Why Win7 Lives On

Description: 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 Subsystem for Linux. We check in on Exchange Server updates after eight months. We cover Cloudflare's announcement of a very welcome Web API firewall, the U.S. 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. We conclude by sharing the results of an interesting poll which illuminates the many reasons why Windows 7 refuses to die.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-787.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-787-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have lots to talk about: why the new Windows Subsystem for Linux, WSL in Windows 10, bypasses the Windows Firewall, and what you can do about it; Microsoft changing some of the ways it's doing updating, in vitro updating for Defender. We'll talk about Cloudflare's new API protection. And then why is it some people are still using Windows 7? Steve takes a look at Ed Bott's poll of users. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 787, recorded Tuesday, October 6, 2020: Why Windows 7 Lives On.

It's time for Security Now!, the show where we cover your privacy, your security with this guy right here, Mr. Steve Gibson of the Gibson Research Corporation, creator of SpinRite and our leader into the world of the dark side of the Internet. Hello, Steve.

Steve Gibson: Yo, Leo.

Leo: How are you today?

Steve: Great to be with you. Episode 787 for, yeah, October 6th.

Leo: We're in the airliner series.

Steve: It's happening, baby.

Leo: Yeah, yeah.

Steve: So there was a bunch of interesting news. Nothing really stuck out except Ed Bott took a poll a couple weeks ago over at ZDNet when they were noticing what a large percentage of their visitors were still coming to their site with Windows 7. So we thought, what? Let's ask people why. And I thought that would just be interesting. Certainly it has a bearing on security, as we know, because the updates terminated at the beginning of this year. And almost 10% of ZDNet's current web visitors are still using Windows 7. So the title of this week's podcast is "Why Win7 Lives On," which I think will be interesting to our listeners. And I was a little surprised until I thought about it, and I thought, oh, okay, yeah, that makes sense. So with that tease, we will get to that.

But first we're going to examine several new and welcome Google initiatives aimed at improving Android sort of in general, and also everybody's web browser security. We're going to look at Microsoft's solution, which is kind of really interesting and cool, for updating aging offline Windows images with the latest Defender definitions. We're going to look at some surprising network behavior from Windows' second Subsystem for Linux, which is where we are now, WSL 2. Their re-architecting of it had a side effect that is catching some people off guard. We're also going to check in on Exchange Server updates eight months downstream of the February patch, and it probably won't surprise any of our listeners to learn that they have not all been updated yet.

We're also going to cover Cloudflare's announcement of a very welcome new free Web API firewall. Also the U.S. Treasury has decided to update their policies regarding ransomware payments, and no one's happy about that. Also, and this was probably the most tweeted thing to me over the past week, Kaspersky's discovery of the use of another UEFI Bootkit, which always makes all of our listeners nervous. We've got a little bit of errata from last week. I'm going to update our listeners about the GRC forums. And then we're going to take a look at the interesting results of this poll that ZDNet conducted - I think it was 3,600 respondents, so a nice chunky big sample - and learn why Windows 7 refuses to die. It's not surprising, but some things were.

Leo: It's tempting to assume it's kind of stupidity of some kind, or they're just old and don't want to change things. But I bet, well, I haven't read it yet, so I'm curious. But I bet it's some better reason than that, I would hope. We shouldn't judge.

Steve: Indeed. And we do have a programmer's keyboard to show everybody for our Picture of the Week.

Leo: It's probably all Escape keys. I haven't looked at it yet, though. So we'll see. Oh, I love it. This is a good Picture of the Week. I like it.

Steve: I had seen it once before, and it struck me as just fun. So this is, for those who don't have video, this is showing us the SuperCoder 2000, which it has the subtitle "Air cooled coding keyboard for professional use." And because this is about programming, it has exactly three buttons: a "0," a "1," and a "Done."

Leo: That's all you need.

Steve: Because one presumes that the fingers will be flying, it's got very prominent air vents to allow it to air cool because that keyboard would tend to overheat otherwise from all the zeroes and ones being pumped into it. And we're sort of used to 3D models and rendering these days. This thing looks real. I mean, I think it's old.

Leo: Somebody built this for sure. Somebody built this for sure.

Steve: Yeah, yeah. So that's like, okay, good.

Leo: I'm not saying somebody used it. But they definitely built it.

Steve: Yeah. You can enter a "0" or "1," and then you can declare when you're done. So basically that sums it up.

So Google is going to be getting even more proactive about Android security. We know that Google runs their own bug bounty program for Android apps which are those listed in the Play Store, the ones that they're curating. That program is called the Google Play Security Reward Program, not a very catchy abbreviation, GPSRP, Google Play Security Reward Program. And through that program independent security researchers are rewarded for locating and responsibly reporting bugs they find in Play Store Android apps. So Google is in essence paying for bug submissions on behalf of its platform apps' publishers.

But one limitation of the GPSRP is that it's restricted to major apps having more than 100 million users. So I guess Google is saying, look, guys, there's a lot of apps there, and don't go finding some bug in an obscure app that 12 people have downloaded. While we would like to have that fixed, we're not that concerned about it. You're not going to get a bounty for that. So the GPSRP, 100 million users minimum. But Google noticed that other apps that handle sensitive data may be performing critical tasks and don't meet that 100 million threshold, which would make them ineligible for GPSRP rewards and are therefore less likely to be tested by external bug hunters.

So what happened was some sharp-eyed person noted a job posting that went online last Wednesday in which 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."

And according to Sebastian Porst, who is the Software Engineering Manager for Google Play Protect, this new team will be focused on apps such as COVID-19 contact tracing and election-related applications, with others to follow. In their coverage of this event, ZDNet quoted Lukas Stefanko, a mobile malware analyst at security firm ESET. Lukas 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. So to Google's credit, having a dedicated team ensures that some of the world's best security talent will put their focus and effort into looking at apps that might otherwise slip under the radar and could end up being exploited with serious consequences.

So anyway, just Google again doing good things for the community. And we see a lot of that from our view of Google. I know that it's controversial. There was some brouhaha recently that erupted in some dialogue over on GRC's forums because I'm using ReCAPTCHA, which of course is Google's CAPTCHA anti-bot solution, and people were

objecting to the fact that Google is Google. But from our perspective here, Google's doing a lot of work to keep things secure. And in fact they are also going to be funding a JavaScript research engine, or essentially offering grants for people. They've established a new grant program to fund research aimed at locating bugs in the industry's browsers' JavaScript engines. The eligible targets are Google's Chrome V8 engine, Firefox's SpiderMonkey, and the Safari JavaScriptCore, each which are the JavaScript processing engines for those browsers. And they did indicate that other engines could be pitched in a grant proposal, but we'll see in a minute why those three make the most sense.

So this program will help and sponsor security researchers and academics to find vulnerabilities hidden inside any of those three JavaScript engines, and the program only has one rule. The bugs must be identified through fuzzing. And we've talked about the virtues and value and power of fuzzing many times in the past. However, since fuzzing tends to be an extremely resource-intensive process, it's traditionally been the province of larger tech organizations that can afford the resources required to set up comprehensive fuzzing operations.

To be done at scale, it typically employs large and expensive cloud computing, resources that are typically beyond the reach of most solo researchers working on their own in the hope of finding a significant bug. And of course the cloud resources need to be paid for, whether or not there's any follow-on bug bounty payoff. And even when bounties are paid, it can be months later, if at all. So in their blog post last Thursday, Google said it created this research grant to address exactly these problems. Under Google's new pilot program, which will have a one-year duration, October 1st last Thursday to October 1st of 2021, security researchers and academics can apply for the funds required to fuzz any of those three JavaScript engines. Google said it will analyze each grant proposal submission and provide an answer to all applicants within two weeks, with approved projects receiving up to $5,000 in funding.

Now, naturally, since Google is a provider of exactly such cloud services themselves, might as well keep the money in the family. The grant funding will actually be up to $5,000 in credits to be used on Google's Compute Engine, which is their heavy computing infrastructure in the cloud. So it runs, as I said, from October 1st, 2020 through 2021. And they're calling the program the Fuzzilli Research Grant, or Fuzzilli - obviously a play on a pasta; right? - F-U-Z-Z-I-L-L-I, named after Google's own Fuzzilli open source fuzzing tool, which supports distributed fuzzing on the Google Compute Engine, and of course Google naturally encourages grant recipients to use. I guess the $5,000 cloud computing grant credit wouldn't do you any good unless you were going to use it on Google's Compute Engine, which is the only place you can use it.

Although Google said that all bugs identified during the pilot program must be reported to the affected vendors, researchers may retain any bug bounty payoffs themselves for any bugs they might find during the program. Samuel Gross, the creator of Fuzzilli and a member of Google's Project Zero team, said: "JavaScript engine security continues to be critical for user safety, as demonstrated by recent in-the-wild zero-day exploits abusing vulnerabilities in Chrome's V8 JavaScript engine."

So I think this is neat. If, for example, any of our listeners might have some time on their hands during the COVID mess, or just during spare time on evenings and weekends, you could figure out how Fuzzilli works. It's all posted. It's open source. It's there on GitHub. Play with it on your own machine to work out the details and come up with an idea, then apply to Google for a $5,000 grant to scale its use up to big cloud iron level. And if you were to get lucky with fuzzing, essentially for free, using Google's cloud infrastructure, you might uncover a previously unknown and valuable flaw. And then you could score yourself some cash without incurring any other out-of-pocket expenses.

I've got the various links to the announcement and to its presence on GitHub for anyone who's interested. Samuel's blog posting about this added, he said: "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 two weeks. If the project is accepted, the researchers may be awarded GCE (Google's Compute Engine) credits worth up to $5,000.

"Researchers can also apply for multiple grants throughout the lifetime of the project. The grants come with the following requirements," he wrote. "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 for Safari, V8 for Chrome and Edge" - and remember any other Chromium-based browsers - "or SpiderMonkey used by Firefox." And when I went over to take a look at and remind myself about Fuzzilli, because we had talked about it before, I saw that it was those three engines for which there is already prepared interfacing open source glue, essentially. So using those three would be far easier with Fuzzilli, although you could certainly do something else if you wanted to bite that off and then propose that to Google.

Samuel said: "All vulnerabilities found must be only reported to the affected vendor." Meaning responsibly and privately. "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 the following requirements." He said: "Any newly developed source code must be published under an open source license that permits further research by others." So, for example, if you did bring up support for some different browser's JavaScript engine, then that needs to be part of your contribution to the open source community.

He said: "An interim report for Google only at the conclusion of the fuzzing, to demonstrate the initial results of the research, so that we can determine the efficacy of the research and make our folks in accounting happy." So justify the $5,000 that got burned somehow. "Furthermore, a final report of some form - for example, a conference paper, a blog post, or a standalone PDF - due no later than six 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, et cetera), and a clear technical explanation of all vulnerabilities discovered throughout the project."

And he concludes: "Researchers are encouraged to base their project on the open source" - oh, here, we were just talking about this - "the Fuzzilli fuzzer, if possible." You know, he's the guy who wrote it, so yeah, he'd like to see it used also, which is ready to go on Google's Compute Engine. So again, a very cool opportunity. If there are any of our listeners who think this might apply to them, I would suggest you think about it. Grab Fuzzilli, play with it, figure it out, see if it makes sense, and then go for a grant. You could end up having a CVE to your name, and maybe a bug bounty payment.

This I thought was really interesting. And props to Microsoft. Microsoft is giving Defender what I described as "in vitro updating." Many enterprise environments used a fixed image of Windows, which they downloaded from Microsoft at some point, to set up their new workstations. And these images are often used for many months at a time. We know that enterprise especially wants to hold onto a given install image as long as they can. The last thing they want to be doing is spreading, "smearing," to use the word I have before, various major feature builds of Windows 10 across their enterprise. It really behooves them to keep them synced.

So as a consequence, a static image of some instance of Windows 10 would be used for a long time. And, you know, there's really no point in updating to a newer download of the same image, since it's just going to be the same download. Microsoft isn't changing those over time. But that's the problem. The trouble is that these images are aging, and so is their built-in Windows Defender definitions, which were frozen at the time that that image was created, making Windows Defender, as is built in, 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 AV 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 in the Windows world since that original image was downloaded and frozen, then those are updated and the system is rebooted, then yes, it, too, will have caught up and had the latest and greatest. But the problem in Microsoft's words is that that still leaves a potentially significant "protection gap," which is the term they used, during which a machine that's booted up and running and on the network could have some seriously outdated Defender definitions until it can update itself.

So 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 AV awareness current before it's being used to create any running systems. I've got their annoying links with a link ID. I've got them both in the show notes, for anyone who's interested, at the bottom of page 3. Those links point to ZIP files, each of which contains two files: an updated and continually updated Windows Defender .CAB file and a PowerShell script which is named "defenderupdatewinimage.ps1."

There's also a link to an upper level announcement, Microsoft Defender Update for Windows Operating System Installation Images in the show notes that would take you to a page explaining how to use those ZIPs and to deploy. But I think it's very cool. It allows you to take a standard Windows install and, in vitro, insert the latest Defender into it so that the moment it comes up and boots, even before it's had a chance to update everything else, it is up to the minute updated with Defender things. And as we know, especially this year, that's become more important than ever because there have been lots of critical things which, I mean, it's what we've been spending a lot of time recently talking about is that people are not applying these updates in as timely a fashion as necessary, and there are threats in the wild that are taking advantage of these. So again, props to Microsoft for getting ahead of this.

I mentioned at the top a change that had been made in Windows Subsystem for Linux v2. It turns out that it quietly completely bypasses its hosting machine's Windows 10 Firewall, which is new behavior. And so I wanted to make sure that none of our listeners would get caught out by this. The first version of Windows Subsystem for Linux, WSL 1, was implemented using a Linux-emulating pseudo-kernel that translated Linux system calls into their equivalent WinNT kernel calls. So under WSL 1, any network traffic would actually be coming from Windows, that is, any Linux-sourced network traffic would actually be coming from Windows and is thereby filtered through the standard built-in Windows application firewall. The Linux distro honors any configured rules because it's behind that firewall because it's also actually behind the Windows kernel.

But with the second release, the second iteration, and it was a huge change, WSL 2 does everything differently. In WSL 2, Microsoft produced a true Linux kernel operating side by side with Windows in a Hyper-V virtual machine and a Hyper-V virtual network adapter. As a consequence of this complete re-architecting, it's a completely different architecture. So unlike with WSL 1, WSL 2 traffic is sent directly to the virtual network adapter, completely bypassing the Windows Firewall. So this is not in itself a bad thing. I would argue it's way more powerful, and it's cleaner, and that this architecture is correct.

It means you've got Linux and Windows side by side, rather than this first pass kludge that they created. But it does mean that an unwary user might mistakenly assume that they had the same merged Windows Firewall protection after updating themselves and their system to WSL 2, and that's not the case. So they could be inadvertently left exposed.

So anyway, just beware that there is no Windows Firewall protecting WSL 2 instances of Linux. It is out there on the raw network adapter, and so subject to any incoming that the Windows application firewall before was protecting. So of course Linux is where the Internet, well, Unix actually, was where the Internet was born. And so it's got firewalls and filters and network protection galore. But you need to turn it on, if that's what you're expecting to have.

And speaking of things not being patched for quite some length of time, we talked about this at the time. On February 11th, 2020, Microsoft released security updates to address a vulnerability in Microsoft Exchange Server 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 Active Directory itself, giving bad guys complete access into a system. That was February 11th, which was Patch Tuesday in February. A month later, any admins who were paying attention would have subsequently learned that any still unpatched servers were then being exploited in the wild by unnamed APT (Advanced Persistent Threat) actors.

So what's the story today, eight months from the time that the patch was released? Somewhat unbelievably, Rapid7 conducted a survey on September 21st and found that out of 433,464 open and Internet-facing Exchange Servers - and of course that's what Exchange Server is supposed to be; right? It's email. It's going to be out there on the public Internet. 433,464 could be observed on the Internet. On September 21st, at least 61% of Exchange 2010, 2013, 2016, and 2019 servers have remained vulnerable to the flaw, despite their having been offered an update. And this flaw, this vulnerability is not theoretical, like Spectre and Meltdown. This is known to be exploited in the wild.

You know, we moan and we groan and we feel rightly sorry when large enterprises are struck, for example, 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 - and this is the count - 267,986 individual Exchange Servers still 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 to be asking for it. And any enterprise that - and there's 267,000, almost 268,000 individual Exchange Servers, eight months from the patch availability, still exploitable. It's just crazy.

And although it isn't a simple vulnerability like a problem with RDP, where you can simply get in without authentication, it's clearly and rightfully a privilege escalation or elevation. You need to have an Exchange account. But it's a perfect example of the idea of someone getting in, doing some reconnaissance, watching the system that they're on, looking at traffic. It would not be difficult to do keystroke logging and capture Exchange credentials or monitor a workstation's traffic and get credentials. As soon as you have any credential on that Exchange Server, then it can be used to potentially get you access to that network's Active Directory. And as we were just talking about last week, that's the keys to the kingdom, then essentially giving you access to much more on that network, allowing a bad guy then to move laterally and get to all the machines there.

So again, it's like, how can it be? I mean, these must just be completely administratively abandoned Exchange Servers, or people who aren't paying attention to the fact. I mean,

I guess for our listeners there's just nothing could be more clear today than that with the use of software comes the need to keep it current. It is an obvious truth that is constantly being reinforced by what we see now every day.

Our friends at Cloudflare just added a free Web API Firewall service for all of their customers.

Leo: Oh.

Steve: Yeah, it's very cool. Last Thursday they announced their new API Shield. I've got a link to the blog posting which announces it in the show notes. They start out by explaining 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 said: "They're also the target of widespread attacks designed to perform unauthorized actions or exfiltrate data, as data from Gartner increasingly shows." Quoting Gartner, they said: "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 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.

So just sort of back up a little bit. A perfect example of a Web API, just to kind of give our listeners some context, that we've talked about not too long ago, was the SSP API. That was the 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 how SQRL works. And I realized that this would also be a general boon for SQRL's server-side adopters.

So I hid and encapsulated all of SQRL's various details behind a very simple abstract authentication API. It was accessible through a simple HTTP query and reply. And in that way Rasmus could simply post HTTP queries using any web agent, in this case PHP, and receive meaningful replies. And this is the model that's evolving for the entire industry. It allows for the creation of elegant and clean interfaces. And in the SQRL use case, I was able to make the interface private, since it only needed to support transactions between Rasmus's PHP code and mine, both which resided on the same server, or also on the same 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.

So like with the web and communications, what we're now beginning to see is, once upon a time, it was a browser being viewed by a human that was pulling content from a web server. Increasingly, it's, well, I mean, a perfect example I was talking about just recently, when I added IoT gizmos. I have now an IoT thermostat which is setting the temperature for the room I'm in, and an IoT hygrometer which is measuring temperature and humidity. Those are definitely communicating, using Web APIs, to their cloud providers in China. And while I'm happy to have them do that, it is the potential abuse of those which would open an opportunity for them to be exploited on my network, which again is why they are sequestered on a network where, if anything were to get loose, it couldn't damage the rest of my network.

Okay. So Cloudflare continued their announcement by saying: "To combat these threats, Cloudflare is making it simple to secure APIs through the use of strong client certificate-

based identity" - which is, I'll just say, very cool - "and," they said, "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." So I'll just note that they've implemented a proper "deny all" model that only permits valid formatted API calls to pass.

So they explained: "A positive security" - which is what they called it. "A positive security model is one that allows only known behavior and identities, while rejecting everything else. It's the opposite of the traditional negative security model enforced by a Web Application Firewall that allows everything except for requests coming from problematic IPs, ASNs, countries, or requests with problematic signatures like SQL injection attempts."

They said: "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 reuse or sharing of passwords." And I'll just note that, for example, once Cloudflare deploys this, once people with Web APIs enable that and able to enable it, then for example Shodan would no longer see any of this. That would just go dark. You would not be discoverable or listable by Shodan. And of course that's what you want. You don't want to be seen. And it's not because you've chosen to change your port and you're using an unknown port, which can still be discovered. It's because you're saying we're going to do mutual TLS authentication and have both ends affirmatively acknowledged.

So they said - this is Cloudflare speaking. "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."

So in other words, they're making it trivial and free for users of their services to issue their own client certificates, which would then be known to their own CA. So then you put this certificate in your devices that you want to authenticate to the Web API which Cloudflare is hosting. When that certificate is presented as part of the TLS handshake, it will be verified. It'll be tied to your account, to your specific API permissions, and it will have been signed by their CA, so it will be valid.

They said: "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."

In other words, they recognize that, yes, creating client-based certificates is strong security, but it's possible in theory to extract the certificate. For example, if it was my residential thermostat making certificate authenticated queries, well, it's sitting, you know, in people's homes. You can buy one from Amazon. It's very much like we talked a long time ago about how it's impossible to actually encrypt DVDs that cannot have the decryption keys discovered because the decryption keys have to be in the DVD player that the consumer has. Which means you can't hide them perfectly.

Anyway, so Cloudflare of course recognizes this. They say: "Requests containing extraneous input may not have been anticipated by the API developer and can cause problems if processed directly by the app. 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 any invalid request or any malicious payload."

They said: "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." So this is way cool. Their announcement continues with a demonstration and examples of API schema definitions that explicitly define valid query formats 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. And note, this is exactly the kind of thing that you would like to have in front of your potentially vulnerable API backend. Fuzzing on the outside would simply be blocked by this firewall because fuzzing is all about throwing unexpected weird stuff at an API. And when it crashes, you go, "Ooh," and then that gives you a place to look to figure out how and why it crashed and then go from there. But with something like this, which is checking any inbound arguments against the schema that the developer defined, even if their own API doesn't enforce it, this drops those things, as Cloudflare said, "at the edge." They go no further. They are just ignored. And you don't even look at the bandwidth of all that stuff coming in. You just don't see it any longer.

So the more I think about what Cloudflare is doing, the more I think I may have them in my own future. As we know, I'm still maintaining a full-height rack of physical servers and network equipment in a nearby Level 3 datacenter. And I'm reminded that three years ago, while I was 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, and with sort of a look of puzzled brow-furrowing.

Leo: Yeah, look at the caveman over there.

Steve: And one of them said - yeah, exactly. And actually, I was the oldster in the group. And one of them said, "What? No one does hardware anymore."

Leo: Yeah.

Steve: And I said, "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 Level 3, but before I'm ready to abandon GRC.com - I mean, we're all getting older, Leo - I'll likely move it to the cloud. And Cloudflare would probably be my first choice for GRC's final home. And it would just make sense. So anyway, these guys just keep doing good things and offering very useful services, just a palette of more and more useful services.

Leo: It's impressive, yeah.

Steve: So the U.S. Department of the Treasury has tightened up on ransomware payments. I'm not sure how I feel about this. And I understand it, but it's easy for them

to do. Last Thursday the U.S. Treasure Department issued revised guidelines which, among other things, are targeting this new phenomenon of ransomware negotiators, and actually also their insurers. I've got a link to their sort of snapshot policy, and also I think it was a five-page PDF.

So in a little snapshot they said: "The U.S. Department of the Treasury's Office of Foreign Assets Control (OFAC)" - which also sounds a little bit like it looks.

Leo: OFAC, yeah.

Steve: Anyway, OFAC. OFAC. Anyway, they're issuing an advisory to alert companies that engage with victims of ransomware attacks, right, so they're alerting the companies that engage with victims of ransomware attacks, so that would be the negotiators and the insurers, of the potential sanctions risks for facilitating ransomware payments.

They said: "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."

So the five-page PDF goes into all the details. But effectively, the Treasury Department wants to be notified of ransomware attacks because large sums of money are crossing U.S. borders and potentially moving into the hands of sanctioned individuals, teams, or governments. And the presumption is that - I got a kick out of this - 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." In other words, you're supposed to get a licensed application to make such payment, and OFAC will probably say no.

So this really puts victims, their negotiators, and their insurers in an awkward hot seat position, especially when circumstances might strongly compel the victim to want to pay the ransom and to take advantage of various intermediaries and insurers. So this whole thing is rapidly becoming a hot mess, as people now say these days.

In what was probably the most tweeted event of the week, UEFI Bootkits - you know, I was tempted to say they're becoming more mainstream. But I don't think they are. The very first UEFI Bootkit was spotted by ESET just over two years ago. We covered it here at the time. Their posting at their site, WeLiveSecurity.com, was titled "LoJax: First UEFI rootkit found in the wild, courtesy of the Sednit Group." And if those words and terms sound familiar, it's because, yeah, two years ago we talked about it.

Two years ago ESET's write-up started off by explaining this a little bit. They said: "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 [ESET] discovered a campaign by the Sednit APT group that successfully deployed a malicious UEFI module onto a victim's system."

So that was two years ago. Today, UEFI rootkits 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 that 30-page PDF is not a fluff piece. It drops right into the technology that they found and reverse engineered. So what we know, thanks to Kaspersky's research, 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 remains difficult. Attackers still need either physical access to the machine, or they need to compromise the target through some sort of complex supply chain attack where the UEFI firmware or tools that work with UEFI firmware are modified to insert malicious code. So again, not trivial. This is not some email worm. Kaspersky's malware researchers said they investigated some suspicious systems and discovered malicious code inside those systems' UEFI firmware.

The code, and I thought this was kind of cool, was designed to install or really copy, and reinstall or recopy, a malicious app as a Windows autorun program after every computer restart. So even if the program was found and removed, at the next system boot it would reappear. It would go into the auto start directory and autorun when Windows was started. So that autorun program acts as a downloader for some other malware components which Kaspersky named the MosaicRegressor malware framework. They said that they had not yet obtained and analyzed all of MosaicRegressor's components.

But the one they did look at contained functionality to gather all the documents from the recent documents folder, put them into a password-protected archive, and presumably prepare the files for exfiltration via some other component. Of course it's typically not difficult to get something out of a system that's been compromised. Especially if you are password-protecting an archive, and it's a well-designed archiver, then it'll be, as we know, maximum entropy noise, not subject to being discovered by anything that might be scanning it on the way out. And if you had time, you could even send it out in little ping echo requests, a little bit at a time, because the payload of a ping can be, I think it's up to a Kbit, or maybe it's a Kbyte, of stuff.

Anyway, the researchers found the UEFI bootkit on only two physical systems, but they had found this MosaicRegressor component on many other computers. Kaspersky believed that the targets of these attacks were all very carefully selected. They were all diplomatic entities and NGOs located in Africa, Asia, and Europe. So that's certainly not random.

They 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 Kaspersky 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 Hacking Team.

And I remember when this happened, we talked about it at the time because we joked that "Hacking Team was hacked." That's what happened. Hacking Team were themselves hacked back in 2015; and their tools, including this VectorEDK toolkit, were all dumped online. According to VectorEDK's manual, the tool was designed to be used with physical access to a victim's computer. So that happened somehow for these well-placed specific PCs that had this rootkit EDK installed. 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 same modus.

So we don't have some new pernicious UEFI attack vector. It remains blessedly difficult to get malware down into our PC motherboards. But we can also see that a sufficiently determined actor, especially one with state-sponsored control - and for that matter, what contemporary motherboard doesn't have some components sourced from China - is able to pull off such an attack. So I got a lot of tweets about this, like uh-oh, another UEFI rootkit. It's like, yeah. But, boy, you know, it's nothing that is going to get any of us. It's the kind of thing that really high-end elite systems are going to have installed on them. And it requires a lot of planning and forethought to make that happen.

I have a piece of errata which appeared in the GRC newsgroups; and I looked for it again, and I couldn't find it. But fortunately, Ian Butterworth tweeted to me, he said: "Correction required. Dr. John Campbell is not an M.D. doctor. He's a retired nurse with a Ph.D." Then he says: "See his YouTube About." And Ian said: "Yes, his content is excellent. I've been watching him for some time." So I did want to correct the record. We mentioned John Campbell last week with that YouTube piece where he was talking about - it's slipping my mind - the Vitamin D content, or the amount of Vitamin D that Dr. - I know it very well, I can't...

Leo: Fauci? Fauci?

Steve: Fauci, yes, thank you. Dr. Fauci's Vitamin D consumption. Thank you, Leo.

Leo: Fauci?

Steve: Anyway, so John Campbell knows his stuff. It's clear from watching his YouTube videos. But not medical doctor, Ph.D. doctor. So thank you, everybody, for the correction.

After my mention last week of GRC's forum availability, we experienced just a perfect level of influx of new members. Today we have more than 2,800 members, and I'm completely happy with the way this is proceeding. It's easier to handle a steady trickle than a stampede, which is what I was hoping to prevent when I actually had something available for people to download and play with and then report on. So people coming in and saying hi and getting to know the place works well. And a very nice community is establishing itself there. So 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 at the moment could easily check to see where I am along the way between here and having SpinRite 6.1 released. So by watching that thread in my blog forum, or as we mentioned also by establishing our RSS feed, you just add a /index.rss to the end of the URL, and then you'll get updates. So registration on the forum is still hidden from the non-podcast world, so those listening who are interested in grabbing their userID of choice before I open the forum more widely are still invited to sign up. It's forums.grc.com/register.

So. Why are people sticking with Windows 7?

Leo: This I've got to hear. I am baffled.

Steve: Yeah. An interesting bit of research by ZDNet caught my eye. ZDNet noticed that today, these days, nearly 10% of their website visitors were still running Windows 7,

more than five years after the release of Windows 10 and, as we all know, 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 - you and I, Leo, know Ed's name well.

Leo: Oh, he's on TWiT all the time, was just on a couple weeks ago. We love him.

Steve: Yup, and he's been in the industry forever.

Leo: Oh, yeah.

Steve: Noticed that their server logs were showing this and set up the poll. So the poll setup read: "Statistically speaking" - this is Ed - "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 and 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" - that's virtually the same - "Windows 10, 10.0 Windows 7, and 3.7 Windows 8. 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 July 14th, 2020, end of support deadline." And he references an article from then.

So, he says: "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." He says: "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" - oh, it's 3,200, not 38, as I said. "Thanks to the more than 3,200 people who responded. I'll share the results next week."

Well, that was a week ago. Now we have the results. He posted yesterday on ZDNet. He said: "Our ZDNet poll drew more than 3,200 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.'"

He says: "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 within their organization. Others pointed to IT departments that are 'understaffed' and 'incompetent' and 'taking their time.'"

Leo: It's [laughing] rude to say, well, that's because I have an incompetent IT department.

Steve: They're not that good. We all wish we had Windows 10, but we're still stuck with [crosstalk].

Leo: That's so rude. So rude.

Steve: He said: "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." Don't you touch that. It's working just fine.

Okay. Next question: "Are you paying for extended support?" That is, has your Windows 7 support not actually ended? He says: "Although Microsoft has stopped releasing monthly updates to the general public, those updates are still available for those" - which bugs me - "for those who purchase Extended Security Updates (ESUs)." He says: "They're not cheap," and they get progressively more expensive, as we know, every year. "And they're not easy for small businesses to acquire," he says, "as I noted at the beginning of this year." Because earlier he wrote a piece titled "You want to keep running Windows 7? Good luck with that, small businesses."

Anyway, he says: "Perhaps that explains why only 6% of poll responders said they are paying for ESUs, with another 3% admitting they're not sure. The remaining 91% are, apparently, going without." And that's the camp I'm in. I'm going without. I'm sitting in front of Windows 7 right now. Works great. He said: "In the longer responses, some people took pains to note that their Windows 7 machines aren't connected to the Internet." Okay. "Others pointed out that they had up-to-date security software." I guess. "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." Fair enough.

Okay. So big third question: "What is the main reason you have not upgraded?" Ed says: "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." He says: "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."

So out of a universe of 2,855 usable responses, here's how they broke down. And this I found really interesting: 42% said compatibility. "The number one reason people are sticking with Windows?" Ed says: "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 include some esoteric equipment, including one person with a legacy CNC milling machine" - Computerized Numerical Control 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." He said: "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." He said: "Their loyalty is impressive."

Okay. 42% compatibility, number one. Number two, a third, 32%, don't want to upgrade. He said: "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..."

Leo: You influenced a lot of that previous group. You know, Steve Gibson says it's a polished turd, so that's why. That's all you need to know. Sorry, I didn't meant to interrupt. Then there's another group.

Steve: Yeah. "Just under half this group praised Windows 7 because 'it just works.'"

Leo: Yeah, or not.

Steve: "A slightly larger group said they believe Windows 7 is better than Windows 10. They praised the user interface - 'much more user friendly,' or 'the last usable version.'"

Leo: You know, all of that's legit. It's not that it's not legit; right?

Steve: Yeah.

Leo: It's just that it's insecure.

Steve: Yeah. Works great. "And called out Windows 7 for its stability." I mean, look at what - this year has been about Windows 10 disaster after disaster after disaster.

Leo: Oh, it's a nightmare. Yeah.

Steve: My printer, I can't print anymore. I just lost all of my programs and documents. They're all gone. What? Anyway.

Leo: Now, did you feel like that's worse with 10 than it was with 7? It feels like it, but I can't - I don't remember specifically; you know?

Steve: Oh, yeah.

Leo: It seems worse, huh.

Steve: Oh, yeah. I don't remember that with 7. 7 was mostly sort of a refrosted XP.

Leo: Right, right.

Steve: They weren't really trying to do that much to it.

Leo: Maybe that's it, yeah.

Steve: They've really gone crazy with Windows 10. Anyway, so he said: "A word that appeared over and over again was 'control,' especially in the context of security updates." He says: "More on that in a minute." He says: "In all, it seems appropriate that the two groups of Windows 7 fans added up to about 7% of the survey respondents."

And then, in something curious, 10% said "Upgrade is too expensive." And Ed wrote: "I was surprised that so many people chose that option, especially when the Windows 10 upgrade is 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.'" Okay. 5% said "Updates are too intrusive." He said: "An unsurprising number of people expressed their extreme displeasure with forced updates, buggy updates, and the feature churn with twice-yearly feature 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 I will decide when to update, not Microsoft.'"

And then, down at 3% was privacy, telemetry, and spying. He says: "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 just 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." And he said: "Perhaps the best response in this group was this one: 'This is BIG BROTHER, brother.'"

Finally, 3% were "afraid to upgrade, can't upgrade, too busy." Ed wrote: "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 just '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" - that would be down at 1% - "said their hardware was too old to upgrade, or that they had tried and failed. But my favorite reply was a single word, just in quotes: 'Laziness.'" Okay.

Leo: Fair enough. Fair enough.

Steve: 3% had training and support issues. He said: "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." And, finally, the least frequently cited reason, at 1% - okay, so what would that be? That would be shy of 300 of the respondents said, "I'm moving to Linux." Ed said: "And finally, it just wouldn't feel right if I ignored the handful of respondents" - oops, no. So less than 1%. He said: "24, to be exact, who said they either have switched to Linux or are just about to do so."

So anyway, looking over those numbers, I was surprised that 42% cited compatibility issues. Ed mentioned some old CNC equipment whose drivers were unsupported by Windows 10. And I think that probably we think too often of Windows systems in the

generic desktop role, desktop workstation role. But for example, my dentist's office is still using Windows 7 for its patient records management. And the app that runs the digital X-ray machine and displays the digital X-ray images I note is hosted on Windows 7. Who knows whether that would run on Windows 10? And I'm sure they're not wondering. It's a case of, if it's not broke, don't fix. So even if compatibility were not an issue, it could also fall into the 32% who just don't want to upgrade. It's like, it's running just fine.

And one advantage of using Windows 7 for non-desktop applications like that, X-ray machine, 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. So anyway, I guess someday that attitude will be considered quaint. And I should just note, I'm now freely using two machines. Today I use Windows 7 and Windows 10 every day, a different workstation at each of my two sites. I'm sitting in front of a Windows 7 machine here. In the evenings I use Windows 10.

And what I think will eventually happen is that Windows 7 will become too old, and things that I want will not be available for it. That's what happened with XP. Both Mozilla and Google had refused to continue updating their browsers just because I was still using Windows XP. Otherwise, it was a perfectly functioning OS. And of course IE had died on it ages ago. And as it happens, just last night I wanted to install the Calm Radio app for background ambient music while I'm working. The Windows 7 app is nowhere to be found. Now it's Windows 8 and 10 only. So as it happens, I ran it on iOS, which worked just fine.

But I'm sure at some point I will rebuild this Windows 7 machine when it just becomes too old, when the things I want to do with it are no longer available on it. And then I'll move to Windows 10. And I have experience with it now. I'm using it in the evenings. And right now I've got Windows Defender, which is keeping itself current and keeping an eye on the system, and all my browsers. Firefox and Chrome are both being updated, as is Edge. So I'm good for the time being. And I know, Leo, you're sort of in Linux; right? Is that where you are now?

Leo: Yeah, Linux and Mac. I don't actually have any Windows machines at this point. I mean, I should. And I'll probably find one somewhere. But, yeah, I don't have any Windows 7, for sure.

Steve: Well, I know what's going to happen is somebody will finally put it on some really tasty hardware that you have to have.

Leo: Yeah.

Steve: And so that'll...

Leo: So what happens is I buy - in fact, there is. There's like - although I'm waiting for Apple Silicon. But there's like a new HP Elite, which they're beautiful computers. But what ends up happening is I just buy the Windows machine and put Linux on it. After about six weeks with Windows I go, that's it. I'm done.

Steve: Yeah, I know.

Leo: I'm surprised that you've survived. But you have to; right? You have to.

Steve: Well, and frankly the developer tools, that's really one of the things that Microsoft really got right in the beginning. They created some world-class developer tools for Windows. And, well, also for the work on SpinRite and Beyond Recall, that's still DOS.

Leo: Right, you had to do that.

Steve: And soon it will be UEFI with no DOS. So I really do. It's just better for me to be developing on Windows.

Leo: I don't know. Emacs with assembly is pretty good. You should really consider it. I'm just saying. I know. You're still using Brief; right?

Steve: No, I had to give it up. It was 16 bits.

Leo: Oh, oh.

Steve: Yeah. And I'm glad. I made the switch. Although I am Visual Studio 2008, so I'm staying as old as I can.

Leo: Visual Studio is pretty nice. I actually use, even on Linux or Mac, I use Microsoft's open source VS Code, which is kind of a simplified Visual Studio. And I really like it. I use it all the time. Right now I'm studying Haskell, and it's a very, you know, the hardest thing about Haskell, the reason I've kind of turned my back on it is the tooling is so nuts. It's just crazy. It's not anything like - it's not like you write a text file and you compile it. It's much more complicated than that.

And finally, the Haskell community has finally addressed that. There's a very good VS Code plugin. Ghcup is a very easy way to install it. And so tooling, I don't - I completely agree with you. Tooling is 99% of it. Even I'm sure for assembler. If you've got the right tools, then the job is a lot easier. Debugging, testing. How do you test - that's the other thing. Do you have some sort of test harness for hard drives? Or how do you examine hard drive behavior?

Steve: So the first thing I did when I returned to this project was to create a development environment that would be comfortable.

Leo: There you go.

Steve: And so because I'm DOS-hosted - and I had a weird problem with Windows 10 because the DOS Samba client, Windows file and printer sharing, was so old for DOS that it didn't do the encryption that Windows 10 was insisting upon.

Leo: Oh, so it couldn't read your drive.

Steve: So I fought with that. But now what I have is I have a side-by-side, or actually Oracle's Virtual Box.

Leo: Ah, perfect. There you go, yeah.

Steve: I'm able to bring up a Virtual Box VM. It's networked into the network, so it's able to see my development directory, so I'm able to run the debugger. I'm using SoftICE, which was the best debugger back in the day. So I've got SoftICE on DOS. And so I can just hit - I just hit G and Enter, and immediately I'm in the code. The debugger pulls the source from the Windows directory. So I'm doing source-level assembler debugging, able to single step and watch the stack and all the registers and all that, but actually running in DOS. And so some of the things I can do there.

Some of the actual hardware stuff I need to use an external machine. So I have there an external box that boots FreeDOS, same network stack, in order so that it can see into my Windows directory. And then I just have a freestanding DOS machine. So having a comfortable development environment, because I plan to be doing this full-time now for the foreseeable future, that's the first thing I needed to establish.

Leo: Yeah, I would imagine, yeah. That's another kind of tooling.

Steve: And it's not easy because this stuff's getting so old.

Leo: Well, then, so are we, Steve. So...

Steve: Yes. We'll all fade at about the same time.

Leo: Better not. Better not fade. Keep going. Steve Gibson.

Steve: I figure I've got another 20 years in me.

Leo: That's good. I like hearing that. Steve is at GRC.com. That's his website. That's where you'll find SpinRite. That's the product he's talking about, his bread and butter, the world's finest hard drive recovery and maintenance utility. Soon development continues apace on v6. And if you buy it now, is it 6 or 6.1?

Steve: We're going to 6.1.

Leo: 6.1.

Steve: We're now at 6.

Leo: If you buy 6 now - I don't know why I forgot that, spaced that out. 6.1 is in active development. You'll get to participate in that. And of course you'll get an upgrade automatically, too. If you're there, check out all the other wonderful things, wonderful things Steve has to offer, including this show. He's got 16Kb audio for our friends in Australia and otherwise bandwidth-impaired. He has a transcript that's unique to his site. No one else has that, either. And that's really a useful tool for people who want to read along and learn from the show. I know people learn a lot listening to this show. You can also get 64Kb audio at his site.

We have both 64Kb audio and video at TWiT.tv/sn. You can also subscribe. There's a YouTube channel, if you want to watch the video only. And if you have a podcast player you could probably subscribe there, too. Security Now!'s been around for 15 years. I would imagine by now they would have added it to their directory. If they haven't, I think they never will. It's too late.

Steve, we will see you next week. We'll be coming off the iPhone announcement. I'll actually be curious. I know you're an iPhone user.

Steve: Yeah.

Leo: If you see any compelling new features, it'll make you want to buy it. Did you upgrade to 11?

Steve: No. I have 10.

Leo: Yeah. So this might be one you want to look at. We'll talk next week about that.

Steve: Okay.

Leo: Thanks, Steve. See you next time on Security Now!.

Steve: Thanks, buddy. Bye.