Black Hat and DEF CON

Description: This week, as expected, we look at some of the events and announcements from last week's Black Hat and DEF CON conference events. Microsoft and Apple have upped the ante for bug hunters, the Chaos Computer Club shreds a hotel's door lock security, a serious philosophical design flaw is revealed to be present in 40 signed device drivers, and Google vows to continue its Incognito-mode battle. We also have some SQRL news, some fun miscellany, and some interesting closing-the-loop feedback from our terrific listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-727.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-727-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. I'm Jason Howell, filling in for Leo this week. Steve's going to dive deep on a number of topics that arose from both the Black Hat and DEF CON conferences. Also Incognito and publisher sites are still playing that cat-and-mouse game. Apple's offering an insane payout for its bug bounty program. That and a whole lot more coming up next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 727, recorded Tuesday, August 13th, 2019: Black Hat and DEF CON.

It's time for Security Now!. I'm Jason Howell. But you don't want to hear from me - I'm filling in for Leo - because you want to hear from the man, the myth, the legend, Steve Gibson. How are you doing, Steve?

Steve Gibson: Jason, thank you for standing in for Leo. He's off to - is it Orlando? - Florida for a podcast conference.

JASON: Podcast Movement, all week, yeah.

Steve: I don't know that I like the idea of calling it the Podcast Movement. But anyway.

JASON: Yeah, that's the name of it, so...

Steve: I guess that's what they decided to call it.

JASON: So that's what they're calling it, so we'll call it that.

Steve: I hope he's having a good movement wherever he is. This is Security Now! Episode 727 for August 13th. Lucky Tuesday the 13th. This happens also to be the

second Tuesday of August, so it's Patch Tuesday. But we have no news about that whatsoever.

JASON: Oh, no.

Steve: We were expecting last week that we would be talking about Black Hat and DEF CON, which concluded this past weekend, and that is indeed the case. So we're going to take a look at some of the events and announcements from last week's paired Black Hat and DEF CON conferences. Before we began recording we were going over whether it was one word or two and how we should capitalize the second half, and we reached no conclusion whatsoever, John chiming in that it's a puzzle.

JASON: It's a real confusing situation, although some people in the chatroom were like, who cares? But I care. I want to know.

Steve: Okay, thank you very much. Yes, well, we do. It's because we care that software doesn't crash more often than it does.

JASON: That's true. Very good point.

Steve: Details matter. Microsoft and Apple announced that they are upping the ante, in some cases quite significantly, for bug hunters, which is interesting. We have the Chaos Computer Club, which is the group in Germany that are hacking things. They shredded a hotel's door lock security, embarrassingly. That also was announced at DEF CON. We have a serious philosophical design flaw which was revealed to be present in 40 signed device drivers. And actually, one of the best-named blog postings I think I've ever seen we'll be getting to.

Google's vowing to continue its Incognito mode battle with sites that are trying to detect Incognito mode. We also have a bit of SQRL news, some fun miscellany, and some interesting closing-the-loop feedback from our terrific listeners. So I think for podcast number 727, post-Black Hat and DEF CON, another interesting conversation.

JASON: Legendary, if I could be so bold. Possibly one of your best episodes. Mind you, we are only beginning to record it right now. But we were talking before the show how Black Hat and DEF CON really, you know, it's like Disneyland. It's like Christmas for security. So for this show, your added fuel.

Steve: And leave your tech at home when you go because - in fact, I think one of the favorite things that we showed last month - or, sorry, last year - was there were two maps had been made of before Black Hat and DEF CON and mid-Black Hat and DEF CON, the number of cell sites in that area of Las Vegas. And there were, like, 15, like the week before the conference, and 50 during the conference. So it's like, uh, okay, wait a minute.

JASON: Somehow people are brave enough to go out there and to stay safe. And that's just one of those conferences that, I mean, who knows? Maybe somebody I'll be sent out there for whatever reason. But I will be frightened if that is the case. And I will probably want to leave my phone at home.

Steve: Bring a large roll of tinfoil, Jason.

JASON: Oh, okay. That's all you have to do.

Steve: Just wrap yourself, and then...

JASON: That's not weird at all, Steve.

Steve: No.

JASON: No, actually, that's normal. That's how I ride to work every day.

Steve: I was going to say, especially there, they would think that was funny. If you poke out some little holes for your eyes, you know, that'd be like, okay.

JASON: There you go. It's kind of like Comic-Con at that point. What is this strange piece of technology I'm staring at, Steve? I don't understand.

Steve: Yes. Our Picture of the Week, a listener sent this to me because for the last few weeks, well, actually, yeah, at least, we've spent a lot of time talking about the ransomware which has been encrypting systems. And last week we spent some time talking about - I was, like, trying to think about how a municipality could explain to its employees that they would really be inconvenienced if they clicked on a link in email which allowed something malicious to crawl into the municipality's network and encrypt all the computers because it was really no fun to have to use a pencil and paper like we used to in the old days.

And in fact we talked about how it was in Georgia that the State Police were now having to write tickets the old-fashioned way, with a pad and pen, rather than being able to just pull it up on a pad. And if they wanted to run someone's license plate, they'd have to actually pick up a microphone and talk into a radio to central dispatch and read the license plate in and get the results back because literally all of that technology was down in the state of Georgia.

So what we have today for our Picture of the Week, for those who are not looking at the video feed, we have a yellow piece of paper that reads at the top: "Computer Problems? NO PROBLEM! Use this manual data entry device." And then we have the abbreviation P.E.N.C.I.L., which of course spells "pencil." And this is "Personal Emergency Non-Computerized Information Lifesaver." Then there's a pencil, a physical, yellow, good old-style yellow pencil has been glued to it. And we have the opposite ends labeled. The pointy end has an arrow pointing to it labeled "ENTER." And of course the end with the pink eraser has an arrow pointing to it labeled "DELETE."

So, yes, and this is what some regions of Georgia are currently using as a consequence of the fact that one of their employees just couldn't resist opening that email and finding out what it was that they were being threatened with or promised or who knows what scam being perpetrated. But anyway, thank you to our listeners for sending that.

JASON: I appreciate the simplicity of this user manual. It gets to the point. There's no question. Enter. Delete. It makes sense to me.

Steve: There is some need...

JASON: At least they didn't have to run the information back. At least they could call it in; right? They didn't have to, like...

Steve: That's true.

JASON: It could have been worse.

Steve: Now, once upon a time it was necessary for us to keep our, well, in modern technology we have to keep our devices recharged. So I suppose the equivalent here is

needing to resharpen the pencil from time to time as it becomes dull and also shorter. And of course we also know, if we're going to really extend this painful analogy any further, that the more you use lithium polymer batteries, which our devices have, the less remaining lifetime they have. Similarly, the more you use this emergency data entry device, the shorter it becomes, thus similarly reducing its remaining available life.

JASON: Thankfully, buying more of this technology is not difficult. The technology has come down in price over the years, so it's pretty easy to stock up on it.

Steve: That is true. That is true. And you know, we really don't trust the battery charge remaining meters in our devices. There have been scams perpetrated on how much life is actually left in the battery. In this instance, with the pencil, again, as I said, really stretching this analogy past its breaking point, there is no doubt about how much life you have left remaining in that particular stylus.

JASON: Yes. And if you happen to snap it in half, it's not going to explode, so that's a positive, as well.

Steve: Yes, that's a very good point. You want to be careful.

JASON: Right.

Steve: Conferences have started, I guess historically they always begin with some major kickoff keynote speech. And that was the case with this year's 23rd annual Black Hat Las Vegas speech, which was delivered by Dino Dai Zovi. He's the mobile security lead at Square. And of course Square has become an important player in our info ecosystem. Dino titled his keynote "Start with Yes." And in it he discussed the ongoing transformation of, I guess it would be, security's role in the workplace.

I mean, when you think about it, we used to have a COO, the Chief Operations Officer, and a CEO, the Chief Executive Office, you know, the so-called "C-Suite" people. Well, there was never a CIO, but today we all have CIOs, Chief Information Officers. So as a consequence of the fact that, first of all, information has become a commodity, and then on top of that the security of that information has not only, I mean, everyone always gave it lip service. Oh, yeah, we've got to have secured information. Well, now it's like a big deal.

So Threatpost wrote up a very nice and succinct summary of the high points from Dino's commentary, which I want to share with our listeners since, first of all, it's a great introduction to this year's Black Hat; but also I think that Dino touches on something very important, which is that there is this, you know, we've often talked on the podcast about the tension that exists between the security guys, who the executives sort of, I do want to say, I guess, they sort of tend to look down their nose at them. Like security is still seen as a necessary evil.

It isn't, you know, it's sort of like healthcare. I mean, it's not obvious that it proactively provides benefit the way marketing does. It's something you have to have because there are hackers. There are bad guys. And there are, unfortunately, there are turncoat employees who could get you from the inside. And so, and of course, the security people tend to speak their own language. It's like, wait, did he just - would somebody repeat that to me in English, please? I don't know what that geek just said to me about why we're not giving them enough money.

So, you know, the point is that's been the way things are. And so Dino titles his keynote "Start with Yes." Whereas it's so often the case, I think, that security people start with no. They're like, of course, that's not the right way to deal with the CEO, who doesn't want you around in the first place and is trying, you know, trying to sleep at night

worried that there's going to be a problem that is going to bite his organization that he'll ultimately be held responsible for.

So Dino says, and this is the summary thanks to Threatpost: "Taking as a first principle the idea that security teams now have the ear of company boards and the C-Suite, the challenge becomes figuring out how to communicate most effectively within this newly collaborative environment, and how to have the most impact organizationally." In other words, Dino is saying this is reality. This has happened. Let's succeed, and let's not try to do it by forcing our will upon those pencil necks. Let's start with yes.

So: "One of the first things to do is to realize that in today's software-centric world, where internal teams rely on software-as-a-service and the cloud for core missions, and where DevOps is becoming the norm, security must become a shared responsibility and resource. Thus, listening to the 'asks' from different division leaders in terms of building security processes that don't cause friction is in many ways Job One for dedicated security personnel." Again, let's not run off in a huff. Let's figure out how to make this sort of inherently fraught relationship work. Dino said: "Saying 'yes' keeps the conversation going, keeps it collaborative and constructive, and opens the door for real change and real impact."

"Digging beyond this umbrella idea," writes Threatpost, "Dino highlighted three transformational principles for boosting the impact of security within organizations. First, work backward from the job to be done. Second, seek and apply leverage, develop feedback loops, and scale with software automation. And, third, understand that culture trumps strategy and tactics every time."

So taking those three, looking in more detail at those. Regarding working backwards from the job, which he highlighted as the first principle: "Job theory says that a job is both a function, but also represents emotional context. He used the example of milkshake research at McDonald's. Market researchers found that most milkshakes were sold before 8:30 in the morning, often as the only item, via the drive-through. In asking buyers what their motivation was, it turns out that they wanted something that would be easy to consume and would occupy a long commute."

So Dino said: "So the job the milkshake was doing wasn't solving hunger, it's alleviating boredom on a long commute. Similarly, security teams need to determine what the job is to be done. Talk to internal teams. Try to understand their struggles. Listen. What are they setting out to do? What adds friction, and what makes things easier? When and why do they interact with security? In understanding what their 'hiring' criteria is for a security solution, as well as the 'firing' criteria, it becomes possible to build in an agile way for the need at hand, rather than spending time overlaying security principles that may or may not be useful, adopted, or practical." In other words, fit security into what's really going on, rather than just make declarations.

And he says, under seeking and applying leverage, Dino touched on the idea of how to have the most impact with limited resources: "Taking Archimedes's classic idea of using a lever as a force multiplier to lift an object much heavier than oneself, it's possible to see this play out in security, with automation as the lever. For example, security is still a small community, and the problems that we tackle can be huge, using fuzzing for finding vulnerabilities as an example of scaling security's effectiveness via automation." He says: "We must work smarter, not just harder, through better software and by applying automation where we can."

Stressing the importance of having automatic feedback loops, Dino said: "We have to build them explicitly, and the tighter feedback loop wins. We have to build security services for observability, so you can understand if the protections are working and also

perform anomaly detection. We have to be able to identify attackers when they're probing, learning, attacking, and succeeding."

And finally, as for culture, strategy, and tactics: "Culture is, of course, the term for what companies value and promote, and how its employees interact and communicate. Without a culture shift towards embracing security, the technical aspects will fail despite any best-laid plans. Dino explained: 'We in security are not outsiders anymore; we're inside communities and companies. We need to use that access to improve things.'

"He noted that making security a shared issue can go a long way to creating a safer organization. For instance, he said, at Square, security engineers have to write code just like everyone else. He says this is a cultural change. There's a lot more collaboration and empathy for how people are operating. A software engineering team would write security features, then actively go to the security team to talk about it and for advice. We want to generate generative cultures where risk and involvement is shared. It's everyone's concern. If you build security responsibility into every team, you can scale much more powerfully than if security is only the security staff's responsibility."

Which I think is absolutely the way it needs to go. I mean, that makes sense. It can no longer be something treated as an afterthought, added later, and then, like, with a lot of finger-pointing going on when something goes wrong. This involves implementing a blame-free post-mortem process when it comes to responding to an anomaly or a vulnerability report. Dino said: "Turn these events into inquiries where you focus on getting at the root causes of the problems. In the end, security teams should see themselves as an extension of internal teams, not a separate division apart." He said: "Instead of saying no, start with yes, and here's how we can help. It's all about cultivating empathy. It's something you practice and grow. This is the way we meet the challenge of leveling up on security."

So I think his points were very well taken. I thank Threatpost for providing their summary of it. And of course it was with that spirit that this year's Black Hat took off. And I think that's exactly the right position. We are, over the last couple years, really seeing an evolution in security's role. It can no longer be something freestanding and separate. It's got to be something that everybody's aware of.

JASON: Importance raising, although we continue to hear about IoT, all the IoT insecurities and these smaller companies that are creating these products but never consider the security implications of them. And it sounds like what he's saying is, when we're creating these products or creating these companies around these products, security should be at the start of that discussion. I mean, do you think that we're heading in that direction? Feels like some places that hits it on the nose, and in other places that seems like a lot to ask. Just for some companies that's never going to be top of their mind. It's all about money more than it is about security.

Steve: I think that's right. I think that security is, for example, like quality. You will have companies that produce a quality product. You will always have companies that produce crap. You know, they just don't care about the quality of whatever it is they're doing. They're operating on the principle of, well, some people will purchase it. And since we're not spending any money to create this crap, we can squeak by and make a living just because some people will buy it.

JASON: Right, for as long as we can get away with it.

Steve: Yes. And so if you think of security that way, that is, as something that a company can produce, a company can produce security. It can produce quality security. Apple probably is a great example of that. We know that Google is involved. We know that Android is continually improving its security. So there are organizations where

security is necessarily their product. It's, yeah, because of what it is they're offering, security and high-quality security is integral in their offering.

But similarly, even though there are companies that should have high-quality security, just as it would be nice if they had a high-quality product which had that high-quality security bound into it, there are companies that are just going to produce crap. And their security is going to be crap. So that's the way it's going to be. The good news is that it used to not be the case that security was regarded as the product, that is, that a company was producing security. And as we're about to be talking about with the, in some cases, breathtaking strides that have been announced in this year's Black Hat and DEF CON by some of the major players who recognize that security is their product, they really are making moves in that direction. So, yeah. Not everybody's going to have it.

Now, there are things we can do that we've talked about on the podcast. Making security easier will go a long way to improving the security of things. It's not exactly on point, but for example the idea that we now have the ACME protocol which Let's Encrypt was the first deployer of, and other certificate providers are beginning to deploy their own ACME protocol offerings, where a server can obtain a free TLS certificate just by running an ACME client on its platform and automatically receive certificates, well, that made security easy. It made it simple, it made it free, and it made it easy. As a consequence, there are a ton of ACME-based certificates now in the world and will always be. So there's a good example of dropping the friction, making it easy, making it free. And so it's like, okay, why not have it? Because there's no reason not to now.

Similarly, we're beginning to see security packages targeted at IoT vendors where they're drop-in solutions that solve the problems that the IoT vendors weren't even bothering with because they didn't want to spend any money on it. Well, if they're available, and they're on GitHub, and it's free and easy, why not have it? So I do think that there will always be crap security. But to the degree that, you know, as the available means of adding security improve, I think that we'll see the companies operating in the margins or on the border, where they're willing to expend a little bit of effort, above none, in order to secure their stuff, if we make it easy for them to do that. If there are players in the industry who decide we're going to lower the bar for the good of all, then those companies can increase security, and it'll only be the real deadbeats that just don't bother.

JASON: Right, right. Yeah, that makes a lot of sense. Microsoft's not a deadbeat, though.

Steve: No.

JASON: Heck, no.

Steve: So they have dangled a $300,000 bounty, which they announced last week at Black Hat, for the discoverers, payable to the discoverers of unknown bugs lurking in their Azure cloud platform. Which is, you know, a nice piece of change. Microsoft launched a dedicated Azure cloud host testing environment which they call Azure Security Lab, ASL. This ASL program allows security researchers to test attacks on this so-called "infrastructure as a service," IAAS scenarios, without impacting customers.

So these hosts, the hosts that Microsoft will be making available, will be isolated from the mainstream Azure production environments which customers use, allowing researchers to attack this ASL, the Azure Security Lab infrastructure, as much as they want to, testing live exploits without fear of bringing down or adversely affecting other Azure clients. Microsoft's Kymberlee Price, who is the principal security program manager for the Microsoft Community and Partner Engagement Programs, wrote in a blog post about this last week.

She said: "The isolation of the Azure Security Lab allows us to offer something new. Researchers can not only research vulnerabilities in Azure, they can attempt to exploit them without fear." She said: "To make it easier for security researchers to confidently and aggressively test Azure, we are inviting a select group of talented individuals to come and do their worst to emulate criminal hackers in a customer-safe cloud environment which we're calling Azure Secure Lab."

Researchers with access to Azure Secure Lab may also attempt scenario-based challenges, meaning I guess Microsoft provides the scenarios and the researchers try to attack them, with top awards of $300,000. Starting yesterday, that is, August 12th yesterday, researchers can apply for this access at Microsoft's website. And on top of the ASL announcement, Microsoft announced that it is doubling its bug bounty rewards for researchers who discover Azure vulnerabilities. At the start of this year, Microsoft announced a bug bounty program designed to find flaws in Azure DevOps, with top rewards of up to 20,000. Now that 20,000 has been doubled to 40,000.

For those who aren't aware, Azure DevOps is a cloud service launched in 2018 that enables collaboration on code development across the breadth of a development lifecycle. The two in-scope services for the bounty program include Azure DevOps Services, which was formerly known as Visual Studio Team Services - I guess I'm going to have to update myself because that's how I still think of it. So now it's Azure DevOps Services - and the latest publicly available version of Azure DevOps Server and Team Foundation Server.

Microsoft has quietly been paying out some nice hefty rewards. Microsoft announced at the conference that it has paid out, get this, a total of $4.4 million in bounty rewards during just the past 12 months across its various programs. So yup, there's some money there. And actually we'll be talking a little bit more about that a little bit later because they also announced the top 75 researchers.

Last month Microsoft initiated a bug bounty program, offering payouts as high as $100,000 for the discovery of holes in identity services and implementations of the OpenID standard. These include Microsoft Account and Azure Active Directory, which offer identity and access capabilities for both consumer and enterprise applications, as well as its OpenID authentication protocol implementation. And, of course, for a company with the cash and resources of Microsoft, spending their money like this really does make sense because why not incentivize people to tell Microsoft rather than to tell, like, sell these things on the dark.

And also, back last March, inspired by the Meltdown and Spectre flaws - we talked about this at the time - Microsoft started another new bug bounty program targeting speculative execution side-channel vulnerabilities. And at the time when we mentioned this, I remember it was interesting, and I'll remind our listeners that it is, kind of oddly, but I guess it kind of makes sense, it's a time-limited program that will be operating only through the end of this year. It offers a quarter million dollars for identifying new categories of speculative execution attacks that Microsoft and other industry partners are not yet aware of.

So again, researchers have been prolific the last year and a half in finding new speculative execution side-channel attacks. So if you are someone who's interested in poking around at the Intel processor and seeing if you can find something else, there's $250,000 potentially waiting for you, if you're successful.

And in keeping with doing this the right way, on Monday Microsoft also implemented explicit safe harbor terms and conditions which clearly outline how researchers who are acting in good faith can safely report bugs without fear of facing any legal repercussions. And of course this is something that ought to be codified into law, in my opinion, for the

benefit of our cyber future. I mean, I know that Leo gets questions, I get questions from individuals who believe they have found a flaw, and they're worried about being attacked by the people whose software is flawed as a hacker if they report it. So we really need to fix that problem. That should not be something people worry about.

Kymberlee Price wrote: "Microsoft is committed to ensuring our cloud is secure from modern threats. We built Azure with security in mind from the beginning, and work to help customers secure their Azure cloud environment with products such as Azure Sentinel and Azure Security Center. If a situation arises, our Cloud Defense Operation Center (CDOC) and security teams work around the clock to identify, analyze, and respond to threats in real time." So again, perfect example of a company who clearly perceives that security is not an afterthought. It is a product that they are offering, and clearly important.

JASON: And Microsoft's been in the business long enough to have seen probably both sides of that coin, too; right?

Steve: Oh, yes, yes. Thus we call it Patch Tuesday for a reason.

JASON: Yes, exactly. Forever, yes. Patch Tuesday is legendary.

Steve: So as I mentioned at the top of the show, one of the presentations at Black Hat I got a kick out of, only because you just have to ask yourself, how can a company do something, in this day and age, so wrong? The Chaos Computer Club, which is a group of researchers in Germany, tackled the question of hotel key, you know, door key security. It turns out that the latest trend in high-end, high-tech hotel door keying is something known as "mobile keys." And in this case "mobile" as in mobile phone.

So I think all of us who travel, and I will be joining the ranks of people who travel next month, and we'll be talking about that a little bit in a minute, will be encountering probably in the future the technology of mobile keys. So the credit card-like thing which we're used to sticking in a slot or sliding down the side of the hotel door will be switching to an app that runs in our phone, whether Apple or Android, you know, iOS or Android. So the site that I was led to from this, and we don't know that these guys are the provider that the Chaos Computer Club attacked. But, boy, is it typical. It's OpenKey.co.

And so the headline reads: "Universal Mobile Keyless Entry For Hotels Worldwide. Improve guest reviews and gain competitive advantage with the latest hotel technology, the industry standard for hospitality technology." And their site reads: "The mobile revolution is here. Guests want to use their smartphones to control every aspect of their stay, and the major hotel chains around the world are responding. Keyless entry, mobile check-in and check-out equal a quantum leap for the guest experience. OpenKey" - this particular company - "makes delivering on guest expectations simple, fast, and affordable." And they say they have four steps.

Step 1: Reservation confirmation email sent 24 hours prior to arrival contains mobile key info and a download link. Naturally, if you're going to be using an app for your phone during your hotel stay, you need to have it in your phone. So by sending your prospective guest a link the day before, the guest is able to go, oh, cool, I get to use my phone to get into my room. So you download the app and so that you're ready when you show up at the hotel.

Step 2: Guest downloads the app and registers with name and mobile number. Step 3: Guest can check in via mobile at a kiosk or at the front desk. The mobile key is sent to the app upon check-in. Step 4: Guest uses smartphone to access their room with the tap of a key button. What could be easier? Unfortunately, the guys at Black Hat demonstrated Step 5: Bad guys can waltz into any locked hotel room they choose.

So, yeah. It sounds really great. It reduces check-in friction. But unfortunately, I guess we shouldn't be too surprised to learn that it wasn't done properly. The reporting on this reads: "Researchers developed an exploit that allowed them to perform an array of malicious functions against these so-called 'mobile keys,'" meaning the apps and their use. "A vulnerability in a popular IoT lock key" - and again, we don't know that it's the site I just showed, but it was cited - "used chiefly by a high-end hotel in Europe" - so these guys focused on one particular European high-end hotel - "allowed researchers to break into hotel rooms.

"The locks in question are dubbed 'mobile keys' because of their reliance on mobile phones as opposed to card-based access such as those based on mag stripes and RFID. Researchers at Black Hat USA 2019 showcased how they were able to circumvent an Internet of Things-connected key system utilized by an unnamed European hotel." And the name of the hotel and the specific IoT system were not identified for safety reasons, as the locks are still deployed in the hotel and have not been updated.

The researchers explained: "We went to do the one thing a mobile hotel key is supposed to prevent: wirelessly sniff someone entering his room or just unlocking the elevator, and then reconstruct the needed data to open the door with any Bluetooth Low Energy (BTLE) enabled PC or even a Raspberry Pi." Okay. So without knowing any of the details, this is horrifically, ridiculously horrific because it is such a trivial problem to solve with a proper design.

JASON: And you apparently have the solution here.

Steve: Well, a solution.

JASON: A solution, okay, you're right, yeah.

Steve: Yeah. And that's the problem. Or that's the annoyance is that, in this day and age, this problem is so trivial to solve. So from what we just heard, at its root, we're talking about some form of classic replay attack. All that's needed to prevent this, for example, is for the door, when challenged to unlock, to provide a nonce for the phone to sign and return. The door contains a software ratchet. In this instance that would be a counter which feeds a secretly keyed AES symmetric cipher. So of course that turns the counter into an unpredictable sequence of, for example, 256 bits.

Each door lock is configured with its own secret key, which is never exposed. The AES cipher which encrypts that counter produces a public elliptic key which is used to verify signatures. So the door lock, when challenged, first checks the signature for the key that is currently valid, the one that it's been using, assuming that, you know, like if it didn't know that the previous guests had departed. So it assumes that. It checks to see if the signature returned from its nonce is one that it's expecting.

If that fails, it checks ahead, because it's able to increment the counter to produce successive candidate public keys; right? So it checks ahead to the next public key to see whether that one can verify the returned signature. If not - and it doesn't only have to test one ahead for robustness. It might test a couple ahead. But if not, if it doesn't find a signature that it's expecting, it simply ignores the request. But if the next key does successfully verify the signature on the request of the nonce that it sent out, it makes that next key permanent, ratcheting forward, thus forgetting the previous guest's key, so that a previous guest can no longer unlock the door from the instant that the next guest has successfully been able to do so.

So, and what this little simple system that I just proposed does is it means that the door locks have no need to communicate with the hotel. Each door lock is able to operate autonomously with its own secret key forever, which determines the sequence, as I said,

of its public keys. The hotel system, the hotel's master system knows each room's secret key, so it's able to autonomously issue the proper private signing key to each guest for the proper room.

And if the system is designed correctly, no one with, like in the case of the Chaos Computer guys, no one who had a copy of the mobile key software and the ability to eavesdrop on the conversation would be able to gain any advantage in doing so because this is inherently, thanks to the fact that the door always challenges a request to enter with a nonce, which needs to then be signed by the private key contained by the mobile key software which it received at the kiosk or during check-in, there's no chance for having a replay attack.

And what's annoying is that this is Crypto 101. Right? This is, you know, this is like not even in the final exam. This is in one of the early tests in Crypto 101, is design a system, blah blah blah blah. So, and this is what I've often said on the podcast, that our modern cryptographic tools that we have today are so powerful and so cool that they can provide any kind of nifty functionality like this that we want. And, for example, it's all available for free in the well-tested and audited libsodium library on GitHub, which runs on all platforms.

So I just have to scratch my head when some, I mean, the website is beautiful, looking at this one particular company. Again, we don't know that they're the ones who had this problem. But I was going to say, if they spent one tenth of the money designing their website that they did designing the technology that they're trying to promote with this website, I was going to say that, but it doesn't even - it's free. It doesn't cost anything to have state-of-the-art technology like this any longer. I just - I don't understand it. It's crazy.

JASON: And they're a company with security as, like, the main component of the business; right?

Steve: Exactly. Right. That's what they're selling is locks.

JASON: Kind of ties back to what we were talking about at the very beginning, you know, with the keynote and the grand vision of companies putting security first. But still, even when the company is about security, apparently it's not important enough to put it through those proper paces, I guess.

Steve: No. And that simple system that I just outlined is, I mean, it's so cool, I almost want to go start a hotel lock company just so I can write the code to do this, except that, again, it's not even worth doing, it's so simple. Unbelievable.

JASON: That should not be a reason to stop you, though, Steve. I would love to see you championing SQRL and whatever you would name your new hotel lock business simultaneously. I think that's possible.

Steve: So there was another presentation that was at DEF CON, which followed Black Hat, which generated some frightening-looking tech press headlines. But there was an interesting moral to this. Okay. So I grabbed three headlines from the press, all reporting on the same story. First one: "Driver disaster. Over 40 signed drivers can't pass security muster," read one headline. Another one: "Researchers find security flaws in 40 kernel drivers from 20 vendors." And the third one: "Over 40 drivers could let hackers install persistent backdoor on Windows PCs." All of that was true. No hyperbole. No exaggeration there.

And when you stop to think about it, okay, think about drivers. It does make sense that this could be a problem. Device drivers occupy a sort of security loophole in our systems

today. They're not the primary OS, so they don't get the scrutiny that the OS gets from its vendor. They're provided by random third parties who probably have the best of intentions, you know, they don't want to produce an insecure driver. But the developers there are likely operating under pressure to ship, and their focus is on the driver working, and not crashing, and being stable, not on it being bulletproof against direct attack. And, finally, the drivers often run down in the kernel alongside the OS with direct access to the system's hardware, the physical hardware, and the high privileges that such direct access requires.

So, I mean, it's sort of like the perfect storm. It's not the OS. It's written by random third parties to make their stuff go. The OS sort of has to accept it, I mean, like agree to take it in, in order to be friendly and compatible with whatever thing the user needs the driver for. And the driver has to, by definition, have kernel-level access permissions. So it was sort of inevitable that we would have a problem.

In ZDNet's words - they wrote about this. ZDNet said: "At the DEF CON 27 security conference today in Las Vegas, security researchers from Eclypsium" - I like that, Eclypsium [E-C-L-Y-P-S-I-U-M] - "gave a talk about" - and this is, as I said, there's sort of a moral here. There's like a double whammy ooh - "gave a talk about common design flaws" - that's what kind of surprised me. Not, like, 40 different problems, but a common design philosophical flaw "they found in more than 40 kernel drivers from 20 different hardware vendors. The common design flaw," ZDNet wrote, is that low privileged applications can use legitimate driver functions to execute malicious actions in the most sensitive areas of the Windows operating system, including the Windows kernel.

"Mickey Shkatov, principal researcher at Eclypsium, told ZDNet in an email earlier last week: 'There are a number of hardware resources that are normally only accessible by privileged software such as the Windows kernel and need to be protected from malicious read/write access from user space applications. The design flaw surfaces when signed drivers provide functionality which can be misused by user space applications to perform arbitrary read/write of these sensitive resources without any restriction or checks from Microsoft.'

"Shkatov blames the issues he discovered on bad coding practices which don't take security into account. He said: 'This is a common software design anti-pattern where, rather than making the driver only perform specific tasks, the driver itself is written in a flexible way to perform arbitrary actions on behalf of user space.'"

Okay. And so when I read that, I was taken aback. That's way worse than what I was expecting. This means, essentially, that the driver is an interpreter that is accepting commands from user space and following them. To me, that's unbelievable. Now, what that means is it's an easier way for a developer to write a driver, is just to sort of create their own little interpreter so the driver is a shim into the OS to allow it access to the kernel, and then their userland application will use that to sort of, like, make it up as it goes along. Which is unconscionable.

Anyway, Mickey explained: "It's easier to develop software by structuring drivers and applications this way" - uh-huh - "but it opens the system up for exploitation." And I'm just - I'm astonished that that's what it turns out we have in our systems right now. Shkatov said his company has notified each of the hardware vendors that were shipping drivers which allow user space apps to run kernel code. Oh, my goodness.

Vendors who issued updates are listed below. And they are, in looks like alphabetical order, AMI, American Megatrends International; ASRock; ASUSTek Computer; ATI Technologies, which is to say AMD; Biostar; EVGA, the display guys; Getac; GIGABYTE; Huawei; Insyde, I-N-S-Y-D-E; Intel (what?); Micro-Star International, MSI; NVIDIA; Phoenix Technologies; Realtek Semi; Supermicro; and Toshiba. Shkatov told ZDNet:

"Some vendors, like Intel and Huawei, have already issued updates." Well, good for them. That's good to know. "Some, which are IBVs" - which are independent BIOS vendors - "like Phoenix and Insyde are releasing their updates to customer OEMs." So they'll go through that second-level channel.

Mickey said he did not name all the impacted vendors because some needed 'extra time due to special circumstances,' whatever those are, and future fixes and advisories will be released in the future. He said he plans to publish the list of affected drivers and their hashes on GitHub after his talk so users and administrators can block the affected drivers. In addition, Shkatov said Microsoft will be using its HVCI, which is the Hypervisor-enforced Code Integrity capability, to blacklist drivers that are reported to them, that is, to Microsoft. However, Shkatov said that the HVCI feature is only supported on the 7th gen Intel CPUs and subsequent. Manual intervention will be needed on older systems, and even on newer Intel CPUs, where HVCI can't be enabled.

Microsoft said, and this is unimpressive CYA nonsense, in my opinion: "In order to" - this is Microsoft. "In order to exploit vulnerable drivers, an attacker would need to already have compromised the computer." Eh, okay, not really. The attacker would need to have userland presence, but that's easy. Microsoft said: "To help mitigate this class of issues, Microsoft recommends that customers use Windows Defender Application Control to block known vulnerable software and drivers. Customers can further protect themselves by turning on memory integrity for capable devices in Windows Security. Microsoft works diligently with industry partners to privately disclose vulnerabilities and work together to help protect customers." Okay. Which is to say someone asked the techies, uh, what can we say about this? What do we have in our stuff that can kind of help? And this is what they came up with which is, as I said, nonsense.

Okay. So the talk was on Saturday. The day afterward, last Sunday, the Eclypsium guys published additional details in what has to be, as I noted at the top of the podcast, one of the best titled blog postings, if not of all time, at least in recent times. It's titled "Screwed Drivers - Signed, Sealed, and Delivered." And there is both - I have it in the show notes - a link to the blog posting, and also they have the whole thing available as a PDF where they go into additional details.

So, unbelievable. We have, basically, we have lazy drivers which allow code running in userland. And remember that a driver, once installed by something, is globally available to any application that knows how to access it, and any application can, which are essentially written to make it easy to do and are way too capable of reaching into the system and performing mischief. And the problem, of course, is that they're going to persist in systems that have them installed and do not update for a long time. So we may be hearing about these problems moving forward in the future. Wow.

JASON: No kidding.

Steve: Not surprising, but really, really irresponsible.

JASON: And I'm looking, I admit I'm looking a little bit further ahead because I knew that this would end up being a ping-pong battle, which is basically the battle between Google and any sites trying to lock down their content in a paywall. You had to know that this was going to be a tennis match that was going to keep hitting back and forth, and yet here we are.

Steve: Yup, it's going to be cat and mouse, Google's battle to allow its Incognito users incognitoness to be incognito. We've talked about this before. We saw this, we knew this was going to be happening in their announcement of the features in Chrome 76. As we know, and I'm now using Chrome 76, as is everybody who's been updating recently, release 76 was intended to close a loophole that various commercial paywalled websites

were using to detect when their visitors were viewing the site through their web browsers' Incognito mode.

As we know, Incognito detection had been implemented because Incognito mode inherently flushed the simple-minded cookie histories that were being used to permit a paywall access compromise and tease, where a limited number of pages could be viewed before the site's paywall would slam shut to block further access. The understandable feeling was that, if a visitor wished to be Incognito when surfing the web, they should have that right. And that should also extend to include the fact that they were choosing to be Incognito in the first place.

And as we know, we talked about this before, prior to Chrome's release 76, Incognito mode simply disabled the JavaScript file system API. Since this was trivial for a site's scripting to detect, Incognito mode visitors were being blocked from having any access. So it was essentially - so if you visited The New York Times, and I think the Washington Post, with Incognito mode, you wouldn't get anything. You would be blocked, saying you are using your browser's Incognito mode. We'd be happy to have you come visit, but please come back with Incognito mode turned off.

Well, that annoyed everybody. So essentially it was let us store our crap on your computer, or you don't get to see any of our site. Because of course that's what cookies are is stuff that a website is storing on your computer. So what Chrome 76 now does is implement a RAM-based file system API so that the file system appears to work, rather than just being like returning an error, it appears to work. But it's inherently volatile because it's just in RAM. And thus it won't store anything permanently.

So we know what happened next; right? A RAM-based file system won't behave exactly like a nonvolatile file system, and that makes it detectable. And yes, some sites such as The New York Times immediately adapted their code to do just that. Chrome's file system API presents a smaller maximum file system size. It says that it's going to be - that there is 120MB of available file system. When the script, the JavaScript queries the file system API on how much space you've got, Chrome 76, rather than saying, error, file system API not available, it says, oh, we're here and open for business. We've got 120MB.

Well, it turns out that's a fixed number. And this is not what's seen when the browser is not in Incognito mode. So that's one feature. Also, writes to RAM present a consistent timing compared to writes to physical media, where the timing will vary. Both these incognito detection bypasses have been demonstrated. And as I mentioned, they've been found to be used in the wild. In the show notes I have a snippet which was taken from The New York Times website. And it says in the JavaScript code: "Quota for an incognito Chrome window is a fraction (10%) of the device memory with an upper limit of 120MB." Then it says, "More info:" and gives a link to someone's posting, bypassing-anti-incognito-detection-google-chrome.html. And then INCOG_MAX_QUOTA = 120, which is exported as a constant. So cat and mouse.

Bleeping Computer reached out for their reporting of this to Google to inquire about these two new Incognito detection methods. They were told that Google stands by their previous statement and position, that they will "work to remedy any other current or future means of Incognito mode detection." So we have another new cat-and-mouse game afoot. And this is dumb, since this is a game that Chrome can and will ultimately win. Chrome can trivially simulate the varying timing of reads from and writes to volatile media. And they can remove the fixed declared RAM-based file system size limitation trivially.

So in Chrome 76 I'm sure they'll do that. And with any luck, the guys who are doing the web design at the other end will tell their bosses, you know, we could do something else, but then they'll just go around that, too. So as Google said, if sites wish to be paywalled,

they should require all visitors to create an account. Once that account is created, then some free access could be metered out under the site's control. But since visitors could still create throwaway accounts, a credit card or other payment means would probably also be needed to provide an anchor to that user's identity. And all of this would, of course, clearly reduce the site's traffic, since many users would choose to go elsewhere rather than to create yet one more account anywhere. And it would also mean that no one could use the site without creating at least a temporary account.

This is all clearly a mess, which arises from the fundamentally irresolvable conflict inherent in the goal of wanting to provide some access to visitors who wish to have complete anonymity. Right? I mean, that's what this comes down to. It's an irresolvable conflict to provide some access to visitors who wish to have complete anonymity. I would argue, and Google does, that anonymity ought to be a feature that a web browser can offer. And that falls in fundamental conflict to the fact that, to provide some limited access, having some way of identifying a visitor is necessary. Which means they're not completely anonymous. So I say bravo to Google. And to websites I say, good luck with that because this is not a game they're going to win.

JASON: Yeah. And of course sites are going to be hesitant to implement some sort of like a mandatory account creation for their visitors because a lot of publisher sites, they just want people to know what's there with little to no friction, but not too much. Like a little bit, but not so much. And so that ended up being...

Steve: Right, I mean, I completely get the idea. And I'm appreciative of the fact that I just don't subscribe to the Washington Post and The New York Times, but sometimes I want to read a story that they cover, and it's nice to be able to do that. Again, I'm not reading the paper. It wouldn't make sense for me to sign up because I'm just not going to get enough use out of it. The only thing I could imagine they could do is if, in the future, we get some form of micropayment system. Then if I went there, and they said this story will cost you five cents, you know, and I'd go, okay, yeah, it's worth a nickel. So ding me a nickel, and I'll read the story. I mean, then it would make sense.

And I wouldn't even mind, if there were a means to do it securely, having a payment balance the way I do on my Starbucks card, you know, having a payment balance on those sites that I could draw a nickel from. That, too, again, it's infrequent. I don't like the idea of paying for something I'm not really, really, really, rarely going to use. At the same time, I'd pay a nickel in order to read a story that I wanted to. And running a positive balance I think makes a lot of sense. We don't yet have that infrastructure in place to make all that work seamlessly.

JASON: I know at some point Google had gone down this road. And there was also a service a couple of years ago called Blendle that was all about paying on a per-article basis. But nothing universal; right. And that might be something more that needs to be hashed out in order for it to be successful. But I feel like Google had something along those lines where, instead of seeing ads on certain partner sites, you paid this monthly fee, and I can't remember the name of it, and that monthly fee kind of...

Steve: Actually, you're right. And I remember I used it. I had it also, yeah. And you know, what this conversation reminds me of is that sometimes ideas which are good are just premature.

JASON: They're too early, yeah.

Steve: Like the Atari 400 trying to sell itself as a home computer. You know, like the first round - or the Apple II, which was a neat machine. But, you know, when Dad brought one home, Mom said, "How much did this cost?" And Dad says, "Oh, honey, you can

store your recipes on this." She's like, what? You know, and so the point is we often see good ideas which were just ahead of their time. So they happened, and they died. And when they came back for the second try, then it was okay.

And so the idea of micropayments may be sort of like that; you know? It may be that now, finally, the media companies are well enough established, people are comfortable with creating a balance, or maybe you use your phone, for example, you know, you have an app in your phone that maintains your balance, and you use something like SQRL to say, here's my micropayment for my app, and it securely deducts from the app in your phone, you know. Those sorts of things are possible in this day and age now.

JASON: Yeah. Like those ideas.

Steve: So once again, as Microsoft is beginning to do - it's becoming a Black Hat tradition - they are ranking the industry's top bug hunters. They announce the top industry security researchers and enterprise partners who are responsibly discovering and disclosing the greatest number of vulnerability and zero-day reports affecting its software products. And many in the security industry are now using Microsoft's annual published list as a guide to the identities of today's top bug hunters.

And I think with good cause, security researchers who rank on this list often tout it was one of their highest career achievements because there is a lot of competition out there. According to Microsoft, this year's top security researcher is Yuki Chen of Qihoo, Q-I-H-O-O, Qihoo 360's Vulcan team. And taking second place is Yuki's colleague, and this name I cannot pronounce, Q-I-X-U-N Zhao, Z-H-A-O, who also won a Pwnie Award for Best Privilege Escalation Bug. All in all, Qihoo 360's Vulcan team managed to place eight researchers in this year's ranking of the top 75. And I have a picture of that ranking in the show notes. You've got it onscreen right now. And I have to say that Asia looks like, just from these names, they are really kind of taking over. I mean, you have to look here to find some Anglo-Saxon looking names. Like, okay, there's Joshua Graham. I found one.

JASON: Danny Grander.

Steve: Scott Bell. There's two. Danny Grander, three.

JASON: Matt Nelson. Four.

Steve: Boy. They're few and far between. Yeah, Mario Gomes? Gomes, maybe?

JASON: Yeah.

Steve: Five. Boy, but, I mean, the rest is like - James Forshaw, okay. But, wow. Yeah. Interesting lineup shown there. So anyway, very cool that there is this kind of ranking, and it is serving as the industry's benchmark for this. And I am about to announce a stunning escalation in bounty payout.

JASON: All right. The mother of bounties. Is that what this is? The mother of all bounties?

Steve: Yes, it's possible to become a millionaire hacker.

JASON: Wow.

Steve: In a single blow. Not to be left behind, Apple has bumped its bounties. Apple just updated the rules of its bug bounty program, announced at Black Hat. These effects are

not available; they haven't come into effect yet. This happens this fall, so a few months from now. Apple has enormously increased the maximum reward for its bug bounty program from what was already pretty good, $200,000, to $1 million, making it by far the biggest bug bounty offered by any major tech company for reporting vulnerabilities in its products.

The $1 million payouts will be rewarded for a severe deadly exploit. So, you know, it's got to be bad. But still, a million dollars. Severe deadly exploit, a zero-click kernel code execution vulnerability that enables complete persistent control of a device's kernel. In other words, the worst possible of all possible things. But if you can find it, you're a millionaire. Well, before taxes. Less severe exploits will qualify for smaller payouts. So it's now possible to become a millionaire hacker overnight by finding just one serious bug in an Apple product.

And Apple's bug bounty program does not only apply to security vulnerabilities in the iOS mobile operating system, which of course has been the focus of all of this in the past, but now also covers, or will when this goes into effect, all of its operating systems - macOS, watchOS, tvOS, iPadOS, and iCloud. I didn't know iCloud was an operating system, but I guess for this purpose it qualifies. So find a bug in iCloud. Until now, for the past three years, Apple's bug bounty program has only rewarded security researchers and bug bounty hunters for discovering vulnerabilities in iOS. And as I said before, remember that that limitation does remain in effect until this fall.

But wait, there's more. Or as Steve Jobs would have said, "One more thing." Actually, two more. Starting next year, Apple will also provide, get this, pre-jailbroken iPhones to select trusted security researchers as part of the iOS Security Research Device Program. That's so cool. These devices will have far deeper access than iPhones available to everyday users. This is just smart on Apple's part. They've been doing some thinking. This includes access to ssh, the root shell, and advanced debug capabilities, allowing researchers to hunt for vulnerabilities at the secure shell level.

Although anyone can apply to receive one of these special iPhones from Apple, the company will hand out only a limited number of these devices, and only to qualified researchers. So you know, if you crawl out of the basement and say, hey, I want a pre-jailbroken iPhone, you're unlikely to get one. You probably have to be somebody who Apple recognizes and a legitimate researcher. But this is smart because this allows the researcher to dig in deeper and find things. So, very cool.

And I said there were two more things. The last thing is, in addition to its maximum reward of a million dollars, Apple is also offering - it turns out a million is not the biggest possible because they're offering a 50% bonus on top of it to researchers who find and report security vulnerabilities in the pre-release software, ahead of its public release. So potentially a maximum reward of up to $1.5 million. So you could be a millionaire after taxes.

This is smart, of course, since finding pre-release bugs is better for everyone. Get them before it ships. So that incentivizes researchers to take time looking at pre-release stuff rather than, like, I'll get around to it eventually because you could get an additional 50% on top of whatever it is that you find. Even if it's not the million dollar jackpot, still it bumps whatever you would have had up by 50%. Applications for Apple's revised bug bounty program will be open later this year, and this will be open to all researchers rather than some limited number of security researchers approved by Apple. So that is to say, they're now going to open the bug bounty program widely, rather than keeping it much more closed.

This expansion and massive boost in the payout of Apple's bug bounty program are likely to be welcomed. I mean, I'm sure they're going to be welcomed by security researchers

and bug bounty hunters who either publicly disclose vulnerabilities they discovered in Apple products, or sell them to private vendors such as Zerodium, Cellebrite, and Grayshift, who as we know deal in - basically create and have a gray market in zero-day exploits for profit. And this seems like a clear smart move on Apple's part, to me. A researcher who does find a showstopping exploit in an Apple device would have been a little hard-pressed to offer it to Apple for a pittance when the likes of Zerodium would be willing to pay much more. Now, a developer can be a good guy, do the right thing, and receive a top-of-the-industry class monetary reward for their efforts.

So I think this represents some good maturity in the industry. And anything that upsets Zerodium and Cellebrite and Grayshift seems like a good thing to me, and this news from Apple would certainly upset them. And who knows? Maybe we're going to see other major vendors respond in kind, if there's a little bit of an escalation war here in bug bounty payouts. Apple has just upped the ante. Microsoft may just decide to follow suit. So that would be good, too.

A listener of Security Now! wrote a nice piece of email regarding SQRL and my upcoming travel to Europe that I wanted to share. The subject was "Dublin Visit." Ben Fletcher wrote: "Dear Steve, I was delighted to hear on Security Now! that you're coming to Dublin, and I'll be at your OWASP talk." He says: "I moved to Dublin recently, having served in the Royal Air Force in the U.K. for the past 16 years. I was originally a mechanical engineer by degree, but quickly realized" - smart man - "that IT was the future. I started volunteering for IT engineering jobs and haven't looked back.

"However, this transition left me with a pretty steep learning curve. You might remember the days when Conficker hit a number of networks. We were not immune. As I frantically searched the web for information to be able to understand the issue and brief my superior officers, I came across Security Now!, having listened ever since and all previous. Suddenly, I was the subject matter expert, deploying and commanding tiger teams to remediate the problem across the U.K. and the world."

He says: "I honestly can't thank you enough for the invaluable service you've provided me and the knowledge I've gained over the years. I'm now working in Grant Thornton as a cybersecurity consultant, doing all sorts of fun. I'm sure your timetable will be tight while in Ireland, but I'd be delighted to take you out for dinner with our team or anything you fancy, whether it be something cultural or touristy. It's the least I can offer you for helping me develop as an IT professional. If there's anything we can do for you while you're in Ireland, please do not hesitate to ask. I look forward to meeting you in person. Kind regards, Ben."

So first off, I very much appreciate the offer, Ben. But Lorrie and I are already committed to sharing a meal with the Dublin OWASP gang. And when we're not doing that, I strongly suspect that we're going to want to just wander around aimlessly, soak up the local environment and culture. But again, thanks.

And I did want to note for our listeners that I've put up a short calendar listing the three forthcoming planned SQRL presentations, the first one - or the next one - here in Orange County, California next Thursday; and then both Dublin, Ireland and Gothenburg, Sweden toward the end of September. The calendar entries contain links to the presentation announcements, as well as links to register to attend the meetings if you're interested and able to.

And so just go to GRC.com/calendar. You can add a .htm to that, if you want. Otherwise the site will do it for you. GRC.com/calendar. Anybody who is interested in attending Orange County, California next week or, what is it, the 24th and 26th of September in Ireland and Sweden, respectively, there are links to - both of the OWASP groups in Europe used Eventbrite in order to manage the invitations for the event.

So I did expect a robust response from our listeners to last week's Steve's File Sync Journey podcast, and our listeners did not disappoint. It's very clear that this topic is of great interest to our audience. So there will be a follow-up extensive results podcast. As it turns out, the way the timing of the SQRL OWASP presentation trips have worked out, with my trip abroad ending in Boston to meet Leo for the LastPass event the Thursday, I think it's October 4th, which is a Thursday in Boston, I'm going to end up being away from Security Now! Tuesday podcasts for two weeks.

So Leo and I will be prerecording the Security Now! podcast for that first Thursday the previous Saturday afternoon before I depart. But for the following Tuesday, since we were unable to find a time soon after we return to record for that one late, instead we're going to prerecord a podcast to fill in that week. And that podcast will be my comprehensive results of my several months, by that time, of exploring multisystem file synchronization solutions. So I will be working on that in the background, experimenting with all kinds of things and putting together a pretty comprehensive review of what I find.

And along those lines, a note from a listener, Cliff Spooner in Utah, struck me as intriguing. And I thought it was very clever. The subject was "SN-726 File Encryption Option." And he wrote: "VeraCrypt with Dropbox." Okay, now, I thought, what? Okay. He wrote: "Steve, after listening to your podcast #726 I thought I would share what I do to secure sensitive information in Dropbox." He says: "I use VeraCrypt." Which, now, remember, VeraCrypt is a whole drive encryption tool; right? I mean, it is what TrueCrypt became. It is the inheritor of the TrueCrypt code. It's been audited and is being maintained, and it's like a strong whole drive encryption. But remember that you can also encrypt a volume, that is, you can encrypt a file and mount that as a drive.

So he says: "I use VeraCrypt, which sounds like a terrible idea because of the large file size of one encrypted container. But Dropbox is magic, and it is the only sync solution that is able to transfer only the changes to the container when they are made. Other sync solutions need to re-upload the entire container." He says: "You will upload a large container once, which may take some time. But after that, it will only be the changes." He says: "Maybe this will help solve some of your issues. Thanks, Cliff."

Now, I don't think that's the solution I would choose, but I loved it from a cleverness standpoint. Now, okay. So first of all, remember that what this does is this creates a large file, and a hard drive is at the lowest level a - can be viewed as a block of sectors, but at the practical level it's a block of clusters, where you typically have, for example, eight-sector clusters where the sector is half a K; eight of them make 4K. So you have 4K clusters. Well, what that means is that changes to the file system are reflected by changes to clusters, and those clusters will be bitmaps, they will be directory chunks, and they will be file system contents.

So it certainly is the case that, if you were to create a large file container, and you use VeraCrypt to turn it into an encrypted file system container, you could then send that to Dropbox, like put it in Dropbox. Dropbox, your local client, would start monitoring it for changes. And what Cliff discovered or noted or knew or found or something was that Dropbox would be very smart about syncing changes to this massive, potentially multi-gig blob. And of course the fact is, when you do update a file, like you've got files living in there, and you're hitting Control Save, Control Save, Control Save like all the time, as I do, you're only changing a few clusters in that file.

And so the Dropbox local client sees that only - so it doesn't know that that's a file system. It sees just a monster single container where a few little 4K chunks of that are changing. And so that's all it's uploading. I mean, arguably, this would be more efficient than other sync solutions, which are resending the entire file over and over and over, if

indeed that's what they're doing. I don't know that that's what they're doing, that they're not getting smarter. But this is very cool.

So I just wanted to share that with our listeners. Again, I don't think that's the solution I'm going to be suggesting because of this. I have in the show notes: "Although I need more time to reach conclusions about robust, flexible, and secure multi-machine file syncing, I do need to acknowledge the massive response I received about one solution in particular: Syncthing." Probably 90+, 95%? I mean, there were, like, other things people have suggested. But by far and away Syncthing, S-Y-N-C-T-H-I-N-G, was most often suggested.

I now have it running on my Drobo, which was nice that I was able to have my own existing local Drobo operating as a Syncthing node. And I have it running on both of the workstations, here and in my second location. Both locations are behind NAT routers, and that secondary site is actually behind chained double NAT because I wanted to place, and I did, I placed a Netgate SG-1100 pfSense firewall/router in front of the ASUS router that we already have there because I wanted to do a bunch of manual fancy port shifting stuff, and IP filtering.

And although I do have static mappings and filters in place at each point with pfSense, and I could manually punch holes through the NAT routing again if necessary, because I wanted to be able to report on Syncthing for our listeners, I first wanted to try firing up these babies with everything behind those NAT routers. Syncthing is UPnP capable, so IF I had UPnP enabled anywhere, which of course I don't, we know that it could have set up port mappings itself. But as we talked about many moons ago on this podcast, it is possible to punch through NAT routers if you have a properly set up Rendezvous server. And Syncthing maintains a Rendezvous server, actually multiple of them. So it can work through NAT when it's done right.

And Syncthing does it right. All of the Syncthing instances that I am running are directly connected to each other thanks to NAT punch-through, without any external relaying, despite the fact that everything is safely behind NAT. Meaning that other IPs cannot see any of my Syncthing instances because they're other IPs. I mean, they're going to get hit. They're going to get stopped by the NAT in the same way that we often talked about NAT being inherently, natively, automatically a very good firewall. And if anyone feels skittish about having a third party performing NAT punch-through for them, Syncthing allows you to operate your own external NAT Rendezvous server.

So anyway, so far I'm very intrigued by this. I'm still going to make myself take a look at other options because there are several that I had not found before I settled on Dropbox, which is what I talked about using last week. But this is, so far, this is - I'm very impressed. And I even have an idea for an interesting hack that might work to bridge Syncthing to hosted cloud providers for some of the benefits that are provided, such as automatic versioning, which you wouldn't have unless you did something more, and the ability to create public links for people who wanted to be able to share arbitrary files which are being synced with Syncthing publicly.

So anyway, lots more to come. That will be a podcast, let's see, it's going to be in the beginning of October. So October, yeah, October 1st. That will be the follow-up on file syncing after lots of experimentation by me on that.

JASON: So 731, 732, somewhere around there.

Steve: Yeah. And also our Picture of the Week last week was one that we had fun with. We always do. It's where the combination lock has the combination right next to the lock saying, in this case it was, "To open the door, press #2 and #4 together, then #3." And a number of people commented. A good friend of mine was familiar with that particular

combination door lock. And he noted that press #2 and #4 together, then 3 is, in addition to being printed on the paper right there on the door, also the default factory set combination for those locks.

JASON: Of course.

Steve: Unbelievable.

JASON: Of course it is.

Steve: Of course it is. Unbelievable. I also posted - I have a nice comment about SpinRite. I posted two weeks ago, actually almost exactly two weeks ago, a post called "Toe Stubbed." I posted it to the SQRL newsgroup. My bookkeeper/operations officer gal who's been with me for about 35 years, Sue, her trusty and crusty very old Windows XP machine which she had been using for GRC business finally died. I remember talking about this a couple years ago because I'd had a mirror - I had a pair of mirrored hard drives, and it is being backed up using Jungle Disk. So all of the accounting information and stuff that was important on that machine was being continually backed up to the cloud. So nothing was ever in danger. But it's a pain to, like, rebuild a machine.

So two years ago I picked the machine up, brought it here. One drive was completely belly-up. The other one was in bad shape. I ran SpinRite on it overnight, brought the drive back to life, was able to clone it to a new pair of mirrored drives, and gave it back to her. One of the reasons that we stayed on XP is that we have some 16-bit code. The very first database we used for managing SpinRite 1 sales, well, 1, 2, 3.1, 4, and 5, until toward the end of 5 I wrote GRC's ecommerce system, which we've been using through the end of 5, well, immediately after finishing 5 I wrote the ecommerce system and then wrote SpinRite 6. And then all of SpinRite's 6 sales has been through that ecommerce system. But earlier customers want to be able to upgrade. And so we've kept that - it was FoxPro v2.6 that ran in a DOS box. And it is very 16-bit code.

Anyway, so this happened again. Two weeks ago, email from Sue, machine was dead. She switched to using her laptop to continue doing email. Wednesday morning, since I couldn't do it because of the podcast on Tuesday, I ran down, picked the machine up. Once again, one of the two mirrored drives was completely dead. The other one was in really bad shape. The system wouldn't boot anymore and so forth. Ran SpinRite on it overnight, came in, it had fixed a bunch of problems, enough so that I was again able to make an image of the drive. But I thought, okay, no more XP. It's finally time to move her forward.

So I decided to go to Windows 7 because Windows 7 much more easily runs XP mode, which I thought I might need. But as it turns out, FoxPro was able to run in the cute little DOSBox tool. You know, DOSBox is - it's called DOSBox - was developed so that gamers who were wanting to do old-school 16-bit DOS gaming would still be able to do so. So it's a neat little lightweight - it's only a few megs - lightweight 16-bit DOS emulator, and that's all that FoxPro needed. It will work just beautifully.

Anyway, the point of all this is that one of the people hanging out in the SQRL newsgroup, Jeff Root, he wrote: "I was a SpinRite version 5 owner," he says, "and when I wanted to get the latest, I just bought the version 6 release." He says: "It honestly never occurred to me that GRC would be able to find my previous purchase in their records, especially since years had passed." He says: "Now you're running FoxPro in an emulator, just in case someone wants to upgrade?" He says: "I wish Microsoft had as much respect and sense of responsibility for their customers. Perhaps if GRC had shareholders, the answer would be different." And then a smiley face.

So anyway, fun story. Sue's machine is up, running 7. No data was lost, thanks to SpinRite. And, yes, if somebody purchased a copy of SpinRite in 1987, we can find you, confirm your purchase, and offer you a discount to SpinRite 6. However, as I did say, when we finally do release SpinRite 6.1, which will be the thing I start on immediately after finishing the SQRL documentation, which means it's not far from now, 6 will have existed for 15 years. So at that point we're going to say, okay, we're going to finally give FoxPro a rest and not continue to upgrade people. Because anyone will have had 15 years to upgrade to 6. But anybody who has 6, as I have always said, will get 6.1 for free, and then we will be moving forward.

So a couple of closing-the-loop bits from our listeners. A listener whose Twitter handle is @spawnandjesus, I think that's how I would pronounce it, he said: "Hey, Steve. I haven't jumped into this week's podcast yet, but I thought I'd let you know your latest ransomware creators are obviously into anime." And I mispronounced it again. He says: "The word R-Y-U-K should be pronounced ree-ook," he said, "as he's a character from a series called Death Note. Also, thanks for the best tech podcast on the wire. I've been listening for over a year, originally discovered your website as a kid in the 1990s." He says: "I love SQRL. I love SpinRite. I'm patiently waiting for the next version to purchase another copy." Wow, well, thank you. I appreciate that. And as I said, I will be getting to that before long.

Graham Booker wrote: "Random SN question." He says: "I've noticed in many of your show notes for Security Now!, you have ~30~ at the end. What's the history/meaning of this?" He says: "The SQRL logo in its place in #725 reminded me of this." And anyway, so Wikipedia has this to say about -30-. Wikipedia says: "-30- has been traditionally used by journalists in North America to indicate the end of a story. It's commonly found at the end of a press release. There are many theories about how the usage came into being, for example, from that number's use in the 92 Code of telegraphic shorthand to signify the end of a transmission in the American Civil War era."

JASON: Wow.

Steve: Also, Wikipedia says: "It was included in the Associated Press Phillips Code of abbreviations and short markings for common use. And it was commonly used, when writing on deadline and sending bits at a time to be typeset, as a necessary way to indicate the end of the article." So that's what the -30- is, although I confess I have switched to using the SQRL logo because it's cute, and it just occurred to me that I could. So there.

JASON: Yeah. I think the SQRL works better than -30-, in my opinion.

Steve: Yeah, I do. And thank you. And Loren Burlingame tweeted: "Steve, I just listened to the latest SN where you were talking about synchronization. I am not sure if you have run across my favorite synchronization utility, SyncBack," and he gives the URL: 2brightsparks, 2 as in numeral 2, brightsparks.com, syncback/sbpro... He says: "But figured it is worth mentioning in case you haven't. It is Windows software and can run from Scheduled Tasks. But it supports everything you could ever want in a sync utility, including synchronization to S3 and other cloud providers (OneDrive, GDrive, et cetera), ransomware protection, versioning, encryption, et cetera. You have to pay for the Pro version to get all those features, but it's worth the money, in my opinion.

"Thank you so much for Security Now!. I never miss an episode, and every Wednesday morning is like a mini Christmas when I see the show in my feed. I should almost mention that I have" - and I was wondering why I had this whole thing here in the notes, but then here's the reason why. "I should also mention that I have also run into all of those issues you described with OneDrive and Google Drive clients, and I stopped using

them in favor of SyncBack. The beauty is that, since I am using OneDrive's (and GDrive's) back-end, I can access the files on mobile platforms, as well."

Anyway, so I did appreciate the fact that Loren commented that he or she had stumbled upon the same problems that caused me to finally start looking for something other than the first kind of more obvious choices, Google Drive and OneDrive. Thank you, but neither of those work. And we will be, in a month from now or so, talking about what it is I have found that does. And I'm going to try to keep up with all the tweets and all of the feedback in the GRC.com/feedback of people telling me what their sync discoveries are so that I can produce a comprehensive report. I think I'm going to be able to do one, and even come up with some interesting hacks that maybe no one has thought of before.

So I think that's going to be a great podcast. It won't have any news, but we'll catch up on two weeks' of news that I miss on the podcast that follows that. So I think it's good stuff.

JASON: Absolutely. GRC.com/feedback, for anyone that wants to participate?

Steve: Yup.

JASON: Right on, GRC.com/feedback. Jam-packed, as usual. What are you going to do next week because you don't have Black Hat or DEF CON to draw from? And nothing happens in security if there's no Black Hat or DEF CON.

Steve: Oh, Jason. That's true, although, well, I mean, yeah, right. There were a couple stories that I just I ran out of page count. This page count generally brings us to about two hours, which is sort of my target, and that's where we are at an hour and 54 minutes right now. So there will be a couple things I did not get in this week that I will be covering next week because they are also interesting. And I have no doubt that another week will bring us another week's worth of security events. That always seems to be the case.

JASON: Always; right? You can go to GRC.com to find everything you need to know about everything Steve is working on. SpinRite, of course, best hard drive recovery and maintenance tool. You can find a copy by going to GRC.com. Information about SQRL, Steve, you've got to be so excited to be this close to finishing the documentation on this. I can only imagine. You've been working on this for a while. Information on SQRL can be found there. Audio of this show, as well as transcripts, if you want to find transcripts of Security Now!, GRC.com. That's where you're going to find them. I don't believe those are serviced anywhere on our side; right? It's only at GRC.

Steve: Correct, correct.

JASON: There you go. So that's where you go. If you want to come to our site for the show, the website is TWiT.tv/sn for Security Now!. There you're going to find the audio, the video links, podcast subscribe links, everything that you need to know, all episodes, so you can start from the beginning. And you'll also find that we record this show, usually record it live on Tuesdays at 1:30 p.m. Pacific, 4:30 p.m. Eastern, 20:30 UTC. And all that information can be found there, TWiT.tv/sn. And, yeah, I think we've reached the end, Steve. Thank you so much, man. It's always fun hopping on with you.

Steve: Jason, it is, and thanks for standing in for Leo. It's always a pleasure working with you.

JASON: You bet. We'll talk to you soon. And Leo and Steve will see you next week on Security Now!. Bye, everybody.

Steve: Thanks, buddy.