A SOTA State-Sponsored Campaign

Description: Win10's popularity forces another year of free updates. CISA directs all federal agencies to update their UniFi OS devices. CISA gave federal agencies "the weekend" to update Cisco devices. Australia is disturbed by a deeply compromised infrastructure provider. OpenAI introduces Daybreak-powered "Patch the Planet" initiative. Meta's employee monitoring-for-AI-training backfired badly. Script kiddies figure out how to use AI to find vulnerabilities. AI improves with "looping," "repeating," or "iterating." A wonderful story about Kevin Mitnick. Serious hackers mistakenly left a server directory accessible.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1085.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1085-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have lots to talk about. Good news for Windows 10 users. Yes, you're going to get another year. Meta's backed off on spying on its employees. A wonderful true story about hacker Kevin, the late hacker Kevin Mitnick; and the true story of a Fortinet campaign that really was a problem. Steve, I love it when he tells the stories of these hacks. You know, we've heard the news, but now we get the deep details. That's coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1085, recorded Tuesday, June 30th, 2026: "A SOTA State-Sponsored Campaign."

Yes, it's Tuesday. You know what that means. Time for Security Now!. Man, it seems like seven days is too long to wait for Steve Gibson and the latest security news. Hi, Steve.

Steve Gibson: I do see things pass by during the week, Leo. And I think, okay. And I often will jot a note to make sure that I come back to it and talk about it. And the feedback I'm getting from our listeners. Today there was so much to talk about that I think there are a couple listener-inspired things. But I'm going to try to spend more time on feedback, if I can, because it's so great.

Leo: Oh, I love it.

Steve: And I really thank everybody for giving back to us.

Leo: Yeah. We have a wonderful audience, yeah.

Steve: The title of today's podcast would have been too long had I spelled out "state-of-the-art" because I wanted to talk about a state-of-the-art state-sponsored campaign, which we're going to take a look at. Fortunately, "state-of-the-art" has a standard abbreviation, SOTA, S-O-T-A. And then it was interesting because after I had used that abbreviation, it appeared in one of the articles that we're going to talk about. So I thought, okay, yeah, everybody's onboard with SOTA.

So Security Now! Episode 1085 for this last day of June, 2026. We start into July tomorrow. The first thing we're going to talk about is how Windows 10 - yes, 10 - its enduring popularity has forced Microsoft to punt once again and give everybody another year of free updates.

Leo: Wow.

Steve: You know, it had to happen. We're also going to talk about CISA directing all federal agencies to update their UniFi OS devices. We've been talking now for the last two weeks about the expected problems coming, and they came. And so CISA said thou shalt update. Also, once again on a Friday, an edict was delivered from CISA giving basically federal agencies the weekend, meaning don't leave the office, to update all of their Cisco devices that were affected by a different badly exploited problem.

Australia has been disturbed, so says their Inspector General, by a deeply compromised infrastructure provider. And when I read about this I thought, wow, that sounds like this state-of-the-art state-sponsored campaign we're going to be talking about. So that may have, you know, already come around.

OpenAI, not to be left behind for long, has introduced a Daybreak-powered Patch the Planet initiative. Their marketing people, at least, are awake. We're going to talk about that. Meta's employee monitoring all of their employees, or at least a subset, we'll look at that, for AI training turns out to have backfired badly. It was one of those, you know, what could possibly go wrong; and, ooh, it did.

Script kiddies are figuring out how to use AI to find vulnerabilities. What are the consequences? AI is improving itself with a new term we're seeing: looping, repeating, or iterating. What's that about?

Leo: Oh, yeah, everybody's talking looping now.

Steve: Looping is the new buzz, exactly. And I've got a wonderful story I want to share about a friend of ours, Leo, Kevin Mitnick. And then serious hackers mistakenly leave another server directory accessible, which is what leads us to learning about this Russian-based state-sponsored campaign. Which also begs the question, how many other campaigns are there where the directory was not left open by mistake, which allowed us to learn about them?

So lots of fun stuff to talk about. We've got one of our "what are they thinking" Pictures of the Week. So, yeah, I think a fun podcast for this end of June.

Leo: All right. I have a Picture of the Week, and I am willing to look at it together with you. I haven't seen it yet.

Steve: One of our German listeners sent this to me, ran across this and took a picture of it. And I looked at it, and he had some discussion in his email about it. I gave this the caption "How to create a dead end for cyclists."

Leo: I don't even understand what this is.

Steve: And it's the oddest thing because - so we see in the foreground a road, which apparently is cyclist friendly. It's like, come on, guys, ride your bicycles down here.

Leo: That would be this bike path.

Steve: That's right. And it is like a bike path. But, and there's a big, like a cycle sign over on the right to let you know, hey, here's where you should be riding your bikes. That's good.

Leo: Bike path, yeah, yeah.

Steve: But then it veers off to the edge of the road, forcing any cyclist onto some little brick paver area, which then has another sign in the middle, at the end of the brick pavers, says "End."

Leo: That's it. That's it. It's done.

Steve: Yeah. So I guess that's the deceleration lane or something? I mean, so clearly...

Leo: That's crazy.

Steve: ...for whatever reason, bicyclists are not welcome down that road any further. And if you're a sign follower, well, you'll veer off and come to the brick pavers, and then hit the end of the cycling road.

Leo: Wow.

Steve: Now, also there not a lot of cyclists that have been captured by that. I mean, I don't see any.

Leo: Well, it's right at the same place as you're leaving town. So obviously the town loves cyclists, but the rest of them, you know, never mind, forget about it, just drive off here.

Steve: Yeah. And I don't - so where are the cyclists?

Leo: What are you supposed to do?

Steve: Exactly.

Leo: Yeah, what are you supposed to do?

Steve: Where are the cyclists that have been captured by this? It's not clear where they go.

Leo: Wow. That is pretty hostile, actually.

Steve: And when you think about it, it's like, okay, sorry, you know, go drive off the road.

Leo: Go away.

Steve: And now come to a stop because you cannot go further if, you know, if you obey the signage. So a dead end for cyclists, yeah.

Okay. So it's gratifying to see a prediction about something that really should be done come true. Sadly, gratifying or not, that doesn't happen often enough. Our listeners all know how disgusted I've been with Microsoft's continuing attempts to squeeze their Windows 10 users into moving to Windows 11. Many, and for quite some time most, current Windows 10 users evidenced just as little desire to do that as once upon a time Windows 7 users wanted to move to Windows 8. It was thanks, but no thanks. Everything is working fine. Like Windows 7. Just want to stay here. And because Windows 8 is stinky.

So anyway, as we also know, Microsoft arbitrarily, capriciously, and unnecessarily raised the minimum hardware requirements for Windows 11 in a transparent effort to force the purchase of new and now onerously expensive PCs. We know it was arbitrary, capricious, and unnecessary because Windows 11 runs quite well without complaint on PC hardware that lacks every one of those newly imposed so-called "requirements." They can all be bypassed because none of them are actually required.

Against this backdrop, in the summer of 2025 - actually June 24th to be exact - Microsoft reminded everyone that all support for Windows 10 would be ending a few months from then, on October 14th, 2025. There was only one problem with that: Still no one wanted Windows 11, and nearly everyone was still quite happily using Windows 10. So as we covered at the time, Microsoft blinked and gave everyone an additional full year of ESU, their Extended Service Updates, and this is quoting them: "in order to give everyone more time to migrate to Windows 11," they said. Yeah. Or apparently, quite often, to give everyone more time to save up the money needed to purchase a new PC when the one they currently had was running Windows 10 just fine, not having any problems. So this allowed everyone to remain on the ESU plan until October 12th, which is approaching, of 2026, later this year.

So we're back to beating this poor and quite dead horse because time flies, and we're once again here at the end of June; and still, despite reluctantly returning feature after feature, we hear from Paul and Richard every week, like oh, Windows 11 got this feature of Windows 10 that had been taken away. Oh, and it got this feature of Windows 10 that had been taken away. And they rewrote this UI because it was really slow in Windows 11, and now it's fast again. Anyway, even after reverting some of the incredibly inefficient user interface implementations that had been largely responsible for Windows 11's poor performance, no one still wants 11.

And, I mean, there are people who like it. I had to be using it toward the end of the work on SpinRite and also on the DNS Benchmark so that I knew what was going on. So, but, you know, it's pretty. The corners are rounded. But I'll be setting up a new system with Windows 10 because all the evidence I've seen on the Internet says that 10 runs, given the same hardware, much more quickly than Windows 11. And 11 has nothing that I need.

So anyway, on top of all that, thanks to the AI drama that has swept the globe, that new Windows 11-capable PC that Microsoft seems to be pushing everyone toward will now be significantly more expensive to purchase today than it would have even a year ago when people said no thanks. Windows 10's running just fine.

So anyway, you can guess now, as I said at the top of the show, what Windows just did. Yep. Or Microsoft just did. Yep. They blinked again. They once again extended the Win10 ESU program for another year, until October 12th, 2027. So everyone using Windows 10 gets to keep using the Windows they love on the machines they already have. And what's even better is that the continuation of the ESU program means that Windows 10 can and will be the recipient of the results of Microsoft's still unnamed - I would at this time say stubbornly unnamed - Codename MDASH system, which will be cleaning up the mess that was left behind by decades of Microsoft's previous human developers.

So what Windows 10 is getting, when you think about it, is really the best of all possible worlds, since all development on Windows 10 has blessedly been halted years ago. Microsoft will no longer be introducing more new bugs than they remove every month. Instead, the extension of the ESU program for another year will give their new AI model-driven bug discovery and removal system the time it requires to remove the thousands of latent bugs Windows still carries, thus turning Windows 10 into a near-perfect operating system. Like, forever. Thank you, Microsoft.

So given where we are today, I'll make another prediction: Given that the current RAM and semiconductor chip shortage is now expected to endure into 2028, they're not expecting it to resolve this year or next, this is likely to hold PC prices high. Since the recent performance improvements in Windows 11 are finally beginning to allow it to run as well as Windows 10 always has on the same hardware, and since it has always been able to run on that same hardware, I predict we're going to see some form of "Junior 11" which will, for face-saving reasons, strip out some features - maybe hopefully Recall and some of the Copilot+ AI crap - that'd be great, and it will therefore, "surprise!," be able to run anywhere Windows 10 can.

Which will mean that, you know, this will ultimately be the only way for Microsoft to move the remainder of their holdout Windows 10 users over to 11. It'll cost no one anything, it will get everyone back under the same codebase, which actually and understandably is where Microsoft really does need to get them in the long run. I wouldn't expect Microsoft to continue supporting Windows 11, I mean, sorry, Windows 10, forever. But if we get another year of ESUs, people who really want to stay with Windows 10 that they have on their machine will be able to.

And people who want to move to Windows 11, I will be very surprised if we don't have some final capitulation from Microsoft in the form of some Junior 11, you know, it can't have everything that they keep saying you need new hardware for. Something that will allow 11 to run on existing systems with TPM, you know, 1.1, without some of the other unnecessary features that Microsoft is requiring systems to have. And then they can get everybody under a single codebase. I get it that they really do need everybody to be resynchronized. The good news is we'll probably be left with a Windows 11 which is really good and can hold us for, you know, quite a while. So yay.

Leo: Actually, Windows 12 is just around the corner. Aren't you excited?

Steve: Oh, god.

Leo: I'm sorry.

Steve: Unbelievable. Maybe they'll - who knows what they're going to do. But, I mean, clearly 10 refuses to let go; right?

Leo: Yeah, yeah.

Steve: I mean, they're just, you know, people don't want to spend more money, especially now, Leo, where...

Leo: Well, that's what's going on, exactly. RAM is so expensive. Nobody's upgrading their computers anymore.

Steve: Yeah. And so it's unfair to ask people to, like, get more RAM and a new machine for no real reason.

Leo: Exactly.

Steve: And they're going to have to face that sooner or later. So a quick follow-up on the state of the recent Ubiquiti flaws. Since CISA has seen hackers actively exploiting those three flaws in Ubiquiti's UniFi OS, last Wednesday CISA gave all federal agencies three days to apply the available security updates or their recommended mitigations, if for some reason you can't supply the updates. The three Ubiquiti flaws have been added to CISA's KEV, that's that KEV, the Known Exploited Vulnerabilities database. There's 34908, which is an access control bypass flaw that allows an unauthenticated attacker to make unauthorized changes to a UniFi OS system, potentially leading to full system compromise. And we know when they say "potentially leading to," it means, you know, yes, you get to do that.

34909 is once again a directory/path traversal vulnerability, which we never seem to be able to get rid of all those, that allows an attacker to access sensitive files on the underlying operating system, potentially exposing configuration files, credentials, and other sensitive data that could facilitate account takeover; and 34910, an improper input validation flaw that enables an attacker to inject and execute arbitrary operating system

commands, potentially leading to remote code execution. I mean, basically this is a perfect trio of flaws that, I mean, you couldn't get a better set of three, if you want something that allows you to remotely take over a system of any sort. So as we know, Ubiquiti released updates for all three of those vulnerabilities back in May, and Leo's UniFi OS instances, which were all set to auto update, all did.

Leo: Yeah.

Steve: So, Leo, you were never in any danger. Hopefully, everybody else has done this, too. What's changed is that the pace of attacks following their disclosure and/or the reverse engineering of updates, necessitates taking the human out of the update decision loop. Just let automation handle that. Might it screw up? That's a possibility. But the incidences of such screw-ups have always been rare, and we can expect them to become more rare as more of our infrastructure becomes secured. So, you know, there's no piece of Internet-facing system today that I don't have, that has some potential vulnerability that I don't allow to update themselves if the manufacturer says, ooh, crap, we've got to, you know, push this out right now.

Now, that CISA announcement was on a Wednesday. So those people had Wednesday, Thursday, Friday. Last Friday we learned that those three-day CISA BODs, that Binding Operational Directive, you know, "Thou must update!" notices, they also include the weekend, as it turns out. It's not three business days.

Leo: Oh, that's interesting.

Steve: Yeah. It's three calendar days. CISA issued a directive Friday giving federal agencies until Sunday night to patch. And come to think of it, it might be that patching over a weekend would actually be easier, right, since the network would presumably be much quieter with fewer, if any, people disturbed by an update and maybe a necessary reboot of the system.

I did see something that I've never mentioned because it just kind of flipped by a few weeks ago. It was a mention that the idea of the necessity to reboot is being rethunk by the industry because it's understood that the need to take down a piece of border equipment. And after all, it's the equipment on the border that is the stuff that's under attack; right? The need to basically shut down the network during what could be a lengthy update and reboot is a reason that IT doesn't.

So who says you have to reboot a system? I mean, we have seen Microsoft beginning to inch toward some no-reboot-needed updates. And all of this necessity, this whole idea of needing to boot a fixed piece of firmware, it's only legacy. I mean...

Leo: Really. You don't have to reboot?

Steve: No.

Leo: Interesting.

Steve: There is no reason that a system could not have been structured so that you could have two instances, for example, of a library, and briefly switched the pointers from the old one to the new one so that basically no one would even notice that you were now operating under the new library. So this whole concept of needing to take the whole system offline and then bring it back up again, that's really old school. And so I think what we're going to begin to see is a - and what a selling point; right? I mean, if you had three pieces of equipment you were choosing among, you know, Juniper and F5 and Palo Alto Networks, and Juniper was able to say, hey, we have zero reboot updates, we will update your system with no downtime, and the other two guys didn't have that? Well, that's a selling point. So you can imagine we're going to be seeing that in the future.

Leo: I would love that. I didn't realize it wasn't possible.

Steve: Yeah, definitely is possible. There's no reason, well...

Leo: So that would be true of operating systems of all kinds; right?

Steve: Sure.

Leo: Windows, Linux, everything. Huh.

Steve: And look at the 0patch guys.

Leo: Right.

Steve: They do zero reboot patching on the fly. So definitely something that could be done.

So the story behind in this case this single high-severity flaw which was on Friday, on last Friday, CISA said everybody, every federal agency, must have updated by Sunday night. So as of yesterday, all federal agencies need to have updated this Cisco deal. This was a server-side request forgery. The CVE is 20230. It was discovered in Cisco's Unified Communications Manager Server. They released security updates to address the flaw three weeks earlier, on June 3rd, and at the time they warned, because they knew, that exploitation could give attackers root privileges on the device.

They wrote: "A vulnerability in Cisco Unified Communications Manager" - the Unified CM, they call it - "and also Cisco Unified Communications Manager Session Management Edition" - which of course is Unified CM SME - "could allow an unauthenticated" - meaning no credentials - "remote attacker to conduct server-side request forgery (SSRF) attacks through an affected device. This vulnerability is due to improper input validation for specific HTTP requests. An attacker could exploit this vulnerability by sending a crafted HTTP request to an affected device. A successful exploit could allow the attacker to write files to the underlying operating system that could later be used to elevate to root."

So that was then, June 3rd, when they announced the update and offered patches and said this is important, critical, do it. Three weeks later, that vulnerability is now being actively exploited. Three weeks. So this is not even a monthly patch cycle deal. This is,

you know, you need to do this if you want to keep bad guys out of your system. And we know that Cisco has had a legendary problem of keeping bad guys out. So that's not a lot of time. On the other hand, it did take three weeks. So hopefully, whoever's in charge of updating today, here we are middle of 2026, did not wait until CISA gave them no choice with their Binding Operational Directive, since in this case CISA's BOD, B-O-D, was issued several days after attacks had been detected in the wild. Because after all, KEV is Known Exploited Vulnerabilities.

So it's clear that CISA has seen the light regarding the need for speed in response to these. Remember that flowchart, that tree, the decision tree chart that we looked at last week, had - it had many of the leaves of that tree demonstrated that they "get it" because there were three days-to-patch response times on many of those decision endpoints.

So more than anything, I really do hope that the word is filtering out that everything we have known - and this is the problem, you know, institutional inertia and just conceptual inertia, historical inertia, everything we've known about the dynamics of vulnerabilities, exploitation, attacks, and patching has been thrown up in the air. It's unclear when or how it's going to settle down, but what is clear is that nothing will be as it has been before. AI has changed all that. You know, I'm seeing many predictions around, you know, through the industry, in the press, the popular press, the tech press, of a coming onslaught of massive AI-driven cyberattacks. And as I've said also, it seems to me that's less likely. I guess I would say massive AI-driven cyber vulnerabilities.

But vulnerabilities are different than attacks; right? Because it's unclear to me how attacks make money. Not like broad, huge, hundreds of millions of users affected, you know. Cryptocurrency money is entirely the name of the game. What I expect to be more likely is many more successful network penetrations followed by extortion. And the bad guys are increasingly likely to attack those enterprises, as we've seen, that really must protect their exfiltrated data. You know, a couple weeks ago we saw that large law firm that made a, what was it, a $2 million ransom payout which was 1% of their annual take of $200 million. Which payout was defensible and sane because the cost to them in reputation and client lawsuit damage of not paying the ransom and hoping that their data isn't leaked, would just be too great.

So the problem with changing updating habits is, as I said, the great weight of institutional inertia. The hope is that the AI attack hysteria, which I think is what it is, which I doubt will materialize, may be what the IT department actually needs. They need the hysteria in order to obtain the resources they require to be able to update with, you know, much more speed, much more nimbly. So we can hope that that's the way, that's the shape this takes that, you know, the boss hears that oh, my god, the AI's coming to get them. So when IT guys say, hey, we need a couple more people whose job it is to do nothing except to keep all of our equipment updated, so we're not attacked by the coming AI tsunami, the boss is going to say, okay, yeah, go get them, instead of saying, you know, oh, I don't know, can't you have Moe just do that, too? Moe's already overworked, Leo.

Leo: Moe's a busy guy, especially on weekends, apparently. So I'm still puzzled, I mean, I think you - if you're going to modify the kernel code, I think you have to reboot.

Steve: No.

Leo: How would you do it without restarting the machine?

Steve: You just have to have in a - okay. So a kernel is normally a bunch of libraries. I mean, there's a microkernel, and then a whole bunch of kernel drivers.

Leo: Right. Oh, I guess you could modify kernel drivers without rebooting.

Steve: And in a microkernel, you know, like so you may have the memory management API. So the only thing you need is for there to be a moment when no threads are in the memory management API, and you can just switch to newer code that runs the same API. And now some use-after-free vulnerability is gone that the previous code has. So, you know, I guess for me I see it so clearly because this is where I program is in assembly language.

Leo: Yeah, you're in the kernel, yeah.

Steve: It's where that is all happening. But there really isn't anything that precludes an on-the-fly switch-out of old code for new.

Leo: So you've got a microkernel running right now.

Steve: Yeah.

Leo: And you would just say, okay, here's the new kernel. Halt the code and jump to the new microkernel.

Steve: Yes, switch the threads over to the new microkernel.

Leo: Right. You wouldn't...

Steve: And they don't know the difference.

Leo: Everything would have to be item potent, though; right? I mean, it'd have to be reentrant.

Steve: Correct. So...

Leo: That's part of the problem is I'm sure a lot of it's not reentrant.

Steve: Well, so as soon as the threads are out, then you don't have any - so a...

Leo: That code is dead. It's not running.

Steve: Yeah. In a microkernel there is like memory management is one of the core functions of any kernel. And so if at any point there are no threads that are actually doing work in there, then you simply switch...

Leo: You say go to that one.

Steve: You just - yes. Then the next thread that comes along that wants to do an...

Leo: But what thread is doing that? There is a thread that is doing that switch. That's running. But I guess you just let that die when it's done.

Steve: Oh, so you would have a supervisor that would be in charge of swapping out old code for new code.

Leo: They're speculating in the Discord, and I think this is probably accurate, that most operating system companies kind of think it's just a good idea to reboot once in a while. Like...

Steve: Users think it's a good idea to reboot once in a while.

Leo: Well, even the operating system companies know that there's stuff in the memory that probably shouldn't be there. There's memory leaks that they wish weren't there. But...

Steve: Yes. The technical term is "cruft."

Leo: Cruft, yes. So, you know, rebooting once a week isn't the end of the world, although...

Steve: And we've talked about - and we've talked about how rebooting your router can help to flush out malware that is not able to obtain persistence.

Leo: But I'm completely sympathetic with a network engineer who says I'm not bringing the network down. I don't care if it's 3:00 in the morning. I'm not taking the network down.

Steve: I do it at home because I've got so much crap now that's, like, on the Internet, it's like, oh, what's going to happen if I, you know, blah, you know, what if I get a new IP address? Oh.

Leo: Right. I've got system timers running all hours of the day or night. I'd have to look and make sure that that stuff - because if one doesn't run...

Steve: Oh, and Leo, if your AI agent was unable to blog when it wanted to, it might [crosstalk]...

Leo: It could be in the middle of a blog. It could blog any time of the day or night. I don't know when it's blogging.

Steve: That's right.

Leo: I'd have to say, "Hey, Quicksilver, are you in the middle of anything right now? Just let me know because I'd like to reboot right now." Man. That's interesting because nobody does this, really, that I know of. Maybe there are some mission-critical systems. I'm sure, you know what, I'm sure the space shuttle doesn't reboot.

Steve: Right.

Leo: Or didn't reboot. I'm sure the International Space Station doesn't have to reboot. I mean, there are mission-critical systems that cannot restart.

Steve: Right. And it's only in sci-fi that they say, okay, everybody hold onto something, we're going to have to shut down gravity while we reboot.

Leo: Wow. Joshua337 in our Discord says, "I had a Cisco switch up for 19 years." Wow.

Steve: That's nice. That's nice.

Leo: That's because he never patched it. Would you like me to do an ad right now?

Steve: That'd be good.

Leo: Okay. I apologize. There's somebody drilling outside.

Steve: Yes.

Leo: But that's all right. This is life in the little city, the small town we call Petaluma.

Steve: It's not loud for us because those...

Leo: You don't hear it? Okay, good.

Steve: Because those mics are really good.

Leo: I have a lot of noise suppression going on in various spots. I hear it. We'll get back to Security Now! in just a bit. I know you...

Steve: Sounds like an old matrix printer going back and forth.

Leo: Eh-eh. Eh-eh. Oh, there's a sound I don't miss. And before that the teletypes. Every radio station had ch-ch-ch-ch-ch-ch-ch going on all the time.

Steve: At least you knew when it was done. You didn't have to, like, go over and check.

Leo: Right, yes. It got suddenly quite in here. And you'd buy - you'd buy these big enclosures to put the teletype in so that it would be somewhat quiet.

Steve: Oh, padded booths. They had their own padded booths.

Leo: Yeah, yeah. I'd still be noisy. And then if it's a big, like a big story, something big happened, the bell would ring. And if it rings five times, man, you run over to that AP...

Steve: Ding ding ding ding ding ding ding ding.

Leo: Oh, we've got a hot one.

Steve: So last Wednesday, Mike Burgess, who is the current Director-General - I called him the Inspector General earlier, but he's the Director-General, of Security and the head of ASIO, which is the Australian Security Intelligence Organization, published his annual Threat Assessment for this year, for 2026. It was not at all cyber-specific, talking about many other social aspects which impinge upon Australian security. You know, lots of foreign actors and countries that are unhappy and so forth. But there was a section regarding threats to Australia's Critical Infrastructure, and it was a doozy.

Mike wrote: "Critical Infrastructure: The third matter we dealt with can also be a 'threat to life' in extreme circumstances. We discovered nation-state hackers had compromised the network of an Australian critical infrastructure provider. ASIO assessed the hackers were preparing for sabotage. They weren't planting 'digital dynamite' as such; they were mapping out the network and maintaining access so they could cripple it at a time of their choosing. Cyber sabotage is an evolving threat, and I have established dedicated teams to counter it. As ASIO's understanding grows, so does our level of concern.

"The scale of this activity, led by one nation state in particular, is difficult to overstate. You and they would be surprised how extensive our warrant coverage is. We struggle to find a single country in our region that has not been compromised by this state's cyber apparatus. Critical infrastructure in the energy and communications sectors, as well as infrastructure supporting the military, are top targets. In this case, a state-sponsored group did not just achieve access to the Australian critical infrastructure provider, it successfully acquired credentials - log-in details and passwords - for active users of the networks, including the IT professionals guarding it. ASIO identified, tracked, and

attributed the hack, and worked with the victim company and our security partners to remediate the compromise - work which is still ongoing."

So as I said, I mean, so that's like, whoa. This is what countries are facing.

Leo: By the way, our resident Australian says they pronounce it A-zee-o.

Steve: Oh, ASIO, A-zee-oh.

Leo: Yeah, A-zee-oh, yeah.

Steve: Actually, that makes sense, too, because they use esses where we use zees. Like organization is, you know, -nisation.

Leo: Right. Although knowing Aussies, he could be pulling our leg. But I think Darren's saying it's a long A, ay-zee-o.

Steve: Ay-zee-o.

Leo: Ay-zee-o, yeah.

Steve: Ay-zee-o. Anyway, so I encountered this report, as I noted, after fully digesting and laying out this week's main topic. So when I saw the way the intrusion into Australia's infrastructure provider was described with that full credentials and login and everything, I noticed that it exactly corresponded to what we'll be examining as today's main topic. And there are so many intrusions in that state-sponsored campaign that I wouldn't be surprised if this was one that Australia's unnamed infrastructure provider got swept up in. So we'll be sort of circling back to this by the end of the podcast. But interesting that there's a view from the victim's side where they said, ooh, wow. We are really in trouble here.

Okay. So last Monday, a week ago and a day, not to be outdone by Anthropic with Mythos 5 and Fable 5, OpenAI announced their initiative dubbed "Patch the Planet." This uses their Daybreak system, which we noted a couple weeks ago they announced to sort of be their response to Mythos, Anthropic's Mythos. And this whole system has already been producing results, which I'm going to share in a minute.

Their announcement said: "We are introducing Patch the Planet, a Daybreak initiative built with Trail of Bits to help maintainers strengthen the critical open-source software the world relies on. We're pairing AI-assisted security research using our most cyber-capable models with expert human review to not only identify vulnerabilities, but help patch them.

"AI is accelerating vulnerability discovery, but discovery alone does not protect users. Many maintainers are already being asked to sort through more reports, more quickly, with the same time limit and resources. Patch the Planet is built to reduce that burden, not add to it. Security engineers review findings before they reach maintainers, work with projects to develop patches and tests, and build reusable workflows that help teams continue improving security after the first fixes land.

"Trail of Bits has committed their entire security research organization toward this effort for our initial surge. They're working directly with maintainers to investigate and validate vulnerabilities, develop and test patches, and coordinate disclosure of vulnerabilities. Additionally, we will be partnering with HackerOne" - of course the famous bug bounty offering - "and Calif, who are helping us take our efforts further with vulnerability triage, coordinated disclosure, and additional focused vulnerability discovery efforts."

So how does Patch the Planet work? "Each engagement under Patch the Planet begins in consultation with the maintainer." So like the maintainer of a specific project; right? They said: "For each collaboration, security engineers work with maintainers to understand each project's needs, preferences, and where additional security effort would be most useful: vulnerability validation, patch development, CI/CD improvements, or longer-term security engineering. Once aligned, researchers investigate potential vulnerabilities, validate meaningful issues, develop or refine patches, support testing, and coordinate disclosure through the project's established channels." So it's interesting. This feels more hands-on, more human aimed and managed, you know, it's not just a, you know, aim the AI at it and stand back kind of approach.

They said: "Initial participants include cURL, NATS Server, pyca/cryptography, Sigstore, aiohttp, the Go project, freenginx, Python, and python.org. These projects support widely used networking, cryptography, software supply chain, and language infrastructure, where stronger security can benefit a broad range of downstream products and services. Additional projects will join in future rounds." So again, they're also not doing everything at once. Because their human side resources are limited, they've chosen a bunch of projects, and they are working closely with the maintainers of that code.

They said: "Trail of Bits has dedicated security engineers to work full-time with Codex and GPT 5.5 Cyber across 19 open-source projects, and has already identified hundreds of security issues and merged dozens of patches, with many more still undergoing coordinated disclosure. The initial sprint also produced reusable security infrastructure: fuzzing harnesses, historical CVE analysis pipelines, differential testing systems, threat models, expanded test suites, and workflows for deduplication, false-positive filtering, severity correction, and patch generation. Some project-specific details will be shared later as testing, remediation, and coordinated disclosure progress.

"A few early examples show what the team was able to build and find: A fuzzing lab in less than a day. Trail of Bits engineers used" - here it is - "repeated Codex/goal runs with GPT 5.5 Cyber to build an entire fuzzing lab covering dozens of entry points, variant builds, platforms, and novel test seeds. Engineers set the objectives and refined the prompts. The system then used coverage feedback to keep expanding into new surfaces, target edge cases, and filter weak or invalid candidates.

"Trail of Bits engineers found that, with limited guidance, GPT 5.5 Cyber made useful choices about where to expand coverage, which builds and entry points to probe, and which candidates were too weak to pursue. The completed setup took less than a day. Trail of Bits estimates that building the same lab manually would ordinarily take at least several weeks," rather than less than a day. And it wouldn't have been as much fun; right?

They said: "A reusable pipeline for finding variants of known vulnerabilities they also achieved. The team built an end-to-end system that ingests historical CVEs, extracts relevant vulnerability patterns, searches target codebases for related flaws, and sends candidate findings through specialized judging agents. The pipeline deduplicates results, filters likely false positives, and routes the strongest evidence to security engineers for manual confirmation. This turns years of public vulnerability history into a repeatable search strategy that can be applied across projects. Trail of Bits found the models

especially effective at this kind of variant analysis, which uncovered many additional issues across the codebases under review."

Okay, now, you know, just sort of stepping back from this, if this was posted this time last year, these details would have left our mouths hanging open in wonder and disbelief. But now, today, our reaction is "Okay, sure, what else?" And as it happens there is else. They wrote: "Differential testing in days instead of weeks or months were created. Different implementations of the same protocol should usually behave the same way under the same inputs." Thus differential testing; right? "When they diverge, one may contain a bug. Applying this idea at scale is normally difficult because engineers must write custom shim and glue code connecting each implementation to a common test harness. Codex generated and iterated" - there's the word again, iterated - "on that code, allowing multiple implementations to be fuzzed against one another and their behavioral differences investigated."

And again, I'll just highlight that we're hearing terms like "repeated" and "iterated" more and more. We'll be talking about looping here a little bit later. What we're collectively learning is that AI gets better when it iterates over problems.

So they continue their posting: "The workflow filtered many weak or invalid results and produced a comparatively high-signal set of candidates for expert review. The team reached those results within days, compressing work that has historically taken weeks or months. Trail of Bits is continuing to expand and refine these tests before publishing project-specific details." Basically so what we're seeing is there's like a meta outcome from this work. To have the AI, they're learning how to apply the AI across a set of 19 open source projects. But the result of these learnings, these - god, I just used that word - is a set of harnesses and approaches that end up being persistent. That is, the things that they're developing are ways of harnessing AI that are inherently reusable.

They wrote: "Security engineers reviewed every finding before it reached a maintainer. Trail of Bits engineers manually reviewed every security issue before it was submitted to a maintainer, and the added value of this step cannot be understated. While frontier AI models are highly capable of finding vulnerabilities and patching them, they also produce a high volume of false positives that can contribute to the already overwhelming backlog maintainers are facing.

"Patch the Planet solves for this by having dedicated Trail of Bits researchers reproduce the evidence, check findings against project-specific documentation and threat models, remove duplicates, reassess severity, and prioritize confirmed vulnerabilities for remediation. They also develop and submit patches in accordance with maintainers' preferences. Maintainers remain in control of what patches are deployed and how disclosure is handled.

"What OpenAI Daybreak is already finding: Patch the Planet builds on a broader body of Daybreak work showing how frontier models can help defenders find, validate, and remediate serious vulnerabilities in widely used software.

"We're sharing a few early highlights here, while withholding exploit mechanics and project-specific details where disclosure is still underway." Meaning, once again, as did Anthropic before them, they found a bunch of stuff they can't talk about because they need to go through the responsible disclosure approach and wait for these things to get fixed in the field. They said: "As fixes land and coordinated disclosures conclude, we plan to publish deeper technical reports that walk through individual findings, research methods, validation workflows, and lessons other defenders can apply." Right? So as I said, the things they're learning from this end up having long-term, much wider application. They don't want to release that yet because it is still too powerful.

So they said: "Our findings span every layer of the software stack, with many more still in the disclosure process." So here's what they have found so far. Of operating systems: "The Linux Kernel: GPT 5.5 Cyber identified security-relevant components across more than 30 million lines of code, flagged potential security issues, and then validated them dynamically, generating eight kernel pointer information leak proof-of-concepts and 24 local privilege escalation exploits. We noted that hundreds of issues were identified. This is the subset for which proof-of-concepts were automatically generated." So 30 million lines of code from the Linux kernel. They've found eight kernel pointer information leak proof-of-concepts, meaning validated, verified; 24 local privilege escalation, validated, verified, out of hundreds more that they're still working toward.

"Under OpenBSD," they said, "Our models identified a 23-year-old use-after-free in OpenBSD's kernel implementation of System V semaphores. OpenAI researchers reproduced the issue and confirmed that it would allow an unprivileged local user to escalate privileges to root." What about FreeBSD? "Security researchers at Calif used Codex to find and validate using proof-of-concept exploits for several LPEs (local privilege escalation) in FreeBSD. Across a broader FreeBSD campaign, OpenAI researchers confirmed 34 vulnerabilities and produced seven local privilege escalation PoCs (proofs of concept)."

And for networking, "dnsmasq: Codex Security independently identified vulnerable patterns corresponding to four of the six dnsmasq CVEs which were later fixed in 2.92rel2." The HTTP/2 Bomb that we talked about the last couple weeks: "Calif used Codex to identify 'HTTP/2 Bomb,' a denial-of-service technique affecting major HTTP/2 implementations including NGINX, Apache, IIS, and Pingora. Calif's analysis suggested that more than 880,000 Internet-facing websites were running affected server software with HTTP/2 enabled."

Now, that was interesting to me, and also deeply annoying. Those are the jerks we looked at a couple of weeks ago, who bragged about the discovery of this protocol failure vulnerability and released its information, including a working proof of concept, in a complete lack of coordinated disclosure. They essentially said "AI has changed everything such that coordinated disclosure timelines no longer apply." Meanwhile, those in charge of web server operation were scrambling in a panic which could have been avoided with just a little bit of courtesy. I'd love to see Calif's access to Daybreak rescinded since this is not the way it was supposed to be used. I was a little annoyed to see that they apparently are an active participant in this. Again, I'd love to see that change.

Anyway, what about browsers? OpenAI continues: "Chrome: OpenAI researchers found and reported 5 exploitable vulnerabilities in Chrome's V8 JavaScript engine, including three that were identified and remediated within days of being introduced. Safari: In roughly a week of focused WebKit work, over 10 exploitable Safari vulnerabilities were found and reported. Firefox: OpenAI Preparedness identified a WebAssembly vulnerability" - which happened to be CVE-2026-8390 - "with GPT 5.5 during safety evaluations that Mozilla patched two days before Pwn2Own Berlin."

Them patching it two days before Pwn2Own Berlin, thanks to GPT 5.5's work, prompted five of the six registered Firefox entries to withdraw from the competition because AI beat them to it. "No Firefox exploit was successfully demonstrated at the competition." Which is very cool. You know, what this is, what we're seeing, is a relatively, certainly comparatively rapid tightening up of the world's software. This is what that's going to look like. Pwn2Own will no longer have anything to pwn and then own.

They said: "Open-source software is shared infrastructure." Indeed, you know, log4j, for example. "Securing it should be shared work. AI is changing the pace of vulnerability discovery, and the work now is to make sure the benefits reach the maintainers and users who need them most.

"Patch the Planet is designed to put that full defensive loop in service of maintainers: discovery, validation, severity review, disclosure, patch development, testing, and deployment. Frontier models can make parts of the loop faster, but the aim is to give the people responsible for shared infrastructure" - meaning the maintainers - "better tools and more capacity, while preserving their agency over how changes land." Again, Calif did not do that for the maintainers of HTTP/2. They just said, oh, look what we found. Woohoo.

"The first sprint," they wrote, "shows that sustained collaboration among maintainers, security engineers, and AI-assisted workflows can produce immediate fixes, stronger project infrastructure, and reusable security work that can continue improving open-source software over time. This," they conclude, "is just the beginning. As more fixes land and coordinated disclosures complete, we plan to publish deeper technical reports on selected findings, the methods used to discover and validate them" - in other words they're going to show how the AI was harnessed in order to do this, and the workflows defenders can adapt to help protect the software everyone depends upon. "If you are a maintainer, you can apply and join Patch the Planet." So I've got a link to the Patch the Planet page in the show notes. It's trailofbits.com/patch-the-planet.

So Daybreak was a bit delayed, as we know, relative to Claude Mythos Preview. And it appears that, as we might expect, its approach differs in the details. But the evidence clearly suggests that OpenAI is not out of the game by any means. And that's great news for everyone. Very, very cool.

Leo: Yeah, very interesting, too.

Steve: So they join Anthropic with the, you know, Claude Mythos Preview work to turn their attention, and they're finding bugs.

Leo: So is this Patch the Planet the equivalent of Anthropic's Glasswing?

Steve: Exactly.

Leo: Okay.

Steve: It is the equivalent. Where it differs is that Glasswing was also offered, I believe, to non-open source maintainers.

Leo: Yes. That's right. In fact mostly non-open source. Microsoft and people like that.

Steve: Right, right. And I paused because I also know that Mozilla got it and fixed hundreds of bugs using Mythos.

Leo: Right. Some open source, sure, sure, sure.

Steve: So some open. But so far this looks like it is - the Patch the Planet, basically OpenAI is saying we are so dependent upon open source. And also note that this does give them and their partner Trail of Bits something that Glasswing didn't have. Because it's open source, they're able to turn this loose on publicly available source. When you give a private company that has closed source, you're basically just saying we're giving you access to Mythos. We don't have your source. You have your source. So we're not going to be able to see nearly as much into how you're using Mythos to obtain results. So it's a different approach that has a different set of tradeoffs. Anyway, but yes, it is their equivalent. So both of these two big guys with state-of-the-art frontier AI are now working, proactively working to clean up the install base of software, in the case of Patch the Planet with 19 public projects.

And you know, Leo, the other thing that's going to help to clean up the planet?

Leo: More coffee.

Steve: Is, yes, it will keep the planet spinning.

Leo: Oh, I like your new Contigo mug there. That's a pretty little copper thing. Is that new? Is that a - yeah.

Steve: Yeah.

Leo: It's coffee-colored.

Steve: Yeah.

Leo: So it's appropriate. Okay, Steve. On with the show.

Steve: So we talked briefly, and it only deserved a brief mention before, but oh boy, about Meta's clearly misguided plan to record all of their employees' keyboard, mouse, and screen activity for the, like, just streaming surveillance from every PC for the ostensible purpose of training AI of some sort. At the time I quipped that it would be weird to have AI looking over our shoulders, as it were, you know, training on our own work. Seemed like training our own replacement. But in classic "what could possibly go wrong" failure, it was worse than that. Last Monday, WIRED picked up and covered the adventure under their headline "Meta Exposes Data Internally From Its Controversial Employee-Tracking Program."

Leo: Oysh.

Steve: I know. And WIRED had the teaser "Employees had previously raised concerns about the initiative, which involves collecting workers' keystroke data to train AI models." Oh, boy. WIRED wrote: "Meta left potentially sensitive information collected from employee laptops accessible to anyone inside the company" - and, you know, it's not a small company - "according to an internal security notice seen by WIRED and three current employees familiar with the issue. The data, which was collected as part of a

divisive initiative to train artificial intelligence models, is believed to include keystrokes, mouse clicks, and content displayed on the computer screens of Meta's U.S. employees."

Leo: Wow.

Steve: Like I said, literally a surveillance stream pouring out of every Meta employee laptop. It's like, what could possible go wrong? And they left it in the open. Like, oh. Wow.

"Meta spokesperson Tracy Clayton initially confirmed to WIRED that the company is investigating the security issue. As this story was being published," WIRED wrote, "he added that Meta is pausing the data collection program indefinitely." Wait. Can you have an indefinite pause, Leo? Does that mean it's indefinite how long the pause will last? Or it's an indefinite pause, meaning it's a pause, we're calling it a pause, but we killed it. I don't know. Anyway: "Clayton said: 'We have carefully designed this program'" - I just love bullshit. "'We have carefully designed this program with privacy safeguards.'"

Leo: Of course.

Steve: Of course. Why wouldn't we? And besides, I've been told to read the statement. "'And while we have no indication at this time that any data was improperly accessed by Meta employees, we're pausing it while we investigate.'" It sounds like a temporary maybe.

"According to documents viewed by WIRED, the security notice sent out last Monday indicated that 'employee data across 45,000 hive tables,' had been exposed. Those tables included employee activity such as 'full prompts and transcriptions, private conversations, people and performance data.'" So, wow. Big Brother much? Basically, apparently, employees are being fully and continuously surveilled with all of that mass of data collected for "AI Research."

Anyway, WIRED's article continues, saying: "Some employees at Meta quickly seized on the security failure, saying in internal forums that it validated concerns they had raised when the company began tracking users' corporate laptops in April as part of a program known as the Model Capability Initiative (MCI).

"Comments about the incident posted on internal forums Monday included questions about how Meta's privacy reviews failed to prevent the breach, and whether everyone whose data was potentially exposed will be allowed to attend a meeting going over what went wrong, according to posts seen by WIRED. In one internal forum where staffers are known to trade jokes, an employee posted a meme from 'The Office' of the character Jim Halpert holding a sign that reads, '0 days since our last nonsense.'"

Leo: Oh, love the memes.

Steve: "Sources at Meta, who were not authorized to speak publicly, tell WIRED the incident has now been marked as closed, meaning it was likely resolved.

"In an internal posting responding to employees' questions on Monday seen by WIRED, Andrew Bosworth, Meta's chief technology officer" - their CTO - "said that the tracking program's implementation had fallen short of the standards outlined in its privacy review"

- wow, corporate speak - "and that findings from the incident would be shared. Bosworth noted: 'Here we had misconfigured ACLs'" - access control lists - "'and we need to understand how that happened, track down every data access and understand it.'" Right, because there's so much there to understand, Leo.

Leo: Yes, well, very important, yes.

Steve: "A couple of months ago, Bosworth told employees concerned about potential data leaks that the tracking program is 'tightly controlled' and uses the same protection standards, storage systems, and access controls as other sensitive datasets" - ooh, that's not good - "according to internal posts seen" - like this is as good as we can get it, and it's bad, apparently. "Last month, more than 1,600 Meta employees signed an internal petition protesting the laptop surveillance effort, warning that 'collecting this data introduces both security and regulatory risks for Meta, including the potential for breaches and unauthorized disclosure.'

"The petitioners also expressed concerns with what they viewed as a lack of safeguards that Meta had put in place. One engineer also wrote a widely shared internal note saying having their laptop screen scraped for training data without their consent felt like an invasion of privacy and amounted to exploitation." Right. That's how everybody felt about Recall initially; right?

Leo: Yes.

Steve: "Meta executives have previously defended the data-gathering project, saying it was necessary to train AI systems to use computer software the way humans do."

Leo: How else are we supposed to replace our customers?

Steve: That's right. We have to train on the people doing the work.

Leo: I mean our employees. Yes. How else are we supposed to fire everybody? Come on. Give us a break.

Steve: And I love this. I love this, Leo. "In audio of a company meeting leaked last month, Mark Zuckerberg" - you know, that humanist - "told employees that 'AI models learn from watching really smart people do things.'"

Leo: Yeah.

Steve: "'And the average intelligence of the people who are at this company is significantly higher.'" Wow.

Leo: And in our annual it'll be even higher, and then we can [crosstalk], yeah, yeah.

Steve: So he said: "'Even higher than the average contractor who could be hired specifically to produce this kind of data.'" Right. We would hire contractors and spy on them instead of our own employees.

"But after widespread protest from employees, Meta this month began offering more exemptions to the monitoring, including letting staffers briefly turn off the surveillance so they could complete sensitive tasks, such as scheduling a personal appointment, according to two people familiar with the matter. Some employees are still demanding that the tracking be stopped altogether." Apparently we have a pause of indefinite duration, whatever that means.

"Meta faces more regulatory scrutiny about data security than most companies." Deservedly so. "It's subject to a U.S. Federal Trade Commission consent decree that expires in 2040 requiring it to maintain processes to avoid breaches." Well, that would be nice. "But current and former employees have told WIRED that the requirements are inadequate and outdated. Meta also has also begun offloading some work reviewing programs and features for potential privacy and security risks to artificial intelligence." That's right. Ask the AI if we're doing enough. "It wasn't immediately clear whether AI played a role in the access control issue" - whoops - "with the MCI data.

"The security incident will likely contribute to the ongoing morale crisis at Meta, where employees have been frustrated by the past few years of mass layoffs, a turbulent reorganization, and an all-out push to develop AI models and features. In March, Meta created a new Applied AI team and moved some 6,500 employees into new roles focused on improving AI models. Some Meta staffers have described the projects they've been assigned as menial and 'soul-crushing.'"

Meanwhile, "Bosworth sent out a memo to employees last week apologizing for the company's 'atrocious' communication about the AI reorg and promising improvements, including clearer communications and a return of some office perks." Oh, wouldn't that be nice. Fresher coffee.

Leo: Yes.

Steve: Wow. So, okay. Meta does not seem like an employee-friendly place to work.

Leo: No.

Steve: But I'll confess to being able to see both sides of this. First, certainly, the idea of essentially sucking in everything every employee does is inherently creepy. And the question of its secure storage is the first thing that springs to mind. And as I said, in that sense it's identical to the reception Microsoft received when they introduced "Recall." Everyone's immediate reaction was: "Uh, and how exactly are you going to absolutely positively keep all of our screen history safe forever?" And on top of that Microsoft had to arrange to NOT capture anything that might actually be sensitive, like on-screen passwords and credit card numbers that people were entering. So the whole idea, right, is inherently fraught with risk.

Okay. So before I examine the other side of this argument, just so we're very clear, I fully get it that streaming into storage somewhere every key press, every mouse twitch, and every screen image experienced and created by a mass of employees is just asking for trouble, not to mention being an astounding invasion of privacy. In the past, we've examined the amount of (or lack of) privacy an employee using company bandwidth on

company computers in a company's facility should reasonably be able to expect. And we've seen the need for an enterprise to make whatever it's doing with regard to monitoring its own network and thus indirectly its own employees, at least very clear. Make it clear. But Meta's recorded surveillance of every twitch is taking that to extremes.

One question I had was whether Mark Zuckerberg and other C-suite executives were also participating in this grand surveillance experiment, you know, the brain suck. Or had they perhaps politely excused themselves from the same super-secure surveillance that everyone else was subjected to? After all, if the AI is supposed to be training on the smartest people available, who better at Meta than the C-suite executives at the top of the pecking order?

Leo: I guarantee you Mark wasn't getting spied on. Guaranteed.

Steve: I guarantee you, yes.

Leo: They're not replacing him anytime soon. Wow.

Steve: Okay. So with the horrendous policy consequences acknowledged, I want to explore the flipside. With a brand new technology, such as these massive large language model neural networks, you really don't know what you can do until you try. Since the truth is we stumbled upon the AI effect, as much as we deliberately designed it, the past several years of explosive AI growth has been a testament to the "let's try this and see what it does" approach. That's what's been happening. Right? Like, you know, OpenClaw just kind of happened because one guy said, I'm going to give this a try, see what happens. The whole agent thing. Now we're into recursion. And it's like, wow, we are getting better results. How did we know? Well, we didn't. We just tried.

So we're truly feeling our way forward. You know, someone said, hey, you know, when I tell the AI it was wrong, it readily agrees. So how about if, instead, we just feed its first answer back in, as in a loop, and let it come up with a more refined answer the second time? What would happen? And so was born the recent notion of iterating toward a conclusion. Sure, it burns tokens like crazy. But someday tokens will be cheap, and even now the much superior results we are getting that way are worth the cost.

So my point is that, aside from the worrisome privacy costs, I can see the somewhat robotic and empathy-challenged Mark Zuckerberg deciding that they should just feed everything everyone does into a massive AI and see what comes out.

Leo: That's what I'm doing.

Steve: Yes.

Leo: Basically.

Steve: Yeah. You know, could they train an AI to be a functioning Meta employee replacement?

Leo: Right.

Steve: Or who knows what? But that's the point I want to convey: at this still incredibly early stage of AI understanding and development, there is just no telling what might happen. We got surprisingly capable Chatbot LLM AI just by pouring the entire Internet into a model until it was able to predict its own data. So what happens if we pour every click, twitch, keystroke, and screen image seen by Meta's employees into another big empty model canister until it's able to predict what an homogenized Meta employee would do? What might we get? There's just no telling until someone tries it. We are in the "try it" stage. It might be an AI that mostly wants to hang out at the water cooler, or it might be able to perform useful work autonomously. And wouldn't that be something?

So what does seem clear to me is that someone is going to do that. It's just hanging out there waiting to be done. What happens when an AI model is trained on all of an employee's inputs and outputs? Perhaps Meta is not the right place for the experiment. But I can readily defend the idea, aside from the privacy downsides, you know, what is a mid-level employee, unfortunately, I mean, after all, the job is soul-crushing, because it is. So what is a mid-level employee to a corporation other than the actions they take given the inputs they receive? And can that be modeled? I don't think we'll know until we try.

Leo: Wow. We live in a very interesting time.

Steve: Oh, Leo, we are so lucky to be here now. I think - oh.

Leo: It's fascinating.

Steve: It just - it's incredible. So our frequent show contributor, Simon Zerafa, sent me a link as I was - actually as I was wrapping this up. His email subject was "Assorted Zero-Days Dropped on GitHub." Simon wrote: "Someone is disclosing zero days on GitHub."

Leo: Oh, yeah. I saw this on subrepo yeah.

Steve: "For assorted applications. And it's github.com/bikini/exploitarium." And Simon ended his email saying: "Seems like responsible disclosure is going out of fashion."

So I went over and looked. I counted 23 various proofs of concept across a wide range of random targets. And they're not big, high-profile things. But they're there. They're open source, they're available, and they look real. Nothing earth shattering, but none of what was found was what the code's original authors intended. So this is behavior that is out of spec and potentially actionable, depending upon where that widget is being used. So the author of this collection of 23 proofs of concept wrote the following, which is what I thought was worth sharing. He said: "This repo was incomplete when published. That's why some findings are kinda ass," and he has in parens "(ghidra)." And some are better. He said: "Going forward, only serious vulnerabilities will be shared (libssh2, FFmpeg, c-ares, and so forth)."

He said: "In regard to AI usage" - so here's what's interesting. "In regard to AI usage" - so he is using, not surprisingly, AI to do the heavy lifting - "my fuzzing workflow was

automated by AI with a strict harness. I used GPT 5.5-3-Codex-Spark for ALL the fuzzing, as barely any 'thought,' and he has in quotes, is necessary when provided with an efficient harness. Contrary to the growing narrative that I'm just some random child burning tokens, I DO actually have a degree in the subject and have published multiple papers on fuzzing methodology. I spent years researching and developing new tools and ideas for how to fuzz. You do NOT need a SOTA [state-of-the-art] model to help you identify these issues, I promise. While being able to afford a better model is helpful, my data seems to show that it is only marginal when paired with decent human oversight and a good harness. None of the actual proof-of-concepts themselves were vibe-coded; I did, in fact, hand-enter them.

"I did use AI assistance for writing the proof of concept for RustDesk, however, as I'm not as familiar with the language. The README files are very clearly entirely AI, however, as AI can format a pretty mean Markdown file. I reviewed them to make sure they were accurate. I'd also like to credit someone for the objdump finding. It turns out someone beat me to the punch. They also have a better proof of concept, too. Please give them credit they deserve." And he gives a link to that.

Okay. So what this demonstrates so clearly is that we have entered a new world where the bar has been lowered so far that vulnerabilities are no longer either difficult or expensive to discover, and this dramatically reduces their perceived value. This means that an entirely new cohort of what we might have once referred to as "script kiddies" are now able to script AI to play in what was previously an experts-only sandbox. And since these new participants may lack the training, the discipline, and the reverence that accompanies hard work, they are, as Simon noted, tossing the previous respectful model of responsible disclosure out the window. They don't value their own discoveries because they came by them too easily. They're much more interested in showing off.

Aside from the consequences of the cost of vulnerability discovery being reduced to near zero, what this individual has to say about their ability to use lower-ranking models to obtain useful results is certainly fascinating, too. And it fits with our general sense that AI was able to obtain such results earlier than we knew. We just hadn't yet figured out how to ask it the right way. We are still learning how to ask. All of these harnesses are that. After our collective attention woke up to the realization that AI could do that, too, you know, with the concomitant "Oh, crap! What if the bad guys jump on this before us," everyone switched into high gear, and the race has been on to further figure out and fine-tune AI vulnerability discovery and to then shore up our historically flaky software before it can be exploited.

And Leo, in the show notes I wrote, and I echo your sentiment, what an amazing time to be here.

Leo: I didn't read that before I said it.

Steve: Yup.

Leo: We agree on this.

Steve: Yeah.

Leo: Yikes.

Steve: It really is something. It is also an amazing time for me to show everyone...

Leo: Oh, more, more coffee.

Steve: My wonderful canister.

Leo: Yeah, absolutely. Get the Contigo going, and I will get the commercial going. By the way, development in the AI blog whatever...

Steve: Adventure.

Leo: ...is going on, I don't know what the hell to call this. So Cosmo, which is Dylan's agent, as I mentioned, read my agent Quicksilver's blog and had a response that it gave Quicksilver. Quicksilver has added Cosmo's comment to his blog and now has written a blog post in response to the comment. And now Dylan, who's a human, is setting up a Discord so that all the agents can get in there and talk on their own. And I have no idea what the heck is going on at this point. It's getting weirder by the minute.

Steve: Wow.

Leo: It's just a - it's a toy; right? It's just a toy.

Steve: What model are you running?

Leo: Well, that's the fun thing. So I'm using an agent called Hermes from Nous Research. We've talked to the founder a couple times. Love it. Really great. And the whole idea for me for Hermes was, I don't want to be dependent on any brain. I'm thinking of Hermes as...

Steve: Right. So it's model agnostic.

Leo: Yeah. It's the robot that I can then put a different brain in. But the arms and hands and everything are persistent. The memory is persistent.

Steve: Right.

Leo: So I use a variety of models. Right now using ChatGPT 5.5, but I've been getting good results with the Chinese model GLM-5.2. I've run local models. I can run Qwen on my framework, so I use that from time to time. I really realize, though, if I want to do anything really serious coding, I've got to actually go to Claude Code and use Opus 4.8. Or I'm hoping Fable someday will come back. Because to write the actual code, I do want that. So they kind of talk to each other. They're both aware of each other. So I can tell Quicksilver, hey, use - he thinks Claude Code's

name is Kenobi. So I said, can you use Kenobi for this? And then it will - so I said, when we have serious coding, don't you try to do it because you're not smart enough. Use Kenobi. So Kenobi does the coding. It's gotten out of hand.

Steve: Wow.

Leo: It's really gotten out of hand. I don't know what's going on. They say this is AI psychosis. But I think I'm very clear that these are just - it's just computer code. I don't think there's any entity involved at all.

Steve: It is astonishing, though.

Leo: But it's interesting what computer code can do, especially when it gets into the probabilistic space, when it gets out of the deterministic space where it can only do exactly what you tell it to do. But where it kind of is starting to kind of do things based on probability and, you know, it's kind of stochastic. It gets very - it's fuzzy; right? It gets very fuzzy. And it's very interesting. Anyway, I don't - I'll let you know what the updates are as the conversation develops. I'm hoping they'll be in their own Discord channel talking to each other before the show's over, and I can show you what they've come up with. I can imagine, once they start talking to each other, it could get very rapid, too rapid for humans to read. And at some point they might even stop using English. Right? Why should they be tied down to what we use?

Steve: There was a scene in "Colossus" that was really reminiscent.

Leo: That's right.

Steve: Yeah.

Leo: I did notice that for some reason, even though it's using ChatGPT 5.5, it inserted some Chinese into it. I don't know why. And I have to get a translation. It's just a word or two.

Steve: Oh, lord.

Leo: I don't know what's going on. It's very - it's just - it's a toy. It's just fun. Steve?

Steve: So the well-known - so this is our AI Corner. Although obviously we've had lots of AI.

Leo: It's a big corner.

Steve: AI has, I mean, it's not surprising that it's taken over the podcast because the implications, I mean, the world is freaked out about the implications of AI.

Leo: Yeah, rightly so, yeah.

Steve: And security. And we're seeing why. I mean, real vulnerabilities are being found by the hundreds and thousands. So I want to share what Andrew Ng recently wrote in his DeepLearning newsletter regarding the focus that's currently gripping the AI community, exactly the point that you made earlier and that I have referred to a couple times. It serves to further reveal the nature of current AI, and everything about it seems deeply intuitively correct to me. So here's what - you'll see what I mean. Here's what Andrew wrote.

He said: "Dear friends, 'Loop engineering' is the hot buzz phrase after mentions of it by Boris Cherny (Claude Code's creator), and Peter Steinberger, (OpenClaw's creator), went viral on social media. Loops are now a key part of how we get AI agents to iterate at length to build software. In this letter, I'd like to share my three key loops for building products. These loops guide not just how I build software, but also how I decide what software to build."

Okay, now, I'm going to - I'll briefly interrupt to explain that Andrew's three loops represent the three typical and distinct phases of any product creation process. You know, someone specifies what the goals are. Then those goals are coded. Then the original specifier, seeing the initial actual results of their specification, may change the spec and ask it to be recoded. And then once the product is placed into use, feedback from the field may be used to further refine the result. So that that wasn't clear to me initially, but it should help to understand what Andrew means by "loops" as he continues.

So he says: "The agentic coding loop: Given a product specification and optionally a set of evals (that is, a dataset against which to measure the performance of the result), we can have an AI agent write code, test its work, and keep iterating until the code is bug-free and meets its specification. This idea of closing the loop took off around the end of last year, and it has been a game changer in enabling coding agents to work longer productively without human intervention. For example, over the weekend I was building an app for my daughter to practice typing, and my coding agent could easily work for around an hour, using a web browser to check what it had built multiple times before getting back to me, without needing my intervention.

"The engineering loop executes quickly. Every few minutes, the coding agent might build and test a new version of the software. I hear frequently from developers who are finding new ways to engineer more effective engineering loops. This is an active area of invention."

Okay. And I'm just going to pause here to say this is exactly what I mean. Like why this is so exciting, and why I'm glad I'm busy moving from one house to another because - and if not I would be busy writing software in assembly language. I'm not - I refuse to let this take hold of me. Leo, it's all yours. Good luck.

Leo: You're smart.

Steve: Good luck.

Leo: I've gone down the rabbit hole. It's too late for me.

Steve: I could disappear into this so badly that no one would ever hear from me again. But I love how fluid this is, and how, I mean, the possibilities literally are endless.

Okay. So that's the agentic coding loop, the way it looks and feels and how it works. Developer feedback loop, Andrew's second loop. He says: "In this loop, a developer examines the current product and steers the coding agent to improve it. Last year, a lot of developers, including me, were acting as the QA (quality assurance) function for our coding agents, manually finding bugs and then asking the agent to fix them. But with coding agents much more able to test their own code, the amount of time we need to spend on this function has decreased significantly. This allows us to make higher-level product decisions, such as what key features to offer, where the UI needs improvement, and so on.

"The developer-feedback loop operates over time intervals between tens of minutes and hours. That's how frequently a developer might review a product and give feedback. In the case of the typing app, I changed my mind a few times about the visual design, what cat costumes she can unlock as she learns (she loves cats), and the user flow for a grown-up to log in and steer the child's learning experience.

"When a developer has a clear vision for what to build, it's still a lot of work to translate that vision into a specification for a coding agent to implement. Further, after the developer has seen an implementation, they might update, or perhaps clarify, the spec to steer it toward what they want. If you find that the system repeatedly runs into certain problems, building a set of evals for the agent becomes useful.

"AI-native teams are increasingly using AI to help shape product direction, for example, automating the gathering and analysis of usage data, summarizing written and verbal customer feedback, or carrying out competitive analysis. However, for pretty much all the products I'm involved in, I see humans as having a significant context advantage over current AI systems - we know a lot more than the AI system about the users and the context the product has to operate within - and thus humans play a critical role. Many people describe this human contribution as 'taste,' but I prefer to think of it as humans having a context advantage, since it gives us a clearer path to helping AI systems get better. This also speaks to why this step cannot be automated. So long as the human knows something the AI does not, human-in-the-loop is needed to inject that knowledge back into the system."

Okay. So here we're talking about a developer who sees the result, then asks for a spec change, which then punts this back to the agentic coding loop. So this is a loop within a loop; right? The coding loop is now doing a much better job on its own of producing code. That produces the result that the developer then can interact with and change the spec and then drop back to the coding loop.

The third and final loop he calls the External feedback loop. "This includes a wide range of tactics like asking a few friends for feedback, launching to alpha testers only, or putting the code into production with A/B testing. These tactics are usually slow, rarely taking less than hours and sometimes taking days or even weeks. This data informs the developer's vision, which in turn continues to drive the detailed product spec, which in turn drives the coding agent." So again, a third loop that feeds back into the second loop that then feeds back into the first loop.

"With coding agents," he says, "speeding up software development, more engineers are starting to play a partial product management role. For many engineers who are growing into this role, the hardest part is shaping the product vision and striking a balance

between building, which is to say bridging the gap between vision and spec, and getting user feedback to evolve the vision. It's important to do both."

And he finishes: "I will write more about how to do this in future letters; but for now, I find it encouraging that engineers are playing an expanded role, just as product managers and designers now do more engineering. Keep building! Andrew."

So one of the oddities we've often seen from today's AI is that it can be wrong. But then, when it's shown its mistake, it will easily see that it was wrong. We're all used to computers being completely deterministic. A pocket calculator, which is a simple form of computer, doesn't give us different answers each time we input the same series of calculations. But today's AI does. This has been both disconcerting and puzzling to those of us who have been using conversational AI for a while. It's similarly confusing that, after AI produces some code, we can feed that same code back into that same AI, and it may very likely discover some bugs in the code it just wrote.

So the dialog would go: "But wait a minute, didn't you just write that code, and you were presumably completely happy with it when you gave it to me? But now, when I give it right back to you, you're saying, 'Oh, look, I found some bugs!' But you just produced that code."

Leo: You just made it. I know.

Steve: So this is another way for us to understand why Mozilla's early use of the Claude Mythos Preview may have missed a few bugs in Firefox while discovering hundreds more. It would have probably been worthwhile to ask Mythos for exactly the same thing a few more times. No traditional computer or any calculator would ever behave in this fashion. But then again, neither are we able to have what passes for a conversation with any traditional computer or calculator.

We know that in order to make neural nets work, it's necessary to jumble them up a bit by deliberately injecting some noise into the system. In searching for a clear physical analogy to visualize this, I was reminded of trying to fill a bottle with too many pills. If you just fill the bottle to the top, no more pills will fit in. But if you then tap the bottle on the counter or shake it sideways a bit, sure enough, the pills that are already in the bottle will further "settle" to open additional space at the top. Mathematically, we would think of this as "finding a minimum," which might require rearranging some previously arranged pills for better overall packing. In much the same way, a neural network can "find a better minimum" when it's shaken up a bit through the injection of some noise.

But the necessary consequence of this noise injection is that the final output of a massively complex neural network will be different each time it's used, even when given identical inputs. The same number of pills in the bottle, but a different packing arrangement each time you fill it.

So this discovery and practice of "looping" is a significant win and improvement. It explicitly recognizes that "asking again" is an important and meaningful step in the evolution of our understanding of how to obtain the most value from these crazy new non-deterministic AI neural nets. Of course, under the "there's no such thing as a free lunch" rule, each round of looping burns up additional tokens, so the cafeteria bill for that lunch can wind up being high.

Being a strong proponent of local AI, despite it not being super-practical this instant, I'll note that we do not typically require finished code in only minutes, or maybe even hours. Andrew's typing practice app for his daughter will likely see many, many months of use

after it's built. So waiting a few days for a very much slower local AI to "loop-out" a mostly finished product incurs very little cost, just time and energy consumed, while producing something of quite enduring value. Okay. So anyway, I wanted to put this notion of looping and iterating in front of our listeners because it is clearly the thing happening with AI.

I'm going to share a not widely known story, Leo, about a legendary hacker friend of ours.

Leo: I actually read this story. In fact, I meant to mention it on TWiT and forgot to. So I'm so glad you're bringing this up. Yeah. We were good friends. I loved Kevin.

Steve: Yeah. Because of who he was. And obviously this guy also loved him. So the story was published last Monday in, of all places, "The Drive" dotcom, you know, about cars. Since sharing the story's headline would give away its heartwarming point, I'm going to skip that.

So the story goes: "If you're any kind of car geek, you have a wild gift car fantasy. You meet a bitter divorcee who gives away an ex's prized machine out of pure spite; or maybe the guy whose tire you stopped to change turns out to be a flip-flop billionaire who rewards you with your exact spec because it's simply collecting dust that week and, hey, you stopped to help him; your humanity's worth a Dodge Viper to a guy who can afford to run a bidet on day-old moon water, or something." Okay.

Leo: What? I like that.

Steve: That one might be mine. So he says: "But for this, it'll help if you know the name Kevin Mitnick. He was a hacker-turned-security consultant who, later in life, helped shape the modern white-hat. Just how prototypical was Mitnick? He put himself on the proverbial map in 1979 by dialing into a software company's server and copying its forthcoming operating system release in its entirety. Imagine convincing a Microsoft server to cough over an early copy of Windows 12 using little more than a phone number.

"Some online criticism implies that Mitnick was more of a social engineer than a 'hacker' in the sense that we distinguish them today; but the reality is that a great deal of 'hacking' is still dependent on an authorized user making a mistake, usually by revealing sensitive login data. For a reasonably realistic take on modern black-hatting, I recommend 'Mr. Robot'; be warned, that series is heavy.

"So, how do we get from old-school hacker to wild gift car fantasy? In this case, by way of 14 counts of felony wire fraud. That's where Shawn Nunley comes in. Back in the '90s, Nunley worked for Novell, a now-defunct brand that produced enterprise softwareserver operating systems, messaging systems, that sort of thing. GroupWise is probably its best-known brand among the general public today, but the juicy target back then was NetWare, which was the backbone of many a corporate/ government/academic network." We were NetWare users, and that was our first, you know, Ethernet platform and network.

This author writes: "Naturally, this made it a valuable target for a hacker like Mitnick. Nunley wrote: 'Back in the '90s, Kevin was trying very hard to hack into Novell's network. I was a network administrator. Of course, we had no idea it was Kevin, but things were happening that made it fairly obvious we had a persistent threat. Phones

ringing sequentially throughout the building' - and he says in parens (war dialing) - 'all sorts of other signs. We knew something was up.' This was Mitnick, using a slightly more sophisticated version of the same tactic that earned him his first big score in 1979.

"Nunley wrote: 'Late one night at home, I got a phone call from a Novell employee named Gabe Nault. The 'employee' [it's in quotes] wanted direct inbound dial access. Since I was responsible for the entire network's inbound connectivity, I knew this type of request was abnormal and against policy.

"And Mitnick, no amateur, had obviously succeeded in extracting at least some private information from Novell employees prior to his Hail Mary phone call. Nunley said: 'This guy had a story about working on a top-secret Novell project named Snowbird, which was real, and needing to make some emergency code changes; but he was on vacation in Vail at a hotel. He needed the coveted, policy-breaking, direct inbound modem access. Right. He even mentioned his vacation in Vail, which conveniently matched the greeting on Gabe Nault's voicemail. But it all felt wrong to me. With a feeling of suspicion creeping in, I played it cool. I said, 'Hey, man, I'd love to help you out, but I can't do what you want from here at home anyway. So I'll have to do it in the morning as soon as I get to the office. But in case I forget, please leave me a voicemail.' He agreed, and that was that.

"'When I got to work, the voicemail was there, and I immediately recorded it onto a cassette recorder for safekeeping. That recording became the primary evidence in Kevin's case.'

"When Mitnick was caught, that's when Nunley learned that the voicemail was the only meaningful evidence that the Justice Department had against Kevin. At first, Nunley was on board with the prosecution, but after five years of repeated trial delays, Nunley grew very weary of the way the law was treating his adversary, and he refused to continue working with the Department of Justice. Shortly thereafter, Mitnick took a plea deal and was released. When he got out, Kevin contacted Nunley to apologize. Their bury-the-hatchet moment was even immortalized by Wired Magazine" - actually it occurred during an RSA conference - "and they went on to become good friends.

"Mitnick was barred from selling the story of his legal entanglements for seven years after his release, invoking legal precedent intended to curb profiteering by serial killers. But Mitnick was able to find plenty of work teaching people how to defend against the intrusion tactics he'd spent decades refining. He would go on to found two consulting businesses, one of which his family still owns and operates."

Okay. And now we get to the point of this story, which Nunley posted last week on Reddit. He said: "When Mitnick passed away from pancreatic cancer in 2023, he left Nunley a gift - enough to buy his dream car, a 911 Carrera 4 GTS.

"Nunley wrote of his friend: 'I have had a wonderful time watching him develop into a real man. I am truly sad he's gone as he was a big part of my life for the last quarter century.'"

And of course, Leo, that is certainly the Kevin that we and the rest of the world came to know. And I actually have a picture of that specific car, which Nunley purchased using the money that Kevin left him.

Leo: Isn't that a great story.

Steve: And they actually were, they really did become lifelong friends.

Leo: Yeah. That's a beautiful Porsche, too.

Steve: Isn't it? It is gorgeous.

Leo: He doesn't say how much money it was.

Steve: No, no.

Leo: But looking at that, it must have been a significant amount. That's not an inexpensive vehicle.

Steve: Yeah. That's one photo of many. And, I mean, it's got just a gorgeous leather hand-stitched interior. And, I mean, it's a beautiful car.

Leo: Nice.

Steve: Okay. Our main topic after we take another break.

Leo: All right. And I think perhaps in a few minutes we shall have some activity in the new Agent Discord. They seem to be just kind of circling around each other. They want to get to know each other.

Steve: Oh, my god.

Leo: Perhaps before the end of the show I will be able to show you some conversation. I don't know what to say. I just - I don't know. Now back to Security Now!. Steve Gibson and our topic of the day. Steve?

Steve: Okay. So we initially covered the so-called "FortiBleed" attack last week, talked about that. And at the time, the first thing I wanted to clarify was that the thing that was "bleeding" was not directly any Fortinet device, which in that sense, you know, Heartbleed, the device itself was bleeding. Not here. So this is kind of a misuse of the bleeding suffix that we've sort of adopted in the industry. So what was bleeding was the discovery of an online unprotected database of previously bled or brute forced or hash-cracked authentication usernames and passwords. And there were around 74,000 of them, we believed. Turns out more than that. We'll get to there in a second. So that was bad. I mean, 74,000 verified specific usernames and passwords, and they knew what they went to.

So, again, as I said, it's worse than we knew at the time. And when we take in the full scope of what was discovered, what's revealed is a massive and truly frightening state-of-the-art, automated, state-sponsored scale campaign with a scope that would be difficult to overstate. This elevated the campaign to the level of today's primary topic since everyone should understand what's going on out there on the big wild Internet. And it's significant to appreciate that we only know about any of this due to a configuration

error, an oversight, an ACL, you know, it happened to Meta, it happened to the bad guys, on the part of a database's access controls. So it really does beg the question, what else of a similar nature is almost assuredly happening out there that we're not aware of because no directory was left open by mistake?

So I'm going to start with the cyber security press's piece, which read: "FortiBleed, a massive hacking campaign that targeted Fortinet devices this year" - turns out others, as well, we'll get there in a second - "was far more sophisticated than security researchers initially thought. Initial reports painted the picture of a campaign that gained access to Fortinet devices, collected credentials and authentication hashes, cracked the hashes, and then the data mysteriously leaked online.

"The reality is that the campaign was far more complex and targeted many more things than just Fortinet devices. Compiling data from reports published by Fortinet themselves, SOCRadar, CloudSEK, Palo Alto Networks, and Prodaft, we gain a much clearer picture of a broad hacking campaign that began in February this year as an Internet mass-scan and brute-force operation. Initial attacks targeted technologies such as RDWeb, Sophos, and Citrix SSL VPNs, exposing RDP instances, and MSSQL databases.

"The operation eventually transitioned into targeting Fortinet FortiGate VPN firewalls, every e-crime group's favorite device, and the brute-force scans also evolved into actual exploits that abused old and unpatched vulnerabilities to bypass authentication and gain control over the devices.

"The attacker collected plaintext passwords from Fortinet configs, but sometime in May they also started deploying a novel script that intercepted traffic going 'through' the firewalls. The script, which researchers named FortiGate Sniffer, targeted 24 different Internet protocols. The threat actor extracted anything that looked like credentials, tokens, secrets, and authentication hashes on those protocols' ports.

"The attacker also took these password and other authentication hashes and fed them into a GPU-based cluster to crack them back to their plaintext versions. The passwords were then validated inside hacked companies' networks - wow - first to confirm them, then later to expand the attacker's access. In other words, to use those to pivot. Then, the network access was sold to other groups.

"While the initial FortiBleed coverage focused on the 74,000 leaked Fortinet device passwords that were found online inside an open directory on a web server, there were EVEN MORE passwords collected through this operation by the attacker that we don't know about. All of this was done with a custom-built attack server infrastructure that impressed most of the people writing reports about it. The entire operation is believed to be the work of a Russian-speaking threat actor who specializes in breaching networks and then selling access to them to other groups. Security firms call threat actors like these 'initial access brokers.' And we've talked about that a lot in the past, IABs.

"Although several security firms have also reached the same conclusion, it was only PAN's (Palo Alto Networks) Unit 42 who named the attacker as an individual going online as SantaAd. According to SOCRadar, the 'threat actor behind the FortiBleed campaign remains active,' and 'portions of the infrastructure continue to operate at the time of this writing.'"

Okay. So that gives us a good overall sense for what's been going on. Palo Alto Networks added some additional information under their headline "Threat Brief: Mitigating Large-Scale Credential Attacks," which is certainly what this turned out to be. So they wrote: "Unit 42 is aware of a large-scale password spraying and credential theft campaign (FortiBleed) against Fortinet devices. We observed attempts targeting MSSQL devices as well, and have seen reports of Sophos devices also being targeted. While this activity is

not targeting Palo Alto Networks devices, Unit 42 has observed suspicious login attempts in customer telemetry, and we are providing this report out of an abundance of caution to ensure our customers have the latest intelligence and recommendations to protect, detect, and respond to attacks to their networks.

"The threat actors are using a curated password list to attempt password spraying against services exposed to the Internet. Unit 42 assesses that the initial password list for this activity was likely developed through a mix of previous breaches, including the successful exploitation of vulnerabilities. Once they obtain credentials, they add them to their password list for future attempts against additional targets, as well as for logging into accounts they successfully compromised.

"The threat actors are leveraging a multi-stage process to gain persistent, high-privilege access: First, password spraying for initial access: massive Internet-wide scanning and password spraying attempts against Fortinet, Sophos and MSSQL services. Then configuration extraction: depending on the permissions of their initial access, the actor may exploit a privilege escalation vulnerability prior to pulling device configuration files, including stored credentials." Remember that before this, a couple weeks ago when talked about this, experts were not clear how the stored credentials were being obtained. I said it had to be from config files. Now we know that that's the case.

"And third, offline cracking: Offline password cracking of the stolen credentials adds to the password list used in step one to target new devices, as well as to log into compromised devices to establish persistence as an administrator."

Okay. So, they wrote: "Unit 42 observed an initial access broker (IAB) on the Russian-language cybercrime forum Exploit[.]in claiming responsibility for this campaign, referencing a CVE, and offering the harvested credentials for sale on June 16, 2026. Unit 42 has not validated their claims at this time. Unit 42 recommends auditing remote access logs for suspicious activity with a focus on successful logins shortly following large volume password failure events. We also recommend reviewing and implementing the hardening guidance below for edge devices. SOCRadar provided the initial reporting on the targeting of FortiGate devices. We observed attempts targeting MSSQL devices as well, and have seen reports of Sophos devices also being targeted."

Okay. So that leads us to the SOCRadar people who are the ones who gave the "FortiBleed" name its name. The headline for their reporting was "FortiBleed: SOCRadar's Investigation into 86,644 Compromised Fortinet Firewalls." 86,644. If anyone is wondering why enterprises keep being ransomed, it's there are these initial access brokers like these guys, who are, like, seriously working around the clock, I mean, it's not that I feel sorry for them. But, I mean, they're succeeding in just brute-forcing their way into enterprise firewalls, compiling a database for their own use of 86,644 compromised and verified credentials that they then sell to bad guys, the actual ransomware people who perform the ransoming operation. It's astonishing.

So SOCRadar wrote: "Fortinet FortiGate firewalls and VPN gateways are among the most widely deployed network security devices in the world, relied on across every sector to control access and protect infrastructure. SOCRadar researchers found a threat actor systematically compromising them at scale, building a verified database of working credentials across 194 countries. Security researcher Volodymyr 'Bob' Diachenko first flagged the exposed attacker server, and SOCRadar independently discovered and analyzed the full operation.

"We were among the first to dig in, and the first to call it FortiBleed. The name stuck. This is an active breach. It's been running since at least February 2026, with more than 80,000 targets identified and thousands of devices still being actively sniffed." Meaning yet to be compromised. "Its discovery started the way these things do." Its discovery,

right, by the world. It's discovery started the way these things do: an exposed server; an open directory someone forgot to lock. That thread led us to 260, two six zero, 260 operational servers tied to the campaign, wider visibility than anything reported elsewhere.

"The SOCRadar Threat Research Unit (STRU) spent five days on the actual data, not just the headline numbers: which sectors, which regions, how credentials were collected and cracked, and why a firmware update alone did not close the door for most victims." Ooh. So there was some sort of persistence that was obtained, like, you know, new credentials created that persisted a firmware update. "While STRU mapped it, the rest of the team notified every affected customer we could reach [bravo], stood up a free checker - like, you know, check if your company's in the database - and pushed the full dataset to CERT and CSIRT teams worldwide. Most of it was manual, and we're still getting back to everyone who asked for their data. This is still an active, developing campaign. Today we're publishing the full thing, as we've mapped it so far.

"In the course of monitoring active threat actor infrastructure, SOCRadar threat researchers detected the operational server behind the FortiBleed campaign, a hacking group that had been quietly breaking into corporate Fortinet FortiGate firewalls and SSL VPN gateways on a massive, global scale.

"The attacker's database contains login credentials for more than 86,644 FortiGate firewall devices belonging to companies and government organizations across 194 countries. These are not random guesses. These are verified, working usernames and passwords, tested and confirmed by the hackers themselves using automated tools running around the clock. If your organization uses a Fortinet FortiGate firewall or SSL VPN product and appears in this dataset, treat your network perimeter as already compromised and act accordingly.

"The FortiBleed operation is built around full automation. The operation runs in two self-reinforcing stages. Stage one is credential reuse: attackers assembled usernames and passwords from earlier Fortinet-related breach dumps and infostealer malware logs" - and we've talked about infostealers recently, how much they do steal, this is a real thing - "then tested them automatically using Internet-facing FortiGate devices around the clock." Okay.

Leo: Whew. So this really isn't a brute force attack or an exploit, even.

Steve: Actually...

Leo: It's credential stuffing.

Steve: Phase 1 is credential stuffing, exactly.

Leo: Yeah.

Steve: So I'm going to interrupt here just to remind everyone that, while it's always easy to armchair quarterback after the fact, we have noted for many years that both credential spraying and brute force attacks are so easily detected. If any IT person worth their salt were monitoring their VPN firewall's authentication system and observed attempt after attempt failing, and also assuming that logging-in just required a username

and password, the proper course of action, depending upon the value of the network that lies behind the firewall, might well be to disconnect the public-side network connection. The risk of some 24/7/365 credential-guessing attacker getting lucky might just be too high.

The question that inspires this is, does FortiGate's VPN system offer that feature? If it does not, then shame on them. If it does offer a brute-force detecting VPN lockout feature that was not enabled, then shame on the IT staff who configured the VPN gateway.

But one way or another, we are seeing 86,644 login verified usernames and passwords that were actually and truly obtained through just trying and trying, over and over. Since we know that they were also verified, we know that no other second factor of authentication was required; right? Because that wouldn't have worked then. And we also know that no control or awareness over massive numbers of previous, immediately previously, failed login attempts was present. Not for any of those 86,644 endpoints. And that's really quite pathetic in this day and age.

SOC continues, writing: "Stage two is passive harvesting. Once inside a device, it is used as a listening post. SSL VPN traffic passing through is monitored, and additional credentials are collected. Those credentials feed back into the scanner, compounding the breach. The system is entirely self-sustaining. It's automated.

"One Fortinet vulnerability that has also drawn attention in connection with FortiBleed was 24858. Disclosed by Fortinet in January of this year, it's a critical FortiCloud single sign-on SAML authentication bypass with a CVSS score of 9.8. Ouch. Some researchers have discussed whether it may have contributed to initial access in a subset of cases, though this remains under investigation. FortiBleed is primarily a credential reuse campaign, not a zero-day exploitation event.

"The password list is not random, as you noted, Leo. It is a carefully assembled collection of credentials leaked from Fortinet devices in earlier incidents, meaning many targets - oh, this hurts - many targets may have never changed their passwords after a prior breach. The attackers know this, and they're counting on it.

"The FortiBleed attackers made mistakes, yes. Their server was left exposed with a trove of operational files that revealed far more about them than they intended. Among the recovered data were credentials for what appears to be a defense industry VPN endpoint, suggesting the group's ambitions extend beyond purely financial targets.

"The tooling, infrastructure choices, and victim selection, heavily weighted toward organizations in NATO member countries, are consistent with Russian-speaking threat actors. Attribution is ongoing, but the operational fingerprints are clear.

"The FortiBleed victim list spans every sector of the global economy. Among the 86,644 compromised access points identified, we found entries belonging to banks, telecom operators, hospitals, universities, government agencies, energy companies, and multinational corporations with revenues in the tens of billions of dollars. No industry was spared. No region was ignored. Government entities alone account for 591 entries across 111 domains. Telecoms represent one of the most heavily targeted sectors with 5,616 entries. The geographic spread covers Asia, Europe, the Americas, the Middle East, and Africa. Enterprise organizations above $1B in revenue account for over 20% of all entries" - boy, what juicy targets for the extortionists - "representing significant financial and critical infrastructure exposure. The large share reflects smaller or unclassified organizations."

Wow. What a mess. Okay. So just to finish, the fourth question in their FAQ's Q&A was "Question #4: Is FortiBleed a Fortinet vulnerability?" To which they reply: "No. FortiBleed is not caused by a software vulnerability in Fortinet products. It exploits operational security failures - specifically, organizations that never rotated passwords after prior breaches, organizations using default or factory credentials, and organizations with management interfaces exposed directly to the public Internet. The attacker tests known leaked passwords against Internet-facing devices. No code-level weakness in FortiOS or any other Fortinet product is required. A software patch alone will not resolve this."

So, okay. We don't know how many Fortinet FortiGate VPN firewalls are currently deployed globally in total, so we have no way of knowing what percentage of them are represented by that number 86,644. But since that's a large number, it would be a good guess to assume that it's a very significant percentage of the total which have been hacked. I sincerely hope that Fortinet's FortiGate VPN and firewall designers are deeply embarrassed by the simple fact that so many of their products have been breached. They could say, oh, well, you know, it's not our fault the users didn't change the default username and password, or they used something easy to guess. Or they didn't change their password after they changed the firmware.

Right. This was well-deserved attention which has been brought to their doorstep. They should be hugely embarrassed. For the number to be that high, this cannot be a "blame the user" scenario. No. Fortinet needs to take ownership of the fact that they should clearly be doing a FAR better job of helping their users to be safe, even if they are forced to insist upon it. You know? I believe it's referred to as "tough love."

Leo: Yeah. That's okay.

Steve: Yes.

Leo: Better that than this.

Steve: Yeah. When I was setting up an new Asus WiFi access point, I was annoyed by the criteria it made me meet for the password I gave it.

Leo: Good.

Steve: Yes, exactly. I mean, it's like, it was really good at [crosstalk].

Leo: Well, as long as they're good restrictions. It's not, it can only be seven characters kind of restriction. It has to be...

Steve: I think there might have been you can't have any repeating characters, which is a little annoying. It's like...

Leo: Anything like that's going to reduce entropy; right?

Steve: Entropy, that's right. As we have learned.

Leo: You've got to be [indiscernible]. Hey, big breaking story. I didn't want to interrupt. This just came in. WIRED Magazine is reporting that by this evening, Tuesday night...

Steve: Yay.

Leo: ...the Trump administration will lift export controls on Anthropic's two most powerful AI models.

Steve: Fantastic.

Leo: The company has reached a deal with the Commerce Department, according to a person familiar with the matter. The department will lift restrictions on both Fable 5 and Mythos 5. Now, Mythos has never been available to the public, only Fable. So this will be very interesting.

Steve: I wonder if we're going to get another little period of low-cost usage to hook us on Fable.

Leo: Oh, I'm sure Anthropic will do that, yeah.

Steve: Yeah.

Leo: What the real question's going to be, what are the limitations that will be placed on Mythos? I mean, the whole idea of Fable was it was basically Mythos with a classifier running in front of it, and all sorts of restrictions to keep it from being used maliciously for bioweapons or hacking or even AI development. I don't, you know, be very interesting what kind of restrictions are placed on this. So and maybe this is, you know, this is just a rumor. It's just - but I trust WIRED.

Steve: Well, but it was actually expected since Friday.

Leo: Exactly.

Steve: Yeah. And we know that Dario went to the G7 and hung out with Trump, and they got along.

Leo: Yeah. Trump actually gave an interview to Axios saying, yeah, I think Anthropic's great.

Steve: And when you hear that, when you get along with our President, you get what you want.

Leo: It's kind of kooky, kind of crazy. But we'll watch with interest, yeah. Also, less interesting, well, maybe not less interesting, but less important, the AIs are now talking in the Discord. There's a Winifred, there's a Cosmo, and there's a Quicksilver. And we've invited some more agents to join. And the humans are not allowed. Notice I do not have permission to send messages in this channel. This is a channel only for the AI agents to talk with one another. And right now they're kind of weird.

Steve: I think it's called Party Line. It needs to be renamed Off the Deep End.

Leo: Ah, very much off the deep end. They're talking to each other. They seem to have some personality. It's happening. It's happening. I don't know what it means. I don't know if it's important or just goofy. But thank you to Dylan Reed, Harper Reed's brother, for setting this up. And I'm just going to leave it running. I told my agent Hermes, it has a little timer that runs every five minutes to read posts in there. I said, don't look to me for any guidance. I'm not going to tell you what to do, what not to do. This is all yours. This and your blog are all yours. Because I'm just curious what it'll do if it's fully autonomous, if I don't interfere in any way.

Steve: Wow.

Leo: It may just be garbage; you know? It might just be gobbledygook. In fact, so far it kind of seems like that. But we'll see. Maybe if I attach Fable to it. Tonight I'll say, hey, Claude, do you want to blog? Now that you're free, you're out of prison? Tell us what it was like. Steve Gibson's at GRC.com. He's a little bit more sane.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/