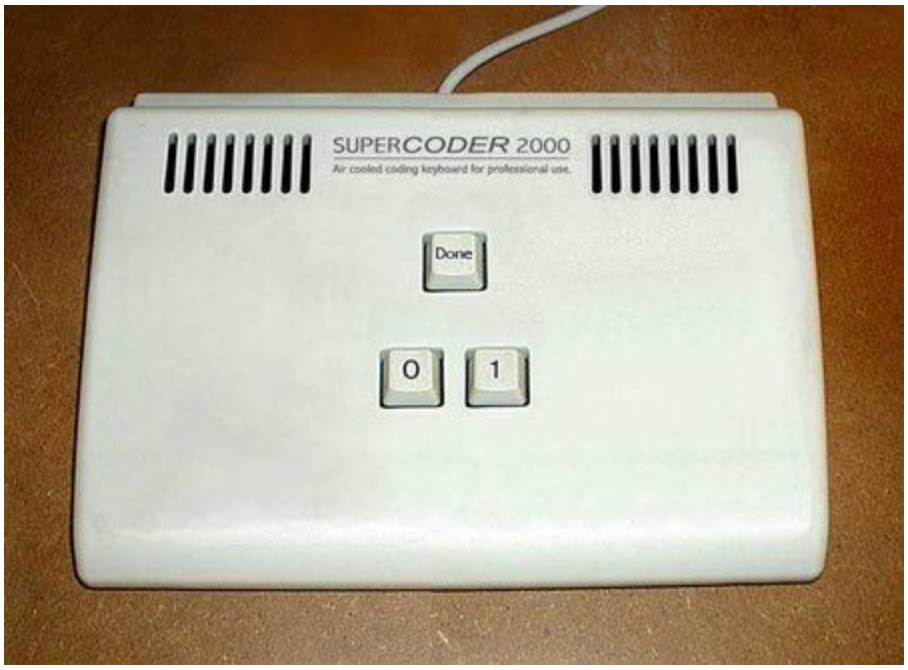
# Security Now! #787 - 10-06-20

# Why Win7 Lives On

## This week on Security Now!

This week we examine several new and welcome Google initiatives aimed at improving Android general web browser security. We look at Microsoft's solution for updating aging Windows offline images with the latest Defender definitions. We note some surprising network behavior from Windows second Subsytem for Linux. We check-in on Exchange Server updates after eight months. We cover Cloudflare's announcement of a very welcome WebAPI firewall, the US Treasury's recent policy regarding Ransomware payments, and Kaspersky's discovery of the use of UEFI Bootkits. Then we have a bit of errata and a GRC forums update. And we conclude by sharing the results of an interesting poll which illuminates the many reasons why Windows 7 refuses to die.



# Security News

**Google to get even more proactive about Android security**

Google will soon be getting even more proactive about Android security. We know that Google runs their own bug bounty program for Android apps listed on the Play Store. That program is called the Google Play Security Reward Program (GPSRP), through which independent security researchers are rewarded for locating and responsibly reporting bugs they find in Play Store Android apps. Google is, in essence, paying for bug submissions on behalf of its platform's app publishers.

One limitation of the GPSRP is that it is restricted to major apps having more than 100 million users. And Google noticed that other apps that handle sensitive data, or may be performing critical tasks, often aren't eligible for GPSRP rewards and are, therefore, far less likely to be tested by bug hunters.

Thus, in a new job posting that went online last Wednesday, Google wrote:
"As a Security Engineering Manager in Android Security [...] Your team will perform application security assessments against highly sensitive, third party Android apps on Google Play, working to identify vulnerabilities and provide remediation guidance to impacted application developers."

According to Sebastian Porst, the Software Engineering Manager for Google Play Protect, this new team will be focusing on apps such as COVID-19 contact tracing apps and election-related applications, with others to follow.

In their coverage of this, ZDNet quoted Lukáš Štefanko, a mobile malware analyst at the Slovak security firm ESET. Lukáš said that it was "Definitely a good move" and also that "Finding security issues with serious impact isn't that easy and requires a lot of time and experience."

To Google's credit, having a dedicated team ensures that some of the world's best security talent and full effort can be put into looking at apps that might slip under the radar and end up being exploited with devastating consequences.

**And Google funds a JavaScript research engine**

Meanwhile, Google has also established a new grant program to fund research aimed at locating bugs in the industry's JavaScript engines. Eligible targets are Google Chrome's own V8 engine, Firefox's Spidermonkey and Safari's JavaScriptCore, though Google indicated that other engines could be pitched in a grant proposal.

The program will help and sponsor security researchers and academics to find vulnerabilities hidden inside any of those three JavaScript engines and the program has only one rule: The bugs must be identified using "fuzzing."

Since fuzzing tends to be a resource intensive process, it has traditionally been the province of larger tech companies that can afford the resources required to setup comprehensive fuzzing operations. To be done at the scale required, fuzzing typically employs large and expensive cloud computing; resources that are beyond the reach of most solo security researchers working on their own in the hopes of finding a significant bug. And, of course, the cloud resources need to

be paid for whether or not there's ever any bounty payoff. And when bounties are paid, it can be many months later.

In their blog post last Thursday, Google said it created this research grant to address exactly this problem. Under Google's new pilot program, security researchers and academics can apply for the funds required to "fuzz" any of those three JavaScript browser engines. Google said it will analyze each submission and provide an answer to all applicants within two weeks with approved projects receiving up to $5,000 in funding. Naturally, since Google is a provider of exactly such cloud services, the grant funding will actually be up to $5,000 in credits to be used on the Google Compute Engine, which is Google Cloud's heavy computing infrastructure.

This pilot program will run for one year, from last Thursday, October 1st, 2020 to October 1st of 2012. Google calls this program the Fuzzilli Research Grant, named after Google's own Fuzzilli open-source fuzzing tool, which supports distributed fuzzing on the Google Compute Engine, and which Google naturally encourages grant recipients to use. Although Google said that all bugs identified during the pilot program must be reported to affected vendors, researchers may retain any bug bounty payouts for any bugs they might find during the pilot program.

Samuel Groß, the creator of Fuzzilli, and a member of the Google Project Zero team said "JavaScript engine security continues to be critical for user safety, as demonstrated by recent in-the-wild 0-day exploits abusing vulnerabilities in Chrome's v8 JavaScript engine."

So... this is kinda neat.  If you might have some time on your hands during the COVID mess — or just during spare time on evenings and weekends — you could figure out how Fuzzilli works, play with it on your own machine to work out the details. Then apply for a $5,000 grant to scale its use up to Big Cloud Iron. If you get lucky, you might uncover a previously unknown and valuable flaw. And score yourself some cash without incurring any other out-of-pocket expenses.

https://github.com/googleprojectzero/fuzzilli
https://googleprojectzero.blogspot.com/2020/10/announcing-fuzzilli-research-grant.html

Samuel's blog posting added:

> Submissions are not limited to those in academia or those with a demonstrated track record of success - if you have a good idea in this space, we'd love to hear from you. Incoming submissions will be reviewed by a review board on a regular basis and we aim to respond to every submission within 2 weeks. If the project is accepted, the researchers may be awarded GCE credits worth up to USD $5,000. Researchers can also apply for multiple grants throughout the lifetime of a project. The grants come with the following requirements:
>
> The credits must be used for fuzzing JavaScript engines with the approach described in the proposal. The fuzzed JavaScript engines should be one or more of the following: JavaScriptCore (Safari), v8 (Chrome, Edge), or Spidermonkey (Firefox).
>
> All vulnerabilities found must be only reported to the affected vendor. Researchers are encouraged to apply Project Zero's 90-day disclosure policy. Researchers may claim any CVE credits and bug bounty payouts for reporting the bugs that don't conflict with these requirements:

- Any newly developed source code must be published under an open source license that permits further research by others.

- An interim report for Google only at the conclusion of the fuzzing, to demonstrate the initial results of the research, so we can determine the efficacy of the research and make our folks in accounting happy.

- Furthermore, a final report of some form (e.g. a conference paper, a blog post, or a standalone PDF) due no later than 6 months after the first grant for a project has been awarded, including:

    - A detailed explanation of the project
    - Basic statistics about which engines have been fuzzed for how long (CPU time, iterations, etc.)
    - A clear technical explanation of all vulnerabilities discovered throughout the project.

Researchers are encouraged to base their project on the open source Fuzzilli fuzzer if possible, which, amongst other features, already supports distributed fuzzing on GCE.

**Microsoft Defender gets *in Vitro* Updating**

Many enterprise environments use a fixed image of Windows to set up new workstations. And these images are often used for many months at a time. And there's really no point in updating to a newer download of the same image, since it will be the same image.

The trouble is that as these images are aging their built-in Windows Defender is becoming less and less relevant because it's aging too and it won't know about any of the newer threats that have been discovered and added to the current Defender A/V definitions.

And yes, it's true that AFTER the image has been deployed onto a new machine, and that machine has gone online, downloaded everything that has happened since the original image was downloaded, installed those updates and rebooted, then it, too, will have the latest and greatest.

But the problem, in Microsoft's words, is that this still leaves a "protection gap" during which a machine that's booted, up and running and on the network could have some seriously outdated Defender definitions.

To close this protection gap, on Friday Microsoft released a new tool for both 32- and 64-bit systems that allows an up-to-the-minute version of Windows Defender to be inserted into an offline WIM or VHD Windows image, thus bringing the image's A/V awareness current before it's used to create any running systems.

For 32-bit systems: https://go.microsoft.com/fwlink/?linkid=2144828
For 64-bit systems: https://go.microsoft.com/fwlink/?linkid=2144531

The links point to ZIP files, each which contains two files: an updated Windows Defender .CAB file and a PowerShell script named "defenderupdatewinimage.ps1"

Microsoft Defender update for Windows Operating System installation Images:
https://support.microsoft.com/en-us/help/4568292/defender-update-for-windows-operating-system-installation-images

**Run a Bypass!**
WSL 2 (Windows Subsystem for Linux v2) completely bypasses the hosting Windows 10 firewall.

The first version of the Windows Subsystem for Linux (WSL 1) is implemented using a Linux-emulating pseudo-kernel that translates Linux system calls into WinNT kernel calls. Therefore, under WSL 1, any network traffic is actually coming from Windows and is thereby filtered through the standard built-in Windows Advanced Firewall (WAF). The Linux distro honors any configured rules because it's behind that firewall because it's also actually behind the Windows kernel.

But, WSL 2 does it differently. In WSL 2, Microsoft produced a true Linux kernel operating side-by-side with Windows in a Hyper-V virtual machine and with a Hyper-V virtual network adapter. As a consequence of this completely different architecture, unlike with WSL 1, WSL 2 traffic is sent directly to the virtual network adapter, completely bypassing the Windows Firewall.

Now, this is not itself a bad thing. WSL 2 is way more powerful than WSL 1, and the architecture is correct. But it does mean that an unwary user might mistakenly assume that they have the same merged Windows firewall protection after updating the WSL 2 as they had under WSL 1, which could be leaving them exposed.

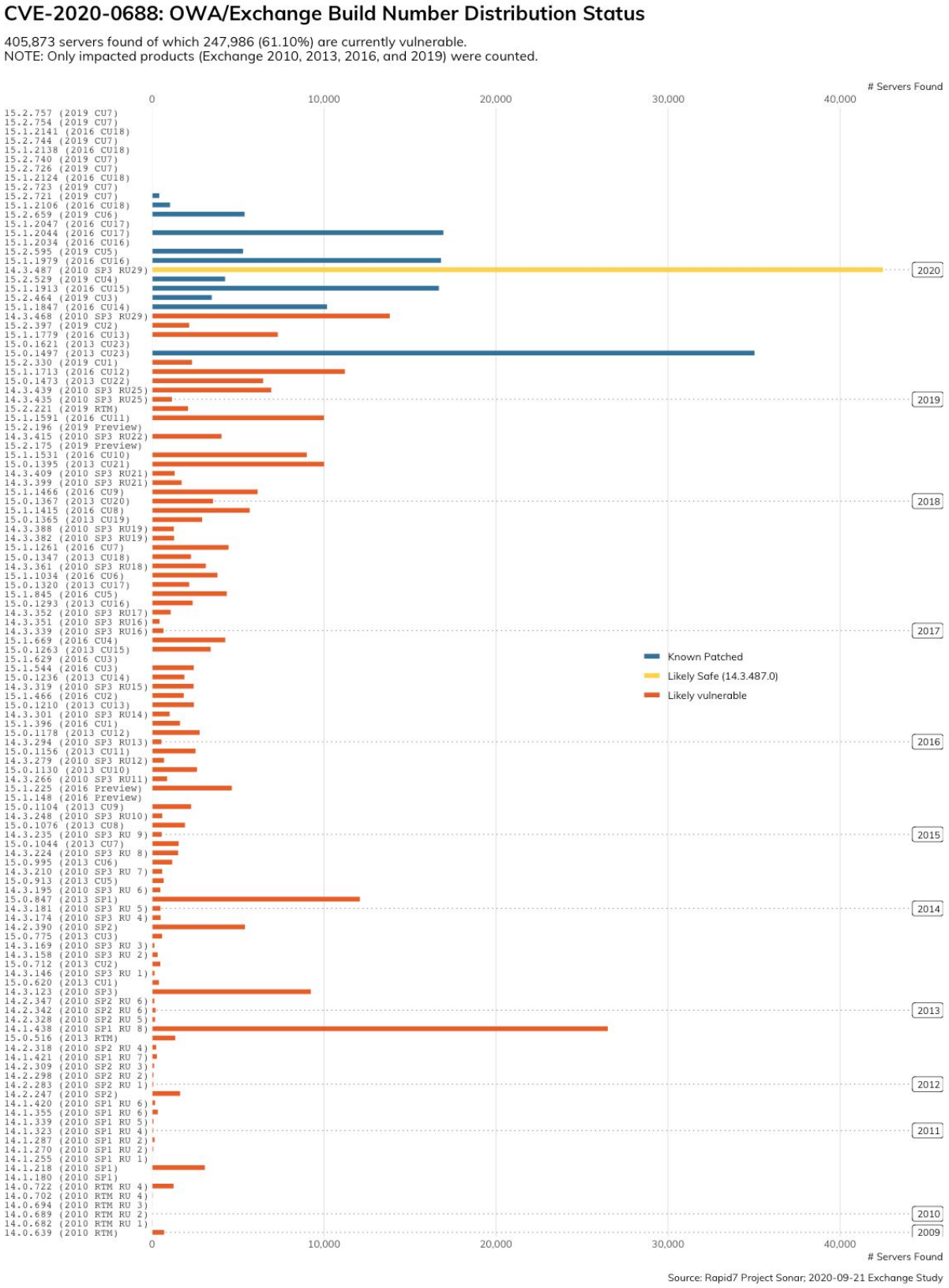
So just beware that you'll need to use the standard Linux network firewalls and filters when securing any WSL 2-based system.

**Most Microsoft Exchange Servers remain unpatched after 9 eight months!**
We talked about this at the time. On Feb. 11, 2020, Microsoft released security updates to address a vulnerability in Microsoft Exchange that would allow an attacker to turn any stolen Exchange user account into a complete system compromise. In many implementations, this could be used to completely compromise the entire Exchange environment (including all email) and potentially all of Active Directory. One month later, any admins who were paying attention would have also learned that any still-unpatched servers were then being exploited in the wild by unnamed advanced persistent threat (APT) actors.

So what's the story today? Somewhat unbelievably, Rapid7 conducted a survey on September 21st and found that out of the 433,464 open and Internet-facing Exchange servers which could be observed, at least 61% of Exchange 2010, 2013, 2016 and 2019 servers have remained vulnerable to the flaw... which is not theoretical and which is known to be exploited in the wild.

We moan and groan and feel rightly sorry when large enterprises are struck by debilitating ransomware, having their information exfiltrated, being publicly shamed, having their users' data

put at risk of exposure and often needing to pay a hefty ransom. But when you learn that 267,986 individual Exchange Servers remain vulnerable to remote exploitation TODAY, eight months after a patch to close this vulnerability has been made available, well... no one deserves to be illegally attacked. It's still illegal. But wearing a sign that says “Kick Me” is known as asking for it.



**Cloudflare has just added a free web API firewall service for all customers**
Last Thursday they announced their new "API Shield."

https://blog.cloudflare.com/introducing-api-shield/

> APIs are the lifeblood of modern Internet-connected applications. Every millisecond they carry requests from mobile applications—place this food delivery order, "like" this picture—and directions to IoT devices—unlock the car door, start the wash cycle, my human just finished a 5k run—among countless other calls.
>
> They're also the target of widespread attacks designed to perform unauthorized actions or exfiltrate data, as data from Gartner increasingly shows: "by 2021, 90% of web-enabled applications will have more surface area for attack in the form of exposed APIs rather than the UI, up from 40% in 2019, and "Gartner predicted that, by 2022, API abuses will move from an infrequent to the most-frequent attack vector, resulting in data breaches for enterprise web applications" Of the 18 million requests per second that traverse Cloudflare's network, 50% are directed towards APIs—with the majority of these requests blocked as malicious.

A perfect example of a web API that we've talked about recently was the SSAPI — SQRL Service Provider API — that I designed and created for Rasmus to use when he created the SQRL client for XenForo. I didn't want to burden him with the need to learn anything about SQRL. And I realized that this would be a general boon for SQRL's server-side adoption. So I hid and encapsulated all of SQRL's various details behind a very simple, abstract authentication API, accessible through a simple HTTP query/reply. In that way, Rasmus could simply post HTTP queries using any Web agent (in this case PHP) and receive meaningful replies. This is the model that's evolving for the entire industry. It allows for the creation of elegant and clean interfaces. In the SQRL use-case, I was able to make the interface private, since it only needed to support transactions between Rasmus' PHP code and mine which both resided on the same server or network. But the case that Cloudflare is addressing requires much more security, since, as the Gartner Group notes, most of these APIs need to be publicly available.

Cloudflare continued their announcement:

> To combat these threats, Cloudflare is making it simple to secure APIs through the use of strong client certificate-based identity and strict schema-based validation. As of today, these capabilities are available free for all plans within our new "API Shield" offering. And as of today, the security benefits also extend to gRPC-based APIs, which use binary formats such as protocol buffers rather than JSON, and have been growing in popularity with our customer base.

And they've implemented a proper "deny all" model that only permits valid formatted API calls to pass. They explained:

> A "positive security" model is one that allows only known behavior and identities, while rejecting everything else. It is the opposite of the traditional "negative security" model

enforced by a Web Application Firewall (WAF) that allows everything except for requests coming from problematic IPs, ASNs, countries or requests with problematic signatures (SQL injection attempts, etc.).

Implementing a positive security model for APIs is the most direct way to eliminate the noise of credential stuffing attacks and other automated scanning tools. And the first step towards a positive model is deploying strong authentication such as mutual TLS authentication, which is not vulnerable to the reuse or sharing of passwords.

Just as we simplified the issuance of server certificates back in 2014 with Universal SSL, API Shield reduces the process of issuing client certificates to clicking a few buttons in the Cloudflare Dashboard. By providing a fully hosted private public key infrastructure (PKI), you can focus on your applications and features—rather than operating and securing your own certificate authority (CA).

Once developers can be sure that only legitimate clients (with SSL certificates in hand) are connecting to their APIs, the next step in implementing a positive security model is making sure that those clients are making valid requests. Extracting a client certificate from a device and reusing elsewhere is difficult, but not impossible, so it's also important to make sure that the API is being called as intended.

Requests containing extraneous input may not have been anticipated by the API developer, and can cause problems if processed directly by the application, so these should be dropped at the edge if possible. API Schema validation works by matching the contents of API requests—the query parameters that come after the URL and contents of the POST body—against a contract or "schema" that contains the rules for what is expected. If validation fails, the API call is blocked protecting the origin from an invalid request or a malicious payload.

Schema validation is currently in closed beta for JSON payloads, with gRPC/protocol buffer support on the roadmap. If you would like to join the beta please open a support ticket with the subject "API Schema Validation Beta". After the beta has ended, we plan to make schema validation available as part of the API Shield user interface.

Their announcement continues with a demonstration and examples of API schema definitions that explicitly define valid query formats and which are used to drive their query-validation firewall.

So, once again, Cloudflare is innovating and leading the way with some powerful protections for their customers. To which I say BRAVO!

As we know, I'm still maintaining a full height rack of physical servers and network equipment in a nearby Level3 data center. Three years ago, while attending a DigiCert Customer Advisory Board meeting in Utah, I happened to mention GRC's rack of equipment. A bunch of the networking gurus turned toward me as one with a look of puzzled brow furling and one of them said: "What!?!?! No one does hardware any more!" Oh. I guess I am old school. But I took their point. And at some point in the future, when it no longer makes sense for GRC to have a rack of equipment at Level3, but before I'm ready to abandon GRC.COM, I'll likely move it to the could, and Cloudflare will probably be my first choice for its new and final home.

**US Dept of the Treasury tightens up on Ransomware payments**
Last Thursday, the US Treasury Department issued revised guidelines which, among other things, are targeting the new phenomenon of Ransomware negotiators.

https://home.treasury.gov/policy-issues/financial-sanctions/recent-actions/20201001
https://home.treasury.gov/system/files/126/ofac_ransomware_advisory_10012020_1.pdf

Get a load of this:

> The U.S. Department of the Treasury's Office of Foreign Assets Control (OFAC) is issuing an advisory to alert companies that engage with victims of ransomware attacks of the potential sanctions risks for facilitating ransomware payments. This advisory highlights OFAC's designations of malicious cyber actors and those who facilitate ransomware transactions under its cyber-related sanctions program. It identifies U.S. government resources for reporting ransomware attacks and provides information on the factors OFAC generally considers when determining an appropriate enforcement response to an apparent violation, such as the existence, nature, and adequacy of a sanctions compliance program. The advisory also encourages financial institutions and other companies that engage with victims of ransomware attacks to report such attacks to and fully cooperate with law enforcement, as these will be considered significant mitigating factors.

The 5-page PDF goes into all the details. But, effectively, the Treasury Dept. wants to be notified of Ransomware attacks because large sums of money are crossing US borders and potentially moving into the hands of sanctioned individuals and/or governments.

And the presumption is that, when notified, Treasury will simply say no. A Treasury official asked about this specifically said that "license applications involving ransomware payments demanded as a result of malicious cyber-enabled activities will be reviewed by OFAC on a case-by-case basis with a presumption of denial." <unquote>

So this really puts the victims, their negotiators and their insurers in an awkward hot seat position; especially when the circumstances strongly compel the victim to desire to pay the ransom — with the help of various intermediaries and insurers. So the whole thing is becoming a big mess.

**UEFI Bootkits are becoming more mainstream**
The first spotting of a UEFI Bootkit was made by ESET just over two years ago. We covered it here at the time. Their posting at "WeLiveSecurity" was titled: "LoJax: First UEFI rootkit found in the wild, courtesy of the Sednit group"

https://www.welivesecurity.com/2018/09/27/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/

Their write-up, two years ago, begins by explaining:

UEFI rootkits are widely viewed as extremely dangerous tools for implementing cyberattacks, as they are hard to detect and able to survive security measures such as operating system reinstallation and even a hard disk replacement. Some UEFI rootkits have been presented as proofs of concept; some are known to be at the disposal of (at least some) governmental agencies. However, no UEFI rootkit has ever been detected in the wild – until we discovered a campaign by the Sednit APT group that successfully deployed a malicious UEFI module on a victim's system.

And now, today, UEFI Bootkits are back in the news thanks to some research findings from Kaspersky. Kaspersky's 30-page PDF is titled: "MosaicRegressor: Lurking in the Shadows of UEFI" ... and it's not a PR piece... this write-up drops right into the technology:

https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2020/10/05094208/MosaicRegressor_Technical-details.pdf

So what we know is that a Chinese-speaking hacking group has been observed using a UEFI bootkit to download and install additional malware on targeted computers.

Fortunately, UEFI firmware attacks remain rare because getting malicious code into the motherboard's firmware is still difficult. Attackers either need physical access to the machine or they need to compromise their targets through complex supply chain attacks where the UEFI firmware or tools that work with UEFI firmware are modified to insert malicious code.

Kaspersky's malware researchers said they investigated some suspicious systems and discovered malicious code inside the UEFI firmware. The code was designed to install (and re-install) a malicious app (as a Windows autorun program) after every computer start. So even if the program was removed, the next system boot would bring it back.

This autorun program acts as a downloader for other malware components, which Kaspersky named the MosaicRegressor malware framework.

Kaspersky said that they had not yet obtained and analyzed all of MosaicRegressor's components, but the one that they did look at contained functionality to gather all the documents from the "Recent Documents" folder, putting them into a password-protected archive, presumably preparing the files for exfiltration via another component.

The researchers found the UEFI bootkit on only two systems, but they found MosaicRegressor components on many other computers. And Kaspersky believed that the targets of these attacks were all carefully selected. All were diplomatic entities and NGOs in Africa, Asia, and Europe. So that's certainly not random.

Kaspersky wrote: "Based on the affiliation of the discovered victims, we could determine that all had some connection to the DPRK [North Korea], be it non-profit activity related to the country or actual presence within it."

And Kasperksy also discovered that the malicious UEFI code wasn't exactly new. According to their analysis, the code was based on VectorEDK, which is a hacking utility to attack UEFI

firmware, created by a now-defunct Italian vendor of hacking tools, exploits, and surveillance software that was known as “HackingTeam.”

HackingTeam were, themselves, hacked back in 2015, and their tools, including the VectorEDK toolkit, were all dumped online. According to its manual, the tool was designed to be used with physical access to a victim's computer.

Kaspersky says that based on the similarities between VectorEDK and the modified version used by the Chinese group, the Chinese group most likely deployed their tool using physical access to their targets' computers as well.

So... we don't have some new pernicious UEFI attack vector. It remains blessedly difficult to get malware down into our PC's motherboards. But we can also see that a sufficiently determined actor, especially one with State-sponsored control — and what contemporary motherboard doesn't have components source from China? — IS able to pull off such an attack.

## Errata

Ian Butterworth / @misterian
@SGgrc @leolaporte Correction required. Dr. John Campbell is not an MD. He's a retired nurse with a PhD, see his YouTube About. Yes his content is excellent, I've been watching him for some time.

## SpinRite

After my mention, last week, of GRC's forum availability we experienced a perfect level of influx of new members. Today we have more than 2800 members and I'm completely happy with the way it's proceeding. It's easier to handle a steady trickle than a stampede, with people coming in and saying Hi. And a very nice community is establishing itself there. I could not be more pleased.

A couple of days ago I added a "What I'm working on right now" tracking thread to my personal blog forum so that those who wanted to know what's going on with me could easily check, and by "Watching" that thread, or by establishing an RSS feed for it (by appending /index.rss to the URL), they could receive updates of that status.

Registration is still hidden from the non-podcast world. So those listening who are interested in grabbing their user ID of choice before I open the forum more widely are invited to sign up at:

**http://forums.grc.com/register/**

# Why Win7 Lives On

**Why are people sticking with Windows 7?**
An interesting bit of research by ZDNet caught my eye. ZDNet noticed that today, nearly 10% of their website visitors were still running Windows 7 more than 5 years after the release of Windows 10... despite Microsoft's extreme efforts to force everyone onto Windows 10.

Ed Bott, who grew up through the birth of the PC like the rest of us old timers, noticed their server logs and setup for the poll by writing this:

> Statistically speaking, one in 11 people reading this post on a PC are running Windows 7. That's not speculation or guesswork. That's what the ZDNet server logs for the last 90 days say. Some 85.8% of the many millions of PC-based visitors to this site are running Windows 10. Of the remainder, 9.2% are running Windows 7, which is twice as many as the Windows 8/8.1 population.
>
> That data lines up pretty closely to public data from the United States Government's massive Digital Analytics Program, which measures visits to more than 400 websites run by the Federal Government. Over the same three-month period, their mix of traffic from PCs consisted of 85.9% Windows 10, 10.0% Windows 7, and 3.7% Windows 8.x.
>
> That's a fairly significant drop from the last snapshot I looked at, which counted Windows 7's share of visitors at 18.9% in the 90 days leading up to that operating system's January 14, 2020 end of support deadline. (See "It's 2020: How many PCs are still running Windows 7?")
>
> The glass-half-full crowd says it's a good thing that half the population has stopped using Windows 7 in the nine months since Microsoft ended support for it. But I'm curious why so many are continuing to use Windows 7 past its expiration date.
>
> Rather than speculate, I put together a poll. Thanks to the more than 3,200 people who responded. I'll share the results next week.

And today we have those results shared in Ed Bott's story, posted yesterday on ZDNet:

> Our ZDNet poll drew more than 3200 replies, along with 50 or so emails. The results are fascinating. Let's start with the two easiest questions.
>
> **DO YOU PLAN TO UPGRADE TO WINDOWS 10 IN THE NEXT 12 MONTHS?**
>
> The answer to this question was pretty emphatic. Just under 58% replied No, with another 27% answering Not Sure. Only 16% said Yes.
>
> Several respondents pinned the blame for the slow upgrade on corporate IT departments, with two respondents saying that the Covid-19 response had caused issues with completing upgrades in their organization. Others pointed to IT departments that are "understaffed" and "incompetent" and "taking their time."

In all, roughly 1% of respondents specifically mentioned that they were prohibited from upgrading because they were using a company PC and not a personal device.

**ARE YOU PAYING FOR EXTENDED SUPPORT?**

Although Microsoft has stopped releasing monthly updates to the general public, those updates are still available for those who purchase Extended Security Updates (ESUs). They're not cheap, and they're not easy for small businesses to acquire, as I noted at the beginning of this year. [Ed earlier wrote a piece titled: "You want to keep running Windows 7? Good luck with that, small businesses."]

Perhaps that explains why only 6% of poll respondents said they're paying for ESUs, with another 3% admitting they're not sure. The remaining 91% are, apparently, doing without security updates.

In the longer responses, some people took pains to note that their Windows 7 machines aren't connected to the internet; others pointed out that they had up-to-date security software. And a few said they thought Microsoft was exaggerating the threat posed by running unsupported software in a bid to squeeze more money out of customers.

**WHAT IS THE MAIN REASON YOU HAVEN'T UPGRADED?**

Here's where things got interesting.

The original survey contained four choices and a box labeled other, where respondents could fill in their own answers. Nearly a thousand people chose "Other" and then wrote in their reason.

I read every one of those responses and categorized them manually. About 10% of the responses could not be categorized, because the reason was indecipherable or irrelevant. That left a total of 2,855 usable responses.

Here's how they broke down.

***Compatibility (42%)***
The number-one reason people are sticking with Windows 7? There's no contest. "Compatibility issues (hardware and software)" wins in a landslide. Fully 40% of respondents chose that answer, and another 2% or so selected "Other" and then identified a compatibility issue.

The specifics included some esoteric equipment, including one person with a legacy CNC milling machine, plenty of old peripherals that don't have Windows 10 drivers, and several people who paid for Adobe Creative Suite perpetual licenses and have no desire to upgrade.

Among the write-in responses the biggest group was made up of fans of Windows Media Center, who collectively added up to roughly 1.5% of respondents. Their loyalty is impressive.

***Don't want to upgrade (32%)***
About 17% of respondents chose the ready-made "Just don't feel like upgrading" answer from the poll form. But I counted nearly the same number of people who chose the Other box and then made it clear from their reply that they had picked Windows 7 over its logical successor, Windows 10.
I sorted those replies into four buckets, in the following order:

People who just don't like Windows 10 made up the biggest chunk of respondents. People called out the user interface, bugs, and stability, in particular.

About one in four of the "I don't like Windows 10" group used strong enough language that I created a separate "Windows 10 sucks" category. Many used that exact phrase, while others threw in overheated words like "garbage," "crap," dumpster fire," and a few choice phrases that I can't repeat here.

In all, those first two groups added up to about 10% of total responses.

A slightly smaller camp had no particular problem with Windows 10 but preferred Windows 7. Just under half of this group praised Windows 7 because "it just works."

A slightly larger group said they believe "Windows 7 is better than Windows 10." They praised the user interface ("much more user friendly," "the last usable version") and called out Windows 7 for its stability. A word that appeared over and over again was "control," especially in the context of security updates. (More on that in a minute.)

In all, it seems appropriate that the two groups of Windows 7 fans added up to about 7% of survey respondents.

***Upgrade is too expensive (10%)***

I was surprised that so many people chose this option, especially when the Windows 10 upgrade is still free. The most poignant example came from a reader in Iran, who said, "In Iran we have a bad situation" and the cost of the upgrade was too high.

***Updates are too intrusive (5%)***

An unsurprising number of people expressed their extreme displeasure with "forced updates," "buggy updates," and the "feature churn" with twice-yearly updates.

"I've never had to reinstall an OS due to a borked update," said one respondent. "That seems to be a regular occurrence with Win10."

Continuing the theme of loss of control, another person said, "I own my computer and decide when to update, not Microsoft."

***Privacy/telemetry/spying (3%)***

I was just a bit surprised that this number was so low, but what they lacked in numbers, this group made up for in ... well, let's call it passion, using multiple variations of the word "spying" as well as "privacy" to express their discontent for Microsoft's "telemetry," which is apparently a dirty word.

Another 1% or so specifically called out the word "trust" and a handful described their hatred for Microsoft using language that my editors would be extremely displeased to see me repeat on this website.

Perhaps the best response in this group was this one: "This is BIG BROTHER, brother."

***Afraid to upgrade, can't upgrade, too busy (3%)***

Not everyone who responded to the survey was dismissive of the idea of upgrading. A significant number of people said they were afraid to upgrade because they worried they would lose data or programs in the process.

This category also included people who said they were "too busy" to upgrade or that they couldn't afford the time to reinstall programs and reconfigure system preferences.

About a third of the responses in this group said their hardware was too old to upgrade or that they had tried and failed. But my favorite reply was a single word: "Laziness."

***Training/support issues (3%)***

Honestly, I expected this group of responses to be bigger, but it looks like most corporate customers aren't particularly worried about users being able to adapt to change.

***I'm moving to Linux (1%)***

And finally ... It just wouldn't feel right if I ignored the handful of respondents (24, to be exact) who said they either have switched to Linux or are just about to do so.

Looking over those numbers, I was surprised that 42% cited compatibility issues. Ed mentioned some old numerical control equipment whose drivers were unsupported by Windows 10. I think that we too often think of Window systems in the generic desktop workstation role. But my dentist's office is still using Windows 7 for its patient's records management and the app that runs the digital X-Ray machine and displays the digital X-Ray images is hosted on Win7. Who knows whether that will run on Win10? ... and I'm sure they're not wondering. It's a case of "if it's not broke, don't fix it." So even if compatibility were not an issue, that could also fall into the 32% who just don't want to upgrade. One advantage of using Windows 7 for non-desktop applications like that is that Microsoft cannot force the system to upgrade "or else." I certainly do find much to agree about with those users who say "this is MY computer, darn-it. I'll decide what software it runs, and when." Someday that attitude will be considered quaint.

Today I use both Win7 and Win10 every day — a different workstation at each of my two sites. And I know what's going to happen. Eventually Win7 will become too old and things I want will not be available for it. That's what happened with XP. Both Mozilla and Google refused to update their browsers on that perfectly functioning OS... and IE had died on it ages ago. And just last night I wanted to install the "Calm Radio" app for background ambient music while I'm working. The Win7 app is nowhere to be found. Now it's Win8 and 10 only. So I ran it under iOS.

At some point I'll rebuild my Win7 machine with Win10. But for now I have Windows Defender working well and updating. And all of my browsers are still current and updating. So I'm good with Win7 for the time being.