Security Now! #798 - 12-29-20

Sunburst & Supernova

This week on Security Now!

This week, as we end 2020, we look at Chrome's backing away from a security initiative, Firefox's move to further thwart tracking, all of the browsers once again saying "No!" to Kazakhstan, the formation of a new industry-wide Ransomware Task Force, this week's widespread WordPress security disaster, the return of Treck's insecure embedded TCP/IP stack, and yes... finally, the long awaited announcement of the release of the ReadSpeed benchmark which serves as a testbed and proof-of-operation for the next generation of SpinRite. And then we look at everything more that has come to light three weeks downstream from the first revelations of the SolarWinds-based massively widespread network intrusion and compromise.

"It's free, but they sell your information."

CARTOONSTOCK.COM
CARTOONCOLLECTIONS.COM

Browser News

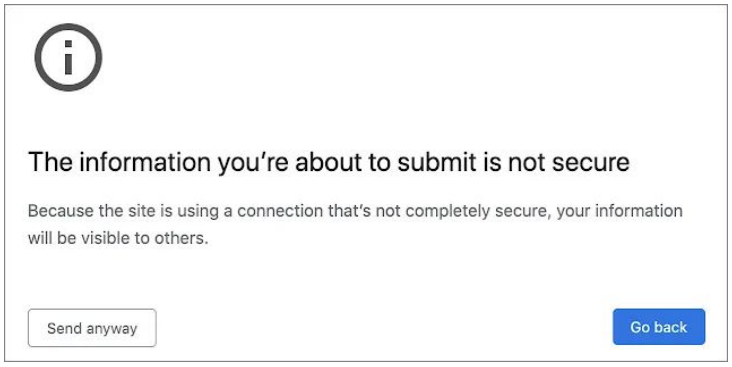
Chrome 87 quickly backs away from Insecure Form Warnings

We've had some fun from time to time at Google's expense with the way they tend to roll out new features that might have an impact on some of their visitors. The most recent of these was the very gradual deprecation of the browser's FTP services. But their very cautious approach is easier to understand in the wake of a change that was just made in the most recent and still current release 87 of Chrome which they then needed to immediately roll back as a result of a surprising hue and cry from their users.

The change was one that we covered before it happened: The enforcement of secure form submission with a warning if the form's target URL is not to a HTTPS url. When we talked about this coming to Chrome 87, I commented that the announcement had surprised me because I had assumed that everyone had long ago been enforcing secure form submissions. It was years ago that we originally noted that even non-secure pages could be submitting their form data securely since it wasn't the URL of the submitting page but rather the URL to which the form was sending its data as a query. Now, while that was true, it was still unadvisable to submit anything from an HTTP page, since being HTTP, anyone being able to establish a man-in-the-middle position would be able to edit the pages contents to, for example, have the page send the form's data elsewhere. But in any event, with the recent release of Chrome 87, Google finally joined the club of browsers that are enforcing secure form submission. So what went wrong? Google apparently implemented their solution differently than everyone else's, since it began producing scary warnings shortly after it went live.

The trouble arose because, as we know, what web pages do it generate queries. They can be GET or POST queries, and in the case of a form submission, the POST's query body contains the data being submitted. And since it's a query, the server then responds. A typical response might be to return a page saying "Welcome back Joey! You have successfully logged on!" ...or whatever. But a more complex reply, in the form of a redirect of the browser to another server URL domain is just as valid. And it turns out that this is also somewhat common.

What Google learned the hard way, was that some of these form-submission redirects are to non-secure HTTP URLs. This doesn't represent any threat to the user's submitted data, since the original form submission URL will be to an HTTPS URL. So the form's data will be transmitted with the encryption privacy and certificate authentication provided by HTTPS. And wherever the receiving server wants to then bounce its user is really up to it. But Chrome was being faithful to Google's "HTTPS or die" philosophy. So it freaked out when, even after the form had been submitted, the user's browser was **then** bounced to a non-secure HTTP page. The user would receive a warning that was designed to be scary:



Of course, the problem was, by that point the data had already been submitted securely and the only thing this message was doing was breaking a redirect chain that needed to be followed for the user's experience to be correct.

Upon the release of Chrome 87, Google began receiving complaints from web sites that their users were suddenly receiving bogus warnings about insecure form submissions. It didn't take Google long to determine that it wasn't the submission itself that was triggering the error, but rather the post-submission redirect.

So, shortly after Chrome 87's first release, Google's software engineer Carlos Joan Rafael Ibarra Lopez stated that they are disabling the feature in Chrome 87 to adjust it, so HTTP redirects after a secure form submission do not generate a warning. He wrote:

> *"After considering the unexpectedly large impact this change had on form submissions that involve redirects through HTTP sites, we have decided to roll back the change for Chrome 87. We expect the configuration to be out later today, at which point it will take effect on the next Chrome restart. I'll ping this bug with updates.*
>
> *We are planning to re-enable the warnings in Chrome 88 (tentatively going to stable on January 19, 2021), but warning only on forms that directly submit to http://, or that redirect to http:// with the form data preserved through the redirect, so it won't trigger for the cases mentioned in this bug where the http:// hop didn't carry the form data.*
>
> *That being said, I still encourage sites to keep https:// throughout the whole redirect chain, as http:// steps still compromise user privacy (by exposing the form target location) even if no form data is being exposed.*
>
> *Apologies for the issues caused by this new warning."*

So, I suppose the moral of this story is that it's more clear why Google was so careful about the deprecation of "browser as FTP client". It's clear that anything that's changed in the existing ecosystem can — and probably will — have some unexpected consequences.

Firefox to begin partitioning its caches

The original designers of the HTTP protocol understood that having browsers download the same static content over and over from a remote web server would be really dumb. So from its first days, browsers employed local storage caching to, effectively, incrementally and conditionally move some of the remote web server's more static data over to the user's side of the connection. This was managed by a clean and simple mechanism that allowed any web content to indicate how long it was allowed to be cached. Content could specify a "Max-Age" for itself, and also a "Not Valid After" expiration time and data. This would cause cached data to self expire. And if a browser already had something in its cache that a new page was requesting, it could send a new query for that existing content with an "If-Modified-Since" header which would tell the receiving server when the cached content had been received. And the server could then check with its copy of the content to see whether it was newer than the one the browser already had, and either reply to immediately do ahead and use the copy it has, or send updated content.

So, caching is, and has always been, crucial to the performance of our web browsers.

Unfortunately, caching, because it inherently stores evidence of a user's browsing history, is also subject to tracking and privacy abuse. Recall that clever hack where a tracking facility would create a off-canvas link with a URL and then probe for the link's rendered color, knowing that the browser would color previously visited links differently than new links. Thus the tracking site could profile which sites the browser's user had previously visited. If we've seen anything its that the user tracking industry will go to any lengths necessary to track and profile us.

The problem is that browsers have been storing all of their cached content in large pools without regard for the domain which sourced the cached item. And this has led to trackers probing the shared cache pool for content. Td The World Wide Web Consortium's (W3C) response to this is known as "Client-Side Storage Partitioning'' — also known as Network Partitioning — and it's coming to our Firefox browsers with release 85 next month, in January 2021.

According to Mozilla, the following network resources will be partitioned starting with Firefox 85:

HTTP cache, Image cache, Favicon cache, Connection pooling, StyleSheet cache, DNS, HTTP authentication, Alt-Svc, Speculative, Font cache, HSTS, OCSP, Intermediate CA cache, TLS client certificates, TLS session identifiers, Prefetch, Preconnect, CORS-preflight cache.

All of that is client-side state history data that could, unless proactively prevented, be sniffed by 3rd-party domains, advertisements, and 3rd-party JavaScript libraries running in our web pages. And while Mozilla's deployment of this new W3C standard client-side storage partitioning will be the broadest implementation of partitioning so far, it's not the first.

The prize for being the first goes to Apple who began partitioning Safari's HTTP cache seven years ago in 2013. We talked about this happening at the time. And Apple then followed through by partitioning even more client-side data years later as part of its Tracking Prevention campaign.

Google partitioned Chrome's HTTP cache last month, in Chrome 86, and the results were felt immediately as Google Fonts lost some of its performance when fonts could no longer be stored in the shared HTTP cache.

The Mozilla team has indicated that they expect to see similar performance hits when Firefox's cache partitioning is brought online. But they are committed to taking that hit to improve the privacy of Firefox's users. And Mozilla said that one positive side-effect of their deployment of comprehensive client-side partitioning is that Firefox 85 will finally be able to block "supercookies" — those files that abuse shared storage mediums to persist in browsers and allow advertisers to track user movements across the web.

When you stand back to look at all of the time and trouble and complexity that has gone into and continues to go into this fight against privacy-invading tracking — all of which is only allowed to happen because it's well hidden and most users are completely oblivious — it would all be a lot easier if all cross-domain tracking were simply outlawed. But that doesn't appear to be on any legislator's radar.

The browsers once again say no to Kazakhstan
As we have reported some time ago, the government of Kazakhstan has been requiring the citizens to install a "Government of Kazakhstan" CA root certificate into their machines. And we know why. This would allow the government to perform what I would call "no complaints" man-in-the-middle interception of all web traffic of any and all of their citizens. A Kazakhstan proxy would accept the remote connections from remote servers, decrypt, reinspect, and then reencrypt the traffic under the Kazakhstan CA's identity, which would be trusted because the citizen's machine would have the matching Root CA installed.

So, Google, Mozilla, Apple, and Microsoft all together said: "No, you don't." All four of those companies' browsers recently updated to block any and all use of that root certificate.

A thread on Mozilla's bug-reporting site first reported that the certificate was in use just over three weeks ago, on December 6. The "Censored Planet" website later reported that the certificate worked against dozens of Web services that mostly belonged to Google, Facebook, and Twitter.

You gotta give these clowns credit for trying. Remember that back in 2015 the Kazakhstan government formally applied to have their root certificate included in Mozilla's trusted root store program. But once it came to light that they were intending to use the certificate to intercept their citizen's data, Mozilla denied the request. And then, shortly afterwards, the government required its citizens to manually install its certificate, but that attempt failed after organizations took legal action.

And then, August before last in 2019 all browsers blocked a similar attempt. Our listeners may recall the Kazakhstan government's dubious statement back then. Reuters reported that "Kazakhstan has halted the implementation of an Internet surveillance system criticized by lawyers as illegal, with the government describing its initial rollout as a test." State security officials claimed they were trying to protect people in Kazakhstan from "hacker attacks, online fraud and other kinds of cyber threats." Right. Kazakhstan President said in a tweet that he had personally ordered the test which showed that protective measures 'would not inconvenience Kazakh Internet users." The President said "There are no grounds for concerns."

So, they've tried to do this again and again they have been blocked. Back in August of 2019 we conjectured that the only feasible path for them, if they insisted upon doing this, would be to create a Kazakhstan national web browser and that would be the only one that would work in-country. But it's unlikely that either Apple or Google or Microsoft would allow such a privacy-violating browser into their App Stores. So it's use would be strictly for desktops. But it's not feasible for a country to kill all use of mobile applications.

So... Browsers 3 / Kazakhstan 0.

Ransomware News

Announcing the RTF — The Ransomware Task Force
The newly christened Ransomware Task Force is a group of 19 security firms, tech companies,

and non-profits which include Microsoft and McAfee. Last Monday the group announced their plan to form a coalition to deal with the rising threat of ransomware. The group will focus on assessing existing technical solutions that provide protections during a ransomware attack. The RTF will commission expert papers on the topic, engage stakeholders across industries, identify gaps in current solutions, and then work on a common roadmap to have issues addressed among all members.

The group's target is the creation of a standardized framework for dealing with ransomware attacks across all market segments. And the single framework will be based upon an industry consensus rather than individual random advice received from lone contractors. The 19 initial founding members reflect the group's commitment to building a diverse team of experts:

- Aspen Digital (policy maker group)
- Citrix (networking equipment vendor)
- The Cyber Threat Alliance (cybersecurity industry sharing group)
- Cybereason (security firm)
- The CyberPeace Institute (non-profit dedicated to help victims of cyberattacks)
- The Cybersecurity Coalition (policy maker group)
- The Global Cyber Alliance (non-profit dedicated to reducing cyber risk)
- The Institute for Security and Technology (policy maker group)
- McAfee (security firm)
- Microsoft (security firm)
- Rapid7 (security firm)
- Resilience (cyberinsurance provider)
- SecurityScorecard (compliance and risk management)
- Shadowserver Foundation (non-profit security organization)
- Stratigos Security (cybersecurity consulting)
- Team Cymru (threat intelligence)
- Third Way (think tank)
- UT Austin Stauss Center (research group)
- Venable LLP (law firm)

The Ransomware Task Force website, including full membership details and leadership roles, will be launched next month, in January 2021, followed by a two-to-three month sprint to get the task force off the ground.

It's going to be interesting to see how this effort develops.

WordPress

This Week in WordPress we have more than 5 million WordPress sites in critical danger thanks to their use of the popular plug-in called “Contact Form 7.” The trouble arises from a lack of sufficient filename sanitization in the plug-in's file upload filter. This allows a file of the form xyz.php\t.jpg to be seen by the upload filter as a benign JPG file whereas it will be seen by the PHP interpreter as a valid PHP script.

The flaw has the CVE designation of 2020-35489 and Astra Security during a security audit for a client whose WordPress site was using the Contact Form 7 plug-in. A representative from Astra Security said: "Seeing the criticality of the vulnerability and the number of WordPress websites using this popular plugin, we quickly reported the vulnerability. The developer was even quicker in issuing a fix."

https://contactform7.com/

I'm impressed with the forthright communication of the Contact Form 7 people. The first thing on their homepage, not buried under some security updates menu, they clearly state:

> Contact Form 7 5.3.2 has been released. This is an urgent security and maintenance release. We strongly encourage you to update to it immediately.
>
> An unrestricted file upload vulnerability has been found in Contact Form 7 5.3.1 and older versions. Utilizing this vulnerability, a form submitter can bypass Contact Form 7's filename sanitization, and upload a file which can be executed as a script file on the host server.

And a ways down the page of chronological postings we find:

> **Heads-up about auto-updates** / August 24, 2020 Takayuki Miyoshi
>
> WordPress 5.5 has introduced the auto-update feature for plugins and themes. Keeping plugins and themes updated to the latest version is a key factor in managing your WordPress site securely. We strongly recommend you enable auto-updates for the Contact Form 7 plugin, but you should also be aware that there are risks involved in the use of auto-updates.
>
> In the following cases, consider disabling auto-updates and doing an update manually:
>
> - You use plugins that extend the functionality of Contact Form 7 (add-on plugins);
> - You use a theme that overrides the CSS style rules of Contact Form 7;
> - Or you apply coding customization of some sort to Contact Form 7.
>
> In those cases, updating Contact Form 7 or one of the plugins or themes that affect Contact Form 7 might bring about incompatibility risks between them, and if you do it automatically, you might not even realize problems are occurring on the site.
>
> Managing your sites securely is your responsibility. Update your plugins and themes in a proper way. If there is a plugin or theme that is an obstacle to updating other parts, you should make a decision to remove it.

So we have a mixed bag about updating. And of course this is only one of a great many plug-ins for WordPress. The only solution for someone who wants or needs to host their own WordPress site is to place the WordPress SQL database on the same machine, and tightly sequester the machine with absolutely minimal access to anything else. Only allow it to send and receive web, email, and DNS traffic, and never fail to treat it as untrusted and suspect.

Security News

Treck's TCP/IP stack strikes again!

The US Cybersecurity Infrastructure and Security Agency (CISA) has warned of critical vulnerabilities in the low-level TCP/IP software library developed by Treck that, if weaponized, could allow remote attackers to run arbitrary commands and mount denial-of-service (DoS) attacks.

The specific 4 flaws affect Treck's widely-used TCP/IP stack v6.0.1.67 and earlier and were reported to Treck by Intel. Two of these are rated critical in severity. And here's the problem: Treck's embedded TCP/IP stack is deployed worldwide in manufacturing, information technology, healthcare, and transportation systems.

The most severe of the four is a heap-based buffer overflow (CVE-2020-25066) in the Treck HTTP Server component that could permit an adversary to crash or reset the target device and even execute remote code. It has a CVSS score of 9.8 out of a maximum of 10. And of course we know what that means: Being in the HTTP server, any embedded device or IoT gizmo that exposes an HTTP service onto the public Internet, whether or not it provides strong access authentication, could nevertheless be compromised remotely.

The second flaw is an out-of-bounds write in the IPv6 component which received a CVSS score of 9.1, so somewhat less critical because it could be exploited by an unauthenticated user to cause a DoS condition via network access — so, just crashing a system remotely. But if that system were an industrial control system, crashing could be bad.

Two other vulnerabilities are an out-of-bounds read in the IPv6 component that could be leveraged by an unauthenticated attacker to cause DoS, and an improper input validation in the same module that could result in an out-of-bounds read of up to three bytes via network access.

Now, this next part I really liked... get a load of this: The CISA's disclosure said “Treck recommends users to update the stack to v6.0.1.68 to address the flaws. In cases where the latest patches cannot be applied, it's advised that firewall rules are implemented to filter out packets that contain a negative content-length in the HTTP header.” Ha! Yeah. So the hacks involve send queries containing a negative content length. Love it. There might be code that's checking for a content length that's greater than some value. But, as we know, in 2's compliment binary encoding, a signed negative value will appear as a very large unsigned value when it's interpreted by logic that's expecting to receive an unsigned length.

If the name "Treck" and the idea of disastrous TCP/IP stacks are ringing some bells, that would be because we first encountered these guys this past summer in June with the so-called Ripple 20 attacks — 19 different horrible security problems in their massively widely used embedded TCP/IP networking products.

They named the large set of vulnerabilities "Ripple 20" due to the ripple effects that are inherent when problems exist at one end of a long supply chain. The potential devastation will ripple out to affect hundreds of millions of devices. In this case, Treck's customers range from one-person boutique shops to Fortune 500 multinational corporations and include HP, Schneider Electric, Intel, Rockwell Automation, Caterpillar, Baxter, as well as many other major international

vendors in medical, transportation, industrial control, enterprise, energy (oil/gas), telecom, retail, commerce, and other industries.

Though we talked about this last summer, it's made even more chilling now, in the wake of the massive SolarWinds SunBurst intrusions. It's one thing to know intellectually that probably most IoT devices are highly vulnerable to remote attack and take over. But what really puts a sharp point on that is the idea that we now know without any question that there are entities bearing us malice who without any question also have the capability of turning theoretical vulnerabilities into fully practical attacks. **Let's hope that never happens.**

Closing The Loop

Anthony Lipke / @AnthonyLipke
@SGgrc I remain a fan of newgrounds a flash site. I haven't seen something take that place for games and animation. That said they're part of great efforts to preserve the content. You're likely already aware but it seems worth mentioning besides just saying flash is dead.
https://www.newgrounds.com/

Adobe's "Flash Player projector content debugger" available for Windows, Mac and Linux:
https://www.adobe.com/support/flashplayer/debug_downloads.html

BlueMaxima's Flashpoint is a webgame preservation project.
Internet history and culture is important, and content made on web platforms including, but not limited to Adobe Flash, make up a significant portion of that culture. This project is dedicated to preserving as many experiences from these platforms as possible, so that they aren't lost to time. Since early 2018, Flashpoint has saved more than 70,000 games and 8,000 animations running on 20 different platforms.

Flashpoint was started in January 2018 by BlueMaxima in an attempt to outrun the disappearance of content prior to the death of Flash. It has since evolved into an international project involving over 100 community contributors, encompassing both web games and animations created for numerous internet plugins, frameworks, and standards.
https://bluemaxima.org/flashpoint/

SpinRite

ReadSpeed is Ready!
Though the timing was never my plan, the first release of the ReadSpeed benchmark went widely public on Christmas Eve. The code had settled and had been stable for quite some time. And I'd had time to get ReadSpeed's home page at GRC ready. I annotated a sample run from my large multi-drive test system, and I made an 11-minute video walk through of that benchmark run with a voice-over commentary highlighting the various events of interest. So I thought, it's finally time to let the world have a crack at it! https://www.grc.com/readspeed.htm

In parallel, there's been a lot of research work going on over in the SpinRite development newsgroup regarding SpinRite's ability to improve SSD performance. Check-out this before and after benchmarking of an OCZ-VERTEX3 SSD:

```
  +-----------------------------------------------------------------------+
  | ReadSpeed: Hyper-accurate mass storage read-performance benchmark. rel 1 |
  |  Benchmarked values are in megabytes read per second at five locations.  |
  +-----------------------------------------------------------------------+
```

Driv	Size	Drive Identity	Location:	0	25%	50%	75%	100
81	60GB	OCZ-VERTEX3		300.1	320.4	324.0	371.5	371.9
				177.9	272.7	347.5	371.1	372.0
				89.0	292.4	291.3	372.4	371.9
				101.9	310.0	299.5	372.0	372.1
				102.2	228.8	335.4	371.8	372.0
				102.9	314.8	275.1	371.4	371.9
				113.4	332.8	278.0	371.9	372.0
				113.6	315.6	320.1	372.0	371.9
				100.5	334.4	341.8	372.0	371.5
				89.8	325.2	335.3	370.9	372.0
				90.2	332.9	336.7	372.4	371.8
				94.4	316.9	311.3	372.1	371.9
				292.1	329.3	325.4	372.1	371.8
				360.6	333.7	308.7	371.1	372.1
				364.2	300.2	308.1	372.0	371.9
				364.8	322.8	321.4	372.0	372.0
				359.7	335.8	310.7	371.9	371.9

Driv	Size	Drive Identity	Location:	0	25%	50%	75%	100
81	60GB	OCZ-VERTEX3		354.9	352.6	352.6	362.4	361.9
				366.2	350.9	354.3	360.2	360.6
				361.2	351.2	362.2	363.4	361.0
				362.3	351.7	346.8	362.7	360.4
				360.8	347.4	354.3	363.2	368.0
				363.3	359.3	352.3	362.8	359.6
				360.3	348.8	344.2	360.4	360.8
				359.7	350.5	344.0	361.4	363.6
				359.2	343.4	347.9	360.7	361.9
				363.2	352.2	352.4	369.5	361.7
				360.8	354.0	345.9	362.9	359.9
				363.2	348.9	347.9	366.4	357.4
				358.0	348.1	348.4	363.3	362.9
				361.7	353.8	355.3	363.5	360.0
				362.1	354.2	352.3	360.6	361.9
				361.7	346.0	351.6	360.3	355.0
				357.8	356.7	354.8	367.3	360.8

So... ReadSpeed, which is, as we know, the test platform for the new high performance drivers I've been developing for the next and all future SpinRite's and "Beyond Recall" products, is now ready for people to play with. The top of the ReadSpeed page declares: "What you discover is going to surprise you." ... and I think that's probably true. So grab any old USB stick that you have lying around, run ReadSpeed for Windows and it will prep the USB drive to boot DOS. And for Linux folks, the bottom of the page offers an image file of an 8 megabyte bootable DOS file system which can simply be 'dd'ed to a USB root device and then booted.

InitDisk is at release 5.
There had been some anecdotal reports of write failures, where InitDisk would fail to reformat a USB stick. Before I used the IntiDisk technology for ReadSpeed, I wanted to get to the bottom of the trouble. Fortunately one of our contributors in the newsgroups had two USB sticks that were doing this. So he sent me a cleaned image of the drive and I was able to reproduce and fix the trouble. It turned out to be caused by some different way that Windows 10 operates since Windows 7 had no trouble. If, under Win10 the existing USB stick format being overwritten did not have any partition table, InitDisk's attempt to change the stick to a partitioned drive would fail, but only under Win10. That's fixed now so that InitDisk, ReadSpeed and SpinRite will all be able to handle that situation.

Sunburst & Supernova

It should not surprise anyone that more intelligence is being continually uncovered about the event that's being called the biggest computer hack in history — and I would argue that it's also been the most embarrassing for the U.S. So, as we wind up this horrific year 2020, with the discovery of this widespread computer network intrusion still being only three weeks old, let's look at the additional significant things that have been learned since we first discussed this two weeks ago:

One thing I want to clear up first, was that two weeks ago I said that SolarWinds' Orion was an appliance. That was incorrect. It's just a software system that can be loaded and run on any qualifying Windows system. So I wanted to correct that for the record.

To my mind, the biggest revelation since the initial discovery of the Orion .DLL being hacked, signed and then delivered to approximately 18,000 SolarWinds customers via a software update, is that compelling evidence has been found of a second entirely separate second backdoor in SolarWinds offerings. And the evidence leads forensic investigators to believe that this second "SuperNova" backdoor — as it has been named — was planted by a second threat actor.

(At this point, if you had any equity stake in SolarWinds you're probably not happy.)

This second piece of "SuperNova" malware is an extremely sophisticated webshell which was also planted in the code of Orion's network and applications monitoring platform. It enabled the attackers to run arbitrary code on any of the machines hosting the Trojanized version of the software.

The webshell is a modified version of a legitimate .NET library DLL named "app_web_logoimagehandler.ashx.b6031896.dll" that's present in SolarWinds' Orion software. And that DLL was very cleverly designed to get its nefarious job done while proactively evading automated defense mechanisms.

By design, the Orion software uses this DLL to expose an HTTP API which allows the host to respond to other subsystems when querying for a specific GIF image. In his technical report published on the 17th, Matt Tennis a Senior Staff Security Researcher at Unit 42 of Palo Alto Networks, wrote that the malware could slip past even careful manual analysis since the code implemented in the legitimate DLL is innocuous and is of "relatively high quality."

https://unit42.paloaltonetworks.com/solarstorm-supernova/

From Matt's introduction to this SECOND threat, we also learn something significant about the attackers that hasn't been widely reported. Matt wrote:

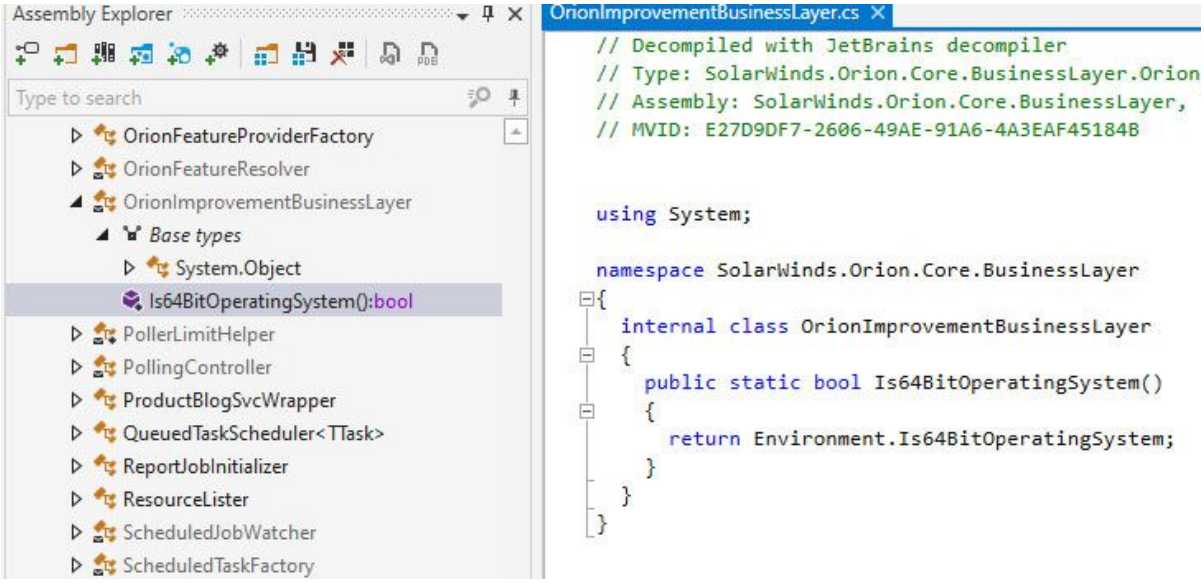
> The actors behind the supply chain attack on SolarWinds' Orion software have demonstrated a high degree of technical sophistication and attention to operational security, as well as a novel combination of techniques in the potential compromise of approximately 18,000 SolarWinds customers. As published in the original disclosure, the attackers were observed removing their initial backdoor once a more legitimate method of persistence was obtained.

In other words, the attackers used their initial backdoor intrusion mechanism ONLY until they were able to obtain keys to the front door. At which point the backdoor mechanism was shut down and went quiescent to avoid any chance of its detection.

In describing the second SuperNova backdoor, Matt writes:

> The analysis shows that the threat actor added in[to] the legitimate SolarWinds file, four new parameters to receive signals from the command and control (C2) infrastructure. The malicious code contains only one method, DynamicRun, which compiles on the fly the parameters into a .NET assembly in memory, thus leaving no artifacts on the disk of a compromised device.
>
> This way, the attacker can send arbitrary code to the infected device and run it in the context of the user, who most of the time will have high privileges and visibility on the network. It is unclear how long SUPERNOVA has been in the Orion software but a malware analysis system shows a compilation timestamp of March 24, 2020.

Now, the timing of that may seem to be very close to the March 6th, 2020 signing of the first instance of the original SunBurst backdoor. But we have also since learned that the initial intrusion by that first actor into SolarWinds was likely made back in the previous October, 2019.



A benign "do nothing" change was found in the SolarWinds source code base dating back to October 2019. It literally does nothing. And researchers conjecture that the change was injected just to allow them to determine whether this change would (a) go undetected and (b) eventually be disseminated out into SolarWind's global customer base.

Matt says of this second intrusion that based upon the findings of the investigation, SuperNova bears the hallmarks of an advanced hacking group that took compromise via a webshell to a new level. He wrote:

> "Although .NET webshells are fairly common, most publicly researched samples ingest command and control parameters, and perform some relatively surface-level exploitation."

He said that taking a valid .NET program as a parameter, and performing in-memory code execution, makes SuperNova a rare encounter, as it eliminates the need for additional network callbacks aside from the initial C2 request. Most webshells run their payloads in the context of the runtime environment or by calling a subshell or process such as CMD, PowerShell, or Bash.

Microsoft believes that this SuperNova webshell is likely the creation of a different adversary than the one that was first discovered by FireEye. Microsoft wrote:

> "In an interesting turn of events, the investigation of the whole SolarWinds compromise led to the discovery of an additional malware that also affects the SolarWinds Orion product but has been determined to be likely unrelated to this compromise and used by a different threat actor."

One argument for this theory is that unlike the SunBurst DLL that slipped into SolarWinds source code repository and was thus validly signed, SuperNova does not have a digital signature.

Kim Zetter, writing for Yahoo! News added some good detail to the saga. Kim wrote:

Hackers who breached federal agency networks through software made by a company called SolarWinds appear to have conducted a test run of their broad espionage campaign last year, according to sources with knowledge of the operation.

The hackers distributed malicious files from the SolarWinds network in October 2019, five months before previously reported files were sent to victims through the company's software update servers. The October files, distributed to customers on Oct. 10, did not have a backdoor embedded in them, however, in the way that subsequent malicious files that victims downloaded in the spring of 2020 did, and these files went undetected until this month.

A source familiar with the investigation told Yahoo News: "We're thinking they wanted to test whether or not it was going to work and whether it would be detected. So it was more or less a dry run. They took their time. They decided to not go out with an actual backdoor right away. That signifies that they're a little bit more disciplined and deliberate."

The October files were discovered in the systems of several victims, but investigators have so far found no signs that the hackers engaged in any additional malicious activity on those systems after the files landed on them.

The original SunBurst malware discovered by FireEye used a Domain name Generation Algorithm (DGA) to obscure the lookups that it was doing to find its command and control server. And let's not forget that this was successful until whatever it was that FireEye discovered.

The cool thing is, as part of its detection avoidance system, the malware incorporated its own internal killswitch. So, working together, Microsoft, FireEye and GoDaddy have decided to cause the malware to shut itself down:

After the obscure domain name is generated, as a subdomain of avsvmcloud.com, a DNS Address record lookup is performed. The resolved address is checked against a hard coded list of networks, one of which happens to belong to Microsoft. And if the IP address matches any of those networks, the malware will update a configuration key named "ReportWatcherRetry" to prevent its own further execution and will then terminate itself permanently.

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 224.0.0.0/3
- fc00:: - fe00::
- fec0:: - ffc0::
- ff00:: - ff00::
- 20.140.0.0/15 ← A Microsoft IP block
- 96.31.172.0/24
- 131.228.12.0/22
- 144.86.226.0/24

Also note that the IP addresses querying for the domain's A record can be obtained — which is where GoDaddy comes in — to reveal the IP of the local DNS server resolving for any still-active malware. Thus it's possible to determine who has any still-active intrusion.

That's all good... but as we've often noted, once a bad guy has been crawling around inside a network, especially a huge and complex network and especially a highly skilled bad guy, it's really never possible to know that everything that they may have done while they were inside there has been found and reversed. Remember that it's quite possible for malware to take up residence inside a printer or a security camera, or pretty much anything else these days.

FireEye was quoted by the Tech Press saying:

"In the intrusions FireEye has seen, this actor moved quickly to establish additional persistent mechanisms for access to victim networks beyond the SunBurst backdoor. This killswitch will not remove the actor from victim networks where they have established other backdoors."

The U.S. CISA's follow-up statement, which talked about the ongoing discovery of the breadth and depth of this attack said: "This APT actor has demonstrated patience, operational security, and complex tradecraft in these intrusions. CISA expects that removing this threat actor from compromised environments will be highly complex and challenging for organizations."

One final point is that the obfuscation used by the domain name generator has been reverse engineered. And logs of previous DNS queries have been obtained and decoded. This resulted in the discovery of many additional infected corporations who have not yet gone public with any disclosures. And a bunch of them are quite tasty.
The biggest names on this list include the likes of Cisco, SAP, Intel, Cox Communications, Deloitte, Nvidia, Fujitsu, Belkin, Amerisafe, Lukoil, Rakuten, Check Point, Optimizely, Digital Reach, Digital Sense and probably MediaTek.

(See a sample of some of the decoded domains on the next page.)

```
882  q1b91c4fdd7q4td56rswoiou0govirsv.appsync-api.us-east-1.avsvmcloud.com          servitia.intern
883  q3b8h3lm9q7eoqa56260kun0e6iuir0e.appsync-api.us-east-2.avsvmcloud.com          sos-ad.state.
884  q3vcrhhcmddh7rl5oi602ou6iuir0grn.appsync-api.us-east-2.avsvmcloud.com          its.iastate.ed
885  q80cgv4eolosbfo4tvef0t12eu1.appsync-api.us-east-1.avsvmcloud.com            gncu.local
886  q882csbrq5oa58d4r6eud0i2st.appsync-api.us-east-1.avsvmcloud.com          escap.org
887  q8bps26mocuq6re4dutru70ct2w.appsync-api.us-east-1.avsvmcloud.com            pageaz.gov
888  q8g11thobvg6d604tvef0b12eu1.appsync-api.us-east-1.avsvmcloud.com            gncu.local
889  sf0q84qdutb323q6eo6e202e2h.appsync-api.us-east-1.avsvmcloud.com          cisco.com
890  q8vmaei8n3dpeui5vr2d32i2voe60be2.appsync-api.us-east-1.avsvmcloud.com          neophotonics.co
891  qb9it88vftri6v84euheoip0e12eu1.appsync-api.us-west-2.avsvmcloud.com          camcity.local
892  qbj26i5jnkrqdac5wh602un0twusouv0.appsync-api.us-west-2.avsvmcloud.com          vms.ad.varian
893  1cmge6dsclrtfejc6e0gdohu0et2w.appsync-api.us-east-1.avsvmcloud.com            sc.pima.gov
894  qfnf6ab6u28je4d5un0b2dioho7r1p0b.appsync-api.us-east-2.avsvmcloud.com          ad.optimizely.
895  qfnf6ab6u28je4i5un0c2dioho7r1p0c.appsync-api.us-east-2.avsvmcloud.com          ad.optimizely.
896  qg1e4bctbk3gdkr4e2sd0bdieo0be2h.appsync-api.us-east-1.avsvmcloud.com          corp.ptci.com
897  qgc2gj97t3sop4i5uhs0be2sd0govir1.appsync-api.us-east-1.avsvmcloud.com          amr.corp.intel
898  qgdubroda1vph414srd6sw0oe2h.appsync-api.us-east-1.avsvmcloud.com            repsrv.com
899  qipotpf1jic4gav5oi60eou6iuir0grn.appsync-api.us-east-2.avsvmcloud.com          its.iastate.ed
900  qit94i5tqf2j9mq5wo11r02irssrc2vv.appsync-api.us-east-2.avsvmcloud.com          ville.terrebonn
901  qj1bggoa06prfj646d6n0g6j02eu.appsync-api.us-east-1.avsvmcloud.com            spsd.sk.ca
902  qj82njdvtfuoi455uhs0be2sd0govir1.appsync-api.us-east-1.avsvmcloud.com          amr.corp.intel
903  qo046rspifbl4k04e2mvri0ge2m0te2h.appsync-api.us-east-2.avsvmcloud.com          coxnet.cox.com
904  qrieo21mr659tfk5wh60iun0bwusouv0.appsync-api.us-west-2.avsvmcloud.com          vms.ad.varian
905  qrjtdj3alnlcj0k4urso2ve2sd0be2h.appsync-api.us-west-2.avsvmcloud.com          aerioncorp.com
906  qvot463cl5rcg5r4urso2ve2sd02e2h.appsync-api.us-west-2.avsvmcloud.com          aerioncorp.com
907  r14ptgkl7qacucu5chsv0ee2h.appsync-api.us-west-2.avsvmcloud.com          bmrn.com
908  r1q6arhpujcf6jb6ervisu10odohu0it.appsync-api.us-west-2.avsvmcloud.com          central.pima.g
909  r1qshoj05ji05ac6eoip02jovt6i2v0c.appsync-api.us-west-2.avsvmcloud.com          city.kingston.
910  r69ncekf56jlkkr6oi602ou6iuir02rn.appsync-api.us-east-2.avsvmcloud.com          its.iastate.ed
911  r6b5cj43deojp665u30c2st.appsync-api.us-east-2.avsvmcloud.com          ah.org
912  r74br8r0cce4m6r6oi60eou6iuir0trn.appsync-api.us-east-2.avsvmcloud.com          its.iastate.ed
913  r75n0q0557bl6nv6oi60cou6iuir0orn.appsync-api.us-east-2.avsvmcloud.com          its.iastate.ed
914  r7kqk893t5lu82j6uhs0ie2sd0iovir1.appsync-api.us-east-2.avsvmcloud.com          amr.corp.intel
```