THE BUMBLEBEE LOADER

Description: This week we'll start off with a bit of fun over the most tweeted by far wacky tech news item. We then get serious with a very worrisome flaw which very likely exists in the WAN interface of the routers that many of us probably own. DDoS attacks have broken another record by a large margin. Both Chrome and Apple deal with, if not emergency, then at least high-priority software updates. We also have another major software repository tightening up its security against supply chain attacks. Then, after sharing just a few, but powerful, bits of feedback, we're going to step through the blow-by-blow operation and actions of the newest and meanest kid on the block with the emergence of a powerful malware loader that gets its name from the DLL it first loads: Bumblebee.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-885.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-885-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Oh, we've got some good stuff to talk about, a very worrisome flaw in the WAN interface of the routers many of us own. That's probably not where you want that problem. Another record DDoS attack and how Google mitigated it. And then a blow-by-blow, step-by-step, list of how the newest and meanest kid on the block works, the Bumblebee Malware Loader. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 885, recorded Tuesday, August 23rd, 2022: The Bumblebee Loader.

It's time for Security Now!, the show where we protect you online with this guy right here, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: Yo, Leo. Great to be with you.

Leo: Good to see you.

Steve: As we start into our 18th year.

Leo: OMG. Wow. Who would have ever thought that? Not me.

Steve: I have the hang of it now. So we've got Episode 885 for August 23rd. This one, I think a lot of our listeners are going to find this one interesting, I hope, because a security firm did a complete step-by-step, this is what we saw happening with a brand new piece of malware which is called Bumblebee, due to the name of the first DLL which contains it, which it arranges to get loaded. And as we'll see, this is not a Bumblebee that you want your enterprise to get stung with. Basically, this thing is taking over the previous means for getting malware onto people's, like, some poor unwitting person who clicked a phishing email in an enterprise. This thing gets in, and it never lets go.

And it's one thing just to kind of wave our arms around and say, oh, you know, malware. But the reason I wanted to take our listeners through this is that you really - it sort of gives you chills when you think about what this thing does. And it is also just a textbook-perfect example of the new approach which is being called "living off the land," where rather than bringing a bunch of stuff in, which has the danger of tripping security alarms, increasingly we're seeing advanced malware using and abusing existing code on machines, thus the term "living off the land." Anyway, that's at the end. We're going to start off with a bit of fun over the most tweeted by far wacky news item ever. Maybe. Well, okay, we've had some wacky ones, so probably not ever. But still, definitely it swamped my Twitter feed.

Leo: Now I'm trying to think what could it because there have been so many wacky stories this week.

Steve: Oh, you will know. I heard you talking about it, I think maybe on Sunday. Anyway, we're then going to get serious with a very worrisome flaw which likely exists in the WAN-facing interface of the routers that many of us probably own. We've got a new record having been broken for DDoS attacks, which Google managed to fend off. And that's broken by a large margin. We have both Chrome and Apple dealing with, if not emergency, then at least high-priority software updates to squash some zero-days that were in active exploit. We've got another major software repository tightening up its security against software supply chain attacks. And then after sharing just a few, but powerful, bits of feedback from our listeners, we're going to step through, as I said, the operation and actions of the newest and meanest kid on the block with the emergence of a powerful malware loader that is called Bumblebee. And of course a great Picture of the Week, too.

Leo: Nice. Bumblebee bumblebee, fly away home. Oh, no, that's ladybug ladybug. That's a different one. This is a Video of the Week this week, Steve.

Steve: It is. I should say, though, that if you're not convinced yet that something like a Canary is what you need, you will be by the end of this podcast

Leo: Oh, boy.

Steve: Because it is exactly the sort of thing that the guys behind the Bumblebee loader would trip over. And in fact the timeline, I was thinking about this already, that there's enough manual side to this that, if you did receive notification of early strange behavior, there would be time.

Leo: Oh, that's good. That's good.

Steve: And so anyway, it does play perfectly into today's topic. So, yeah, we have a Video of the Week. I had not seen this before. I don't know, YouTube - oh, I was looking for something else, and it came up and said you might think this is interesting. I thought, oh, it's a little freaky how correct you are, Google. So we're playing it now. And I played it for Lorrie, and she's like, what?

Leo: Last year my trainer said, "I saw a UFO last night." I said, "You did? What did it look like?" He described what you're seeing right now. And I said, oh, that ain't no UFO. What is it, Steve?

Steve: So it is - and describing it can't do it justice. I mean, it would be - I guess we're sort of used to stuff happening near Earth, so maybe it wouldn't be that surprising. But it's extremely cool. What you're showing is a video of the light reflected, sunlight reflected off of a train of 53 Starlink satellites which had been - and this was taken a couple days ago - which had been launched from Florida on Friday. And this has been going on for a while as Elon's company is getting the Starlink constellation of satellites up in the air. But, I mean, the idea of - and this is just some random guy with a cell phone recording this. I mean, and it is, like, it's a series of dots spaced somewhat evenly. I mean, it's kind of perfect that they're not perfectly even because I guess these have been released from the transport vehicle that got then up into orbit.

Leo: Yeah. They do this every time. They fly in formation, and then they deploy to their locations. But when they first come off of a - it's like a MIRV. You know, when they first come off of the rocket they're all in formation. Isn't that cool?

Steve: It's just too cool.

Leo: It's a "train," they call it.

Steve: Yes. Nothing to do with security, but just something very cool.

Leo: Yeah. And if you ever, you know, you can see it. It's happening all the time. They've done many, many launches. So if you get a chance, it's worth seeing.

Steve: So since we're on the topic of nothing to do with security, I needed to respond to what was by far the most tweeted to me news item in a long time. And our listeners who are naturally on top of their game felt about this pretty much as I do. For more than 2.5 decades, the highly respected Microsoft engineer Raymond Chen has been blogging.

Leo: Oh, I do know what you're talking about. I love this.

Steve: I knew you would. Last Tuesday he posted a blog entry that was just so weird that everyone picked up on it. Raymond's posting was titled "Janet Jackson had the power to crash laptop computers." Now, the fact that this was assigned a CVE number

(CVE-2022-38392) has apparently lent it more credibility, or at least more notoriety, than I think it deserves. And the fact that the CVE refers to Raymond's blog as its sole reference seems to be somewhat self-referential. Raymond cites the CVE which cites Raymond.

And I'm actually wondering whether it might have been a slow blog week, and Raymond may have needed a bit of filler. So his blog post opens with two lines: "A colleague of mine shared a story from Windows XP product support." Okay, well, that wasn't recent, presumably. Anyway, he said: "A major computer manufacturer discovered that playing the music video for Janet Jackson's 'Rhythm Nation' would crash certain models of laptops."

Okay. So from the CVE we learn that this was "certain models of laptops" circa 2005, so 17 years ago. The CVE's formal description says - because, you know, if it's a CVE you need a formal description. It says: "A certain unnamed 5400 RPM OEM hard drive, as shipped with laptop PCs in approximately 2005, allows physically proximate attackers to cause a denial of service," it says, "device malfunction and system crash via a resonance frequency attack with the audio signal from the 'Rhythm Nation' music video." If this wasn't April, I mean, it's not April 1st; right? So, really? So we can see now why the tech press thought that this was just too wonderful to pass up. On the other hand, we have a CVE that was apparently issued based upon what amounts to "a friend told me" rumor. No mention of the make or model of the 5400 RPM OEM hard disk that should be kept away from discos.

So this begs the question of just how low the bar has been set for issuing CVEs. This is not an attack, although, okay, there are a vocal group of people who feel that any playing of Janet Jackson's "Rhythm Nation" should qualify as a form of terrorism. And neither is it a bug that needs to be fixed, nor malware that needs to be expunged. There's no action that can or should be taken today. It's from 17 years ago. So why give this, you know, "heard it from an XP support guy" a CVE in 2022? I have no idea.

And of course those who've been following this podcast will recall that video, which was also cited in some of the coverage of this "Rhythm Nation" hard drive DDoS attack, where somebody - and we showed this on the podcast; right? - was monitoring the dynamic throughput of an array of spinning hard disk drives while screaming at the array at the top of his lungs. And sure enough, the throughput visibly dropped during the screaming. And as we noted at the time, the throughput dropped because modern mechanical hard drives have crammed their tracks so closely together, I mean, actually they are now overlapping each other. It's like this is what engineers do; right? They engineer DRAM so tightly that neighboring rows interfere and cause bits to flip. Well, they've also crammed hard drive tracks so closely together that they have become quite sensitive to any exogenous vibration. And in fact, the way they're mounted in the server chassis can be critical.

Now, Raymond, of course, also referred to the famous video showing that 1940 collapse of the Tacoma Narrows Bridge. In the same way that "Rhythm Nation" was able to rub some hard drives the wrong way back in 2005, the coincidentally timed gusts of wind through the Tacoma Narrows rubbed the bridge the wrong way, until it disintegrated. Anyway, I felt that this podcast needed to at least acknowledge this story that everyone tweeted to me over the last week, and that most of the tech press had a lot of fun talking about, as did we here.

In a not-so-fun and much more relevant issue, during one of the recent DEF CON presentations in Las Vegas, a team of four Argentinean researchers from the cybersecurity company Faraday Security detailed their discovery of what was subsequently classified as a CVSS 9.8 zero-click remote code execution vulnerability in the interface stack which Realtek provides in the SDK for their hardware's use with the

very popular open source eCos operating system. Since they had previously responsibly disclosed their discovery, and since Realtek had patched the flaw in March, their presentation during DEF CON provided full disclosure of the all technical details needed to replicate the attack. After all, it's been a few months.

Consequently, there is now exploit code released publicly for this critical security vulnerability affecting networking devices which use Realtek's RTL819x System on a Chip. And those devices number, unfortunately, in the tens of millions. Being of the turnkey consumer "plug it in and forget it" variety, there's little chance that most of these tens of millions of devices are ever going to be updated. Many will have long since gone out of warranty. Since this Realtek System on a Chip RTL819x is incredibly popular, we're talking about devices that many of us probably already have, since the chips are used by more than 60, six zero, vendors including ASUSTek, Belkin, Buffalo, D-Link, Edimax, TRENDnet, and Zyxel.

And again, zero-click on the WAN interface. The vulnerability presents on the WAN interface. And because it's a stack-based buffer overflow, it allows for no operator needed compromise of the host upon receiving a UDP packet from the public Internet. The DEF CON presentation left nothing to the imagination. The SIP, you know, SIP protocol, the ALG, the Application Layer Gateway function that rewrites SDP data has a stack-based buffer overflow. This allows an attacker to remotely execute their code without authentication via a crafted SIP packet that contains malicious SDP data. Which ends up getting written onto the stack and then executed.

We've spoken about the abuse of application layer gateways in the past. Remember that ALGs are essentially enhancements to the baseline NAT routing functionality, which allows NAT to handle what would otherwise be NAT's interference with the details of specific NAT-unfriendly protocols. The simplest example is the original FTP protocol where the client instructs the server which port it has opened to receive the reverse connection from the FTP server. The router's application layer gateway monitors the outgoing data, sees the port being specified by the FTP client by looking in the packet as it's leaving the router, and then either opens that port in its WAN side interface so that the remote FTP server can connect in, or modifies the outbound port specification to a port it wishes to open. The point being that it allows a NAT router to become transparent to the otherwise NAT-hostile FTP protocol.

Now, in this case the trouble exists in the application layer gateway logic for handling SIP, that's the session initiation protocol used in VoIP systems. It's not clear whether disabling SIP's ALG, if that's even an option in the router, would help.

Johannes Ullrich, who's the Dean of Research at SANS, says that a remote attacker could exploit the vulnerability for the following actions. They could crash the device, okay, that's easy; execute arbitrary code, a little more tricky; establish backdoors for persistence, that's what you want to do; reroute or intercept network traffic, you know, turning it into a proxy; basically, take over any vulnerable router. And he warned that if an exploit for CVE-2022-27255 were turned into a worm, it could spread throughout the Internet in minutes.

Now, while Johannes is technically correct about a worm, as I've been saying recently, a massive worm attack no longer makes any sense. They made sense back when email viruses just existed to see whether they could. But in today's world, it's about money. Anyone who's capable of writing a working worm would also be capable of using the router as a proxy to bounce malicious traffic; or to quietly mine cryptocurrency; or to enslave the router into the service of one of today's massive botnets, as we'll see in the next story; or to pivot into the network behind the router to see whether there might be something juicy worth attacking somewhere on the router's LAN.

For their part, the four security researchers said that devices using firmware built around Realtek eCOS SDK before March of 2022 are vulnerable. Period. Full stop. Users are vulnerable even if they do not expose any admin interface functionality. Attackers may use a single UDP packet to an arbitrary port to exploit the vulnerability. And this vulnerability will likely affect routers the most, but some IoT devices built around Realtek's SDK may also be affected.

Realtek's own vulnerability report - I went looking for it and found it - is two pages and provides no guidance. There's no list of manufacturers or makes and models of affected products, nothing. There's no action that any responsible end user can take. There's no obvious way to know whether any particular router or IoT device might be affected. The only recourse would be to proactively verify that your router is running the latest firmware available for it from its vendor, and to hope that they care enough to update their firmware for the model of router you have. If your system can run one of the alternative router firmware systems such as DD-WRT, I think that's what I would do. That would be one way to move it to safety, into a platform that is being continuously kept up to date.

So many of these sorts of things have come out through the years, problems that are unlikely to ever be fixed, that it's possible to imagine what must exist, the sort of massive known vulnerability database that both nation states and sufficiently large criminal enterprises must now be maintaining. You want to get into which organization? What equipment do they have on their border? Look up all of the known exploits over time that have been available against it and start working down through the list until you get in.

Seventeen years ago, back when we were launching this podcast, that scenario would have seemed like pure speculative fiction. Today, I'd lay money down that such databases must now exist all over the world. And it's difficult to see how this changes. There are no plans in motion right now because anything would take a while to have any effect. Nothing is going on that's going to like change the way the world is working now. And the way the world is working now is deeply broken.

So those who can afford to be truly aware and concerned about security could choose to use a non-consumer router on their borders, such as something running pfSense, as I do at my locations. But that still leaves the much larger majority of end-user consumers potentially vulnerable for decades to previous old vulnerabilities. And every month more of these surface. And routers are not being updated with nearly the speed or reliability that they should be. And as I said, we don't seem to be taking any action.

Okay. Forty-six - it's hard to say this. Forty-six million requests per second. After standing up to the largest-ever DDoS attack on behalf of one of its "Cloud Armor Adaptive Protection" customers, Google said it had blocked a record-breaking HTTPS-based DDoS query attack that hit, at its peak, a whopping 46 million requests per second. That puts it 76% higher than the 26 million RPS attack which we talked about previously which had been mitigated by Cloudflare in June. Google chose not to disclose the target of this attack, its customer, behind this protection, but said that it believes the attack was carried out with the help of the Meris botnet.

To put the scale of this attack in perspective, it's an HTTPS request rate equivalent to receiving all of the requests to the Wikipedia domain which is one of the top traffic domains in the world which Wikipedia would receive during one 24-hour period, take all of those requests and compress them into just 10 seconds. It's that much traffic. And this thing went on for quite a while. Google's report of the attack - and Leo, thank you. We got the chart showing the shape of the attack, peaking at 46 million RPS. Google's report of the attack contains lots of interesting details. Here's what they shared.

They said: "Starting around 9:45 a.m. Pacific time on June 1st, an attack of more than 10,000 requests per second began targeting our customer's HTTP/S Load Balancer. Eight minutes later, the attack grew to 100,000 requests per second." They said: "Cloud Armor Adaptive Protection detected the attack and generated an alert containing the attack signature by assessing the traffic across several dozen features and attributes. The alert included a recommended rule to block on the malicious signature."

They said: "Our customer's network security team deployed the Cloud Armor-recommended rule into their security policy, and it immediately started blocking the attack traffic. In the two minutes that followed, the attack began to ramp up, growing from 100,000 requests per second to a peak of 46 million requests per second. Since Cloud Armor was already blocking the attack traffic, the target workload continued to operate normally." Meaning their site wasn't adversely affected. It stayed on the air. Everything was fine. "Over the next few minutes, the attack started to decrease in size, ultimately ending 69 minutes later at 10:54 a.m. Presumably, the attacker determined they were not having the desired impact while incurring significant expenses to execute the attack." Now, I would argue that point. I suspect that the attack cost the attackers exactly nothing other than the exposure of the IP addresses of their fleet of infected consumer routers hosting the Meris botnet. But maybe not even that.

They said: "In addition to its unexpectedly high volume of traffic, the attack had other noteworthy characteristics. There were 5,256 source IPs from 132 countries contributing to the attack. The top four countries - Brazil, India, Russia, and Indonesia - contributed approximately 31% of the total attack traffic. The attack leveraged encrypted requests (HTTPS) which would have taken added computing resources to generate."

Again, Google appears to be deliberately missing the whole point. If there were 5,256 observed source IPs, then that crypto burden will have been well distributed across the globe. And then we learn that HTTPS pipelining was also in use, further limiting the crypto overhead. Google said: "Although terminating the encryption was necessary to inspect the traffic and effectively mitigate the attack, the use of HTTP pipelining required Google to complete relatively few TLS handshakes."

Right, and thus also much less burden on the attackers. It required them, the attackers, to similarly terminate relatively few TLS handshakes. The attackers were establishing a single TLS connection, then attempting to flood that connection with pipelined HTTP requests. There's no reason to believe that any IP that's flooding HTTPS requests down the pipe will ever generate a valid request. So those IPs should have and hopefully were simply dynamically blacklisted.

They said: "Approximately 22%" - which was 1,169 - "of the source IPs corresponded to Tor exit nodes, although the request volume coming from those nodes represented just 3% of the attack traffic." Now, let's stop there for a second. That's interesting. 22%, so just shy of one-fifth, of the total source IPs were coming from Tor, yet its traffic was just 3%, which is what we'd expect; right? I mean, Tor incurs a huge latency burden and bandwidth burden on its user in return for giving you some hope for privacy on the Internet. Anyway, sort of an interesting data point.

They said: "While we believe Tor participation in the attack was incidental due to the nature of the vulnerable services, even at 3% of the peak" - which would have been greater than 1.3 million requests per second - "our analysis shows that Tor exit nodes can send a significant amount of unwelcome traffic to web applications and services."

And finally: "The geographic distribution and types of unsecured services leveraged to generate the attack matches the Meris family of attacks." And I'm sure they couldn't resist poking at a few of those IPs and confirming that, yup, in fact that was Meris.

Anyway, they said: "Known for its massive attacks that have broken DDoS records, the Meris method abuses unsecured proxies to obfuscate the true origin of the attacks."

So yes, if it was Meris, then they were bouncing their traffic through proxies like routers that had been compromised. Remember we've talked about how if a router exposes its plug-and-play port, or the plug-and-play service to the WAN interface, it's possible for someone to just set up a proxy and say, you know, send any traffic incoming to that IP and use it to reflect traffic. Anyway, so we know that attackers were bouncing their botnet's traffic through intermediate proxies in order to protect the IPs of their actual bot agents, which they did not want revealed.

So I couldn't help - another just random point. I couldn't help but note that in their redacted report because Google, in their posting of this, they had screen shots which were redacted to hide the identity of their customer. They used text blurring to obscure the identity.

Leo: Uh-oh.

Steve: Whoops.

Leo: Uh-oh.

Steve: And we all know, remember, we covered it here. In some beautiful work that we covered a while back, we learned about that blurring text is not secure. We learned it doesn't work. If someone is interested in learning what original text lies behind the blurred instance, they can identify the details of the typography from all of the examples of non-blurred text, then iteratively guess the text that's behind the blur, employ the same blurring of their guess text, and then compare the result of the two blurrings, one they control and one they do not.

Leo: We'll leave this as an exercise for the listener.

Steve: Yes. Google's report then switches into marketing mode, bragging about their technology, which anyone would have to agree works. So we're now living in a world where those whose Internet web services must remain online in the face of attacks will need to bear the added cost of the privilege of doing so by putting themselves behind Google or Cloudflare or one of these big pipe DDoS protector services because otherwise you just, I mean, aiming that much traffic at anyone else, I mean, it's just like, what's the point?

Leo: Right.

Steve: It seems like just stomping on a gnat using a planet. Just would be ridiculous.

And Leo, I'm going to take a sip of water. Let's tell our listeners...

Leo: Okay, let's do it.

Steve: ...why they're glad they're here.

Leo: Yeah, we use Amazon for DDoS. They also - anybody who has a lot of bandwidth can do that, DDoS protection. Cloudflare does a great job. And I think this is as much a war of press releases as anything else because I think Cloudflare just recently had an almost as big DDoS mitigation.

Steve: Yeah, in June.

Leo: In June, right, yeah. So it's like, well, our DDoS was bigger than yours.

Steve: The entire entry point for this Bumblebee loader is spear phishing.

Leo: Of course. Of course.

Steve: And as we'll see, I wrote all this before I knew the advertising.

Leo: You didn't know. Yeah, yeah.

Steve: It is ultimately, as we'll see, that hapless person in the enterprise who clicks on an email and takes some actions which will clearly specify that begins this entire thing. And it just must be keeping IT people up at night.

Leo: Oh, and owners. It's terrifying.

Steve: Yeah, yeah. Okay. So last Tuesday Google updated our Chrome browser for desktops to squash an actively exploited high-severity zero-day flaw in the wild.

It's tracked as CVE-2022-2856. That's only four digits. Interesting. As we know, CVEs are allocated now in blocks, and so various people allocate them as they choose. So the fact that it's four digits and a low number doesn't really tell us anything. Anyway, the issue is a case of what they termed insufficient validation of untrusted input, which is like Microsoft saying, yeah, that was a security bypass. Okay. Right. Security researchers Ashley Shen and Christian Resell, both on Google's TAG team - remember, you know, their Threat Analysis Group - are credited with reporting the flaw last month on the 19th of July.

As usual, there's no upside for Google to sharing anything more with us beyond "Please be sure that your version of Chrome now ends in .102." They did add that "Google is aware that an exploit for CVE-2022-2856 exists in the wild." In addition to stomping on that 5th of the year actively exploited flaw, that update to blah blah blah .102 addressed 10 other security flaws, most of which relate to the most common use-after-free bugs that we just keep encountering in this code, which appear in various Chrome components. They also fixed a heap buffer overflow in the downloads portion of Chrome.

So, this is number five this year. Previously we had a - the first of the year was a use-after-free in animation. We had two type confusion bugs in V8 and a heap buffer overflow

in WebRTC. So, and now with number five being a rather vague "insufficient validation of untrusted input," okay. Anyway, so if you're using one of the non-Chrome Chromium siblings - Edge, Brave, Opera, or Vivaldi - just be sure to keep yourself updated there, too, because they would all be susceptible until they're updated with the latest update to Chromium, you know, their common core.

And not to be left behind, last Wednesday Apple released high-priority security updates for iOS, iPadOS, and macOS platforms. This was to remediate a pair, two zero-day vulnerabilities which were being exploited by threat actors to compromise Apple's devices. There was the CVE ending in 32893, which was an out-of-bounds bug in WebKit which could lead to the execution of arbitrary code by processing a specially crafted web content; and 32894, an out-of-bounds bug in the OS kernel that could be abused by a malicious application to execute arbitrary code with the highest privileges.

So again, these are not theoretical. These were found being used to perpetrate malicious ends, so that was pushed quickly in all of our devices. I got little notices everywhere. So Apple clearly felt that this was worth getting out into the world. They said that they had addressed the issues with improved bounds checking, so that's good because those were out-of-bounds bugs, so you want to do a little more bounds checking to keep them from going out of bounds. And they said also that they were aware that the vulnerabilities "may have been actively exploited." Uh-huh. And please update immediately before you do anything else.

So as usual, they didn't disclose anything additional either regarding these attacks or the identities of the bad guys who may have been using them; although, as usual, it's almost certain that they were involved in targeted intrusions. Again, you're not going to spray this around because you want to keep it quiet and get as much use out of it as you can. And since we're counting zero-days so far this year, this latest update brings Apple's total of actively exploited zero-days to six for the year. As with Chrome, we had four before these latest pair, and we've covered them all in the podcast in the past. So anyway, two more added to the list, and iOS, iPadOS, and macOS Monterey all need to be updated.

As we know, RubyGems is the official package manager for the Ruby programming language. And in a welcome response to the increasing threat and prevalence of supply chain attacks, the RubyGems repository has become the latest platform, following NPM and PyPI, to require multifactor authentication for its more popular package maintainers. Specifically, owners of "gems," as they are referred to, having more than 180 million total downloads are now, as of last Monday, August 15th, required to enable multifactor authentication.

The RubyGems management said: "Users in this category who do not have MFA enabled on the UI and API or UI and gem sign-in level will not be able to edit their profile on the web, perform privileged actions, for example, push and yank gems, or add and remove gem owners, or sign in on the command line until they configure MFA." In other words, they'll log in, try to do anything, and they're going to get a nope, sorry, you're too popular. You've got more than 180 million downloads. You should be proud of that. But just, you know, come on. Let's do multifactor authentication here.

And as gem downloads approach that magic mandatory 180 million count, as soon as downloads pass 165 million cumulative, their maintainers will receive reminders to turn on multifactor authentication before the download count hits the magic 180 million, at which point they no longer have a choice.

So this is further welcome, of course, in the package ecosystem to improve the past's casual approach to software supply chain security, which no one took that seriously until we started discovering lots of malicious packages in our repositories. So as we know, adversaries are increasingly setting their sights on open source code repositories, with

attacks on NPM and PyPI having snowballed by a combined 289% since 2018. Researchers from Checkmarx, Kaspersky, and Snyk have all uncovered a large number of malicious packages in PyPI that could be abused to conduct DDoS attacks and harvest browser passwords as well as Discord and Roblox credential and payment information.

So now RubyGems joins the ranks of NPM and PyPI which are all tightening their security. And, you know, yay. I mean, it's nice to see this happening. It's clearly something that is easily done. As I said before, we still don't have an answer to the IoT problem, to this problem that we've got very sophisticated devices packaged in $5 light switches and plugs, with no one standing behind them, where vulnerabilities are being found by researchers, and there's just no infrastructure in place to fix them. And it's not like this is like slowing down. The rate at which this stuff's being created is accelerating, and there's nothing on the horizon that suggests a way to fix this. And even if there were, or once there is, it would still take decades for it to work its way through. So, bad.

Okay. We have some very neat closing-the-loop bits. Thomas Tomchak tweeted. He said: "Hey, Steve. When you register a domain, you do have the option to register a technical contact as well as the owner." He said: "When I have registered domains in the past for friends, I always make sure they are listed as the domain owner, and myself only as the technical contact." He says: "Yes, it's still under my registrar account, but at least that shows proof of ownership, and they could probably transfer it to a new account at the same or different registrar should I get hit by a bus unexpectedly."

So I'm so glad that Thomas thought to remind me and us of that. And I want to acknowledge that several other of our listeners sent notes to the same effect, which I saw, and thank them. Of course that's the case. And it had completely slipped my mind. Domain registration records provide for completely separate Owner, Administrative, and Technical contacts. I'm so used to always pointing all three of those at myself that I completely forgot the power of the flexibility that they could provide. Now, this of course begs the question, what would a domain registrar do if the assigned owner of a domain which is, after all, just a name and email address wished to take control of the domain in the event that the Admin or Technical contact was unresponsive? Would that provide the degree of safety that we're looking for?

I don't know. In an attempt to answer that question definitively, I found an ICANN FAQ, you know, an FAQ. Question 12 that they asked themselves read: "I can't access my domain name or my domain name management account because the domain name was registered by someone else, such as my web developer or administrative contact. What now?" And their answer, ICANN's answer is: "You may not be able to access the domain name if you are not the administrative contact/registrant of record of the domain name. You should contact the individual or entity who registered the domain name to obtain access credentials/details or update the domain name's administrative contact/registrant of record."

Then they said, in the second paragraph: "You should contact your registrar right away if your domain name manager/administrative contact is unreachable, has gone out of business, et cetera, to update your information. Once you're able to become the administrative contact/registrant of record, this will ensure that you have full control of managing your domain name and allow you to find someone else to help you manage your domain name, if you so choose. It's a good idea to keep a record of your domain name management credentials at all times, even if you choose to outsource some administrative/management duties to a third party."

Okay. So the formal names of the three things you can register, there's the Owner, there's Administrative, and there's Technical contact. And in ICANN's response they keep referring to Administrative contact. Well, okay, that's one of the three; right? Administrative. But there's also Owner. So I dug around some more, and I couldn't find

anything formal or official. But anyway, I just wanted to put on the record that a bunch of our listeners said, "Hey, Gibson, did you forget about that?" It's like, yeah, I did. So thank you.

Leo: I don't think it solves anything. It's the same issue. Because in effect ICANN's saying, well, if you can prove to your registrar that you're you, which is what it would require to change the administrative owner, well, then I guess we'll give you, I mean, it's putting it back on the registrar. I don't think it solves anything.

Steve: So what I was suggesting last week was that we needed, like that the system needed to be upgraded to provide something like a next-in-kin sort of effect so that there would be some means for dealing with a sudden lack of management. So our listeners said, hey, you know, you do have three things in a domain name registration record. And so it would be possible for you to ask the person who's managing your domain name to point one or more of those at you, rather than all at him. So it's something. But again, you still don't have, like...

Leo: There's no backdoor.

Steve: There won't be an access to the registration record itself.

Leo: Plus you need the cooperation of this person who is presumptively not cooperating.

Steve: Right.

Leo: That's the problem. If they're cooperating - this is why you should register it yourself, I think. But, you know, that's what I would say. Do it yourself. Don't trust them.

Steve: Okay. So a listener, Dagan, he said - well, I should preface this with I am truly flattered, honored, and humbled that this podcast has had as much impact on people's lives through the years as it has. I suppose that my true love for technology - and Leo, you share that love with me, you always have - and computing can be a bit infectious. So in that sense I'm gratified that we've had the opportunity to infect so many of our listeners with this bug.

Leo: Infect, okay, yeah, okay.

Steve: Yeah.

Leo: All right.

Steve: And it's a bug that I've been in its grip throughout my entire life, so this tweet from Dagan really hit home. So I asked its author if I could share it, and I received his permission.

So he wrote just to me by DM. He said: "Steve, I wanted to privately drop you a brief personal note of thanks. I've been listening to Security Now! religiously since 2011, and I truly believe your weekly show has had a significant impact on my career and, as such, my life. When I started listening, I was a happy nerd that loved the idea of security. Nine years in, I discovered and was credited for identifying two CVEs in a Red Hat product. Last year I was credited for two more CVEs in a SUSE product, and this year yet another vulnerability in a second SUSE product.

"And finally, just last weekend, I was fortunate enough to have the opportunity to present at DEF CON, where I demonstrated chaining three of five vulnerabilities together to fully compromise a Kubernetes cluster from the outside. I will also be speaking at KubeCon this year, where I will talk about securing clusters from risks introduced by third-party applications." He said: "I'd happily buy you a nice bottle of wine to express my most sincere gratitude, but I'll settle for a SpinRite license instead."

Leo: Aw, that's nice.

Steve: He said: "Your show has changed my life, Steve. Thanks." Now, just this morning, you, Leo...

Leo: Ah, I was wondering if you got that. Okay.

Steve: Yes. Forwarded a very heartwarming story from someone who said his career was catalyzed by this podcast. So as we start into year 18, I want to share what he wrote as an example of what is possible if anyone else out there might be in need of a bit of a nudge. So this person wrote: "Leo, I must say I'm quite honored to have received a reply directly from you. I would like to share a brief story you and Steve might like to hear.

"In 2015 I recognized Steve's name on a list of podcasts. I knew it from back in the mid-to-late '90s when I used ShieldsUP! frequently. I left IT after the dot com bubble around 2002-ish. I followed a path in science, but ultimately I couldn't find a good job in that field. I was in poverty, on Medicaid, and in significant debt. It wasn't just me I met my wife and our son was just born during this time. It was rough. I listened to Security Now! and the ad for ITPro.TV. I signed up, studied, and passed the CCNA.

"In a matter of days I had a good-paying job as a network engineer. I have since moved on into cybersecurity, getting a CISSP. I now work at a job I love, and my family is able to live debt-free with good healthcare and everything else that goes with this terrific career."

Leo: Yay.

Steve: "I have you and Steve to thank for this." Well, and of course his own initiative, too. He says: "Your show made this field approachable and fun. Words cannot begin to express my gratitude and appreciation. You guys change lives. Please keep up the great work. Michael."

Leo: Thank you. Thank you. Yeah. Yeah.

Steve: So thank you, Michael, for sharing your story. You know, the field of cybersecurity truly is a growth industry. There is a crying need for trained cybersecurity professionals. And, boy, is it interesting and fun.

Leo: Yeah, yeah.

Steve: So, you know, anybody else who's looking for something to do, you can follow in the footsteps of these guys.

Leo: Okay. I'm not crying. You're crying. Okay.

Steve: I know. I got choked up the first time I read it.

Leo: It was a beautiful - but, you know, I have to say, every time, and I know you know this, too. But every time we go out, we meet people, I hear these stories again and again. And so thank you for the job you do, Steve. And I always tell them, it's not us, it's you. But thank you for letting us be part of your life, and I'm glad we could help.

Steve: Wouldn't be happening, Leo, if it weren't for you.

Leo: Okay. What do you want to do now?

Steve: Let's tell our listeners about our last spot and see whether it also...

Leo: Another inspiration. All right. You know, I tell you what, we don't do this for the money particularly. We do it because it's a privilege and an honor, and it's something we really enjoy. I'm don't think - I'm not saying anything you don't feel. But it is always nice. It is always nice to hear from people and know that we've made somewhat of a difference. So thank you very much for those kind words. I'm going to promote this new book from...

Steve: Dennis Taylor.

Leo: Dennis Taylor, that's it.

Steve: We should mention that, because we were talking about it before we began recording.

Leo: Be good for the book club, yeah. Oh, you didn't mention it in the show?

Steve: No.

Leo: Oh, it was before. Yeah.

Steve: Yeah. This is the guy who did the Bobiverse trilogy.

Leo: Mr. Bobiverse, yes.

Steve: Of four books. And I was planning to be reading something else, but one of our listeners said, uh, you know how much you like the Bobiverse. He wrote something else called "The Singularity Trap." And I was talking to Leo before the show. I have not been able to put it down. I've been reading it during the ad reads.

Leo: No. Oh, you're going to finish it before the show's over.

Steve: Well, I'm going to finish it tonight, unfortunately.

Leo: Nice. Oh, nice.

Steve: It is so fun.

Leo: Fantastic.

Steve: So it's Kindle Unlimited, so if you are a reader it won't cost you anything, in the same way that the Bobiverse books didn't.

Leo: Oh. Oh, good. Oh, nice. Okay, good.

Steve: And you learned that it's got the same guy reading it as who read the Bobiverse for Audible.

Leo: Ray Porter's fantastic, yeah.

Steve: And everyone was raving about the Audible side of the Bobiverse.

Leo: Yes.

Steve: So anyway, it's called "The Singularity Trap." It is the same style, the same wit. It's like really fun. And it's a book where you have no idea what is going to happen next. It's a whole new concept that we've - it's not a rehash of let's go attack the aliens thing. It's a new idea, and really cool. So this one I can, I mean, I'm at, sadly, at 93% or

something of the book, and I'm going to be sad to have it end. And then I went looking for more stuff, but he's not written anything else. He's going to do some more Bender books on the Bobiverse side I guess next.

So anyway, okay. The Bumblebee Loader. It's recently become a big deal on the malware front. Symantec in June wrote: "Bumblebee, a recently developed malware loader, has quickly become a key component in a wide range of cybercrime attacks and appears to have replaced a number of older loaders, which suggests that it's the work of established actors and that the transition to Bumblebee was pre-planned. By analysis of three other tools used in recent attacks involving Bumblebee, Symantec's Threat Hunter team has linked this tool to a number of ransomware operations including Conti, Quantum, and MountLocker. The tactics, techniques, and procedures used in these older attacks support the hypothesis that Bumblebee may have been introduced as a replacement loader for Trickbot and BazarLoader, since there is some overlap between recent activity involving Bumblebee and older attacks linked to these loaders."

And even earlier than that, on the 7th of June, Cyble, the security firm we were just talking about last week, wrote: "In March 2022, a new malware named Bumblebee was discovered and reportedly distributed via spam campaigns. Researchers identified that Bumblebee is a replacement for BazarLoader malware, which has delivered Conti ransomware in the past. Bumblebee acts as a downloader and delivers known attack frameworks and open source tools such as Cobalt Strike, Shellcode, Sliver, Meterpreter, et cetera. It also downloads other types of malware such as ransomware, trojans, and more."

And last Wednesday, the global security firm Cybereason, based in Boston, Massachusetts with offices in London, Tel Aviv, Tokyo, France, Germany, South Africa and Singapore, published in their "Malicious Life" blog a detailed technical description of the operation of this extremely dangerous new entry onto the malware scene. They titled their report: "Bumblebee Loader - The High Road to Enterprise Domain Control."

And this podcast would be remiss if we didn't take some time to bring our listening audience up to speed about this emergent threat. So Cybereason's report explains that they analyzed a case that involved a Bumblebee Loader infection which allowed them to describe in detail the attack chain from the initial Bumblebee infection to the compromise of the entire enterprise network.

Okay. So let's begin with a couple of bullet points to set the stage. "The majority of the infections with Bumblebee," they said, "we have observed started by end users executing LNK (.lnk) files" - still hasn't gone away - "which use a system binary to load the malware. Distribution of the malware is done by phishing emails with an attachment or a link to the malicious archive containing Bumblebee." And I'll be expanding on all of this here in a minute.

They said: "Bumblebee operators conduct intensive reconnaissance activities and redirect the output of executed commands to files for exfiltration. The attackers compromised Active Directory and leveraged confidential data such as users' logins and passwords for lateral movement. The time it took between initial access and Active Directory compromise was less than two days." And I'll be sharing a timeline breakdown in a minute.

"Cybereason GSOC" - that's their Global Security Operations Center - "has observed threat actors transitioning from BazarLoader, Trickbot, and IcedID to Bumblebee, which seems to be in active development and generally the loader of choice for many threat actors. Attacks involving Bumblebee must be treated as critical. Based on GSOC findings, the next step for the threat actors is ransomware deployment, and this loader is known for ransomware delivery."

Okay. So let's take this step by step. A spear phishing email is received containing an archive or a URL link to an external source to download the archive. As we know, the malware is encapsulated in an archive to prevent the archive's contents from being tagged with the Mark of the Web (MOTW) which would complicate its execution. The user extracts the archive and mounts the resulting ISO (.iso) image. Newer releases of Windows will happily mount .ISO images, thus exposing the ISO's file system files. The content of the mounted ISO image is a .LNK file executing the Bumblebee payload upon user interaction.

So the operators behind an instance of Bumblebee host malicious websites that implement a drive-by download. To infect the system, an end user has to first manually decompress the archive containing the ISO file - on the other hand, if it's a ZIP, Windows will do that for you now, too - mount the file, and then execute the Windows shortcut LNK. This is all done as part of a phishing email where the user fully believes that they are doing the right thing, that installing this or that is needed, or updating something that's needed before they can proceed. So the user is without question unwittingly complicit in the success of this entire penetration and intrusion. All of the other mechanics is about avoiding everything the user's enterprise security people have done to keep bad stuff out, despite what dumb stuff their users may do.

So the LNK file has an embedded command to load and execute the Bumblebee Dynamic Link Library, the Bumblebee DLL, using the already and always present odbcconf.exe in what has become the increasingly popular "living off the land" approach of using what's already available in the system. And these days plenty is, in modern systems. So in this context, odbcconf.exe is called a LOLBin. A response, which has the extension .rsp, a response file is also used where some Bumblebee-specific name .rsp contains the reference to the Bumblebee DLL. So specifically, the LNK file's target property contains the string odbcconf.exe space -f space, and then this Bumblebee-specific name ending in .rsp, the response file. And the .rsp file contains a reference to, again, the Bumblebee-specific name .dll which is the Bumblebee payload.

Now, if anyone's curious, you can see this for yourself. In any version of Windows, open a command prompt and type "odbcconf /?" and you'll receive a pop-up from ODBCCONF showing a list of its command-line options. And sure enough, among them is /F which takes a response file as its argument. Basically it's a command stream which is fed into ODBCCONF. So in this case, this loads and runs the Bumblebee DLL, at which point all is lost because a hostile executable has made it into the user's system and has been started.

The Bumblebee DLL injects code into multiple running processes in order to establish a strong foothold on infected endpoints, and the newly launched odbcconf.exe process creates Windows Management Instrumentation calls to spawn two new processes from the wmiprvse.exe, which is the Windows Management Instrumentation Provider Service. Once again, both of these newly spawned processes are existing Windows executables where malicious code is dynamically injected into their process space once they've been started.

So the first of the two is wabmig.exe. That's the Microsoft Contacts import tool. It's injected with Meterpreter agent code. Meterpreter is a Metasploit attack payload that provides an interactive shell from which an attacker can explore the target machine and execute code. It's deployed using in-memory DLL injection. As a result, Meterpreter resides entirely in memory and writes nothing to disk. The second existing Windows EXE that is spawned and then injected into is wab.exe. That's Microsoft's address book app. After being launched, it receives an injection of the Cobalt Strike beacon, which we've covered. We did a podcast on it a while ago.

Bumblebee performs privilege escalation by loading an exploit for CVE-2020-1472 - Zerologon, which we talked about at the time - into rundll32.exe. Bumblebee uses a User Account Control (UAC) bypass technique to deploy post-exploitation tools with elevated privileges on infected machines. Specifically, it uses an existing trusted binary - again, part of Windows - fodhelper.exe. This prevents Windows from showing a UAC window when it's launched. Fodhelper is the executable used by Windows to manage features in Windows settings. Again, living off the land. And it uses this to bypass any appearance of what's going on.

This fodhelper is exploited to run cmd.exe /c space and then rundll32.exe. Then we give it the DLL, which is a path to the Cobalt Strike dll, comma, and then MainProc where the Cobalt Strike DLL is the Cobalt Strike framework beacon, and MainProc is the exported function which Cobalt Strike exports in its DLL in order to run. As we know, Cobalt Strike is an adversary simulation framework used to assist in red team attack operations. Unfortunately, bad guys use it to conduct actual post-intrusion malicious activities. It's a powerful modular framework with an extensive set of features that are used to, you know, you can do command execution, process injection, credential theft, and more.

And speaking of credential theft, after obtaining its foothold and elevating itself to system privilege without any further user interaction or UAC permission, Bumblebee performs credential theft through two methods. The first method is to trigger a memory dump of Windows' LSASS process. LSASS is Windows Local Security Authority Subsystem Service. Within LSASS's memory footprint are the keys to the kingdom, including both domain and local usernames and passwords. They're all sitting in the memory space of the LSASS process. So Bumblebee dumps the memory of this process using procdump64.exe, also living off the land, to obtain access to this sensitive information.

The second method of credential theft used by Bumblebee is registry hive extraction using good old reg.exe. The HKLM SAM, which is the Security Account Manager database, is where Windows stores information about user accounts. That's dumped. HKLM Security, which is the Local Security Authority (LSA), stores user logins and their LSA secrets. And HKLM System contains keys that could be used to decrypt and encrypt the LSA secret and SAM database.

Bumblebee issues three commands of the form reg.exe space save space hklm\sam and then a path to the program where that registry hive should be saved. In this case they give the example c:\ProgramData\sam.save. Then the same command for \system and for \security, saving their dumps into system.save and security.save. So that creates a trio of files containing the dump of those three system-critical registry hives. Then the LSASS dump and those three registry hives are all compressed using 7z and exfiltrated back to command central.

At that point the human operators behind Bumblebee process the retrieved credentials offline, attempting to extract cleartext passwords. The observed time between credential theft and the next activity is about three hours. So this stuff all goes back to wherever. Somebody ruminates on it, figures out what the username and passwords are, but basically reverse engineers cleartext, and then comes back three hours later.

After the attackers have gained a foothold within the organization's network, they gather information using tools such as nltest, ping, netview, tasklist, and ADFind to collect a wide range of information related to the organization. They collect information such as the domain names, users, hosts, and domain controllers. We talked about ADFind in episodes 789 and 790 back in October of 2020. It is a powerful Active Directory exploration tool meant to aid in the administration of Active Directory systems by regular Active Directory admins. Unfortunately, it's been turned against those administrators.

Bumblebee uses Cobalt Strike agent for lateral movement. Their analysis, that is, the Cybereason's analysis, observed multiple connections from the Cobalt Strike process to internal addresses on RDP, Remote Desktop Protocol over TCP port 3389. And of course now they've got credentials for all that stuff, so they're able to log in. After lateral movement, the attacker persists on the organization's network using the commercial remote management software AnyDesk.

After the attacker has obtained a highly privileged username and password, they access the Volume Shadow Service Shadow Copy, which is, again, built in, Windows' built-in facility to create backup copies and snapshots of computer files or volumes while they are in use. So Bumblebee accesses the remote Active Directory machines using the Windows Management Instrumentation command-line WMIC and creates a shadow copy using the vssadmin command. In addition, the attacker steals the ntds.dit file from the domain controller. Ntds.dit is a database that stores Active Directory data, including information about user objects, groups, and group membership. And the file also stores the password hashes for all users in the domain.

In order to obtain maximum privileges on the Active Directory domain, the threat actor executes the following four steps: Creates a shadow copy of the machine file's volume. Lists all available shadow copies, storing the result in a file. Copies the Active Directory database, that ntds.dit file, as well as registry hives containing credentials and sensitive data from the shadow copy. Compresses the output directory for exfiltration. And finally, the threat actor uses a domain administrator account obtained previously to move laterally across multiple systems. After initial connection, they create a local user and exfiltrate data using the open source Rclone software.

Wikipedia describes Rclone: "Rclone is an open source, multithreaded, command line computer program to manage or migrate content on cloud and other high-latency storage. Its capabilities include sync, transfer, crypt, cache, union, compress, and mount. The rclone website lists supported backends including S3, and Google Drive." In the instance observed and monitored by Cybereason, the rclone.exe process transferred approximately 50GB of data to an endpoint with an IP address over TCP port 22 (SSH), located somewhere in the U.S.

So what does all this tell us? The first and most obvious thing we've learned is that you do not want to have your enterprise stung by the Bumblebee loader. It'll definitely ruin your whole day. But speaking of "day," Cybereason compiled the entire event into a timeline. Taking everything we've just stepped through, here's how it stretches out in time. So we have T0 at initial access, when the unwitting user clicks the link, thinks they're doing the right thing, opens the archive, mounts the ISO, runs the program inside. Actually it's the LNK file that'll be contained in the ISO. Reconnaissance using nltest, net, and whoami commands, at T0 plus 30 minutes. When we get to four hours, we're at command and control, loading the Meterpreter agent. Also at four hours is privilege escalation using the Zerologon exploit.

Two hours later at T0 plus six we have command and control with Cobalt Strike beacon execution. Also at the same time credential theft through the registry hive. Reconnaissance 30 minutes later at six hours plus 30 using ADFind, the Active Directory find, ping, and curl. Okay. Now we have a big jump. At T0 plus 19 hours is the credential theft and privilege escalation. That's when we get the LSASS memory dump with procdump64.exe. That was at T0 plus 19 hours. At 22 hours, so three hours later, credential theft. That's where ntds.dit exfiltration occurs with Active Directory full privilege.

Two hours later, now we're at one day later, 24 hours later than the initial T0, is lateral movement using Cobalt Strike SOCKS tunnel over RDP. And data exfiltration using Rclone presumably finished at T0 plus three days. So a full 72 hours from start to finish.

So this thing, as I said, isn't over before it starts. It does take some time. There is a human in the loop at that point where the registry hives and the LSASS dump have been compressed with 7z and sent back to headquarters. Three hours goes by before they've got that figured out, and then they come back to do some real damage using all of the credentials that they've been able to obtain.

So one lesson we learn from all of this is why local privilege escalation vulnerabilities form such a crucial part of this and so many attack chains. Remote Code Execution sounds like the worst possible nightmare. And indeed it's not good. But none of that was needed here. If malware were truly constrained within a user's low-privilege account, much less damage could be done. But the old saying, "If wishes were horses, beggars would ride," reminds us that wishing won't make it so. With everything we've seen of the continuing and apparently worsening trouble Microsoft is having securing Windows, which they refuse to just leave as is and fix because it's obvious they can't do both there is virtually zero chance that once a single piece of malicious software gets loose inside a user's machine that the rest of the organization will not fall.

And this brings us back to that first fatal click, that innocent action taken by a user on the inside, that initiated the collapse of even the most carefully constructed enterprise's security. Those in charge of their organization's security must be living in a state of constant terror over what any one of their employees might do next.

Leo: Rightly so.

Steve: Yes.

Leo: Yes.

Steve: And what we see here with Bumblebee loader is exactly what transpires.

Leo: It's interesting. It's educational once in a while to go through the step-by-step of how these things work. It's fascinating.

Steve: I think it's important because it helps make it real for people.

Leo: Yeah, yeah, yeah.

Steve: It's like, you know, oh, that happened to somebody else. It's like, here's exactly how this could happen to you.

Leo: Yeah. Well, there you go. If you haven't seen the show notes, this would be a good time to get them and look at that step by step and really understand what the threat is. The show notes are GRC.com, that's Steve's website. You also can get of course a copy of the audio. He's got 16Kb audio, 64Kb audio, and very nice transcripts, as well. Plus the show notes. I think the transcripts and show notes are going to give you all the additional detail you need.

While you're at GRC, pick up a copy of SpinRite, Steve's bread and butter, the world's best mass storage maintenance and recovery utility. Version 6 is current, but if you buy 6 today, you get 6.1 when it comes out. And Steve's working hard on that all the time, I know, except when he's reading this book. And then, you know, all bets are off. He's almost done. He's a fast reader, don't worry.

Steve: I am, actually.

Leo: After the fact you can also get on-demand versions at our site, TWiT.tv/sn. We've got audio and video for you. I don't know why you'd want video, but you could get it. Well, I know why, so you could see the video of our Image of the Week, that's why.

Steve: Yeah, the satellites moving. It's spooky.

Leo: Yeah, it's a video.

Steve: It's spooky.

Leo: You also, let's see, what else. Oh, there's a YouTube channel you can watch. In fact, a good way to share clips is on the YouTube channel because they have the feature to do the little clip thing. And of course you can also subscribe in your favorite podcast player and get it automatically the minute it's available. Just search for Security Now!. It's been around, as you know, 18 years now. Everybody should have a copy of it somewhere. Keep it up.

That's actually how Michael first wrote to us. He wrote me a note saying how come I only see the last 10 episodes in the feed? And we just do that to keep the feed from getting big. If we had 885 episodes in there it'd be hundreds and hundreds of megabytes. And we don't want you to have to download that on a regular basis. That's kind of how feeds work, you know, you download it each time. So we keep it to 10 titles. But all the titles are available at Steve's site and our site.

You want to watch us do it live, get the first edition? The first edition, while the ink is still wet? It's easy to do. Just go to live.twit.tv of a Tuesday afternoon, 'round about 1:30 Pacific, 4:30 Eastern, 20:30 UTC, right after MacBreak Weekly. People who are watching live often chat with us live at irc.twit.tv or in the Club TWiT Discord. After the fact you can chat, and people often do, in our community forums.

Steve's got his own forums, GRC.com. We also have ours at twit.community for all of the shows. And we also have a Mastodon instance, which is a federated Twitter without all the security flaws. That is at twit.social. Both are free to join. But I do approve everybody to keep the spammers out, so it might take a day or two for me to let you in. But please do knock: twit.community or twit.social. Steve, have a great week. Go finish your book.

Steve: Thank you, my friend. Unfortunately it will not live out the night because I'm not going to be able to put it down, and I'm close enough to the end, although even at this point I'm on pins and needles. I have no idea what's going to happen.

Leo: You're almost done, and you still don't know what's going to happen. That's a good sign.

Steve: It's really good.

Leo: Exciting, yeah. Well, I just bought it and downloaded it. So I'll be listening soon.

Steve: Cool.

Leo: Actually, as soon as I finish Stacey's Book Club, "Klara and the Sun," which is also excellent. It's about an artificial intelligence, an artificial friend that you bring into your house to be your companion. And it's told from her point of view, so it's really fascinating.