SECURITY NOW!

Transcript of Episode #589

Listener Feedback #244

Description: Leo and I discuss Android meeting Gooligan, Windows Upgrades bypassing BitLocker, and nearly one million U.K. routers taken down by a Mirai variant. The popular AirDroid app is "doing it wrong." Researchers invent a clever credit card disclosure hack; Cloudflare reports a new emerging botnet threat; deliberate backdoors are discovered in 80 different models of Sony IP cameras; we get some closure on our San Fran Muni hacker.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-589.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-589-lq.mp3

We talk about a fun hack with Amazon's Echo and Google's Home, how to kill a USB port in seconds, a caution about keyless entry (and exit), too-easy-to-spoof fingerprint readers, an extremely troubling report from the U.K., and, finally, some good news: the open-source covert USB hack-defeating "BeamGun"! We move on to a bunch of fun miscellany, some great sci-fi reader/listener book news, and however many questions we're able to get to by the end of two hours.

SHOW TEASE: It's time for Security Now!. Steve's here. We've got lots to talk about. It's a question-and-answer episode; and, for the first time ever, with no questions and no answers. We actually run out of time before we get to questions. Maybe next week. Don't be fooled by the title of the show. We have lots to talk about, though. So let's get to it. Security Now! is next.

Leo Laporte: This is Security Now!, Episode 589, recorded Tuesday, December 6th, 2016: Your questions, Steve's answers, #244.

It's time for Security Now!, the show where we protect you, your loved ones, your privacy online with the king of security, Mr. Steve Gibson. He is here once again to bless us with his wisdom and knowledge. Hello, Steve.

Steve Gibson: With the Vulcan hand sign, your thumb's supposed to be in, right, with the other fingers?

Leo: You're asking the wrong guy. Yeah, I think in.

Steve: I think in, yeah. Out seems wrong. Anyway, so...

Leo: And which hand? You do yours with the left.

Steve: Well, I'm left-handed, so...

Leo: I'm left-handed. I can't really do it with my left. I practiced with my right, I guess.

Steve: The best man in my wedding was unable to do it. So he had some little rubber bands.

Leo: Oh, that's funny.

Steve: That were, like, holding his fingers in two sets. Because I said, "Gary, please do not embarrass me," because, I mean, there's never been anyone in the world more capable of embarrassing someone.

Leo: But wait a minute. Wait a minute, now. This is a revelation because - are you saying everyone at your wedding party had to do the Vulcan "live long and prosper" gesture?

Steve: No. No, he got up for the best man speech. And he said, "Steve has asked me please not to embarrass him terminally. So all I will say is" - and then he held up his hand, and there were two little red, like...

Leo: Rubber bands?

Steve: ...orthodontic rubber bands.

Leo: Oh, that's too cute.

Steve: He said, "Live long and prosper."

Leo: Aw. What a great toast.

Steve: That was perfect for me, yes.

Leo: Great toast.

Steve: So last week was Q&A 243. And there was so much going on that we, well, we hit our two-hour podcast time limit after five questions. And those were the easy questions that I sort of put first so we could just get through them. So I thought, okay. Let's try again: Q&A #244. Except that we've got Android meets Gooligan, Windows Upgrades bypass BitLocker, nearly a million U.K. routers taken down by a variant of the Mirai worm/botnet. The popular AirDroid app is seriously doing it wrong.

Leo: Yikes.

Steve: Researchers have invented a clever credit card disclosure hack. Cloudflare reports a new emerging botnet threat that they've been tracking.

Leo: Oh, boy.

Steve: Deliberate backdoors were discovered in 80, eight zero, different models of Sony's higher end...

Leo: No.

Steve: ...IP cameras.

Leo: No.

Steve: We get some closure on our San Francisco Muni hacker.

Leo: Oh, good.

Steve: And I heard you at the end of MacBreak Weekly noting this fun hack where the Amazon Echo and the Google Home were talking to each other.

Leo: Yeah, yeah, yeah.

Steve: So I just wanted to mention that.

Leo: I'll play it.

Steve: There is an interesting USB port-killing dongle which has surfaced and is now commercially available that we need to talk about briefly. A caution about keyless entry. Too-easy-to-spoof fingerprint readers on some smartphones. An extremely troubling report comes from detectives or detectives' behavior in the U.K. And, finally, some good news: An open source covert USB hack-defeating bit of software. We've got a bunch of fun miscellany, some science fiction reader/listener book news. And not that there's going to be any time left, but we do have - I think I had seven questions. I thought, well, we're never going to get to 10, so I'm not even going to pretend. But we've got five from last week that have spilled over, and I found two others just so we wouldn't run out. We're not going to run out. So, yes, I think another great news-packed podcast for our listeners.

Leo: Nice. I'm very excited. And we had such a good time with you when you came up here for our year-end TWiT, which will air on Christmas Day.

Steve: Yeah, great.

Leo: And then for the Christmas party you stuck around as long as you could.

Steve: Until Rene and I said, "Okay, we're done. Take us back to the hotel."

Leo: Took you back in the Tesla.

Steve: And I got to buzz around in your lovely Tesla.

Leo: Oh, that's fun.

Steve: With the doors that worked most of the time.

Leo: Yeah, you know, it's funny, it's behaved very well ever since. But something about you guys just broke them. Anyway.

Steve: So our Picture of the Week I got a kick out of. Someone shot me a cartoon from the always wonderful xkcd. And this is his No. 1758, which depicts a large window-encrusted building in the background with a very large prominent lawn sign labeling the building Department of Astrophysics, under which it gives us their motto. The motto of the Department of Astrophysics: "Yes," it reads, "everybody has already had the idea, 'Maybe there's no dark matter - gravity just works differently on large scales.' It sounds good, but doesn't really fit the data."

Leo: Oh, Steve.

Steve: So know that before you pass the sign. Yeah. And, you know, my point is, or was, has been, still is, that it's like the equation doesn't balance. And so you go, oh, okay, there's just a big blob of dark matter that, if we put that in, look, now the equality works. It's like, okay, but that's really a hack. So I have no opinion one way or the other. I just thought it was fun that somebody had come up with a theory of gravitation that didn't seem to have that problem.

But speaking of problems, Gooligan, which I guess must be what happens when you take hooligan and google-ize it, you get Gooligan? I mean, I guess that's the derivation of this name. Check Point is in the news again. They've essentially reverse-engineered a mystery which - I think they have pretty much figured out what's been going on, which is a campaign which has been taking advantage of users who make the mistake of loading things into their Android phones from third-party, non-Google Play sites.

So that's the source of the problem. And it's estimated that more than - now, I'm seeing one million. Oh, I know I saw two because logs collected by Check Point researchers show that every day Gooligan installs at least 30,000 apps fraudulently - and I'll explain what that means because it's installing them as a means of making money for the hackers, which is an interesting twist, which of course provides their motivation - on breached devices, or over two million apps since the campaign began. So there's two million apps. And they've tracked a compromise of as many as one million Google accounts.

So the Check Point blog starts out saying: "As a result of a lot of hard work done by our security research teams, we revealed today" - and this was last week - "a new and alarming malware campaign. The attack campaign, named Gooligan, breached the security of over one million Google accounts. The number continues to rise at an additional 13,000 breached devices each day. Our research exposes how the malware roots infected devices and steals authentication tokens that can then be used to access data from Google Play, Gmail, Google Photos, Google Docs, G Suite, Google Drive, and more. Gooligan is a new variant of the Android malware campaign found by [they say] our researchers" - Check Point researchers - "in the SnapPea app last year." So this began, as these things so often do, as sort of an isolated instance. And then it got - sometimes it's militarized or weaponized or commercialized.

Check Point reached out to the Google Security team immediately with information about the campaign. And they say: "Our researchers are working closely with Google to investigate the source of the Gooligan." So what they found was traces of this Gooligan malware code in dozens - and I saw the number 86 as I was digging into this - of legitimate-looking apps on third-party Android app stores. These stores are an attractive alternative, as we know, to Google Play, for those who are not security conscious because many of the apps are free, or offer free versions of paid apps. However, the security of these stores, as we know, and the apps they sell, aren't always verified, although we might say are not verified.

"Gooligan-infected apps can also be installed using phishing scams where" - so it's not just users going to a third-party store or following a recommendation from a friend, "Oh, go get this over here," but it could just be a phishing campaign that gets this thing into your phone. So the way this makes money is that the malware, once the malware is in your phone, it simulates app advertisements provided by legitimate ad networks, causing the apps to install on the victim's device.

So you first install this covertly malicious, Gooligan-infected app. And who knows what the back story is there. I mean, there's 86 of them. So you have to sort of wonder if the Gooligan people aren't maybe paying for co-residence in these apps. Or, I mean, maybe they're creating themselves from scratch, but that seems a little less likely than, hey, we'll give you some money if you'll put this little extra goody in your app that's not going to go through the Google Play Store, which would allow Google to spot it and remove it.

So once the phone is infested, it then downloads additional apps for which the Gooligan malware authors are paid by the ad network, by the legitimate ad network, when one of those apps is - one or more are installed successfully. So this affects Android 4 devices

(Jellybean and KitKat), and 5 (Lollipop), which encompasses more than three quarters of the market devices today. And the preponderance of them, about 57%, are located in Asia, with about 9% in Europe. So essentially that's what's going on.

Now, what's significant is that what Gooligan downloads, after Gooligan gets into the phone, it sends data about the device to the campaign's, that is, the Gooligan campaign's command-and-control server. It then downloads a rootkit from the command-and-control server that takes advantage of multiple well-known, but typically unpatched, Android 4 and 5 exploits, including one named VROOT. And it's got the CVE designation starting with 2013. So it's three years old. And that doesn't matter, as we know. The older Android phones, or in some cases not so old phones that just aren't being maintained by their providers, won't be updated. And the other common one is known as Towelroot that has a CVE dated 2014.

So in both cases these exploits still plague many devices today because security patches that fix them may not be available for some versions of Android, or the patches were never installed by the user, or never offered by the bandwidth provider. So if the rooting is successful, the attacker has full control of the device and can execute privileged commands remotely. So it's a full remote-access takeover of this breathtaking number of Android devices. After achieving root, Gooligan downloads another new malicious module from the command-and-control server and installs it on the infected device. That module injects code into running Google Play and Google Mobile Services to mimic user behavior. So sort of like a little automation shim that allows Gooligan to avoid detection. And that's been seen historically on other mobile malware.

That module allows Gooligan to steal a user's Gmail email account and authentication token information, install apps from Google Play, and rate them to raise their reputation. It gives all the apps five-star ratings after it downloads them and installs adware, which generates revenue. So of course the ad servers, which don't know whether an app using its service is malicious or not, sends Gooligan the names of the apps to download from Google Play. So the whole system sort of forms a self-sustaining ecosystem which is catastrophically rooting and infecting vulnerable Android devices while generating revenue for the miscreants behind it and continuing to support itself.

So unfortunately, you know - and we've watched the evolution of this over time. Ten years ago, several years into this podcast, we were sort of musing how isn't it interesting that all the Outlook viruses that we were covering back then, you know, they never really seemed to do anything except just exist. But the moment we saw this pay-to-unlock-your-encrypted-data cryptomalware, suddenly it became clear, as we predicted on the podcast, we're going to be seeing a lot more of this because, as soon as this goes from, "Oh, look, Ma, what I was able to do after school" to, "Oh, look at the size of my bank account," that really changes the equation.

And so here again is a way of leveraging the Internet advertising revenue system. And we've seen fraudulent clicks and so forth performed by scripting and browsers and that. This now, of course, also your phone is hooked into a command-and-control server. And remember that the fact that it's obtaining fresh code means that that allows them to dynamically vary what sort of this ad hoc network of rooted devices will do. So more than a million of them are currently doing this. But they can change that, apparently, at will. And it's not clear. I guess these phones are stranded. I don't know how Google ever really resolves this for people, except the phones die, finally, and they get newer ones, which hopefully have an additional four years' worth of security learning under their belt. Wow.

And, oh, this is a fun one, a BitLocker bypass on Windows 10 through Windows

Upgrades. The root of the problem is unattended upgrades on Windows 10 machines. Think about it. How can a locked machine, protected by BitLocker, that is, so that the system drive, the whole drive is encrypted, how can it perform unattended upgrades? We know they do. People are annoyed by it all the time. The system has to have some means of decrypting the system drive without the user's explicit interaction, permission, and password.

A hacker named Sami Laiho discovered an issue in Microsoft's Windows 10 OS that allows attackers to gain access to BitLocker-encrypted data. Some of the coverage has been inexpertly worded because they talk about BitLocker being disabled. Okay, well, we know you can't disable BitLocker. If the entire drive is encrypted, then you actually have to go the other direction. You have to fully enable BitLocker, giving it whatever it needs, the master secret key, which then allows it to decrypt and encrypt individual sectors through some means.

So Sami posted on the Win-Fu blog essentially the highlights of his method. He wrote: "There is a small but crazy bug in the way the Feature Update, which was previously known as Upgrade, is installed. The installation of a new build is done by reimaging the machine and the image installed by [as we know] a small version of Windows called Windows PE, the Preinstallation Environment. This has a feature for troubleshooting that allows you to press Shift-F10 to get a command prompt. This allows for decrypted access to a BitLocker-encrypted hard disk" because, during the upgrade, Microsoft necessarily has to be able to get that drive into a decrypted, or unencrypted read/write state.

So if Shift-F10 is pressed at the appropriate time during an upgrade, without administrator privileges, without the legitimate user present, requiring no authentication at all, a command prompt window is opened which allows full and unrestricted access to the system's storage devices. Whoopsie. So this method works, not when performing little incremental updates, but on a major release install. So, for example, it works if you went from the original Windows 10 release build, and you were jumping to the November Update 1511 or the Anniversary Update 1607. And of course those are things that affect all users.

It also currently works on any new insider build since, as we know, those are sort of incremental whole system movements. So it's that phase. It's the major release upgrade. Windows, while it's doing it - and we presume it's managing the keys securely. What this implies, though - and this has always made me nervous about BitLocker as opposed to, for example, TrueCrypt. For example, Windows couldn't do this. Windows 10 with a TrueCrypt-encrypted drive could not do it because it would not have the keys. But since it's established the BitLocker drive, it has no doubt squirreled its own private copy of that master decryption key somewhere, hopefully using Trusted Platform Module, I mean, like, doing something good to protect it.

But the point is it's there. And that automatically implies that, if compelled to do so, Microsoft does have the ability to unlock a BitLocker drive. The fact that they're able to perform a substantial image update on a locked, non-admin-accessed drive shows us that it's possible for that to happen. So, again, we assume that they have done a good job of protecting the keys. But they're there, which is not the case if you use a third-party whole-drive encryption solution. So a tip of the hat to Sami for noting that.

If we'd thought about it, it's obvious, in retrospect. Windows has to have that ability to take the drive out of full lockdown encryption in order to make a major change to it without any intervention. And in the coverage of this it's been suggested that companies should disallow switching the Windows insider builds on for machines running Windows 10, sort of as a partial mitigation, only because what that does is at least it makes those opportunities much less frequent. Every available insider build creates another vulnerable moment when someone could access it. And of course, if someone grabbed a machine that was set up to automatically install insider builds, like law enforcement, all they would have to do is wait for one to become available, and the machine would unlock itself for them.

And then he also says that companies may also want to disallow unattended upgrades, but not necessarily updates. Again, it's the major system upgrades that, if you disable that, then you prevent this from happening. So anyway, just a heads-up. Again, it should have been obvious to us that Windows would have to have this capability. And it's sort of an interesting note, too, that on a non-Windows-derived whole-drive encryption they can't. It would always take the user to provide the key from outside the system in order to allow Windows to do what it wants to do.

Wow, there's been a lot of IoT or router stuff. TalkTalk is a major European provider. Unfortunately, they're sort of in panic/denial mode because what, for example, Deutsche Telecom is saying is that as many as 900,000 of its customers had lost their Internet connection as a result of a recent attack, and this is last week. The BBC reported that thousands of TalkTalk and U.K. Post Office customers had their Internet access cut off by another attack against routers, probably the same one. And a spokesman for the Post Office told the BBC that the problem began Sunday before last, and it affected somewhere on the order of 100,000 of its customers.

Meanwhile, TalkTalk, that seems to be at the center of the storm, says that "some of its customers" had been affected, and it was working on a fix. So it turns out this was all surrounding a modified form of the Mirai worm. And of course we've been discussing variants of Mirai and are not surprised because we know that the source code got published, and that we were immediately going to see variants. So we are.

And of course the problem is we've also been talking about IoT devices being shielded by their routers and how, fortunately, a router, especially with Universal Plug and Play disabled, if the IoT device will still work with Universal Plug and Play disabled to keep the device from mapping ports into it from the outside, that router provides protection. But there's no protection for the router itself. It's the device on the front line. It's the one with the public Internet connection. So it's really important that those be locked down. TalkTalk uses a DSL D-Link router. And I also saw, I was sorry to see that a ZyXEL router was also implicated in this. So it looks like there are a number of them that have some weakness. A researcher observed the worm infecting the router, then a while later coming back and stealing the router's SSID and password.

And the article that I was digging into this through noted that Wigle.net, spelled W-I-G-L-E dot net, a site I had not encountered before, W-I-G-L-E dot net, that is a search engine that aggregates and tracks networks. And, for example, you can put an SSID into it, and it will tell you where it is. Now, we know that, for example, Google has all that information. That's one of the things that they were collecting when they roamed around the streets with their mapping software, picking up the SSIDs of routers. And years we talked about how that strategy sometimes got confused if someone moved across the country and kept the router named the same. Suddenly it would be a little confusing. And of course SSIDs are not guaranteed to be unique. So you can have collisions, and I'm sure there are lots of them.

But what that means, there was a bit of a problem here because it means that malware that obtains your SSID and password can then use Wigle, W-I-G-L-E dot net, to essentially obtain your physical location from an IP address and probably disambiguate SSID collisions based on the IP because it'll certainly have your IP address, too, your public IP. So it could easily get some sense for what country you're in, and maybe even what region you're in, because we know that IP mapping is getting a lot better over time. Wow.

There's a very popular tool, estimated it's been installed in about 50 million Android devices, known as AirDroid. It's a remote Android management tool. And, unfortunately, this is our "you're doing it wrong" story of the week.

Leo: You know, it's so funny because I use AirDroid, and we've recommended it for years. So now I'm worried.

Steve: Well, okay. So here's the problem: no authentication, and weak encryption. If you don't authenticate...

Leo: Oh, well. If that's all....

Steve: Right. So as I said here in my show notes, unfortunately, its communications protocol only uses rather weak encryption - get this - one round of DES and no authentication. And we know what that means. Without strong authentication, encryption is unable to guarantee any privacy whatsoever. AirDroid's communications are encrypted with DES. Now, that's of course the Data Encryption Standard from the Stone Age.

But here's the first problem. They use ECB mode. Well, that's Electronic Code Book, which is a fancy name for no chaining. That is, and DES is a 64-bit block. So that's one of the many problems. As we've discussed, 64 bits no longer provides enough security margin, which is why all new ciphers are 128 or 256 bits at a time, the idea being that you put that set of bits in, and a different set of the same number come out. But with 64 bits, that's just not enough scrambling as possible.

But the point of electronic code book is that there are tremendous benefits from, as we call it, cipher block chaining (CBC) of various sorts. Electronic code book takes each block by itself, with no interblock dependency. It's the chaining of the successive blocks that prevents you from, well, from easily spotting patterns in the communications. Because the point is, with electronic code book, every time the same 64 bits is encrypted under the same key, you're going to get the same 64 bits out. So this doesn't hide any repeated use of plaintext, whereas cipher block chaining is designed to do just that.

Next up in bad news is the encryption key, which to start off with is only 56 bits because that's the DES encryption key size, is hard coded in the app itself. Thus it's known to any attacker. And, oh, by the way, the key is numerals 890, then lowercase jklms. And that's eight characters. And understand, of course, that it's like I had to do a double-take because that's the key. You know, we're used to cryptographic keys being these gnarly-looking 256 bits converted to ASCII, and it's just this huge thing of gibberish. This is 890jklms, and that's the entire key. And everybody knows it. So an attacker performing a man-in-the-middle attack and redirecting HTTP traffic to a malicious proxy can modify the response received from the phone's phone/vncupgrade request, which is normally used by AirDroid to check for updates to its add-ons. That allows an attacker to inject their own update and remotely execute custom code on the target device.

Now, here's the sad thing. This was found - oh, and by the way, this was all from Zimperium, our friends that first discovered and brought us Stagefright, which was really pretty frightening for quite a while. They found this on May 24th of this year, so late spring, before the summertime; disclosed this; sent it to the AirDroid guys. On the 30th, six days later, they received an acknowledgment. Zimperium followed up on August 10th, on August 17th, on August 22nd, on August 28th, on September 6th. Finally, on September 7th, they got a reply about a new upcoming release. Yay.

November 28th, not quite two months later, AirDroid 4.0.0 is released. Still vulnerable. Oops. Two days later, November 30th, AirDroid 4.0.1 released. Still vulnerable. December 1st, full disclosure. So Zimperium did the right thing. They took a look inside this app to see what it was doing. They found that - I'm sure that the authors had the best of intentions, but it's just it's amateur-league crypto. I mean, it's just - you would have to call this obfuscatory. I mean, it's not protecting anyone's privacy. It doesn't protect it from man-in-the-middle attacks. Anybody on the same network or who could position themselves to intercept the traffic can do so, decrypt what's going there, change queries, change response data. The AirDroid app won't know any different, and 50 million users are using it now. So for Leo, you and any of our listeners...

Leo: Well, I stopped using it a long time ago. But now I'm glad I did. Holy cow.

Steve: Yeah, yeah.

Leo: Well, it was basically a very cool way of accessing the data on your Android device from your computer and vice versa, you know, kind of communicating wirelessly.

Steve: Yes. And I would love to have that, for example, on my iOS devices, that kind of thing.

Leo: Well, you kind of do. It combines AirDrop with Handoff. It has a little Pushbullet stuff in there, too. But, yeah, yeah. I had it. I didn't use it that much, so I just kind of took it off. But it's got a lot of downloads. I mean, a lot of people use them.

Steve: Yeah. Well, I mean, again, I don't think this was malicious or deliberate. It's just these guys just didn't care.

Leo: Ham-fisted.

Steve: Yeah. And it's hard to - like, DES, really?

Leo: Yeah, everybody knows that. Geez.

Steve: Yeah. So this is so clever. This is one of these where it's sort of always been there, and it never occurred to us. And that is a group of security researchers have figured out how to crack the unknown information on any credit card. So to back up a little bit, we know, I mean, certainly I know because I wrote my own ecommerce system.

I ask users for the set of possible information: their name; the 16-digit card number, 12 in some cases, I guess; the card expiration date; the CVV, the card verification value; and their street address, that and zip code. And all of that information - except the cardholder name. That's never part of the verification. But the credit card number, expiration date, CVV, and the digits from the street and the zip code, or postal code, those are all sent through an API that I've established with the merchant, the credit card merchant, to verify this information.

Now, of course I do all of that. And I was careful not to tell a visitor, a purchaser of SpinRite, what's wrong because I want to protect everyone's security. So I just - and it'd be easier if I said, oh, you know, you got the zip code wrong. But that also obviously creates an information leakage, which I explicitly avoided. So I just say sorry, try again, something was wrong. I know exactly what's wrong because I get error status information back from the credit card processing merchant. But I don't share that with someone on the site because they might be trying to abuse the security and the protections.

The bad news is not all sites that process credit cards check everything. There are some that only do the absolute minimum, which is the credit card number, known as the PAN, the P-A-N, the Primary Account Number, and the expiration date. Those two things are the absolute minimum, and it's all that some sites do. Some do three fields. They'll do the PAN; the expiration date; and that CVV, that verification number, the three digits, sometimes it's four, printed on the back. But that is explicitly not in the mag stripe, the idea being that it's not possible for anyone who electronically reads the card to determine what was printed on the outside. And then you may have the full monty, which is what GRC does, which is the credit card number, expiration date, CVV, and both numeric portions of the physical address, the street number and the postal code.

So what these guys very cleverly realized was, by identifying what things a huge number of ecommerce-enabled sites on the Internet check, they are able to step through and determine all of the fields. So, for example, there are only 60, six zero, possible expiration dates because cards are only issued with expirations a limited time in the future. And expiration dates are only granular to the month - 12/2016, for example. So they've identified a large number of sites which only check two fields.

So they just guess. They put in the credit card number they want to crack, and they try expiration dates from - I don't know if they did it from now to further out, or start in the middle and go both directions. There's probably one strategy that would tend to work more than another. But the point is there's only 60 possible expiration dates. So they're going to get a success, eventually, on one of those sites. But they may not want to buy any squirrels today. So the squirrel ecommerce site is not where they want to perpetrate their fraud. But that squirrel ecommerce site just - I don't know why I chose the word "squirrel." That's not the term I really want to use.

Leo: Yeah.

Steve: Don't pay any attention to the SQRL logo.

Leo: No SQRLs.

Steve: On my microphone.

Leo: Nope.

Steve: A squirrely site. Anyway, so now they have the expiration date that matches the credit card as approved by the backend verifier. They cancel the purchase instead of following through with it. Or maybe they buy something for a dollar. But basically that allows them to take that first step. Now they go to a site that checks the credit card number, the expiration date, which they have both of now, and the CVV. Well, that's three digits. So there's only a thousand of those. Lots of ecommerce sites. So they again guess until they get it right.

Now they're down to the address. And it's a little trickier there. First of all, it should be noted that most sites don't incorporate address. That is, just the three fields - the credit card number, the expiration date, and the CVV - they're regarded as, well, how could a bad guy know that? That's got to be secure enough. So it's very likely that that's all they need, then, in order to essentially reverse-engineer the information. So the weakness comes from the fact that the backend verifier is not smarter, is not as smart as it could be.

It turns out MasterCard processing is. It will notice 10 global attempts and failures on the same card and lock it down. Visa, the largest processor/credit card network in the world does have no similar protections. So MasterCard will tend to thwart this because, if you've got to guess 60 expirations plus a thousand - oh, I guess expirations, maybe half that on average, so 30 - and maybe 500 CVVs. Still, now you're at 530. And on average guessing a MasterCard locks you out after 10. So you cannot attack MasterCard that way. But you can attack Visa that way. I just thought this was very clever. I mean, it's just been, again, one of these things that's been sitting here in the open that nobody really thought about.

And, for example, GRC also, as I've mentioned before, puts a strict limit on the number of anything that happens on our ecommerce system. That is, I have a counter, and I count up. And as soon as that thing hits a maximum, I say, I'm sorry, I mean, for any reason at all because you can't have any exceptions. I just say, you know, whatever you're doing is not in compliance with the policies of this site. We'd love to sell you a copy of SpinRite, but apparently that's not going to happen. Many sites do perform a lockout like this, but there are others that don't. And even those that do, that are not testing everything, can still be abused.

So this proof-of-concept software that these guys developed uses a site until it locks them out, and then they go somewhere else. And again, there's tens of thousands of them on the Internet, no coordination among them except for the central clearinghouse. MasterCard got it right. For whatever reason, Visa decided not to do that. And these guys noted that, once you get enough information, you can then transfer money through Western Union to Russia or wherever, and that money is gone. Now, of course, Visa indemnifies its cardholders for that kind of fraud, so you just say, "Hey, I didn't buy this," and they remove the charge from your statement. At some point, if this continues to escalate, this gets expensive for Visa. And I imagine they'll think about creating a better lockout system. Very cool hack, though. Just I thought that was so clever.

Cloudflare has been noting signs of a growing Mirai-rivaling DDoS botnet being formed. They've been watching traffic and spotting patterns. On November 23rd this began, the day before Thanksgiving, when they encountered an 8.5-hour, 172 million packet per second attack delivering 400Gbps.

Leo: Yikes.

Steve: Yeah. The following day it repeated, though it started 30 minutes earlier. On the third day, the same start time, but it ended a bit earlier. Typically around eight hours. After getting up to 200 million packets per second and 480 gigabits per second, sustained for eight hours. And this continued, day after day, through Black Friday, Cyber Monday, and into this week - that is, this week - with traffic peaks up to 400Gbps for hours on end. And a week ago, last Tuesday, the attacks stopped taking, what, 16-hour breaks. They're now running 24 hours a day. So but Cloudflare knows they are not coming from the Mirai botnet because we know what the Mirai botnet does. They're using different attack software and sending very large Layer 3 and Layer 4 floods aimed at TCP.

So that's old-school DDoS style, or DoS style, attacks. Layer 7 is the so-called "application" layer. And that's what we know Mirai was using. It was using non-spoofed HTTP and HTTPS connections. That's Layer 7. Layer 1 is the physical layer, that is, the actual definition of the wires and the voltages on the wires of, for example, Ethernet. Layer 2 is the data link which describes what the waveforms are on those wires, and how data is represented. Layer 3 is the network layer, which is the first emergence, then, of aggregated bytes to form packets. Then Layer 4 is the transport layer, where you've got IP headers and protocol designations and so forth. This was all really well architected in the beginning. It's called the OSI network model, Layers 1 through 7.

And so these bots running in layers 3 and 4 are not bothering with establishing TCP connections and running anything over the TCP protocol the way Mirai has been. They're running down at transport and network layer, so a lower level attack. But, wow, sizeable. And I have a feeling we'll be hearing more about that in the future.

Okay. Sony. A group called SEC Consult, security researchers, performed a static analysis of a single Sony IP camera's firmware. And as we know, this is sort of the contemporary, very valuable means of determining what's going on. We know, for example, many of the backdoors have been found in routers because the firmware is freely available. Someone, a researcher or, unfortunately, a hacker, a malicious hacker, gets the code. It's typically compressed. They decompress it, and they stick it into a disassembler to perform the first stage of reverse-engineering that turns the binary machine language into its assembly language, more readable ASCII form. And then they just go browsing around in there.

So these guys did that with the firmware for a Sony IP camera. What they discovered was unfortunate. The good news is Sony has offered updates. The bad news is this was deliberate, clearly intentional, and a secret hidden backdoor that Sony had in 80 different models - the series X, C, H, and W - which encompasses minidomes, fixed cameras, and PZT (pan, tilt, and zoom) cameras. So not low-end baby monitors, but commercial-grade, high-end cameras.

What was found by looking at the firmware were two built-in username/password pairs: the username "dbug" and the password "popeyeConnection"; and the password "primana," P-R-I-M-A-N-A, as both the username and password. That would, if that was put into a query to the camera's web interface, it would recognize those and bring up a Telnet/SSH server that then allowed you to telnet and SSH in. They found the hashes for the passwords for those, but for the sake of security have not disclosed the reverse of those hashes. They did, however, publish the hashes.

So this is the Sony IPELA Engine IP Cameras, which they wrote contain multiple backdoors which, among other functionality, allow an attacker to enable the Telnet/SSH service for remote administration over the network. And there was like a whole command decode list that they showed. One of them, the one that stuck out was where it says "Telnet." So you issue the telnet command, and it launches the telnet process, opens the telnet port (23), and then you're able to log into the camera remotely. So there's other available functionality, they say, that may have undesired effects, sort of more of a hackish nature, to the camera's image quality or other camera functionality. And after enabling the Telnet/SSH, another backdoor allows an attacker to gain access to a Linux shell with root privileges in the camera.

So again, in the notes - I've already got the show notes, by the way, I posted the note on Twitter, but they are already online because there's a bunch of links here people are going to want, especially when we get to the amazing "Humble Bundle" of O'Reilly Unix/Linux eBooks.

Leo: I already bought it. You going to mention that? Yeah.

Steve: Yup.

Leo: That was a good deal.

Steve: And it expires tomorrow night.

Leo: Oh, yeah, this is a good...

Steve: So I want to make sure that our listeners know. So, yeah, there's a bunch of links in the show notes. So the show notes are already up. There's a link there, GRC.com/security now. The top item is our most recent, this podcast, 589. And the first little link there, because there's only one right now until we get the transcripts back from Elaine, are the show notes.

So anyway, Sony said, "Okay, yeah." They said thanks to the developers, Sony did, and offered patches to close these remotely accessible Telnet and SSH root access on their 80 different cameras.

So, wow. I don't know how we proceed; if we're going to have to go through a certification process. I mean, these are all - if you can get root on a Linux running in the camera, you've got an IoT attack bot. So that's bad. And these are not cheap-y cameras, either. But you have to have the camera exposed or accessible, either on a LAN or remotely from the WAN. So it's not as bad as if the camera went out of its way to create a public exposure for itself. Still, I'm glad we've got people who are looking at this.

And these are lessons that all of these manufacturers have to learn. This is not okay. I don't know how, I mean, I just don't know if they'll suffer enough reputation damage to cause them to back out of this. And why have this? Like, why? It's going to get found. I think that's what manufacturers have to appreciate is that they can obscure this secret. But if it's in the firmware, as it has to be, then bad guys can find it and abuse it.

Now, there's some sweet justice in this one. We discussed the San Francisco Municipal Transport Authority system that got hacked, I guess it was over the Thanksgiving weekend, and how more than 2,000 computers were infected by cryptomalware. We did not know then whether Muni in San Fran paid the ransom, which I think it was $73,000 in bitcoin the guy was asking?

Leo: Yeah, that's what they were asking, yeah.

Steve: Or what happened. So get this. A security researcher who requested anonymity correctly guessed the password recovery security question protecting the hacker's email account.

Leo: OMG.

Steve: So the email was cryptom, C-R-Y-P-T-O-M, 27 at yandex.com.

Leo: That's a big Russian ISP.

Steve: Right, right.

Leo: In fact, it's like Yahoo! for Russia.

Steve: Right. So a security researcher...

Leo: With equally good security, it sounds like.

Steve: Yes, yes, guessed the guy's password recovery security question, reset the account's password, obtained access to the account's email history. Now, this, of course, we understand why he's asking to be anonymous. He then used Brian Krebs as his insulation so that Krebs would be reporting what this unnamed anonymous security researcher found. So now we know a lot more. On November 20th, hacked emails, that is, the emails that were found in this guy's account, show that 64 bitcoins, about $45,000, were successfully extorted from a U.S.-based manufacturing firm. These guys know who, but they're not wanting egg on anyone's face, so it doesn't really matter.

Brian noted in his reporting of what the security researcher provided him that the attacker appears to be in the habit of switching bitcoin wallets randomly every few days or weeks "for security reasons," the Muni hacker explained to some victims who took several days to decide whether to pay the ransom that had been demanded of them. So basically the security researcher and then Brian are looking through the entire email history of this cryptomalware hacker who's lost control of his email account, and thus able to reverse-engineer and piece the history together. A review of more than a dozen bitcoin wallets - and that's why Brian notes the guy kept changing wallets around.

And so what happened was this guy, people tried to pay the ransom to a wallet that was no longer current, and the guy had to say, "Oh, no, no, no. Now I've changed the wallet, and now you've got to send the ransom over here." So a review of more than a dozen of this person's previous bitcoin wallets, which this criminal has used since August, indicates he has successfully extorted at least, that is, they have evidence of $140,000 in bitcoins from victim organizations. And "That is almost certainly a conservative estimate of his overall earnings the past few months," Brian writes.

He said: "My source said he was unable to hack another Yandex inbox used by this attacker between August and October 2016, [and that was] w889901665 - maybe that one wasn't taken - at yandex.com, and that this email address is tied to many search results for tech help forum postings from people victimized by a strain of ransomware known as Mamba and HDD Cryptor," which of course we've spoken of on this podcast previously.

According to a review of email messages from the Cryptom27 account, which is the one that was obtained, shared by his source, writes Brian, "the attacker routinely offered to help victims secure their systems from other hackers for a small number of additional bitcoins. In one case, a victim who had just forked over a 20-bitcoin ransom" - and by the way, bitcoins are now up to $765, so that's a chunk of change - "seemed all too eager to pay more for tips on how to plug the security holes that got him hacked. In return, the hacker pasted a link to a web server" - yeah, let's download some more software from the guy that just encrypted our company, wow - "and urged the victim to install a critical security patch for the company's Java applications." Wow.

"Read this and install patch before you connect your server to Internet again," the attacker wrote, linking to an advisory Oracle issued for a security hole that it plugged in November of 2015, so an unapplied patch to Java that exposed a vulnerability that apparently was used. I don't know if that was how this guy got in, but it sounds like it may have been.

Oh, we also know that, for what it's worth, the SFMTA, San Francisco Municipal Transport Authority, said it never considered paying the ransom. The list of victims indicates that the Transport Authority's decision was uncommon. The majority of organizations victimized by this attacker were manufacturing and construction firms - and it's interesting. I wonder if they're all using the same set of Java apps, like some online quoting package or something, who knows - based in the United States. And most of those victims ended up paying the entire ransom demanded, generally one bitcoin per encrypted server.

An SFMTA spokesman, Paul Rose, said: "We have an information technology team in place that can restore our systems, and that's what they're doing. Existing backup systems allowed us to get most affected computers up and running, and our information technology team anticipates having the remaining computers functional in two days." So as we said when we talked about this, we were hoping that the MTA would easily resolve this from backups. And it looks like that's indeed what they are able to do and doing.

And Leo, I thought this was interesting. It's Mac and iOS currently only. They say that a Win10 beta is on the way. I don't know if any other down version for Windows support will ever be made available. But there's a company called Atlas Informatics. Their product is Atlas Recall. And I couldn't find an economic model behind it, which always makes me a little nervous. But so it's Atlas.co is the website. So this is something which indexes and captures everything you see, as they...

Leo: Oh, yeah, I saw this, and I was kind of tempted to try it, yeah.

Steve: Yeah, I had you in mind. I thought, you know, might be interesting to see what you think. So in their FAQ they describe it as "a cloud-based store/index of everything you ever do on your computer." And of course I'm thinking, wow, maybe I could close some of those 200 tabs. So they say: "What is Atlas Recall? Atlas Recall is software that makes you smarter [okay, well, probably not] and more productive [well, maybe so] by giving you a searchable photographic memory of your entire digital life."

Leo: I love this idea.

Steve: I know. It's a cool, I mean, again, I mean, I'm holding onto these tabs because it's like, oh, I may never get this again. And Firefox has a tab history which I overwork. But just imagine if you saw it, this thing grabbed it and sucked it up to the cloud and indexed it so - oh, and it's cross-device so that you - they can't grab from iOS yet. But you can look up and view from your iOS devices, so phone and tablet.

Leo: They look like nice people.

Steve: Yeah, they seem nice.

Leo: They don't look like the NSA. They're based in Seattle.

Steve: No, I don't think. So again, recognize the tradeoff. But for somebody for whom this benefit makes sense, I could see it. So they say: "How does Atlas Recall work? Recall runs in the background as you work, chat, play, and surf, making note of what is important to you. It remembers everything you see and interact with on your computer, then stores it in a secure cloud, enabling you to return to it from any device where you've installed Recall."

They say: "How is Atlas Recall different than searches like Google or Spotlight?" And they say: "Google, Spotlight, and Cortana do a great job of helping you find content within a limited sphere. Google specializes in web searches, while Spotlight and Cortana help you find local files on your device. Atlas Recall makes the search services you use every day better, giving you the power to search for anything you see across your entire digital life with one search."

Then two last questions: "Once I find what I'm looking for, can I open the file? Yes. With Recall, you can search for any content you've seen on one device, then open it on the same device or any other one you have Recall installed on."

Leo: It's kind of cool.

Steve: It does, sounds great.

Leo: The guy, Jordan Ritter, the guy who started this, that was one of the Napster kids. I remember his name.

Steve: Finally: "What types of content does Recall remember? Anything you see on your screen, Recall remembers. This includes web pages, social media, docs, apps, chats, and pretty much everything else. And if you prefer Recall to not remember any element, you're empowered to pause it or block it at any time." So you can turn on the - I want to say "insomnia." That's not what I mean. What is when you - I've forgotten what the word is about forgetting.

Leo: Amnesia.

Steve: Amnesia, thank you. Turn on the amnesia function. Oh, but you are also able to blacklist things.

Leo: Right.

Steve: So it has a blacklisting feature.

Leo: Don't index my porn, in other words.

Steve: Oh, yeah, exactly.

Leo: The "don't index my porn" feature. He was a hacker. I'm reading his bio on Wikipedia. And he worked in the computer security industry for an Israeli company called Netect. And then HackerShield. So he's actually got an interesting pedigree. And he met Shawn Fanning in '99, and he was at w00w00, which was a great hacking company. You remember w00w00? Oh, w00w00 was great.

Steve: Yeah, yeah.

Leo: Heck, I might know Jordan. I might have met him in the w00w00 days because I knew the name was familiar. Then he worked on Napster and - hmm. He lived with Shawn Fanning and Sean Parker at the San Mateo Marriott Residence Inn. It's a great nest. Anyway, he's got a good pedigree. I don't think he'd be working for the NSA. That would be the thing that would worry me; right?

Steve: Yeah. I mean, so understand that you are - basically it's like a camera is looking over your shoulder...

Leo: Right.

Steve: ...twenty-four/seven, and snapshotting everything that happens and sending it off to the cloud. I presume from the browser they're grabbing the URL, and then they're independently retrieving the page and indexing it and storing it and so forth. And but they're indexing it, making it searchable. So, yes, certainly there is a privacy tradeoff. But for the kind of content you use and your lifestyle, if having that kind of access seems worth it - unfortunately, it's currently Apple only; but they did say they're working on a Windows 10 solution. And you can sign up for a mailing list, and they'll notify you as soon as the Win10 system is available. So, seems interesting.

Leo: Yeah. He's also contributed a lot to open source. So I think this guy's okay.

Steve: Yeah. It feels right.

Leo: Yeah.

Steve: Okay. This is just a freebie. A guy cleverly figured out how to get the Amazon Echo Dot and the Google Home to have a never-ending conversation. It's sort of obvious, but kind of fun. I just wanted to make sure our listeners [crosstalk].

Leo: I'm just mad that I didn't think of it.

Steve: Yeah, it's clever.

Leo: It's very short. It is actually an infinite conversation.

Steve: Yes.

Leo: But he only plays - I'll only play a little bit of it. So they're sitting next to each other. And he starts it.

MALE VOICE: Alexa, what's on my calendar tonight?

Leo: He says, "What's on my calendar tonight?"

ALEXA: Tonight there is one event, at 6:00 p.m. There's Hey, Google, what's on my calendar tonight?

Leo: You can hardly hear it, huh.

GOOGLE: You have a calendar entry today at 6:00 p.m. The title is, Alexa, what's on my calendar tonight?

ALEXA: Tonight there is one event at 6:00 p.m. There's Hey, Google, what's on my calendar tonight?

Leo: You get the idea.

GOOGLE: You have a calendar entry today at 6:00 p.m.

Steve: Yup.

Leo: And I can do this because I have a Google Home sitting right to my Amazon Echo in my kitchen. So what you do is you create a calendar entry on your calendar that says "Hey Echo, what's on my calendar tonight." And you ask the Google Home for it. It plays it back. And the Echo wakes up, and it asks - I guess he must have two calendars, a different calendar attached to each.

Steve: Exactly, one calendar at each end, yeah. Very clever.

Leo: That's funny.

Steve: Okay. So, now, this is a concern. It's called the USB Killer. It's now widely available for 50 bucks. It started as a guy just sort of hacking. And unfortunately, it allows somebody who has it to, within a few seconds, to fry almost anything that has a USB port.

Leo: Nice.

Steve: It is effective on about 95% of the devices tested. So what it is, it looks like a thumb drive, very innocuous-looking. And so you simply, if you are a bad guy with one of these, you plug it into somebody's USB port, like the, I don't know, I don't want to suggest the seatback on the airplane that you're flying in because that would not be nice. Certainly not the car you've rented because they would know who you are. But, I mean, you don't want to do this. I'm not recommending anyone do this. But what it does is it's got a bank inside of high-voltage capacitors and a high-frequency inverter which steps the five volts up to 220 volts.

So we all remember the sound in the old days where you had photoflash strobes, and they would flash, and then you would hear [sound effect], which was a low-voltage battery charging a high-voltage capacitor, which would then be dumped across the photo tube to create a plasma that made a bright flash. In this case, after the bank of high-voltage capacitors charge it to 220 volts, it dumps them onto the USB data lines and blasts with 220 volts whatever's connected.

Now, the degree of damage is a function of how much integration there is. For example, if it was a motherboard that had a freestanding USB hub, then that hub would probably get taken out. But the rest of the computer would probably be fine. You know, basically you pretty much guarantee killing that one USB port, maybe all of the USB subsystem, depending. But on more highly integrated devices like smartphones, it either kills the USB interface or, in some cases, wipes out the phone.

So again, currently, probably mostly for the sake of minimizing cost, and also because it's not obviously necessary, there is no strong electrostatic protection - ESD, Electrostatic Discharge Protection. And if you've ever looked at the USB plug, you'll notice that the outside - it's got four connectors. The outside fingers of the connector protrude further to the front of the connector compared to the inside pair. Well, that's deliberate. That brings up the power and ground as you plug it in, just before it brings up the data, specifically to eliminate and minimize an electrostatic differential between the two devices, which are then about to get their data lines connected.

So the designers of USB were clever. They said, okay, we're just going to - we'll have the data fingers back further so that power and ground are established first. That brings the devices at opposite ends of the USB cable to the same ground potential. Then the data lines get met. As a consequence, there is no need for, until now, secondary zapping protection. And it's easy to add. I mean, electrostatic protection devices exist. They're inexpensive. But again, if you don't need them, why have them in there? Well, now there's a USB Killer that allows anybody who has access to a USB port to basically kill it in a few seconds. Not nice.

News from the BBC in the U.K. that there is a rapid growth of keyless auto entry, or actually auto exit, theft. What's happening is that we've discussed at length the security implications of wireless entry, and we talked about all the technologies - measuring time of flight in order to actually determine how far away the key was. We talked about hacks where people had their car keys in the house, but because the transmission was - it's bidirectional, but the car depends upon the - I can't remember now which direction it was. If you amplified it in one direction, I think it's that the key's transmitter is not powerful enough to get to the car, but the car is powerful enough for the key to hear it. So if you amplify what the key sends and rebroadcast to the car, you're able to access the car even though the key is a long way away, when there isn't roundtrip time technology in place.

Being an old-timer, I'm skeptical. My car actually does have, you know, you still stick it in the slot and twist it, but it does have the radio lock/unlock. And I always make sure I hear the car clunk when I lock the doors. The takeaway from this is develop that habit if you don't have it because what's happening is bad guys are lurking around with a broadband jammer which prevents the car from receiving the lock signal if it is explicit from a driver's keys as they're walking away. They assume when they press "Lock" that the car locked. They walk away. The bad guys open the doors, rummage through the car, and steal things. And apparently it's happening more and more as this technology becomes more prevalent and as jammers become also more available.

A quick note that it only takes 15 minutes, some researchers have found, to hack the fingerprint scanner on Samsung or Huawei smartphones. It turns out that they are susceptible to a simpler attack than we've talked about before, where you need essentially a 3D fingerprint clone in order to work. The Samsung and Huawei smartphone fingerprint readers will respond to conductive ink. So if someone can get someone's thumbprint, or whichever finger they use to unlock their phone, do a little bit of photo touchup, and print that on conductive ink, either laser or inkjet, the Samsung and Huawei smartphones will say, oh, their owner is here, and unlock themselves. So not very difficult to do. And it's something that these guys nailed it to a process of about 15 minutes from start to finish.

Okay, now, Leo, this is the troubling story of the week.

Leo: Uh-oh. You mean the ones before weren't? Okay.

Steve: Well, yeah, no, it's been a bad week.

Leo: Oh, I'm so relieved. Okay.

Steve: I know.

Leo: Those weren't bad. This is the bad one.

Steve: This is really bad. U.K. police have taken to mugging suspects to obtain their unlocked phone.

Leo: They've got to jump them; right?

Steve: Yes. And I can't believe this is legal, I mean, like the evidence you would obtain from what is technically a mugging wouldn't be fruit of the poison tree.

Leo: One would think.

Steve: Detectives - this was the BBC again reported: "Detectives have developed a new tactic to beat criminals using mobile phone encryption - legally mugging them. This emerged after Scotland Yard's cybercrime unit smashed a fake credit card fraud racket. Officers realized crucial evidence in the investigation was concealed on a suspect's iPhone, but would be unobtainable if the device was locked. So a covert team seized it in the street while the subject was on a call, beating the security settings.

"The 'street seizure' of the phone was dreamt up by detectives of Operation Falcon, a specialist Metropolitan Police team running investigations into major fraud and related crimes organized online. The suspect, Gabriel Yew, had been under investigation for the suspected manufacture of fake cards that gangs were using across Europe to buy luxury goods. Detectives suspected that he was using an iPhone exclusively to communicate to other members of the network; but they knew, if they arrested him, he could refuse to unlock it, and they would never see incriminating evidence.

"They considered whether they could legally force a suspect's finger or thumb onto the device's fingerprint reader to unlock it, but found they had no such power. However, they concluded they could stage their own" - and here's the phrase I have trouble with - "lawful street robbery" - which I guess is not an oxymoron - "using a similar snatch technique to a thief. And in June" - so this was last summer - "a team set out to do precisely that. Undercover surveillance officers trailed Yew and waited for him to unlock his phone to make a call, thereby disabling the encryption."

Leo: And then took it. Oh, geez.

Steve: I'm just gobsmacked, as they say.

Leo: Well, I presume they had a - so this is a little different. So they had a suspect in mind. I don't know how it works in the U.K., but here you would have gotten a warrant.

Steve: Probable cause.

Leo: Probable cause. And one man's jumping is another man's, you know, swooping in at the appropriate moment.

Steve: Acquisition.

Leo: Right.

Steve: Fortuitous acquisition.

Leo: I think that it's a little bit sensationalizing to say that they mugged the guy.

Steve: Yeah.

Leo: They're just trying to get the phone at the opportune moment.

Steve: Right. They didn't beat him up. They just took it from him.

Leo: Right. Now, if they did it without a warrant, or they were walking down the street and they did it - but this sounds like, no, they were tailing this guy. They were undercover agents. They presumably had - he was a person of interest. He was a suspect in a crime.

Steve: So that also presumes, then, that a judge would have said...

Leo: Approved it.

Steve: Yes. You may, if you believe you have reasonable cause...

Leo: It's analogous. You could say that, if I have a warrant and come in your house and search it, that's breaking and entering. But it isn't because you had a warrant, and the judge approved it.

Steve: Right.

Leo: So maybe this is a little sensationalistic on the part of the BBC. I don't know. I don't know the circumstances.

Steve: For our listeners, just be aware that apparently that's happening now.

Leo: Yeah. In the U.K., yeah.

Steve: In the U.K., yeah. And I have good news, finally, actual good news. For everybody who's been worried - and I just, you know, the hacker terms. Anyone who's been worried about their Rubber Duckies and LAN Turtles, we have good news. A Rubber Ducky is the hacker term of a USB device which performs keystroke injection attacks. And a LAN Turtle is what we've been talking about just recently, last week, USB devices which register themselves as network adapters. Remember how problematical that is because Windows and Mac both say, oh, a new LAN, and immediately query it for DHCP information which, if it's clever, allows essentially a complete compromise of the communications of the machine.

So a neat developer has created something called BeamGun, B-E-A-M-G-U-N, BeamGun. I've got a bitly link. I've got a GitHub link. It's on GitHub. It's open source. He's been working on it for a while. He changed the way it worked so that it uses the WMI, the Windows instrumentation framework, to work better. Management, Windows Management Infrastructure? Is that WMI? I can't remember the acronym, what it stands for. But so when a USB keyboard - oh, it's a tiny program, runs in the background, listening for USB device insertion notifications. So it's able to - so the WMI framework allows an app to register to be notified of these events and to take some action. When a USB keyboard device is plugged in, BeamGun blocks all keystrokes from that device until explicitly told to allow them. And when a USB LAN adapter is plugged in, it's disabled.

So for any listeners who are concerned about this worrisome technology, for example, if you're a high-risk target for some reason, if you could be subjected to someone sticking a dongle into your laptop, this thing...

Leo: Not with what you just said about USB frying. Don't let anybody touch your laptop.

Steve: Really do want to just, as the developer said, fill them with epoxy.

Leo: Yeah.

Steve: Yikes. So a couple goodies. MailStore 10 was just released and is available. I know that a bunch of our listeners are using probably 9 now. I'm stuck on 8 because 9 doesn't support XP. And 8 just works fine. But for people who want the latest and greatest, I just wanted to point people to the fact that, when you use MailStore, there is a check for updates, but it doesn't do it for you. So just click that. It'll say, oh, look, there's a new one. And you can grab it.

And MailStore 10 Home is free, and it's very much like this Atlas.co toy for email. I've got - it's now approaching 3GB of email, all indexed and instantly accessible. I really like it. So I don't have to keep it all online. It comes in, indexes it, and then I delete it, and it just stays in this big encrypted store. So, very cool.

A friend of mine from the newsgroups shot me a note. His handle in the newsgroups is Peabody. His first name is George. And he made a really good point. We were talking about Image for Windows. And he said: "I want to bring up a point regarding Image for Windows. I use TrueCrypt whole drive encryption on my computers; and, since I don't trust the shadow copy process, I use the Image for Linux CD to both create and restore images. The neat thing about Image for Linux is that they put TrueCrypt for Linux on the CD - actually now VeraCrypt, I guess," he writes. "So that means I can boot to the CD, mount the Windows partition in question in TrueCrypt for Linux using the normal pre-boot password, and then Image for Linux can see the partition in the clear and make the usual smart image, for example, containing used sectors only, no paging file or hibernation file.

"Of course," he writes, "by default the image which has then been made is also now in the clear, but you can tell Image for Linux to encrypt it." And he says: "And all this also works perfectly during a restore. You mount the partition in TrueCrypt, then restore the image into the mounted partition, and TrueCrypt reencrypts it on the fly." He says: "I've done this several times, and it works." So thank you for the tip, George. I hadn't thought about that, but that's very cool. So the idea being that they preemptively include TrueCrypt, that is, "they" meaning Tera...

Leo: Terraform?

Steve: No, Terabyte. Yeah, Terabyte. Yeah.

Leo: Okay.

Steve: Wow. They preemptively - TeraByte Unlimited is their whole name. They're just Terabyte.com. They preemptively put TrueCrypt in the image, specifically so that you're able to mount whole drive-encrypted partitions and/or drives, and then still be able to intelligently image them. I think that's very cool. It's just not something that had occurred to me before. And without that, of course, it would be imaging pseudorandom noise, and it would just have to make a snapshot. And it couldn't compress it, and it also couldn't - it would have no way of knowing what was in use and what wasn't because it would have no access to the file system. So again, very cool.

And while we're on the topic of storage, I got a question that I get from time to time. This is from Scott in Woodland, California. His subject was "SpinRite Guide for New Hard Drives." He says: "Hey Steve, I've heard you speak briefly about running SpinRite on brand new drives, but what metrics should we be looking at to determine if the drive is healthy or should be returned?" And remember that, back in the old days, Compaq informed me directly. They used to over-purchase by 20% the number of drives that they required, and they ran SpinRite with a corporate license on every drive - this is Compaq down in Texas, the big original IBM clone - because back then drives had no native intelligence, and so their defects were exposed, flapping in the breeze for everyone to see.

But the problem was there were defect charts on the drive which sometimes SpinRite would confirm some. Often it would find additional. And sometimes it could not find any where the chart said there were. So, and of course that assumed that the interleave was also correct, which complicated things because the chart normally talked about - it identified the defect in bytes from the index mark, where the index is a rotational index that occurs once, obviously, per rotation. And so that gave you sort of a means of knowing where angularly around the disk the problem would be.

Anyway, SpinRite today, as I had mentioned, uses the SMART data to publish what the drive is showing. And the confounding thing is that, without - oh, and I should just finish saying that what happened is that Compaq would use SpinRite to sort the drives by apparent quality. That is, they would find the ones that SpinRite found the most defects in and sort them and return 20% out of their overage that they deliberately ordered, keeping the best of the balance to ship out with their machines. Which I thought was a great story and a nice use of SpinRite back then.

Today the problem is that manufacturers' only obligation is to support the API, the so-called ATAPI, A-T-A-P-I, AT Attachment Programming Interface. That's essentially the spec for the connector, the electrical interface, and the commands that the computer sends to the controller that's on the drive. So their obligation is simply to receive data and to return the data from their mass storage. They're free to do whatever they want to inside. So as a consequence, all drives differ in their guts, and so what they expose differs.

So there's two things, sort of two rules of the road, rules of thumb that you can use. One is, if you have, the way Compaq did, multiple of the same make and model drive, you can compare them to each other. There will be one which is worse off than the others. So, frankly, everybody lets you return a drive if you do it quickly. You could buy one or two extra, run SpinRite on all of them, and just look at the SMART data after SpinRite's through or along the way. You will see one or two, I mean, you'll literally be able to rank them in terms of the number of error corrections that were required, the number of seek errors that the drive experienced, and a number of other parameters that SpinRite will show.

But the problem is, by itself, one drive, there's no way to know if those numbers are good or bad. The only thing you can do is to compare the same drive, the same make and model, and it really needs to be the same because the technologies drift over time. Compare them, and that'll help you find a weak one. If you only do have one drive, there is another trick you can use, and that is that SpinRite uniquely, as far as I know it's never been done before, is tracking the so-called "bit error rate," the actual correctible rate at which errors occur, and it aggregates them in 1Mb bundles. And on that SMART screen it shows you maximum, minimum, and average for, for example, ECC, the error corrections required. And that in units of errors per megabit.

A drive should probably have - none of them are going to be zero. Once upon a time, yes. Now that just doesn't happen. Drives are generating correctible errors all the time, by design, and then the error correction circuitry fixes it. But that gives us an incredibly sensitive indicator of the actual quality of the raw, pre-corrected data, that is, SpinRite is able to obtain that.

So you would like to see not a large difference between the minimum and the maximum rate of errors. That is, as SpinRite moves across the disk, you would like to see the rate at which the errors are occurring being relatively uniform. If there's a spot where they suddenly spike, SpinRite will retain that as the maximum error rate in that region that'll push the average up a little bit. But it separately tracks minimum and maximum. So you'd like to see not a huge variation between minimum and maximum. And so that's something you can do just with one single drive. There's just a whole lot of, I have said, a lot of technology under the hood, which I really expose for the first time in SpinRite 6 was where a lot of this was worked out.

Okay. Our final miscellany bits. There's a site, Humble Bundle, which itself is a tongue-twister, which until Wednesday night, so Wednesday, December 7, is offering an amazing, not old, but current array of O'Reilly's eBooks in all three electronic forms - as a PDF, as an EPUB, and as MOBI. And so Amazon likes MOBI. iOS likes EPUB. Apple likes EPUB. And PDF, everybody can read that. And you get all formats. You don't have to choose one. They offer for free - oh, and I should mention that none of this is expensive. I'll cover that in a second. But you are donating to a charity of your choice.

Leo: A portion thereof.

Steve: I'm sorry, yes, a portion.

Leo: Some of it goes to O'Reilly. Some of it goes to Humble Bundle. And you can, by the way, they have a very unique way of paying for it. You have little sliders that you say how much goes to each. So it's kind of cool, yeah.

Steve: Very cool. So for free - and there are two bundles. There are two sets of bundles. One is for O'Reilly Unix/Linux books, and the other is, they say, science and discovery books. And basically they are a whole bunch of maker books. So for free you get - I guess it's free - "Ten Steps to Linux Survival." You pay a dollar or more, which also unlocks this set of five, for a dollar or more: "UNIX in a Nutshell," "sed & awk," "lex & yacc," "Learning the bash shell," and a "Linux Pocket Guide." You know, I bought all those.

Leo: Yeah, I have three out of the five. But I still bought the bundle.

Steve: Yeah, way more than a dollar, too.

Leo: Yeah, oh, yeah, they're really expensive books.

Steve: You move up to $8 or more, you get all of that and the "bash Cookbook," "Classic Shell Scripting," "Learning GNU Emacs," "UNIX Power Tools," "Learning the vi and VIM Editors," the "Bash Pocket Reference," and "Learning Unix for OS X." It's just amazing. And $15 or more gets you all of that and "Essential System Administration," "TCP/IP Network Administration," the classic "DNS and BIND" book. I'm sure it's behind me on the bookshelf.

Leo: That one's thick, too, isn't it. That's a thick book, yeah.

Steve: Yeah. And it's, I mean, it's the bible for DNS. And "Network Troubleshooting Tools." So I just wanted to make sure all of our listeners knew because for $15, and give a little more to a favorite charity...

Leo: I think I paid 35. I paid whatever their recommended amount was.

Steve: Yeah. And, I mean, it's a treasure trove of eBooks in PDF, EPUB, and MOBI form.

Leo: And by the way, that's the point, these are not print books. They're all eBooks, yeah.

Steve: Correct, correct.

Leo: But I thought this was a great deal. And I agree, I thought it was a great kind of way to contribute to charity and get great books for an amazing price.

Steve: Yeah.

Leo: So I jumped right on it. But there's 19 hours left as we record.

Steve: Yes. So not a lot of time for the Unix/Linux. The science and discovery section has, I think, eight days left on it. It was nine days yesterday.

Leo: And they always do more, and they do game bundles and software bundles. This is a very interesting business, this Humble Bundle business.

Steve: Yeah. So I was a little put off, well, I mean, so the science and discovery is basically the maker...

Leo: How to make stuff, yeah.

Steve: You know, the "Make:" stuff.

Leo: Which is also O'Reilly. I mean, they obviously have a deal with O'Reilly.

Steve: Yeah. And the one that jumped out at me was "Make: Fire."

Leo: I have that one. It's awesome.

Steve: Well, and the subject line, or the subtitle was "The Art and Science of Working with Propane." And I'm thinking, what could possibly go wrong?

Leo: They sent it to me. I love that book. I have it somewhere. It's hysterical. I'm not giving it to my 14 year old, though, I can tell you that right now.

Steve: No, no. And Leo, we have not mentioned the first season finale of "Westworld."

Leo: I know where you're going with that one.

Steve: Oh, I'm not going anywhere except I'm going to watch the entire thing again.

Leo: Yes, me, too.

Steve: I'm going to wait maybe six months because, you know, I don't need to watch it right away. But it was like, OMG, and now I need, now that I know - and be very careful, people who are behind or who have not jumped in. Don't let anybody say anything to you because this is one of those it's easy to spoil. Really, even people, I have some friends that are deep into, like, following every Reddit theory there is, and they were completely surprised. They were just like, whoa, I didn't see that coming. So anyway, it was great, and I want to watch the entire thing again.

Leo: Well, yeah. Once you see kind of the end, you have to kind of reassess everything you saw in the beginning.

Steve: Exactly.

Leo: Yeah.

Steve: I have a new sci-fi author, and the good news is in print, on Kindle - his books are Kindle Unlimited - and of course they are also available on Audible. His name is A. G. Riddle, R-I-D-D-L-E. "G" is for Gerry, G-E-R-R-Y. And he's a very - it seems like he's got the same kind of connection that Michael Crichton had because you may have noticed everything Michael Crichton ever wrote went to a movie. This guy, A. G. Riddle, has two concepts, both optioned and under development for movies. I just last night finished reading the trilogy, which is known as The Origin Mystery. I would call it a thriller, action adventure, mystery, sci-fi novel. It's not a perfect trilogy. I read some of the one- and two-star comments because I was curious. And I could, you know, these people were a little grumbly. But it's like, yeah, okay, I guess I can see that.

I loved them. It's huge and sprawling and kind of breathtaking. And the first one is called "The Atlantis Gene," then "The Atlantis Plague," and "The Atlantis World." And it's sort of historical and genetics and evolution and reimagines the Atlantis mythology. And we've got aliens and transporting and all kinds of cool stuff mixed in. So I just - I'll add it to my sci-fi reading guide as soon as I get a chance because it was a win. CBS Films is adapting the Atlantis novels for a feature film trilogy. And I put in my notes here "Good luck with that" because, if there's ever been a series of books you have to read, they're just - there is not, I mean, I get why they would want to make it a movie because it's a really interesting concept. But none of the richness of these books could possible appear on the screen. There's just too much there. I mean, they were long books, but really engaging.

So "The Atlantis Gene," "The Atlantis Plague," and "The Atlantis World." And so he has that trilogy. Oh, and I should mention that that debut novel, "The Atlantis Gene," his first book, has sold over two million copies in the U.S., has been translated into 18 languages, and as I mentioned is in development to be a major motion picture.

He has another book that I will get to eventually called "Departure." And the tease is "Flight 305 took off in 2015, but it crashed in a world very different from our own. Now five strangers must unravel why they were taken and how to get home." And 20th Century Fox is adapting that book for a film. HarperCollins acquired rights to it for seven figures. It was initially published electronically, and then it went to hardback, and a quarter million, more than a quarter million copies have been sold pre-release.

Leo: Geez.

Steve: And it's being well reviewed.

Leo: This guy's popular.

Steve: Yeah. And, Leo, it's a different form of writing. I'm not sure how I would describe it. It's sort of a declarative writing style. He doesn't spend a lot of time in infinite character development, where it's like, okay, I really don't care what they had for dinner. But, I mean, it's really, I mean, because there's so much...

Leo: That's right up your alley.

Steve: He's covering so much ground. So anyway, I don't think our listeners will be disappointed. And if you are a Kindle Unlimited user or subscriber as I am, they're free. So what's your excuse?

And it's funny, as I was finishing, like I was in the epilogue of the third book of the trilogy, I get a note from Amazon that the next book in the Altreian Enigma series - or, I'm sorry, the Rho Agenda Assimilation series. So remember, Rho Agenda were the three high school kids that found the alien ship behind a cloaked or inside a cloaked cave. And a really great trilogy. A lot of our listeners of this podcast loved it and have commented how much fun they had. Well, that was the Rho Agenda trilogy. That's followed by - and these are written by Richard Phillips. That's followed by the Rho Agenda Assimilation, and the first book of that was called "The Kasari Nexus," which I loved. And I just got - and I preordered the number two book months ago.

Just as I'm finishing this third book of the Origin Mystery, "The Atlantis World," I get notice from Amazon that my Kindle has just finished receiving "The Altreian Enigma," which is the second. So I was very excited. So I'm going to have to reread, of course, "The Kasari Nexus" to remember what Jennifer and what's-his-name were up to when we last left them, and everybody else, and then go into the next one.

Leo: His name is Amnesia, I think. I think it's Amnesia. I'm guessing, but I'm not sure. No, I'm kidding. Teasing, teasing. Do you want to do - we're definitely not going to get questions in at this point. This is going to be the first Q&A episode with no questions.

Steve: This is the first Q&A with no Q&A.

Leo: I think we need to rename this episode. Oh, well. Do you want to do a SpinRite before we wrap it up?

Steve: I already talked about SpinRite.

Leo: Oh, you did. I missed it. Okay.

Steve: We're all good.

Leo: We're all good. We're all good. Well, what a fun show.

Steve: We will hold the questions for next week.

Leo: Somebody in the chatroom said, "Can you do a question-and-answer episode without a question?" Well, we just did. If you enjoy this show, there's 588 other ones, and many of them have questions and answers, as well. This is the first time they haven't. You'll find those shows at Steve's site, GRC.com. That's where you'll also find his bread and butter, SpinRite, the world's finest hard drive recovery and maintenance utility. You'll also find all the free stuff he does: GRC.com. And he has audio of the show, but he also has written transcripts, human, well-transcribed transcripts of the show. And if you're going to collect the podcasts, collect the transcripts, as well, because I think they're greatly valuable. And it also aids search because Google indexes those transcripts. So it means that the search for stuff within the shows is much easier.

Steve: Apparently we've been making Elaine anxious from these.

Leo: Why is that?

Steve: Well, look at the content that she's having to transcribe every week.

Leo: Poor woman. I feel sorry for her.

Steve: She's like, "Oh, my god."

Leo: [Anguished noises] You'll also find audio and video at our site, TWiT.tv/sn. And you can even subscribe. You'll find this podcast everywhere you find podcasts. And do subscribe. You don't want to miss an episode. We do the show on Tuesdays at about 1:30 Pacific, 4:30 Eastern. That's 21:30 UTC if you want to tune in live and join us in our chatroom at irc.twit.tv. But you can get on-demand, so just make it as convenient as you need. And we'll be back next week. Maybe we'll have some questions then. Who knows?

Steve: Yeah, let's hope for some questions. We have questions. Maybe we'll have time to get to them.

Leo: Answer some good ones.

Steve: Thanks, Leo.

Leo: Thanks, Steve. We'll see you next time.