevitable differences:

> Facebook may occasionally find critical security bugs or vulnerabilities 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:
>
> 1. *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.
>
> 2. *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 clear when Facebook will disclose these issues.
>
> **Vulnerability Disclosure policy**
>
> *In a nutshell,* Facebook will contact the appropriate responsible party and inform them as quickly as reasonably possible of 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.

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.

Here are some details.

Reporting

- 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 will use contact methods including but not limited to emailing security reporting emails (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.

Mitigation & Timeline

- 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.

Disclosure

- Depending on the nature of the problem, there may be a number of disclosure paths: 1) we may disclose the vulnerability publicly, 2) we may disclose it directly to the people using the project, or 3) 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.
- 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.

Additional Disclosure Considerations

- 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,*** we might initiate the disclosure prior to the 90-day deadline when the delay might adversely impact the public.
    - ***If a project's release cycle dictates a longer window,*** we might agree to delay disclosure beyond the initial 90-day window, where reasonable.
- 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.

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 through our Bug Bounty Program.

https://www.facebook.com/security/advisories/Vulnerability-Disclosure-Policy

Facebook's new "WhatsApp Security Advisories" page

While we're on the topic of Facebook, they have also published a Security Advisories page for WhatsApp: https://www.whatsapp.com/security/advisories

The home page is pretty much boilerplate about their commitment to security and how they will work diligently to quickly address any security problems found in WhatsApp. That page also provides the rationale behind the Security Advisories:

> Due to the policies and practices of app stores, we cannot always list security advisories within app release notes. This advisory page provides a comprehensive list of WhatsApp security updates and associated Common Vulnerabilities and Exposures (CVE). 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.

The page for the list of CVE assignments for their problems during 2020 is gratifyingly short. In fact it's surprisingly short. Which is nice to see after watching Microsoft breaking monthly records month after month: https://www.whatsapp.com/security/advisories/2020/
The page lists a total of 6 CVE's, one of which was issued last year, in 2019.

U.S. Agencies Must Adopt Vulnerability-Disclosure Policies by March 2021
And speaking of vulnerability disclosures, we also have the flip side: Rather than how will an organization which finds vulnerabilities report them to those responsible... How does an independent security researcher who discovers a vulnerability in a U.S. government website go about reporting that?

To clarify this, last Wednesday, Bryan Ware, the Assistant Director of CISA, the US Cybersecurity & Infrastructure Security Agency, blogged about the formalization of the U.S government's policy.

In his posting Bryan explained that last November CISA asked for feedback on a draft binding operational directive which would require most executive branch agencies to create a 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 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," this one merited it. And even though the comment round spanned every holiday from late November to early January, the quantity and quality of feedback was nothing less than stellar. 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 our 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 agency VDPs, which was something they hadn't considered before, and they agreed that they should.

A few comments suggested some ways of thinking about the problems that would remove ambiguity around scope — by including vulnerabilities and misconfigurations — reporting requirements (which stop when everything is in scope), and how to respond to anonymous vulnerability reports. (No response 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 push-button, and requiring deadlines might create perverse incentives where a lower severity-but-older vulnerability takes organizational precedence over newer-but-more-critical bugs. And that imposed deadlines might also cause rushed fixes. The final directive makes clear that the goal of setting target timelines in vulnerability disclosure handling procedures is to help organizations set and track performance metrics; they are not mandatory remediation dates.

And he noted that many of the comments helped them to vastly expand and enhance the implementation guidance, particularly the FAQs. He said "We intend our guidance to be living, and we'll update the FAQs as questions come."

https://cyber.dhs.gov/bod/20-01/
https://cyber.dhs.gov/assets/report/bod-20-01.pdf

So... the result was published last Wednesday as "Binding Operational Directive 20-01"

The final report notes three of its primary objectives, saying:

Choosing to disclose a vulnerability can be frustrating for the reporter when an agency has not defined a vulnerability disclosure policy –the effect being that those who would help protect the public are turned away:

The reporter cannot determine how to report: Federal agencies do not always make it clear where a report should be sent. When individuals cannot find an authorized disclosure channel (often a web page or an email address of the form security@agency.gov) they may resort to their own social network or seek out security staff's professional or personal contact information on the internet. Or, if the task seems too onerous, they may decide that reporting is not worth their time or effort.

The reporter has no confidence the vulnerability is being fixed: If a reporter receives no response from the agency or gets a response deemed unhelpful, they may assume the agency will not fix the vulnerability. This may prompt the reporter to resort to uncoordinated public disclosure to motivate a fix and protect users, and they may default to that approach in the future.

The reporter is afraid of legal action: To many in the information security community, the federal government has a reputation for being defensive or litigious in dealing with outside security researchers. Compounding this, many government information systems are accompanied by strongly worded legalistic statements warning visitors against unauthorized use. Without clear, warm assurances that good faith security research is welcomed and authorized, researchers may fear legal reprisal, and some may choose not to report at all.

SO... all of this gets addressed and pursuant to this "Binding Operational Directive" U.S. agencies must have their individual vulnerability policies in place by March of next year.

The Tor Project Membership Program

Last Monday the TOR project announced their new membership program as a new means for nonprofit and private sector organizations to financially support the Tor Project's work.

As we well know, the Internet was designed to work. Audaciously and incredibly at the time, it was designed to robustly interconnect up to 4.3 billion endpoints — all at once. That was only under IPv4. But, as we also well know, it was not initially designed to incorporate authentication or privacy, and its operation is deeply hostile to the provision of anonymity... which is the high bar the Tor Project has set for itself.

In their announcement they wrote: "For a while, we have 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."

The five founding members are: Avast, DuckDuckGo, Insurgo, Mullvad VPN, Team Cymru. Perhaps that number will grow over time. The Tor Project is unique and it clearly has a place in our global Internet ecosystem.

Intel's latest microcode patches

In the middle of July, Intel released updated microcode for a dauntingly long list of processors. The microcode, which can be loaded by Linux whenever it boots, is posted on Github and I have link in the show notes: https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files

The there's a separate link for their 10th generation (Ice Lake) microcode: https://github.com/intel/Intel-Linux-Processor-Microcode-Data-Files/releases/tag/microcode-20200508

Microsoft has packaged, and last week released, their 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, they are all only a possibility for information leakage at the margins, I am certainly not going to ever bother.

Okay, 'ever's a long time. We don't know what the future holds. But aside from being great to discuss here from a theoretical computing security standpoint — which it certainly was — there's absolutely no reason why any end user running a personal workstation, even in an enterprise environment, would have any need for any of this. As I said the last time we talked about this, I really could see Intel eventually offering two separate families of processors: their "highest performance" family — which is what we would all get — and 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.

So, anyway, the affected processor families are Amber Lake, Avoton, Broadwell, Cascade Lake, Coffee Lake, Comet Lake, Haswell, Kaby Lake, Skylake, Valley View, Whiskey Lake. And across that set of processor families there is a separate Windows 10 update for each of eight different versions of Windows 10. (I thought there was only going to be one version of Windows now? Apparently I was quite wrong about that.)

https://software.intel.com/security-software-guidance/processors-affected-transient-execution-attack-mitigation-product-cpu-model

Lawrence Abrams has compiled all of the various links for the various Windows 10 releases. So rather than repeating that here, I've provided a link to his coverage of this at Bleeping Computer: https://www.bleepingcomputer.com/news/microsoft/new-intel-microcode-updates-for-windows-10-fix-cpu-hardware-bugs/

Miscellany (COVID-19)

"Finally Confirmed! Vitamin D Nearly Abolishes ICU Risk in COVID-19"
Chris Masterjohn has his PhD in nutritional sciences.
https://chrismasterjohnphd.com/covid-19/finally-confirmed-vitamin-d-nearly-abolishes-icu-risk-in-covid-19

This was a double blind, randomized control trial conducted in Spain, just published last Thursday in JAMA, the Journal of the American Medical Association. All patients admitted had acute respiratory infections. All patients were treated with the hospital's standard best practices. But a subset of the patients, chosen at random and unknown to the patients and to their doctors (thus, double blind) were also given a metabolized form of Vitamin D. It was a small trial of 76 hospitalized patients divided in a 2-to-1 ratio. 50 were given VitD, 25 were not.

Richard Tan @richard_tan
https://www.youtube.com/watch?v=V8Ks9fUh2k8
Worth watching on control group in Spain. Video upload 6th Sept, so recent. 50 patients given Vitamin D, 26 were not. 1 (2%) of 50 patients on VitD went into ICU, 0 deaths, 13 (50%) of the 26 without VitD went into ICU, 11 deaths.

Dr. John Campbell, 740K YouTube subscribers
https://grc.sc/783

SpinRite

Spinrite work is progressing nicely. The update to the early previous work from 2013 is finished and has been well tested with great results. So the next thing up will be the amalgamation of the new support for the older IDE/ATA mode controllers with the AHCI controllers. Then I'll get it packaged up for its first broader testing by everyone here who wants to see what's up with their mass storage hardware. :)

IoT Isolation Strategies

So, California's recent heat and humidity wave finally broke through my longstanding IoT resistance. I decided that I wanted to obtain remote control of 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 A/C kick on at 5am to begin cooling the place off. And I also wanted to have 24/7 temperature and humidity monitoring. So I purchased an Emerson Electric WiFi cloud-based Thermostat and a continuous monitoring and logging thermometer and hygrometer. Emerson Electric is a high-end commercial equipment provider, but they have a consumer branch, and I'm very pleased with my $91 thermostat purchase. I don't need a touch screen or a 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" from the Chinese company Shenzhen Intellirocks Tech. Co., Ltd.

At my other location, Lorrie and I were annoyed by the old school mechanical timer we had turning some little popcorn lights on at dusk and off after were were asleep. So I found a nice little 4-pack of WiFi outlets for $23 from Hong Kong based Gosund.

All of these IoT technologies tie 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/outdoor remote sensor via a low power 433 MHz coded broadcast beacon.

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, and to similarly check the temperature and humidity at my office, I'm doing so by contacting server's 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 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 might 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. But we're also 15 years into this podcast and we've recently covered critical vulnerabilities in the 3rd-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.

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 defensive toward the U.S.

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. 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 looks, it IS a computer; it IS in my home; it IS 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 those multifarious potential threats, both my home and workplace networks are completely safe from external intrusion and attack.

Are yours?

We've spoken about this before, but I thought that it would be useful to do a refresher.

All of 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. At home home we have an ASUS RT-AC68U which offers that feature. The guest WiFi has a different SSID name, its own password and, crucially, no access to anything other than the Internet.

It may, however, have access to UPnP, any UPnP might be programmable to allow incoming traffic to enter the non-guest network. That's something I haven't tested because UPnP is the first thing I disable when setting up a network. Since UPnP is, by design, an entirely unauthenticated protocol, it's now all the more important that it be disabled.

You'll also want to make certain that the Guest WiFi has no access to the router's management web interface. It shouldn't, but be sure. And it's also absolutely 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 often, and you definitely want foreigners to get into it NEVER. Your browser will remember its insane login username and password, so don't just make one of them impossible to guess. Take advantage of the fact that you have one of each and set both to something having as much entropy as possible.

So... at home, when I was setting up the 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. I gave it my guest login username and password, so that's what the little plug is using.

But let's suppose that your WiFi router doesn't have a Guest mode option. There's really no safe way to share a single WiFi access point with IoT devices where everyone is on the same wired and wireless network.

I thought about this a bit. For example, an IoT device will almost certainly have a fixed MAC address. And it will always be configured with DHCP. You just have no choice. That means that if your router allows for MAC-based DHCP assignment of fixed IPs, you COULD arrange to sequester all of your IoT devices within a fixed IP region, 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 great protection since the devices would be on the same broadcast domain and they would have access to your router's management interface.

A better solution is to pull that retired 2.4 Ghz-only WiFi router or access point out of the garage and set it up as an isolated Guest / IoT network. That's what I did at my workspace. So I have two acess points whose network traffic is isolated from each other. We'll talk about that in a second. But because I'm kind of a belt and suspenders guy, I also disabled the 2.4 GHz radio of my primary WiFi access point — all I need there is 5.0 Ghz. and, without exception, all of my IoT devices (the three that I have) are 2.4 GHz only, and that's not likely to change anytime soon. So, by disabling low-band on my primary secure WiFi and separating the radio frequency bands with IoT using the low band and my iPhone, iPads and Laptops all using the 5.0 GHz high band, there's just no way that the IoT devices can ever see into my privileged WiFi.

And 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), or one of those nifty little Ubiquiti

EdgeRouter X's, either of which use an individual NIC for each physical port, you can strongly isolate the IoT network by placing it on its own subnet. You can leave your home on 192.168.x.x. and setup the IoT network on 172.16.x.x. That way they will be on entirely separate Ethernet broadcast domains and the IoT network will have no way to see anything else. And you could add some port firewall rules to further block any traffic that might attempt to cross over.

Most routers are not a NIC-per-port, but have a single LAN NIC internally connected to a switch to give the router four or five external Ethernet port connections. So the question is, how do you isolate networks in this instance? We talked about this many years ago because creating separate isolated networks can be so handy. We can think of a NAT router as a one-way valve. Traffic can freely flow upstream to the ouside but any unsolicited traffic attempting to flow downstream will be blocked.

So, take any retired router and place it upstream of your primary network WiFi router. To do this, plug the IoT WiFi access point (which could also be that upstream router) into your WAN Internet conenction — for example your cable modem. Then plug your primary network's secure router into that router. Your primary network's secure router functions as the one-way valve to protect your internal network — not only from the Internet but no also from the uncertainty of your own IoT network.

Yes... it's more involved and it'll take some work, but once it's setup it's true peace of mind. And you'll also have a truly isolated guest network, not only for your growing stable of IoT devices, but also for any human guests who might come by.