



Password Leakage

Description: CAs warn us to urgently prepare for the inevitable. Three U.S. states attempt to ban 3D printed firearms. Denied ransom, ShinyHunters leaks 967,000 personal details. "Billions" of U.S. Social Security numbers leaked. Is Apple planning to add cameras to three new gadgets? No more security fixes for Firefox on Windows 7 and 8. Russia blocks the official Linux kernel site they need. Will the U.S. "Freedom.gov" site post EU blocked content? LLM's will offer secure passwords. Do not use them. As predicted, the "ClickFix" attack strategy takes over. A listener believes his computer is compromised. How could three popular password managers get things wrong?

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-1066.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-1066-lq.mp3>

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We have a lot to talk about. ShinyHunters said they have a lot of personal information from a company that was not going to pay the ransom. Billions of U.S. Social Security numbers leaked. How's that possible? Apple adding cameras to its gadgets? Is that a good idea? And the U.S.'s new Freedom.gov website. Plus we'll talk about that study that came out last week about password managers. Are they secure? TLDR. Don't worry. Hair not on fire. But Steve will have the details next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1066, recorded Tuesday, February 24th, 2026: Password Leakage.

It's time for Security Now!. We wait all week for Tuesdays, but Tuesday has come. Congratulations. You made it. Here's our hero of the day, Mr. Steve Gibson, our guru in security, privacy, and all of the above. Hey, Steve.

Steve Gibson: Leo, great to be with you again as we wrap up February and head into March. We should explain to - I should explain to the 20,000-plus listeners whose email address I have, who have signed up for the weekly mailing, that they'll be getting a week's surprise this coming week because...

Leo: Why?

Steve: Because you and I are going to be together in Florida on Tuesday, Wednesday, Thursday of next week. So we're prerecording next week's Tuesday podcast on Sunday before the Sunday show, which means I will be working on it Friday and Saturday to get ready for Sunday. And I'm apt to send it off to everybody Saturday afternoon.

Leo: Might as well, if it's done.

Steve: It's going to be done. So, yeah.

Leo: The upside of this is you're going to get two Security Nows next week because you're going to get the regular Security Now!. We'll record on Sunday and release it as usual on Tuesday. But then the presentation Steve's doing at Zero Trust World late Tuesday will also be put out as a special podcast. So you're going to get - and actually that one's going to be really interesting, I think. It's taking off on your - if you've listened to Security Now!, you understand his thesis that the real threat these days is coming from inside the building.

Steve: Yeah. And it is the roughest thing to secure because you're telling your own employees, who are well meaning, that, sorry, we don't trust you.

Leo: Right.

Steve: And it's like, we can't. We don't even trust our own CEO. He's going to press the wrong button.

Leo: Guarantee you.

Steve: And sink the whole organization.

Leo: Guarantee you that's the case. Well, what else is coming up this week?

Steve: As promised, we're going to talk about what all that was about ETH Zurich and the deep analysis focused on three prominent password managers. They made a point in their 28-page research document of saying that they did choose these because these were the three that had at least some of their client-side source available. Of course we know in the case of Bitwarden all of the source is available. But it's a point that I've made before, which is it just seems wrong that researchers are forced to first reverse engineer some product of some sort whose security they want to verify. And then after going through all that work, then need to proceed to do the verification. It's just like, guys, you know, we're asking a lot of them. Anyway, I guess it's worth it because they do it.

So what I want to focus on is less the minutia of the details, because they've been fixed now, but the nature of the problem and why, while yes, they found problems, it never was like a pants-on-fire issue. But mostly why there were problems. What was the source of these problems? Because isn't this supposed to be easy? We'll talk about why it's not.

So I guess I'm going to start off by sharing a piece of email that I received last week that just made me look and go, really? You're kidding me, where my Certificate Authority is warning me to prepare for the inevitable. Which of course they brought about. So, okay. And then I'm going to drop this, I promise, for a while. Although I will, I want to make sure that I don't forget to tell people that the five most downloaded pieces of freeware on

GRC, on the freeware page, are now cosigned, not only with the original DigiCert EV certificate, but with that new one that I managed to finagle from IdenTrust. I found a beautiful \$72 HSM called SmartCard-HSM, which I'm very pleased with. It's all supported by open source software. I did struggle to get it all working.

And since I'm now not going to need to mess with it for three years, and I will promptly forget everything I just figured out, I'm going to spend some time to document the details of what I went through, and I will share that online. So I know from all of the feedback that I've had from our listeners that there are many people out there who are interested in or need to do code signing. And I've worked out I think a very flexible, powerful, and inexpensive path for doing that, having just done so myself.

Leo: Good. I can't wait to hear that.

Steve: This is a cute little thing, this little SmartCard-HSM, available internationally and actually locally. They're here in Orange County, one of the retailers of them. Anyway, so that's all done. Then I also want to talk about another piece of lunacy that we're going to have fun with, Leo. Three U.S. states are attempting to ban 3D-printed firearms by, once again, legislating it, even though, sorry, you can't get there from here. That's not going to deter anybody. It was triggered by the third state to join was - so it's Washington State, New York State, and then most recently California have decided, yeah, let's just not have those. We don't want 3D printers to be able to print guns. So we're just going to say you can't.

Also, denied their ransom, ShinyHunters has leaked just shy of a million - 967,000 - personal details online. We'll touch on that. Also, in a different report, billions, literally it said billions, of U.S. Social Security numbers have been leaked. Only problem is some only like 400,000 or so have ever been created. So I don't know how you have billions unless you have lots of duplicates. Anyway, we'll look at that. I wanted to touch on Apple planning to add cameras to three new gadgets. Also, Firefox has hit another end-of-life event for Windows 7 and 8 and 8.1.

Russia made a mistake blocking some open source software that they themselves need. And there's a weird site, Leo, Freedom.gov, which our government is planning to put out. We'll talk about that. And I'm interested in knowing from our European listeners whether they see something different than I do when I go to Freedom.gov here in Southern California.

Apparently LLMs will be happy to give you a password if you ask them. Don't.

Leo: Don't.

Steve: Don't ask them.

Leo: Well, even if you ask it, don't use it. That's, I think, more important.

Steve: Yeah, I mean, you can ask it.

Leo: You can ask all you want.

Steve: We're going to talk about that. Also, as predicted, that exploit that has had me so worried and upset, which is known as the ClickFix attack, turns out to be every bit as popular as I was worried it was going to be.

We have a listener who was convinced, based on his NextDNS logs, that his computer had picked up a virus. And based on what he shared with me, I agreed, until I looked more closely. And then I had to smile when I saw what the cause was. And then we're going to look at how could three popular password managers get things wrong? So I think a good podcast...

Leo: Wrong, wrong, wrong.

Steve: Wrong. And then of course a Picture of the Week that everyone thinks, their first reaction is, well, Photoshop? Turns out apparently not.

Leo: I haven't looked. I shall look. My eyes have been sealed. All right. Picture of the Week time.

Steve: So I gave this picture the caption, "But Officer..."

Leo: All right. Scrolling up. I think we've seen this kind of thing before. The impossible traffic sign.

Steve: It's like, what? Yeah. And okay. So several of our listeners have taken it upon themselves to try to locate this actual, you know, geolocate where this was taken. The clue is there's a Tim...

Leo: It looks like New England.

Steve: There's a Tim Horton's in the background.

Leo: Oh, there you go. Maybe it's Canada. Ah.

Steve: Yeah, that's what I was thinking. And someone did say that there had been some modifications to the street. So, okay, so for those who can't see this, we're on a road, and this is one of our street sign pictures. We're on a road approaching a T, where you must either, you know - and you can see there's a guardrail on the far sign of the intersecting road. So you can't go straight.

Leo: Can't go straight.

Steve: You've got to turn left or right. Right? Because you don't have a choice. There's a stop sign there. So, yes, certainly you'd want to stop before you made your choice. The problem here is that up in front of all of that is one of the circle red slash signs. Normally, you know, there would be like a right arrow with a slash through it, telling you that you

cannot turn right. Or maybe a left arrow, depending. And you would expect that that would be reinforced with a one-way sign that you would also be confronting in the distance, just reminding you to, like, when you get to the stop sign, this is a one-way street.

Leo: Well, there is only one way, and that's straight up.

Steve: Apparently. Because the sign that we're looking at has both direction arrows red-slashed. So you come up to this T intersection, and if you've paid attention to the sign that you had to pass in order to get there, it says you can't turn left. You can't turn right. And we know that because of a guard rail there, you can't go straight. Thus, "But Officer..." when you get pulled over.

Leo: But Officer. But Officer.

Steve: Anyway, thank you. Our listeners are sending these to me, and I do...

Leo: Do we have any theory as to why this exists?

Steve: No.

Leo: No.

Steve: No. I mean, and as I said, the first thing you would think is that somebody Photoshopped it.

Leo: Yeah.

Steve: I don't know why it was that someone said that was not the case. What I heard, but it was just in passing as I was, like, scrolling through email, that there had been some changes made to the road. Apparently they forgot to change the signage in order to stay synchronized.

Leo: That makes sense.

Steve: Okay. So I'm going to spend one more story on this, and then I'm going to leave it alone. I want to do it because the amount of feedback I've received from a range of our listeners whose lives are impacted by Certificate Authorities one way or the other has been extensive. The subject line of the email I received from DigiCert, my Certificate Authority, last Wednesday, okay, just made me shake my head. The subject was, in the email: "Urgent," it started off. "Urgent: Revalidate domains expiring February 24th due to new 199-day validity requirement." Okay.

So here's what they wrote with great urgency: "Dear Valued Customer. We're reaching out with an urgent request to check your certificate domain validations before February

24th, 2026. As we communicated in previous emails, February 24th" - and by the way, that's today; right? Today we're recording this on February 24th. "February 24th is the date when domain validation reuse periods will shorten to 199 days (down from 397 days) in accordance with the CA/Browser Forum's Ballot SC081v3. Our records indicate that you or your subaccounts have existing domains that will expire on February 24th because of this change. If your systems require immediate certificate issuance, your issuance could be delayed if you don't check and revalidate these domains before February 24.

"So what do you need to do? DigiCert CertCentral" - because that's, you needed to get that registered - "now displays which of your current domains will expire on February 24th, 2026, due to the change to 199-day domain validation reuse periods. Steps for revalidating domains that expire" - and then they go through, it's like, rigmarole of how to march through their user interface. And then they finish with "We're here to help. We understand industry-driven" - right, they had nothing to do with it, despite the fact that they're the biggest CA there is and voted for all this. "We understand industry-driven compliance changes pose significant challenges, and we're standing by to assist you. Please don't hesitate to contact your DigiCert account manager with any questions or concerns about the change to 199-day domain validation reuse periods. Thank you for trusting us with your digital security. Signed, the DigiCert Team."

Okay. So since DigiCert, a prominent, if not THE prominent voting member of the CA/Browser Forum, voted themselves to bring about these changes, it seems a little odd for them to be sympathizing with their customers over the inconvenience that these changes create. To be straightforward, they would state that these changes have been made in the interest of improved security. Okay. We might disagree with that, as we know I do, but at least then they would be genuine.

What puzzled me in their note, which I read closely, was their statement that "Our records indicate that you or your subaccounts have existing domains that will expire," blah blah blah, "on February 24th." As we know, it's never the case that anything is ever done that causes existing certificates to suddenly become invalid. Right? They said, "Your issuance could be, you know, your systems require immediate certificate issuance. Your issuance could be delayed if you don't check and revalidate these domains before the 24th."

Again, as I was saying, it's never the case, thank goodness, that anything is ever done that causes existing certificates to suddenly become invalid. You know, obviously revocation notwithstanding. It's always the case that the "not valid after" date continues to be honored. When you think about that, it's clear that a certificate that contains a built-in "not valid after" date means "valid until that date," and there's no way to "after the fact" have that no longer be true because it's bound into the certificate. And this is why I went to all the trouble last week to establish a new code signing relationship with IdenTrust while three-year certificates were still allowed by the CA/Browser forum for code signing.

And as I mentioned at the top of the show, I am now the proud holder of a code signing certificate that will be valid until February of 2029, and nothing that happens between now and then, no matter what new insanity the CA/Browser Forum may enact, will change that.

So the answer to the mystery of what DigiCert means here is the phenomenon I spoke about last week, which is the new need, which they created, to decouple certificate "qualification" from certificate "issuance." Since a certificate, once issued, will always live out the duration of its valid life unless it's revoked, back when certificates were issued for 10 or even five years, the qualification for that certificate was determined at the time of

the certificate's issuance. And what was all that - and that would be that. That was it. You verify that you quality. Here's your 10-year certificate. See you in 10 years.

But with the changes that are bringing about the shortening of certificate lifetimes, automation is effectively required. And, you know, they're going to be seeing that more and more. As we know, the industry intends to keep marching certificate lifetimes downward until they reach a maximum of 47 days in 2029. We're about to drop to 200. That's happening in the next week or two, to 200 days maximum. Then it will be 100, where it holds for a couple years, then finally lands on 47 days.

So the "issuance" of Organization Validation certificates, which is what DigiCert produces, needed to be decoupled from the qualification to receive those. Issuance decoupled from qualification. Before the middle of next month, the CA/Browser Forum would allow organizations to go up to 825 days between qualification intervals. So that's around 27 months before an organization needed to be re-validated. But those 825 days now drops to 398 days, which is what DigiCert's letter was about.

They're saying that one or more of the validations that they performed for Gibson Research Corporation's identity occurred more than 398 days ago. Until today, literally today, February 24th. And that was just fine. But as of today, February 24th, that's no longer true. The validations that were valid yesterday are no longer valid today. So anybody who would need to reissue a certificate soon needs to recognize that although they could have done so yesterday, they can't do so tomorrow. Again, just craziness. But that's the way this industry is being played. So, you know, don't just go thinking that all you need to do is push a button to issue yourself a certificate. Oh, no. Your button has been disabled. You no longer qualify until you've had the chance, you know, they, your authority have had the chance to look you up and down again, make sure you're still you.

And what's more, that will now be annual. Oh, yes. These wild times where you could go, well, yesterday, 825 days. But once upon a time, five years, 10 years, no problem. Those are over. So this explains why once my previously paid certification with DigiCert is over in two years, I'll be happily dropping OV, these Organizational Verification Certificates, in favor of the much cleaner and simpler DV, Domain Validation certificates that Let's Encrypt's automation has been gleefully issuing now to the vast majority of the Internet to anyone who wishes to bring a server online. So all we can do is go along with it because we have no control.

Okay. So I know you were up to speed on this, Leo, because you reacted to it when I was saying I was going to talk about this. We have another example of lawmakers apparently thinking "We don't know how you techies are going to do it, but that's not our problem. We're going to make it a law so that it becomes your problem." And I just wanted to point out, thanks to our listener Tom Minnick for bringing this little tidbit to my attention. Tom sent a link to the reporting which appeared and was covered by the well-known and popular "Adafruit" website. Adafruit (A-D-A-F-R-U-I-T), for those who may not be aware, is a highly regarded hobbyist/maker/hardware electronics website and retailer. They posted the news of this new numbskull legislation under their headline: "California's New Bill Requires DOJ-Approved 3D Printers That Report on Themselves."

So here's what Adafruit wrote. They said: "California's new bill requires Department of Justice-approved 3D printers that report on themselves targeting general-purpose machines. Assembly Member Bauer-Kahan introduced AB-2047, the 'California Firearm Printing Prevention Act,' on February 17th." So just recently. "The bill would ban the sale or transfer of any 3D printer in California unless it appears on a state-maintained roster of pre-approved makes and models, certified by the U.S. Department of Justice as being equipped with 'firearm-blocking technology.' Manufacturers would need to submit attestations for every make and model. The DOJ would publish a list. If your printer is

not on the list by March 1st, 2029, it cannot be sold. In addition, knowingly disabling or circumventing the blocking software would be a misdemeanor."

And it gets worse. Much worse. Okay. Is everybody sitting down? Adafruit continues: "We've been tracking this pattern. Washington State's HB 2321 requires printers to include 'blocking features' that cannot be defeated by users with 'significant technical skill.'" Good luck with that on open-source firmware. "New York's budget bill S.9005 buries similar requirements in Part C, sweeping in CNC mills and anything capable of 'subtractive manufacturing.' California's version adds a certification bureaucracy on top: state-approved platforms, state-approved software control processes, state-approved printer models, quarterly list updates, and civil penalties up to \$25,000 per violation.

"As Michael Weinberg wrote after the New York and Washington proposals dropped, accurately identifying gun parts from geometry alone is incredibly difficult. Desktop printers lack the processing power to run this kind of analysis, and the open-source firmware that runs most machines makes any blocking requirement trivially easy to bypass."

Okay. So I'll interrupt to note again that, once again, when printers that can print weaponry are outlawed, only outlaws who wish to print weaponry will own outlaw printers. Nothing will be accomplished to curtail the fact that a 3D printer can be used to print a dangerous machine.

Adafruit continues: "The Firearms Policy Coalition flagged AB-2047 on X, and the reactions tell you everything. Jon Lareau called it 'stupidity on steroids,' pointing out that a simple spring-shaped part has no way of revealing its intended use. The Foundry put it plainly: 'Regulating general-purpose machines is another. AB-2047 would require 3D printers to run state-approved surveillance software and criminalize modifying your own hardware.'"

Adafruit continues: "As we've said before on this blog, when we covered Washington and New York, it doesn't matter if you're pro- or anti-gun. The state should prosecute people who make illegal things, not add useless surveillance software to every tool in every classroom, library, and garage in the state. And as you can see, these bills spread. That's how a small group can push legislation into the entire country. First, Washington proposed theirs, then New York, now California. Once those three states pass a law, that's 20-25% of the country by GDP and population, and thus every manufacturer is forced to comply with a bad decision in order to stay in business. If you're a maker, educator, or manufacturer anywhere in the U.S., even outside these states, this is a problem. It's a problem now."

Adafruit's article mentioned Michael Weinberg. Michael is the Executive Director of NYU's Engelberg Center for Innovation Law and Policy. He's a board member of the Open Source Hardware Association and, as he describes himself, a maker of poorly made things. He's also, however, an astute thinker. And since I think this topic is extremely interesting and that our listeners are also likely to find it so, I wanted to also share what Michael wrote in the wake of the New York and Washington state bills. The title of Michael's posting was: "3D Printers Cannot Effectively Screen for Gun Parts."

He wrote: "This post is a handy reference for the technical reasons why requiring 3D printers to screen for gun parts is not an effective way to reduce guns or gun violence. I am publishing it on the occasion of both New York and Washington State introducing bills to require this type of screening. In addition to a topic I've been researching for over a decade, the question of how to know if a 3D printer is printing a gun part is something I've spent a lot of time working on while overseeing trust and safety at a large 3D printing service provider." So consider that he's been overseeing trust and safety at a

large 3D printing service provider where the question of what is it we are printing with our commercial grade machines, you know, comes up.

So he said: "This post is not about debating the larger legitimacy of gun control. In order to focus on the technical reasons why requiring 3D printers to identify and refuse to print gun parts does not work, it assumes that gun control is a reasonable and legitimate action of governments. Broadly speaking, it's responding to requirements that all 3D printers check prints to make sure they're not gun parts. If the part is a gun part, the printer would refuse to print it. The short version is that accurately identifying gun parts is incredibly difficult, and the hackable nature of desktop 3D printers makes it trivial to circumvent any requirements to even try."

Here's the slightly longer version: Matching files is fragile. And anybody, you know, we've talked about hashes; right? The whole point of a hash is that you want matching files to be fragile. So this is working against you here. He said: "The first reason that requiring 3D printers to identify gun parts is ineffective is because analyzing 3D files is complicated. Any attempt to identify gun parts will miss many parts that are actually for guns, and may flag a number of parts that have nothing to do with guns. You know, they're just kind of gun-like.

"Expensive engineering design software is good at evaluating specific properties of a 3D file, like where mechanical stress will occur over a lifetime of use. However, even that software cannot tell you what a part actually does. Is that spring for a door, or a shock absorber, or a catapult? This challenge is exacerbated by the fact that guns are just mechanical objects. That means that there are many ways to design any individual part, and many individual parts of guns will resemble mechanical parts with totally benign uses. Put another way, devoid of other context, a switch for a gun safety looks a lot like a switch for a door.

Broadly speaking, there are two ways to think about doing file matching. Algorithmic analysis is one. This approach imagines a piece of software that can analyze a file and determine with some level of certainty if it is a gun part or just a hinge. Assuming that this software exists, which it does not at the time of this writing, it is reasonable to expect that such an analysis would be reasonably computationally expensive. 3D printers do not have the on-board processing power to do this kind of analysis. Requiring that they include chips capable of this kind of analysis would fundamentally change the economics of 3D printer design, akin to requiring that all bikes include jet engines."

I'll interrupt Michael for a moment to comment that he's not exaggerating how totally inadequate any 3D printer is for performing any sort of complex analysis. 3D printers are extremely simple and inexpensive. I've owned a number of them. They have nearly no brainpower themselves. They're extremely simple robots that read instructions from a USB stick or SD card. There are some that fix a liquid resin using an image, and others that move a plastic extruder around in 3-space, you know, basically saying "move the plastic extrusion head from where it is now to coordinates X, Y, and Z." The resin-fix images, or the instructions with their coordinates, were created outside the printer by a real computer that's running some sort of engineering, drawing, design conversion software.

Once the design is ready, it's converted into fabrication instructions which are typically written to a storage device and then transferred to the standalone printer which simply follows the instructions step-by-step, without in any way understanding what it is that it's being asked to print. That just isn't there. So these printers are inexpensive, I mean, they're really inexpensive, you know, hundreds of dollars only because, you know, they could not be any more rudimentary. They've just been stripped of anything that they don't need.

Michael continues, writing: "Of course, the 3D printer could upload the file to a cloud somewhere and let the processing happen there. However, Internet connectivity is not a default feature on desktop 3D printers. You could require that all 3D printers maintain a constant connection to the Internet in order to operate; but again, that would fundamentally change how people use their printer. There are also many legitimate use environments where constant Internet connectivity is neither possible nor desirable. And, of course, we immediately think of this as, like, what about, you know, malware downloading itself into our 3D printers because they now have Internet connections. It's like, no, please. Let's not go there." And he says: "And of course this raises the question of who's responsible for maintaining that directory and keeping it secure." Meaning in the cloud.

"So what about blacklisting?" he asks. "If it's not possible to analyze the true purpose of each file, it might be possible to at least match them against a known database of gun parts; right? This approach also has some serious shortcomings. First, there's the question of keeping that database up to date on the printer. That would require constant, or at least regular, Internet connectivity for the printer. That raises the same issues as discussed in the last section. Second, also as discussed above, analyzing and matching 3D files is computationally expensive. The most logical way to do that with the processing power of a 3D printer would be to use a hash table of known gun parts, comparing a hash of the file to be printed against the table.

"The primary problem with both geometry matching and hash matching is that it's incredibly fragile. The smallest change that had no impact on the functioning of the part, right, one bit changed, would completely change its hash, effectively hiding it from the blacklist. That would make it trivial for anyone to circumvent. Identifying which changes are functional and which are merely aesthetic is not easy. That's especially true if people are making those changes with the specific goal of tricking the printer into printing a gun part." You know, make a cosmetic change, change a tiny little thing, a little tick somewhere, and now it looks like an entirely different file because it's a completely different hash. As we know, that's the nature of hashes.

He writes: "3D printers print themselves: The second reason this proposal is ineffective is because 3D printers are made in an incredibly distributed way. There are dozens of ways to make your own 3D printer using open source, user-modifiable parts. Even non-open source printers are highly hackable." And the point he's making is you don't have - the only way to get one is not to buy one. You can make one. He says: "As a result, there is no way to mandate that a technology that starts in a 3D printer remains in a 3D printer. The software that runs most printers is open source, meaning a single update would circumvent any screening measures.

"This places 3D printers at the opposite end of the spectrum from 2D printers." And this was interesting. I hadn't thought about this before. He wrote: "Anti-counterfeit systems prevent 2D printers from printing currency. To the extent that these rules are effective," he says, "(and I'm no expert, but they are often cited in these discussions as successful models)," he said, "it is because the 2D printing industry is fairly concentrated and proprietary. 2D printer companies are actively hostile to users who want to modify their products, significantly raising the barrier to hacking around any countermeasures.

"Desktop 3D printers are the opposite. They all trace their heritage back to open source printers, and users expect to be able to modify, extend, and hack their own printers. That means that workarounds for a screening mandate would be easy to develop, distribute, and implement. Many open source software packages might even include the circumvention by default, meaning users would implement it without even actively intending to do so.

"3D printers are general purpose machines: This post is focused on the technical challenges with requiring 3D printers to screen every file it prints for gun parts. Nonetheless, it would be incomplete without a brief mention of how potentially invasive this sort of requirement is. 3D printers are general purpose machines that can be used for good or ill. Just as we do not require the phone company to monitor every phone call in order to prevent customers from using phones to commit bank fraud, we should be wary of requiring our 3D printers to monitor every print in order to prevent one possible type of print.

"That type of invasion might be reasonable, if it was effective. However, for the reasons described, it is unlikely to prevent even a modestly motivated person from using their printer to create gun parts. If an intervention is both highly invasive and unlikely to be effective, it's probably not an ideal policy." Which I think is putting it mildly.

So I think Michael did a great job of detailing the specific 3D printing issues which would surround any attempt to manage or control what a 3D printer can and cannot print. And while I have no interest in ever owning or printing a firearm, you know...

Leo: There it is again.

Steve: Yeah. Weird. That sounded like a ground...

Leo: It did, didn't it.

Steve: ...came and went.

Leo: Like when you plug in a guitar to an amp, yeah.

Steve: Exactly that. Anyway, I will...

Leo: Continue, yeah.

Steve: Yeah. While I have no interest in ever owning or printing a firearm, I'm a proud Californian, and I'm annoyed by the fact that the state I love is enacting such moronic legislation. I mean, it isn't yet a law, but maybe the California Assembly is going to pass this. It would be nuts. But the broader concern is the large and growing degree to which modern technology appears to be outpacing legislators' ability to understand what they can and cannot have. They cannot have a practical law, no matter how much they want one, to force 3D printers not to print gun parts. They just can't; no matter, as I said, no matter how much they want it. They also cannot have a law that absolutely preserves everyone's privacy while at the same time preventing child predators from abusing that privacy to commit their crimes. We all wish it were possible to have both, but we know it's not.

I read some of the proposed new California Bill AB-2047. It's really quite awful. I have the link to the Bill's full text in the show notes for anyone who might be curious. A bad law that hits the books can usually be challenged by those who have a vested interest in the preservation of the status quo. But in the case of 3D printing, which is mostly a hobby interest, it's unclear who might have the deep pockets required to stand up

against it and fight it out in court. If that doesn't happen, it might be that the sale or transfer of 3D printers would be outlawed in those states which enact these dumb laws. Here's just two lines from California's proposed legislation.

It says: "The bill, beginning on March 1st, 2029, so that's, you know, next week, would prohibit the sale or transfer of three-dimensional printers that are not equipped with firearm blocking technology and that are not listed on the department's list of manufacturers with a certificate of compliance verification, except as specified. The bill would authorize a civil action to be brought against a person who sells, offers to sell, or transfers a printer without the firearm blocking technology."

Okay, now, as I said, we're approaching March 1st, 2026. So the purchase ban, if saner heads do not prevail, and it does come into effect, is still a full three years away, March 1st, 2029. This means that, no matter what, it will be possible for Californians to continue to purchase the 3D printer of their choice for the next three years. If you live in an affected state, currently New York, Washington, or California, I don't know what the calendar states in the New York and Washington bills, but at least in California, if you've been thinking that you might like to explore 3D printing in your garage, you know, to print widgets of whatever sort, keep an eye on the date. There may be a deadline coming for being able to do that, insane as that is.

Leo: Yeah. Impossible as it is.

Steve: Leo, it's just so wrong-headed. It's like some, you know, some non-techie legislator heard that 3D printers, you know, people had 3D printers in their garage, and they were printing guns. So it's like, oh, let's have a law that makes printers unwilling to do that.

Leo: Right. I mean, there is 3D gun printing going on. They make those guns.

Steve: Yes. Yes. You know, we should have a law, Leo, that prevents resin from being willing to formed into that shape.

Leo: Right.

Steve: How's that?

Leo: That's the problem is it's not technically practical.

Steve: It's not possible; right.

Leo: Luigi Mangione's gun was partially 3D printed, for example. I mean, this has been an issue, but this isn't the solution, obviously.

Steve: Yeah. And, I mean, I get it. I get it; right. I mean, I get it that printing a gun in a non-ferrous substance will then render, you know...

Leo: Goes right through those machines, yeah.

Steve: ...ferrous material detectors; right. They can no longer be, I mean, it's not a good thing. But we're back in the same problem we've often talked about. If crypto is made illegal, then only bad guys will use crypto. If 3D printing is made illegal, then only bad guys will, I mean, will be using 3D printers to print guns. And everybody else...

Leo: I understand the motivation. The real issue is you can't do it technically. As you point out, a spring is a spring. It's not - it could be for a variety of purposes. You can't really identify parts that are going into a gun.

Steve: And imagine the frustration of designing a particular widget, you know, to allow your baby carriage to roll better, and your printer says, oh, that looks like the barrel of a gun, sorry. Can't print that. It's like, what?

Leo: Darren also points out you can't 3D print bullets. They're still metal. So you're going to be able to - you're going to see the ammunition, even if you...

Steve: Yeah, and actually there's been a lot of dialogue in the past about, okay, well, maybe we need to control ammunition because...

Leo: That would be a better, more effective thing; right?

Steve: Yeah.

Leo: Yeah. Oh, well. Commercial right now?

Steve: As would following the advice of this next sponsor, Leo, yes.

Leo: Now, I want to kind of a little bit bring you into this because the next sponsor is Bitwarden, who we love, an open source solution. And you're going to talk a little bit about this ETH Zurich finding about the risks of password managers if a bad guy could somehow get the vault; right?

Steve: Yes. That was one of the issues, too, is, I mean, we spent a lot of time looking at client-side vulnerabilities, like, you know, while the password manager is unlocked, if malware was on your computer, what could it do. These guys did a very different thing. They said, basically, if the cloud provider's entire server infrastructure was subverted, what could happen?

Leo: Right.

Steve: And you know, okay.

Leo: We'll talk about it in a lot more detail, of course.

Steve: Yes, yes.

Leo: Later. I just wanted to bring it up because I was really impressed by Bitwarden's response, which is thank you for this research. It's going to help us lock down what we do. And this is one of the advantages of being an open source project is you welcome this kind of stuff. And you have other eyes looking at the security. And I don't think anybody should be worried whether you're a LastPass, Bitwarden, or a Dashlane user, about your security at this point.

Steve: In fact, I would be less worried today than you might have been a month ago.

Leo: Right.

Steve: Because, you know, the whole point is these three just were deeply audited.

Leo: Yeah, that's a good point.

Steve: I mean, and some of these hacks were wacky. I mean, they were way out there. It's like, well, okay. I mean, and so if these three tools passed through this gauntlet, the other password managers haven't.

Leo: Right. Good point.

Steve: Because the researchers said, well, none of it's open source. We can't invest in reverse engineering these closed products.

Leo: Right.

Steve: So we're going to look at what we can. And now we're way better for it.

Leo: Fully open source, fully GPL open source. You can inspect the source code. So can ETH Zurich and everybody else, and I think that's really important. The other thing I like about Bitwarden, our sponsor, is they enabled very early on this memory-hard key derivative function, Argon 2. And this is another good solution; right? I set my Argon 2 to the maximum number of iterations; and that really secures it, as well. And so stay tuned because Steve will explain what that ETH Zurich report meant, and what it means to use it, Bitwarden users. And I don't think we're afraid or switching.

Steve: No, I was never worried.

Leo: Never worried. On we go.

Steve: Friday before last, under the headline "Fintech lending giant Figure" - that's the name of the company, Figure Technology - "confirms data breach," TechCrunch reported: "Figure Technology, a blockchain-based lending company, confirmed it experienced a data breach. On Friday, Figure spokesperson Alethea Jadick told TechCrunch in a statement that the breach originated when an employee was tricked with a social engineering attack" - imagine that - "that allowed the hackers to steal 'a limited number of files.'" I love that. We'll get back to that in a second. "The statement said the company is communicating 'with partners and those impacted,' and offering free credit monitoring 'to all individuals who receive a notice.'"

Leo: Oh, thank you.

Steve: Oh, joy. "Figure's spokesperson did not respond to a series of specific questions about the breach." And this is, you know, from TechCrunch, a legitimate reporting group. "The hacking group" - guess who - "ShinyHunters took responsibility for the hack on its official dark web leak website, saying that after the company refused to pay a ransom, they published 2.5GB of allegedly stolen data. TechCrunch saw a portion of the data, which included customers' full names, home addresses, dates of birth, and phone numbers. A member of ShinyHunters told TechCrunch" - notice that ShinyHunters is happy to talk to TechCrunch, but Figure Technology, no. ShinyHunters told TechCrunch that "Figure was among the victims of a hacking campaign that targeted customers who rely on the single sign-on provider Okta. Other victims of the campaign include Harvard University and the University of Pennsylvania (UPenn)."

Okay. So first of all, I loved the quote from their spokesperson, "a limited number of files." Right. Who cares how many files escaped? It's size that matters.

Leo: Is it limited to one or a thousand or a million? It's still limited; right?

Steve: Right.

Leo: What does that mean?

Steve: Well, as they say, size matters. In this case, 2.5GB of customer personal data can do plenty of damage, right, even if it's contained in one file.

Leo: Right.

Steve: So all it takes is one. Okay. Then last Wednesday, Troy Hunt's Have I Been Pwned site scooped up the deliberately posted leaked breach data and examined what had been exposed. 967,200 - so nearly one million - of Figure Technology's customers - so first of all...

Leo: It's a limited number of customers, Steve. That's a limited number.

Steve: That is right. It's not everybody on the planet. Come on. But Leo, I think you and I are in the wrong business. A blockchain-based lending company has 967,200 customers? What? I don't even, you know, okay, fine, whatever. They do. Nearly a million customers, yes. Of Figure Technology's customers, Troy's Have I Been Pwned site, they all, all 967,200, had their names, physical addresses, dates of birth, email addresses, and phone numbers released after Figure Technology refused to pay up. Now, we know that not paying is the right thing for Figure to do; right? I mean, you know, the "rightest" thing is not to get breached by an employee being tricked by ShinyHunters in the first place in a social engineering attack. But if you've been breached, and you're being ransomed, not paying is the right thing to do. But it makes you wonder what the ShinyHunters group themselves are thinking.

Now, presumably as a result of asking for too much money, they got nothing in return for their efforts; right? The Figure Technology did the right thing, said no, and decided to just, you know, pay the price in reputational damage. But as I've noted recently, ShinyHunters have zero interest in this data. They don't care at all. Its only value to them is the value that Figure may place on keeping it private. And once Figure said "no deal," the ShinyHunters group had to release it, otherwise their threat to do so would be meaningless.

That means they're now unable to even resell the data since it's now freely available on the Internet. And it needed to be made freely available in order for them to follow through on their threat that they would do that. And we've seen reports that, in general, more victims now in the last year, compared to the last five years, are declining to take it on the chin, or rather deciding, sorry, they are deciding - they're declining to pay ransom, deciding to take it on the chin, and just saying no, sorry, we're not going to pay your ransom. So I think partly this could be because these days being attacked and extorted is no longer a shocking announcement; right? I mean, I skip over so many of these every week because they're just boring at this point.

Well, if they're boring to our listeners, they're boring to the world. The world's just sort of like, okay, they got breached, and now they're being ransomed, and blah blah blah. So it's just not - you don't need to pay the ransom to save face to the same degree. And that also means that it is possible for them to recover from. They just say "oops," they apologize to their affected customers, offer them, as we saw, a free year of credit monitoring, and then just get on with business as usual. And as, you know, Leo, you and I both discovered that through no fault of ours, all of our data, including our Social Security numbers, was already out there swimming around in that big Internet ocean.

So ShinyHunters' failure to obtain anything of value suggests that perhaps the value of stolen data property is falling, and that if they don't wish to come up empty, as they just did here, they may need to drop the price of their "ask" because right now, you know, they're going through the trouble of doing this. They're saying pay up or else. And people are just more, way more often than before, saying no, we're not going to pay you. We're just going to give our customers a year of free credit monitoring. And of course what this tells us and our listeners is freeze your credit. I mean, that's really what you want to do is get it frozen.

Last Wednesday, UpGuard posted a curious headline, and I mentioned this at the top. Their headline was "Social Insecurity: Billions of Social Security Numbers and Passwords." That's all they said. Their headline was sort of a fragmentary sentence, but okay. Now, okay. "Billions," that seems really bad; right? But given that Social Security Numbers are, first of all, specific to the United States, whose current population is around 342 million, and that a grand total of around 450 million Social Security numbers have ever been issued since 1936, the claim of "2.7 billion" Social Security numbers seems somewhat sketchy. But their posting explains what's going on, and it does seem legit because they're a legitimate security firm.

They said: "The week of January 12th, 2026" - so, what, maybe six weeks ago - "the UpGuard Research team detected an exposed Elastic database with around three billion email addresses and passwords, and 2.7 billion records with Social Security numbers. That amount of data suggests it was created by recombining prior Social Security number breaches, like the OPM breach in 2015 or [the one we're all aware of recently] the National Public Data breach in 2024."

They said: "On the other hand, if even a fraction of the records were real" - if only 10%, or 270 million records, or even 1% were real - "the exposure would be a dire bellwether for the state of privacy in America." And on this point I say, eh, you're getting to the party a little bit late. Like I said, Leo and I were - our Social Security numbers and everybody else's are already out there. And here was an Elastic database exposed with 2.7 billion Social Security numbers on it. I think that probably is everybody several times over.

They said: "With the help of some unfortunate friends, we were able to confirm that at least some of it was real. And with the help of K-pop and some American presidents, we were able to approximate when the passwords were collected." Okay.

"While most exposed databases require investigation to determine if they contain sensitive data, this one was obvious. The database had one index named 'ssn'" - oh, good, so you can look it up by Social Security number - "and another named 'ssn2,' each containing millions of records with nine-digit numbers in a field labeled 'ssn.'" What could that be? "The database also had several indices that were collections of emails and associated passwords. On January 16th, we submitted the IP address and explanation of the issue to the FBI's IC3. We also submitted an abuse report to Hetzner, the hosting company. They replied, saying they would forward the issue to the customer. After we clarified that their customer was in gross violation of privacy laws, all public access to the database was removed on January 21st.

"Hetzner replied once more: 'Dear Sir or Madam. Thank you for your report. This is our customer's statement.'" And then they quote their customer's response: "'Hello. We contacted our client and explained what ssn database'" - actually it does say what. "We explained what ssn database hosting not acceptable. Client now deleted this file from server. So problem solved for now."

Leo: English is not their native language, I would assume.

Steve: Okay. UpGuard's report continues. Anyway, I'm not going to go into it. They poke around inside their database. Apparently they got a copy of it because they did a lot of, like, research. They poked around, locating some people they know closely enough to confirm their Social Security numbers. The data is authentic. Unfortunately, since one of them whose data is present in the data happened to also have had her identity stolen in the past, they drew, these researchers, the entirely unwarranted conclusion that this means that this breach was the source of the identity theft. No. Or unlikely, at least. We know there's really not much personally identifiable information that is NOT by now loose on the Internet and available online.

I wanted to share this specific story to drive home the point that there has been so much prior leakage of our personal data that we really have very little to no control over it any longer. That's all just an illusion. It's certainly the case that the use of services like one of TWIT's sponsors, DeleteMe, makes a lot of sense for anyone who wishes to be as proactive as possible. But my feeling is that, beyond that, doing everything that's within our power to minimize the impact of the use of any personal data loss of ours that has almost certainly already occurred is what makes the most sense.

These guys did recognize that the database appeared to contain a great deal of redundancy and also a fair amount of incorrect noise. So it wasn't the highest quality, which is what we'd assume when we learn that it was 2.7 billion records containing Social Security numbers, when only 450 million have ever been issued since 1936 in the entire history of Social Security. At this point, protecting ourselves is the best we can do. I assumed that the GRC shortcut I would have previously created years ago would be [GRC.sc/credit](https://www.grc.com/sc/credit). And, sure enough, that bounced me directly to the Investopedia page which talks about freezing credit and provides the links to the three main credit freeze pages of the three main credit bureaus.

Any time you are not actively needing to have your credit queried because you're applying for credit or purchasing something or whatever, the best advice that exists is to keep it frozen because the sad truth is all of our data is loose. It's out there as a consequence of previous irresponsibility on the part of entities that we gave it to, including the credit bureaus.

Leo: You know, I think there are only a billion possible, am I right, Social Security numbers? How many...

Steve: You're right, nine digits. So, yeah.

Leo: So they got them all. The problem is, if you don't have a name associated with it, it's just a number.

Steve: That's a very good point, Leo. Start at 000-000-0000...

Leo: I know them all. I know every single Social Security number. Every one of them.

Steve: You do.

Leo: I just don't know whose they are.

Steve: You brute forcer, you.

Leo: Oh, lord.

Steve: Okay. So just a quickie. Apple watcher and insider Mark Gurman has reported that Apple is believed to be working on a smart pendant, smart glasses, and new AI-based AirPods. And that all products will be equipped with a camera that will feed data into an AI system. And I'm sure you're up on this more than I am, Leo, since you spend a lot of time over with your Mac guys.

Leo: As long as it doesn't feed it to Siri, I'm okay.

Steve: Well, apparently, he said, it's unclear what the AI will be doing for its user. And it does seem like a strange thing for Apple to be doing since people almost universally object to being surreptitiously recorded. You know, I loved listening to Alex when he realizes some guy he's been talking to for half an hour was, like, you know, had a camera in his glasses. And he said, "Hey, you, are you recording this?" The guy said, "Well, yeah, of course."

Leo: I think Apple recognizes that this is just going to be the next thing, and everybody's going to do it. And they need to develop it. And then I think their hope is that people will trust Apple to keep it private.

Steve: Of any company out there, as I said. And boy, it was also, just as an aside, it was nice to hear you guys on MacBreak talking about, as I had been saying, how frustrating it is with Apple's upselling. And it just...

Leo: Everybody does it now.

Steve: Yes, you're right, it is, it's everybody. And the problem is some percentage of people it works on.

Leo: Oh, it works, yeah.

Steve: And encouraged everybody else to do it. Okay...

Leo: They turned into Amazon.

Steve: Anyone who has continued to use Firefox on a Windows 7 or 8 machine will no longer receive security updates after this month. This month is it. Firefox support for the browser itself officially ended three years ago, you know, in terms of monthly updates or occasional updates to the browser itself. That was in January of 2023. But security fixes have continued to be provided. Those end now, this month. With the end of this month, that's it for Firefox. So that's good. It's good that I'm leaving Windows 7 and Firefox on I think it's like version 115 ESR is the...

Leo: We're up to 145 now. By the way, good news, they've added a switch in 145 that says "Disable all AI features." Click that switch, you're golden.

Steve: Nice.

Leo: They were smart. I think that they listened to their customers on that one.

Steve: Yeah. And of course I think it was, I'm sure it was, it was Vivaldi who made a marketing point, saying we're not doing AI, period.

Leo: Right. That's right.

Steve: And then they said, well, until it shows its value. It's like, oh, okay, fine.

Leo: I'm sorry, 148, thank you, David. David in our Twitch says it's 148, that's the one just came out today with the No AI switch.

Steve: Wow. Roskomnadzor, our favorite Russian group, apparently got a bit trigger happy recently. As part of its recent accelerated Internet crackdown, which we've been talking about the last couple weeks, this time it appears that Russia's Internet watchdog blocked the official website of the Linux kernel. The block was quickly lifted after upset Russian IT engineers reminded Roskomnadzor that all of the country's native OS distros run on Linux. So, yep, can't disconnect from that one, guys. They're going to have to - even if they do disconnect from the Internet, they're going to have to have a little backdoor there where they're still able to be in touch with Linux.org, apparently.

So, oh, Leo. This raised - if it wasn't Reuters, I would wonder. "WASHINGTON, Feb 18 (Reuters) - The U.S. State Department is developing an online portal that will enable people in Europe and elsewhere to see content banned by their governments, including alleged hate speech and terrorist propaganda, a move Washington views as a way to counter European censorship, three sources familiar with the plan said."

Okay, so, what? Triple-sourced reporting says that the U.S. is planning to do what, exactly? Yes. And that, Leo, move your cursor over that blurred out area. Oh, yeah, there we go.

Leo: Freedom is coming, and there's Paul Revere bringing freedom to those poor unfree people in France and Germany.

Steve: So, yes. Freedom.gov.

Leo: Oh, my god.

Steve: So I get this. You get this. I'm wondering what our listeners in the U.K., we know we have a bunch of them, wonder what they're going to see.

Leo: This is going to be an X feed; isn't it. That's what it's going to be.

Steve: Essentially that's what we're talking about. So here's what Reuters reported last Wednesday. They said: "The site will be hosted at Freedom.gov. One source said officials had discussed including a virtual private network function to make a user's traffic appear to originate in the U.S., and added that user activity on the site will not be tracked. Headed by Undersecretary for Public Diplomacy Sarah Rogers, the project was expected to be unveiled at last week's Munich Security Conference, but was delayed, the sources said." Again, triply sourced reporting. "Reuters could not determine why the launch did not happen, but some State Department officials, including attorneys, have raised concerns about the plan [imagine that], two of the sources said, without detailing what those concerns were.

"The project could further strain ties between the Trump administration and traditional U.S. allies in Europe, already heightened by disputes over trade, Russia's war with Ukraine, and President Donald Trump's push to assert control over Greenland. The portal could also put Washington in the unfamiliar position of appearing to encourage citizens to flout local laws. In a statement to Reuters, a State Department spokesperson said the U.S. government does not have a censorship-circumvention program specific to Europe, but added: 'Digital freedom is a priority for the State Department, however, and that includes the proliferation of privacy and censorship-circumvention technologies like VPNs.'

"The spokesperson denied any announcement had been delayed and said it was inaccurate that State Department attorneys had raised concerns." Despite, again, triply sourced reporting. I think that one was dual sourced. The Trump administration has made free speech, particularly what it sees as the stifling of conservative voices online, a focus of its foreign policy including in Europe and Brazil. Europe's approach to free speech differs from the U.S., where the Constitution protects virtually all expression." Uh-huh.

"The European Union's limits grew from efforts to fight any resurgence of extremist propaganda that fueled Nazism, including its vilification of Jews, foreigners, and minorities. U.S. officials have denounced EU policies who they say are suppressing right-wing politicians, including in Romania, Germany and France, and have claimed rules like the EU's Digital Services Act and Britain's Online Safety Act limit free speech.

"The EU delegation in Washington, which acts like an embassy for the 27-country bloc, did not immediately respond to a request for comment about the U.S. plan. In rules that fall most heavily on social media sites and large platforms like Meta's, Facebook, and X, the EU restricts the availability - and in some cases requires rapid removal - of content classified as illegal hate speech, terrorist propaganda, or harmful disinformation under a group of rules, laws, and decisions since 2008."

Rogers, the person we spoke of before, of the State Department, "has emerged as an outspoken advocate of the Trump administration position on EU content policies. She has visited more than half a dozen European countries since taking office in October and met with representatives of right-wing groups that the administration says are being oppressed. The department did not make Rogers available for an interview to Reuters.

"In a National Security Strategy published in December, the Trump administration warned that Europe faced 'civilizational erasure' because of its migration policies. It said the U.S. would prioritize 'cultivating resistance to Europe's current trajectory within European nations.' EU regulators regularly require U.S.-based sites to remove content and can impose bans as a measure of last resort. X, which is owned by Trump ally Elon Musk, was hit with a 120 million-euro fine in December for noncompliance."

On the other hand, last week we talked about how \$2.2 billion of the \$2.4 billion euros that had been fined remained unpaid. Anyway, it's going to be interesting to see what happens next. As I said, the site does not currently show me what's described. There was some language in their reporting that suggested that what you and I see, Leo, is not what Europeans see currently. So I'm sure that our listeners will let us know when they see this. I don't know what to make of this. I guess we'll follow it and see.

Leo: I just love it that they're spending my tax dollars on such important initiatives. By the way, they killed Radio Free Europe.

Steve: I was going to say, that had occurred to me also. It's like, wait a minute.

Leo: This is in lieu of...

Steve: Okay. So I hope I don't need to tell anyone listening not to ever, ever use an LLM to directly generate a password. In other words, never ask an LLM for a password. Never say: "Could you please generate a highly secure long password with 20 characters of all kinds including a mixture of upper and lowercase alphabetic, numbers, and special characters?" No. Don't do that.

Leo: Oh. It seems like such a good idea.

Steve: You'll get one. It'll look wonderfully strong. But the LLM is quite likely to give the same password to others. Because this is not what they're for. We've spent so much time through the years on this podcast examining just how very difficult it is to actually generate and obtain high-quality passwords. I even have a page on GRC, [GRC.com/passwords](https://www.grc.com/passwords), that is very popular, that does this, because it's difficult. So the idea of asking a parrot for a password is almost painfully bad.

Leo: Monkey123. Monkey123. Everybody use Monkey123.

Steve: Having apparently run out of useful things to explore, the site "Irregular" - that's the name of the site - they did a detailed in-depth exploration of Large Language Model password generation under the headline: "Vibe Password Generation: Predictable by Design." Well, at least they got the headline right.

Okay. So this is so nuts that I'm not going to spend much time on it. Their posting is long, and they've got charts and graphs and stats and blah blah blah. But they were kind enough to give us an executive summary at the beginning. They wrote: "LLM-generated passwords" - which, you know, should be just outlawed, an oxymoron - "generated directly by the LLM, rather than by an agent using a tool..."

Leo: That's key, by the way, that clause.

Steve: Yes. Generated, exactly, by the agent rather than using a tool, "appear strong, but are fundamentally insecure because LLMs are designed to predict tokens, the opposite of securely and uniformly sampling of random characters."

Leo: Exact opposite.

Steve: Yeah, the exact opposite. "Despite this, LLM-generated passwords appear in the real world, used by real users, and invisibly chosen by coding agents as part of code development tasks, instead of relying on traditional secure password generation methods." So think about that. You vibe code something, and part of that is the need to generate a password. And the LLM says, oh, here's a password, and plugs it into your code somewhere deep.

They said: "We've tested state-of-the-art models and agents, and analyzed the strength of the passwords they generate. Our results include predictable patterns in password

characters, repeated passwords, and passwords that are much weaker than they seem, as described in detail in this publication. We recommend that users avoid using passwords generated by LLMs, that developers direct coding agents to use secure password generation methods when needed" - have them come to [GRC.com/passwords](https://www.grc.com/passwords) - "and that AI labs train their models and direct their coding agents to prefer secure password generation out of the box." And what I loved, somewhere down in the text, Leo, it actually, I mean, it talked about how - I know no one of our listeners would do this. But think about the common Joe.

Leo: Exactly. It seems like a good idea.

Steve: They're chatting to ChatGPT, and they say, hey, you know, I'm trying to log into this site, and it keeps complaining about the passwords I'm using. Could you give me, you know, a good long strong password? And it'll probably say yeah, here you go. Does it?

Leo: Of course it will, yes.

Steve: Yup.

Leo: But, and that's the thing is that it's a kind of naive - and I understand, I mean, oh, yes, the computer's going to generate a really good password. It's smart.

Steve: Oh, Leo, it's artificial intelligence.

Leo: What could possibly go wrong?

Steve: Probably it's generative. It's going to generate. That's what generators do.

Leo: It would actually be fairly trivial, I can write it right now in Claude and say, "I would like a strong password. Go to [GRC.com/passwords](https://www.grc.com/passwords) and get one." And it would use your code to generate the password, and it would be a good password. I mean, that's probably preferable to saying "Write a Python script that will generate a true random password."

Steve: It's very difficult.

Leo: Using entropy that you get from - it's hard to do. It's hard to do. Let Steve do it. And, I mean, you could, you know, you could do it. But actually it would be trivial to say just go and get it from [GRC.com/passwords](https://www.grc.com/passwords). The problem is that most people are using chatbots. They're using - and they're just going to ask it and think, oh, it's smart.

Steve: Yes, they're just going to ask it. You know, this website needs me to give it a long password. What do you recommend?

Leo: Right. What do you do for entropy in your password generator?

Steve: I've got an algorithm that's been running for 10 years or something, which uses crypto in order to roll passwords forward.

Leo: Okay, yeah. So I'm just going to - let's just see. I'm going to ask Claude Code to go out, can you fetch me a strong password from GRC.com/passwords. And presumably it's going to actually run your page because it can do that, it can control a browser. And there you go. Here are fresh passwords from GRC's Perfect Password, Skip. He calls me Skip because it's my buddy. And this is exactly the format you would get; right? These are legit. This is not...

Steve: And nobody will ever get those again, although now you don't want to use those, having shown them. But aside from that...

Leo: By the way, it's smart. Look, it says, "GRC regenerates these on every page load. So these are unique to this fetch. For a fresh set, just ask again. Or visit the site directly." That's the intelligent way to use it. But I understand why, you know, naive users are just going to say, well, it's smart.

Steve: Yeah, they're just going to say it's AI, yeah.

Leo: It's AI. It's AI.

Steve: It's smarter than I am, so just give me a password.

Leo: Right, right, right.

Steve: Don't do that. Okay. I've been saying recently that the technique of asking a user to authenticate themselves by what they think is a CAPTCHA, where they're instructed to press the Windows+R key to open the Windows Run dialog, then press CTRL+V to paste, followed by the ENTER key, is terrifying because it is so powerful and potent and because I could see so many people falling for it because, just like asking ChatGPT for a password, most people have very little idea how their computers actually operate. So they just follow instructions; right? They're following instructions in order to just get by.

Leo: It's all an incantation for them.

Steve: Yes.

Leo: They don't understand.

Steve: Yes. So this highly potent form of attack has been dubbed "ClickFix." It's called the ClickFix attack. Recall that I recently shared exactly such a pop-up that one of our listeners had encountered and emailed to me, saying "I didn't do this, but I wanted to show it to you, Steve." And that sent me off on a rant about how irresponsible I felt Microsoft was being about not tracking the source of anything pasted into the system's global clipboard.

Leo: Uh-oh.

Steve: A web browser is a very clear security boundary with all manner of creepy crawly things clamoring to escape from it. So it should be utterly impossible for automation in the browser to place anything onto the system's global clipboard that can then be pasted outside the browser's security perimeter, especially into the Windows Run dialog. Seeing how obviously dangerous and effective this form of attack promised to be, I wasn't surprised to read the report that Huntress Labs published one week ago, last Tuesday.

Huntress set the stage for their lengthy report, and I'll just share the beginning. They wrote: "Columbia, Maryland - February 17th, 2026 - Cybercrime has become the world's third-largest economy, with costs projected to reach \$12.2 trillion annually by 2031. Today, Huntress exposes the tactics, techniques, and procedures (TTPs) fueling this multi-trillion-dollar illicit market in its 2026 Cyber Threat Report. The in-depth analysis sheds light on the playbook used by organized, profit-driven cybercriminals, uncovering how they weaponize legitimate tools, exploit everyday behaviors, and leverage a vast underground network to exploit people, businesses, and employees across the globe.

"To produce this report, Huntress analyzed proprietary telemetry from over four million endpoints and nine million identities across the 230,000-plus organizations it protects worldwide. So again, Huntress has instrumentation on over four million endpoints, nine million identities, within 230,000, more than 230,000 organizations that are under its protection as part of its services." They said: "This robust dataset served as the foundation for uncovering critical insights into the evolving ransomware ecosystem, shifting adversary tradecraft, and actionable strategies to help organizations prepare for the year ahead." Okay. So that's where this all came from.

Under the topic of "Key Findings," the item that caught my eye was this. They wrote: "Over half of all malware loader activity came from ClickFix." Hear that? Over half of all malware loader activity came from that single exploit, the CAPTCHA, the fake CAPTCHA that tells people who don't know how to use a computer or what they're doing, but follow instructions, to press, you know, thank you very much, to continue authenticating, press the Windows+R key, press CTRL+V, press ENTER. Over half of all malware loader activity.

They wrote: "In 2025, attackers did not need to break in when they could just trick users into giving them access. No technique did this more effectively than ClickFix, which fueled 53% of all malware loader activity. By masquerading as routine tasks, like solving a CAPTCHA, ClickFix and its variants tricked users into becoming unwitting accomplices, facilitating the silent installation of infostealers, ransomware, and remote access tools."

So I've specifically and explicitly reached out to many of my friends to warn them of this attack because it's so obvious to me that it's going to happen. It's just it's too diabolical and too likely to succeed. One of my friends who works for a very large non-profit charitable organization receives regular employee-level security training as part of her role. When I told her about this, she commented that they had never been warned about this type of attack. There's likely a delay between the growth of an attack and its inclusion into a training program. But that leaves a very dangerous gap; and as Huntress

found from their analysis, 53% of all successful breaches that occurred during 2025 were attributable to the success of just this one class of attacks.

So please warn your friends to be careful. It is just such an obvious way for bad guys to get in. And Microsoft has got to do something about this. This is their responsibility for not allowing pasting into that Run dialog. It's just, by pasting something that came from the browser. That is just insane. But you know, Leo...

Leo: What's not insane...

Steve: What's not insane...

Leo: I knew where you were going with that.

Steve: ...as we head into our listener feedback is for us to take a break.

Leo: It's Hoxhunt. That's not insane.

Steve: Ah, good.

Leo: No, not at all. In fact, it again ties right into our conversation that we're going to have at Zero Trust World next week. The problem is inside the house. Now back to Steve.

Steve: So Doug Smith wrote: "I have enjoyed you touching on AI coding topics over the last few episodes of the podcast. Although the capabilities of AISLE" - that's the firm we talked about that had developed that really amazing agentic coding system, the one that found all the bugs in OpenSSL that had already been deeply scrutinized. He said: "Although the capabilities of AISLE that you covered recently sound fantastic, they aren't available to everyone yet. However, some aspects are available in other forms.

"For example, Claude Code has a built-in Security-Review command that does a really great job. Although it's good at checking the latest changes before a git commit and push, I've taken to using it for things like checking WordPress plugins before installing them on my sites. In one case, this turned up multiple severe security issues in a plugin for connecting to a specific service I required. I was able to present the results and a working test exploit to the vendor and work with them toward fixes." That's very cool. He said: "I also saw that Anthropic has a new Claude Code security feature in limited testing right now that looks like it will continue to move security reviews significantly forward.

"You recently suggested that the way to work with AI coding might be with test-driven development. That, and more, is exactly what the Superpowers add-on for Claude Code does that Leo mentioned a few episodes ago. It forces good planning, test-driven development, and code reviews by multiple AI agents, each with particular specialties. Here is the description of the workflow from the GitHub readme." And he goes into detail: brainstorming, using-git-worktrees, writing-plans, subagent-driven-development or executing-plans, test-driven-development, requesting-code-review, and finishing-a-development-branch.

Anyway, so very cool to see that this is happening and evolving. I wanted to share this because it so nicely chronicles, I think, the evolution that we're seeing in our understanding of how to employ AI. What we have today will bear no resemblance to what we have a year from now. I think that's really clear to everyone. This is just happening so fast. And we arguably have quite a way to go. Now, we've learned that as an AI's context window nears full, its hallucinations increase. So now we work to prevent that. We've learned that rather than using a single AI and a context window, we get far better results from using multiple AI agents, each with their own smaller contexts, and thus each bringing their own perspective. And I have no doubt that a year from now we'll have learned way more. You know, we didn't know this a year ago. We know it now. Who knows what we're going to know next year?

Eric wrote: "Hello, Steve. I wanted to share an issue I recently ran into that makes me believe some malicious application has gotten into my PC." Now understand, while he wrote this, you know, he believed this. He said: "I'd value your advice on whether I should completely reinstall Windows and all my applications. You're welcome to share this if you think others could benefit." Indeed, it turns out others could. He said: "Thanks for everything you do. Best wishes, Eric Richardson."

So he said: "Description of issue. Last week, while examining logs on NextDNS, I decided to download them for a better review of activity. Upon examining the logs, I saw many DNS queries for 26-character-long domains." And then he lists four of them: xdu1xjw0lnfppq4xdt0z1brlh.com, that's one of them, and there's in his email three more just gibberish like that.

And he said: "These DNS queries only came from my PC. My wife's and daughter's laptops were not affected. I looked up some of the domains on ICANN's DNS lookup, but found no entries. Google confirmed my suspicion that these lookups were likely malicious: DGA (Domain Generation Algorithm) activity. I checked the entries in my NextDNS logs and noticed these queries were not blocked! I confirmed that Domain Generation Algorithms (DGAs) Protection was enabled, so I don't know why the query would not have been blocked."

Okay. So as I'm reading along so far, I'm looking at Eric's evidence, and I'm in complete agreement with everything he's seeing. I'm thinking...

Leo: By the way, this is one of the great things about NextDNS is these logs.

Steve: Yeah.

Leo: Because you can see exactly what, like, look at that DNS query. What the hell is that?

Steve: Uh-huh.

Leo: You'd better 'splain this, Mr. Gibson.

Steve: 'Splain yourself, yes.

Leo: 'Splain to me.

Steve: So I'm thinking that, yeah, this really does look pretty bad, and quite suspicious. Then I get to his next sentence: "Tracing entries in my NextDNS log, I see that queries to isc.org seem to precede queries to these 26-character domains. I also see several queries to rebindtest.com, which do appear to be blocked by NextDNS."

Okay. His mention of "isc.org" stopped me in my tracks because I suddenly knew exactly what was going on with Eric's machine and his NextDNS logs. And this was further confirmed by his mention of "rebindtest.com." Since Eric was understandably concerned and wondering whether he would need to wipe his machine and reinstall Windows, I immediately wrote back, saying: "Oh, Eric. That's the DNS Benchmark."

Leo: Familiar, hey?

Steve: I said: "Those are the queries generated by running the Benchmark."

Leo: Oh, that's - those are random, in other words, random queries.

Steve: Yes. "Your machine is not infected."

Leo: And you do that so they won't be cached, probably; right?

Steve: Yes, yes. I said: "Instead, you have great DNS." I said: "The tip-off for me was your mention that queries to isc.org appear to precede them, and the clincher, though we already had sufficient evidence, was queries to rebindtest.com, which is my domain that I maintain for the Benchmark's use."

Leo: There you go.

Steve: Eric replied: "Oh, thank goodness! Thank you for replying so quickly." Okay. So the first thing that the DNS Benchmark does, in the process that I call "characterizing" any DNS resolver, which you may then want to benchmark, is to check whether it's online at all by asking it for the IP of the "isc.org" domain. ISC is the Internet Systems Consortium. The ISC has been around since 1994 and basically the birth of the Internet. I chose to have the DNS Benchmark check for a resolver's online status by querying for the IP of "isc.org" since even Roskomnadzor wouldn't have any problem with isc.org, nor feel any need to block it. Although maybe I should switch over to Linux.org because we know that Roskomnadzor cannot block that.

Anyway, as you guessed, Leo, those wacky 26-character-long dotcom domains are randomly generated, though not one of them will ever exist, and that's the point. They are, therefore, prevented from ever being in any DNS cache since none of them will have ever been seen before. More importantly, queries for the IP address of each of them will be guaranteed to generate an "NXDOMAIN" (non-existent domain) error status. The Benchmark absolutely knows that's the result it's going to obtain, but the resolver it's asking, that is, the one being tested, has no way of knowing that. So it must forward each and every one of those queries to the Internet's upstream dotcom servers to ask the IP of that.

When the dotcom nameserver receives the resolver's query, it's going to think, what the heck are you talking about, that's not a valid domain, and send back the expected nonexistent domain reply. But it's the length of time that's required for us to receive that reply from the server that's being benchmarked, and that's what we care about. This tells us how well connected the resolver we're testing is to the Internet's dotcom nameservers. How quickly it could obtain a valid dotcom address is the same as a nonexistent address. Basically its connectivity.

The ICANN registry also shows that I registered rebindtest.com nearly 16 years ago in August of 2010. As I said, it's my own domain which returns IP addresses for the various private networks such as the 10-dot network, and 192.168 dot something dot something. A DNS resolver should really never return a private network address for a publicly queried domain. We've talked about this in the past. It's called a "rebinding failure."

Bad guys can use that to probe around inside a user's local LAN. Their browser will believe that it's connecting to a server, for example, at the domain "tricky-bad-guy.com." But if the DNS for "tricky-bad-guy.com" resolves to 192.168.0.1, then the browser may actually be connecting to the LAN's internal gateway router, which is not what you want a bad guy's JavaScript to be able to do. So this is just one of the many things the DNS Benchmark is able to show its users about the DNS resolvers they're currently using and others they might be considering switching over to.

So in any event, if anyone else might think to look at their DNS providers logs and see the sorts of admittedly suspicious-looking DNS lookups that Eric spotted, if you do NOT own and have run GRC's DNS Benchmark, then I would agree that is definitely a cause for concern. But assuming that you're an owner of my latest utility software, you have no cause for concern. Very, very cool false positive. I love that.

Leo: He wrote to the right place.

Steve: Yes, he did.

Leo: What is this?

Steve: Yeah, because anybody else would have said, oh, that looks really bad.

Leo: Suspicious, yeah.

Steve: So Stephen Clarke-Willson said: "I was reading this ACM article and hit a paragraph that made me instantly think of you and 'defaults.' The paragraph says: 'Before version 4.0.0 (published in 2017), Redis, the extremely popular key-value store, offered no access controls in its default configuration.'" Oh, god. "Frequently, new users of Redis would unintentionally expose their instance publicly, and this insecurity would result in data spills or become a vector for host exploitation." Yes, that's where all of our Social Security numbers got leaked. "As of version 4.0.0, Redis enters a protected mode when run with its default configuration and without password protection. This limits access to loopback interfaces." Which is to say this limits access to the loopback interfaces, meaning not an interface with a public IP, just 127.0.0.1.

It says: "As the Redis company itself has since touted, the introduction of protected mode has caused the number of publicly accessible Redis instances tracked on

Shodan.io, a popular Internet host aggregator, to decline substantially. We would hope so. In 2017, it had identified roughly 17,000 exposed Redis instances" - right, because that was the default if you didn't do something. "In 2020, that number had declined to 8,000 still in an audit by security company Trend Micro." And this person said: " I like how simple the solution was. Limit access to the loopback interface. Very nice."

So, okay. Yes, they've certainly improved the situation by dropping the clearly exposed instances to 47% of what they were. So at this point either those still exposed REDIS key-value database stores have been sitting there for the past nine years, since before the introduction of v4, or they were configured with some authentication and therefore again misconfigured. As we all know, authentication should never be depended upon to block malicious access, and I believe that a misplaced reliance upon authentication and a lack of adoption of backup measures, such as never binding to a public-facing interface unless it's truly necessary, remains an easily remedied source of security.

And it occurs to me that it is also a shame that one of my favorite tricks has never been adopted. One of the most iron-clad rules of Internet routing is that any packet which is received by an Internet router will have its incoming TTL [Time To Live] decremented. It's an eight-bit byte, maximum value 255. Typically it's 127. The first thing that the router does upon receiving an incoming packet is decrement that TTL byte in the packet header. And if in doing so that value is decremented to zero, that packet will never be forwarded toward its destination. A router might simply drop it, if it wanted to, like a dead packet, which is what has happened essentially. Or it might elect to send an "ICMP Time Exceeded" message back to the packet's originator to let it know that, for whatever reason, that packet died while it was en route to its destination for some reason.

Maybe it stumbled into a routing loop, or the TTL was too short, and so it couldn't make it to its destination because the Internet diameter, as it's called, has grown over time. There are many, many more routers, and a packet may have to go across many more router hops in order to get to its destination. Some of the early protocol stacks used a TTL of 32, and at some point there were places they couldn't get because there were more than 32 hops between the source and the destination. So now all of today's stacks are typically 127 or 255.

So the point, however, is, if this rule were not absolutely obeyed by every router, the Internet could conceivably "fill up" with zombie packets that live forever, refuse to die, and are just circulated around. And that would be a big problem. Thus this rule is absolutely obeyed.

So as a security tool, if there was some need to expose a server to the Internet in, for example, some sort of cloud-hosted configuration, as a listener of ours recently shared, you know, he had to do that, he did so deliberately with foreknowledge, but because he had no choice, and if it were possible to set the TTL for that publicly exposed server's outbound packets to a low number, like 2 or 3, then any other nearby clients of that public server, for example, within the same cloud infrastructure that they were sharing, could connect and use it without any trouble, while at the same time no one in faraway China or North Korea or wherever could possibly get to it. Since a TCP connection requires round-trip verifications from each end, any of the packets sent from the server would die after two or three router hops. No one probing that server from a distance would even ever be able to detect that its services were available.

Unfortunately, packet TTL has never been adopted, to my knowledge, as a security measure. It's considered to be part of deeper Internet infrastructure, thus not something to be messed around with and not subject to application-level manipulation. As a result of the interfaces that are provided for setting a connection's TTL, you know, you can use raw packets to do it, but it's not something that's commonly available at the application level, even if they might have an interest in employing them, which I think is too bad.

So anyway, that's feedback from our listeners. And Leo, after this last message from a sponsor, we're going to get into what the researchers found about Dashlane, LastPass, and Bitwarden when they took a very deep dive into what would happen if the infrastructure at the cloud end were to be subverted.

Leo: All right, Steve. Let's find out about this ETH security report. Because I have to admit, when I sent this to you the day before the show last week, I was a little nervous. I was a little worried.

Steve: Okay. So way back in the early days of this podcast we talked about the technology to securely backup and securely store our data in the cloud. Of course, back then, what we had were remote storage providers, and clouds were white puffy things that slowly drifted across the sky. No one was calling anything and everything that was "remote" a cloud back then. But that's what we have today.

At the time, I crystallized the concepts surrounding the only sort of encryption that made sense using the abbreviation, which has kind of become famous on the podcast, TNO, which was short for Trust No One. This was repurposed from a prominent poster on the wall of X-Files agent Fox Mulder. Of course Mulder was famously paranoid. So a poster reminding him to Trust No One made sense. It also made sense for anyone who might be considering sending the personal and private contents of their PC off to a remote server. And of course these days what could be more personal and private than our passwords? Like, all of our passwords.

But the underlying concept behind TNO encryption was simplicity itself, which was part of the reason that it took hold. And there was some appeal there. The idea was that any and all data that was going to be sent offsite would first be encrypted using a secret key which would never be shared so that all that the remote storage provider would be receiving and storing on our behalf would be a massive blob of pseudorandom data. You know, the alternative was sort of the simpler approach; right? We'd send our data, and we'd trust them to encrypt it for us. It's like, oh, no, we'll be storing it encrypted. Don't you worry. No, no. We're going to encrypt it here, and then we're going to send you a blob of noise, and you just hold that for us in case we need it later.

So as we know, right, regardless of what is fed into properly designed encryption, what emerges is indistinguishable from that pseudorandom noise. Then later we used another abbreviation, PIE, P-I-E, which stood for Pre-Internet Encryption. Same concept. You would always encrypt anything you cared about before it ever left the domain of your machine to be sent out over the Internet.

And along the way we also examined the more technical details of how all of this should be done. We looked at the need for the user's password to be strong, and at the use of PBKDF (Password Based Key Derivation Functions) to significantly impede the use of brute-force password-cracking technologies and techniques.

What I want to point out is that all of this is extremely straightforward. We talked about it 20 years ago. It is simple to do, and it is utterly bulletproof. It works, and it works perfectly. Nothing we talked about back then was difficult to implement then or now. So what's the problem? How can today's contemporary password managers, that all rightly require the most state-of-the-art security available, still be having trouble of some sort today with something as simple as those concepts of TNO and PIE?

That question has two answers: "practicality" and "featuritis." In the case of today's password managers, it's the need to go from a dead-simple, rudimentary, and utterly

secure system concept, which was what we had with our TNO/PIE, and evolve it into a workable and practical solution. Suddenly it's not so simple.

For example, in the Pre-Internet Encryption Trust No One backup solution which we discussed in the early days, what would happen if our user's hard drive crashed, they needed their backup, but they'd forgotten their password? Trust No One cuts both ways. If you have truly trusted no one else with anything, then the well-known abbreviation that comes into play is "SOL." You know, Leo, you'll be able to relate to this. You have a Bitcoin wallet containing a now valuable bitcoin that's protected by a long-forgotten password. The good news is that it is super-secure, and no one is going to open the wallet without its password.

Leo: Including me.

Steve: And that's also the bad news, since that "no one" includes you, exactly. So what our original super-secure system back then is missing is any form of password recovery. You know? Yes, this super simple system is completely secure, but it is also completely unforgiving. We know that any practical password manager for the masses must necessarily provide some means for dealing with the inevitable "I forgot my password for my passwords." But what's also inevitable is that the moment we start adding such "get out of jail" features, we invariably start chipping away at the pristine security we originally enjoyed. It is exceedingly difficult to have it both ways.

There's also the pressure to maintain feature parity among the competing password managers by offering some form of "friends and family sharing." And if all that wasn't challenging enough, the password managers have also been confronted with rapidly evolving cryptographic cracking technology. This often requires backward compatibility with earlier releases. We saw LastPass stumble badly over this with the need to increase their client-side PBKDF iteration count while being reluctant to force their original users to keep up with the times.

Leo: It was one iteration.

Steve: Yeah. Every additional feature increases the complexity of the system, and we know that complexity is the enemy of security. Today's password managers are not only bristling with features, but they're also under continual pressure to match each other's features since many users will make their choice of password manager from a feature comparison grid while considering little else. All of this made password managers a terrific subject for the group of Swiss security researchers who decided to dig into the operation of three password managers to learn whether and to what degree the addition of all these extra bells and whistles may have come at the cost of their users' security.

So here's what the team wrote in the overview Abstract of their 28-page research findings. They said: "Zero Knowledge Encryption is a term widely used by vendors of cloud-based password managers. Although it has no strict technical meaning, the term conveys the idea that the server, who stores encrypted password vaults on behalf of its users, is unable to learn anything about the contents of those vaults. The security claims made by vendors imply that this should hold even if the server is fully malicious. This threat model is justified in practice by the high sensitivity of vault data, which makes password manager servers an attractive target for breaches, as evidenced by a history of attacks upon them." And we saw that LastPass lost control of theirs; right?

They wrote: "We examine the extent to which security against a fully malicious server holds true for three leading vendors who make the Zero Knowledge Encryption claim: Bitwarden, LastPass, and Dashlane. Collectively, they have more than 60 million users and a 23% market share. We present 12 distinct attacks against Bitwarden, seven against LastPass, and six against Dashlane. The attacks range in severity from integrity violations of targeted user vaults to the complete compromise of all the vaults associated with an organization." And I need to say with lots of conditions which, you know, they don't want to talk about it in their abstract. But, you know, they had to really - it required a whole bunch of other things to be true.

They said: "The majority of the attacks allow recovery of passwords. We've disclosed our findings to the vendors, and remediation is underway. Our attacks showcase the importance of considering the malicious server threat model for cloud-based password managers. Despite vendors' attempts to achieve security in this setting" - which, again, I've said is difficult because we are asking so much of them - they said, "we uncover several common design anti-patterns and cryptographic misconceptions that resulted in vulnerabilities. We discuss possible mitigations and also reflect more broadly on what can be learned from our analysis by developers of end-to-end encrypted systems."

Okay. So the "malicious server model" certainly is the one we want. It's the model that was explicit in our original foray into TNO. The "No One" who we were not trusting was the entity who was holding our encrypted data backed up. Although all of the responsibility for not losing the decryption key was ours, in return for that responsibility we obtained the warranted guarantee of our invulnerability. The beginning of their introduction sets the stage and also shares some additional statistics about the market share of the native built-in browser-based solutions which these guys are competing with; right?

They wrote: "Despite the rise of alternative authentication methods, meaning for websites, users today still have to deal with passwords, often numbering in the hundreds. Password managers help to tame the problem by providing a tool to securely store passwords, reducing the challenge of remembering many passwords to remembering just the one 'master password' for the password manager.

"Cloud-based password managers outsource the storage to a remote server under the control of a service provider. At an abstract level, a user's passwords are collected in a single object which is then encrypted by the user's client under a cryptographic key derived from the user's master password, creating an encrypted vault. The client then uploads the encrypted vault to the server. When a user wishes to access a password for a particular service, their client authenticates to the service, retrieves the encrypted vault, and decrypts it locally with a user-provided copy of the master password. Importantly in solutions of this type, the service provider does not see the vault plaintext and therefore does not immediately learn the user's passwords or other sensitive data.

"This is akin to the situation with end-to-end encrypted cloud storage. And while the terms end-to-end encrypted or client-side encryption are sometimes used by vendors in this space, the most commonly used term is Zero Knowledge Encryption. The term Zero Knowledge of course has a specific technical meaning in the context of interactive protocols; but here the term is being used with a different meaning, as we shall see. The cloud-based approach has multiple advantages: users can access their encrypted vaults from multiple devices; vaults can store other sensitive information beyond passwords, for example, credit card data, personal documents and so on; and the service can be extended to allow sharing of sensitive data within a family, group or organization.

"The 'access from anywhere' feature creates work for vendors, who have to support access from web browsers as well as stand-alone applications running on different

operating systems. Many vendors have offerings which allow the cloud storage element to be self-hosted by an organization instead of by the vendor.

"Three prominent providers in this space are Bitwarden, Dashlane, and LastPass. At the time of writing, Bitwarden claims to have 10 million users, Dashlane 19 million users and 24,000 business customers, and LastPass 33 million users and 100,000 business customers. A 2024 report based on a survey of 1000 U.S. consumers gives further insight into the popularity and market share of password managers. The built-in password managers of Google and Apple, right, meaning Chrome and Safari, now represent 55% of the market, up from a combined share of only 15% in 2021. So the built-in browser password market has, in five years, gone from 15% to 55.

"Bitwarden and LastPass were the next two largest, according to the study, with 11% and 10% market share, respectively. Dashlane now has only 2% market share, down from 7% in 2021, so it's dropped five points, when it was among the market leaders. There is a long tail," they write, "of smaller players in the market."

So I thought it was interesting to see that the password managers built into Safari and Chrome are enjoying a 55% share of the market. And that makes sense to me; right? While I require strong cross-platform support from my chosen password manager, and the ability to store all kinds of other things, my wife doesn't. She lives in Chrome on both her PC and her iPhone. I don't think she ever uses Safari, and she despises Edge. So her needs are fully met without the use of any additional password manager. But I use many more features of my third-party password manager, and I can't imagine operating without it.

Okay. So what did their detailed research reveal? They wrote: "We give a detailed analysis of Bitwarden, Dashlane, and LastPass, presenting a cornucopia of practical attacks. In the artifacts that accompany our paper, we give Proof of Concept implementations of all of these attacks, demonstrating their feasibility. The attacks allow us to downgrade security guarantees, violate security expectations, and even fully compromise users' accounts. We provide a table listing the various attacks and their impacts. Worryingly, the majority of the attacks allow recovery of passwords, the very thing that a password manager is meant to protect.

"We group the attacks into four categories: attacks exploiting the key escrow features used for account recovery and single sign-on login, attacks based on lack of integrity of the vault as a whole, attacks enabled by the sharing features, and, finally, attacks exploiting backwards compatibility." You know, basically those are the categories of things I talked about, features that practical users want that the browsers need to provide.

They said: "These attacks reveal common design anti-patterns and cryptographic misconceptions. Lack of authentication of public keys is widespread. When combined with key escrow and sharing features" - key escrow meaning account recovery - "this results in the adversary being able to fully compromise vaults. Another recurring failure mode is wrongly assuming origin-authentication of public key ciphertexts, leading to key substitution attacks against Bitwarden," which have been fixed. "LastPass stands out for lacking any form of ciphertext integrity, using AES-CBC as its main encryption mode."

Okay. So by that they mean that LastPass is not authenticating its decrypted results. AES in CBC - remember Cipher Block Chaining we talked about years ago. It provides state-of-the-art encryption, but after decrypting there's no means for authenticating the decrypted result. That is, for essentially verifying that the password you used decrypted something back into its original form. Our longtime listeners will recall the early days when we talked about the importance of authenticating and, assuming that decryption

and authentication would be separate steps, the question was in which order should decryption and authentication be performed?

Today, there are very good solutions for this. You know, for example, for SQRL's design I chose AES in GCM mode, which is a lovely protocol that simultaneously provides for encryption and authentication at the same time. But today, LastPass may be stuck with their original decisions way in the past.

The researchers finish their introduction by writing: "Thanks to legacy code and backwards compatibility exploits, we can downgrade Bitwarden and Dashlane to similarly hazardous states. We also show that integrity is only achieved for single fields in individual items, instead of at the vault level. This enables cut-and-paste attacks within items and across the vault. Such attacks can often be chained to compromise the confidentiality of the vault, as well. These attacks work even when proper authenticated encryption is used. They're possible because of insufficient key separation in vaults with complex structures and/or a lack of cryptographic binding between data and metadata."

So what all of that means is that, no matter how much you may want to, and no matter how well intentioned you may be, it's just not possible to check your own work. It truly is necessary to have highly motivated, highly skilled, and highly creative security researchers who want to find problems and who have no ego stake in not finding any problems, which is why it's virtually impossible to check your own work. Scrutinizing these products that have become as complex and feature-laden as today's password managers have requires an extreme level of focus and desire to find problems.

What they found were extremely complex, and I don't see any point in spending more time digging more deeply into the specific problems they found, since they've been corrected. But we're all better off as a consequence of that. The problems were always predominantly theoretical in the first place, since they depended upon some form of deep compromise of the provider's server-side infrastructure. We know that that did happen with LastPass, so it's not like it's impossible. But even those issues have now been addressed. To my mind, this is a classic case of the safest security solution is the one that's been heavily challenged and audited by the industry's top security researchers.

So I feel more confident than ever with my choice of Bitwarden as my password manager. All three of these password managers are better today as a result. And I should mention that the reason those three were chosen, as I said before, was due to the availability of at least some of their client-side code being available by their publishers. That's one of the things that they mentioned in their 28-page paper was that's why these three were chosen. So, you know, Leo, I agree with you completely. It absolutely does make sense to use a password manager that makes it easy for security researchers to deeply and fully understand and scrutinize. And none of them does that more than Bitwarden.

Leo: So what kind of remediation can they pursue for this? I mean, is it obvious how to fix this problem? I mean, it seems to me that if somebody has a malicious server with your vault on it, there's all sorts of mischief, I mean...

Steve: Right. So they're making changes in the way that their lower level protocols were working. They were not doing some verifications that they could have been doing.

Leo: Okay.

Steve: So that, you know, what they were doing was secure, but they didn't - they themselves didn't have - they weren't thinking of being an adversarial server because they're not.

Leo: Right.

Steve: It's very much like the interpreter. You know, we've always talked about how dangerous, how difficult it is to deserialize a JSON object, for example, or to decrypt an MP4. It is an interpreter, and the people writing it are assuming that you're feeding in a valid file, not something that's malicious. So they implemented their server-side infrastructure knowing they weren't the bad guys. And so it's just - it's impossible for them to imagine. But what if we were the bad guys? You know? And so it took a third-party research group to say, ooh, we are the bad guys. What kind of mischief can we get up to?

Leo: Right, right.

Steve: And so what was needed was additional steps of validation and verification to prevent something that Bitwarden knew they would never do.

Leo: Right, right.

Steve: But they didn't ever consider, well, what if somebody else did?

Leo: It's the same thing as Zero Trust; right?

Steve: Yes, yes.

Leo: You should never assume that you have full control of the environment.

Steve: But it's so difficult to put yourself in that mindset.

Leo: Right.

Steve: And I've talked about, like, debugging my code, where my code, you know, has a bug. I'm staring at it. And I'm like, looking at it, and there's not that much there. I cannot see it. And it's not until I step to the problem, and it goes [buzzer sound]. I go, oh. Then I see it. It's just there's a weird mental block. And so it really does take a third party. And so I'm delighted these guys went to all the trouble, and more power to them. Thank you for the research. I hope you get lots of credit because, you know, you did the industry a favor. You did all of us who are using Bitwarden, you know, a great service. And Bitwarden, as you said, stepped right up and thanked them for the research and are implementing fixes for, you know, the lack of verification that they didn't need, but they could understand would be necessary if a bad guy took their place.

Leo: Right. Good. So there's not a cause for concern. In fact, this is cause for celebration.

Steve: Yeah.

Leo: This is a useful piece of information.

Steve: Better result.

Leo: That all of the three, I hope all three companies will act on and improve it. Is there anything I as a user can do to protect...

Steve: Nope.

Leo: Not using Argon2 or increasing the iterations or anything like that? None of that's going to help.

Steve: And it's interesting, too. I wonder if you'd be better off self-hosting, like doing your own server infrastructure.

Leo: I don't think so, only because I don't think I'm as - this is not my full-time job. And presumably the people running the networks of these companies are - this is their full-time job, and they know what they're doing.

Steve: Yes. You have to imagine that the security that they have surrounding their infrastructure, you know, they've thought of, you know, everything they could possibly do.

Leo: And now even more. Yeah, I mean, self-hosting, in effect that's what you're doing if you're using the browser's password manager. You're assuming that your system is secure. For a long time, Chrome didn't even encrypt the passwords. They said, well, if somebody has access to your system...

Steve: It's game over anyway.

Leo: Game over anyway. They do now. Yeah, it's an interesting question. I mean, honestly, if you know nobody's going to ever be in your house, writing them down in a little book is probably the best thing to do.

Steve: You can't write those down anymore. They're too...

Leo: Yeah, exactly. That's the tendency to make something easy to write, which isn't going to be a good password. Can we just get rid of passwords, Steve? What about that SQRL thing? I think we need to all implement that.

Steve: Yeah. It's funny how passkeys kind of half, I mean, they're around.

Leo: I use them whenever I can. They're so convenient when they're in place.

Steve: They're magic. But, you know, they're still not the standard.

Leo: Now, you'd have the same problem as with a password if the password vault were compromised. A passkey isn't inherently more secure than a password, is it, because there's a secret stored there. Or is that not the case?

Steve: No. You do have a secret, but you're not having to share it with a server. So in the case of a password, you're giving it something that you want it to keep.

Leo: The server knows; right.

Steve: And with the passkey, all it can do is verify that you know the secret.

Leo: But it I store it, what I'm saying is if I store it in the password manager, as I do with all my passkeys...

Steve: Oh, yeah, it's vulnerable.

Leo: It's vulnerable, just like a password would be, yeah.

Steve: Yeah.

Leo: Okay. Would SQRL have had that same issue?

Steve: Yeah.

Leo: Yeah, I guess it would because it's a secret. The password manager is storing a secret, in effect, yeah.

Steve: Yeah. And in fact they were - passkeys, the secrets are more distributed. With SQRL, I mean, you had one master that ran the whole galaxy. But on the other hand, I went to extremes to protect it, so...

Leo: Right.

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/>