Sisyphus

Description: Steve's original plan to explain Google's terrific innovations in web performance, known as "QUIC," were derailed by the overwhelmingly worrisome security news, with significant new problems from Linksys, Belkin, ASUS, and others. So this week's podcast is pure, and rather sobering, news of the week. We'll cover Google's QUIC as soon as time permits!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-443.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-443-lq.mp3

SHOW TEASE: It's time for Security Now!. You remember the myth of Sisyphus, the fellow who rolled the boulder all the way up the hill with great effort, and then it just rolled all the way down, and he had to start over again? It's a new form of hell, and Steve Gibson says we're in it, ladies and gentlemen. Finally, our Explainer in Chief has become dejected by the state of security in the world around us. What's wrong with the WeMo and a lot more, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 443, recorded February 18th, 2014: Sisyphus.

It's time for Security Now!, the show that protects you and your loved ones, your privacy online, your data online. Here he is, the data protector, the Explainer in Chief...

Steve Gibson: And now your homes, Leo. We're going to start protecting people's homes.

Leo: Homes, too, yes, because we're going to talk about the Internet of Things today. Ladies and gentlemen, I give you Steve "Tiberius" Gibson of Security Now! fame, and of course GRC.com, where his wonderful program SpinRite makes its home. Hi, Steve.

Steve: Hey, Leo. Great to be with you again, as always. So this is one of those days where I changed course - and the name of the podcast, even - midway through. Some people may see that the word "Sisyphus" is in front of them, or that that was what the podcast was named, somewhat curiously. It was going to be called QUIC, Q-U-I-C. That

is the name of some really good work that Google is doing. I mean, I kind of grumble, like, that when I now export my Google - my Security Now! notes, all the links are actually links through Google, so they're monitoring anyone who clicks on links in the PDFs that I support, because I authored it in Google Docs, which annoys me. So there are things that, like, eh, okay.

But I tell you, the tech that Google is developing for making the web faster, getting pages in front of us. We've talked about SPDY, S-P-D-Y, was the enhanced protocol over TCP that Google was working on. And I was reminded of that because, when I noted last week or the week before that Firefox had gone to v27, one of the features in there was that they had added support for SPDY v3.1. Well, there's another effort, which I think is - it's a little more aggressive, and it has some compatibility problems because it's using UDP rather than TCP. And there is just problems connecting with UDP because, for example, the web doesn't need UDP. It uses TCP. Well, what they've done is they've taken all the work they've done before and said, okay, can we do this over UDP? And, if so, what would we get? Well, we can't talk about that today because there is just too much bad news.

Leo: [Laughing] Oh, I know that.

Steve: Thus I renamed the podcast for today "Sisyphus," from, what, the Greek mythology, the guy that was rolling the big boulder up the hill. And of course it was a never-ending task, and when he stopped it rolled back down, and he had to start again. Well, when I finally got through putting everything together for today, I was so depressed.

Leo: [Laughing] No, no, say it ain't so. Not - not you.

Steve: It's like we're losing this battle. And I have to say that, if you, after we're done, if listeners just sort of sit back and think about what they've heard, they're going to realize that this world, this future where, like, there are actually all of these hackers fully engaged in poking holes in things is way more true than I would have imagined a decade ago. I just sort of thought, okay, we'll get - it's math. Math, we can write this stuff correctly, and we're not going to have problems. But, boy, I mean, we're losing. We're falling behind. And one of the things that we'll see is that it's the commercial interests rushing new features to market, I mean, wanting to, like, now we've got people's thermostats, and like with NEST, and motion detectors, and smoke detectors, and doorbells, and all this stuff is like, oh, let's put it on the Internet. Won't that be fantastic. Wherever you are, using your smartphone, you'll be able to check in. And it's like, whoa, okay.

And this has already been given a name now. It's called the "Internet of Things." And it is, it's one justification for IPv6 because, boy, if every - if all the appliances that you have in your home are on your WiFi, and then somehow sticking their tendrils out through your router onto the Internet so that you, out on the Internet, are able to access them, then things get a lot more complicated. And we see, basically, all of today's depressing news is the consequence of this. It is manufacturers offering features in an unfortunately insecure fashion and, in several - and in many cases clearly making an effort to do it right, but failing to do so.

And the only thing you can - the only reason you can really understand this is that really

secure-oriented people, security-oriented people, weren't involved when they should have been. It looks like this was - security was added as an afterthought. And again, often it's like, now we're seeing, oh, military-grade encryption. Well, yes. But if you hand out the key inadvertently, then it doesn't matter that it's military grade. And in fact, one of the mistakes that we'll talk about today Belkin has made with their WeMo technology. Their firmware upgrades are digitally signed and contain the key for signing.

Leo: [Laughing]

Steve: Which means you take any WeMo device, you reverse-engineer it, you obtain the key, and now you can send your own firmware to anyone's WeMo device. Oh, Leo. I mean, it's really bad.

Leo: [Groaning]

Steve: Okay. So this week on Security Now! we've got a tiny tweak to Firefox. We've got to talk about Kickstarter's data exfiltration that made the headlines and affected many of us who are Kickstartees. And the predicted new wave in mega DDoS attacks. Bitcoin's protocol finally has a problem, which has caused Mt. Gox all these problems. CryptoLocker is getting incredibly greedy. Google bought a little authentication trio of guys that everyone was asking me about because of my work with SQRL. And then we've got problems with Linksys, ASUS, and Belkin.

And then I want to briefly talk about a couple sci-fi updates and the fact that SQRL now, the SQRL UI page, I've been talking for several weeks about how I've been working with the SQRL UI. It's now public. You can see a lot of the user interface. And in fact it's an interesting way to sort of understand SQRL because this is - what I'm designing is what users who know nothing about SQRL, not our listeners, but people's siblings and parents, I mean, really designed to be easy to use. As a consequence, anyone can understand it. So that's also public. In fact, it just - it's bit.ly/sqrlui, if anyone's interested. And for those people who are listening live, the show notes for this podcast are already up on the server.

Leo: Did you see that Google had bought an authentication company that uses audio authentication?

Steve: Yep, that's who I was talking about. It's just three guys - three guys and a patent, it looks like.

Leo: Oh, well.

Steve: And they've only been around for two months.

Leo: SlickLogin is the name of it, yeah.

Steve: Yes. And I think what happened is Google said, oh, you know, we've got an office in Israel, and they're Israeli developers.

Leo: They hired them for the people.

Steve: Exactly. They're smart guys. They basically bought three smart people. And I'm not convinced that sending inaudible crypto from your computer to your phone makes any sense. I mean, I'm proposing doing it with a QR code, basically, that. But we don't know anything else. They also talk about somehow using geolocation and WiFi and NFC. They mix a whole bunch of stuff together. But then they say their secret sauce is that your computer whispers to your phone. It's like, okay.

Leo: The secret sauce gets me every time; right? Because...

Steve: Well, and the concern is they say, and you need no web, you know, it'll have zero, very low impact on your web server. Just five lines of code enables this on your site, which tells you that it is a third party.

Leo: Right. JavaScript, and it's calling a library somewhere else.

Steve: Well, no, no. It's calling them. And so...

Leo: It sends the sound to them. It has to; right? Yeah.

Steve: Yeah. Or they provide the sound.

Leo: Oh, the sound's coming from the website to you, that's right. Yeah, yeah.

Steve: Well, but again, from them, from the company, not from...

Leo: But we've already talked about this. There's no - Google will never adopt a technology that is open and doesn't require Google and their participation; right?

Steve: And of course, again, that's my ultimate response, is that SQRL is two-party. I designed it so it's just you and the site.

Leo: Google's always going to want to introduce themselves in the middle of it.

Steve: Yeah.

Leo: Or somebody like them. Let's start with the tech, with the news, the security news.

Steve: Well, before I get into doom and gloom, I just did - I wanted to thank John Woods. @JohnAlanWoods is his Twitter handle. And I just - this came in at 1:26 p.m. on the 13th of February via the web. And so I saw, in my feed, @SGgrc, he said: "Just used SpinRite 6. Incredible. What a fantastic tool. Files are back!"

Leo: Well, that was fast. Yay.

Steve: So it's like, that's - it doesn't get any better than that. Ran SpinRite 6, files are back, thank you very much. And John, I did - I already responded and thanked him for...

Leo: Files done.

Steve: Appreciate that. So Firefox made a little bump. In case our listeners wonder what that was about, I saw my Firefoxes updating themselves and said, wait a minute. We just got that, like, a couple days ago. Well, this was a tiny one. This went from v27 to 27.0.1 and basically just fixed a couple little problems which cropped up in 27. So this is unusual that Mozilla does a non-major update because they've been doing the major updates now so frequently that normally they're just able to sweep up anything that they need to in a major update. But this one they needed to fix. So there really wasn't much to see there. As I mentioned before, it reminded me about SPDY, which put me onto QUIC, which is Google's really cool protocol for speeding up web stuff, which I hope we'll talk about next week because I really want to. I'm ready to go. But as we'll see, the week's news just overwhelmed us.

Many of our listeners noticed that, or actually received email, I was even getting email from other people who knew that I used Kickstarter and were warning me in case I hadn't seen Kickstarter's email directly. So they got breached. Yancey Strickler, who is the cofounder and head of communications at Kickstarter, tweeted on Saturday, just this most recent Saturday, what, February 15th, he said: "On Wednesday night, law enforcement officials contacted Kickstarter and alerted us that hackers had sought and gained unauthorized access to some of our customers' data." Actually two customers. "Upon learning this, we immediately closed the security breach and began strengthening security measures throughout the Kickstarter system."

Now, I and our listeners, that first question leaves us with some questions. It's like, wait a minute. How did you immediately close the breach? How did some external law enforcement official know and tell you, like you didn't know it yourself, and so forth. I mean, so there's things we don't know, things we may never know. In fact, since no additional information has come out, and we've sort of moved past it now, I think we know all we're going to. He did calm some worries, saying that: "No credit card data of any kind was accessed by hackers. There's no evidence of unauthorized activity of any kind on all but two Kickstarter user accounts."

And then he said: "While no credit card data was accessed, some information about our customers was. Accessed information included usernames, email addresses, mailing addresses, phone numbers, and encrypted passwords. Actual passwords were not

revealed. However, it is possible for a malicious person with enough computing power to guess and crack an encrypted password, particularly a weak or obvious one." So that tells us that, thank goodness, Kickstarter, as you would expect, being in the technology business, did things right. And in fact they did them right from day one.

Leo: Hallelujah. Hallelujah. Somebody who did things right for a change.

Steve: Yes. In a Q&A that they posted, they asked themselves the question, how were passwords encrypted? And in fact many people were authentically asking the question. And so their response is: "Older passwords were uniquely salted and digested with SHA-1 multiple times." Which, you know, that's very good for, like, the dawn of Kickstarter. And then they said: "More recent passwords are hashed with Bcrypt." And of course Bcrypt, we've discussed, is a good password-based key derivation function. It deliberately uses the Blowfish cipher as its scrambling mechanism, and they chose Blowfish because Blowfish has a very slow key schedule algorithm.

Remember that in a symmetric cipher, whatever the input key is, the so-called "key schedule" is the algorithm which normally expands that data. It takes whatever the key length is, 128 bits, 256, whatever, and expands it out to a larger amount of data because typically then you have multiple rounds of a simpler cipher, and each round is keyed uniquely with different data from this so-called key schedule, which was expanded from the actual input key. So Blowfish, it's a well-known characteristic of Blowfish that, while it's a very good cipher, and it's stood the test of time, of course it was designed by Bruce Schneier, and maybe Ferguson, I don't remember, but certainly Bruce was involved. And it's still going strong. It's not preferred now because it does take so long to set up the key.

Well, in something that you want to take long, like brute-forcing a password, you want something that is going to take a while and cannot be easily short-circuited So they did that. But even before that, they were using unique salt and unique means per account. Which means that they can't take just an SHA-1 rainbow table or existing dictionary and apply it against that. And they were doing multiple iterations of SHA-1. They don't give us a count, but the fact that it's more than one means that from the beginning they understood how to do this.

So it's annoying that things got loose. In their email they stressed the need - and I would say maybe overstressed, given how well protected Kickstarter users were, I mean, except for the fact that our customer names, email addresses, phone numbers, mailing addresses and so forth got loose, I mean, that's a big breach of personal data. But they were also then prompting people to change their password if they used the same password on multiple sites. I mean, that's not bad advice. The better solution is not to use the same password on multiple sites. But they've really gone out of their way to protect users. So we'll keep our eye out for any indication that these passwords have been breached. But they really did protect their users as well as anyone could expect them to.

Leo: I changed my password. But you're right, given all...

Steve: Oh, yeah, I definitely changed my Kickstarter password. I did that immediately, yeah.

Leo: And I had, you know, I was using a strong generated password. Somebody in the chatroom, Jeff, suggested that perhaps it's because they use Amazon Payments on Kickstarter that they had to adhere to a higher standard? I don't know.

Steve: Could be. It would be nice to believe.

Leo: Amazon holds the credentials. So even if you didn't do anything right, the best they'd get is the password because they're not going to get a credit card or anything.

Steve: The problem, of course, is that Amazon is notorious themselves for not being HTTPS Everywhere.

Leo: Right.

Steve: You have to be secure briefly. But stealing Amazon sessions is trivial.

Leo: I doubt very, I mean, I'm sure they had a back channel at Amazon for the payments. Who knows? I don't know, actually. Shouldn't say that. That may not be. I think it pushes you to Amazon, come to think of it, when you make a payment on Kickstarter.

[Talking simultaneously]

Steve: Yeah, and I think that technology is well done. I don't mean to disparage Amazon. I think they've done a good job. But Amazon, it's time to switch everybody over to HTTPS. So it's time. It's past time.

Leo: Time. It's time.

Steve: So there's been a record broken, unfortunately.

Leo: A bad one.

Steve: A bad record.

Leo: One you don't want to break, yeah.

Steve: And that is that CloudFlare blogged about deflecting the largest DDoS attack they've ever seen, the largest one anyone has ever seen, now touching 400Gbps.

Leo: We had John Graham-Cumming, who works at CloudFlare, on TWiT on Sunday. He talked a lot about that. Really interesting. Really interesting. And he has a great whitepaper on how that happens on the CloudFlare site.

Steve: Yes, in fact, I've got a link here in the show notes to John's link. It's titled "Understanding and Mitigating NTP-based DDoS Attacks." And you just heard NTP, this is what I talked about, about a month ago, is that this is now the - it's the new darling of the bots which are attacking because it gives on the order of, what is that, 550-some amplification of bandwidth.

Leo: You can start with a megabit connection and get a 400 or even 500Gbps attack with a megabit connection.

Steve: Yes. So, I mean, what they wrote, there was some interesting data here. In the CloudFlare discussion of this, they said "Not Just Theoretical." And they said: "Monday's DDoS proved these attacks aren't just theoretical. To generate approximately 400Gb of traffic, the attacker" - now, this is their own metrics, this specific attacker. So to generate this 400Gb that they deflected, "the attacker used 4,529 NTP servers." Okay. So that's a little over 4,500 network time protocol servers scattered all over the Internet. And that map you showed, I also have it here in the show notes, are the locations of those network time protocol servers.

Leo: Quite evenly distributed globally. Just everywhere.

Steve: Yeah. And so you'll notice that, or you'll remember from my discussion of how this works, that somebody is sending requests to those 4,500 servers for the list of all of up to 600 other requests they've received for the Internet time. And so the request, however, has its source IP spoofed. It uses source IP spoofing so that those 4,500 servers believe that the target of the attack, in this case it was an account, it was a website that CloudFlare was in the business of protecting from these attacks.

Leo: Yeah, he didn't name names, either. We don't know who.

Steve: Right. Right. And so somebody sent, sprayed these 4,500 NTP servers with requests for their list of people who've contacted them, the most recent 600, spoofing the source IP so they all aimed, all 4,500 of those servers aimed their responses, focusing it down to a single IP that was the target of the attack. So the blog says the attacker used 4,529 NTP servers running on 1,298 different networks. So almost 1,300, two shy of 1,300 different networks. So three or four NTP servers per network, scattered, as you said, Leo, globally. On average, each of these servers, each of the 4,500+ NTP servers, sent 87Mb of traffic to the intended victim on CloudFlare's network. And this is one of the other problems with this attack is that more so than DNS servers, which have been used in similar reflection attacks, NTP servers tend to be running on big iron. DNS generally isn't very high bandwidth. It doesn't necessarily have high bandwidth connections. NTP servers generally are running on routers, big iron routers, and they've got NTP service just sort of running as a side effect of just the fact that they're running a system that offers all of these common Internet servers.

Leo: It's kind of a default service commonly on servers.

Steve: Exactly.

Leo: So they may have disabled it, I mean, unless their clients need it or - but it seems like it's just not - it's always on; right? I mean...

Steve: This is the problem. It's on by default. And so this posting goes, it continues, saying: "Remarkably, it is possible that the attacker used only a single server running on a network that allowed source IP address spoofing to initiate the requests." They said: "While NTP servers that support MONLIST" - that's the command, MONLIST - "are less common than open DNS resolvers, they tend to run on beefier servers with fatter connections to the network. Combined with the high amplification factor" - which remember is like 500-plus, 550 times - "this allows a much smaller number of NTP servers to generate very large attacks. For comparison, the attack that targeted Spamhaus used 31,000 open DNS resolvers to generate a 300GB DDoS attack. On Monday, with one seventh the number of vulnerable servers, the attacker was able to generate an attack that was 33% larger than the Spamhaus attack."

Leo: [Whistling] Wow.

Steve: So it's interesting, too, because in the research I did for this I ran across, it might have been John's posting, or maybe lower in that blog posting, where the author was sort of musing about, you know, I'd like to talk to whoever it was that even came up with this MONLIST command. It's like, why? It's like, why do you care? Why would you have a command to tell some arbitrary NTP server to tell you who it's been talking to? It's like, who cares?

Leo: But that's a long list, presumably, and that's what fuels the DDoS because sending that response back to the target is going to jam its pipes, especially if it comes from many, many, many NTP servers. There are other similar amplification attacks. We've talked about them before. You could ask a router for its tables, things like that; right? A BGP router.

Steve: Well, normally routers that you talk to are TCP, and TCP cannot be spoofed.

Leo: Saves that, right, right.

Steve: Yes. And so that's the great danger.

Leo: Has to be UDP, okay.

Steve: Yeah, it's got to be UDP because that doesn't involve the three-way handshake

that, among other things, verifies the IP address at each end.

Leo: It is trivial to configure the NTP server to ignore that request.

Steve: Yes. And that is - so there are several things you can do. You can not have raw sockets [clearing throat] available to the operating system that doesn't need them, and then software is unable to spoof the source IP. Or the ISP could block the departure of packets that do have spoofed source IPs. And this is called "egress filtering." We've talked about it in the past. Any ISP is, like, they're hosting some set of IPs. It's like, if it's a cable modem company, they're like 24-dot IPs; right?

So that ISP absolutely knows what its IP space is. All they have to do is block packets egressing, leaving their boundary, their network border, that don't say they're coming from their IP space. Because for this attack to work, it's necessary for packets to be leaving an environment with a different IP that are lying about generating that source IP. And so all you have to do is just drop those packets. Put a filter, a simple filter rule in your outbound routers that interconnect the ISP to the Internet, and this problem goes away. You can no longer spoof source IPs. And that's a general mitigation for all spoofed source IP reflection attacks across the board. And then, of course, the ultimate solution, which we will never reach because many of these servers are in closets, they're the famous server in the closet that just sits there, it hasn't been rebooted for 10 years, and it won't be. But mitigation of this would be good, too.

That is to say, as you said, Leo, it turns out it's a simple configuration line. You add a line to the NTP config file saying, eh, don't tell people who we've been talking to. Just disable that MONLIST command, it's trivial to do, and the problem also goes away that way. So that would be nice. Now, it is the case also that, if you're generating 87Mb from the average router, probably, running NTP, all aimed at one location, that should show up as a blip on the radar of the people in the NOC, the Network Operations Center, for the people running these routers, or these NTP servers, probably a service running in a router.

That ought to raise a question in them. It's like, wait a minute. Why do we have a bunch of routers that are all suddenly generating 87-plus megabits of traffic, aimed in a certain direction? And all they have to do is look at it and go, oh, we're being used, our server, this router probably is being abused by some attacker, and used to do NTP amplification and reflection. So that would motivate them to simply fix their server. So this may be something over time that gets fixed. But, wow.

And the other point I wanted to make was that, while this attack was happening, other aspects of the Internet, not the victim, were affected. And that's a consequence of the way the Internet works. Remember when I first talked about the idea of when - it was in our first podcast about Ed Snowden and how traffic was concentrated - I was using the example of Google, how as data was coming closer and closer in terms of hopping from one router to the next, it was approaching Google, and this was where I used the original, at that time it was speculation, but it's since been confirmed, that all the NSA had to do was tap outside of Google because the traffic would have been concentrated down by the time it got to Google so that you could just, like, monitor nearby, and you'd be getting all of Google's traffic.

Well, similarly, an attack of this magnitude means that it might have even been bigger than 400GHz, but routers several jumps away were overloaded and unable to handle the attack because, again, it's a concentration. And so if you had less capable routers, which

were fine for normal day-to-day Internet traffic, but the point is that even before the last few stages of concentration, an attack this size would have and did in fact bring down other routers just that were trying to do their job of forwarding the traffic. But there was too much incoming for them to send. And other sites that were not even related ended up being knocked off the Internet because the point is this was so overwhelming that, several steps away, the attack got so big that non-targeted hardware could no longer handle it. I mean, it's a...

Leo: That's pretty impressive.

Steve: ...weird phenomenon, yeah.

Leo: But it's something you probably can only do once. I would imagine the people who perpetrated this had it in their back pocket for a while and then used it and know that now NTP will probably be widely patched or disabled and so they probably can't do it again.

Steve: It's going to take a while. I mean, the people in the know are not happy that this happened because this was theoretical. In fact, there's a link at the top of this article. I have it in there, Leo, the us-cert.gov link. That's their report about the concern over NTP, oh, I'm sorry, over UDP reflection attacks. And they list many different services which can be used to attack.

Leo: DNS, NTP, SNMP, NetBIOS, SSDP, CharGEN, QOTD, BitTorrent, Kad, Quake Network Protocol, and Steam Protocol.

Steve: And if you scroll down a little bit, they also show the amplification factor.

Leo: Right. NTP is the best one of the bunch.

Steve: Yes. And so that's what we're seeing. But in fact John was saying that SNMP, that's another UDP...

Leo: He mentioned that one, yeah, yeah.

Steve: Yes. That's Simple Network Management Protocol. And that's the one where - and in fact that may be what you were referring to.

Leo: Spamhaus. That was the Spamhaus attack was SNMP. Wasn't it?

Steve: Yeah, but because there are SNMP queries where you say, tell me everything about what's going on in your router, and it dumps a phenomenal amount of information out.

Leo: I would just hope that this - now that the word's out, that these things are - it seems like they're easily fixed for future reference; right?

Steve: Yes, absolutely. Absolutely easily fixed. It's just, again, this is like - and this is, if this particular podcast has a theme, and unfortunately it does, it's this sense that we're not going to win this. I no longer believe that we're going to win this. I think new stuff is happening at an accelerating rate. Everyone knows how I feel about new. It's just bad. In fact, someone commented that they got a kick out of the fact that the screenshots I have for the forthcoming SQRL UI, which I posted, and he said - it was a listener who said, "I got a kick out of the fact, Steve, that they're XP screenshots." It's like, eh, oh, yeah, they are XP.

Leo: [Laughing] Of course they are. What else would they be? You think Steve would install Windows 8 just for screenshots? Come on.

Steve: No. It's new. It's bad. And so..

Leo: The factor that Cert gives is actually low on SNMP. I think that John said it was actually much higher. It would be...

Steve: I think it's vastly higher, Leo. Maybe they're doing a little misdirection, hoping that the bad guys don't notice that, yeah. SNMP, I mean, I use SNMP. I've got, like, a screen here that monitors what's going on at GRC. And so I'm querying using SNMP, the various hardware that I have in our Level 3 facility, our data center, constantly getting its packet counts and so forth. And I remember when you traverse the SNMP tree, it's like this dotted decimal notation which has all been standardized, and they're called, I can't remember, there's a name for that. Not SMBs, that's Server Message Blocks. But there's something which is like this well-established protocol for enumerating the contents of network equipment over Simple Network Management Protocol. But, boy, you can ask it a question it'll give you a huge answer to.

Leo: The Spamhaus attack was open DNS. I'm sorry.

Steve: Right, it was DNS.

Leo: Let me correct myself on that, yeah. So UDP is, in a way, the problem. You're not going to get rid of UDP. But we certainly should look at all protocols that use UDP and mitigate.

Steve: Yes. And you could look, you could go to, at the protocol end, or you could tell ISPs, there's now a...

Leo: Ah, this is what he was suggesting, yeah.

Steve: Yes. There is a well-established standard for egress filtering, where ISPs simply drop packets which are carrying spoofed source IPs. And that would end this problem. I mean, and even if all ISPs didn't, it would mean that it would be much more difficult to find, you'd have to explicitly then find machines in non-filtering ISPs that weren't doing this blocking. But still, blocking really does make sense.

Leo: Interesting. Interesting.

Steve: Okay. So there's that.

Leo: Yes.

Steve: Now we have the Linksys router Moon Worm.

Leo: The hack of the week. Now what?

Steve: Oh, my lord. So I wasn't aware, until I followed some links, where this was discovered. But because the very - Ars Technica reported, Windows IT Pro was doing reporting on it. And all that was said was that it was a Wyoming-based ISP. Well, it turns out it's an old friend of ours, Leo, Brett Glass. I've known Brett, you've known Brett for decades.

Leo: Great, great tech journalist, yeah.

Steve: Yes. He was at InfoWorld for a long time while I was there. And he is really a - he's a techie's techie. So what Brett wrote I thought was interesting because it was in a posting to one of the Ars Technica pieces. He said: "I'm the ISP who discovered the worm."

Leo: Brett was an early WISP, Wireless ISP guy. I don't know if this was his WISP, or it must be.

Steve: No, no. Yeah, it is. He is still, because he's, like, in the boonies of Wyoming, and there was no Internet service anywhere. And he and I've talked about him, like, getting Pringles cans and, like, going up to water towers out in the middle of nowhere and, like, aiming the can, or maybe it wasn't a can, it might have been a - I can't remember the name of the antenna he was using, but - Yagi. I think it was a Yagi is the name I was thinking of. I mean, so he, like, was really pioneering bringing Internet to farmland out in the middle of nowhere; and, as you said, being a wireless ISP.

So he said: "I'm the ISP who discovered the worm. It was not saturating our entire ISP's bandwidth." Actually that was a misstatement that Ars Technica's reporting made. He said: "We don't let a customer commandeer unlimited bandwidth. But it was consuming users' bandwidth allocations, slowing down their legitimate activity and interrupting streams and VPN connections." He says: "I discovered the worm when these users called

to complain of poor performance, and I employed a packet sniffer to investigate the cause. I've now seen hits from thousands of exploited routers. But the number of infections worldwide is likely to be much larger, hundreds of thousands, if not more. Just as the media is beginning to chatter about the 'Internet of Things,' we're starting to see serious exploits of these things. The potential for harm and invasion of privacy is breathtaking, and it's of great concern that Linksys was recently sold by Cisco," he says, "(which has ample security resources) to Belkin, which does not."

Leo: It's not that Cisco was doing anything right with Linksys anyway, but all right, yeah.

Steve: Yeah, I was going to say, not like...

Leo: They may have had resources, but they didn't apply them, so.

Steve: So Brett poses the rhetorical question: "Will Belkin be able to handle this breach as Cisco could have?"

Leo: Could you describe what this malware does? This is new; right? We've not talked about this before.

Steve: Nope, this is brand new. I'm just trying to see if there's anything else. Brett says: "In any event, the security exploit that's used by the worm will work in all current and recent Linksys routers, including..."

Leo: Ooh. Because Linksys said older E-Series routers.

Steve: I know. That was not true.

Leo: Ooh.

Steve: All current and recent Linksys routers, including the entire E-Series. And he says: "(Yes, E1200s with the most recent firmware v2.0.06..."

Leo: Probably the bestselling router out there, by far.

Steve: Which is why Brett brings it up. I mean, Brett knows his stuff. He says: "(...are vulnerable after all), the Valet series, and some with WRT part numbers, for example the WRT160. However, this particular worm seems to focus on the E-Series and appears to be aimed at marshaling a botnet. So far, it does not appear that the malware flashes itself in, so it can be removed by a reboot. But it appears that any router with stock firmware that's exposed to the Internet at a public IP address" - which is to say all of them - "can be reinfected even if it has a secure password." And finally he says: "I am

continuing my research into this worm, as are the folks at SANS, who are fantastically bright and competent. After I informed them about the malware, they duplicated 99% of my work in less than a day and forged on ahead."

So that was from Brett, who discovered this. I'm just looking to see if there's anything else to add. We know that it connects to port 8080 and runs a CGI script on the router, which downloads and executes a 2MB program, which then scans other vulnerable routers. But it lives at this point only in RAM, and reboot clears it. Now, of course, the problem is that, unless we get firmware which solves this problem, we are subject to reinfection, and other worms can be designed which may not be as nice behaving. So this apparently, people who've looked at this say this appears to just be replicating itself in order to show that it can.

Now, Linksys has put out an official statement, saying: "Linksys is aware of the malware called 'The Moon.'" By the way, it was named "The Moon" by the guys at SANS because, as they took it apart, they found some HTML pages and images from the movie of that title, "The Moon." Which, by the way is kind of a funky...

Leo: What's the movie "The Moon"? I don't even know that.

Steve: Oh, it's good good. Yeah, it's definitely worth watching. When "Oblivion" came out early last summer, which was still one of my favorite movies of last year, Tom Hanks - I'm sorry, not Tom Hanks, Tom Cruise - several people said, oh, that's a little reminiscent, a little derivative. And so I then saw "The Moon" because it did not get great reviews. And it's kind of - it's funky, but it's definitely, if you're a sci-fi person, I would recommend it.

Anyway, so Linksys says it's "...aware that the malware called 'The Moon' has affected select older Linksys E-Series routers and select older Wireless-N access points and routers." So as you say, Leo, they're stressing older. But I believe Brett. He wouldn't say this otherwise. "The exploit to bypass the admin authentication used by the worm only works when the remote management access feature is enabled. Linksys ships these products with the Remote Management Access feature turned off by default." So that's significant and happily mitigating. "Customers who have not enabled the Remote Management Access feature are not susceptible to this specific malware. Customers who have enabled the Remote Management Access feature can prevent further vulnerability to their network by disabling the Remote Management Access feature and rebooting their router to remove the installed malware."

Leo: We tell everybody to do that; right?

Steve: Yes.

Leo: That's the remote - that's WAN administration. Is that what the...

Steve: We've, yes, the advice from this podcast from day one has been that is a bad idea.

Leo: You don't need it. Yeah, you don't need it.

Steve: Yeah. Often, you know, there have been manufacturers who have it on by default, enabled by default. So absolutely disabling it if you don't need it is what you should do. However, the mistake here is that, A, this was an exploit of the URL parsing in the server listening on port 8080, running in the router. And so it was one of these buffer overflow, super long queries that was able to get - is able to execute shell code, essentially. And that then was enough to get the router to go download the malware, which is much larger, 2MB, into itself and run it.

So in their Linksys knowledge base there's a bunch of information per model number and per router. I created a bit.ly shortcut for it: bit.ly/themoonworm, all lowercase, bit.ly/themoonworm. And that will take you to a URL which is ridiculously long, which is Linksys's knowledge base for this problem. And it says there: "The Moon malware bypasses authentication on the router by logging in without actually knowing the admin credentials. Once infected, the router starts flooding the network with ports 80 and 8080 outbound traffic, resulting in heavy data activity. This can be manifested as having unusually slow Internet connectivity on all devices."

So if anyone feels like something has gone wrong, and you have a Linksys router, this may be what's gone wrong. But on the other hand, remember, if you didn't turn on - if you've got firmware that has this off by default - it would be interesting if that's where this older firmware comes in.

Leo: Ah, maybe that's what we're saying, is we - yeah.

Steve: Yes, is that the older firmware had it enabled by default, and they're not saying that. So the newer stuff has it disabled by default. But a responsible network person who knew they needed, who had a reason for needing remote management access, would have given it like a really good password and thought, okay, I'm probably safe. And so the takeaway here is no, unfortunately, there is some sort of buffer overrun in the parsing of the query string which allows remote hacker-provided code to take over your router. So you definitely want to turn off Remote Management Access until this is fixed and patched and verified.

Leo: The Linksys remediation page also says check the box that says "Filter anonymous Internet requests." And I guess the impact on that would be it would break multicast, or I don't know what you'd need that for. Anonymous Internet requests.

Steve: Anonymous Internet requests.

Leo: Make sure that's not checked, whatever the hell that is.

Steve: That doesn't sound good.

Leo: You don't want to allow anonymous Internet requests, do you? Really?

Steve: The only thing I could think is that you're probably - you're challenged for a login, admin and password, and then your requests carry that response back in the headers, and so it's no longer anonymous. So, but you'd need it, I would think you'd need it the first time in order to make an anonymous request in order to be challenged for the logon credentials. But definitely turn it off unless you know you...

Leo: I can't think of a reason why you'd want that, yeah. All right.

Steve: Okay. Continuing the hit parade. We're one down.

Leo: And of course putting DDWRT on your Linksys is probably the best idea.

Steve: Yes. In fact, that was another piece of advice I saw elsewhere was, by all means, install third-party good firmware. Buy the hardware from these people that make the blue boxes, and then stick real community-designed...

Leo: Put software on it.

Steve: Real software.

Leo: Put some good software on.

Steve: Oh, lord. So speaking of Belkin, it was bizarre that Brett talked about Belkin's purchase of Linksys and wondered what it meant and then also brought up the Internet of Things, both rather prescient here, because this is - just today, February 18th, this morning, we get the news - and in fact in my notes I said, "Speak of the devil." Belkin. Slashdot has the story. And this is regarding Belkin's so-called "WeMo," W-e-M-o. And I didn't even think to wonder what it actually stands for because I was going to say, if it was WaMo, then this is way mo' access than you intend to give to your household appliances.

Leo: I think it's like remote, remo. I don't know. I've used them. I love them. We reviewed them. They're great.

Steve: Uh-huh.

Leo: I don't think they're that widely used.

Steve: Half a million of them, or half a million users, apparently, they're claiming. At

least that's only one sixth as many people who have Java in their device.

Leo: You can control your WeMo with If This, Then That, so you can have it respond to various things. There's motion detectors. There's turn-on lights. They're all Internet connected. So...

Steve: If This, Then That? That's a cool little thing. That sounds like a little logic flow.

Leo: Oh, you haven't seen that site?

Steve: No.

Leo: IFTTT.com. Oh, you're going to love it.

Steve: Wait wait wait wait wait. You're telling me you're giving some website access to the devices in your home?

Leo: Oh, sure. What are they going to do, turn off the lights? Oh, there's a lot of stuff you can do with If This, Then That, though.

Steve: Oh, nothing could go wrong with that.

Leo: No, no, no, it's wonderful. It puts the Internet to work for you.

Steve: That's right.

Leo: How could you not? I use it. It works with the Nest. It works with the Hue lights. So for instance, when the sun goes down, I have my lights turn on.

Steve: Because of some third-party Internet site that's, like, sending commands to your home?

Leo: Yeah.

Steve: Oh, my god.

Leo: I just sign into my If This, Then That, and I create a recipe. It works with Twitter, Facebook, Instagram, Craig's List, Google Drive, Foursquare.

Steve: Well, then, in that case, I have a context for you, Leo. The context of this next story is that hackers don't even need to break into If This, Then That to gain access, direct access, to the devices you have in your home, which are light switches, outlet plugs, motion detectors. They also announced this year they're partnering with Mr. Coffee, the Crock-Pot company, and Sunbeam, and others, to bring home automation to your favorite everyday appliances. What could possibly...

Leo: That would explain why the other day I came home and my coffee maker was on, my thermostat was turned all the way up, all the lights in the house were on, and the TV was set on a porn channel. I thought it was my kids.

Steve: Run away, Leo.

Leo: No. I'm just...

Steve: You've lost control of your home.

Leo: I'm teasing you. That's what happens when you have children. It's the same.

Steve: So a security researcher who has a little firm, IOActive, Inc. - I had his name here, and I don't see it. Oh, Mike Davis. They put out a press release. There's a link to it. There's that PDF, oh, I'm sorry, not the PDF, the link above in the show notes, Leo. It says: "IOActive, Inc., the leading global provider of specialist information security services, announced today" - that is this morning, as we're recording this - "that it has uncovered multiple vulnerabilities in Belkin WeMo home automation devices that could affect over half a million users. Belkin's WeMo uses WiFi and the mobile Internet to control home electronics anywhere in the world" - yes, even from the Ukraine, when you're not anywhere near the Ukraine - "directly from the user's smartphone." And for that matter, anyone else's. I'm adding that, of course. Didn't mean to embellish their press release.

"Mike Davis, IOActive's principal research scientist, uncovered multiple vulnerabilities in the WeMo product set that gives attackers the abilities to remotely control WeMo Home Automation-attached devices over the Internet; No. 2, perform malicious firmware updates; No. 3, remotely monitor the devices in some cases; and, No. 4, access an internal home network." So they can use this as a beachhead. They can use the installed Belkin WeMo device essentially as a proxy to get onto your home network.

"Davis said, 'As we connect our homes to the Internet, it is increasingly important for the Internet of Things device vendors to ensure that reasonable security methodologies are adopted early in product development cycles. This mitigates their customers' exposure and reduces risk.'" Yeah, no kidding. "'Another concern is that the WeMo devices use motion sensors, which can be used by an attacker to remotely monitor occupancy within the home.'

"Once an attacker has established a connection to a WeMo device within a victim's network, the device can be used as a foothold to attack other devices such as laptops, mobile phones, and attached network file storage."

So little details here, drilling down into this. As I mentioned to you, Leo, at the top of the podcast when we were talking about this craziness, the reason I named this podcast "Sisyphus," for some reason Belkin's firmware signing key is included in the firmware. "The Belkin WeMo firmware images that are used to update the devices are signed with public key encryption" - oh, that sounds good - "to protect against unauthorized modifications." Okay, good. "However, the signing key and password are leaked on the firmware that is already installed on the devices. This allows attackers to use the same signing key and password to sign their own malicious firmware and bypass security checks during the firmware upgrade process."

Meaning that, okay, so it's nice that public key crypto, encryption, is being used for the firmware. It should not, however, include the private key and the password. Whoops. So currently it does. Then they use SSL only for privacy, not for authentication. And in fact we're going to loop around back to this misuse of SSL a little bit later in the podcast. So what does that mean? We've talked about how SSL provides two things: It provides privacy through encryption, and it provides authentication through the use of the whole SSL security certificate hierarchy rooted in certificate authorities.

But "Belkin's WeMo devices do not validate Secure Socket Layer certificates, preventing them from validating communications with Belkin's cloud service," that is, all the communications, see, these things, when you access one of these devices from your smartphone, your smartphone is connecting to Belkin's cloud, and the devices have static connections to Belkin's cloud. And so that it's through the cloud intermediary that you're able to be notified to change your temperature in your house, turn on the crockpot, and so forth. Unfortunately, even though they're using SSL, the devices don't do certificate validation. So it is completely possible for the SSL connections to be intercepted and bad guys to take them over that way.

As this report says: "This allows attackers to use any SSL certificate to impersonate Belkin's cloud services and push malicious firmware updates and capture credentials at the same time. Due to the cloud integration, the firmware update is pushed to the victim's home regardless of which paired device receives the update notification or its physical location." So it's not good. And they use, in order to do NAT traversal, we've talked about NAT traversal in the past, there's the so-called "STUN" and "TURN," S-T-U-N and T-U-R-N, protocols.

And it turns out that they've been implemented in a way that allows them to be abused. Their report says: "The Internet communications infrastructure used to communicate Belkin WeMo devices is based on an abused protocol that was designed for use by VoIP, Voice over IP services, to bypass firewall or NAT restrictions. It does this in a way that compromises all WeMo devices' security by creating a virtual WeMo darknet where all WeMo devices can be connected to directly and, with some limited guessing of a secret number, controlled even without the firmware update attack." So, I mean, this just - it's really not good.

Finally: "The Belkin WeMo server application programming interface" - so their API - "was also found to be vulnerable to XML inclusion vulnerabilities, which would allow attackers to compromise all WeMo devices." So then this wraps up. It's like, okay, wait a minute. Was this disclosed responsibly? I mean, this is big news, and this is not good news.

They said, under "Responsible Disclosure": "IOActive feels very strongly about responsible disclosure and, as such, worked closely with CERT on the vulnerabilities that were discovered. CERT, which will be publishing its own advisory today, and that is online, made several attempts to contact Belkin about these issues. However, Belkin was unresponsive. Due to Belkin not producing any fixes for the issues discussed, IOActive

felt it important to release an advisory and recommends unplugging all devices from the affected WeMo products."

Leo: That's too bad.

Steve: So, yeah. So this is, again, as I said...

Leo: Can I - I mean, really, seriously, so they can turn on lights in your house. And they can monitor your motions, I guess, if you have them everywhere in the house. I mean, most people wouldn't have them everywhere in the house. What is the worst thing that could happen from this?

Steve: Okay. It's a function of what you plug this into.

Leo: Well, yeah. Okay. If you plug it into your iron lung, they could kill you. But I doubt you plug your iron lung. Mostly this is about plugging in lamps so that you can turn them on remotely. The motion sensors, I guess, if you had them everywhere in the house, then a bad guy could see if anybody was home. But they're not everywhere in the house. They're maybe in the living room. I mean, nobody's going to - they're too expensive to put everywhere in the house.

Steve: Well, so how about remote access to your internal network? That's not good.

Leo: But so they can get into my WeMo and then use the WeMo to hack my...

Steve: As a proxy, because it's on your network, and it has to know...

Leo: Is that theoretical, or is there actually a way to do that?

Steve: There's actually a way to do that. We're not wanting to talk about that too much.

Leo: Okay. All right.

Steve: It's so bad.

Leo: All right. So because they could easily get into the WeMo itself - the WeMo's a dumb device, but they could use that as a gateway into the WiFi network.

Steve: Because it's on your WiFi network.

Leo: Bypassing any WPA or any kind of security.

Steve: Well, because it's on your network. It had to have your network key in order to be on your network. And it's using your network and your network connectivity in order to get out onto the Internet. So if its firmware is abusable, or even if its cloud services are abusable, then that allows a way for people to get into, I mean, this news just hit, Leo. And so we will, not long from now, be hearing about exploits of this in the same way that, you know, unfortunately, obscure vulnerabilities get exploited. And it's like this weird authentication bypass on the Linksys routers. Well, someone discovered that there was a bad...

Leo: But that's your router. This is a thing that turns the lights on and off. I mean, it's not...

Steve: It is a router. They've turned this into a router.

Leo: I guess it is, right, yeah. Right.

Steve: This is a router. Yeah, and you mentioned that they were dumb and not smart, and that's the problem. Unfortunately, they are underpowered so that they're not able to do an SSL certificate verification.

Leo: Yeah, yeah, yeah, of course.

Steve: And so it's like, oh, well, we'll use SSL.

Leo: If I put it on the guest network, for instance, would that make it more usable?

Steve: Isolating it from your - yes, isolating it from your home network, or setting up a second router that is only for your questionable security devices. That makes a lot of sense.

Leo: And then definitely don't attach it to your iron lung.

Steve: And if you're a person, if you've got a Comcast router that has a public WiFi hotspot in your living room, hook it up to that one.

Leo: Right. Share that with the world; right?

Steve: Exactly.

Leo: By the way, we found, somebody emailed me, there is a switch. You can go to the settings at Comcast and disable it if you dig deep enough.

Steve: Yes. And many people also commented, in our coverage of that, that if you just use your own hardware...

Leo: Yeah, I use an Apple router on a Comcast network. Obviously they can't, you know, don't have a Comcast WiFi device. And no one should. All right. No, I'll consider - I don't have any WeMos because I didn't find them all that useful. But that's the real attack, is it's an unguarded pathway into your WAN.

Steve: Yes. Well, that's the worst thing. I mean, yeah, having your lights turn on and off and having, I mean, see, here's the problem, Leo. Next up is webcams and nanny monitors and baby monitors and, I mean, this is - we're seeing the tip of the iceberg. You can imagine Belkin has 20 more devices on the drawing board. I was talking about this SkyBell that a friend of mine has which is a video doorbell. And it's the same thing. It's in the cloud. And so what we're seeing is we're seeing, because these are also price sensitive, they're running the smallest processors, the least amount of memory, they're really cut-down technologies. But then they're leveraging technologies designed for full-scale systems, like SSL certificates. But, whoops, self-signed certificates work just fine because we're not checking to make sure that they're signed by a valid certificate authority because we can't afford the storage.

Leo: Right.

Steve: Remember, I mean, Hong Kong Post Office, you know, you'd have that...

Leo: These are little devices. They're little doohickeys.

Steve: Yeah. But they're, like, playing in the major leagues now. They're now on the Internet, and they're bridges into your home network.

Leo: And there will be lots of these. I mean, Nest and Hue and WeMo, I mean, all of these, Sonos, all of these, there'll be lots of these.

Steve: And so when you start off saying, well, what damage could be done because it turns my table lamp on and off, it's like, today. Wait till - and we need to get a handle on this. And the Internet of Things, as Brett said, and as even the IOActive guy said, it's here. It's coming.

Leo: But for now keep it off your iron lung. And banks, if you're using WeMos to open the safes, the vaults at a given time, probably shouldn't. That'd be a mistake.

Steve: I'm not using any of this stuff, Leo. Like I said, I'm glad...

Leo: Well, you're using XP, Steve. We know you're not using it.

Steve: And I have a 2001 car, and I'm quite happy that I still have to have a key that I put in and turn.

Leo: How long before you move to a cabin in the wilderness? All right. Moving on. I will take that under advisement. And I long ago retired any WeMos in my house, but just because they were dumb.

Steve: How long have they been around?

Leo: Oh, a year or so, yeah.

Steve: These WeMelos.

Leo: WeMo. WeMo.

Steve: Okay. And ASUS routers are exposing the entire contents of the USB drives that are plugged into them. Oh, by the way.

Leo: Great.

Steve: Uh-huh. So ASUSTeK has also been nonresponsive. People are now finding a text file on the drives which they intended only to be used - the idea, the back story here is that ASUS routers now allow you to plug a USB, a big USB storage stick into the back of them, 64 gigs or whatever are very common now. And that creates a shared file server for your home. You're unfortunately sharing it a little more widely than you believed.

Default settings for the FTP server allow WAN-side anonymous login. ASUS calls this feature "limitless access rights." And the authors of this letter call this "madness." And there's something called AiCloud where it turns out those usernames and passwords were stored in plaintext in a file available for download without logging in. So that gives attackers access there. So what people are finding on their storage device, that's how people are being awakened to this, is a file called WARNING_YOU_ARE_VULNERABLE.txt. And the file is a text file that just appears on your home's internal network storage that says, "This is an automated message being sent out to everyone affected. Your ASUS router (and your documents) can be accessed by anyone in the world with an Internet connection. You need to protect yourself and learn more by reading the following news article." Then there's a link to nullfluid.com/asusgate.txt.

And it says: "Below is a list of all the vulnerable IP addresses that have been leaked." And I don't remember the number. I think it's 4,000 have been found so far. "If you are reading this, you are vulnerable, too." And then there's a link to a Pastebin URL with the

list of all the IP addresses. "Solution: Completely disable FTP and AiCloud immediately. I hope we helped. Sincerely, g."

Now, once again, this has been known for a long time. A guy named Kyle Lovetts on June 22nd, so more than a year ago, attempted to report this to ASUS. And he created a SecurityFocus posting, which is now there in the archive. He says: "Timeline: Contacted ASUS two weeks ago under my online handle account around 06/06." So around June 6th. "Second email sent on 06/10 when discovered first unauthenticated file disclosure. Received only one response back stating it was not an issue. Sent a third email on June 14th. Only response received was an acknowledgment that my email was received. Attempted to call their development or incident team and was told that somebody would call me back on 06/17. Sent another email today under my real name." Never received any communication from ASUS.

So he says, under "Mitigation and temporary fixes": "Users need to be alerted to turn off AiCloud service immediately." So this is another cloud service designed to connect your router to the cloud, and it's been done wrong. "All web access to both HTTP and HTTPS need to be halted until proven safe. UPnP services need to be turned off." And he says in parens, "I'd say that's a safe bet is for all home routers to turn it off." And of course that's our standard advice, too. He says: "Disable FTP and Samba services until the problem is fully understood and patched, if possible." So, oh, and he says, by all means, change the default username and password. Do not leave it as it was. So this is, again, breaking news. This has just come to light. And ASUS has unfortunately not been responsive to this. So we can hope that they will be shortly. At this point anybody using this AiCloud technology needs to turn it off because, from the outside, from the WAN, if the AiCloud services are turned on, you do not have the protection which you assumed ASUS had been providing, based on this reporting. Ouch.

Leo: Wow.

Steve: Yeah.

Leo: So, okay. No ASUS. No Linksys. No Belkin. Okay. Pretty much you're putting the entire tech industry out of business. You understand that.

Steve: Not a good - not a bunch of good news today.

Leo: Putting themselves out of business is what they're doing.

Steve: Well, yes. And this is the problem. I mean, okay. Anyone can be forgiven for making a mistake. That's not a problem. But in every instance, these companies have known for months. They've been notified by responsible researchers who discovered this and tried to say, oh, my god, everybody who's using your cloud service is vulnerable. Everybody who's using - your half a million that you're bragging about WeMo customers, you're basically installing little routers that are breaching your security every time you install one of these things. Fix this. And the problem is these companies don't respond. They're in the consumer electronics business. They just want to sell hardware. And unfortunately, with what they're doing, with the services they're offering, comes responsibility. Oh, it's...

Leo: If the Navy can't keep itself safe, what are we supposed to do? You've seen this story. Iran was hacking the Navy?

Steve: Oh, lord.

Leo: In fact, apparently the Iranian hacking of the Navy is more extensive than originally thought. The Wall Street Journal said yesterday the cyber attack targeted the Navy Marine Corps Internet, which is used by the Department of the Navy to store websites, store nonsensitive information, handle voice, video, and data communications for the Navy. They've been in the network since November. They're just living there, having fun, cooking hot dogs.

Steve: APT, Advanced Persistent Threat.

Leo: Yeah. From Iran. Who knew.

Steve: Yeah, yeah.

Leo: What else you got? This is the Sisyphus episode, ladies and gentlemen.

Steve: I know. This is just - now you under...

Leo: We've been pushing that boulder up the hill and, yeah...

Steve: Now you understand why I just thought, oh, my.

Leo: ...crush us. Yes.

Steve: I'm so depressed. So we did - I don't remember if you did it before we began recording or not, but...

Leo: No, we mentioned this on the air, yeah.

Steve: Oh, yeah. Okay, good.

Leo: SlickLogin stuff, yeah.

Steve: SlickLogin. Basically, I think, Google didn't - they weren't in love with the technology. They just bought some smart programmers.

Leo: They call that an "aquihire," a-q-u-i-h-i-r-e. They hire by acquisition.

Steve: Ah, yes.

Leo: Aquihire.

Steve: A malware researcher was surprised when Gmail blocked some malware that he was deliberately sending to some other researchers. The reason this surprised him is that he deliberately zipped the malware with a - yup. There's Sisyphus.

Leo: There's Sisyphus, rolling the boulder up the hill.

Steve: That's great artwork.

Leo: This is nice. This is an old '70s animated short about the myth of Sisyphus.

Steve: Yeah. So this researcher had been - I got distracted by the beautiful animation. The researcher had been for years encrypting malware in a ZIP. And the reason you can do that is that the...

Leo: I'm just superimposing Sisyphus over...

Steve: That's great.

Leo: Over your face.

Steve: That's wonderful.

Leo: I'm sorry. Go on. He's still pushing that boulder, by the way, yeah.

Steve: Yes, he's going to be pushing it for a long time. We won't be running out of things to talk about, Leo. There's that for sure. So he'd been zipping his malware in ZIPs, encrypting them using the password "infected," and that's standard industry practice. When I've been downloading these various CryptoLocker malware, they all come, they're all posted, and they come to me zipped and under the password "infected." And what's so nice about ZIP is that it is not just locking the archive. It is encrypting it. So it's encrypting it under that password. And but that's the way it gets through AV scanners is encrypted. So what was discovered just recently is that Google has begun peeking inside ZIPs and checking, probing the ZIP to see if it is encrypted under the - I have to turn off my video, [indiscernible].

Leo: I'm sorry. [Laughing] Can't resist it. Rolling that boulder.

Steve: Those are great.

Leo: [Grunting]

Steve: Poor guy. You feel sorry for him.

Leo: Yeah, yeah. And then of course he gets to the top, and what happens? It just rolls right back down. He's going to start over now.

Steve: [Indiscernible]. So anyway...

Leo: So that's interesting. So Google, now, we know that Google scans inside ZIPs. This is news that they scan inside password-protected ZIPs.

Steve: That's the news. And so this researcher was curious, this Brian Baskin who did this work, he was curious. So he tried - he first reencrypted under a different password, and Google still flagged it. And he said, what? How can that be? So then he did an experiment where he took the most frequently used 20 passwords and put "malware" encrypted under those. And Google only saw the one encrypted under "infected." So what he realized was, once Google tripped over a zipped malware, that was zipped under "infected," it also remembered the filename so that, when he rezipped it under a different password, he had left the filename the same, and Google still said, nope, once caught, never good. So nothing, no matter what he did...

Leo: That's weird.

Steve: Isn't that weird?

Leo: It's using a CRC or something to identify the file? Obviously something beyond the name.

Steve: Probably just the name. It just...

Leo: Oh, it is using the same name.

Steve: We just saw something that was malicious with this name, so we're not allowing the same name to be used again.

Leo: Got it, got it, got it, yeah. I mean, he's trying to protect you, obviously.

Steve: Well, and the idea would be that, if a - see, and this is what's puzzling is that malware researchers, like for example I'm distributing CryptoLocker with ZIPs, with that password. I'm not emailing them to people. But, I mean, you'd have to deliberately type in the password "infected" in order to...

Leo: Right. And that's the point of it.

Steve: Exactly.

Leo: Do you think this is because the password was "infected" that Google just tries that? Or are they able to somehow magically see inside ZIPs that are password protected? No.

Steve: No. And that's just it. They can't because it's true encryption.

Leo: It is really encrypted, yeah.

Steve: Real encryption. So what they're now doing is they must be inspecting...

Leo: They're trying the password.

Steve: Yes. They are trying that password for every ZIP that comes their way.

Leo: That's bizarre. I don't know why you would waste time doing that. Because it's acknowledging that it's infected.

Steve: Yeah, it's saying. I mean...

Leo: No bad guy's going to mail you something with the password "infected."

Steve: Right. And if you open it and execute...

Leo: It's your fault.

Steve: Exactly. Yeah. And that's the theory under which GRC makes the malware available for people who want to test their networks and test their AV, with all kinds of warnings and cautions. Still, hey, you're hopefully an adult. Treat this responsibly.

Leo: I want to hear a response from Google on this.

Steve: Yeah, be interesting.

Leo: I want to know why they're doing - or whether this is a false finding somehow. It's odd. Okay. Don't know. Don't know.

Steve: So CryptoLocker is getting greedy. I posted an updated version of CryptoLocker. Actually, I think there's a newer one that I didn't update my malware page with, which one of my frequent Twitter followers, he saw Simon Zerafa mentioning to me that there was something newer that he'd come across. I think then he messed around with it with a VM. He was a little surprised when, yes, he got the standard CryptoLocker "You Are Locked" screen. You can insert your own four-letter word in place of "lock." And they now want five bitcoins, Leo. Now, remember once upon a time...

Leo: Well, Bitcoin's gone down. So are they really greedy or just adjusting to the fluctuating price of bitcoin?

Steve: Bitcoin has been hurt by Mt. Gox's problems and various other problems and Russia outlawing them and so forth.

Leo: Bitcoin's down on Gox to 285 now.

Steve: Well, Mt. Gox is essentially out of the game at the moment. I mean, it is still a hundred dollars at, like, three other exchanges. It's actually north of - I'm sorry, did I say a hundred? $600, north of $600. And so that's $3,000 is what they're asking for.

Leo: Wow.

Steve: And it was once 300. So it's 10 times - it went from 300 to 3,000 that they're now asking for. Which I think is probably asking too much. I think, yes, as we talked about last week, a law firm lost all their documents. They might pay $3,000 if they were sure they could get it back. But, wow, lots of users are just going to say, okay, well, I just, you know, so we lost all of our photos of Aunt May. Too bad.

So, Bitcoin Protocol. Until recently, we really did believe there were no problems with it. I mean, it has withstood a huge amount of analysis. Now, it turns out that something known as "transaction malleability" has surfaced. And it is what brought Mt. Gox into so much grief. IEEE Spectrum had a nice piece where they explained what's going on. They said: "In order to understand transaction malleability, you need to know that the balances of all Bitcoin addresses are maintained on a public ledger, and that the changes made to this ledger are what constitute a transfer of funds. When a transaction is broadcast to the network, it is relayed with a digital fingerprint that identifies it. Bitcoin miners then scoop it up, verify it, and send it on to the rest of the network for confirmation. Once the transaction has been confirmed, there's no way for that same

person to spend those same bitcoins because they are being checked against the public ledger.

"The malleability feature" - or defect - "allows a person to intervene. Right after the transaction request has been sent, it's possible to modify the fingerprint and create a duplicate transaction. So now you have two unconfirmed transactions flying around the network. They are both for the same exact payment, but they have different fingerprints, and only one of them can be added to the public ledger. So Andreas Antonopoulos, chief security officer for the Blockchain.info Bitcoin wallet, said: 'The first one that is confirmed will be accounted for in the blockchain and will become the definitive record. The other will be dropped as a double spend attempt.'"

So all of that is working. That is, the system is fundamentally working. It turns out that it's an error in the way Mt. Gox was auditing the blockchain to verify transactions, something that they alone were doing in a particular way, so we could blame it on their implication, which allowed attacker to spoof non-successful transactions which were created, deliberately created by this malleability. They could spoof non-successful transactions in order to get Mt. Gox to issue refunds. So the community feels that this is something that needs to be resolved in the protocol, that is, needs to be firmed up, and an agreed-upon way to establish auditing needs to be established within the community.

But here's the point, is all bitcoin users need to do for now to be safe is to deliberately not issue transactions too rapidly. If you allow 10 minutes for a transaction to take hold in the network, then you have nothing to worry about. There's no way then for any mischief to be accomplished by bad guys. But this is a little softer than we thought the Bitcoin protocol was.

Leo: And now the last bit of Sisyphean bad news.

Steve: So we talked about how one of the problems with the WeMo devices is that they're not authenticating SSL certs. It turns out they're not alone. Netcraft, that's been around forever, the really great Internet usage monitoring, characterizing service, has discovered dozens of "rogue," they call them, self-signed SSL certificates used to impersonate high-profile sites. So remember, so what this means is, like, if Facebook, they're not a CA, they're not a certificate authority themselves. So they purchase their certificates from someone, for example, actually I think it is my certificate provider, DigiCert. It was when I noticed that Facebook was using DigiCert that I thought, well, if they are, then it must be recognized by everybody, so I'll use them, too. And as I mentioned last week, I'm so happy with DigiCert.

So the point is that Facebook's certificate will be signed by DigiCert. And the browser knows, has DigiCert's certificate in its list of certificate authorities that it trusts. So the browser is able to verify that a third party, DigiCert, signed Facebook's certificate. A self-signed certificate is, for example, it'll say www.facebook.com, and it'll just be - it'll be signed by Facebook, which really doesn't provide any value because anyone can make their own certificate that they sign.

And in fact there are some websites that want SSL but don't want to pay the freight of having DigiCert or GoDaddy or any of the other cert providers sign their certificates, so they'll self-sign them. And you get an error. Your browser gives you an error when you try to go there, saying, whoa, this certificate is non-trusted. And then you can decide if you want to go ahead and trust it anyway because you'll get the privacy side, the encryption of SSL, but you don't actually know who you're connected to. That is, the

certificate says www.facebook.com. But if you have agreed to accept a certificate that signed itself, then it could be anybody who made that certificate, not Facebook, who had to prove their identity to the certificate authority in order to get the certificate authority to sign their certificate. So the system works. But it only works if there's this chain of trust.

So what has been found is a repository, essentially, of self-signed certs impersonating high-profile sites. And the danger is that there would be applications that real people use that aren't checking, in the same way that WeMo, AC plugs and light switches and so forth, are not checking. What if more important things weren't checking? Well, researchers from Stanford University and the University of Texas at Austin found broken SSL certificate validation in Amazon's EC2 Java library, Amazon's and PayPal's merchant SDKs, integrated shopping carts such as osCommerce and Zen Cart - which are very popular - and AdMob code used by mobile websites.

What that means is that they're accepting SSL connections and not checking to see if the certificate - they're looking to see if it's valid. Does the checksum - is that correct? But they're accepting self-signed certificates. And it also turns out that online banking apps for mobile devices, which are of course tempting targets for man-in-the-middle attacks, are also falling short. They're also not checking certificates. In an analysis that was made, 40% of iOS-based banking apps tested by - and here's the company we talked about earlier, IOActive - are vulnerable to such attacks because they fail - 40% of iOS-based banking apps because they fail to validate the authenticity of SSL certificates presented by the server; 41% of selected Android apps were found to be vulnerable in tests performed at Leibniz University of Hannover and Philipps...

Leo: Leibniz.

Steve: Leibniz. Oh, Leibniz, yeah, of Hannover, and Philipps University of Marburg in Germany. So what we have is a situation where users are trusting, and for whatever reason, apps are initiating SSL connections - so, for example, you would have a Bank of America app. And it's initiating on your behalf a connection to BofA. And I'm just making that up. I don't know if BofA is one of them or not. But, for example, BofA. So you're using the BofA app in your device. And it is giving you privacy because it is using SSL. But when it's connecting to BofA at BofA.com, the BofA is sending back a certificate. The app itself is assuming, sort of like the wrong logic. It knows it's connecting to BofA, and it knows that BofA is valid. So it assumes the certificate that it's going to get from BofA is valid.

But if a man in the middle intercepts that connection, that man in the middle can return a certificate for BofA that is not signed by a real certificate authority, but just it's self-signed. And the app, 40% of apps in iOS, 41% in Android that were checked, are not verifying the remote certificate. They're assuming it's correct. And that assumption opens those apps up to attack by man-in-the-middle impersonation.

And there's no doubt, now that this news is out, that we're going to see people saying, oh, let's - I want to test these apps and find out for myself which ones are vulnerable. And if anyone uses those apps while they are vulnerable, their information is not safe. It could be decrypted by the man in the middle, that then turns around and reencrypts it as it goes off to BofA. So essentially you lose all of the privacy that the app is trying to provide to its user. And, you know, a substantial portion of apps today are not checking, not verifying the SSL certificate chain as they should.

Leo: I don't know if he knew ahead of time about this or he just is prescient or what, but Kaspersky said this exactly. He said banking apps on Android in particular are going to be the next big problem.

Steve: Yeah, yeah.

Leo: I use a - I do a lot of banking, frankly.

Steve: Online?

Leo: On my phone.

Steve: On your phone?

Leo: A lot of it, yeah.

Steve: So we will need to have some - it'll be interesting to see, I mean, it's not hard to do, actually. I bet some researchers will figure out which apps are vulnerable because it'd be nice to know.

Leo: Yeah.

Steve: So we're into my last little notes. I did want to mention that, as I mentioned last week, I mentioned to you, I don't know if we were recording or not, that I was going to see "Ender's Game" for the second time with my buddy on Saturday who had never seen it. I liked it more the second time, I think because my expectations were appropriately set. I had read the book, and of course no movie ever lives up to what the book is. You just can't, in a couple hours, provide as much richness and detail, no matter what you do. But I thought it was fine. I thought it was, I mean, I was much less impressed the first time than I was the second time around. So I just wanted to mention that.

And Jenny and I saw "Robocop" on Monday. I guess that was yesterday. And it far exceeded both of our expectations. I didn't have much expectation, admittedly. But I was very impressed. It was definitely worthwhile. So if people have been on the fence, if they're in love with the original one and are wondering, I think it looked, I mean, it was good. I liked it. And I did see a preview for something called "Transcendence" coming out mid-April. Oh, it looks fabulous.

Leo: And we know how accurate previews are.

Steve: Yeah.

Leo: Not a good indicator of the strength of a movie.

Steve: True, true.

Leo: I always wonder if the best parts were in the preview.

Steve: I don't know how this movie could be bad, given what we saw. It just looked, I mean, it was Daniel Suarez-esque in the notion of, I mean, it's not giving anything away because the previews do, of someone having their consciousness uploaded to the 'Net. And things don't go well, we'll just put it that way.

Leo: Johnny Depp.

Steve: Yes.

Leo: Johnny Depp is in it.

Steve: Yeah, it looks great. And I wanted our listeners to know that the SQRL UI page is up now, No. 6 of, oh, I think there's, like, 20 of them now. And it's where I'm currently working. I've been talking about this the last few weeks, that I am now at work on the UI. And I think you'll see anyone looking at the SQRL UI that I've got designed so far will have, first of all, you'll get a sense for what the typical user will see. And it's simple and easy to use. I mean, that's where I'm struggling is to make it so. But anyway, you can go GRC.com/sqrl, look at the SQRL Login in the main menu. But I did create a bit.ly link that takes you right there: bit.ly/sqrlui. So just "sqrlui."

Anyway, I'm really happy with the way it's coming. It's slow going, but I'm intending, as you'll see there, what you'll see is finished product. I mean, these are - I am designing the experience that the user will have as they use the product. And then it's a matter of wiring up the code behind the UI. So we're making good progress.

Leo: Excellent. Excellent. Steve is the man when it comes to security. We do this show, Security Now!, each and every Tuesday now. Yeah, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 21:00 UTC on TWiT.tv, if you'd like to watch live. If not, don't forget, after the fact, always available audio and video. In fact, Steve does an unusual thing. He orders transcripts. Elaine Farris writes those. He puts those in his website along with 16Kb audio versions at GRC.com.

So when you head over there to get SpinRite, and I know that's why you're going over there because that is the finest, world's finest hard drive maintenance and recovery utility, pick up a copy of Security Now!, as well, and all the other good stuff that Steve does absolutely free.

A Q&A episode next week, news allowing, so you can leave your questions there, too: GRC.com/feedback. And if you want to get full-quality audio and video of this

show, we have it at TWiT.tv/sn. And of course you can subscribe wherever finer podcasts are aggregated. We'll see you next Tuesday. Thank you, Steve.

Steve: Thanks, Leo.