SECURITY NOW!

Transcript of Episode #547

GRC Is Down

Description: Steve and Leo discuss the overzealous DDoS attack ongoing against GRC.com, an ECDH key-stealing exploit, a buffer overflow problem in glibc, innovations in data storage, and Bruce Schneier's Worldwide Survey of Encryption Products.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-547.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-547-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. You know stuff's been going on with his site. He explains what's going on, also talks about some pretty amazing stuff, including an experiment by hackers to show how you could get somebody's secure encryption keys by listening through the wall. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 547, recorded Tuesday, February 16th, 2016: GRC Is Down.

It's time for Security Now!, the show where we protect you and your loved ones online - maybe not ourselves - with Steven Gibson, the man at GRC.com. That's the website for the Gibson Research Corporation where you get SpinRite, the world's best hard drive maintenance and recovery utility. Maybe not today.

Steve Gibson: Well, maybe not.

Leo: Hi-ho, Steverino.

Steve: Yo, Leo. Well, I had a different title for today's show this morning, actually. This morning the title was Security Week Roundup or something. First I was going to talk just about the Elliptic Curve Diffie-Hellman key stealing exploit, which will be revealed on March 3rd at an upcoming security conference, where through a wall separating a laptop that's just sitting idly and researchers on the other side of the wall, the secret key used for GPG encryption was lifted from the laptop wirelessly. But that's not really the top of the news.

Well, and then, then what happened was just this morning came news of a very worrisome, and it's so fresh off the wire we'll be dealing with repercussions, I imagine, for a while. An eight-year-old deeply buried buffer overflow in DNS lookup in the GNU Library that is everywhere. I mean, basically you take the Linux kernel and glibc, you put them together, and that's the Linux API. So this is in IoT stuff. This is in desktops. It is everywhere. And it turns out that a specially crafted reply to a DNS request can take the machine over. So it's like - so then that was going to be the top of the news.

Then GRC went down for the third time, I think, or fourth, depending upon how you count. I've been fighting, well, and losing, because that's what one does these days, a denial of service attack. The first one occurred around 3:30 Saturday afternoon. I was actually visiting an elderly sick neighbor of mine who's been dying of cancer for the last eight or nine years.

Leo: Oh, sorry, yeah.

Steve: And he's so grumpy that I'm - he has no friends, and I'm the only one who will ever volunteer to, like, drive him around.

Leo: Good on you.

Steve: So, well, anyway, so I didn't know it, but we were attacked for the first time when I was over hanging out with him.

Leo: Isn't that sweet. Isn't that special.

Steve: And so I came back home, and GRC was off the air.

Leo: I saw the tweets at the time, and I wasn't sure what was going on. I figured, oh, Steve forgot to renew his certificate or something again, you know.

Steve: Oh, don't I wish.

Leo: Yeah.

Steve: Yeah.

Leo: That'd be a lot easier to fix.

Steve: So I drove over to the datacenter, which is about 50 minutes away. And the light coming in, the light showing traffic coming in was on steady.

Leo: Solid, yeah.

Steve: Yeah, solid. And I have some frontend equipment which, like, looks at traffic and filters it and deals with it in various ways. And coming out of that, it was just kind of like [stuttering sounds], just sort of, like, really sad. And I thought, oh, that's not good. So I then did a packet capture and, in a very short span of just a short time, grabbed 50,000 UDP packets and quickly stopped the capture so it didn't overflow things, and then took a look at it. And it was all inbound DNS responses. So any listeners to this podcast know that we were subject to, victims of, a DNS reflection attack. And this is something, this is not the latest attack, you know, the NTP attacks are...

Leo: Time server attack, yeah.

Steve: Right, the Network Time Protocol. Those have been in vogue a little more recently. The reason, though, that those have sort of been sort of disappearing is that it's possible to reconfigure the NTP servers to avoid that attack, basically bolt them down better. The problem with a DNS reflection attack is that there's really nothing that a DNS server can do. Sort of by its definition, a query packet comes in - and remember the idea was with DNS we want it to be very lightweight, low overhead. So it doesn't use TCP, where you have a three-way handshake to establish communications, to number your packet streams and all that. Instead, it's a very simple query response. And so for that we use UDP, a much lighter weight protocol, where it's just a packet that bounces off of a server with a request to look up the IP address for a given domain. And so the server that receives it does the lookup and then sends the response back to the requester.

Well, if you're able to spoof the source IP of that packet, if a bad guy or, I mean, anybody with an attack tool - and of course these are a dime a dozen now. The big problem with DDoS is that it's trivial to do. It was sort of a novel thing when I was attacked back in 2001. About 15 years ago was when Wicked, that script kiddie Michael, who I managed to track down and found him and broke into his chatroom and said, "Hey, what's up? Why are you bothering me?" It was novel.

Well, of course in years since then the scale of these attacks have increased. And we talk about it from time to time. There are now serious third-party services, like CloudFlare comes up often, that specialize in protecting high-value sites, like gambling sites that have to be up during the big game, and other sites that protect them from DDoS attacks. The problem is those services are - they're more expensive than GRC's total revenue by several factors. We're not a big operation here. It's me and Sue and Greg. And so the future is uncertain, to put it frankly. So what happened was...

Leo: Man, is that effed up.

Steve: I know.

Leo: I just can't believe how just messed up that is. That's, by the way, this is kind of how you got into this whole security thing, right, was a DDoS many years ago.

Steve: It definitely drove me way deeper into the technology. I had ShieldsUP! at the time, and I was sort of doing Internet security stuff. And 15 years ago it was interesting. It was intriguing. It was like, oh, this is interesting, you know. I mean, I'd been talking about it and writing about it, and I'd heard of them, but I'd never been the victim of one.

So it was interesting. But aside from the fact that, I mean, no SpinRite will sell while GRC is down, so it zeroes our revenue and, frankly, potentially puts us out of business. I've got other stuff...

Leo: I feel a little responsible for this because, as you know, we have people who are trying to put us out of business, as well. I don't know what I did to cause that. But...

Steve: Well, and, see, the...

Leo: I hope that that's not the same kind of group of people. I bet it is, though.

Steve: I will never know. There's no attribution behind this. People said, oh, what, you know - okay. So what happened was Saturday...

Leo: Let's take a break. Let's take a break because this is...

Steve: Okay.

Leo: We're going to get going here, and I don't want to...

Steve: We've got a lot more news to talk about.

Leo: We've got other stuff, too. So we'll talk - but actually kind of the anatomy of a DDoS attack would be very interesting. And you do know a lot about what's going on. Despite the fact you don't know who's doing it, you can deduce quite a bit.

Steve: Oh, I - yeah, yeah.

Leo: And we should say this is not a highly skilled thing to do. That's the sad thing is it's...

Steve: And your first comment was, "Are you sure you want to talk about it?" And I have to say I've kept a low profile about attacks because I recognize defense is so expensive, and it is virtually impossible to determine who's behind it without launching an FBI investigation. I mean, and I'm just a small fry compared to the attacks that are going on all the time. And I remember talking to the FBI about it years ago. But anyway, let's do the first sponsorship, and then we'll continue.

Leo: Unbelievable. Just unbelievable. What a world we live in. All right. Back we go. Steverino?

Steve: Okay. So the attack, the first attack was Saturday afternoon around 3:30 Pacific time, lasted about 90 minutes, and then stopped by itself for about half an hour. And then resumed for about 60 minutes and then stopped for, like, four hours. And then started up again in the evening on Saturday. And it was going for maybe 15 minutes, and then it sort of changed.

And I have never needed to get a hold of anyone at Level 3 because GRC has never, in the time that I've been with Level 3, suffered any denial of service attacks. I got to know the guys at Verio, Andy Peterson and John Knowles, really well back in those days, 15 years ago. So I didn't know anyone at Level 3, and this was a Saturday, you know, on Valentine's Day eve. But Level 3 was on the ball. They had monitored these, and the attack was substantial enough that it was beginning to interfere with the traffic of other customers with whom I share what's known as the "aggregation router," the final router that my little subnetwork connects to. So they shut down the interface. They just completely took me down hard.

I got them on the phone. They explained what was going on, and I told them what I knew about the attack. And so I've been working with them for the last couple days. And the attacks have been intermittent. We were able to get back up, I think it was Sunday morning, Valentine's Day, around 10:00. I said, okay, let's, you know, I got them back on the phone, I said let's try bringing it up, and we were okay. And I was hoping, because it was sort of intermittent, I thought, well, you know, maybe someone's just using me as a target to test a new tool. I mean, I haven't done anything to upset anyone, as far as I know. And so I figured, you know, because it wasn't, like, on and steady, but it was sort of coming and going, I thought, well, you know, maybe they're just, you know, I just came to mind. So I was hoping that I'd dropped off of their mind. And we were okay all of yesterday.

And then today at around 11:15 we got hit hard. And I got on the phone with Level 3. And I had tried to log into my management portal, as they call it, that lets me see what's going on. And in fact just now, while you were doing the sponsorship, I did log in, navigated through, and I can now - I'm looking at the chart now, and we had 12.875Gb incoming on the 14th, on Valentine's Day.

Leo: A second?

Steve: Yes, 12.875 billion bits per second.

Leo: So if you had Google Fiber, gigabit fiber, that's 1Gb. You're talking almost 13 times that.

Steve: Thirteen, yes. And so this is why their only recourse is to just shut me down.

Leo: Right, I mean, because...

Steve: I mean, I want to be shut down.

Leo: Because when you buy, you know, when we buy server access, you might get, I mean, it's common to get 100Mb, and usually we get, nowadays you get a gigabit. But the cost for a 13Gb pipe to the Internet that would only get used fractionally 99.99999% of the time is prohibitive. Right?

Steve: Right, right. And, for example, I have a 100Mb interconnect, as it's called, to the router upstream of us. And I buy 10. I have what's called CDR, a Committed Data Rate of 10Mb. So I pay for 10Mb per month, or, well, 10Mb connectivity, essentially. And then it's billed on what's called the 95/5 rule. So, for example, I often see that we're spiking at 50 or 60Mb while somebody downloads Security Now! podcasts because those are the biggest files that I host. But the idea is that, in general, my agreement with Verio is that I'll stay at or below 10 - I mean, not Verio, sorry, with Level 3 - that I'll stay at or below 10Mb in, like, average usage. And if I am above that on a sustained basis, then I get charged for the overage.

So, I mean, and that's fine. That I can afford. The economics works. GRC's up all the time. I've been with Level 3 for at least eight years, never had an outage. I couldn't be any happier with their service. And they've been wonderful to work with during the last three days. But there's not a lot we can do. I mean, this is a world-class 13Gb flood. And it takes, essentially, it takes a class of service far beyond what we've ever needed in order to deal with that. So I've got email into the front office of Level 3 because they do offer mitigation services, but they're expensive. So I don't know how it's going to turn out. An hour or two ago we had to just give up. We had to just unplug from the Internet.

Leo: So mitigation services just give you a fatter pipe, essentially, on demand; right?

Steve: Well, yeah. The idea is that we would - well, and, see, here's the other problem that I didn't mention is we're not a normal website. ShieldsUP! and the DNS Spoofability Test and all these different things that I've built over time, that requires packet-level operation. I'm sending out SYN packets in order to do TCP half-opens to check people's ports. You don't do that. No normal website does that. So it's also very likely that the stuff that GRC has been about is fundamentally incompatible with life behind a proxy.

Basically, proxying these connections are what most websites do. Most websites just need HTTP and HTTPS. That provides them the web services they need. But that's, you know, that's not what GRC is. GRC has been all these other interesting fun things that I've been able to do because I've just had a really good provider that gave me unfiltered access to the raw Internet traffic. So if it were necessary to hide behind some sort of upstream proxy, we couldn't offer any of these additional services. And I don't know that I can afford that.

So anyway, I mean, this is all - you're getting it in real-time as it's happening because I've been dealing with the guys at Level 3. They said, well, we'll see what we can do, but we can't do much for you because you have to buy this stuff that you need. And then they were closed on Washington's Birthday yesterday, so I left voicemail and email. I got an email bounce back saying we're not here. Now I've had some correspondence this morning. But frankly, all of Tuesday is the podcast. So I told Level...

Leo: I'm going to email John Graham-Cumming, our friend John Graham-Cumming over at Cloudflare and see if there's something we could work out, maybe an ad trade or something, to get you some Cloudflare. Because that's the best one. I mean, that's the...

Steve: Well, I mean, the ideal solution is that this stop. I mean, that, you know, I went, we went...

Leo: And you don't know who's doing this? They haven't sent you a note saying "Ha ha"?

Steve: No.

Leo: No.

Steve: No. No idea. And...

Leo: So this could be at any number of - it could have nothing to do with you. It could be, you know, Brian Krebs gets this all the time because it's bragging rights, oh, see, we took down GRC. It could be because of your relationship with us. I mean, I shouldn't be so egocentric as to assume that's it. I hope it's not. But it could be. You just don't know.

Steve: Yeah, well, yeah. I mean, and so I guess, if there's a takeaway from this, there are a couple things. And one is that people should understand how vulnerable websites are to denial of service attacks. It should be a feather in no one's cap that they took GRC down. You know, 100Mb will take GRC down because we just have a simple connection to the Internet. And the economics of that works for us. So it's not like we're some super-hardened high-tech security firm that's, like, hard to kill. We're not.

And the tools, this also requires no skill on the attacker's side. It's a matter of using any Linux or, in some cases, Windows machine because, despite my attempt to get Microsoft not to put raw sockets in Windows, that's what this enables. Raw sockets is what this - you have to have raw sockets to do this. Microsoft went ahead and did that. And then the argument was, well, Linux machines have raw sockets, too. It's like, yeah, they do. And so any tool running on a Linux machine simply sprays DNS queries to publicly available DNS servers.

And DNS servers by definition are publicly available. They're like web servers. They have to be on the Internet in order to serve the people who are querying them for DNS in the same way that web servers provide pages for people who are querying for those. So UDP packets, these little short DNS queries, are sent out with GRC's IP, 4.79.142.200. That's GRC.com. So you simply deliberately rewrite this little IP packet with UDP protocol, saying that that's the source of the query. And you spray these to the world's DNS servers. They get them; and they go, oh, for some reason GRC wants me to look up this IP address, okay, and sends the response. And so the result is that all of the DNS servers that are part of this attack are flooding my one IP with their responses, for which I never asked.

But we've seen incredible traffic aggregation, 13Gb worth of traffic that this attack peaked at on Valentine's Day on Sunday. And in fact Level 3 quoted 10 when I talked to them this morning. That's the first time I had any sense for the scale of this attack because, it's funny, I have had such a perfect experience with Level 3 that I hadn't logged into the portal for five years because everything was working perfectly. So I was unable over the weekend to log in because after six months they just block it until you contact them. So I did that this morning. They gave me access to my own interface at Level 3. And that's when, during that first sponsorship, I saw that - I'm looking at this graph with some serious spikes of bandwidth.

So there is really no way to track this person down. It's like, oh, you know, go figure out who it is. It's nice to say that, but the reality is you can't. Now, and I was thinking a little bit about the whole concern over cybersecurity and so forth. Well, remember that in order for this to be doable, wherever this person or people's networks are, they are allowing packets to egress from their network onto the Internet with an obviously false source IP. Inside of any network, like inside of Cox, it's going to be 24.78 dot something dot something, whatever Cox's range of IPs are. But if the bad guy or a machine under the bad guy's control were generating UDP packets with GRC as their source IP, it's impossible for that to be true as that packet leaves, just to use Cox again, leaves Cox's control, leaves that ISP's network.

The router that is there on the boundary is seeing a packet leaving that says it originated from GRC.com, but it originated inside of Cox's network. So if the router was doing egress filtering, if it was - and it's also known as "reverse path." If it looked at the packet and said, should this be coming from inside this network, and it obviously shouldn't, the packet should be dropped. And if networks were dropping packets with spoofed source IPs, this problem goes away.

Now, the argument is, well, yeah, there are other ways to do DDoSes. You could, for example, swamp a person with page access queries, you know, valid requests for their home page. And if they have a lot of scripting involved, and it's expensive to generate pages, then that'll bring down their server. And that's the case. But something like the whole problem with spoofed source IPs, I mean, we talked about this years ago. Again, I haven't wanted to poke a stick in anyone's eye because I recognize how completely vulnerable I am.

Leo: Well, and how infuriating it is that ISPs aren't doing their job.

Steve: You could argue that, yes, that all...

Leo: There's no excuse for not doing that.

Steve: Right. There really isn't. There are no valid...

Leo: Now, there could be - they could be, you know, in the Philippines. I mean, they could be places where people don't care or don't know any better.

Steve: Ah. And there's a solution for that, too. Because, again, packets - because routers, for example, on the border between the U.S. and the Philippines, the routers know if, you know, like where GRC is. Remember, routers have routing tables. And so they know geographically where networks are. And if packets are coming in from an interface that makes no sense, a router at the border of the U.S. could say, no, sorry, GRC is over here, it's not out there on the other side of the ocean. And so, again, so we could even have international protection. I mean, this idea could work.

But one of the themes of this podcast is inertia. Look at how difficult it was just to retire a hash, you know, the SHA-1 hash. No, no, please, don't stop, we need it for an extra six months, and back and forth and all this. So here we're talking about routers that are already overworked. And this might break some - this might subtly break some things. I mean, there's been lots of dialogue and conversation about this over the years because this is a well-known problem. It is completely fixable, yet there is zero traction on it. And as a consequence, a simple tool from, I mean, this could be, for traffic of this size, I had no idea that we were dealing with a 13Gbps flood.

Leo: It's amazing, isn't it. Wow.

Steve: That's more - yeah. We've seen that one host could generate DNS queries because there is some size amplification. Depending upon the query you generate, the answer could be much larger than the query. And that's a so-called "bandwidth amplification" aspect, where, say, one host, one evil host on its own 50Mb connection, it's using all of its 50Mb to send out UDP queries of a certain size. And these are very - these could be very short, very small packets. But the response can be 50 times larger. I was going to say a hundred, but that's probably - but generally DNS responses are not that big. But they can be larger. But so say it's just 10 to one. So 50Mb outbound, bouncing off of DNS servers all over the place, generates 500Mb inbound to me, and I'm off the 'Net. I mean, that's all she wrote.

Leo: So it could be even just one IP address sourcing all of this traffic.

Steve: Well, no. That's what - until I knew how big this attack was. I thought it...

Leo: If it were a gigabit, maybe, but...

Steve: Well, or if it, yeah, if it were a gig, yes. But 13Gb, so that's a network of hosts, probably compromised bots that are installed in random, unwitting people's machines. And there's someone in control of this botnet. And as somebody tweeted, "Remember, Steve, these are available for rent." You can do DDoS for hire now, too. So there's a business model behind - and of course we've also talked about the extortion. Many times prior to a big, like in the case of a gambling site, a big boxing match, the site will receive email saying, "Pay us X amount of money by this date, or you'll be down for the prizefight." And they elect not to, and sure enough, they're down.

Now, those big sites, then, learned the lesson over time and moved behind DDoS protection. This is not something I have looked into before because I've never had the problem before. We have been eight years at, well, we've been, like, 15 years, actually, attack free, until the last few days. I'm going to have to look into it with Level 3 and see what services they offer.

Leo: Yeah. We buy it for TWiT.

Steve: Okay.

Leo: I mean, you know, that's - but it's part of the cost of business.

Steve: And then of course the question will be, how will it affect what services GRC is able to offer?

Leo: Right, right.

Steve: Because we may just be reduced to being a simple website.

Leo: You know, it's like swatting. You don't want to talk about it if you've been swatted because you don't want to encourage other people to do it because it's trivially easy. All you have to be is an a******.

Steve: Yeah, well, and that's the problem. It's why by no means do I want to present a challenge to the world's botmasters because it's easy to force me off the 'Net. It's not like I have some magic power. There is no such thing in this case. They're just, you know, this is pure, raw traffic flow.

Leo: Well, that's an important point to make, by the way. You have accomplished nothing impressive. Whoever's doing this, you have accomplished nothing impressive.

Steve: Especially for me. I've got a 10Mb...

Leo: Yeah. Steve's not trying to stop you.

Steve: ...CDR with Level 3. Talk about overkill. 13Gb, just use 100Mb. That's all it takes.

Leo: You're overdoing it, guys.

Steve: Yeah. So anyway, no show notes. No podcasts available from me. No services available from GRC.

Leo: I'm so sorry. Your Internet, your personal Internet works.

Steve: No. Well, yeah, yeah, because I'm with Cox, and I've got a great connection with them. But, so, for example, I wasn't able to send email to you guys and Elaine, the show notes and things. I will be able to this afternoon. I'll reconfigure not to use GRC's SMTP server because it's unplugged right now. It's off the 'Net, along with everything else. So I have no idea when, if, or ever GRC will be back. We'll just have to - I will do what I can. But today is podcast day, and that's had my focus. I've explained to Level 3 that I can't even get over to the datacenter to look at the traffic until after the podcast. So we will continue with the show.

Leo: Amazing. How did this - how do we have people like this in the world who are just direly malicious? I mean, you provide - 99% of GRC is a free service and very valuable to a lot of people.

Steve: I think it's, well, but the Internet provides people with anonymity. And this is only another form of the abuse of that anonymity. We see the abuse of that anonymity all the time. I mean, Twitter has been, you know, I've got a fabulous experience with Twitter. I love it. I've got great, great followers, and they provide great information. For me, Twitter is a win. But I know for many people it's just - it's a nightmare. And you look at the postings that are being made to so many blogs which are, again, are just - the conversations devolve quickly into nothing because of anonymity. And so this is - there's, like, zero cost to an attacker in attacking me, and massive cost to me in being attacked. And I would argue massive to the degree that GRC is useful to the industry and to thousands of people. There's a cost to the community for, like, forcing us off the 'Net.

Leo: Right. That's what's very sad about it is, yeah, that's really sad. Okay.

Steve: So. We'll put that aside for the moment, and we'll see what the future holds. I do not know at this point. But if anyone wants to say "I give," I give. I do. This is not something I can afford to fight or plan to fight or have the time to fight. So I hope I don't have to.

There was a really interesting problem found - oh, boy - in a core library. Code was changed back in May of 2008. And this is in what's called glibc. Anybody who's done any work with the C language, which is the majority of programmers, sort of before this recent change to PHP and Python and everything, I mean, C is what Unix was originally written in. C is what Windows was written in. And it's the old-school development language. It's what Mark Thompson and so many other developers choose to write in, even today.

The compiler itself just compiles sort of the armature, the core of the language. It understands four loops and variable definitions and compiling complex arithmetic expressions and that kind of stuff. But anytime you use a function call, that's in a library. It's the C library. And that was - the idea of the language itself being small was one of the brilliant ideas that Kernighan and Ritchie, and there was one other guy in the beginning [Ken Thompson], at AT&T had. And the idea was just create a small language, and then we use an external function library to give it teeth, to allow it to do things. And the beauty of that is, is it's extensible then. So what's happened is over time this C library has grown. And so things like opening a file, allocating a memory buffer to do something with, or even the printf function, those are all in the C library.

So this is part and parcel of any C-based source code. You have the compiler and the C library. You can sometimes dynamically link to it where your program stays small, and it actually makes calls into the library. Or it can be the library, the functions you use can actually be sort of copied into your code so it's bound together. But either way, if it's in C, there's a C library.

So part of the GNU Project was to create a library for its use that was intellectual property, open source, and fit the license requirements that the project wanted and so on. And, for example, any Linux system has the C library as part of it. Now, there are leaner libraries which, because the C library over time has gotten very big and very powerful, many things, because memory's gotten so cheap, just drop the whole library in and don't worry about it. But there are, for example, in lightweight embedded devices that really need to keep themselves slimmed down, they may use a different library than glibc.

But since May of 2008, so coming up on eight years ago, introduced in v2.9 of glibc was a mistake. And the mistake was that, if the program needed to resolve - I'm looking at the irony of this because it's DNS, which the one look I had at our traffic on Saturday was a DNS reflection attack, as I first mentioned. And this is a mistake in DNS. If the program or the IoT device or the desktop machine or the server or whatever it is, I mean, that's how ubiquitous this is. And that's why this is a bit breathtaking, and that's just happened this morning, the news of this came. If that thing looks up a DNS address in sort of the default way, the default way is to ask for either an IPv4 or IPv6, in DNS parlance, there's an A record - the A stands for address, you're asking for the address - or an AAAA. And I always got a kick out of that because the IPv4 is a 32-bit address. So it's the original A record.

Well, since IPv6 is 128 bits, that's four times 32, so they named it AAAA instead of like, you know, A2 or AA or something, to represent the fact that it's actually the equivalent of four A responses since it's IPv6. So if the program, as most do, just says look up this IP address, and sends out both the A and the AAAA query, a subtle mistake in the way the buffers are managed allows the responder or - and I have to say it, Leo, I loved when Rene said "person in the middle."

Leo: Well, you know Rene. He's Canadian. Very politically correct, yeah. It's a person in the middle.

Steve: Oh, I loved it; you know? I thought, okay, we're being gender neutral. It's not a man-in-the-middle attack. For Rene, it's a person in the middle. So it's like, yes, we don't know whether you are a man or a woman, but we wish you weren't in the middle no matter who you are and how you put your pants on.

So the problem is that this is DNS. And remember that DNS is not in a - there's no security wrapper for it. Unlike HTTP, where there is a security wrapper, if we use HTTPS, we have TLS to protect the protocol. There's no wrapper on DNS. DNS packets are just out there in the wind.

So this is extremely simple to MITM, or PITM, I guess, for Rene, person in the middle. And the reason is, if somebody were sniffing the traffic, all they have to do is respond before the actual response comes. That is, somebody who's in the middle is almost by definition nearer to the questioning device than the DNS server that those queries are bound for. So what that means is that person who sees an outbound, the bad guy that sees the outbound DNS query simply responds maliciously and can take over the device. I mean, this is so you get a sense for how bad this potentially is. And so the Google security folks did a blog posting this morning, and they said - and some of this is recap, but this is big enough, I want to share this.

They said: "Any Unix-like operating system needs a C library, the library which defines the system calls and any other basic facilities such as file or device open, malloc, printf, et cetera. The GNU C library is used as the C library in the GNU system and in GNU Linux systems, as well as many other systems that use Linux as the kernel." And of course even more widely than that.

And they wrote: "The Linux kernel plus the GNU C library form the Linux API. All the versions of glib since 2.9 are affected." And so in their blog they said: "Have you ever been deep in the mines of debugging and suddenly realized that you were staring at something far more interesting than you were expecting? You're not alone. Recently a Google engineer noticed that their SSH client segfaulted" - that's the equivalent of a blue screen. A segment fault is where an access is made outside of the valid range that has been allocated for a segment of memory - "segfaulted every time they tried to connect to a specific host. That engineer filed a ticket to investigate the behavior, and after an intense investigation we discovered the issue lay in glibc and not in SSH as we were expecting. Thanks to this engineer's keen observation, we were able to determine that the issue could result in remote code execution." Cue spooky music.

"We immediately began an in-depth analysis of the issue to determine whether it could be exploited, and possible fixes. We saw this as a challenge, and after some intense hacking sessions we were able to craft a full working exploit." So again, that means that the world is currently full of DNS query functions everywhere where, probably if the nature of the target were known, that is, you'd have - the malicious guy would have to know what was making the query. But given that, responding with a special malicious packet or packets can give the attacker control. This is really bad. And it's, again, a perfect case in point of why appliances that are connected to the Internet will have this in them, and they have to be maintained over time. Here's something that has been in place unseen for eight years. So it is, by definition, everywhere.

Continuing with their posting: "In the course of our investigation and to our surprise, we learned that the glibc maintainers had previously been alerted of the issue via their bug tracker in July of 2015," so last summer. "We couldn't immediately tell whether the bug fix was underway." Oh, and by the way, the reason, it was so critical they took it out of their bug tracking - that is, the glib guys did - because it's like, we can't even let this get known.

Leo: Whoops.

Steve: Uh-huh. So, says Google: "We worked hard to make sure we understood the issue and then reached out to the glib maintainers. To our delight, Florian Weimer and Carlos O'Donell of Red Hat had also" - and those are two of the maintainers of this glibc library - "had also been studying the bug's impact, albeit completely independently. Due to the sensitive nature of the issue, the investigation, patch creation, and regression tests performed primarily by Florian and Carlos had continued 'off-bug.' This was an amazing coincidence, and thanks to their hard work and cooperation, we were able to translate both teams' knowledge into a comprehensive patch and regression test to protect glibc users."

So they say: "Our initial investigations showed that the issue affected all the versions of glibc since 2.9. You should definitely update if you are on an older version, though." Okay. So what they're saying is that, if you had something older than eight years ago, there are so many other problems with that, that don't stay where you are just because of this. Get the one that is newer, that fixes everything that's been fixed in the last eight years and this, too.

And then they say: "If the vulnerability is detected, machine owners may wish to take steps to mitigate the risk." In their posting they have a proof-of-concept detection. At this point it's all - this is just, again, this is just a few hours old. So I've got a link in the show notes which I'm not sure how I'll get them to you, or to anyone. Well, I'm sure - oh, in fact, if you just google right now, if you google "glibc," the first hit that at least came up for me was this because it's already tracking as something that is very serious. So if you just google "glibc" you get the link to this.

And I'm sure shortly there will be some safe proof-of-concept tests that a user could use, hopefully. I wonder how that would work? You'd have to respond to a DNS query from your system. Maybe what could happen is that somebody could create a service, they could do a benign DNS server so that you do a DNS query to a benign test server, which will send back packets that will crash your computer. Actually, that's probably what it would do. It would crash your machine. But in doing that you would definitely know the glibc that had been bound into your system or your whatever, you know, your Amazon Echo was vulnerable to this and could potentially in the future be taken over until this is patched.

So they said: "The glibc DNS client-side resolver, which is what this is, is vulnerable to a stack-based buffer overflow." And the function is get address info, getaddrinfo(). And, for example, I use that in the Windows Winsock Library all the time. That's the way you make a DNS query in many API environments. So the getaddrinfo() library function. "Software using this function" - and I should say that which is anything that connects to the Internet that isn't hard-coded with an IP address will need to do a DNS lookup. And most things do because IPs may change. And if some device was hardcoded, it would be stranded.

So they say: "Software using this function may be exploited with attacker-controlled domain names, attacker-controlled DNS servers, or through a" - and they said man-in-the-middle. We're not being gender-specific here, so "person-in-the-middle attack. The getaddr function is used for DNS lookup. Due to a mismanagement of the receiving buffers, the replies from parallel A and AAAA queries may overwrite the receiving buffers." Now, these conversations, the glibc stuff, points to a patch at Sourceware.org. So they've got a patch that fixes it.

What they were able to verify was that at most 64K of data could be force-fed into the querying party, which is catastrophic. In a simple test, they were able to gain control of the instruction pointer in order to point it where they want to. So that means that you load your payload of up to 64K and then point the instruction pointer, not where the requesting device wants it to be, but into the attacking payload, and you're in control. So, staggeringly worrisome. I don't see anything that mitigates this.

Now, they say, if you were to do things like have a firewall which would drop DNS packets greater than 512 bytes, that would be good. That's not clear. Oh, yeah, I think it is clear that this requires a larger than 2K response. So the DNS response must be 2048+ bytes. So, and that's an unusually large, although not impossibly large, but an unusually large DNS reply. So if you had something upstream that was blocking larger packets, that might break some valid queries, but it would definitely block the invalid ones. So, for example, maybe ISPs will consider blocking incoming UDP answers of a size of, maybe, now, they said greater than 512. I don't know why they wouldn't block it greater than 2047. Who knows?

Anyway, the industry is still responding, or maybe getting ready to respond to this. I think we'll know more in coming weeks. But, wow. Understandably, the reason it floated to the top of the news for the week is it's the worst-case scenario. It's a function core to Internet-connected devices, in probably every Internet-connected device, been there for the last eight years, so that's all Internet-connected devices. We didn't have IoT eight years ago. And it does require malicious traffic. But as these guys noted, not even a man in the middle, if you have a malicious DNS server, then an advertisement or JavaScript or a web page, anything that asks a browser to do a lookup would fall through that library, cause your system to look up a DNS query. The response from the malicious DNS server would then crash in the worst, or in the best case, or in the worst case take over your system. So, I mean, I can't think of anything. I mean, this is as bad as it gets, frankly, in a problem that is, like, almost completely ubiquitous.

Leo: Yeah, that's really the scary thing. I mean, every system has this on there.

Steve: Yeah.

Leo: It's not - you wouldn't have a glibc on Windows or - you'd have it on Mac. I should check my Mac because it's Unix-based. But I don't know if you'd have it on Windows.

Steve: Mac is FreeBSD, as I understand. And I think there's a variation of glibc for FreeBSD. But there's been some evolution over time, so I don't know where they stand now. It may have been forked. There was a fork of glibc that happened at one point, and I was just trying to absorb a lot of information this morning to put all this together. And we'll know much more in the coming days. And I'll certainly have greatly expanded coverage of this. Probably next week we'll know way more. But, yeah.

And what I'm hoping is someone will jump on - it would be very simple, there is a proof of concept showing a packet that will crash a device. So it would be trivial for someone like a Kaminsky or somebody to bring up a test server that says, you know, you open a DOS box or a command prompt, and you do an nslookup of this particular, you know, it would be, you know, I-am-a-malicious-DNS-response.com, and if it crashes your system, you know you're vulnerable because a DNS response should not be able to do that. So hopefully we'll have a test before long.

Leo: You're right. According to [Rorx] in our chatroom, OS X uses a libSystem, which includes a FreeBSD version of libc. The G in glibc is GNU, and of course FreeBSD is not a GNU product. So that would make sense.

Steve: Right, right.

Leo: Let's take a little break. When we come back, speaking of Macs, Error 53, Steve's take. I can't wait to hear it.

Steve: Well, and actually it's some interesting informed response from two of our listeners, so...

Leo: Ah.

Steve: Yeah.

Leo: All right. Okay, Steverino.

Steve: So a listener who is actually an Apple i-device repair person responded to our discussion of Error 53 last week and DM'd me. This is Dale Perkel. He said: "Steve, I am an infosec professional during the day" - thus a podcast listener - "and an independent Apple repair business at night. In South Africa, where I live, there's very little official Apple support, except possibly from the mobile operators who charge huge amounts, and generally repair time is on the order of two weeks. I offer an overnight service at an affordable price and hence have repaired hundreds of Apple devices in the past two years.

"Error 53 is extremely frustrating, as there are legitimate reasons to replace the Touch ID sensor, such as accidental or liquid damage, or failure of the home button mechanism. I always advise my customer that this will void the warranty, if there is one, and Touch ID will be disabled, to which they consent. Why, when an update is then installed, weeks or months later, should the iPhone be irrevocably bricked, with no possibility of recovering data? If there were criminal or unauthorized access happening, surely the attacker wouldn't take the opportunity to upgrade to the latest iOS. The window of opportunity is essentially infinite for a determined attacker.

"So why can the iOS update not just detect the missing or changed fingerprint reader and destroy the contents of the secure enclave without turning the entire iPhone into a very expensive paperweight? I trust Apple to provide the best possible security, and they rarely fail. But this is akin to needing to buy a new car if you lose your car keys."

Leo: Somebody else says, if you lose your key to the door, the locksmith doesn't come and throw the door out. He fixes the lock.

Steve: Right. And he says, "Thanks for the great show. I look forward to it every week. Cheers, Dale." And I'll just add that, and I won't go through this, the second one actually was from a listener who gave me a link to a really neat woman who repairs iPhones, I guess as her business, because she's got a YouTube video talking about this problem with the Error 53, and it shows a stereomicroscope and lots of equipment used for dealing with this. And she goes on at some length, I think it was a 53-minute-long video, so not worth sharing. But basically it's that she explains something that I didn't quite understand, which was that the iPhone 5s, which first had the fingerprint reader, did not behave this way. The bricking is new with the iPhone 6.

So this is one of the things that caught people off guard initially was that you can have the screen change. You can change the button. Some people, like just for vanity reasons, they want a black face with a white home button or vice versa, you know, they want something special. Or they need the button replaced and so a third-party site does it and says, you know, this is going to - your Touch ID won't work anymore, and they go, oh, yeah, okay, fine. Better than not having, I mean, not having a home button work at all. But then the 5 Series phones didn't care. That stuff was deactivated, but the phone continued.

Leo: Yeah, 5 didn't have Apple Pay. Only the 5s.

Steve: Right.

Leo: I think it has to do with Apple Pay.

Steve: Okay. So what I understand is the 5s did not do this.

Leo: Oh, okay.

Steve: It was the 6 that...

Leo: It wasn't till the 6; all right.

Steve: Yes. Yeah. So anyway, I was hoping...

Leo: That would actually just lead me to think it's a bug as opposed to something Apple intended to do. Apple, of course, isn't being very...

Steve: And what has happened since? I've not kept in touch.

Leo: Nothing.

Steve: I thought maybe - no kidding. So, wow.

Leo: I mean, Apple, I think Apple will do something. Apple never moves fast in this kind of thing.

Steve: And what's their official position on third-party service? Are they anti-third-party service?

Leo: Yeah. They don't sell - you can't buy official Apple parts.

Steve: Interesting. So where are the people - so they're cloned parts?

Leo: I don't know where they get them. I don't know where they get them.

Steve: They must be cloned parts.

Leo: Apple, yeah, I'm not sure exactly what the story is with that. Am I wrong? But I believe - I'll ask - I'm asking the chatroom. I believe that, you know, iFixit, Kyle Wiens at iFixit has had some stuff to say about this. You know, there is this right to repair movement, where you should have the right to repair your own stuff. And of course iFixit makes the tools available, makes the parts available, makes the manuals available. But even Kyle says, you know, I don't think this was intentional on Apple's part. Because there's lots of ways you could handle this that wouldn't brick the phone, but would protect security.

Steve: Oh, and Leo, don't ever, don't ever try to delaminate the screen of an iPhone. Jenny brought me her cracked faceplate.

Leo: Oh, no, you have to replace the whole - it's a unit.

Steve: And all she had, she said, "Here, Steve. Can you, like, replace the screen?" And so I sort of, I started to, and then I saw the instructions that involved the blow dryer to heat the glue. It's like, oh, my lord.

Leo: Apple doesn't want you going inside.

Steve: Oh, no, no, no.

Leo: They even have proprietary machines in the back of the Apple Store to do a lot of this stuff, to separate it and stuff. They've designed machines. They, no, Apple's never wanted, going back to the Macintosh, where they used Torx screws because it was so rare to find a Torx head. No, Steve never wanted you to get in those things.

So, and Apple's always had a kind of a mixed relationship with even Apple resellers. Essentially they've put them all out of business. My friend Tom Santos, who ran MACadam in San Francisco, was always complaining because he could never match Apple's prices. Apple's always going to cost him more than it costs. So Apple's never really liked the third parties that much. I don't think they did this to the third parties. I don't think that - that's what it looks like, but I don't think that's what they did.

Steve: So if they continue the policy, then what this is, is like a wakeup call event. And the new wisdom will be, well, if you really insist on using a third party to replace your button, absolutely never update your phone.

Leo: Yeah. But even that's going to, I mean, Apple does updates without asking all the time.

Steve: Yeah, yeah, yeah.

Leo: You don't own your phone. When you buy a iPhone particularly, you don't own your phone. Some manufacturers, even on Android, some manufacturers are worse than others. But Samsung has a similar situation on their Galaxies. They have the Knox Enclave. They have a fingerprint reader. If this happens on a Samsung phone, you can still use the phone. You just can't use Samsung Pay, you know.

Steve: Yeah.

Leo: But Apple's, you know, there's such a brisk resale market for stolen iPhones that Apple, I think, is responding to that, as well. That's why they have, in effect, a kill switch. You can't take an iPhone and reactivate it unless you know the original password and all these things. And I think this is part of that. I don't think the Error 53 is intentional. I think that's...

Steve: And for what it's worth, from a security design...

Leo: It's the right thing to do.

Steve: Yes. It is absolutely. We on this podcast would stand behind the concept that you mess with a fingerprint reader, then all bets are off.

Leo: Right.

Steve: So, yeah. Although, you know, I agree with Dale that wiping the enclave, that is, wiping the secure information store which was tied to the input device, to me that's - if Apple wanted to allow people to do what you've explained they really don't want to allow people to do, they could. But again, I don't, as you've said, there's really no motivation on their part. They're not saying this is third-party serviceable. So take it to Apple if you want it fixed, and we're sorry if there aren't any nearby with convenient hours and if the backlog is five months. Wow.

So I did, in the wake of our conversations about the whole Get Windows 10 thing, I have two things. First I want to update us on that ridiculous CheesusCrust...

Leo: CheesusCrust. Now, you saw Ed Bott's takedown, yeah.

Steve: Yes, exactly. But in the wake of me mentioning last week that I had just set up a new Windows 7 machine and had used the sanctioned Microsoft update which added the switches, and in my experience completely and beautifully solved the problem, many people tweeted, saying, where was that again? How do I find that? And so I felt it was so important that I, at the time, created a bit.ly link so that everyone would always be able to find it. So it is bit.ly/no-gwx.

Leo: This is the GWX Control Panel, right, that we've talked about.

Steve: No.

Leo: This is something else.

Steve: That's the beauty, yes, this is Microsoft's sanctioned means, and it's beautiful. So what I did was I set up a brand new Win7 Ultimate. And then, before installing any - and that was Service Pack 1, which is the latest and last service pack. I then added this one update that they provide. There's one for Windows 7, and there's a different one for 8.1. And so when you do that, you get a couple new features that allow you to then disable, formally disable all GWX from then on. And so I did that and then went patch happy. It was 144 patches the first round, and I did the optional ones that made sense and so forth, and never had a problem.

So again, bit.ly/no-gwx. And that will redirect you to a Windows page where you can find the official way of doing it. And you don't need the GWX Control Panel. There is much more to it than that. Unfortunately, because they don't really want you to do this, there's no, like, the Fixit button that we've talked about. They could easily have made this a Fixit button, where you just press it, and it says, okay, good.

Leo: It's just a registry, it's a registry script, basically, yeah.

Steve: Right. And in this case it adds some features. It is a deeper change that adds features to the Group Policy system and the registry. But you still have to go in, and then use GP Edit and navigate down into the tree, find a new item that this patch adds to Group Policy, to say no, never ask, never do anything to do with Get Windows 10. And it's effective. So bit.ly/no-gwx. And write it down, and it works.

Okay. So last week - and I have to apologize because I didn't give the story the focus I should have, as we all know that I missed a crucial line in CheesusCrust's posting where he said that he had turned off the three pages of privacy settings, and this was all the traffic that was generated. But I was just intending to cover the story. I didn't go into it any deeper. Ed Bott at ZDNet did a beautiful takedown of Forbes' hair-on-fire, oh my god, Windows 10 is even worse than you thought story. And there's been lots of retractions and backings away, and apparently CheesusCrust has disappeared and so forth.

Leo: He wisely pulled the site down.

Steve: Yes, yeah. The number one issue was where, as Ed wrote, Mr. Crust...

Leo: Mr. Crust.

Steve: "Mr. Crust reports he 'configured the DD-WRT router to drop and log all connection attempts via IP tables through the DD-WRT router that was being produced by Windows 10 Enterprise.'" Well, now anyone who's network-aware is going, oh, well, of course it generated a lot of traffic because it kept trying.

Leo: It keeps trying.

Steve: Yes. It wasn't that it was actually wanting to communicate that many connection attempts. It's that they were being blocked. So rather than succeeding with whatever little conversation it wanted to have with Microsoft or with the CDN provider that they were using, and then it would have said, okay, fine, it didn't understand why it wasn't able to communicate. And so it just kept doing so. So what we don't know, I mean, it would be nice to have a much more responsible reporter who understands the way Windows 10 Enterprise is going to work.

Oh, and Ed also notes that it's not clear why he chose Windows 10 - "he" meaning Mr. Crust, or we'll just call him Cheesus - why he chose Windows 10 Enterprise. It's not an end-user machine. But Ed notes that the Enterprise version has way more power that an enterprise would, I mean, could and would use to quiet down all of this telemetry stuff, if that's what they wanted to do, down in the Group Policy stuff. So, yeah, there's those three little pages of friendly setup time stuff that you're able to turn off. But you can go far more deeply using Group Policy, which is what an enterprise would do if it wanted to turn things down. So anyway, it was a bogus story.

Leo: Also the big one that had 1,339 hits was a Teredo server, which is an IPv6 to IPv4 conversion server. And of course if you...

Steve: Right.

Leo: It's not giving anything up to Microsoft. This is how it's - this is normal.

Steve: Right.

Leo: If you say block it, it's going to keep trying.

Steve: I think they were UDP on port 137, and...

Leo: Yeah, it was doing NetBIOS stuff, too, yeah.

Steve: Yeah. So anyway...

Leo: Nothing to worry about.

Steve: I should have spent more time on the story. I apologize for that last week. But now we're up to speed and current.

Our friend Bruce Schneier did something that I think is brilliant. In his most recent blog, and I have a link in the show notes, he did a worldwide encryption product survey. He provides it in several different forms, one as an Excel spreadsheet. But I love it because I can't think of anything that could better drive home to our otherwise rather clueless U.S. legislators that encryption is already out of the box. Encryption has escaped. Encryption is not something you can legislate against. I mean, they tried, back in the make the keys only 40 bits for export on SSL, back in the very first days. And then that decision, as we've talked about, is still hurting us because we had weakened security for so long.

So he wrote: "Today I released my Worldwide Survey of Encryption Products. The findings of this survey identified 619 entities that sell encryption products. Of those, 412, or about two-thirds, are outside the United States, calling into question the efficacy of any U.S. mandates forcing backdoors for law enforcement access. It also showed that anyone who wants to avoid U.S. surveillance has over 567 competing products to choose from. These foreign products offer a wide variety of secure applications, from voice encryption to text message encryption, file encryption, network traffic encryption, anonymous currency, you name it, providing the same levels of security as U.S. products do today."

And then he pulled out some bullet points. He says: "There are at least 865 hardware or software products incorporating encryption, from 55 different countries." So you can even choose the country of origin. "This includes 546 encryption products from outside the U.S. representing two-thirds of the total. The most common non-U.S. country for encryption products is Germany, with 112 products. This is followed by the U.K., Canada, France, and Sweden, in that order. The five most common countries for encryption products, including the U.S., account for two-thirds of the total. But smaller countries like Algeria, Argentina, Belize, the British Virgin Islands, Chile, Cyprus, Estonia, Iraq, Malaysia, St. Kitts and Nevis" - whatever that is. Nevis?

Leo: Yeah, Nevis. It's in the...

Steve: Is that somewhere you've gone?

Leo: Yeah, yeah.

Steve: Nevis.

Leo: Caribbean.

Steve: Oh, Nevis.

Leo: Yeah, it's beautiful.

Steve: Of course it is. Sounds beautiful. Tanzania and Thailand.

Leo: Tanzania, yeah.

Steve: Tanzania. Actually, I practiced pronouncing that before the show and then forgot.

Leo: I know. It's hard to get the syllables right on that, yeah.

Steve: "...and Thailand each produce at least one encryption product." All of those crazy, like Estonia, well, not - Estonia's actually a techno leader. I'm surprised they don't...

Leo: No, in fact, you know, that's the one you're going to have to worry about because they love this stuff. They're very literate and very smart, yeah.

Steve: Yeah. And he says: "Of the 546 foreign encryption products we found, 56% are available for sale, and 44% are free; 66% are proprietary, and 34% are open source. Some for-sale products also have a free version. At least 587 entities, primarily companies, either sell or give away encryption products. Of those, 374, or about two-thirds, are outside the U.S. And finally, of the 546 foreign encryption products, 47 are file encryption, 86 are email encryption, 104 are message encryption, 35 are voice encryption, and 61 are virtual private networking products."

So I thought this was great. Hopefully this will turn up in some congressional testimony where this serves to demonstrate that third-party solutions are available, and those will get used by anyone who wants them, anywhere in the world. Unfortunately, encryption is math, and it is free, and it is available.

Leo: Yup.

Steve: So my favorite person - oh, and I forgot to put a photo in the show notes because I just - it gives me the willies every time I look at his face. If you can, click the link in the show notes.

Leo: My computer has crashed, so I can't.

Steve: Ah, okay. Oh, I hope that wasn't...

Leo: We just have to imagine it.

Steve: Okay, fine.

Leo: It's a gray screen of death.

Steve: Anyway, it's my favorite person, James Clapper, who really...

Leo: Oh, lord.

Steve: ...really upset me - and we've talked about it on the podcast, our listeners know - when he just bold-facedly lied to the Senate when he said that - this is back in the pre-Snowden, remember, this was like six months before the Snowden revelations, when he thought he could get away with lying and saying that we're not collecting any data on Americans.

Leo: Of course not.

Steve: Is, you know, what he said. And he's like, scratching his head. And, you know, Jon Stewart at the time said - showed it several times. And he says, "This is what's known as a 'tell.'" Anyway, so he's at it again. The Guardian reports, and I'll just - this is short. I'll share this. They wrote: "The U.S. intelligence chief has acknowledged for the first time that agencies might use a new generation of smart household devices to increase their surveillance capabilities." Yeah, yes, yes.

Leo: Shocking.

Steve: Yeah, I know. Shocking, but he actually said it is what's amazing. "As increasing numbers of devices connect to the Internet and to one another, the so-called Internet of Things promises consumers increased convenience. The remotely operated thermostat from Google-owned NEST is a leading example. But as home computing migrates away from the laptop, the tablet, and the smartphone, experts warn" - and we've been doing that here - "that the security features on the coming wave of automobiles, dishwashers, and alarm systems lag far behind.

"In an appearance at a Washington think-tank last month, the Director of the National Security Agency" - now, that's a different guy, that's our friend Admiral Mike Rogers, and he's also no friend of privacy - "said that it was time to consider making the home devices more defensible" - whatever that means - "but did not address the opportunities that increased numbers and even categories of connected devices provide to his surveillance agency. However, James Clapper, the U.S. Director of National Intelligence" - and I can't believe he still has his job - "was more direct in testimony submitted to the Senate on Tuesday as part of an assessment of threats facing the United States."

Clapper said: "In the future, intelligence services might use the Internet of Things for identification, surveillance, monitoring, location tracking, and targeting for recruitment or to gain access to networks or user credentials." So, yes, our friend is still at it. And I know that our listeners are interested in sequestering their IoT stuff on its own network where it can't get to the rest of their network. And, boy, you know, with something like this glibc vulnerability, which is, I mean, a perfect example of why this is important because all that has to happen is some IoT device is identified that has glibc in it, and anyone sniffing traffic of that device going out to resolve the IP of its mothership responds with a malicious packet, and they're in and on your network. Ugh.

A fun little non-security and also non-shivering tidbit just popped up. This will be a presentation given tomorrow, Wednesday, so that would be the 17th, at tomorrow's Society for Optical Engineering Conference in San Francisco, reporting on a group that have developed what they call "5D Data Storage by Ultrafast Laser Writing in Glass." Yes, folks, we are approaching the glass cube storage technology that we've been talking about and wanting for about 10 years.

So the story that covered this wrote that: "Scientists at the University of Southampton have made a major step forward in the development of digital data storage that is capable of surviving" - and here it is - "for billions of years. Using [what they call] nanostructured glass, scientists from the university's Optoelectronics Research Centre have developed the recording and retrieval process of" - again, they call it five-dimensional, I'll explain why in a second - "digital data by femtosecond laser writing."

Femtosecond, that's very fast. The storage allows unprecedented properties including 360TB per disk data capacity. And if we're talking about the disk they showed in the article, it's like a one-inch diameter disk. It was a tiny little thing. It's actually not glass. Well, it's quartz. And so it's good glass. So what they're demonstrating is, in this small disk, they can now - so we're getting sort of down to the near molecular level, 360TB per little tiny disk, with thermal stability up to a thousand degrees Centigrade, and virtually unlimited lifetime at room temperature. Or, if at 190 degrees C, it would only then last for 13.8 billion years. So...

Leo: Now, you hit it with a hammer, it's another matter entirely.

Steve: Exactly. And of course we also know that, if the reader breaks, well, that's a problem.

Leo: Yeah.

Steve: Because you've actually got to...

Leo: But at least the medium is okay. Yeah? That's good, yeah.

Steve: Yes, isn't that cool? I mean, we've talked about long-term archiving and what a problem...

Leo: Well, and CDs and DVDs are not; right?

Steve: Right. They are not.

Leo: No magnetic material is.

Steve: Nope.

Leo: And CDs and DVDs corrupt over time.

Steve: Yup, yup. So they said this opens "a new era of eternal data archiving."

Leo: Now, you've got to store it in a way that would be like an alien technology, or future generations could look at it and go, oh, that's probably ones and zeroes. Maybe we can decode this. It'd have to have something kind of obvious; right?

Steve: Right, right, right.

Leo: Not a ZIP disk.

Steve: And so they're calling it a self-assembled nanostructure, created in infused quartz. And the reason they used this 5D is that you have, first of all, the regular 3D that we're familiar with - length, height, and width - so its location in three-space in this crystal. And then they're able to encode a size and orientation which is absolutely, I mean, when I say stable, we're talking incredibly stable, 13.8 billion years stable at 190 degrees and, like, forever at room temperature. So it's very cool. Basically, so it's laser writing a blank quartz disk, and that's 360TB per little disk, and lasts forever. So, again, this is engineering paper, really cool technology. It'll be some time, and maybe it will never see the light of day. Maybe something will eclipse it. But wow, very cool. And absolutely archival.

Two pieces of errata. Last week I kept saying text-to-speech, text-to-speech, text-to-speech when we were talking about the Amazon Echo. And that's of course the reverse direction of what I meant. Text to speech is text in, speech out. I was talking about speech recognition and the various technologies and, you know, like does the Echo listen and send back, you know, is it sending it all back to the mothership, or is it doing local recognition and so forth.

Leo: I should have noticed that, too. Speech to text, obviously, yeah.

Steve: I just wanted, yeah, exactly. And then a Twitterer, DangerDad is his handle, he just corrected me. He said I was referring to 80 billion degrees of the German Stellarator.

Leo: I noted that.

Steve: Yeah. I meant million.

Leo: I thought that was very high.

Steve: Oh, yeah.

Leo: Hey, a million, a billion, when you get that hot...

Steve: Eh, you know. Oh, and I also said it was sevenfold symmetry because it's got a seven in the name. And so that tripped me up. I knew that it was fivefold symmetry, so I also wanted to fix that because I had said sevenfold symmetry.

Some real quick updates, just because our listeners are demonstrating a real interest on the Zeo EEG headband. Last week, when we last spoke, there had been three auctions, for 100, then 355, then 460, totaling 915 that that URT outlet seller or reseller had sold. And I just refreshed the page. They have now sold - they created a new auction, I think it was either - they had just created it when - I think they had just created it shortly after we started talking one week ago. That auction has now sold 1,208 of the devices.

Leo: They've got to be running out at some point.

Steve: Yeah. They privately told me they had about 1,600. They had finally gotten around to counting them because they figured, you know, we'd better count these. We may actually sell these in our lifetime. So they've got a few hundred left at this point. And I'm really glad because what I'm seeing is the rate of sales is tapering down. They'll still have a couple hundred probably for stragglers of the podcast who, like, oh, my god, I want one. But what it means is that pretty much everyone who is listening to the podcast and thinks this would be cool has been able to buy, has been able to get one. Because it is neat. It's only $40 for the whole system. You need an Android device. But I just - it would have really annoyed me if they'd only had a hundred, yet that would have never satisfied, obviously, now, because it's been, like, I don't know, 2,500 or so have been sold to our listeners. And it is slowing down. So it looks like people who want them, have them, and that's all I could ask for.

Leo: The power of the Security Now!.

Steve: And finally...

Leo: Advertisers, listen up. Next time pay. Oh, sorry.

Steve: From Beirut, Lebanon. Or is it Beirut?

Leo: It's Beirut. Beirut.

Steve: Beirut, Lebanon. And this individual, thank you, asked me to withhold his name because I could never have pronounced it. His last name took up about half the line. And so thank you. But he is, actually, I'm not sure he's a SpinRite user. But he was wondering, the subject was "Files gone from hard disk." And he wrote: "Hello, Steve. I have a two-year-old desktop that suddenly started causing problems. For example, it would show the 'recover the Windows option' during boot sometimes, and sometimes it would just boot normal. After a couple of times, the recovery option started popping up every time, and the recovery options were of no use.

"I took the hard disk off the desktop and connected to my laptop using a SATA USB extension to get the data out of it. At this point, to my surprise, some data were present, and some others were not, even though I was sure the data was there, and I have not deleted them. Is this even possible? I mean, do bad sectors cause data to just disappear? I ran a recovery software" - but it sounds like not SpinRite - "to check if the files were deleted, and it found nothing. What would possibly have gone wrong? How would SpinRite help in such a scenario? Thank god I had a very recent backup and just lost three days of work." And then he says: "In case you're going to talk about this on SN, I would appreciate if you do not disclose my name. Thanks a lot. Regards."

And so this is interesting because it's a little different than what we've talked about, although I have mentioned it. And the answer is, if there is a problem in the file system metadata, that is, the directory structure. Any file system has files that themselves contain the data that the file system stores. But there's always management information. Back in the day of DOS we talked about the FAT, the File Allocation Table, which showed which clusters had been allocated. And it was by scanning that that the system could quickly tell how much space was available remaining on the disk by how many clusters had not been allocated and had zero bits set in the FAT. But all file systems have some means of managing the files. And it's not the file data, so that makes it the metadata, or commonly called the "directory structure." The symptoms absolutely sound like there was a flaky metadata sector.

And so it's very much, you know, our listeners, I'm sure, are familiar with how a file system can be viewed as a tree, a hierarchy of folders which are branches that contain folder that contain folders that contain folders. And so this creates a branching structure. So it might be that a sector contains pointers to other folders in the tree, or pointers to files. And if that sector cannot be read, then the files that it points to, or the folders and everything else downstream, disappears. And so it sounds like that this drive was sort of on the edge. Sometimes it could get read, and then Windows would boot. Sometimes stuff was missing. And it was missing because the metadata that time wouldn't read.

It is a classic case of SpinRite being able to recover. Sounds like he had a backup that was only three days old. He didn't need it. But for what it's worth, he could have used SpinRite, I mean, this is like exactly. And it even sounds like it wasn't that far gone. It wasn't the typical hard crash where everything is just gone now, and nothing but SpinRite will bring it back. It sounds like it was just getting to the point where it wouldn't have been, you know, it was no longer reliable. SpinRite probably could have sucked it back to life in no time. So for what it's worth, if you find yourself in that situation, and you don't have a recent backup, and you do have a copy of SpinRite around, it'll probably fix it for you.

Leo: You don't have a lot of experience probably with Linux or Mac files. Well, you do with the Mac. Do you have a favorite file system? Is it NTSF? Is it exFAT? Is it HFS? Do you have an opinion?

Steve: I don't. I don't have a useful mature opinion. One of the things that SpinRite 7...

Leo: I'll take a childish opinion. That's okay.

Steve: One of the things that SpinRite 7 will do is have knowledge of all of the different file systems. I'm going to take SpinRite from physical data recovery to true file system recovery. That's the plan. And, you know, I forgot to mention last week when I talked about changing the BIOS in order to get SpinRite 6 to work, I think I scared people off. And I forgot to mention that anybody who has SpinRite, all SpinRite owners, will get 6.1 at no charge.

So I did want to make sure everyone understood that I consider that my obligation. Everyone says, Steve, that's very generous. But many people I know are buying SpinRite to support me and GRC and the podcast, under the assumption that 6.1 is going to come, and there's a future there. So I wanted to make sure people understood that. I'm happy to do that. I'm very excited about 6.1. And once SQRL is behind me, that's what I'm back to.

Leo: Good. We're waiting with bated breath, Mr. G., as you know. What is this little teacup you've got? How many shots of caffeine can you get in that thing?

Steve: Well, I have one of these. This was, when I was married, this was the fancy place setting.

Leo: That's the china, yeah, yeah.

Steve: And [Julia] made off with all of the rest, only because this one just sort of was a stray. And I found it years later, and I thought, well, fine, I'm going to just drink from the good china.

Leo: A little memory. A little memory. I have some - being on my third wife, I have some old wedding gifts lying around. Lisa's put most of them in the garage, though.

Steve: But I do like it because I draw from the tank, you know, I have the big silver tank.

Leo: What is that, tea? Oh, I see. So you are using coffee, but from your Contigo.

Steve: Exactly.

Leo: Yes.

Steve: Exactly. Because the whatever it was, the Nirvana Thermal Thermos is still in shipment from China.

Leo: The Temperfect.

Steve: The Temperfect. The Temp Imperfect.

Leo: Good lord. So I do, sometime I want to - and I'm sure you'll do this work. It is possible to make a UEFI bootable key, like a USB key. Because I get them all the time. And I'm really curious what would be involved in that. I guess you'd have two partitions, one that would have a certificate of some kind in there to - I'm not sure what would be involved for a secure boot.

Steve: I have not gone there yet. But...

Leo: Because I'd like to be able to make, you know, some secure boot compatible keys. I don't want to have to turn off secure boot.

Steve: You know, I'm trying to think of the name. I think it's called REUFI, R-E-U-F-I. I have something on my Mac that allows me to dual boot, and it's the way I was able to verify that SpinRite could run on the Mac only if it fixed the keyboard. So I actually had - I have a version of SpinRite that does run on the Mac because Mac uses a USB-like keyboard. But I used, I think it was called REUFI.

Leo: I'm sure it's not. That would be "rufi," and probably not the best acronym ever.

Steve: Could it be REDOS, R-E-D-O-S? Check that. Is your machine back up?

Leo: I don't know. Yeah, got my machine working. By the way, unplugging a variety of USB devices, sometimes a device hanging off your USB can hang up something. I don't see REUFI, rufi.

Steve: I don't have my [crosstalk].

Leo: Somebody called the radio show with a great new boot manager, which I of course have immediately forgotten. Was that from the caller, REUFI? Maybe I need a dash. What was the name of the boot manager he called about? But anyway, it sounded like a much better boot manager, and it did handle UEFI. That secure boot's a challenge.

Steve: Yes, it is. Yup.

Leo: Let me see if I can find it here. Rufus.

Steve: Rufus, that's it.

Leo: Rufus, hello, Rufus.

Steve: Yes. It's very capable. It may do everything you - I haven't looked at it for a while, but it just...

Leo: Create bootable drives the easy way. Oh, this looks good.

Steve: Yeah, it made short order of my need to dual boot my Mac, so that I was able to have DOS boot and run SpinRite in one partition, and then the Mac in the other.

Leo: And they do, on the front page, they have the example of making a Tails bootable USB key, which is the primary use I'd have for this. Okay. Don't know if it handles UEFI, but Rufus. Rufus. Let us take - oh, and there's rEFIt, refit.sourceforge.net.

Steve: Ooh, rEFIt, that was the one.

Leo: REFIt.

Steve: Yes, that's the one. Although that may be different. Oh, and that's an EFI boot menu, it says.

Leo: EFI boot menu. There you go. I think this is the one. This is the one, yeah, rEFIt kind of idea.

Steve: Yeah. Much better than Rufus.

Leo: Let's take a break. Much better than Rufus. No, Rufus is fine, too. But better than rufi, rEFIt. Hey, there he is, Steverino. Hey ho, Steverino.

Steve: So my last security topic for the week is another side channel attack which is a little spooky just because these guys understand the fun of some drama, and they set up a presentation which is just wonderful. The show notes have the two pictures showing their setup. The first picture is on one side of a wall, where they've got - they just sort of jigged this thing together. So they've got a laboratory two-channel power supply that is running an SDR, a software-defined radio, providing power for that. And that's hooked to sort of an antenna device. And the output of the software-defined radio then goes to a laptop, which is doing frequency spectral analysis of the information it's receiving. On the other side of the wall sits a laptop that looks like a Lenovo with some version of Windows, just sort of sitting there, doing nothing. What these guys have achieved is, in a matter of minutes, extracting the Diffie-Hellman elliptic curve keys from that laptop.

Leo: No. No.

Steve: Yeah.

Leo: Say it ain't so.

Steve: Yeah. So they say in their write-up: "We show that the secret decryption keys can be extracted from PCs running the ECDH" - that's the Elliptic Curve Diffie-Hellman - "encryption algorithm using the electromagnetic emanations generated during the decryption process. By measuring the target's electromagnetic emanations, the attack extracts the secret decryption key within seconds from a target located in an adjacent room through a wall." Then they say, "ECDH (Elliptic Curve Diffie-Hellman) is a standard public key encryption algorithm used in OpenPGP." And they give the RFC and the NIST specs. They write: "We attacked the ECDH implementation of GnuPG's libgcrypt 1.6.3, the latest version at the time the paper was written. The attack asks for decryption of a single carefully chosen ciphertext, iterated a few dozen times."

I should mention that the few dozen iterations is more for the sake of improving the signal-to-noise ratio than, like, they needed to actually see it multiple, multiple times. And so my point is that that's the kind of thing that an evolution of this attack could reduce or minimize, if they really, I mean, probably if they just spent a lot more time. Instead, it was just easier for them to iterate the same thing in order to get multiple samples of the machine doing the decryption, and then sort of sum them together or average them together in order to separate more of the signal from the noise. And then they use time frequency signal analysis techniques to extract from the electromagnetic leakage emitted by the target laptop during the execution of the ECDH decryption.

So, okay. So what this comes down to, and we've talked about this in other contexts before, is that the particular implementation, this is not a problem with the laptop. They're all going to leak unless they're, like, TEMPEST hardened. Those are not ones that we have. They're going to leak. And it's not a problem with the math of Elliptic Curve Diffie-Hellman, which is good. It's a problem with the implementation. And it's one of the reasons that, for example, for SQRL, I chose Dan Bernstein's library because Dan is a stickler. If he's anything, he's a stickler for detail. And the crypto that I chose for SQRL has no execution paths based on secret material. That's the key.

The mistake is it's easier to write crypto where the key affects the instruction sequence that the computer follows. If you do that, this is what happens because the instruction sequence will result in subtle variations of emitted radiation, or power consumption, or even maybe some inaudible sound. And that's the problem is that something that is supposed to be secret has that, and that's why we call it a side channel. It's not an attack on the encryption channel. It's an attack on a side effect of decrypting the information, a side channel attack.

And so, in their own little Q&A, I just grabbed two out of more, Q1 and Q4. The first question was, "How vulnerable are GnuPG and other applications that use libgcrypt now?" So they say: "We've disclosed our attack to the GnuPG developers and worked with the developers to implement countermeasures. GnuPG's libgcrypt 1.6.5, containing these countermeasures and resistant to the key-extraction attack described here, was released concurrently with the public posting of these results," although that's recent. So the vulnerable one was 1.6.3; 1.6.5 is current. So the fix exists. The patches exist. If this is a concern, you'll want to update GnuPG as soon as they publish code that has this fixed.

And then they wrote specifically: "Libgcrypt 1.6.5 completely changed their implementation of the elliptic point curve multiplication, using the double-and-always-add algorithm. This is slower than the previous implementation, but more resistant to side channel attacks since the sequence of high-level arithmetic operations does not depend on the secret key." And that's what you need.

This is going to end up becoming clearly standard practice moving forward, although it's obviously not yet going to be standard practice for one of the ways we evaluate the security of ciphers is not just bit banging and how wide is the block size and how many iterations does it go through, and if we reduce the iteration strength, is there a lower iteration, a weakness that we can find, blah blah blah blah. It's going to end up as important is going to be is this thing, algorithm, is the algorithm used to implement the math absolutely secret key neutral? Meaning that no secret bits are involved in changing what the processor does. And as an example, it may slow it down. Everyone's always wanting this stuff to be as fast as possible. But that comes at a cost of some leakage.

Now, there may certainly be scenarios where it's not a problem. If you're using some appliance in a sub at the bottom of the ocean, and you're around people you trust, then there's nowhere for it to leak to. But there are certainly instances where you'd like to know that you've got a side channel attack-proof solution. And then, oh, I love this one. Q4 was, "What if I can't get physically close enough to the target computer?"

So they respond: "For RSA and ElGamal" - those are two previous side channel attacks that they had demonstrated - "though not yet for ECDH, similar attacks have been demonstrated from large distances." And get this. One was what they called - and we talked about this at the time - "laptop chassis potential, measured from the far end of virtually any shielded cable connected to the laptop - such as an Ethernet, a USB, an HDMI, or a VGA - can be used for key extraction as we demonstrated in a paper presented at CHES '14."

So two years ago, that's when we talked about this. If you had any shielded wire hooked to a machine, the shield carried this noise. There was just no way for it not to. And so no matter how long the cable was, hooking to the shield at the other end, they were able to extract the key from RSA and ElGamal, not yet from ECDH.

And then they also presented at Crypto '14: "Using acoustic emanations measured via a microphone," they write, "can also be used to extract keys from a range of several meters, as we showed in a paper presented at Crypto '14." So it is - I think it's clear that, from what we're seeing, this is no longer just theoretical. We've talked about similar attacks where software running in the same virtual machine, like sharing a host with another virtual machine can actually feel what the other virtual machine is doing and extract keys just by being co-resident in the same hardware, even if there's no actual contact between the two. I think moving forward we're going to end up with this whole idea of this being secret key neutral as another important requirement and attribute of encryption technology. And that's our podcast.

Leo: Nice. Your timing is impeccable, Mr. G. I'm sorry about your site. Is it still down?

Steve: Oh, yeah. Oh, yeah. I'll be dealing with it for the next we don't know how long.

Leo: Well, I mean, I don't know to what extent you want to pursue this. We could talk to - I'll send a note to John Graham-Cumming over at, as I mentioned, at CloudFlare. You know, SquareSpace, you could set up a temporary SquareSpace site where you could do eCommerce, if you wanted to continue to sell SpinRite.

Steve: Understand that all of this takes huge amounts of cycles. I mean, it's just me.

Leo: Yeah, I know. You know what?

Steve: And, I mean...

Leo: The thing is, they'll move on, I presume. And they usually do. They'll move on. The question is if they're aiming at you intentionally, or if it's just you happen to be a convenient target, or what's going on?

Steve: And that's, see, it's so tempting to try to gain some...

Leo: You can't.

Steve: ...insight. And when the attacks lasted an hour, an hour and a half, 30 minutes, and they came and went, I thought, okay, you know, it's a Saturday afternoon. Somebody's testing an attack tool.

Leo: Or a kid was home from school, and he - yeah.

Steve: Yeah. And then on I guess it was the next day, on the 14th, yeah, on Valentine's Day, you know, 13Gb of attack. So, and the other attacks were large also. I just wasn't seeing them. So anyway, I'll be working with Level 3, and I'll let my Twitter followers know what I know. And I hope that I am allowed to be back on the Internet because I like to be on the Internet.

Leo: Yeah. It's a new form of vandalism, kiddies. And I do mean kiddies. I can't really send you to his site, but normally he's at GRC.com. That's where you'll find SpinRite and all that great stuff when it comes back. It'll be back in a couple of days, I'm sure. You can also follow Steve, and you should now, on Twitter, @SGgrc, because he'll let you know when the site is back and whatever he learns. And of course he's always tweeting and responding to questions. Might have questions next week, if the site comes back, GRC.com/feedback. If not, well, I'm sure Steve has plenty of other things he could talk about. And let's see, what else? We do have the podcast. You can at least get that. And you know what, if you email us the show notes, we'll make sure that that's on the show note page on TWiT.tv/sn.

Steve: I'll do that. Elaine has tweeted and said that she would get the audio from you rather than me recompressing it for her as I normally do.

Leo: Yeah, okay.

Steve: So she's aware of what's going on, too.

Leo: It'll go up in a little bit. TWiT.tv/sn. Of course you can subscribe everywhere. Thing is, you can't kill this podcast because it's multi-homed. You'd have to have more than 13GB, gigabits, gigabytes. You'd have to have more than that because it's everywhere.

Steve: Although there is no podcast without SpinRite, so we do need SpinRite.

Leo: Yeah, yeah. We'll have to figure something out on that one. GRC.com will be back, I promise you, one way or t'other. I'm sure you're going to be hearing from some fans who want to help out. Thanks, Steve. We'll see you next week on Security Now!.

Steve: Thanks, Leo.

Leo: Bye-bye.