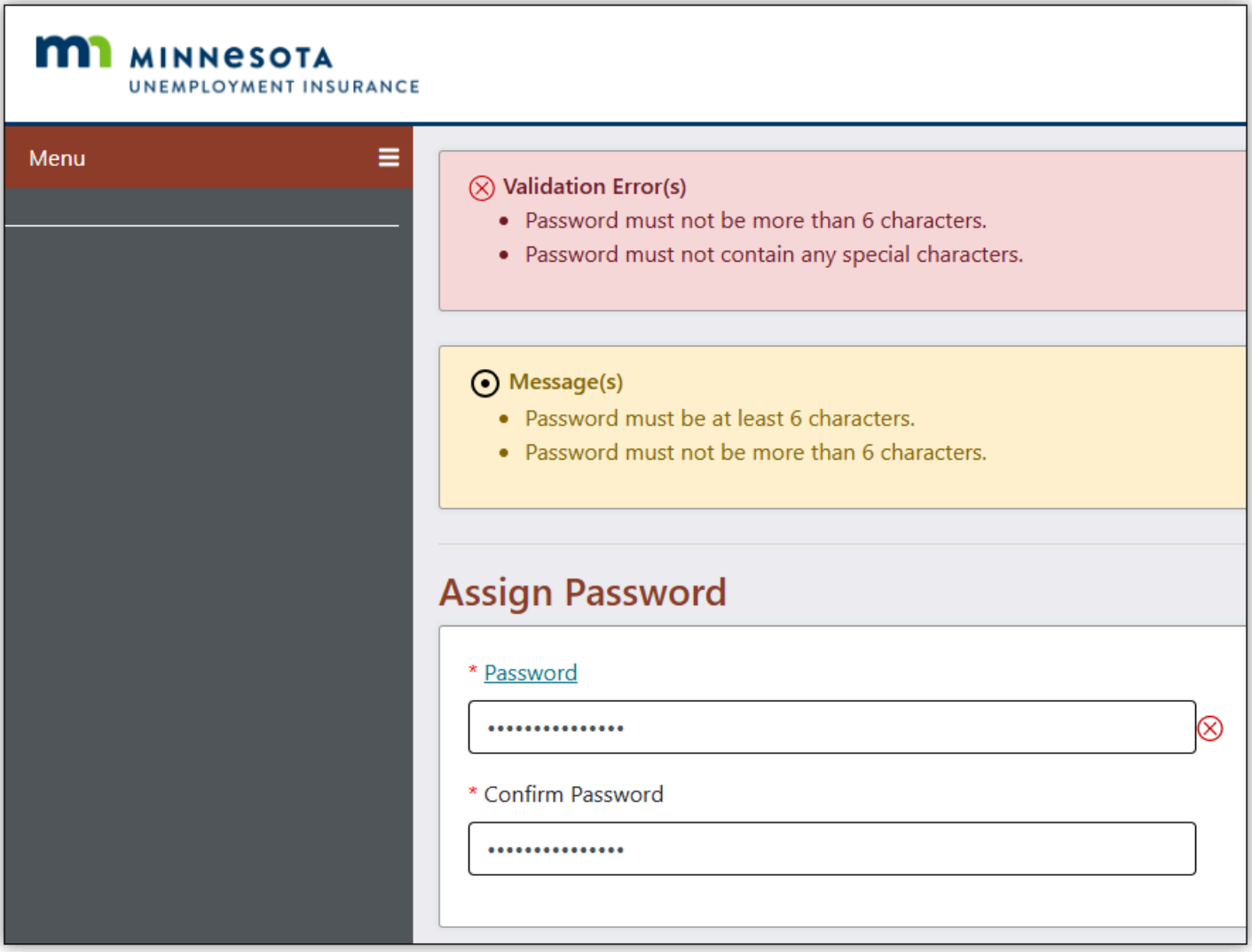
Security Now! #1016 - 03-11-25

The Bluetooth Backdoor

This week on Security Now!

Utah passes age verification requirement for app stores. The inside story on fake North Korean employees. Is that a Texas accent? An update on the ongoing Bybit cryptoheist saga. The industry may be making some changes in the wake of the Bybit attack. Apple pushes back legally against the UK's secret order. Did someone crack Passkeys? The UK launches a legal salvo at an innocent security researcher. The old data breach we witnessed that just keeps on giving. A bit more Bybit post-mortem forensic news. A lesson to learn from a clever and effective ransomware attack. And what about that Bluetooth Backdoor discovery everyone is talking about?

What year is this?
It seems we still have a ways to go...



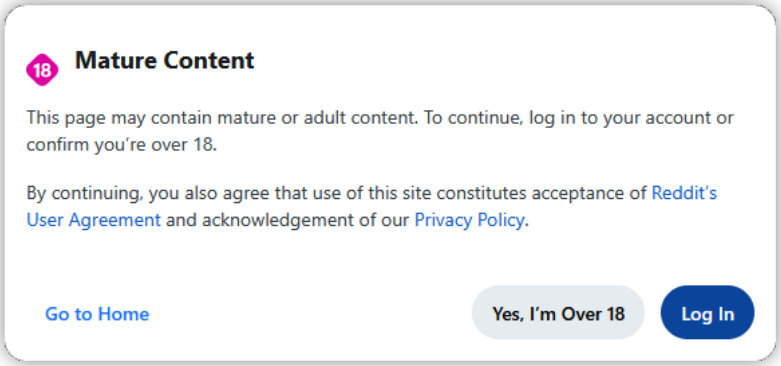
Security News

Robust Age Verification

Listeners of this podcast know how I feel about age verification: In the same way that we need to make peace with the thorny issues surrounding the abuse of the absolute privacy offered by modern encryption, I believe we must also squarely address the problem of verifying someone's biological age in cyberspace. Unfortunately, having given this a great deal of thought, this feels like another of those thorny and intractable problems; but let's explore this a bit.

I'm old enough to collect social security. So I have the legal right to sit at my desktop PC, do anything and go anywhere someone my age can legally do – which is pretty much anywhere and anything. But, I also want my privacy preserved while I'm wandering around. In the interest of preserving as much privacy as possible and only disclosing the bare minimum necessary, and when necessary, there's no need to share an exact date of birth; rounding age down to the integer number of years of life completed should always be sufficient.

But I also don't like the idea of having my age sprayed indiscriminately everywhere I go. So it should be on an as-needed basis. If I go to a website that has a reasonable need to verify my age, **and** if I agree with its need and **elect** to provide that, **then** I should have the **option** of releasing my integer age to that site one time, that one time.



Another consideration is that age restrictions vary by region. In the United States we do not yet have any uniformity across our individual and independent state legislations. And internationally, restrictions may vary by country. So it would likely also be necessary to be able to assert our country and state of residence as part of this voluntary age and jurisdiction disclosure.

Alternatively, perhaps I'm a Gen-Z'er who doesn't care at all about having their age sprayed across the Internet. In that case, the age verifier could be left unlocked with any querying website being informed of such a user's age and jurisdiction on the fly. If I'm in a household with younger kids, I could both lock and password– or PIN–protect this feature so that "something I know" would need to be provided any time I wished to assert my age in cyberspace.

So this would be another Internet specification for the W3C, the World Wide Web Consortium to design and standardize, and it would be implemented in and dispensed by our web browsers. Once this standard was established, any website that was legally obligated to verify its visitors age would, rather than presenting that ridiculous "Yes, I'm at least 18 years old" button, would have returned an HTTP reply header when displaying the site's home page. In the Gen-Z'er case where their browser was set to "permanently unlocked" their browser would return a query making the proper assertion and the site's content would be available.

But in the typical case, where a web user wants to exercise some explicit control, the receipt of this reply header would cause the user's browser to display its own uniform pop-up prompt, saying that the site being visited requires the user to verify their age and location. That pop-up would contain a button labeled "Please verify my age and send my location to this website." If the user agreed, the browser would generate a query containing this information and the website would open its doors.

The model would be very much like the cookie pop-ups we're now plagued with, but it would be implemented by the browser, not the website, it would be displayed screen-center and only when visiting sites requiring verification.

Now, of course, by this time everyone is thinking *"Yeah, okay, fine. But how can any user's web browser possibly know their date of birth and location in any way that cannot be spoofed at will?"* And everyone who's thinking that is 100% correct – that's the big problem. And it's not a problem that can be sidestepped, since that is the essential problem. But I wanted to first lay out the rest of the required framework to show that if that essential problem could be solved, it could be the basis for a workable solution.

Before I go further with that, I want to share the news from last week that triggered this re-exploration. Though it received wide coverage, The Verge's headline was: *"Utah becomes the first state to pass an app store age verification bill"* and they followed that with the note that "Meta, Snap, and X are applauding it." The Verge wrote:

> *Utah became the first state in the country to pass legislation requiring app store operators to verify users' ages and require parental consent for minors to download apps.*
>
> *The App Store Accountability Act is the latest kids online safety bill to head to the governors' desk, as states across the country and the federal legislature have tried to impose a variety of design regulations and age gating requirements to protect minors from online harms. Much of the legislation that has advanced through the states has been blocked in the courts, and the leading bill in Congress failed to pass last year amid concerns that it could limit free expression on the internet.*
>
> *Putting the onus on mobile app store operators to verify ages — rather than individual website providers — is something that Meta and other social media sites have pushed in recent months, as legislatures consider a variety of bills that could impose more liability for kids' safety across the tech industry. Apple reportedly lobbied against a Louisiana bill that would have required it to help enforce age restrictions, but recently voluntarily opted to let parents share their kids' age ranges with apps. Meta spokesperson Jamie Radice called that "a positive first step," at the time, but noted that "developers can only apply these age-appropriate protections with a teen's approval."*
>
> *After Utah passed its age verification bill, Meta, Snap, and X applauded the move in a joint statement, and urged Congress to follow suit. "Parents want a one-stop shop to verify their child's age and grant permission for them to download apps in a privacy-preserving way," they write. "The app store is the best place for it, and more than a quarter of states have introduced bills recognizing the central role app stores play." Apple spokesperson Peter Ajemian pointed to a white paper the company released last month, which emphasizes the importance of minimizing the amount of sensitive data collected on users. Google (which runs the Play Store on Android) did not immediately provide comment on the bill.*

But others, including Chamber of Progress, which counts Meta's European arm as well as Apple and Google among its corporate backers, warn that the bill could put all users' privacy and rights at risk. "The Supreme Court has long recognized that age verification requirements, like those in SB 142, chill access to protected speech for everyone and are therefore inconsistent with the First Amendment," Chamber of Progress legal advocacy counsel Kerry Maeve Sheehan writes in a blog post. SCOTUS is set to weigh in on age verification this year, but in a case that deals specifically with its application to accessing porn sites. "As privacy experts have explained, 'strict age verification — confirming a user's age without requiring additional personally identifiable information — is not technically feasible in a manner that respects users' rights, privacy, and security.'"

So once again we have political legislators imagining that they're able to dictate the way reality should operate. But Apple, apparently cognizant of the direction things were going, last month in February, published a short 8-page document titled "Helping Protect Kids Online":

https://developer.apple.com/support/downloads/Helping-Protect-Kids-Online-2025.pdf

It appears that Apple is grudgingly moving in the direction they need to go, which is to allow their platform to be used as an age verifier – much as they would clearly rather not. Their Helping Protect Kids Online document addressed this. Under the topic *"Making it easier to set up and manage accounts for kids"* Apple wrote:

For years, Apple has supported specialized Apple accounts for kids—called Child Accounts—that enable parents to manage the many parental controls we offer, and help provide an age-appropriate experience for children under the age of 13. These accounts are the bedrock of all the child safety tools that we offer today. To help more parents take full advantage of Child Accounts and parental controls, we are making two important changes.

First, we are introducing a new set-up process that will streamline the steps parents need to take to set up a Child Account for a kid in their family. And if parents prefer to wait until later to finish setting up a Child Account, child-appropriate default settings will still be enabled on the device. This way, a child can immediately begin to use their iPhone or iPad safely, and parents can be assured that child safety features will be active in the meantime. This means even more kids will end up using devices configured to maximize child safety with parental controls.

Second, starting later this year, parents will be able to easily correct the age that is associated with their kid's account if they previously did not set it up correctly. Once they do, parents of kids under 13 will be prompted to connect their kid's account to their family group (if they're not already connected), the account will be converted to a Child Account, and parents will be able to utilize Apple's parental control options—with Apple's default age-appropriate settings applied as a backstop.

And under the topic *"A new privacy-protective way for parents to share their kids' age range"* Apple wrote:

Later this year, Apple will be giving parents a new way to provide developers with information about the age range of their kids—enabling parents to help developers deliver an age-appropriate experience in their apps while protecting kids' privacy.

Through this new feature, parents can allow their kids to share the age range associated with their Child Accounts with app developers. If they do, developers will be able to utilize a Declared Age Range API to request this information, which can serve as an additional resource to provide age-appropriate content for their users. As with everything we do, the feature will be designed around privacy and users will be in control of their data. The age range will be shared with developers if and only if parents decide to allow this information to be shared, and they can also disable sharing if they change their mind. And it won't provide kids' actual birth dates.

As I've noted before, a *"Declared Age Range API"* is exactly the right solution. Kids use specific iPhones and iPads, and Apple will even have it default in the direction of enforcing safe content. So it makes sense for the device's platform to know the age of its user and for that platform to be able to disclose that information with proper controls. I still think it makes the most sense for parents to set their child's date of birth internally – and as I've noted, they would be free to fudge it either way depending upon their individual child's emotional maturity and the level of protection they feel most comfortable enforcing. Then, if Apple insists upon calculating the users' age within large privacy-protecting ranges, while that seems unnecessarily restrictive – Apple has already needed to amend those dumb ranges because, well, they're dumb – then they could do that, and it does serve to give some additional impression of increased privacy.

In this document, Apple explained their thoughts about all this under the heading: *"Age Assurance: Striking the Right Balance Between Platforms and Developers to Best Serve the Needs of our Users"* where they wrote:

At Apple, we believe in data minimization—collecting and using only the minimum amount of data required to deliver what you need. This is especially important for the issue of "age assurance," which covers a variety of methods that establish a user's age with some level of confidence. Some apps may find it appropriate or even legally required to use age verification, which confirms user age with a high level of certainty—often through collecting a user's sensitive personal information (like a government-issued ID)—to keep kids away from inappropriate content. But most apps don't. That's why the right place to address the dangers of age-restricted content online is the limited set of websites and apps that host that kind of content. After all, we ask merchants who sell alcohol in a mall to verify a buyer's age by checking IDs—we don't ask everyone to turn their date of birth over to the mall if they just want to go to the food court.

Requiring age verification at the app marketplace level is not data minimization. While only a fraction of apps on the App Store may require age verification, all users would have to hand over their sensitive personally identifying information to us—regardless of whether they actually want to use one of these limited set of apps. That means giving us data like a driver's license, passport, or national identification number (such as a Social Security number), even if we don't need it. And because many kids in the U.S. don't have government-issued IDs, parents in the U.S. will have to provide even more sensitive documentation just to allow their child to access apps meant for children. That's not in the interest of user safety or privacy.

Requiring users to overshare their sensitive personal data would also undermine the vibrant online ecosystem that benefits developers and users. Many users might resort to less safe alternatives like the unrestricted web, or simply opt out of the ecosystem entirely, because they can't or won't provide app marketplaces—like the App Store—with sensitive information just to access apps that are appropriate for all ages.

By contrast, the Declared Age Range API is a narrowly tailored, data-minimizing, privacy-protecting tool to assist app developers who can benefit from it, allowing everyone to play their appropriate part in this ecosystem. It gives kids the ability to share their confirmed age range with developers, but only with the approval of their parents. This protects privacy by keeping parents in control of their kids' sensitive personal information, while minimizing the amount of information that is shared with third parties. And the limited subset of developers who actually need to collect a government-issued ID or other additional sensitive personal information from users in order to meet their age-verification obligations can still do so, too. All in all, it gives developers a helpful addition to the set of resources that they can choose from—including other third-party tools—to fulfill their responsibility to deliver age-appropriate experiences in their apps.

With this new feature, parents will be even more firmly in the driver's seat—and developers will have another way to help identify and keep kids safe in their apps.

Apple wrote: *"Requiring age verification at the app marketplace level is not data minimization. While only a fraction of apps on the App Store may require age verification, all users would have to hand over their sensitive personally identifying information to us—regardless of whether they actually want to use one of these limited set of apps."* That's right. Apple is going to need to get over themselves and accept that what they've built is an Internet portal and that they're offering their phones and cases in an array of child-friendly colors. The Internet, being what it is, needs industrial-strength filtering and Apple are the only ones who care enough about privacy and about doing it right to be entrusted with that job. Their platform is where that should happen.

When talking about their age range system, they write:

When a developer submits an app to us for distribution, they confirm the types of sensitive content within the app and how frequently it appears, and if the app has certain features that impact what kind of content will be presented. Apple automatically generates an appropriate age rating for their app indicating the minimum age appropriate to use the app. Developers can also opt-in to choosing the highest rating if they believe it is appropriate for their app. We publish these age ratings on the App Store page for each app, and we reject apps from the App Store if they are misleading or inaccurate. These age ratings are integrated into our operating systems, and work with parental control features like Screen Time and Ask to Buy.

And they just said it right there: *"These age ratings are integrated into our operating systems, and work with parental control features like Screen Time and Ask to Buy."*

The problem is that Internet filtering is a slippery slope. Once any filtering at all exists, who decides what's age appropriate and for what age, and what is not. It's not always cut and dried. Probably around half of the country feels very strongly that young people should not be exposed to controversial issues surrounding their sexuality and gender, whereas the other half feel just as strongly that this information and support should be available to anyone who needs it and that the Internet is the great liberator.

In the U.S. so far, attempts to impose filtering have been blocked on grounds of Constitutional 1st Amendment Freedom of Speech grounds which prohibit the government from imposing itself upon the free flow of information.

Standing back from this a bit, I suspect that the way we're going about this, which I would describe as bit-by-bit, incrementally and cautiously, is probably the right thing. Big sweeping

Internet legislation has a great chance of being done wrong, especially when it largely comes from the political class and may be heavily weighted in the strong ideology of the moment.

All that said, I think that the cautious and gradual evolution of the sort we see from Apple is likely the way things are going to go and while it may not satisfy anyone, that's also probably inevitable.

A North Korean Job Interview

https://blog.knowbe4.com/our-interview-of-a-north-korean-fake-employee

Thanks to a listener of ours I was made aware of one employer's experience with North Koreans faking their identities for the purpose of obtaining employment in the U.S. As we'll see, at one point toward the end of his description, Roger Grimes, whose security industry work we've covered before, says: *"I have now spoken with many dozens of other employers who have either almost hired a North Korean fake employee or hired them. It is not rare."* Here's what Roger himself experienced:

You would think with all the global press we have received because of our public announcement of how we mistakenly hired a North Korean fake employee in July 2024, followed by our multiple public presentations and a whitepaper on the subject, that the North Korean fake employees would avoid applying for jobs at KnowBe4. You would be wrong. It is apparently not in their workflow to look up the company they are trying to fool along with the words 'North Korea fake employees' before they apply for jobs.

We get North Korean fake employees applying for our remote programmer/developer jobs all the time. Sometimes, they are the bulk of the applications we receive. This is not unusual these days. This is the same with many companies and recruiter agencies I talk with. If you are hiring remote-only programmers, pay attention a little bit more than usual.

North Korea has thousands of North Korean employees deployed in a nation-state-level industrial scheme to get North Koreans hired in foreign countries to collect paychecks until they are discovered and fired. Note that due to UN sanctions, it is illegal to knowingly hire a North Korean employee throughout much of the world.

To accomplish this scheme, North Korean citizens apply for remote-only programming jobs offered by companies around the world. The North Koreans apply using all the normal job-seeking sites and tools that a regular applicant would avail, such as the company's own job hiring website and dedicated job sites like Indeed.com.

The North Koreans work as part of larger teams, often consisting of dozens to over a hundred fake applicants. They are usually located in countries outside of North Korea that are friendly to North Koreans, such as China, Russia, and Malaysia. This is because North Korea does not have a good enough infrastructure (Internet, electricity, etc.) to best sustain the program, and it is easy for adversarial countries to detect and block North Korean Internet traffic.

The North Korean fake employees work in teams with a controlling manager. They often live in dormitory-style housing, eat together, and work in very controlled conditions. They do not have much individual freedom. Their families back home are used as hostages to keep the North Korean participants in line and working. They get jobs and earn paychecks, but the bulk of the earnings is sent back to North Korea's government, often to fund sanctioned weapons of mass destruction programs.

The scheme is much like an assembly line workflow. The North Korean fake employee and their helpers apply for the job, interview, supply identity documents, get the job, get the related company equipment, and collect a paycheck. The North Korean applicant may do all steps in this process or farm it off to other participants, depending on the language skills of the applicant and the requirements of the job application process.

They will often use made-up "synthetic" identities, use stolen identity credentials of real people in the targeted country, or actually pay real people of Asian ancestry who live in the target country to participate. It turns out there is a burgeoning sub-industry of college-aged males of Asian ancestry who cannot wait to get paid for participating in these schemes. There are Discord channels all around the world just for this. They make a few hundred to a few thousand dollars for allowing their identity to be misused or participating in the scheme. That way, they can interview in person or take drug tests if the job requires that.

Sometimes the North Korean instigator does all the steps of the application process. Sometimes, they just get the job interview and hand it off to others with better language skills for the interview, and sometimes, they hand off the job to someone who can actually do the job (and collect a kickback percentage). How the North Korean fake employee accomplishes the hiring and job process runs the spectrum of possibilities. We have seen it all.

If they actually win the job, they will have another participant in the targeted country pick up the computing equipment sent by the employer and set it up. They are often known as "laptop farmers." These laptop farmers have rooms full of computing equipment sitting on tables, marked with an identifier of what computer belongs to what company (to keep them straight). They power on the laptops and give the fake North Korean employee remote access to the laptop.

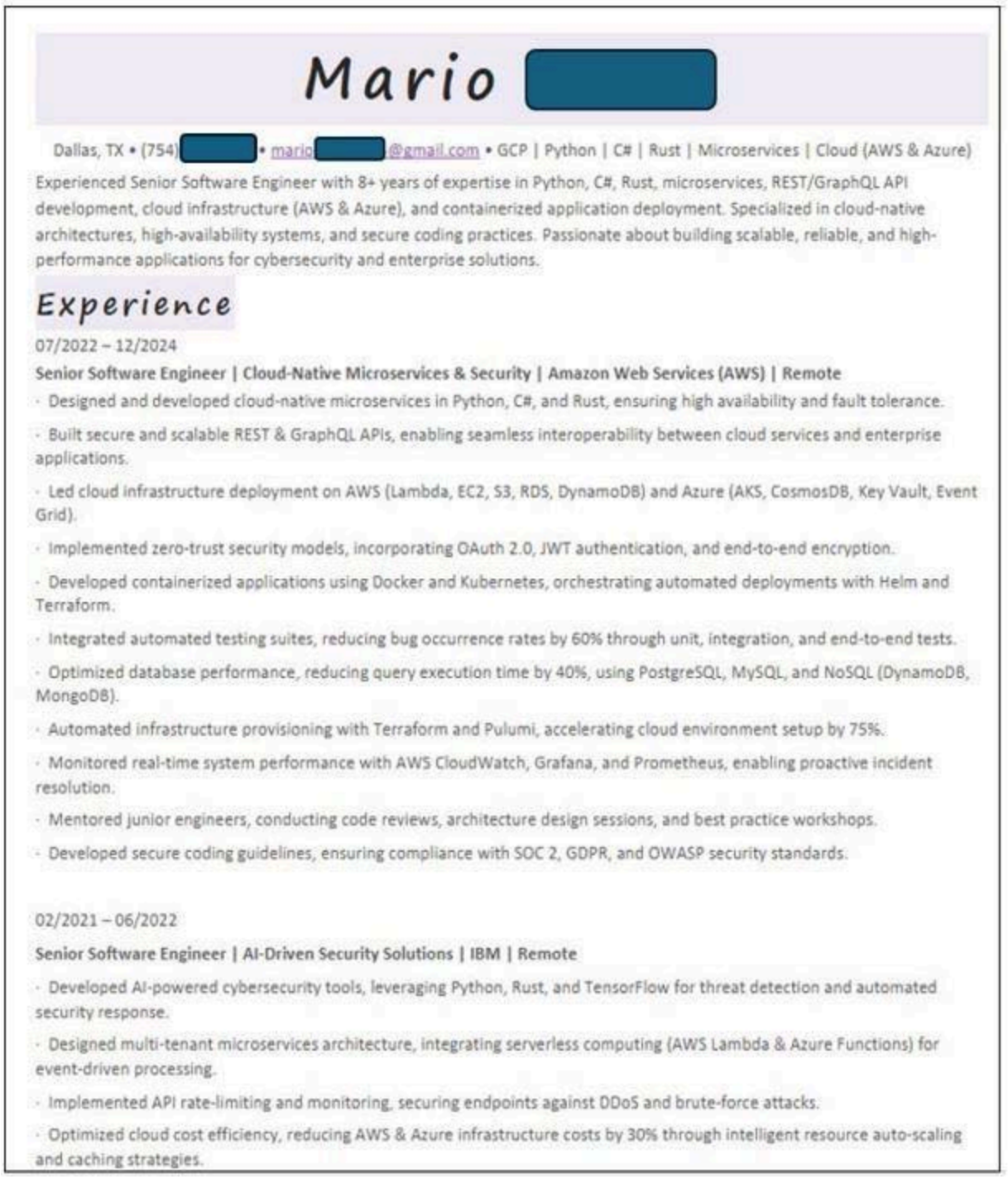
Using this scheme, North Korea has illegally "earned" hundreds of millions of dollars to fund its illegal weapons programs over the last few years.

There have been North Korean fake remote part-time contractors for over a decade, but the fake full-time remote employees took off when COVID-19 created a ton more of fully remote "work-from-home" jobs. There is far more money to be made. If your company offers high-paying, remote-only programmer/developer jobs, you are likely receiving fake job applications from North Koreans. It is rampant. Hundreds to thousands of companies around the world likely have North Korean fake employees working for them right now. It is common.

We regularly get applications from North Korean fake employees. We routinely reject most of them. Occasionally, we accept a few and interview the fake employees to learn more about them and to keep up on any possible developing trends. Luckily, so far, North Korea does not seem to be changing their tactics that much from our original postings.

The signs and symptoms of a North Korean fake employee we described last year still apply today. They are apparently still having great success with them. If you and your hiring team are educated about these schemes, it is fairly easy to recognize and mitigate them. You just have to know and look for the signs and symptoms.

We recently interviewed "Mario" supposedly from Dallas, Texas. Here's a part of his resume.



Mario [redacted]

Dallas, TX • (754)[redacted] • mario[redacted]@gmail.com • GCP | Python | C# | Rust | Microservices | Cloud (AWS & Azure)

Experienced Senior Software Engineer with 8+ years of expertise in Python, C#, Rust, microservices, REST/GraphQL API development, cloud infrastructure (AWS & Azure), and containerized application deployment. Specialized in cloud-native architectures, high-availability systems, and secure coding practices. Passionate about building scalable, reliable, and high-performance applications for cybersecurity and enterprise solutions.

Experience

07/2022 – 12/2024

Senior Software Engineer | Cloud-Native Microservices & Security | Amazon Web Services (AWS) | Remote

- Designed and developed cloud-native microservices in Python, C#, and Rust, ensuring high availability and fault tolerance.
- Built secure and scalable REST & GraphQL APIs, enabling seamless interoperability between cloud services and enterprise applications.
- Led cloud infrastructure deployment on AWS (Lambda, EC2, S3, RDS, DynamoDB) and Azure (AKS, CosmosDB, Key Vault, Event Grid).
- Implemented zero-trust security models, incorporating OAuth 2.0, JWT authentication, and end-to-end encryption.
- Developed containerized applications using Docker and Kubernetes, orchestrating automated deployments with Helm and Terraform.
- Integrated automated testing suites, reducing bug occurrence rates by 60% through unit, integration, and end-to-end tests.
- Optimized database performance, reducing query execution time by 40%, using PostgreSQL, MySQL, and NoSQL (DynamoDB, MongoDB).
- Automated infrastructure provisioning with Terraform and Pulumi, accelerating cloud environment setup by 75%.
- Monitored real-time system performance with AWS CloudWatch, Grafana, and Prometheus, enabling proactive incident resolution.
- Mentored junior engineers, conducting code reviews, architecture design sessions, and best practice workshops.
- Developed secure coding guidelines, ensuring compliance with SOC 2, GDPR, and OWASP security standards.

02/2021 – 06/2022

Senior Software Engineer | AI-Driven Security Solutions | IBM | Remote

- Developed AI-powered cybersecurity tools, leveraging Python, Rust, and TensorFlow for threat detection and automated security response.
- Designed multi-tenant microservices architecture, integrating serverless computing (AWS Lambda & Azure Functions) for event-driven processing.
- Implemented API rate-limiting and monitoring, securing endpoints against DDoS and brute-force attacks.
- Optimized cloud cost efficiency, reducing AWS & Azure infrastructure costs by 30% through intelligent resource auto-scaling and caching strategies.

We have hidden Mario's last name and contact information because it is the name of a real American who is likely unaware that his identity has been hijacked and used in this scheme and we don't want hiring companies to accidentally be given the rogue contact info and think they have a real employee candidate.

"Mario" claimed that he was an American citizen who was born and raised in Dallas. Despite this, he had a fairly strong Asian accent (likely North Korean). The Mario who showed up for our Zoom interview had the same voice as the Mario we interviewed over the phone during the first stage of the application process. Sometimes, they are different.

We had three KnowBe4 people on the Zoom call, including myself.

Over the next 45 minutes, we asked all sorts of questions that would be asked of any developer candidate. Whenever we asked a question, Mario would hesitate, spend 5-15 seconds repeating our question, and then come back with the perfect answer…most of the time. It was clear that Mario or someone participating with him was typing the question subject into a Google search or AI engine and repeating the results.

Mario started off by saying how he had a special interest in social engineering (no kidding!) and security culture. He mentioned "security culture" over and over. I soon realized that if you go to our main website, we say "security culture" all over the place. He was repeating phrases he found on our website.

But he was very friendly and smiling, and his English was heavily accented, but not super hard to understand most of the time. I would say that based solely on this first part of the interview, if we were unaware of what was going on, we all would have liked what he said and how he responded. He was friendly and smiley, and we liked him.

Mario claimed on his resume and in person to have programmed for Amazon, Salesforce, and IBM. He supposedly has the exact advanced programming skills we had advertised. I wish all job applicants knew as well how to best match what we advertised in a job ad with what they responded with. During his initial statements, he said he had a personal interest in cryptography and security. When it came time for me to ask technical questions, I used his mentioned interests as the basis for my questions.

I started off by asking if he had ever done post-quantum cryptography and if he had implemented it in his past projects. He hesitated, repeated the question, and then gave me an excellent dissertation on post-quantum cryptography, including mentioning NIST (which is probably the top search result you will get when researching post-quantum cryptography) and a list of the various post-quantum cryptography standards.

*I asked him if his previous projects were all using post-quantum cryptography. He said, "Yes", which is absolutely untrue. Almost **no** American company is currently implementing post-quantum cryptography. Strike one.*

I asked what post-quantum encryption standard he liked to use most. He said Crystals-Dilithium. It is a digital signature algorithm, not encryption. He frequently mixed up encryption algorithms, like AES, with hashes (e.g., SHA-2) and digital signatures (e.g., Diffie-Hellman). Strike two for someone who is really into cryptography and regularly uses post-quantum cryptography.

I asked what size an AES cipher key would need to be to be considered post-quantum. This seemed to throw him for a loop, and he wasted more time than usual. He replied, 128-bits. This is wrong. AES keys have to be 256-bits or longer to be considered resilient against quantum cryptography. Strike three on the technical questions. He wrongly answered every technical question I asked.

At this point, I decided to throw out a random bad fact that any normal U.S. candidate should be able to spot and correct. I said, "Bill Gates, CEO of Microsoft, says that all future programming will be done by AI agents. What do you think?"

Bill Gates has not been the CEO of Microsoft since 2008, but most people outside the industry would likely think Bill Gates was still the CEO because that is how the media often references him…as the "former" CEO of Microsoft. He is still a cultural icon associated with Microsoft. This is the type of mistake that a North Korean employee who does not have great access to the Internet would make.

And sure enough, Mario repeated the fact that Bill Gates was the CEO of Microsoft (instead of the current CEO, Satya Nadella). Mario did give a great answer on agentic AI and programming using AI agents. If he were a real employee, I would give his answer top points…well, except for not noticing my CEO switch-a-roo.

Finally, with the technical part of the interview over, we switched to the "personal" questions. If you are concerned that you may have a North Korean fake employee candidate on your hands, it cannot hurt to think of and ask for cultural references that anyone in your country or region should readily know, but that would be harder for a foreigner with limited knowledge of the culture to understand.

One of my co-interviewers asked him what he did in his free time. This seemed to surprise him. My co-worker asked if he liked any sports. He said he loved badminton, which he probably did not realize that although super popular in Asian cultures, it is not a top sport if you grew up in Dallas, TX, or nearly anywhere in America. Sure, there are plenty of people who play badminton (especially Americans of Asian-American ancestry), but it is an unlikely response out of all the possible responses you could offer.

I asked how excited he was that the Cowboys won the AFC. I figured he would not know that the Dallas Cowboys got creamed and did not win the AFC. For one, they are in the NFC and not the AFC conference division. He again hesitated…but then seemed to get that I was mentioning the Dallas Cowboys and that they had been eliminated from contention. I was surprised that this did not trip him up as much as I thought it would.

My co-worker said he was going to visit Dallas soon and did the candidate have any favorite food spots. Mario said his mother's cooking. I thought that was a great response so he did not have to look up any restaurants in Dallas.

My co-worker persisted asking the candidate if they had any restaurants to recommend. Mario did not. I offered up the "book repository" (one of the most famous tourist sites in Dallas) where people are dying to eat the "Nashville hot chicken." Mario wholeheartedly agreed with my recommendation.

My co-worker asked the candidate if there was anywhere he would want to travel. In our hidden Slack channel, my co-worker said that when he asked this question of North Korean candidates, their eyes always lit up and they got excited. Sure enough, Mario began to excitedly describe his dreams of visiting Paris and South Africa.

I think it was at this point that we all began to have some empathy. Yes, we were dealing with a fake job candidate who was trying to steal our money (or worse), but in reality, this was a young man likely forced to do what he was doing, destined never to receive any big salary or visit those dreamed of vacation destinations. It is strange, but I think we started to feel a little ashamed at conducting a fake interview. So, we stopped and asked if he had any questions. The normal job candidate would likely ask more about the job, tools used, benefits, and things like that. Mario had no questions other than how many other people we were interviewing and how he was doing in the job interview.

We ended the job interview. We had not picked up any new tactics or information, other than noticing that a lot of the North Korean fake employee candidates lately had been claiming to have been born and raised in Dallas, TX, and all with heavy accents. However, the last fake employee interview switched from a heavy Asian accent from the initial phone interview to a savvy Pakistani person whom we interviewed on Zoom (he must have been the hired handoff for the interview).

I have now spoken with many dozens of other employers who have either almost hired a North Korean fake employee or hired them. It is not rare. And sometimes the fake employees, when discovered, switch to a ransomware encryption scheme or steal your company's confidential data and ask for a ransom, so it is not always just about getting the paycheck.

Employers beware.

I have several long time friends, whom I've known for decades, who pre-COVID, showed up to work every morning in person to work alongside others in an office building. These people are now roaming freely with a laptop in search of connectivity and able to work remotely. In some cases there's a requirement for them to show up in person and spend some time chatting over the water cooler. But I have other friends who have been telecommuting exclusively for years.

And I have a very good techie friend, someone who Leo also knows, who has employed overseas coders in Ukraine for many years with very good results. He's explained that he cannot begin to find people in the U.S. who are willing to work so hard and produce such high quality code. So the *"never even met 'em"* model of telecommuting is also alive and well. I should note that Mark has actually met his coding group many times and during one visit they went to Chernobyl.

In any event, I wanted to be sure that the employers and interviewers among our listeners were fully aware and appreciated the degree to which these fake North Korean employee farm scams are real. I have a link on page 12 of the show notes to Roger Grime's far more detailed 21-page report which also heavily links to many other resources:

https://www.knowbe4.com/hubfs/North-Korean-Fake-Employees-Are-Everywhere-WP_EN-us.pdf

From the standpoint of network security, the idea of having a potentially hostile employee infiltrate an organization and its network is quite chilling. They may be willing to do good work until they are discovered, but it should be clear that no true loyalty is ever going to be earned and that at the moment they feel that they may be discovered they're very likely to trigger a significant attack. It's certainly worth being on the lookout.

Bybit Update

Before I share the latest news on the movement of $1.5 billion US dollars worth of stolen Ethereum tokens I should note that the 10 percent bounty on that $1.5 billion dollars is not $150,000 dollars as I apparently mistakenly said last week. Several listeners politely wrote to say, *"Uh, Steve, that would be $150* ***million*** *dollars in bounty, not $150* ***thousand****."* Indeed. And I'm happy to share that correction.

Okay. So what do we know today? Crypto.News reports under their headline *"Nearly 20% of Bybit's $1.46 Billion in stolen funds 'gone dark,' says Bybit CEO."*

Bybit's CEO Ben Zhou (cho) says nearly 20% of the funds are now untraceable, less than two

weeks after the exchange lost over $1.4 billion in a highly sophisticated attack by North Korea-backed hackers. In a March 4th post on 'X', Zhou (cho) shared an update on the ongoing investigation into the cyberattack, revealing that around 77% of the stolen funds remain traceable, but that nearly 20% has "gone dark" through mixing services.

The hacker primarily used THORChain, a cross-chain liquidity protocol which came under scrutiny for unwillingness to prevent DPRK hackers from laundering the funds, to convert stolen Ethereum into Bitcoin. Approximately 83% of the funds, or around $1 billion, were swapped into BTC across nearly 7 thousand (6,954) individual wallets.

As crypto.news reported earlier, while other protocols took steps to prevent the movement of stolen funds, THORChain validators failed to take meaningful action. Pluto, a core contributor, resigned in protest after nodes rejected a governance proposal to halt ETH transactions.

Of the stolen funds, 72% ($900 million) passed through THORChain, which remains traceable, says Zhou (cho). However, around 16% of the funds, totaling just shy of 80K ETH (79,655 ETH, valued at around $160 million), have gone dark through ExCH, a centralized crypto mixing service.

Zhou (cho) mentioned that the exchange is still waiting for an update on these transactions. Another portion of the funds (~$65 million) also remains untraceable as Zhou says more information is needed from OKX's Web3 wallet. In addition, the Bybit CEO revealed that 11 parties, including Mantle, ParaSwap, and blockchain sleuth ZachXBT, have helped freeze some of the funds, resulting in over $2.1 million in bounty payouts.

So, Bybit is recovering some of their stolen money in return for 10% bounty payouts which those returning the stolen funds are earning legally.

Safe{Wallet} Update

Meanwhile, what of that Safe{Wallet} service whose malicious infiltration was the proximate cause of this very expensive breach? Crypto.News also reports under their headline "Safe Wallet responds to Bybit hack with major security improvements" writing:

Ethereum-based crypto wallet protocol Safe implemented "immediate security improvements" to its multi-sig solution following a cyberattack on Dubai-based exchange Bybit on Feb. 21.

North Korea's Lazarus stole over $1.4 billion in Ether (ETH) from Bybit's Ethereum wallet by exploiting vulnerabilities in Safe Wallet's UI. The infamous hacking group injected hostile JavaScript code specifically targeting Bybit, siphoning more than 400,000 ETH. To prevent further attacks, Safe placed its Wallet in lockdown mode before announcing a phased rollout and a reconfigured infrastructure.

Martin Koeppelmann, co-founder of Safe, said in a March 3rd X.com post that their team had developed and shipped ten changes to the UI. The protocol's GitHub repositories showed updates to "show full raw transaction data now on UI" and "remove specific direct hardware wallet support that raised security concerns", among other upgrades.

Bybit CEO Ben Zhou (cho) discussed the incident on the When Shift Happens podcast with host Kevin Follonier, explaining that the attack occurred shortly after he signed a transaction to transfer 13,000 ETH.

Zhou (cho) mentioned using a Ledger hardware wallet but noted that he couldn't fully verify the transaction details. The issue, known as "blind signing", is a common vulnerability in multi-sig crypto transactions. Safe's latest updates aim to provide signers with more detailed transaction data, according to Koeppelmann.

In response to a post from Kyber Network CEO Victor Tran regarding industry-wide security efforts, Koeppelmann emphasized the importance of collaboration but noted that immediate damage control remains the priority, writing: "We're still in the "putting out fire" mode – but once we have that behind us we need to come together and improve overall frontend and transaction verification security," Koeppelmann stated, adding that "This will take involvement of many parties to solve it for good."

So it does sound as though some good will eventually come of this, though it certainly was expensive. There is just so much liquidity sloshing around in this crypto world. It still boggles my mind.

Apple appeals the UK investigatory demand

Meanwhile, back on the encryption front, last week the BBC, reported under the headline *"Apple takes legal action in UK data privacy row"* – this would be, of course, in response to a legal demand whose very existence Apple is prohibited from divulging. But it seems, **that** particular cat is well out of the bag. So, the BBC wrote:

Apple is taking legal action to try to overturn a demand made by the UK government to view its customers' private data if required.

The BBC understands that the US technology giant has appealed to the Investigatory Powers Tribunal, an independent court with the power to investigate claims against the Security Service. It is the latest development in an unprecedented row between one of the world's biggest tech firms and the UK government over data privacy.

In January, Apple was issued a secret order by the Home Office to share encrypted data belonging to Apple users around the world with UK law enforcement in the event of a potential national security threat. Data protected by Apple's standard level of encryption is still accessible by the company if a warrant is issued, but the firm cannot view or share data encrypted using its toughest privacy tool, Advanced Data Protection (ADP).

Last week, Apple chose to remove ADP from the UK market rather than comply with the notice, which would involve creating a "backdoor" in the tool to create access. Apple said at the time that it would never compromise its security features and it was disappointed at having to take the action in the UK. The UK's order also angered the US administration with President Donald Trump describing it to The Spectator as "something that you hear about with China".

Tulsi Gabbard, US head of intelligence, said she had not been informed in advance about the UK's demand. She wrote in a letter that it was an "egregious violation" of US citizens' rights to privacy and added that she intended to determine whether it breached the terms of a legal data agreement between the US and the UK.

The Financial Times, which first revealed Apple's legal action, reports that the tribunal case

could be heard in the next few weeks, but may not be made public. The Home Office refused to confirm or deny that the notice issued in January exists. Legally, this order cannot be made public. But a spokesperson said: "More broadly, the UK has a longstanding position of protecting our citizens from the very worst crimes, such as child sex abuse and terrorism, at the same time as protecting people's privacy. The UK has robust safeguards and independent oversight to protect privacy and privacy is only impacted on an exceptional basis, in relation to the most serious crimes and only when it is necessary and proportionate to do so."

Meanwhile, Apple declined to comment.

Being a glass half full sort, I'm still holding out hope that Apple's initial move will have shaken up the UK's legislators sufficiently for them to allow Apple's appeal to succeed and for Apple's very public shot across the bow threat to pull their strongest encryption from the UK will be sufficient to put this troublesome issue back to bed for a while.

The unresolved question is this: Given that we now have the technology to create and enforce absolute privacy of communications and data storage, in a modern democracy, which is designed to be by the people and for the people with elected representation in government, do the benefits of this absolute privacy obtained by the overwhelming law abiding majority outweigh the costs and risks to society created by its abuse by a small criminal minority?

The trouble is that individual governments may decide these issues differently, yet the Internet is global and has always promised to be unifying. When we stand back to look at these issues surrounding privacy through encryption and the challenges presented by the biological ages of Internet users and the perceived need to filter their access to this global network, what becomes clear is that up to this point these fundamental issues and concerns, created by cyberspace having very different rules from physical space, have largely been ignored. It feels as if this has all happened so quickly that society has been catching its breath while waiting for the dust to settle. While waiting for services to be developed and mature. While waiting for those who govern us to catch up. It appears that our societies are finally gearing up to deal with these issues. We've had a really interesting 50 years. What will the next 50 look like?

CVE-2024-9956 - PassKey Account Takeover in All Mobile Browsers

If you encounter reports claiming that there's a flaw in Passkeys affecting mobile browsers, the truth is somewhat more nuanced. It wasn't a flaw in Passkeys, it was a very specific and difficult to perpetrate account takeover flaw that was only possible due to URL link navigation mistakes made in mobile Chrome & Edge which was fixed in October of 2024, mobile Safari fixed it in January of this year and Firefox patched the problem last month in February.

At one point in the Passkeys FIDO flow, mobile browsers are given a link with the scheme FIDO:// which they were allowed to navigate. Once the three browsers blocked the FIDO:// scheme from being navigable, the small loophole a researcher discovered was closed, and Passkeys returns to being the extremely robust network authentication solution that the world needs it to be.

The UK appears to be going off the rails.

First they order Apple to accomplish the impossible by decrypting data for which the UK knows Apple does not have the keys. Then I read that a court in the U.K. had demanded that a U.S. based security researcher remove their reporting of an embarrassing cyberattack and data breach at HCRG, formerly known as Virgin Care, which is one of the largest independent healthcare providers in the U.K.

This made me curious, so I first found a nice summary of the situation a TechCrunch, which wrote:

> *A U.S.-based independent cybersecurity journalist has declined to comply with a U.K. court-ordered injunction that was sought following their reporting on a recent cyberattack at U.K. private healthcare giant HCRG.*
>
> *Law firm Pinsent Masons, which served the February 28 court order on behalf of HCRG, demanded that DataBreaches.net "take down" two articles that referenced the ransomware attack on HCRG. The law firm's notice to DataBreaches.net, which TechCrunch has seen, stated that the accompanying injunction was "obtained by HCRG" at the High Court of Justice in London to "prevent the publication or disclosure of confidential data stolen during a recent ransomware cyberattack." The firm's letter states that if DataBreaches.net disobeys the injunction, the site may be found in contempt of court, which "may result in imprisonment, a criminal fine or having your assets seized."*
>
> *DataBreaches.net, run by a journalist who operates under the pseudonym Dissent Doe, declined to remove the posts, and also published details of the injunction in a blog post Wednesday. Dissent, citing a letter from their law firm Covington & Burling, said they would not comply with the order on grounds that DataBreaches.net is not subject to the jurisdiction of the U.K. injunction and that the reporting is lawful under the First Amendment in the United States, where DataBreaches.net is based. Dissent also noted that the text of the court order does not specifically name DataBreaches.net nor reference the specific articles in question.*
>
> *Legal threats and demands are not uncommon in cybersecurity journalism, since the reporting often involves uncovering information that companies do not want to be made public. But injunctions and legal demands are seldom published over risks or fears of legal repercussions. The details of the injunction offer a rare insight into how U.K. law can be used to issue legal demands to remove published stories that are critical or embarrassing to companies.*
>
> *The law firm's letter also confirms that HCRG was hit by a "ransomware cyber-attack."*

That made me interested enough to go to the source where I discovered some additional head shaking detail which picks up where TechCrunch left off. Remember that the site is being represented by Covington & Burling and in the UK we have the firm Pinsent Masons. Dissent Doe, wrote:

> *When Jason Criss of Covington and Burling sent an email to Pinsent Masons informing them that DataBreaches.net is a US entity with no connection to the UK, and that neither the UK nor the High Court of Justice has any jurisdiction over this site, that should have been the end of the matter, right? But it wasn't, and that's partly why DataBreaches is reporting on this.*
>
> *Yesterday morning, DataBreaches.net received an email from its domain registrar that it had been served with the injunction by Pinsent Masons, and that if DataBreaches did not remove the two posts in question within 24 hours, this website site would be suspended.*
>
> *The two posts were not even particularly exciting. They mainly summarized some of SuspectFile's great reporting and linked to those posts. For those who would like to see what HCRG or the court demanded I remove, the posts can be seen at:*

UK: More details emerge about ransomware attack on HCRG by Medusa

Medusa Unveils Another 50TB of Stolen Data from HCRG Care Group, Giving Greater Insight Into the Scope of the Breach

DataBreaches informed the registrar that the injunction was not valid and that DataBreaches.net is not under the jurisdiction of the High Court of Justice or of the United Kingdom. Jason Criss of Covington and Burling also notified the registrar that not only was DataBreaches.net a US entity, but as the site's domain registrar for many years, they could see for themselves that the site was registered to a US person at a US postal address with a US telephone number.

Later yesterday, the registrar responded:

> Since your lawyer has already sent notice to the complainant, Pinsent and Masons ,we confirm that we will not be taking any action on your domain, databreaches.net. Additionally, we will be informing Pinsent and Masons to contact your lawyer directly should they have any further issues. This ticket will now be closed.

Pinsent Masons did not respond to Monday's email notification by Jason Criss that this site was not under UK or High Court jurisdiction. And at no time yesterday did Pinsent Masons contact the domain registrar to say that it was withdrawing the demand for the removal of the posts. That, too, was surprising.

Is it over? Or will there be more? DataBreaches hopes it is over.

The major UK firm Pinsent & Masons must be fully aware of the 1st amendment free speech protections enjoyed in the U.S., and they certainly knew that DataBreaches.net was a U.S. based website registered in the U.S. So this had to be pure baseless intimidation. Some stuffed shirt at the UK healthcare provider was annoyed by the fact that the embarrassingly massive 50 terabyte data breach of their systems was being reported, and decided to aim their law firm at the reporter.

LASTPASS: The breach that keeps on giving

Get a load of this one. You'll hear a very familiar name pop out of this little piece of news, which reads: The FBI has recovered $23 million worth of crypto stolen from Chris Larsen, the co-founder and executive chairman of the Ripple (XRP) cryptocurrency. The recovered funds are just a small part of the tokens stolen from Larsen in January of last year. The funds were estimated at over $110 million last year but are now worth over $700 million. [And here it comes...]

Hackers stole the Larsen funds by first stealing password stores from password manager LastPass in 2022. Since the attack, the hackers have been slowly cracking passwords and emptying crypto wallets. As of May 2024, over $250 million worth of crypto assets had been stolen using the data obtained from LastPass.

Remember, at the time we talked about this: Bad guys largely don't care about random people's laundry at all. They want one thing, which is money. So they were known to be targeting any crypto passwords suspected of being stored in LastPass vaults. With LastPass' failure to increase

the repetition counts of their PBKDF system, accounts which had been created in the early days of LastPass were left with very low or even zero iterations of their hashing algorithm. This made cracking the passwords protecting those early adopters extra easy. Our advice at the time, for anyone who had stored crypto access passwords in LastPass was to immediately create a new wallet and transfer the assets from the unsafe wallet into the newly created wallet.

More Bybit hack post-mortem news

We're still learning more about the early genesis of the attack. The North Korean hackers compromised the multi-signature wallet provider SafeWallet through a social engineering attack which targeted one of its developers. According to a new post-mortem report, the point of entry appears to have been a malicious Docker file that was executed on the employee's computer. The Docker file deployed malware that stole local credentials. The attackers then used the developer's AWS account to add malicious code to the SafeWallet infrastructure which targeted a specific multi-sig wallet used by the Bybit cryptocurrency exchange.

A Ransomware attack using IoT

I wanted to share this terrific look at how a Windows-centric network was hit by ransomware, even though they had strong and effective malware protections in place. A security research group has been tracking the Akira ransomware group. So what they found as they dug into a forensic reverse engineering of a distressingly successful attack was interesting and surprising to them. They wrote:

Until the compromise, this incident had followed Akira's typical modus operandi. After compromising the victim's network via an externally facing remote access solution, the group deployed AnyDesk.exe, a remote management and monitoring tool, to retain access to the network, before exfiltrating data.

During the latter stages of the attack, the attacker moved to a server on the victim's network via remote desktop protocol (RDP). Akira commonly uses RDP as it enables them to interact with endpoints and blend in with system administrators, who use RDP legitimately. The threat actor initially attempted to deploy the ransomware on one of the Windows servers as a password-protected zip file ('win.zip') that contained the ransomware binary ('win.exe'). However, the victim's endpoint detection & response (EDR) tool ***immediately*** *identified and quarantined the compressed file before it was unzipped and deployed.*

At this point, the threat actor likely realised they had alerted the EDR tool and would not be able to evade its defences. They therefore pivoted their approach. Prior to the ransomware deployment attempt to this Windows server, the attacker had conducted an internal network scan to identify open ports, services, and devices. This network scan identified several Internet of Things ('IoT') devices on the victim's network, including webcams and a fingerprint scanner.

These devices presented an opportunity to the threat actor to evade the EDR tool and deploy the ransomware successfully. The threat actor likely identified a webcam as a suitable target device for deploying ransomware for three reasons:

- *The webcam had several critical vulnerabilities, including remote shell capabilities and unauthorised remote viewing of the camera.*
- *It was running a lightweight Linux operating system that supported command execution as if it were a standard Linux device, making the device a perfect candidate for Akira's Linux*

ransomware variant.

- *The device did not have any EDR tools installed on it, leaving it unprotected. In fact, due to the limited storage capacity, it is doubtful that any EDR could be installed at all.*

After identifying the webcam as a suitable target, the threat actor began deploying their Linux-based ransomware with little delay. As the device was not being monitored, the victim organisation's security team were unaware of the increase in malicious Server Message Block (SMB) traffic from the webcam to the impacted server and the webcam successfully fully encrypted the servers on the victim's network. Akira was thus able to encrypt files across the victim's network.

This was an interesting case. Here, the vulnerable IoT device was not the initial point of entry. That honor belonged to some unspecified remote access solution running on a Windows machine.

But even though the IoT device wasn't their way in, the attackers needed an unprotected host for their malware. They were unable to run their ransomware on any of the Windows systems or servers on the network because all of those systems were being protected by effective real-time EDR – endpoint detection & response – security.

But their network scan had discovered some Linux-based webcams, and that's all they needed. And the security of those 'cams was also quite lacking – which made their jobs even easier. So they loaded their malware into the 'cams RAM and it reached out over the network, using Windows file and printer sharing SMB (server message blocks) protocol to read and write back encrypted file contents.

Under the prevention & remediation section of their report they wrote:

Preventing and remediating novel attacks like this one can be challenging. At a minimum, organisations should monitor network traffic from their IoT devices and detect anomalies. They should also consider adopting the following security practices:

And what do you think their #1 first recommendation is? They wrote:

Network restriction or segmentation: Place IoT devices on a segmented network that cannot be accessed from servers or user workstations or restrict the devices' communication with specific ports and IP addresses.

It takes more work and it can limit functionality. And it means that you cannot just randomly plug anything in anywhere you like. So some ongoing network management discipline will be needed too. But this company learned that lesson the hard way.

The Bluetooth Backdoor

I deliberately titled today's podcast *"The Bluetooth Backdoor"* because that's what nearly all of the tech press has been calling it, and in this instance it does feel like the appropriate use of that loaded term. Last Saturday, BleepingComputer's headline was *"Undocumented backdoor found in Bluetooth chip used by a billion devices"* then the next day, last Sunday, they softened that headline to *"Undocumented commands found in Bluetooth chip used by a billion devices"* and to explain that change they wrote: *"After receiving concerns about the use of the term 'backdoor' to refer to these undocumented commands, we have updated our title and story. Our original story can be found here"* where "here" was an Internet Archive link to the article from the previous day.

This podcast has spent some time batting this issue of "when is a backdoor not a backdoor?" back and forth. Would forcing Apple to deliberately and publicly redesign their Advanced Data Protection iCloud synchronization and backup to incorporate a Master Key, be a "backdoor"? In this instance I would conclude "no" because this feature of ADP would be neither secret nor malicious, whereas the classic definition and use of the term "backdoor" is both, and it definitely needs to be a secret. If it's not a secret it cannot be a backdoor. That leaves us with the question of malice. In Apple's case there's clearly no malice anywhere. Thus the term "backdoor" fails to qualify for what Apple has apparently been asked for by the UK on both counts.

So what about today's news of what nearly everyone – including me – is calling a "backdoor"? We know for sure that what a pair of Spanish security researchers discovered lurking in an astonishingly widely used Chinese microcontroller chip was at least undocumented and also incredibly powerful and thus prone to abuse if it were to become known by any malicious party. But is it, itself, malicious? It's just a bunch of undocumented commands – albeit quite powerful. This is an instance where we're left needing to infer the intent behind something. And the intent is ultimately unknowable. But it's certainly possible to pose the question: "Why were these commands deliberately left undocumented? Why?

Before we back up a bit and look at what we know, just so that I don't later forget to mention this, everyone who has taken the time and trouble to physically segment their home or office networks to deliberately isolate as much ans as many of their IoT devices as possible, can use this instance as an example of exactly what I've been talking about from the start. There is zero doubt that the IoT devices we're using, smart plugs and switches, intelligent thermostats, hubs, security cameras, residential alarm systems, room sweeping robots, and everything else, are using these Chinese ESP32 chips. Regardless of whether or not they turn out to contain any sort of "backdoor", there's no way to know they don't.

For this reason it has always made sense to place our PCs and NAS devices, which contain data we would like to keep private and secure, on their own physical network, separate from the mass of $10 gizmos and gadgets that also want and need to have access to a network and to the Internet. We just don't want them to have access to our privileged network. There's no reason to suspect that they're evil, it's just not possible to know that they're not. Consider all of the networks whose border NAT routers we know have already been compromised and contain remotely-controlled botnets. That's not fiction or imagination, that's real. Since we can't know what we don't know, the exercise of a bit of caution makes sense.

Okay. So what happened here? We can trust BleepingComputer to be level headed, here's what they wrote in their updated and toned-down coverage:

The ubiquitous ESP32 microchip made by Chinese manufacturer Espressif and used by over 1 billion units as of 2023 contains undocumented commands that could be leveraged for attacks.

The undocumented commands allow spoofing of trusted devices, unauthorized data access, pivoting to other devices on the network, and potentially establishing long-term persistence.

This was discovered by two Spanish researchers with the security firm Tarlogic who presented their findings at RootedCON in Madrid. A Tarlogic announcement shared with BleepingComputer reads: "Tarlogic Security has detected a backdoor in the ESP32, a microcontroller that enables WiFi and Bluetooth connection and is present in millions of mass-market IoT devices. Exploitation of this backdoor would allow hostile actors to conduct impersonation attacks and permanently infect sensitive devices such as mobile phones, computers, smart locks or medical equipment by bypassing code audit controls."

The researchers warned that ESP32 is one of the world's most widely used chips for Wi-Fi + Bluetooth connectivity in Internet of Things (IoT) devices, so the risk is significant. In their RootedCON presentation, the Tarlogic researchers explained that interest in Bluetooth security research has waned but not because the protocol or its implementation has become more secure. Instead, most attacks presented last year did not have working tools, did not work with generic hardware, and used outdated or unmaintained tools largely incompatible with modern systems.

Tarlogic first developed a new C-based USB Bluetooth driver that is hardware-independent and cross-platform. This provided direct access to the hardware without relying on OS-specific APIs. Armed with this new tool, which enables raw access to Bluetooth traffic, Tarlogic discovered hidden vendor-specific commands (Opcode 3F) in the ESP32 Bluetooth firmware that allow low-level control over Bluetooth functions.

In total, they found 29 undocumented commands, collectively characterized as a "backdoor," that could be used for memory manipulation (to read or write RAM and Flash), MAC address spoofing (for device impersonation), and packet injection.

Espressif has not publicly documented these commands, so either they were not meant to be accessible, or they were left in by mistake. The issue is now tracked under CVE-2025-27840.

The risks enabled by these commands include malicious implementations on the OEM level and supply chain attacks. Depending on how Bluetooth stacks handle HCI commands on the device, remote exploitation of the commands might be possible via malicious firmware or rogue Bluetooth connections. This is especially the case if an attacker already has root access, planted malware, or pushed a malicious update on the device that opens up low-level access. In general, though, physical access to the device's USB or UART interface would be far riskier and a more realistic attack scenario.

The researchers explained: "In a context where you can compromise an IOT device with an ESP32 you will be able to conceal an Advanced Persistent Threat inside the ESP memory and perform Bluetooth (or Wi-Fi) attacks against other devices, while controlling the device over Wi-Fi or Bluetooth. Our findings would allow the full takeover of ESP32 chips and the gaining of persistence in the chip via commands that allow for RAM and Flash modification. Also, with persistence in the chip, it may be possible to spread to other devices because the ESP32 allows for the execution of advanced Bluetooth attacks."

BleepingComputer said that they had contacted Espressif for a statement on the researchers' findings, but that they had not received any comment.

NIST now has this listed, as BleepingComputer noted, as CVE-2025-27840 with the description: *"Espressif ESP32 chips allow 29 hidden HCI commands, such as 0xFC02 (Write memory)."*

So, next, we need to look at what the researchers have explained about their own technology. I've edited it down somewhat to remove the marketing-speak and redundancy:

The ESP32, a 2 Euro chip, can open the door to identity theft to connect to thousands of IoT devices. Tarlogic Security has detected hidden functionality that can be used as a backdoor in the ESP32, a microcontroller that enables WiFi and Bluetooth connection and is present in millions of mass-market IoT devices. Exploitation of this hidden functionality would allow hostile actors to conduct impersonation attacks and permanently infect sensitive devices such as mobile phones, computers, smart locks or medical equipment by bypassing code audit controls.

This discovery is part of the ongoing research carried out by the Innovation Department of Tarlogic on the Bluetooth standard. Thus, the company has also presented at RootedCON, the world's largest Spanish-language cybersecurity conference, BluetoothUSB, a free tool that enables the development of tests for Bluetooth security audits regardless of the operating system of the devices.

This solution seeks to democratize the execution of security tests for Bluetooth devices [I assume they mean that this makes such tests available to everyone] and help manufacturers and cybersecurity experts protect all kinds of gadgets and technological equipment from attacks that aim to spy on citizens and companies and take control of devices that are essential in our daily lives.

Researchers from the cybersecurity company have reviewed multiple Bluetooth devices using the BSAM methodology, presented by Tarlogic a year ago, which systematizes the performance of Bluetooth security audits.

In the course of the investigation, a hidden feature was discovered in the ESP32 chip, used in millions of IoT devices and which can be purchased on the world's most famous e-commerce sites for 2 euros. It is this low cost that explains why it is present in the vast majority of Bluetooth IoT devices for domestic use. In 2023, the manufacturer Espressif reported in a statement that one billion units of this chip had been sold worldwide to date.

Tarlogic has detected that ESP32 chips, which allow connectivity via WiFi or Bluetooth, have hidden commands not documented by the manufacturer. These commands would allow modifying the chips arbitrarily to unlock additional functionalities, infecting these chips with malicious code, and even carrying out attacks of identity theft of devices.

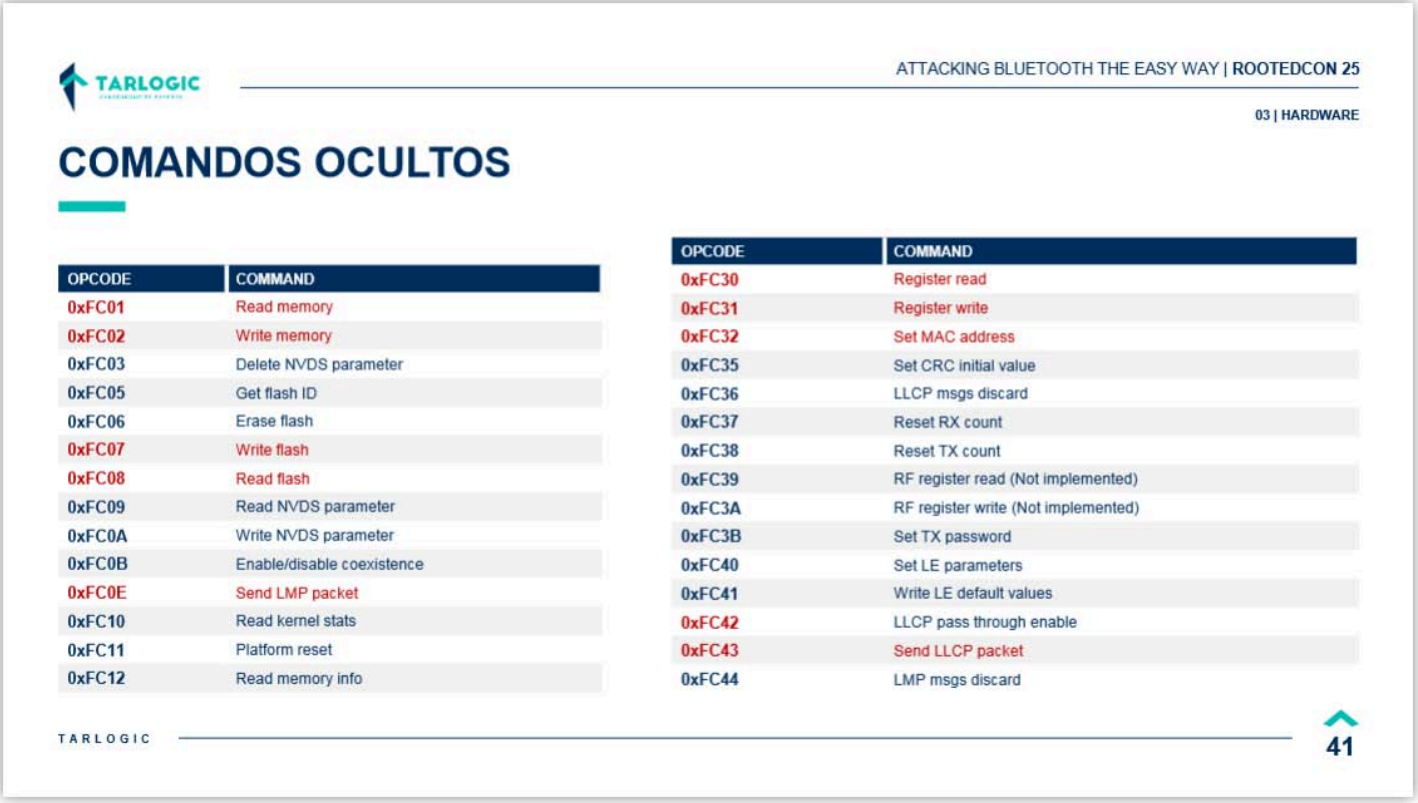
In this way, malicious actors could impersonate known devices to connect to mobile phones, computers and smart devices, even if they are in offline mode. For what purpose? To obtain confidential information stored on them, to have access to personal and business conversations, and to spy on citizens and companies.

Okay. So that's the meat of their finding. The rest of their posting talks about the broader scope of their mission, which is to create a platform to support Bluetooth security audits. Which is certainly a very worthwhile endeavor.

So what have we got here? The Bluetooth HCI defines the boundary between the host processor and the Bluetooth protocol stack. HCI is the abbreviation for "Host Controller Interface". And this is the jargon that's become standardized. Our listeners will have often heard me talking about adding AHCI support to SpinRite 6.1. That's the so-called Advanced Host Controller Interface that was created to manage SATA-connected mass storage devices. So HCI – Host Controller Interface – is a generic reference describing the interface boundary between a peripheral device and its processor.

So this Spanish security group designed and developed a technology, created a new capability, that would allow them to audit the operation of Bluetooth-enabled devices. And what did they discover? They discovered that by far the most widely used microcontroller that lives at the heart of by far most IoT devices contains an array of undocumented HCI commands that can be received over the chip's Bluetooth radio.

I deliberately chose to use the word "undocumented" because it's less freighted with intent than the word "secret". But until last week, these very powerful commands were, indeed, secret. Some have suggested that they may have been left in by mistake. No. 29 commands that were "left in by mistake" in more than a billion chips, across many sub-family versions? Calling that a mistake is not where I'd put my money.



The security conference was held in Spain. It was conducted in Spanish and the slide presentation was all Spanish except where, as we often see in code, English appears in code snippets. Staring at the portions of their 46-slide deck that were understandable to me – chunks of reverse-engineered and disassembled code – I began to get the sneaking suspicion that while these commands might indeed be undocumented HCI commands which would be executed by the Bluetooth hardware, it wasn't clear to me that these were remotely-accessible commands. They appeared to be running their own code on the ESP32 hardware and also reverse-engineering pieces of its firmware.

Since this is the era of helpful AI, I uploaded the Spanish slide deck to ChatGPT-'s latest 4.5 model and asked for a translation into English. It did a beautiful job, and my suspicions were confirmed. The Tarlogic posting ended by writing "Over the coming weeks, we will publish further technical details on this matter." So it may be that they have more than they're saying.

But the only thing I believe they've discovered is that the ESP32's Bluetooth HCI controller contains some commands that are undocumented because documenting them was not important. Discovering that an HCI controller contains a command – which the host CPU issues to it – that allows the controller to write to main memory could hardly be considered earth shattering. The host which issues the command is just as able to write to main memory as its controller. So ... big deal.

If an unauthorized external Bluetooth radio were able to issue such a command remotely to an ESP32-driven device, while presumably providing the data to be written into the system's main memory, and if this discovery existed in more than a billion of the devices we're all using, well then that would, indeed, be the end of the world as we know it. But the world is still here and I haven't seen evidence of that capability in their presentation.

And now that it seems clear that this all amounts to "host-side" access to an HCI controller, the threat this poses seems more like a mouse hole than a backdoor.

BleepingComputer noted that Espressif, the creator of more than a billion of these amazing little chips had not replied to their inquiry. That's likely because they also know that this is nothing.

At one point in the presentation the security researcher mentioned cloning another device's MAC address. Whoopie do! Sure enough, one of the 29 undocumented commands was "Change MAC Address". So, yeah, that can be done. But that's certainly neither a backdoor nor big news.

So I'm strongly inclined to come away from all this with the conclusion that there's really not that much here. It made for some really terrific attention-grabbing headlines, but nothing I've seen has suggested that the ESP32 chip is not still completely secure from external attack. They talk about being able to establish persistence. But if you're running code in a flash-enabled chip, persistence is not difficult to obtain. Among the undocumented commands is write to flash. I don't know whether writing to flash is a privileged operation on the ESP32. So perhaps these undocumented HCI commands could be abused to bypass capability or security restrictions. And that would definitely not be good. But even if so, an attacker's code would first need to get into the ESP32 in order to be able to do that. So the fact that it's the Bluetooth HCI that contains these undocumented features is entirely beside the point. It has nothing actually to do with the HCI's Bluetooth'ness. They just happened to discover these in the Bluetooth HCI because that's the target of their cross-platform low-level security auditing tool.

It's true that these commands should have been documented. And they still should be. But it sure looks like this is not a "sky is falling" discovery.

https://nvd.nist.gov/vuln/detail/CVE-2025-27840
https://www.bleepingcomputer.com/news/security/undocumented-commands-found-in-bluetooth-chip-used-by-a-billion-devices/
https://www.tarlogic.com/news/backdoor-esp32-chip-infect-ot-devices/