RAMBO

Description: Microsoft's "Recall" uninstallability is a bug. YubiKeys can be cloned. How worried should you be? When was that smoke detector installed? We share and discuss lots of interesting listener feedback. Is WhatsApp more secure than Telegram? Does Telegram's lack of security really matter? Elevators in Paris have problems, too. There's a fourth credit bureau to be frozen. Can high-pitched sound keep dogs from barking? A reminder of a terrific Unix 2038 countdown clock. A new Bobiverse sci-fi book and new Peter Hamilton novel. Why does SpinRite show user data flashing past? And TEMPEST is alive and well in the form of the latest RAMBO attack.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-991.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-991-lq.mp3

SHOW TEASE: Coming up on Security Now!, I am Mikah Sargent subbing in for Leo Laporte as he is on vacation. And we have a great show ahead. Steve tells us all about how YubiKeys can be cloned and how maybe that's not something you have to worry about. But, I mean, if you are, you know, a leader of a nation-state, maybe it is something you need to be concerned about. Plus we talk about a lot of listener feedback, including whether WhatsApp is more secure than Telegram, what is going on with elevators in Paris, and of course we cover the very interesting RAMBO, an attack that involves air-gapped systems, special encoding, and the ability to read RF signals coming from RAM. Very interesting stuff coming up on this episode of Security Now!.

MIKAH SARGENT: This is Security Now! with Steve Gibson and me, Mikah Sargent, Episode 991, recorded Tuesday, September 10th, 2024: RAMBO.

It's time for Security Now!, the cybersecurity show that you tune into every week so you can panic and feel anxiety, but then feel better because Steve Gibson is here to help you feel better. I am Mikah Sargent subbing in for Leo Laporte this week, who is enjoying a nice time away on vacation. And I am pleased to be joined by the true star of this show, Steve "T" Gibson. Hello, Steve.

Steve Gibson: Hello, Mikah. It's great to be working with you this week. And on top of the normal level of anxiety that this show tends to induce, today's podcast is titled "RAMBO." So if you were thinking, well, we're just going to have a calm little nothing podcast, no, that's not the case. We've got all kinds of good stuff to talk about.

We're going to cover Microsoft's Recall, which briefly looked like it was going to be uninstallable. Microsoft decided, no, that's a bug. Also YubiKeys can be cloned. How worried should we be? When was that smoke detector installed? Yeah, oh, this is probably one you missed, Mikah, so you're going to enjoy this one. This actually relates to last week's Picture of the Week.

We're also going to share and discuss lots of interesting listener feedback. Is WhatsApp more secure than Telegram? Does Telegram's lack of true security really matter? Turns out that elevators in Paris have problems, too. The relevance of that will be made clear. There's a fourth credit bureau that should probably be frozen. Can high-pitched sound keep dogs from barking? We'll be harkening back to a long-ago podcast. We also have a reminder of a terrific Unix 2038, you know, "end of the world as we know it" countdown clock.

There's a new Bobiverse sci-fi book - that early series was very popular among our listeners - and a new Peter Hamilton novel, also another one of our sci-fi faves. Why does SpinRite show user data flashing past? And the main topic of the show, RAMBO, tells us that TEMPEST-style attacks are alive and well. So a lot of good things to talk about. And of course we've got a great picture, sort of a classic Picture of the Week. And Benito pointed out that it also fits right in with today's recent Apple news theme that you just were talking about over on MacBreak Weekly. So I think a great podcast for our listeners.

MIKAH: Absolutely. In fact, I thought this was specifically picked for the Apple event, so I was kind of pumped.

Steve: Okay. So this week's Picture of the Week, again, I owe such a debt of gratitude to our listeners, who keep sending me these wonderful things that they encounter. I gave this picture the headline "The Very Definition of Form Over Function." So what we have, gates of various sorts seem to be a recurring theme. We've often, like, found gates out in the middle of a field, like what is going on here? In fact, we had one really famous one, a gate, and a bunch of sheep were standing behind it as if waiting for it to open, even though they could have walked around it. So never really clear what was going on with those sheep. But in this - oh, and another one of my favorite gates was a gate that blocked people from passing through with a series of horizontal bars, meaning that it was also a ladder that you could easily use to climb over the gate. Like, okay, maybe these bars should have been vertical.

Well, in this case this "form over function" gate is blocking sort of a long corridor, clearly meant to prevent people from passing. There's some sort of locking mechanism and handle and so forth over on the left. So it makes it look like this gate will not open unless you're authorized. Yet they wanted to celebrate the Apple. And so more than a third, maybe not quite half of the gate is a large apple made out of the same bar material, except the center of the apple is open. I mean, you can just go through the gate...

MIKAH: Squeeze right through.

Steve: ...by ducking down and moving through. And at first I thought, okay, maybe this was not really meant to keep people out. Except that, if you look on the outside the gate, like in the margins outside of that gate area, they have extra bars there, extending to the very edge, definitely intended to keep anybody from squeezing around the side. But you don't have to squeeze around the side because the center of the gate is a wide open body of an apple. Anyway, you know, there are so many of these pictures that we've shared where you want to find the person...

MIKAH: You do; right?

Steve: ...who was in charge of this design and say, okay, now, like the apple. Is this actually supposed to keep anybody out? Like what's your thinking here?

MIKAH: What were you thinking? What were you thinking? You have to tell me.

Steve: Okay. So The Verge carried some news that really makes you wonder what's going on at Microsoft. Their headline read: "Microsoft says its Recall uninstall option in Windows 11 is just a bug." In other words, don't get your hopes up that we're going to allow our illustrious forthcoming "Recall" feature - which as we know most people don't want - to be removed from Windows. That was a bug, not a feature.

So The Verge wrote: "While the latest update to Windows 11 makes it look like the upcoming Recall feature can be easily removed by users, Microsoft," they wrote, "tells us it's just a bug, and a fix is coming." Meaning that that option will be removed as like the bug fix. They wrote: "The Deskmodder spotted the change last week in the latest 24H2 version of Windows 11, with KB5041865 seemingly delivering the ability to uninstall Recall using the Windows Features section." And I grabbed a snapshot of that for the show notes. And you can see it very clearly. Underneath the Print and Document Services option is Recall, which is checked, has the blue checkmark. And that's just above the Remote Differential Compression API Support, which I guess everybody wants to have. So, you know, there it is, suggested that if you were to uncheck that and click okay, Recall would be removed from your life, but only for apparently that release.

In a statement to The Verge, Microsoft's senior product manager, Brandon LeBlanc, said: "We're aware of an issue where" - I love it, it's an issue. "We're aware of an issue where Recall is incorrectly listed as an option under the 'Turn Windows features on or off' dialog in Control Panel." Which we would all argue is exactly where it should be. But Brandon said: "This will be fixed in an upcoming update."

So the Verge goes on to tell us much of what we already know, which is to say why many of us wish that the checkbox would remain. But The Verge also adds a bit of news. So they wrote: "The controversial Recall AI feature, which creates screenshots of mostly everything you see or do on a computer, was originally supposed to debut with Copilot Plus PCs last June. Microsoft was forced to delay the feature after security researchers raised concerns." Like multiple rounds of concerns; right? "Microsoft says it remains on track to preview Recall with Windows Insiders on Copilot Plus PCs in October," so that's next month, "after the company has had more time to make major changes to Recall." Which nobody would argue it needs.

They said: "Security researchers initially found that the Recall database that stores the snapshots of your computer every few seconds was not encrypted, and malware could have potentially accessed the Recall feature. Microsoft is now making the AI-powered feature an opt-in experience instead of on by default, encrypting the database, and authenticating through Windows Hello." Now I'll just pause here to note that Windows Hello has been broken multiple times, and they're not going to be saying, well, you know, we have this Recall feature, but people are really uncomfortable with it. No. They're going to be saying, we have Recall. It's jiffy quick spiffy wonderful, and you definitely want to...

MIKAH: Use it. You want it. It's amazing.

Steve: Oh, my god, you're not going to - after a year, you're going to wonder how you ever got along without it. So, yeah, it may be opt-on, but it's sort of like, if anybody's been using Windows recently and tried not to back up their computer using one of Microsoft's facilities, you know you have to say no, no thank you, no, I'm really seriously sure I don't want to use your backup because, oh, no, they want you to back up to OneDrive.

Anyway, The Verge said: "We did ask Microsoft whether it will allow Windows users to fully uninstall Recall, as this appearance in the Windows features list suggests. But the company only confirmed this was just 'incorrectly listed' for now." They said: "It's possible that Microsoft may need to add a Recall uninstall option to EU copies of Windows

11 to comply with the European Commission's Digital Markets Act," you know, the DMCA. "Microsoft has already had to add an option to uninstall Edge in the European Economic Area countries, alongside the ability to remove the Bing-powered web search in the Start menu." So, you know, maybe Europe is going to come to our aid. Although the problem is, you know, Microsoft knows if these copies of Windows 11 are in the EU or not. So that may not help us.

And, you know, when you really think about it, what does it mean that Windows has a feature that presents a clear and present privacy and security danger to all of its users, which Microsoft knows full well many of its users feel extremely uncomfortable about, and where it's obvious that the feature could be readily removed from Windows, yet Microsoft refuses to allow their users to do so. One thing that means for certain is that GRC's forthcoming freeware, which will totally neuter and remove Recall, promises to be quite popular.

MIKAH: Okay. Let's talk about this. Okay. So first and foremost, I 100% agree, and right now there have to be at least 10 people inside of the company who are cringing at the fact that this was discovered in the first place because it's such a clear, easy way to say, if it can be removed, and we're not giving people the ability to do so, now that's there. That's so bad that they've shown that it can be taken away, and now they're not giving people the opportunity to completely take it away. Secondarily, though, do you have concerns that - because you've created other tools like this. So I know that you have a better understanding of this. Freeware that removes something that Microsoft says can't be removed or shouldn't be removed, does that introduce any issues in the system? Or do you have concerns about that?

Steve: I guess I would say we'll find out. So what this most reminds me of is what we went through with Internet Explorer back in the early days of Windows. Microsoft was so committed to having their own web browser built into Windows that they told the world that it could not be removed. You know, there was no way to remove IE from Windows. And so, you know, that was the story that we heard, I mean, and this was where a lot of the antitrust problems came from back in the beginning. So we kept hearing, oh, no, the browser is an integral component. It's deeply integrated into Windows and cannot be removed, until the EU said, you know, we think that's wrong.

And so then Microsoft made it removable. So, you know, it was only unremovable because Microsoft didn't want it to be removed, until they were forced to say, okay, well, I guess we'll let people turn it off. And in fairness, there are some parts of Windows that have always been and almost still are dependent upon IE components. So, you know, it was integrated into Windows.

What's interesting here is that the progression of this demonstrates that it is an add-on to Windows. I mean, Windows 10 doesn't have it. Windows 10 is apparently going to be getting it. Windows 11 doesn't have it. Windows 11 is definitely going to be getting it as a feature of Copilot. So I think it's, I mean, Microsoft could certainly - Microsoft can do anything they want to with the OS; right? So they could arrange to make Windows dependent upon it in some fashion. But to your point, clearly the fact that there is a Remove It feature now, and they're removing the Remove, rather than removing Recall, suggests that Recall can be removed.

MIKAH: Yup.

Steve: So I will - and if it turns out that it cannot be removed, that is, like it literally cannot be removed from the system, or that every Windows Update brings it back if it's taken out, you know, whatever, then the least I could do would be to have my thing set up a little background service in Windows whose job is to absolutely, you know, kill it

when it appears, or turn it off when it gets turned on, I mean, just basically, you know, take responsibility...

MIKAH: A Whac-A-Mole.

Steve: ...for keeping it shut down. Now, and we know that that is possible because Microsoft has said, if you're doing something sensitive, you can stop Recall from snapshotting your system while you're doing something that you specifically don't want it to watch. So I just...

MIKAH: Which is everything for me.

Steve: Exactly. And so it's certainly - and I do have a great name for it. I'm just keeping it quiet for now. But yeah. Leo, I'll never forget him laughing when I told him that the freeware that was going to prevent Microsoft from upgrading your Windows 7, of course I called it Never10. Which he really liked. Anyway, I've got something good for this one.

MIKAH: Oh, good.

Steve: Yeah. As soon as it actually happens and becomes a problem, I'll spend, I mean, it's only going to take a couple days to create something that does the job.

MIKAH: Nice.

Steve: Okay. So many, many, many people sent me a link to, like, at least as many pointers to the recent YubiKey exploit stories as I received with news about this RAMBO attack because people wanted to hear what I thought about this RAMBO attack. And also, hey, Gibson, look, your favorite key has a problem. So of course this is due of course to the fact that Yubico themselves largely credits me, thanks to the listeners of this podcast, with discovering them at an RSA conference where I met Yubico's primary mover and shaker and co-founder, Stina Ehrensvard. And then, you know, the podcast put them on the map and really helped them going.

Now, it's clear that this would certainly have happened for Yubico sooner or later, and knowing Stina, probably sooner. So it was just fortune that I happened to be someone, you know, who had a microphone, who recognized the cleverness of what they had created back then. And of course the YubiKeys have evolved dramatically since that first thing that was basically a keyboard, a USB keyboard emulator, which was very clever. And so the world is changed. But Yubico has remained the leader.

Ars Technica's headline about the recent discovery was: "YubiKeys are vulnerable to cloning attacks thanks to newly discovered side channel." And their subhead read: "Sophisticated attack breaks security assurances of the most popular FIDO key."

The researchers at NinjaLab, who performed the research and previously informed Yubico of their findings, so much so that Yubico has already solved the problem for any new keys that they then sell, NinjaLab said: "In the present work, NinjaLab unveils a new side-channel vulnerability in the ECDSA [Elliptic Curve Digital Signature Algorithm] implementation of Infineon 9." That's the actual chip inside the YubiKey that does the crypto on any security microcontroller family of the manufacturer. Meaning of Infineon.

They said: "This vulnerability lies in the ECDSA ephemeral key (or the nonce) modular inversion and, more precisely, in the Infineon implementation of an Extended Euclidean Algorithm." They said: "To our knowledge, this is the first time an implementation of the EEA" - that's the Extended Euclidean Algorithm - "is shown to be vulnerable to side-channel analysis," they said, "contrarily to the EEA binary version. The exploitation of this vulnerability is demonstrated through realistic experiments" - and we'll discuss how

realistic they are in a minute because it takes something to make this happen - "and we show that an adversary only needs to have access to the device for a few minutes." Although I'll put "access" in air quotes, as we'll see.

They said: "The offline phase," that is, after the access for a few minutes, "took us 24 hours. With more engineering work in the attack development, it would take less than an hour." So yes, it's possible after we've demonstrated the problem to improve its performance.

They said: "After a long phase of understanding Infineon implementation through side-channel analysis on a Feitian 10 open Java Card smartcard, the attack is tested on a" - so that was a smartcard using the same chip. So they actually developed the attack on something no one had ever heard of. Then they thought, okay, to get some press we're going to see if the YubiKey is vulnerable. So they said: "The attack is tested on a YubiKey 5Ci, a FIDO hardware token from Yubico. All YubiKey 5 Series," they said, "before the firmware update 5.7.11 of May 6th, 2024 are affected by the attack. In fact, all products relying on the ECDSA of Infineon cryptographic library running on an Infineon security microcontroller are affected by the attack." In other words, all kinds of other things, too.

They said: "We estimate that the vulnerability exists for more than 14 years in Infineon top secure chips. These chips and the vulnerable part of the cryptographic library went through about 80 CC certification evaluations of level AVA VAN 4 (for TPMs) or AVA VAN 5 (for the others) from 2010 through 2024." And they said: "A bit less than 30 certificate maintenances." Okay. So in other words, this has had the crap tested out of it.

MIKAH: That's what I thought it meant. Okay, good.

Steve: Yeah, over and over and over. So, like, no problems were ever found. And everybody's using this in the industry because it is the industry standard secure microcontroller, super high volume, super low cost, and that's what's in the YubiKey 5 series, as well as many other secure tokens and HSMs, for example, of different kinds.

Okay. So in his reporting of this for Ars Technica, Dan Goodin wrote: "The attacks require about $11,000 worth of equipment and a sophisticated understanding of electrical and cryptographic engineering. The difficulty of the attack means it would likely be carried out only by nation-states or other entities with comparable resources, and then only in highly targeted scenarios. The likelihood of such an attack being used widely in the wild is extremely low," as in nil. "Roche said that two-factor authentication and one-time password functionalities are not affected."

So this is a specific function among many functions in this cryptographic library. So two-factor authentication, one-time password functionalities not affected. So it's very likely, and I didn't dig into this enough to get a sense for, of all the things that are affected, what are. But FIDO is. So FIDO2 passkey stuff, that'll be a problem.

Okay. So he said: "Tuesday's report from NinjaLab outlines the full flow of the cloning attack as: First the adversary steals the login and password of a victim's application account protected with FIDO." Right? So it's not like the key is all you need. You still need the login and password first.

"The adversary gets physical access to the victim's device during a limited timeframe without the victim noticing. Thanks to the stolen victim's login and password for a given application account, the adversary sends the authentication request to the device as many times as necessary while performing side-channel measurements." In other words, you give your login and your password to authenticate, then the account says now you need to use your key in order for us to verify that you're in physical possession of the

key. So the bad guys have to make all that happen so that the device is actually doing successful passkey or FIDO authentications over and over and over.

Then they said: "The adversary performs a side-channel attack over the measurements and succeeds in extracting the ECDSA private key linked to the victim's application account. The adversary can sign in to the victim's application account without the FIDO device and without the victim noticing." Now, notice that they already could because they had the key somehow; right? So what this is doing is, it's allowing them future access, which they already got, you know, present access for because they had the guy's login and password and their device, like in a very intrusive way so they were able to do things with it, use it, successfully authenticate. So it was during multiple successful authentications under the scrutiny of this $11,000 worth of equipment that they're getting the private key whose entire purpose is not to do something now, but to do something in the future.

And then of course they've got to get this back - it's like a "Mission Impossible" episode. They've got to get this back to the user. And wait till you hear that they had to crack the key open in order to do any of this anyway because that's part of it, too. So they had to glue it back together after cracking it open, and getting it back to the guy before they know that it's ever been taken so that they won't go and change the key because anyone who knew this had been done would stop using it; right? So it's like, okay, fine. So this allows them to in the future sign into the victim's application account using the stolen login and password, and the then-stolen elliptic curve private key, which allows them to do this in the future.

And then Dan says: "The list," which we just finished, "however, omits a step, which is tearing down the YubiKey and exposing the logic board housed inside." He says: "This likely would be done by using a hot air gun and a scalpel to remove the plastic key casing and expose the part of the logic board that acts as a secure element storing the cryptographic secrets. From there, the attacker would connect the chip to the hardware and software that take measurements as the key is being used to authenticate an existing account. Once the measurement-taking is finished, the attacker would seal the chip in a new casing and return it to the victim." So...

MIKAH: Okay. This isn't easy. This isn't realistic for the average - even the person who had the equipment, like there's still so much that could go - what could possibly go wrong? So much. It'd be hard to do this.

Steve: Yeah, it's not like you scan it in some guy's pocket...

MIKAH: Right.

Steve: ...from 20 feet away or something. So to put this into context, Dan adds, he says: "The attack and underlying vulnerability that makes it possible are almost entirely the same as the one that allowed NinjaLab to clone Google Titan keys in 2021. The attack required physical access to the token for almost 10 hours in the case of the Google Titan keys." He says: "The attacks violate a fundamental guarantee of FIDO-compliant keys, which is that the secret cryptographic material they store cannot be read or copied by any other device. This assurance is crucial because FIDO keys are used in various security-critical environments, such as those in the military and corporate networks.

"That said," he writes, "FIDO-compliant authentication is among the most robust forms of authentication, one that's not susceptible to credential phishing or adversary-in-the-middle attacks. As long as the key stays out of the hands of a highly skilled and well-equipped attacker" - with a hot air gun and a scalpel, I'll just interject - "add $11,000 worth of equipment and the ability to get it out of your possession for the time required

to do it, and who already knows your login name and password. It remains among the strongest forms of authentication."

He says: "It's also worth noting that cloning the token is only one of two major steps required to gain unauthorized access to an account or device. An attacker must also obtain the user password used for the first factor of authentication. These requirements mean that physical keys remain among the most secure authentication methods.

"To uncover the side channel," he finishes, "the researchers reverse-engineered the Infineon cryptographic library, a heavily fortified collection of code that the manufacturer takes great pains to keep confidential. The detailed description of the library is likely to be of intense interest to cryptographic researchers analyzing how it works in other security devices."

Okay. So what we have here is Yubico in the spotlight only because it is by far the most successful and well-known user of high-security token hardware by Infineon that, despite years, 14 years of previous reviews and extensive analysis by the industry, was finally found to have an extremely subtle flaw that could be used to extract its secrets; and even and only then through the use of quite high-end expensive engineering equipment, including the need to physically compromise and crack open the key. And even then, the attacker would still need knowledge that only the key's legitimate owner and user probably possesses.

So Infineon has fixed their problem with a firmware update. But in the interest of security, Infineon's firmware is not field upgradeable. So Yubico has obtained the improved hardware from Infineon and is now offering keys that have this fixed. Whether or not anyone should or would bother to update is up to them. But this attack seems so far-fetched, I mean, literally, Mission Impossible 9, and is so far out of the realm of ever happening to anyone - and, after all, we're just using the keys to contain additional factors of login credentials - that I can't imagine this is worth another thought.

MIKAH: Can I ask you, you say in the interest of security, Infineon's firmware is not field-upgradable.

Steve: Right.

MIKAH: How is it, and it's probably just not obvious to me, why is that a security thing, to say you can't upgrade the firmware and fix this? Why does it have to be new hardware?

Steve: So what it actually is, the firmware is in ROM, in literally old-school, it's called Masked ROM, where the bits are actually little bits of metal mask which are present or not. Literally like physical connections making ones and zeroes. As opposed to it being in flash memory, where it is dynamically writeable. And the reason is, if you are able to put the key in and change the firmware, then an attacker could put it in and change the firmware to be insecure.

MIKAH: Got it.

Steve: So you just, you absolutely, like the top-level security says we fixed this in the factory so that you can't ever change it.

MIKAH: Got it. So it is the, like, the fruitcake firmware. It's so dense, so it's nearly concrete, as opposed to firmware that exists on the other side where it's a software update. This is literally, like, actual physical firmware update.

Steve: I would say that we're using the term "firmware" because it's the code that drives the microcontroller inside. But it's actually so firm that it's hard.

MIKAH: Yeah.

Steve: It's actually hardware firmware.

MIKAH: Got it, yeah. Wow, that's cool. That's really cool. But you're saying, though, given that you basically have to have a significant other who is secretly a spy from some nation-state organization to, A, be able to guess that your username and password is this, or have access to that in the first place, and can get that away from you and can keep you entertained by something for 10 or more hours, I think it's more in this case, to be able to pull this off. You and I probably don't need to go buy a new YubiKey.

Steve: I'm not the least bit concerned. You know, Yubico's been totally responsible. They immediately, in concert with the announcement from NinjaLab, they put out their own explainer that said, yes, it turns out that our supplier and the supplier of everybody else on the planet has a problem. This is the nature of it. And we've responded the only way we can because these are not field-upgradeable, is if you really are concerned, we'll offer you a new key. But, yeah, really, you know, we're really talking nation-state level exposure for this to be a problem.

MIKAH: Got it. So, yeah, I will keep my YubiKeys, thank you very much. I have 5 Series, and so whenever you first said this I thought, oh, dear.

Steve: Yeah, not a problem.

MIKAH: All right. We are back from the break. I am Mikah Sargent subbing in for Leo Laporte this week. And it's time once again for Steve to take it away.

Steve: Okay. So we've got - that was pretty much the big news of the week, along with RAMBO that we'll get to in a minute. I have a bunch of interesting listener feedback that I wanted to share because the GRC mail system has been working overtime, I should say. So our Picture of the Week podcast before last was that signage which was intended to have its blank field proudly filled in with the date since there had last been any sort of accident on the job. But as we remember, instead of that, it cited that they'd had no accidents since a specific person, who everyone presumably knew, had left the job. So, you know, this site has been accident free - yeah, there it is. It said "Since Joe left."

MIKAH: Aw.

Steve: Okay. So I recalled at the time that we'd had a similar non-sequitur once before for our Picture of the Week in the form of a close-up photo of a smoke alarm that also had a blank space where its installer was expected to fill in a date. But during the podcast two weeks ago I was unable to recall what had been written there when we showed that Picture of the Week before. One of our listeners whose online moniker is "Mr Nobody 2" was kind enough to remind me.

Okay. So the smoke alarm had, as I said, had a field where its installation date was meant to be filled in by its installer. So it said: "Installed On:" and followed by a blank space. In this case, the person filled the information in so that it read: "Installed On: The Ceiling."

MIKAH: Oh, that's great.

Steve: It was wonderful. So anyway, I wanted to remind everybody of that similar repurposing of the original intent. Like, you know, looking at it you wouldn't know that your smoke alarm was installed on the ceiling, so yeah.

MIKAH: Mine, unfortunately, I can see here, it's just not - it doesn't have anything on it.

Steve: Well, and I'm not really sure because, you know, these things start to beep when the batteries get low. So maybe technically you're supposed to replace them every 10 years or something, like maybe the battery isn't the only problem. Maybe the smoke sensing sensor could go bad. You know, so like, oh, well, replace it when you need to change the batteries. But after a decade you should just really go get a new one. Anyway, just in case anyone was wondering, the smoke sensor is on the ceiling, if you weren't secure what surface that was.

So anyway, as I said, there was not a huge amount of news. And I got caught up in the terrific listener feedback that I'd been receiving. As our longtime listeners will remember, many years ago we used to deliberately alternate episodes between security news and listener feedback, where we would do a pure feedback episode. We dropped that approach over time in favor of always doing some of both, which is our normal routine, as we are this week. But a little more feedback this week.

And now I should note how pleased I am with the way GRC's email system has worked out. The nature of the feedback by email is completely different from Twitter; and having it, you know, coming into my own email client makes it significantly easier to manage. So I'll just remind everyone that in order to send feedback directly to me at the email address securitynow@grc.com, and that's not listed anywhere at GRC because I'm intending this to be for podcast listeners.

So, you know, people say, hey, I looked around, I wanted to send you a note because I know you're talking about this all the time, but I couldn't find the address anywhere. Right, because it's only for people who hear my voice. And my voice says "securitynow@grc.com." That's the email address. So again, you need to register your sending email address with GRC. You do not need to subscribe to any of the three mailing lists that you'll find there. Just being registered allows my system to prevent all incoming spam from anyone sending to securitynow@grc.com who's not registered. And that's a blessing because all I ever get there are actual listeners' email, and it's wonderful.

So anyway, I'm also hearing from many of our listeners who really appreciate receiving the weekly show title, summary, Picture of the Week, and the show notes link by mail every Tuesday morning before the podcast. So subscribing to the Security Now! List will automatically make that happen for you.

Okay. So Angus MacKinnon. He said: "Steve, I see on WhatsApp all the time that your messages are encrypted. Is WhatsApp secure? I thought WhatsApp had Signal embedded."

Now, of course, last week's podcast was all about the fact that Telegram, which claims security and boasts of its reputation for security, was not truly offering end-to-end messaging encryption, with the single exception that two - and exactly and only two - parties who were both online at the same time could deliberately enable point-to-point encryption for their conversation. So I'm sure that Angus just wanted some assurance that WhatsApp's similar claim of encrypted messaging is actually legitimate. And as he noted, since WhatsApp is based upon the open Signal protocol, all messaging is always fully encrypted, even in multi-party groups, you know, up to a thousand people in a group. So and in fact, since it's based on Signal, there's no way to use it or Signal in an unencrypted mode. That's all they offer. So 100% yes for WhatsApp.

Now, at the same time, Andy Pastuszak shared some useful points which he feels favors Telegram. He wrote: "Steve, I'm a user of Telegram as well as Signal. The definition of anything less than end-to-end encryption as not being true encryption would make a LOT [all caps] of services not encrypted, even outside the messaging space." He says: "There are almost no cloud providers that offer true end-to-end encryption. Dropbox, OneDrive, and Google Drive don't. Online calendar, to-do lists, and note-taking apps don't really either. And the ones I find that do, charge A LOT for the privilege. Some of the end-to-end encrypted note-taking apps I looked at charge well over $100 per year for their basic plan.

"Telegram," he says, "is obviously NOT [all caps] end-to-end encrypted. But it is encrypted in transit and encrypted at rest. For the things I use Telegram for, all I really need is encryption in transit. If I really need end-to-end encryption, then I use Signal.

"The other nice thing about Telegram is how group chats work. How many times have you been part of a group SMS text, and asked to be removed from it? And that works great until someone responds to an old message that you're still included on, and then all of the sudden you're part of the conversation again. With Telegram, you leave a group chat or channel, you're gone until you rejoin.

"And Telegram fully supports Siri and CarPlay. I can easily say 'Hey, Siri, send a Telegram message to Joe' while driving, and it will happily do that. Signal does not have Siri or CarPlay support yet. So if you want something better than SMS, with Siri and CarPlay and Android Auto support, and you're aware of the encryption limitations, Telegram is an excellent choice." Okay. I agree...

MIKAH: Lot of caveats.

Steve: Huh?

MIKAH: I said a lot of caveats there.

Steve: Yeah.

MIKAH: You like this, but you don't care about the - yeah, sorry, go ahead.

Steve: Yeah. No, you're right. I agree with Andy that true end-to-end encryption is rarely needed or necessary. I use iMessage among my iPhone-using friends. As we know, its encryption is what Matthew Green described as modern state-of-the-art, you know, true end-to-end encryption. But the messages I'm sending are about what time we're meeting for dinner, or whether they saw some random piece of news. Hardly anything that would ever be of interest to anyone else.

Andy is obviously a sophisticated user who understands exactly what's going on. After all, he's listening to this podcast. So there's nothing to disagree with him about. One of the points of his sophistication is that he knows that when he truly needs end-to-end encryption, it's time to switch to Signal for that. He said so. But a big part of what Matthew Green wanted to convey - although unfortunately it was only being read by people like us, so it didn't come as a huge surprise - was that the typical Telegram user - not Andy - was extremely unlikely to have any such sophistication and thus appreciation of the distinction between Telegram and Signal or WhatsApp.

So Matthew told us that what he was growing increasingly annoyed about as the years rolled by with Telegram not making any significant improvements to the security of their messaging technology, was that they were essentially riding on the coattails of all of the other fully, truly end-to-end encrypted messaging platforms, while all the while claiming privacy parity with them while choosing to not actually offer it.

So Andy, I don't take issue with anything you said except you're the exception, not the rule, among Telegram users, of which there are, you know, a billion who don't understand all of this.

John Hickin said: "Steve, we rented an apartment in Paris where a sign was present in the elevator, but in French, of course. It was put up by owners who were annoyed when renters (AIRBNB) forgot to close the outer door after leaving the elevator, thus rendering it stuck in place so nobody on any other floor could recall and use it." And he said, "Cheers, John."

Okay. So I enjoyed John's note which related, of course, to last week's Picture of the Week, remember, which suggested that if the elevator didn't "go," its occupants should try jumping up and down a bit, which, you know, should give anyone the willies. Presumably that would allow the elevator to know that they were present, although one would imagine that pressing a floor button would serve that purpose.

Anyway, apparently in Paris still to this day they're using those quaint elevators where its user first closes an outer door on the floor, and that door remains on the floor to close off the elevator shaft while the elevator is not there, and then the elevator itself is only responsible for closing the inner door of its own carriage. So as John notes, if people leaving an elevator leave the outer door open, which is not under automation, then the elevator is unable to close that outer door, so the elevator is unable to move. And anybody pushing for the elevator to go to their floor will end up having to take the stairs and be an extra-annoyed Parisian, which may go a little ways to explain how they feel about Americans visiting Paris. So, you know, American tourists are, well...

MIKAH: You can just leave that blank. It fills itself in.

Steve: Craig Taylor said: "Hi, Steve. Longtime listener. I wanted to provide you with some additional information on the article you cite for Freezing Credit after the NPD breach. The article you reference for freezing credit only mentions three of the four major credit bureaus at which you need to freeze your credit. Innovis is missing from that article." He says: "Our article at CyberHoot has a collection of many of the primary and secondary credit bureaus." And I have a link in the show notes. And he finished, saying: "Great coverage, and thanks for doing what you do."

Okay. So Craig is a co-founder of CyberHoot, and the page he linked to does indeed provide more comprehensive coverage of the various credit bureaus with links to each bureau's individual credit freeze resources page. The GRC shortcut "npd" - that's what I thought it was, it's not NDP as he had. He got those backwards. Or maybe I did, I transliterated them. Anyway, it's NPD, for checking the NPD breach database. That shortcut, grc.sc/npd received the largest number of referring clicks ever, of all time, and that was just a couple weeks ago. So it hasn't even had that long to age. And GRC's "credit" link, grc.sc/credit, to the Investopedia page is next in line, just behind it, in second-place runner-up position. The /npd was something like 10,000-plus clicks. And the grc.sc/credit link was 9,000 some.

So I know this topic is, not surprisingly, of significant interest to this podcast's listeners. Since I want to make Craig's more comprehensive listing of credit bureau credit freeze links readily available, I've created another new GRC shortcut. This one is "freeze." That points to Craig's excellent page about identity theft. So the link is grc.sc/freeze.

Since I'd only previously frozen my own credit at the big three - TransUnion, Experian, and the infamous Equifax - I immediately used the new link to Craig's page to find the link to Innovis. I went there and froze my credit. To Innovis's credit (pardon the pun), it was the easiest of any of the freezing experiences. No need to create an account. You just fill out an online form - which, by the way, contains all the data that's already been

made public in the breach, so it's not news to anyone - and your credit is immediately, from that moment, frozen against anyone's inquiry. Innovis then sends, by postal mail, a credit freeze confirmation letter which contains a 10-digit PIN. So you'll want to hold onto that. That PIN can subsequently be used to then manage your freeze status at Innovis. It was so quick and easy that I cannot imagine why anyone who cares about this would not do it. So again, the GRC shortcut to get to Craig's page at CyberHoot is grc.sc/freeze.

And I should mention that Craig's quite comprehensive page mentions an additional five lesser-known bureaus which also offer credit freezing. I didn't bother with them, but if you want to be fully covered you may wish to. I mean, like, why not? And Craig, thanks very much for bringing this additional major credit bureau and your page to our listeners' attention. That's much appreciated.

And I now send email out every morning with a summary of the podcast and a link to the show notes. So I've already received feedback from a listener who read the show notes and asked me, okay, what about these other five credit bureaus? I mean, they're there. Do we need to freeze them? So I don't know how to answer that. I remember Innovis. And I remembered that when we talked about this years before, there was a fourth bureau. I couldn't remember it when this came up again. And so, you know, I didn't take the time to dig into it. It was Innovis. So they're at least in number four position, probably big enough to count. The question about whether you should bother, like, how far down the list should you go, essentially, I don't know. I know that when I needed to apply for an Amazon card because I wanted my Amazon purchases to be on their card because they gave you extra points...

MIKAH: Really good deals, yeah.

Steve: Yeah, I found out which of the major three they used, and I did a temporary suspension of the freeze to allow Amazon to check my credit and then issue me a card, and then the freeze automatically snapped back on. So I know that Amazon uses one of the big three. Technically, anyone, you know, so the question is you're freezing your credit because you don't want someone to issue, some bad guy to obtain credit in your name. Well, if they're querying a random - if the company granting the bad guy credit in your name is querying some random credit bureau that may not have the best information for you, then I guess anything's possible; right? So, yeah, if this is a concern for you, lock them all down.

My feeling is that none of these bureaus received my permission at any point to collect this information about me; you know? They're just doing it. And I do think that probably the top four covers, you know, virtually all of the use. Presumably not absolutely all, otherwise these other five wouldn't be around. So I guess I don't really have a good answer to the question. But, you know, if credit bureaus keep popping up, do we just run around freezing them all? I don't know. And besides, all the information is now public.

MIKAH: Yeah, exactly. It's already out there. I wish that - so, you know, part of the requirements by the federal government involve providing the three major credit bureau reports every year, you get one free from each of the three. I think that, A, if they can network to do that, they should network to let you do security freezes all together with just one button. And I know this is just wishful thinking, but they've proven that they can work together in that capacity. Let's let them work together in letting us freeze easily, and unfreeze, all at once.

Steve: Well, and, you know, when we talked about this a couple weeks ago, the way the system should actually work is that all of the bureaus should always be frozen by default. And then if you are applying for credit, as I did for Amazon, or like someone's buying a home or buying a car or making any large purchase, anyone who wants credit should do

something to specifically authorize that person to have access to their credit report. So, you know, you receive a PIN from the credit supplier that that entity's going to use. You give them this long access PIN, which they are then able to use to obtain access to your report directly from the granting agency. I mean, there are ways we could make this work that don't even require, you know, fancy computers or being online or anything because that's still an issue. Not everybody - you don't want to like require that you have Internet access in order to do all this because all this predates the Internet.

But anyway, the system's broken, and we're just sort of limping along as we go. But I do think it was so easy to add a freeze to Innovis that I can't imagine why anybody who has taken the time to freeze the other three would not go ahead and do number four because it does round out the top four. And I think that's probably worthwhile.

MIKAH: Agreed. That's very easy.

Steve: Yeah.

MIKAH: I did it while we were talking. Perfect.

Steve: Nice, nice. Adam Tyler said: "Hi, Steve. I was curious if you or a listener have found a commercial version of the portable dog killer device?" He says: "I'm not really looking for a laser gun, but something that could sit on the fence line to deter a barking dog, ideally automatically activated, and a battery design that made sense. Lithium ion with a little solar panel would be sweet." He said: "Anyway, love the podcast. Glad you're going past 999. I also only had an X/Twitter account to DM you and am very happy to see you've moved over to email. Regards, Adam Tyler."

Okay. So Adam is, of course, referring to one of this podcast's favorite past episodes which we've re-aired a number of times through the years because it tells a fun story which ends with a moral of the surprising benefits that can arise from being active rather than passive. I first shared that youthful adventure of mine on the occasion of the 50th anniversary of the laser. The device I designed and built when I was in high school was not a laser, though the beam of high-intensity directed sound energy it produced was likely coherent.

Now, 12 years ago, back in 2012, when this podcast was only seven years old, I recreated that device after so many of our listeners commented that their neighbors' barking dogs were ruining their lives. Since I didn't have the web forum technology running that I have today, I created a Google group called "Portable Sound Blaster" for public discussion of this, and I published the final electronic design of the device which I had created on a page at GRC, naming the project "The Quiet Canine." If you're curious, you can find it under GRC's website menu under "Other," and down at the bottom is "The Quiet Canine." I think you can also just google "the quiet canine," and it comes right to my page.

Now, on that page I wrote: "The good news is that we arrived at an extremely simple, inexpensive, and easy-to-build design for a small, lightweight, and painfully loud handheld sound emitter." And then the page shows the design. But then under the caption "The Bad News" I wrote: "Many of these final TQC (The Quiet Canine) v2.2.2 devices were assembled and tested by those following and participating in the Portable Sound Blaster group at Google. The devices were invariably incredibly loud and high pitched. While their dads were assembling and testing the devices downstairs in the garage, their upstairs teenagers were complaining about the piercing sound penetrating their heads. And of course dogs were at least as well able to hear it, and at much greater distances.

"But in no event was this able to function as any sort of barking deterrent. Dogs heard it, and at great distance, but they didn't care. We soon came to appreciate that my own original "point blank" blasting of the original "Portable Dog Killer," as I named my first device when I was in high school, was required for the device's effectiveness. No dog next door, let alone down the block, will care about a high-pitched sound. It needs to be blasted directly into the dog's face at a very short distance.

"Now, this means that while this device would not be useful for silencing dogs at a distance, it would likely be extremely useful and effective as a personal defense device for people walking, postal workers on foot delivering mail, and joggers who are harassed and threatened by overly aggressive canines on the loose. Although we cannot and do not offer any specific guarantees, it's difficult to see how any attacking dog would not be stopped in its tracks by a close blast of incredibly loud and high-pitched sound." That's what the website says.

So the bottom line is my particular use-case, which I described in that story, turned out to be unique. I specifically designed and used that first device back in the early 1970s to train an incredibly aggressive, I mean, really rabid dog not to jump on the fence which bordered the sidewalk which was terrifying passersby, causing them to fall off the sidewalk into the street. I saw it happen a number of times, and that's what motivated me to basically train the dog not to run at strangers by blasting it in the face several times point blank when it did that to me. And after a couple days it just kind of peered around the side of the house to see who was there. It completely changed its behavior.

But anyway, unfortunately what we learned was that a lot of people have a problem with dogs barking. And I wish there was a solution for that, but this isn't it. You know, I don't know that there is one except to try to talk to the dog's owner. And unfortunately many dog owners who have loudly barking dogs are strangely unsympathetic to the complaints of neighbors. So I don't know. I don't have a solution for it. I wish there was one. For what it's worth, Adam, sorry.

But many of our listeners, due to this story, have occasionally sent me links to commercial devices that do this. They do exist. Unfortunately, based on all the experience of those who've built these devices, and these things really did, they were super loud, and they really did work, none of them stopped dogs from barking. So I doubt that any of the commercial devices do that. And they're probably just weak imitations of what we originally had.

MIKAH: I want to say I think it depends on the dog because I have actual experience with - so when I first moved to California back in 2019, I have two small, small dogs. One is a pure Chihuahua, the other is a Chihuahua fox terrier mix. Really small dogs. And I had in the past, when I lived in a home in Missouri, where there wasn't anyone attached directly next to me, I could leave them in the home during the day. They would hang out on the sofa, they would eat, they would drink their water, I'd come home, everything would be fine. And I was not cognizant of the fact that being attached to others in this town home meant that they would hear sounds through the walls that would frighten them, and they would bark at them.

So I would go off to work, and I didn't know this was happening, but they were barking while I was gone. The way that I found out was - I can remember the day because it cranked my anxiety up because I got an email from the townhome people, and they said, hey, we've had complaints about your dogs barking. I thought, oh, my god, I'm going to get kicked out. This is awful. What am I going to do? I ordered this little, it looks like - and it's probably what you've seen. It looks like a little tree house. And it was intended to be used outdoors, and you'd, like, hang it. And it hears the dogs bark, and it lets out a series of high-pitched [mimics sound].

But the sound that I'm making is not, whoops, the sound that they make. Much different, much higher pitched. And what it's intended to do on a sort of scientific level is to disrupt the dog's central nervous system to cause the dog to take its attention off of whatever is causing it to bark, and focus on that instead.

Steve: Yup.

MIKAH: And if that happens enough times, it will break them of the pattern of choosing to bark at whatever they're barking at. That's why some animals or some manufacturers make little devices to go around the neck, they're not shock collars, but they actually put out a spray of citronella, and the same thing happens. That spray, whenever they're barking, they suddenly sniff that, it breaks their pattern of paying attention to whatever it was they were barking at. Anyway, all of that's to say it actually did work for my small dogs.

Steve: Yay.

MIKAH: And I actually, in the place we live now, we're back in a home that's not attached to other people. But there are, on the other side of us, we have neighbors that because there's plants and stuff in the way, the dogs can't see, they can only hear. And so it has been a little frightening for them at first. And so I actually pulled those out of storage, hung them in the yards, and that has significantly reduced the barking as well.

Steve: Nice. Okay, so your mileage may vary.

MIKAH: Your mileage may vary, yeah, exactly.

Steve: Nice.

MIKAH: I think it depends, yeah. If you've got a big dog that's maybe prone to some aggressiveness and doesn't quite react as quickly to - because I think it depends also, you know, if you've got - if you're more of a prey animal than you are a predator, then those small sounds are going to draw your attention more than if you're a bigger dog, I think.

Steve: Yeah, it'll just eat the little tree house.

MIKAH: Yeah. I have tree houses like this for breakfast.

Steve: Actually I just got some feedback by email, which was written to securitynow@grc.com, while you were sharing the news about Melissa. Apparently the lesser known security bureau links, many of them are broken. Two are broken, one goes to a Wix page that doesn't go anywhere, and another one is a subsidiary of Experian, so you've already got that covered among the top four. So it looks like Craig, who is the co-founder of CyberHoot, will want to click on those links himself and get them fixed up or remove them. I mean, and if you've got something calling itself a credit bureau that's going to a Wix website...

MIKAH: Yeah.

Steve: I don't think we need to worry about freezing anything there. And you may not want to give all your personal and private information...

MIKAH: Exactly.

Steve: ...to those people either. And that was from Joey, I think it was Joey Albert who just sent me email, and I just saw it. Yeah, Joey Albert. So thank you, Joey, for that. And we got everybody informed during the same podcast.

Okay. So John wrote: "Hey, Steve. I stumbled across this very cool-looking hexadecimal clock face with ticking hands showing the time in the venerable Unix time and thought you, Leo, and the rest of the listeners would love to see it, too. Check it out at," and then he's got a URL that I could read, but it's in the show notes, and I've got something better coming up. He said: "All the best to 999 and well beyond, John."

Okay. So we've previously encountered this wonderful version of the Unix clock. Thinking that I would probably have created a GRC shortcut for it previously, sure enough, I searched for and found it, created almost exactly two years ago on September 18th of 2022. And the shortcut itself is, not surprisingly, "2038," so go to grc.sc/2038.

Unix time is represented by a 32-bit signed integer which has been incrementing once per second since midnight of January 1st, 1970. In what's known as signed two's complement format, the most significant bit of a number's binary representation is reserved for the number's positive or negative sign, with the bit set to '1,' meaning that the number is negative. Now, this works out naturally when doing two's complement binary math, which is the system used by all contemporary computers. For example, subtracting 10 from 5 should produce negative 5, and that's what happens if negative values have their high bit set. So the system works beautifully.

However, Unix time could and arguably should have been defined as an unsigned 32-bit integer since it was meant to be used for timekeeping into the future, not the past. But as it is, the result of Unix time being a signed value means that negative values represent times before 1970, extending back to 1901, which is not highly useful for things like timestamping database entries and so forth, which is what we use this for.

The good news is that all modern Unix-like systems, and even some of the Unixes themselves, well, all of the Unixes themselves, have long ago switched to 64-bit time representations. But as we always see, there are surprising corners of technology that are slow to update. So it's entirely foreseeable that there will be some breakage somewhere when we finally get to 2023, 14 years from now. I'm sorry, 2038, 14 years from now.

Okay, now, this specific clock site is very cool and very nerdy - and thus, you know, very appealing - since those 32 bits are broken into four 8-bit bytes, with each of the four bytes determining the position of each of the clock's four hands. Since each 8-bit byte can have any one of 256 values, the clock has 256 "ticks" around its face. And since trouble begins once the high byte, represented by the red hand, reaches its halfway point, because we're only able to use the positive half of all the values in a 32-bit signed integer, when that red hand is pointing straight down, something's going to break somewhere. So this graphic makes it very clear that we're well on our way toward the Unix apocalypse.

Now, I have to say I would dearly love to still be doing this podcast 14 years from now and to be able to cover and discuss the events of the end of 32-bit Unix time. I'm not sure I'll still be doing this in 14 years, but I would not be surprised if something didn't break. So we'll see what happens.

Norbert shared: "Bobiverse book number 5 came out on September 5th," so five days ago. The book is titled "Not Till We Are Lost." I just wanted to let everybody know. The Bobiverse series has been a big hit among our listeners. And so Norbert, who said "Thanks for the podcast," I will say thanks for the notification that there is now a fifth Bobiverse book.

MIKAH: The Bobiverse is unique for me in that I very rarely enjoy, and I know this is kind of weird among nerds, I very rarely enjoy science fiction epics or anything like that, science fiction books in general. I like science fiction shows. But when it comes to books, typically if I'm going to read fiction, it's going to be high fantasy or some sort of fantasy. Bobiverse really hooked me from the get-go. And again, I was surprised myself and went into it expecting that I wouldn't keep listening to it. And so it caught me off guard in really enjoying it. So I was very pleased when September 5th rolled around, and I was able to get the next one.

Steve: Oh, cool. So you already knew.

MIKAH: Yeah. Yeah, I did, because I had it in my wish list already to get it as soon as it was available.

Steve: Nice. So I wrote here in the show notes, I said the Bobiverse books are pretty easy to breeze through. But for anyone who's interested in really sinking their teeth into something that promises to be far more substantial, our listener Simon Zerafa sent me a note that one of this podcast's favorite sci-fi authors, none other than the great Peter F. Hamilton, is releasing his next novel next week. Now, that's the good news. What may be bad news, depending upon your need to achieve immediate closure, is that this is book number one of a two-part novel series. The good news is there's only two of them.

In the past, as with for example "Pandora's Star," which left us hanging quite a while for the story's conclusion in "Judas Unchained," and later it was the same with Peter's "Dreaming Void" series, which had a number of books, you know, Peter's famous for laying down a lot - and I mean really a lot - of foundation in his novels, so that things are really finally just, you know, really get moving just as the first novel ends. And it's like, oh. So, you know, that may not bother everyone, I get it, but it bugs the crap out of me. So I'm sure I am going to patiently wait for the publication of the series' second and concluding book because then I'll be able to purchase both books at once, and I'm sure I'll do that in order to read them back to back.

The first book's title is "Exodus: The Archimedes Engine." And the synopsis, probably taken from the back cover of the hardback, just to give its reader a sense for what's to come while not being a spoiler, it reads: "Forty thousand years ago, humanity fled a dying Earth. Traveling in massive arkships, these brave pioneers spread out across the galaxy to find a new home. After traveling thousands of light-years, one fleet of arkships arrived at Centauri, a dense cluster of stars with a vast array of potentially habitable planets. The survivors of Earth signaled to the remaining arkships that humanity had finally found its new home among the stars.

"Thousands of years later, the Centauri Cluster has flourished. The original settlers have evolved into advanced beings known as Celestials and divided themselves into powerful Dominions. One of the most influential is the Crown Celestials, an alliance of five great houses that controls vast areas of Centauri. As arkships continue to arrive" - right, remember they were all called by the announcement that we found a great place. So "As the arkships continue to arrive, the remaining humans and their descendants must fight for survival" - I don't know why - "against overwhelming odds" - I don't know why - "or be forced into serving the Crown Dominion."

Okay. So it sounds as though the Crown Dominion has become old and corrupt and, you know, bad. So this thing says: "Among those yearning for a better life is Finn" - who probably becomes the focus of this - "for whom Earth is not a memory but merely a footnote from humanity's ancient history. Born on one of the Crown Dominion worlds, Finn has known nothing but the repressive rule of the Celestials, though he dreams of the possibility of boundless space beyond his home. When another arkship from Earth, previously believed lost, unexpectedly arrives, Finn sees his chance to embrace a greater

destiny and become a Traveler" - with a capital T - "one of a group of brave heroes dedicated to ensuring humanity's future by journeying into the vast unknown of distant space."

Okay. So at this point this is not any sort of recommendation because I haven't read, you know, that first book. I'm certain, as I said, I'll read both of them once they both become available. So if anyone listening, I mean, it sounds like another fun Hamilton adventure. And, you know, boy, when Hamilton gets going with a series, they can really be a lot of fun. Lots of new, you know, brain-stretching tech in there. So if anyone listening does decide to jump on the first book, knowing that they may be left with a classic Hamilton cliffhanger, please DO send your review to me at securitynow@grc.com, and I'll share what you think without any spoilers.

Hadrian said: "Hi, Steve. Longtime reader, then listener, then viewer. I recently bought SpinRite. Not yet needed for recovery, but I now have a burning question. Am I the only one who looks at the raw data display and then suddenly says, 'Hey! I know which file that was!'?" So I got a kick out of Hadrian's note because, though no one else has ever specifically mentioned it that I can recall, I, too, will often see something I recognize flash past on SpinRite's Real Time Activities display. Of note is that SpinRite did not always show that.

Back before mass storage drives were able to manage their own defective sectors, SpinRite needed to and did handle all of that itself. This meant that sectors which were embedded in clusters that had been found to be defective would need to be relocated, then replaced by good clusters. So that region of SpinRite's Real Time Activities UI page once was used to track all of those changes and show totals by counts and by bytes of everything that SpinRite had done in moving things around within the file system in order to knit the file system back together after making those changes.

But at some point, once all drives became able to handle defect relocation autonomously, although SpinRite would still induce a drive to perform the needed relocation, now that would happen below the level of the file system. That meant that I was able to remove all of that logic from SpinRite. But that also meant that I needed to remove all of the tracking, totaling, and displaying of that work which SpinRite no longer needed to do. And that left a big empty display region in SpinRite's user interface. So I decided to fill that hole with an updating snapshot of the data that was passing by, so that SpinRite's user could literally see the data that SpinRite was working on. It's ended up becoming one of SpinRite's more popular user-interface features. So Hadrian, you're not alone in staring at the screen and saying, hey, I know what that was that just went by.

And I should tell you that I bought, during the work just last year on SpinRite 6.1, I realized from one of our other testers that it was possible to buy bad drives in bulk from eBay sellers. I thought, what? But I thought that was great, I need those. So I bought several boxes of bad drives. They're not very expensive because they're bad. Mostly they're just heavy, so it costs a lot to ship them. But I did that, and I'll just say drives you buy from eBay have not been wiped. And so I was - since I needed these bad drives to test SpinRite, I was doing that. And, you know, I would switch to the real-time activity screen, and I would see other people's data flashing by on the screen.

MIKAH: Other people's "data."

Steve: Uh-huh.

MIKAH: Oh, my. I've seen their "data."

Steve: Yes. So one of my announced and planned products which I will be doing, not immediately, but next to immediately, would that be, what's that next - no. Anyway, it's

a product that I've already got the trademark on it, I'm ready to go, it's called Beyond Recall. And it will be a high-speed secure drive wiping utility which you can use for USB thumb drives or spinning drives that you connect to the computer. Anyway, clearly a need to make secure data removal quick and painless, as I have a feeling it'll be very popular among this podcast's listeners.

MIKAH: Beyond Recall, not to be confused with your upcoming freeware to get rid of Recall.

Steve: Yes.

MIKAH: Okay.

Steve: And that's a very good point. I've noted that there is a collision of the name Recall and Beyond Recall. So I don't want to - I like calling something that is a secure drive wiper "Beyond Recall."

MIKAH: Yeah, makes sense.

Steve: Yeah. But so I think once you hear the name that I've got for the Recall product, everyone will agree we've got to stick with that one.

The final piece of feedback leads us nicely into this week's topic. The feedback was sent by a UK listener named Laura, who wrote: "Hi, Steve. My name is Laura, from the UK. As I have a Master's degree in Cybersecurity, I came across this article and hope you would be interested in talking about this both for me and everyone else." She said: "I love the show, and I'm so glad you're going past 999 as I have a standing appointment with you and Leo" - in this case me and Mikah - "every Tuesday evening..."

MIKAH: Wow.

Steve: "...that no one is allowed to interrupt." And then she says, in parens, "(my ex tried)."

MIKAH: Oh, no.

Steve: So don't know if that was the deal breaker with the ex, Laura, but okay. So she said: "I've included the link below." And this particular link is one to Cyber Security News, and in the URL I can see that it says "rambo-attack-air-gapped-systems." So thank you again, Laura. Oh, and she also said: "P.S.: Leo, love the new attic." So she's apparently a video watcher of the podcast.

MIKAH: Yeah, a watcher.

Steve: And I don't - what time zone in the UK, I'm not sure where that would put her. But obviously Tuesday evenings she already has the video available. So I don't know what that means relative to us making it Tuesday afternoon/evening.

MIKAH: Oh, that's going to be - because I think it's like six hours plus.

Steve: Ahead?

MIKAH: Yeah, yeah, they're definitely ahead. Back when I was in Central time it was six hours ahead of me. Now I'm not - so hold on.

Steve: So it's two or four. Or would it be...

MIKAH: So it's 11:41 p.m. right now in London. So they're plus eight hours, 11:41 p.m. So that's really late.

Steve: Yeah, yeah. Maybe that's what happened to the ex.

MIKAH: Yeah. Maybe Laura watches live, and then that way at least it's only until, like, midnight that she's watching us on the podcast.

Steve: Oh, that would make much more sense, wouldn't it. Yes, yes, yes, of course. In that case, hi, Laura!

MIKAH: Hi!

Steve: Happy to read your feedback.

MIKAH: And it is time to talk about RAMBO with Steve Gibson.

Steve: So, thank you, Mikah. Many of our listeners forwarded news to me of this latest side-channel attack brought to us by none other than another clever researcher at, you can probably guess, Israel's Ben-Gurion University of the Negev. These guys are the ones who have brought us so many bizarre ways of exfiltrating data from computers through the years that no one would be surprised that we have another one.

It was easy to see how much attention this latest bit of research drew from the security press since the many links I received from our listeners - and thank you all, by the way, for sending them. Basically you all voted for this week's topic by informing me of, like, hey, Steve, did you see this, did you see this, did you see this? Yes. They were from widespread security-related publications. Before I dug into what it was all about, I was hoping that the reason for all the attention was not only because the new attack was named "RAMBO," and I was not disappointed. So I decided that RAMBO should be this week's main discussion topic.

And also, everyone knows that I have a difficult time ignoring access to raw research. The worst case is having to decipher something that some public relations person wrote that doesn't contain any of the really good nitty-gritty. But in this case we have 18 pages of pure delicious research written by the researcher himself, Mordechai Guri, which explains his new attack in detail.

So the Abstract of Mordechai's research says: "Air-gapped systems are physically separated from external networks, including the Internet. This isolation is achieved by keeping the air-gap computers disconnected from wired or wireless networks, preventing direct or remote communication with other devices or networks. Air-gap measures may be used in sensitive environments where security and isolation are critical to prevent private and confidential information leakage.

"In this paper, we present an attack allowing adversaries to leak information from air-gapped computers. We show that malware on a compromised computer can generate radio signals from memory buses." Thus RAM and RAMBO. "Using software-generated radio signals, malware can encode sensitive information such as files, images, keylogging, biometric information, and encryption keys. With software-defined radio hardware and a simple off-the-shelf antenna, an attacker can intercept transmitted radio signals from a distance. The signals can then be decoded and translated back into binary information. We discuss the design and implementation and present related work and evaluation results. This paper presents fast modification methods to leak data from air-gapped computers at 1,000 bits per second. Finally, we propose countermeasures to mitigate this out-of-band air-gap threat."

Okay, now, the first thing I'll note is that while 1,000 bits per second will not allow you to send Windows, or even a Windows update over the air, a modern state-of-the-art cryptographic private key is only several kilobits in length, so the keys to the kingdom could be broadcast from just such a compromised machine, over and over, every few seconds. And since it would just appear as random RF noise, no one would ever be the wiser.

And unlike most malicious code whose purpose is readily revealed through inspection, any code that's being used to generate radio signals from memory buses will just be puzzling for any forensics researchers. They'd stare at it and scratch their heads and never have any idea what the heck such code was doing. They couldn't even be certain it was doing anything that was malicious. It wouldn't appear to be doing anything at all since the designers of this code are using a far-fetched side channel of normal data processing to get their message out of the machine.

The second thing to note is that one of the consequences of today's heavy use of encryption is that we've grown to rely upon it completely. What this means practically is that today we're far less worried about storing our sensitive encrypted data in far more accessible places, such as in the ubiquitous cloud. It's "Who cares if it's in the cloud. It's encrypted; right?" Sure thing. That's true. Right up until the time someone figures out how to exfiltrate the comparatively tiny secret key that's protecting the otherwise far less secure data.

So my point is, thanks to the application of cryptography virtually everywhere today, we now concentrate vastly more value into a handful of bits. So whereas 1,000 bits per second cannot be used to transfer a massive database, those few thousand bits are the secret that's protecting a massive database in the cloud. Then a few seconds worth of transmission is all that's needed to crack that database wide open.

Reminding us that air-gapping and air-gap exploits have a significant and deep history, Mordechai explains. He writes:

"Enforcing an air gap in a computing or networking environment involves physically and logically isolating a system, network, or device from external networks or communication channels. This can be done by disconnecting network cables, disabling wireless interfaces, and disallowing USB connections. In addition, it must be ensured that the isolated system has no direct link to any external communications infrastructure.

"Despite air-gapped networks being considered highly secure, there have been incidents demonstrating that air-gapped networks are not immune to breaches. Stuxnet is one of the most famous air-gapped malware. Discovered back in 2010, Stuxnet was a highly sophisticated worm that targeted industrial control systems, particularly those used in nuclear facilities. It exploited zero-day vulnerabilities and used several methods, including infected USB drives, to jump the air gap and spread across isolated networks.

"The Agent.BTZ worm was another type of air-gap computer worm with advanced capabilities and targeted type. It was specifically designed to spread through removable media, such as USB flash drives, and infiltrate computer networks, including those highly secure or air-gapped. According to reports, that worm affected the U.S. Department of Defense classified networks." Which is not easy to get into. It did. He said: "Notably, more than 25 reported malware in the past targeted highly secured and air-gapped networks, including USBStealer, Agent.BTZ, Stuxnet, Fanny, MiniFlame, Flame, Gauss, ProjectSauron, EZCheese, Emotional Simian, USB Thief, USBferry, Retro, and Ramsay." So plenty of them.

And then Mordechai discusses his new air-gapped attack. He writes: "In order to exfiltrate information from an infected air-gapped computer, attackers use special

communication channels known as air-gap covert channels. There are several types of covert channels studied in the past 20 years. These attacks leak data through electromagnetic emission, optical signals" - like LEDs blinking - "acoustic noise" - like changing the noise of the fan of the computer - "thermal changes" - actually like ramping the CPU up, which generates more heat. Now, those are extremely low data rate changes, but they are changes. "And," he says, "even physical vibrations. In this paper, we show how malware can manipulate RAM to generate radio signals at clock frequencies. These signals are modified and encoded in a particular encoding allowing them to be received from a distance away." And I'm actually going to focus on the encoding because that's the really cool part of this.

He says: "The attacker can encode sensitive information - keylogging, documents, images, biometric information, et cetera, and specifically secret keys - and exfiltrate it via these radio signals. An attacker with appropriate hardware can receive the electromagnetic signals, demodulate and decode the data, and retrieve the exfiltrated information.

Attacks on air-gapped networks involve multiphase strategies to breach isolated systems by delivering specialized malware through physical media or insider agents, initiating malware execution, propagating within the network, exfiltrating data using covert channels or compromised removable media, establishing remote command and control, evading detection, and covering tracks."

And finally: "In the context of the RAMBO attack," he says, "the adversary must infect the air-gap network in the initial phase. This can be done via a variety of attack vectors. An attacker could plant malware on a USB drive and physically introduce it into an air-gapped network. An unsuspecting insider or employee might connect the USB drive to a computer within the isolated network, unknowingly activating the malware and allowing it to propagate and exfiltrate data through the same USB drive or via covert channels.

An insider with access to the air-gapped network might intentionally introduce malware or provide unauthorized access to external parties. This could involve transferring sensitive data to personal devices using covert communication methods like steganography to hide data within innocent-looking files. An attacker could also compromise hardware components or software updates during the supply chain process."

I'll interrupt to note that the particular power of this attack is the degree to which its effects would be unsuspected and undetected. I mean, like, the computer's working fine; right? And it's not connected to anything. Nothing is going out. So an adversary, for example, might introduce their RAMBO-enabled malware into a device driver that's known to be used and needed by the targeted system. Since no one would ever imagine that a device driver update could suddenly turn a PC into a covert short-range transmitter, the updated drivers might be delivered as part of a very careful and very clean offline CD or DVD carried update. In other words, you can't infect a CD or DVD. And that's all that would be required.

Mordechai continues: "Once these components are installed within the air-gapped network, hidden malware might activate and communicate with external parties. Note that APTs (Advanced Persistent Threats) in the past have targeted highly secured and air-gapped networks. Recently, in August of 2023, researchers at Kaspersky discovered another new malware and attributed it to the cyber-espionage group APT31, which targets air-gapped and isolated networks via infected USB drives." So that's still going on.

"In the second phase of the attack, the attacker collects information - keylogging files, other files, passwords, biometric data and so on - and exfiltrates it via the air-gap covert channel. In our case," he writes, "the malware utilizes electromagnetic emissions from

the RAM to modulate the information and transmit it outward. A remote attacker with a radio receiver and antenna can receive the information, demodulate it, decode it into its original binary or textual representation."

Okay. For the actual generation of the RAMBO RF signals, he explains: "When data is transferred through a RAM bus, it involves rapid voltage and current changes, mainly in the data bus. These voltage transitions create electromagnetic fields, which can radiate electromagnetic energy through electromagnetic interference (EMI) or radio frequency interference (RFI). The radio frequency range of electromagnetic emanation from the RAM bus mainly depends on its specific clock speed, measured in megahertz or gigahertz. This clock indicates how quickly data can be transferred between the CPU and memory. The emanation levels are influenced by other bus characteristics, including its data width, clock speed, and overall architecture. Faster RAM buses such as DDR4 and DDR5 having wider data paths can lead to quicker data transfers with increased emissions."

He said: "When data is read from or written to memory, electrical currents flow through the RAM chips and the associated traces on the printed circuit board. These electrical currents generate electromagnetic fields as a byproduct, which radiates electromagnetic energy. To create an electromagnetic covert channel, the transmitter needs to modulate memory access patterns in a way that corresponds to binary data. For instance, they could alter the timing or frequency of memory access operations to encode information.

"The sender and receiver must establish rules that define how memory access patterns translate to binary values. For example, reading or writing an array to the physical memory at a specific timing interval might represent a zero, while another interval represents a one. The receiver detects and decodes the EM emissions caused by the modulated memory activity. This could involve sensitive radio frequency receivers or electromagnetic field sensors."

Okay. So he says: "One algorithm used OOK" - which is his abbreviation for On-Off Keying modulation, he says - "a basic form of digital modulation used in communication systems to transmit data over a carrier wave. In our case, the OOK modulation involves turning the carrier wave on and off to represent binary data, where the presence of the carrier wave generated by memory activity corresponds to one binary state ('1'). The absence of the electromagnetic carrier wave corresponds to another binary state ('0').

"Note that to maintain the activity in the RAM buses, we used the MOVNTI instruction" - which is an Intel instruction - "which stands for Move Non-Temporal Integer. It performs a non-temporal store of integer data from a source operand to a destination memory location. This instruction is primarily associated with optimizing memory operations for certain types of data transfers, particularly in cases where the data is not to be reused immediately. For the beginning of the transmission, we used the preamble sequence 01010101, allowing the receiver to be synchronized with the transmitter."

And I'll just interject to say that essentially what they discovered was that that particular instruction, MOVNTI, generates the most noise. They found a specific Intel instruction that was a noisy instruction in terms of the amount of radiation it produced.

And he finishes, saying: "For the fast transmission, we used Manchester encoding. In this encoding, each bit of the binary data is represented by a transition or change in signal level within a fixed period. Manchester encoding ensures a constant number of signal transitions, making it useful for clock synchronization and error detection."

Okay. Now, I smiled when I saw that Mordechai had chosen to use Manchester encoded signaling since it's extremely simple and straightforward, and it's likely the best solution for his need. Manchester encoding is still in wide use today due to its simplicity and

robustness, though it dates back to 1948 where it was invented and first used to store and retrieve data on the magnetic storage drum for the University of Manchester's Mark 1 digital computer. So this thing's been around for a while.

Because Manchester encoding provides such an elegant solution to a common problem, so common in fact, as I noted, it's still being used today - for example, it's the way our consumer IR remote controls send their data to our television sets and stereos. And even RFID tags use Manchester encoding. And since it provides another interesting dip into pure communications engineering and abstract computer science, I want to take some time to explain how it works.

The problem Manchester encoding beautifully solves is known as "clocking." If you have a single-bit channel, as with RAMBO, where we have a radio signal that's either on or off, or a wire that's either carrying a current or not, you know, the current's being switched on and off, or a remote control's infrared LED that's either on or off, a significant problem arises when a long series of some number of ones or zeroes occurs in a sequence because the question quickly becomes, exactly how many zeroes or ones was that?

If some time passes without anything happening, for example, the radio is off or on for a while, how is the receiver to know precisely how many bits were just transmitted? If the sender and the receiver both had perfectly and exactly equal knowledge of time passage, it would theoretically be possible to just count the elapsed time between a change from On to Off or Off to On, then divide that time by the time per bit to determine exactly how many "bit times" had elapsed. The guys at Manchester may have initially tried that back in 1948. But in their case, slight variations in the speed of their drum storage rotation rate would have quickly shown them that they needed some system that would be far more tolerant of slight timing variations. And even today, two clocks are never precisely synchronized, nor are they ever running at precisely the same rate.

Communications designers have solved this problem by creating systems known as self-clocking encoding. Self-clocking encoding systems ensure that something always happens often enough for the receiving end to stay synchronized with the sending end, even if their timing is not precise. And Manchester encoding, first used 76 years ago, does exactly that. Here's how it works: The key to understanding any encoding is to recognize that the signal is no longer the data. Mordechai mentioned simple on-off keying which is an unencoded system. With simple on-off keying, the signal IS the data. But that's where we run into trouble if many zeroes or ones are sent in an uninterrupted sequence. So any encoding that we employ breaks this simple relationship between the signal and the data that the signal is intending to send.

Okay. To talk about this, we'll refer to the signal as being "low" or "high," whereas the data bits are a zero or a one. So "low" or "high" in the case of RAMBO would mean the RAM transmission is either on or off. The transmitter is sending or it's not. A one data bit is encoded as a low followed by a high, whereas a zero bit is encoded as a high followed by a low. In other words, RAMBO transmits a one bit by having its RAM transmitter first not transmitting anything, then having it transmit. And it sends a zero by having its RAM transmitter first transmitting a signal, then switching it off and not sending any signal.

The best way to think of this is that in Manchester encoding a one bit is encoded as a transition from low to high, whereas a zero bit is encoded as a transition from high to low. This means that a so-called "bit cell" - which is the period of a single data bit - always contains two opposite states, both a low and a high, and the direction of the transition between those two states is the bit cell's data, a zero or a one. If the bit cell contains a transition from low to high, radio off to radio on, that's RAMBO sending a one. And if the bit cell contains a transition from high to low, that's RAMBO sending a zero.

Okay, now, if you think about this for a second, you'll see we have a problem. In order to send a pair of ones, we need to have back-to-back low-to-high transitions, in other words, RAMBO radio off to RAMBO radio on transitions. But at the end of that first bit the radio will already be on, and the next one we're sending requires the radio to start by being off. We solve this problem by completely ignoring any inter-bit cell transitions. In other words, only the transitions occurring in the middle of a bit cell carries any data. The transitions occurring in between are ignored.

Okay. So now, assuming that you've been following along, you're wondering how the receiver can tell the difference between the data transitions occurring in the middle of the bit cells and the transitions we're supposed to ignore which may or may not occur in between the bit cells in order to get ready for the next bit. Manchester encoding provides the answer because every bit cell must always contain a transition, whereas there may or may not be a transition in between two bit cells.

Now, if you doodle with a piece of paper, with a pencil and paper for a bit, you'll quickly see that any receiver can perfectly "lock onto" the location of the bit cells the very first time a transition is missing, since that can only be the period in between bit cells. That means that the next transition must be in the exact center of a bit cell as far as the transmitter is concerned. So if the receiver knows only approximately the rate at which the transmitter is sending bits, that's now sufficient to allow it to judge whether an inter-cell transition opportunity has passed, and when the next guaranteed-to-always-be-present transition occurs. And when that happens, the receiver updates its self-clocking "lock," which prepares it to judge whether the next transition occurs quickly, meaning that it's an inter-cell transition, or not until it's expecting the next data bit transition.

This simple system works so well that it was used in the earliest Ethernet physical layer standards. And as I mentioned earlier, it's still used today by consumer home entertainment infrared remote controls, as well as RFID and near-field communications.

Mordechai had considered both simple on-off keying and Manchester encoding. He wrote: "Our analysis shows that the Manchester encoding is more relevant for the requirements of the RAMBO covert channel due to two main reasons: One, the encoding aids in clock synchronization between the sender and receiver; and, two, the frequent transitions make it easier to detect errors caused by signal loss, interference, or distortion. However, it's important to note that Manchester encoding doubles the required bandwidth compared to direct on-off binary encoding, as each bit requires two signal states within the bit interval."

Okay, so how did all this turn out? "Keylogging," he wrote, "can be exfiltrated in real-time since UNICODE is only 16 bits per keystroke. A 4096-bit RSA encryption key" - you know, the keys to the kingdom that you don't ever want to get loose - "can be exfiltrated in 4.096 seconds, and biometric information and small files such as .JPGs and small documents require a few seconds at the system's fast speeds." They conclude: "This indicates that the RAMBO covert channel can be used to leak relatively brief information over a short period." And they were also able to receive this information - get this - at a distance of up to 700 centimeters. For those of us who grew up using the Imperial system of measurement, as I did, 700 centimeters is 23 feet. So this is useful and impressive.

And for those who remember the days of using the Pringle can to get increased WiFi range, I would imagine that if you got yourself a well-tuned and well-aimed Pringles can, you might be able to significantly improve on that performance distance. For at least this first round of research, they seemed less focused upon distance than feasibility. They've certainly shown that their RAMBO system is feasible.

It's been known for a long time that electronic devices generate and radiate electromagnetic interference while they're in use. The somewhat strained acronym TEMPEST stands for "Telecommunications Electronics Materials Protected from Emanating Spurious Transmissions." So TEMPEST-hardened devices are those which incorporate specific countermeasures designed to block or mask any useful information-carrying emanations from electronic equipment.

We can hope that any air-gapped machines which have been deliberately disconnected from any traditional form of data communications will have also been shielded so that none of the noise generated by the system's motherboard is able to find its way into the surrounding environment. It would be necessary, of course, to first infect that machine with RAMBO technology malware. But if that could be accomplished, any otherwise unprotected machine could be turned into a RAMBO transmitter.

MIKAH: Wow. This is what I have to say about all of this. I was able to follow along with every bit of this, and that is what I really appreciate is that I actually got what was going on here and was able to piece it together. So I always appreciate that about what you do. Because as much as sometimes this security research can seem to go over one's head, you really did a great job with explaining what is going on here. And I just - I think it's amazing.

We talked earlier about the person who made the fence, right, and wanting to get into the heads of the person or the people who made that fence with the hole in it shaped like an apple. I want to know what was going on that they were able to - what was it, Mordechai? - yeah, Mordechai was able to even come up with this idea. Was he sitting there, was Mordechai sitting there and was like, I wonder if I keep this antenna next to the device, and I'm looking at the RF signals coming by, and then you isolate those RF signals. Or if it was like, I'm thinking about RAM one day, and then I think, maybe we could do something. It's just so cool just to conceive of this stuff. It's amazing.

Steve: Yeah, yeah, yeah. It is really fun.