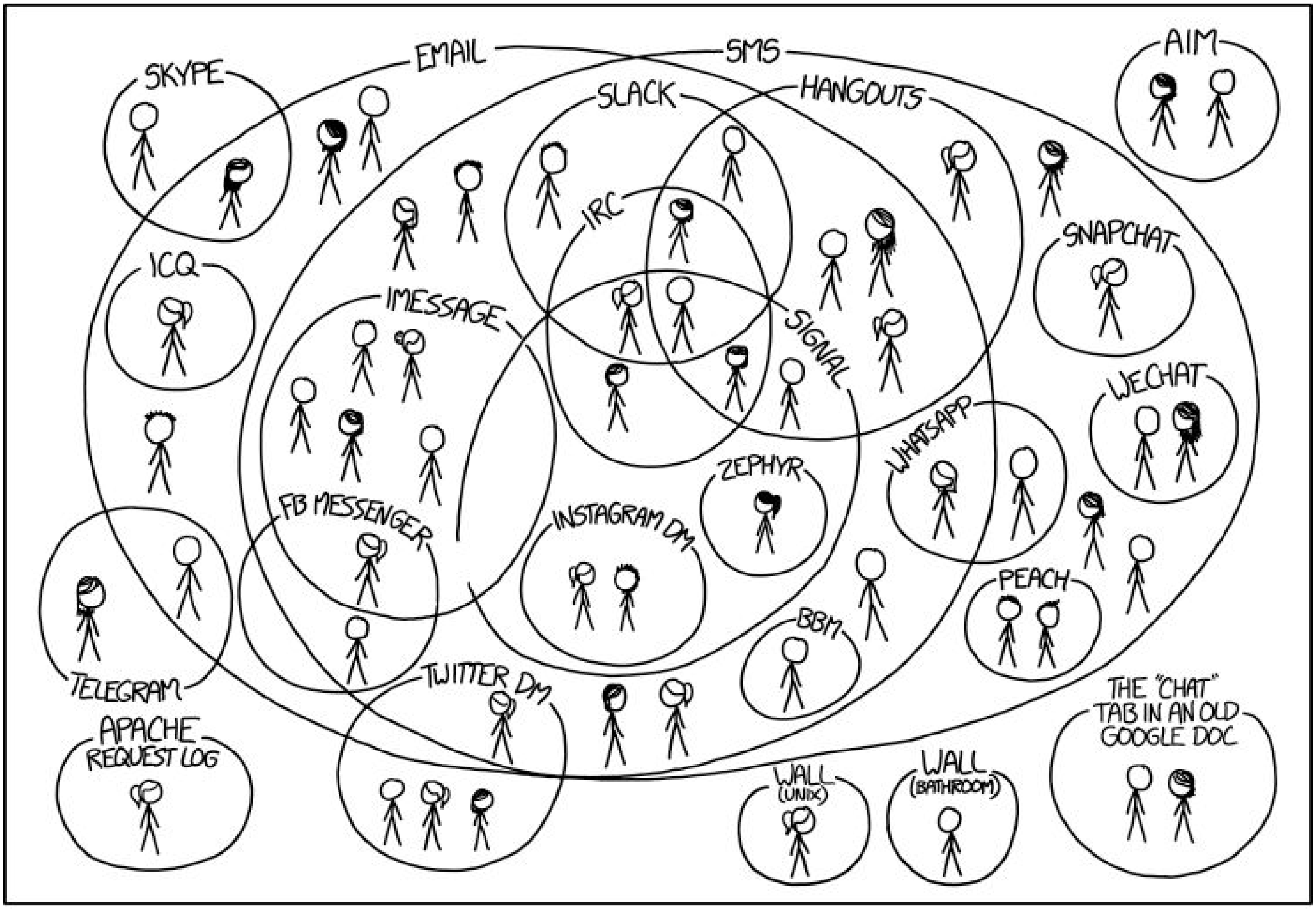
Security Now! #805 - 02-09-21

SCADA Scandal

This week on Security Now!

This week we begin with a collection of interesting and engaging news surrounding Google's Chrome browser. We look at a high-profile Windows Defender misfire, and at new WordPress plug-in nightmares. We check-in on the world of DDoS attacks and cover the meaning of three new critical vulnerabilities in SolarWinds' software. We have a bit of closing the loop feedback from our listeners, an update on my work toward the next SpinRite, and then we look at a near-miss disaster in a poorly designed industrial control system.



I HAVE A HARD TIME KEEPING TRACK OF WHICH CONTACTS USE WHICH CHAT SYSTEMS.

Browser News

Google has been busy with Chrome

A trio of interesting security affected Chrome last week.

First off, at the end of the week, saying only that “The extension contains malware”, Google unceremoniously removed an extremely popular Chrome extension — “The Great Suspender” — from the Chrome Web Store repository and caused all two million plus Chrome browsers, where it had previously been installed, to immediately remove it from their installations. The Great Suspender was very popular with those wishing to run Chrome on memory-lean machines. Chrome Tabs are known to consume a great deal of RAM, and The Great Suspender's claim to fame was that it could suspend tabs and release their memory. Thus allowing its users to retain the tab reference without the RAM cost of keeping the tab's instance resident.

However, back last November, things began to turn sour in Great Suspender land. A November 3rd posting on Github, which started a thread of 449 comments, started with the TR;DR of:

> TLDR: The old maintainer appears to have sold the extension to parties unknown, who have malicious intent to exploit the users of this extension in advertising fraud, tracking, and more. In v7.1.8 of the extension (published to the web store but NOT to GitHub), arbitrary code was executed from a remote server, which appeared to be used to commit a variety of tracking and fraud actions. After Microsoft removed it from Edge for malware, v7.1.9 was created without this code: that has been the code running since November, and it does not appear to load the compromised script. The malicious maintainer remains in control, however, and can introduce an update at any time. Well, they could until Google nuked the extension from their store.

The more detailed discussion was interesting, I thought, so I've excerpted some of the best bits from it:

> @deanoemcke, the original developer, chose to step back from the extension in June 2020. As a replacement maintainer, he chose an unknown entity, who controls the single-purpose @greatsuspender Github account. Much was suspicious about this change, including mention of payment for an open-source extension, and complete lack of information on the new maintainers identity. However, as the new maintainer did nothing for several months, it was believed that there was simply a failed transfer. In October 2020, the maintainer updated [the] chrome store package. The update raised red flags for some users, because the changelog was not modified and there was no tag created in GitHub. On investigation, it appeared that the extension was now connecting to various third-party servers, and executing code from them.
>
> This led a few users to panic, however, on closer investigation, it appeared that the third-party servers were part of an alternative to Google Analytics: and the changes shipped along with a new, though unexplained, tracking deactivation. It appeared that deactivation works. We would later discover that this was wrong.
>
> The discussion continued, however, because the new update also requested additional permissions, including the ability to manipulate all web requests. That lets the extension do

> what it pleases, including inserting ads, blocking sites, forcible redirects.... This change was supposedly in order to enable new screenshot functionality, but that was unclear, and probably shouldn't be needed.
>
> Furthermore, the web store extension has diverged from its GitHub source. A minor change in the manifest was now being shipped on the chrome web store, which was not included in GitHub. This is a major concern: though again, it has a possible innocent explanation. While some think it is illegal given the license on the code, this may not be a GPL violation.. Because the minified script is not part of the extension, the license does not apply to it. Because of Web Store rules, the extension itself can be unpacked and inspected in full, human-readable form, likely satisfying the copyleft restrictions.
>
> As a final red flag, no part of the web store posting has been updated to account for this. @deanoemcke remains listed as the maintainer, and the privacy policy makes no mention of the new tracking or maintainer. It has been several months since the transfer, but almost nothing reflects that change.
>
> @deanoemcke did respond to the thread, after a significant delay. He confirmed much of what is above, including that the secret changes are limited to analytics and are disabled by the flag. However, he hasn't yet clarified what his relationship or basis of trust with the new maintainer is, nor has he explained why the initial post mentions a 'purchase'.
>
> On November 6th, @lucasdf discovered a smoking gun that the new maintainer is malicious. Although OpenWebAnalytics is legitimate software, it does not provide the files executed by the extension. Those are hosted on the unrelated site http://owebanalytics.com/, which turns out to be immensely suspicious. That site was created at the same time as the update, and is clearly designed to appear innocent, being hosted on a public web host, and being given a seemingly innocent homepage from the CentOS project. However, the site contains no real information other than the tracking scripts, appears to have been purchased with BitCoin, and is only found in the context of this extension. Most importantly, the minified javascript differs significantly from that distributed by the OWA project.

I'll finish quoting this writer by observing that he's being extremely kind in his description of the clearly bogus CentOS page. I went over to http://owebanalytics.com/ to take a look around. First of all, it's http:// and not https://. So I thought "Ah! I should probably switch that to https:// to see what it's certificate looks like."

The certificate has a 90-day life, so we wouldn't be surprised to learn that it was signed by Let's Encrypt. But what's weird is that the Common Name of the certificate is cdn.owebanalytics.com. So the certificate is invalid for the domain that's serving it. When I saw that I thought that perhaps the article was wrong and the actual domain was https://cdn.owebanalytics.com. But, no. There's no one home at that domain.

https://github.com/greatsuspender/thegreatsuspender/issues/1263

There's a lot more for anyone who's interested. I have a link in the show notes. It's easy to imagine that some party with less than completely charitable intentions might offer someone who originally developed and has since been thanklessly maintaining a browser extension that's been steadily growing in popularity through the years, some cash to buy them out. And how that might be an appealing opportunity after many years of tireless effort. The seller, who likely still feels some responsibility, hopes that it's all going to work out and wouldn't want to disparage the project's new owner. Yet, there's probably some reason why control of a well-regarded and highly-used open source extension was worth some money to its purchaser. And, indeed, there were some activities discussed back in November that appeared to provide some hint of what was to come.

We don't know what finally happened to trip an alarm to cause Google to yank the extension from the Chrome Web Store. But the writing was certainly on the wall. And we've talked before about web browser extensions being allowed to have the power to filter and modify all web content coming to and from the browser. It presents a sobering danger.

The GitHub thread did note that:

> The Great Suspender has been removed from the Chrome Web Store. To recover your tabs, see issue #526. The code in the github repository is currently safe. The most recent tagged release happened before the transfer of ownership. To use that version, and avoid needing to finagle URL's, enable Chrome developer mode, download and extract a copy of the code, then navigate to your extensions menu and select 'Load Unpacked Extension'.

Let's hope that whoever purchased the extension lost money and that they and others will be disincentivized from attempting to purchase and subvert other browser extensions. What we’ve just witnessed is a worrisome reality of our current web browser ecosystem. I know that I and many of our listeners rely heavily upon extensions for both Chrome and Firefox and I would never want to have to use browsers without any.

The second thing that happened…
was that on Thursday, Tenable and Microsoft both provided information about the otherwise under-mentioned update to Chrome that occurred the same day. Tenable's posting explains what they know and I’ll extract some bits from what they wrote:

https://www.tenable.com/blog/cve-2021-21148-google-chrome-heap-buffer-overflow-vulnerability-exploited-in-the-wild

On February 4, Google published a stable channel update for Chrome for Desktop. This release contained a single security fix to address a critical zero-day vulnerability that had been exploited in the wild. The vulnerability is a heap buffer overflow vulnerability in Chrome's V8 engine whose discovery is credited to Mattias Buelens, who reported the flaw to Google on January 24.

Google noted that they are “aware of reports that an exploit” for this vulnerability “exists in the wild,” which we interpret to mean that in-the-wild exploitation attempts have been observed. Google's bug report for the vulnerability is unsurprisingly restricted to allow users time to apply the relevant patch.

In a bit of interesting timing, this flaw was disclosed to Google just one day before a significant revelation from Google. On January 25, Google's Threat Analysis Group (TAG) published a blog post detailing the discovery of an ongoing campaign conducted by nation-state actors (believed to be North Korea) which is targeting security researchers interested in collaborating on vulnerability research. The report specifically mentions that the threat actors circulated a link to their potential victims to a malicious website that led to successful exploitation on systems that were fully patched for both Windows and Google Chrome. This was corroborated by Microsoft, which published their own blog post about the attacks, surmising that a Google Chrome zero-day was likely used to target researchers.

https://blog.google/threat-analysis-group/new-campaign-targeting-security-researchers/

What Microsoft discovered and shares is amazing and kind of horrifying...

> Over the past several months, the Threat Analysis Group has identified an ongoing campaign targeting security researchers working on vulnerability research and development at different companies and organizations. The actors behind this campaign, which we attribute to a government-backed entity based in North Korea, have employed a number of means to target researchers which we will outline below. We hope this post will remind those in the security research community that they are targets to government-backed attackers and should remain vigilant when engaging with individuals they have not previously interacted with.
>
> In order to build credibility and connect with security researchers, the actors established a research blog and multiple Twitter profiles to interact with potential targets. They've used these Twitter profiles for posting links to their blog, posting videos of their claimed exploits and for amplifying and retweeting posts from other accounts that they control.
>
> Their blog contains write-ups and analysis of vulnerabilities that have been publicly disclosed, including "guest" posts from unwitting legitimate security researchers, likely in an attempt to build additional credibility with other security researchers.
>
> While we are unable to verify the authenticity or the working status of all of the exploits that they have posted videos of, in at least one case, the actors have faked the success of their claimed working exploit. On Jan 14, 2021, the actors shared via Twitter a YouTube video they uploaded that proclaimed to exploit CVE-2021-1647, a recently patched Windows Defender vulnerability. In the video, they purported to show a successful working exploit that spawns a cmd.exe shell, but a careful review of the video shows the exploit is fake. Multiple comments on YouTube identified that the video was faked and that there was not a working exploit demonstrated. After these comments were made, the actors used a second Twitter account (that they control) to retweet the original post and claim that it was "not a fake video."
>
> The actors have been observed targeting specific security researchers by a novel social engineering method. After establishing initial communications, the actors would ask the targeted researcher if they wanted to collaborate on vulnerability research together, and then provide the researcher with a Visual Studio Project. Within the Visual Studio Project would be source code for exploiting the vulnerability, as well as an additional DLL that would be executed through Visual Studio Build Events. The DLL is custom malware that would immediately begin communicating with actor-controlled C2 domains. An example of the VS Build Event can be seen in the image below.
>
> In addition to targeting users via social engineering, we have also observed several cases where researchers have been compromised after visiting the actors' blog. In each of these

> cases, the researchers have followed a link on Twitter to a write-up hosted on blog.br0vvnn[.]io, and shortly thereafter, a malicious service was installed on the researcher's system and an in-memory backdoor would begin beaconing to an actor-owned command and control server. At the time of these visits, the victim systems were running fully patched and up-to-date Windows 10 and Chrome browser versions. At this time we're unable to confirm the mechanism of compromise, but we welcome any information others might have. Chrome vulnerabilities, including those being exploited in the wild (ITW), are eligible for reward payout under Chrome's Vulnerability Reward Program. We encourage anyone who discovers a Chrome vulnerability to report that activity via the Chrome VRP submission process.
>
> [And note that it is now believed that it was this, just patched, Chrome 0-day that was being used to compromise the system's of trusting security research collaborators.]
>
> These actors have used multiple platforms to communicate with potential targets, including Twitter, LinkedIn, Telegram, Discord, Keybase and email. We are providing a list of known accounts and aliases below. If you have communicated with any of these accounts or visited the actors' blog, we suggest you review your systems for the IOCs provided below. To date, we have only seen these actors targeting Windows systems as a part of this campaign.
>
> If you are concerned that you are being targeted, we recommend that you compartmentalize your research activities using separate physical or virtual machines for general web browsing, interacting with others in the research community, accepting files from third parties and your own security research.

The third Chrome story that every security-related outlet carried last week was the discovery of a unique use of Chrome's "sync" feature for command & control and data exfiltration.

Last Thursday, a Croatian security researcher (who I'm certain would appreciate being referred to by his initials "BZ" rather than have me attempt to pronounce his name), said last Thursday that during a recent incident response, he discovered that a malicious Chrome extension was abusing the Chrome sync feature as a way to communicate with a remote command and control (C&C) server and as a way to exfiltrate data from infected browsers.

As we know, multiple Chrome web browsers which are logged into the same Google account will automatically share and synchronize a set of configuration settings, tabs, favorites, extensions, browser history and so forth. Each browser connects to the Google Mothership to check-in, and Google hands out updates, as needed, which may have come in from other Chrome browsers logged into the same account.

What's diabolically clever about this is that this communication, encrypted under Google's own security certificates, would typically go completely unnoticed by anyone. It would slip right through any corporate firewalls. Data could be encoded into long Base64-encoded URL tails, and Google would simply send them out to other browsers on the same account.

"BZ" said that in the incident he investigated, attackers gained access to a victim's computer, but because the data they wanted to steal was inside an employee's portal, they downloaded a Chrome extension on the user's computer and loaded it via the browser's Developer Mode.

The extension, which posed as a security add-on from security firm Forcepoint, contained malicious code that abused the Chrome sync feature as a way to allow attackers to control the infected browser. This way, the extension could be used as an exfiltration channel from inside corporate networks to an attacker's remotely located Chrome browser instance, or as a way to control the infected browser from afar, bypassing local security defenses.

“BZ” explained that blocking access to the Chrome sync server at client4.google.com would not work because that domain is used for many other things such as by Chrome to detect an Internet connection. Instead of doing that, BZ urges companies to use Chrome's enterprise features and group policy support to block and control what extensions can be installed in the browser to prevent the installation of rogue extensions like the one he investigated.

Security News

Defender thinks Chrome is Malware

With the subhead of “no good deed goes unpunished” ... No sooner had Google quickly updated Chrome to remove the 0-day flaw in its V8 engine that was being actively exploited in the wild to, among other things, attack security researchers... than Microsoft's enterprise version of Windows Defender decided that Google's modifications were malicious.

Yes... Microsoft Defender Advanced Threat Protection (ATP), which is the commercial version of the ubiquitous Defender AV and also Microsoft's premiere enterprise security solution, was having a bad day and labeled Google’s recent Chrome browser update a backdoor Trojan.

Based on Twitter reports posted by dismayed sys admins, Defender ATP is currently detecting multiple files which are part of last week's Chrome v88.0.4324.146 as containing a generic backdoor trojan named "PHP/Funvalget.A."

Though this might have normally been met with somewhat more calm, in this case Defender's alerts raised some alarm and quite a stir in enterprise environments due to the recent multiple software supply chain attacks that we're all quite aware of recently. So, sys admins were awaiting a formal statement from Microsoft to confirm that the detection is indeed a "false positive" and nothing to worry about.

Fortunately, the built-in no-charge version of Microsoft Defender AV that we all have in our personal Windows 7, 8 and 10 systems was **not** suspicious of this new release of Chrome... which is fortunate or it would have been a far more widespread mess. But that does make one wonder about the detection differential between Microsoft’s commercial and consumer AV’s? Why, exactly, did the enterprise AV freak out whereas the consumer AVs remained quiet?

Microsoft **did** later confirm that the “Funvalget” detections for those Chrome files were indeed a false positive due to what Microsoft termed “an automation error” — whatever the heck that means. I suppose they needed to call it something, and calling it what it really was, a “false positive” probably can’t get past the P.R. folks.

I should mention that over for the past few years I've had to exclude Defender from poking into many of my own development directories. Unless I do that, shortly after I build a new executable from source, Defender will slide up a red warning saying “Not to worry! It's all good! Defender has found and removed the threat that just appeared.” Of course, that's my own code, just freshly built from source — it couldn't be any cleaner. And recently they've all been DOS executables. I noted that if some malicious Trojan were to find itself running in DOS (which actually it probably couldn't even do) it would not be happy.

What must be upsetting Defender is the lack of digital signatures on my freshly built EXE's. We’ve seen through experimental evidence that Defender places a LOT of weight upon the reputation of the certificates which sign today's executables. It's gotten to the point where it's difficult to keep Defender from complaining when any executable is **not** signed.

Today, Windows refuses to load any unsigned device drivers into its kernel. Developers can force the issue by disabling Windows' driver signing enforcement. But I'll bet that we're not far from the day when Windows will be elevating that requirement to the desktop. There might be something like a UAE dialog that users are forced to push past whenever they wish to run any unsigned executable, or an executable signed by a certificate that hasn't yet established a spotless reputation for itself.

More Critical WordPress Plug-in Problems

In this case we have more than 800,000 WordPress sites vulnerable to several critical and high-severity cross-site request forgery (CSRF) flaws which are present in the popular “NextGen Gallery” WordPress plug-in.

NextGen Gallery allows sites to accept uploads of photos in batch quantities, import metadata and edit image thumbnails. But researchers discovered two CSRF flaws – one critical and one high-severity – in the plugin. A patch was released to address the flaws in version 3.5.0, in the middle of last December. And the researchers responsibly waited until yesterday before publicly disclosing details of the flaw.

https://www.wordfence.com/blog/2021/02/severe-vulnerabilities-patched-in-nextgen-gallery-affect-over-800000-wordpress-sites

WordFence's Ram Gall wrote in their disclosure of the vulnerabilities, yesterday of “a critical severity vulnerability that could lead to Remote Code Execution(RCE) and Stored Cross-Site Scripting(XSS). Exploitation of these vulnerabilities could lead to a site takeover, malicious redirects, spam injection, phishing, and much more.” He said:

> We initially reached out to the plugin’s publisher, Imagely, the same day, and provided full disclosure the next day, on December 15, 2020. Imagely sent us patches for review on December 16, and published the patched version, 3.5.0, on December 17, 2020.
>
> Wordfence Premium users received firewall rules protecting against these vulnerabilities on December 14, 2020. Sites still running the free version of Wordfence received these rules 30 days later, on January 13, 2021.
> NextGEN Gallery is a popular WordPress plugin designed to create highly responsive image galleries. It is clear the plugin’s developer took care to integrate security in the code of the

> plugin. NextGen Gallery has a single security function, is_authorized_request, that is used to protect most of its settings:
>
> This function integrated both a capability check and a nonce check into a single function for easier application throughout the plugin. Unfortunately, a logic flaw in the is_authorized_request function meant that the nonce check would allow requests to proceed if the $_REQUEST[‘nonce’] parameter was missing, rather than invalid.
>
> This opened up a number of opportunities for attackers to exploit via Cross-Site Request Forgery. One feature of NextGen Gallery is the ability for administrators to upload custom CSS files to be used to style galleries. While the file uploaded had to end with the .css extension, it was possible to upload arbitrary code with double extensions, (e.g., file.php.css). While these files would only be executable on certain configurations, such as Apache/mod_php with an AddHandler directive, this could still result in remote code execution on any vulnerable configurations.
>
> Unfortunately, it was also possible to achieve code execution even on configurations not vulnerable to double extensions. NextGen Gallery has a separate feature that allows users to specify how galleries are viewed via a “Legacy Templates” feature, which also uses the is_authorized_request function for security. Thus, it was possible to set various album types to use a template with the absolute path of the file uploaded in the previous step, or perform a directory traversal attack using the relative path of the uploaded file, regardless of that file’s extension, through a CSRF attack.
>
> This would result in Local File Inclusion (LFI) and Remote code Execution (RCE), as the uploaded file would then be included and executed whenever the selected album type was viewed on the site. Any JavaScript included in the uploaded file would also be executed, resulting in Cross-Site Scripting (XSS).
>
> As a reminder, once an attacker achieves Remote Code Execution on a website, they have effectively taken over that site. XSS can likewise be used to take over a site if a logged-in administrator visits a page running a malicious injected script.
>
> This attack would likely require some degree of social engineering, as an attacker would have to trick an administrator into clicking a link that submitted crafted requests to perform these actions. Additionally, performing these actions would require 2 separate requests, though this would be trivial to implement and we were able to do so during our testing. Finally, the site would require at least one album to be published and accessible to the attacker.

I'm impressed by these WordFence people. We've run across them several times in the past year as WordPress problems have been receiving increased scrutiny, from both attackers and security researchers. I'm glad I am no longer running WordPress on any of my own servers. But I understand that others may have little choice. As was noted above, WordFence offers both a free and paid version of their WordPress firewall. If I had to be running WordPress, I would give WordFence a serious look. They really do appear to be on the ball and quite worthwhile.
Since its these plug-ins that appear to be causing all the trouble, if you’re running WordPress lean, with no plug-ins, then I think you could safely skip it, or definitely go with the free

WordFence. But if you can't resist enhancing the base WordPress with plug-ins galore, then I'd definitely add one more — specifically, that WordFence protection.

Checking in on DDoS attacks

Last Thursday, NetScout posted a notice about the abuse of Internet-exposed Plex Media servers SSDP protocol being used in reflected and amplified DDoS attacks.

https://www.netscout.com/blog/asert/plex-media-ssdp-pmssdp-reflectionamplification-ddos-attack

We haven't talked about DDoS attacks for awhile, so thought that the numbers being cited here by NetScout were interesting. Here's what they wrote:

> Plex Media Server is a personal media library and streaming system which runs on modern Windows, macOS, and Linux operating systems, along with variants customized for special-purpose platforms such as network-attached storage (NAS) devices, external RAID storage units, digital media players, etc.
>
> Upon startup, Plex probes the local network using the G'Day Mate (GDM) network/service discovery protocol to locate other compatible media devices and streaming clients. It also appears to make use of SSDP probes to locate UPnP gateways on broadband Internet access routers which have SSDP enabled; when a UPnP gateway is discovered via this methodology, Plex attempts to utilize NAT-PMP to instantiate dynamic NAT forwarding rules on the broadband Internet access router.
>
> On January 7, 2021 Baidu Labs, in a Chinese-language weblog post, described a UDP reflection/amplification DDoS attack vector leveraging Plex Media Server instances running versions of the Plex software prior to 1.21. In early February 2021, NETSCOUT Arbor were notified that reflection/amplification DDoS attacks which appeared to leverage abusable Plex Media Server instances were actively taking place on the public Internet.
>
> According to an announcement published on Plex's Web site on February 5, 2020 Plex Media Server instances which have either been deployed on a public-facing network DMZ or in an Internet Data Center (IDC), or with manually configured port-forwarding rules which forward specific UDP ports from the public Internet to devices running Plex Media Server, can potentially be abused as part of possible DDoS attacks.
>
> These actions can have the effect of exposing a Plex UPnP-enabled service registration responder to the general Internet, where it can be abused to generate reflection/amplification DDoS attacks. In order to differentiate this particular attack vector from generic SSDP reflection/amplification, it has been designated as Plex Media SSDP (PMSSDP) reflection/amplification. To date, approximately 37,000 abusable PMSSDP reflectors/amplifiers have been identified on the public internet.
>
> Amplified PMSSDP DDoS attack traffic consists of SSDP HTTP/U responses sourced from ports UDP port 32414 and/or UDP port 32410 on abusable Plex Media Server instances and directed

towards attack target(s); each amplified response packet ranges from 52 bytes – 281 bytes in size, for an average amplification factor of ~4.68:1.

Observed single-vector PMSSDP reflection/amplification DDoS attacks range in size from ~2 Gbps – ~3 Gbps; multi-vector (2–10 vectors) and omni-vector (11 or more vectors) attacks incorporating PMSSDP range from the low tens of Gbps up to 218 Gbps. As is routinely the case with newer DDoS attack vectors, it appears that after an initial period of employment by advanced attackers with access to bespoke DDoS attack infrastructure, PMSSDP has been weaponized and added to the arsenals of so-called booter/stresser DDoS-for-hire services, placing it within the reach of the general attacker population.

To date, more than 5,500 PMSSDP reflection/amplification DDoS attacks have been observed on the public Internet, leveraging approximately 15,000 distinct abusable PMSSDP reflectors/amplifiers.

It should be noted that a single-vector PMSSDP reflection/amplification attack of ~2 Gbps – ~3 Gbps in size is often sufficient to have a significant negative impact on the availability of targeted networks/servers/services. The incidence of both single-vector and multi-/omni-vector reflection/amplification attacks leveraging PMSSDP has increased significantly since November of 2020, indicating its perceived utility to attackers.

And just how prevalent have DDoS attacks become these days? To find out, BleepingComputer opened an eMail dialog with Richard Hummel, NetScout's Manager of Threat Intelligence. Richard wrote that “The total number of [Plex Media SSDP] attacks from Jan 1, 2020, to present day, clocked in at approximately 5,700 — compared to the more than 11 million attacks in total we saw during the same time frame.”

Three more NEW vulnerabilities have been discovered in SolarWinds’ software

I hope that this podcasts' listeners are aware of the extremely disturbing fact that we keep encountering instances of what I'll term the principle of “wherever we look we discover new problems.” Today, problems are being discovered in two ways:

First, the old fashioned way, where we discover malware in some system, then reverse engineer the malware to discover how it got in. And we looked, last week, at the extreme measures the SolarWinds hackers went to in order to avoid exactly that form of reverse engineering.

The new modern way of finding vulnerabilities is, apparently, simply by looking closely at pretty anything and discovering that “Oh look! ... it's full of security weaknesses! Who knew?!” It's this new second reality that has turned vulnerability discovery into a potential career! Just find some company that has the wisdom to offer bounties for the discovery of their bugs, then take a close look at their code... and before long you can probably use cash to buy yourself a new car.

Last week, another case illustrating this disturbing truth just came to light thanks to the many researchers who have started looking more closely — for the first time ever — at the code being shipped by SolarWinds. TrustWave's most recent SpiderLabs' Blog, posted last Wednesday, was titled: “Full System Control with New SolarWinds Orion-based and Serv-U FTP Vulnerabilities” https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/full-system-control-with-new-solarwinds-orion-based-and-serv-u-ftp-vulnerabilities/

Martin Rakhmanov posted in the first person:

> “In this blog, I will be discussing three new security issues that I recently found in several SolarWinds products. All three are severe bugs with the most critical one allowing remote code execution with high privileges. To the best of Trustwave’s knowledge, none of the vulnerabilities were exploited during the recent SolarWinds attacks or in any “in the wild” attacks. However, given the criticality of these issues, we recommend that affected users patch as soon as possible. We have purposely left out specific Proof of Concept (PoC) code in this post in order to give SolarWinds users a longer margin to patch but we will post an update to this blog that includes the PoC code on Feb. 9.”

That was last Wednesday and, true to his plan, today is the 9th and proof of concept code has indeed been published. There are now public PoC's for the following disturbing three new vulnerabilities:

- SolarWinds Orion Platform (CVE-2021-25274): Improper use of Microsoft Messaging Queue (MSMQ) could allow any remote unprivileged user the ability to execute any arbitrary code in the highest privilege.

- SolarWinds Orion Platform (CVE-2021-25275): SolarWinds credentials are stored in an insecure manner that could allow any local users, despite privileges, to take complete control over the SOLARWINDS_ORION database. From here, one can steal information or add a new admin-level user to be used inside SolarWinds Orion products.

- SolarWinds Serv-U FTP for Windows (CVE-2021-25276): Any local user, regardless of privilege, can create a file that can define a new Serv-U FTP admin account with full access to the C:\ drive. This account can then be used to log in via FTP and read or replace any file on the drive.

It's not my intention to single out SolarWinds. Yes, they're currently in the hot seat. But we would be wrong to assume that we just happen to be finding all manner of serious problems with the only company whose offerings have been closely scrutinized. The only sane assumption would be that the software published and in wide use by many other similar entities would crumble just as quickly if and when it were to be subjected to a similar level of close expert scrutiny. The entire industry just assumed that SolarWinds' was sufficiently good and careful about network security, which is what they were selling, after all. It probably said that somewhere on their website, next to their customer list, which has since been taken down.

Because such close expert scrutiny is expensive, no one is doing that to most of the industry's software. We now believe that highly skilled and talented Russian hackers were caught having closely scrutinized SolarWinds’ systems and code. But the evidence begs the question: What other software company's work has this group also examined closely? And what did they find?

Closing the Loop

ndom91 / @ndom91
@SGgrc Steve, I love you, your work, and the show, but for the love of God - it's called "lib", like "libertine" and not "laib" like "library" ?? (referring to the "libgcrypt" segment in SN804) #securitynow #twit

David Stricker / @strickdd
You seem to be sending mixed signals on this week's SN. In the past you've indicated that everything should have an auto-updating mechanism. This week it sounds like you don't trust them. Is it just Notepad++ you don't trust or is it because it prompts to update where as Chrome does it in the background?

AndyMan7 / @Man7Andy
Hello Steve, big fan of Security Now. I recall you mentioning a remote desktop management tool on one of the shows but can't remember the name. Would you help remind me what that was and if you have any recommendations for a safer tool for IT people to do remote sessions to PCs?

[The tool is simply called "Remote Utilities" (https://www.remoteutilities.com/) and I continue to be SO impressed with it. Lorrie uses it constantly, literally continually, daily, to manage the array of remote laptops being used by her clients who are doing at-home neurofeedback training. My tech support guy, Greg, who has a computer repair consulting business on the side has completely switched over to using it and has hundreds of machines under remote management. And I am increasingly using the system to manage several of my own machines. It is an absolute win.

The other thing that I love about it is that it is purchased ONCE and it's not a subscription. And they offer a free license: "Our free license allows you to add up to 10 remote computers in your Viewer address book. You can use the free license in a business and personal setting. Only one free license key is allowed per individual, company or organization. For more information, please see our EULA."

So, I'm glad you asked, Andy, since these guys really deserve a look!]

SpinRite

"Discovering System's Mass Storage Devices..."

SCADA Scandal

So now we turn our attention to the small city of Oldsmar, Florida home to approximately 15,000 residents. Oldsmar lies about 16 miles northwest of the much more widely know city of Tampa.

The first signs that anything might be amiss at Oldmar's municipal water treatment plant appeared last Friday morning, when a plant operator noticed someone had remotely accessed a system that controls chemicals and other aspects of the water treatment process. The operator reportedly didn't think much of the event since his supervisor and co-workers regularly logged into the remote system to monitor operations.

But then later that same day, around 1:30 in the afternoon, the operator watched as someone remotely accessed the system again. He could see the mouse on his screen being moved to open various functions that controlled the water treatment process. This unknown person then opened the function that controls the input of sodium hydroxide — popularly known as lye — increasing it by 111-fold. The intruder increased the level of sodium hydroxide to 11,100 parts per million from the normal proper level of 100 ppm.

Lye, or sodium hydroxide is used in very small amounts to treat the acidity of water and to remove metals. It's also the active ingredient in liquid drain cleaners. And, yes, in higher levels — such as 111 times normal, it's highly toxic. Had the change not been reversed almost immediately, it would have raised the amount of chemical to toxic levels.

[I'll pause our story here to wonder why it's even possible to adjust, through any automation, the amount of lye to a level that's 111 times normal. That seems like a fundamental oversight in the design of the system. Sure, perhaps allow a range of 0% to 200%. But certainly not up to 11,100%.]

Fortunately, the operator immediately changed the setting back to the normal 100 ppm. And supposedly, even if the malicious change had not been reversed, other routine procedures in the plant would have caught the dangerous level before the water became available to residents. It apparently takes 24 to 36 hours for treated water to hit the supply system. So in this case no poisoned water ever escaped.

The local county's Sheriff's Department did not immediately respond to a question asking whether the utility required personnel to use two-factor authentication to gain remote access to interfaces such as the one that was breached in Oldmar. The Reuters new agency, citing an interview with managers, reported that Teamviewer was the application used to gain remote access, but the department didn't immediately respond about the requirements for authentication.

Jake Brodsky, an engineer with 31 years experience working in the water industry, said it's not at all uncommon for water utilities to make such interfaces available remotely. While he frowns on the practice, he said that the managers were probably correct in stating that the public was never in any danger.

In an interview, Brodsky said: “There’s a bunch of different things [water utilities] look for, and if they see anything out of kilter, they can then isolate the storage water. The danger here is relatively minimal as long as you catch it soon enough and there are multiple checks before that happens.”

Of course, if intruders can remotely tamper with a process, they may also be able to tamper with the safety redundancies in place. If Brodsky were advising Oldsmar officials on better securing their water treatment plant, “the first thing I’d probably do, and this almost doesn’t cost anything, is you disable the remote access,” he said. When remote access is required, as occasionally is the case, connections should be manually allowed by someone physically present and the access should time out after a brief period of time.

“I can’t imagine leaving a connection like that open and exposed to the world,” Brodsky said. “This is cheap and easy. All you do is call the operator and you get the access.”

Wow. There has been a **LOT** of talk — and no obvious action — through the years about the vulnerability of SCADA systems. SCADA being the abbreviation for “Supervisory Control And Data Acquisition.” We probably really are incredibly vulnerable in this country. There are just too many instances where convenience has dictated policy.

I sincerely hope that all managers who are responsible for the operational safety of their industrial plants, of every description, hear about what happened in Oldsmar and take it to heart. With any luck, it will have been a wakeup call.