SECURITY NOW!

Transcript of Episode #509

LOGJAM: Imperfect Forward Secrecy

Description: After covering the week's most significant security news, Leo and I closely examine the week's most significant news, a major new vulnerability in the Internet's TLS protocol known as "Logjam."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-509.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-509-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. I am here. Today we're going to talk about Logjam. Last week it was Venom, this week it's Logjam. There's always a new insecurity. This one may be pretty significant for SSL interactions, both for you the user and for the web server. What is Logjam, what does it mean, and how to fix it, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 509, recorded May 26, 2015: TLS Logjam.

It's time for Security Now!, the show we talk about security in - in which we talk about security - with this guy right here, Steve Gibson, the Explainer in Chief. Hi, Steven.

Steve Gibson: Hey, Leo. This was supposed to be a Q&A. And I was ready to go with the Q&A until everybody in the security press decided that they had to talk about Logjam. Lots of my Twitter followers said, oh, my goodness, another problem with secure connections on the Internet. And this thing has so many aspects and nuances to it, like maybe now we understand what some of the slides that Snowden leaked that seemed puzzling and, like, sort of hard to understand, maybe now we know what they were saying.

Leo: Oh, my god.

Steve: What capability that the NSA has that we just couldn't really rationalize before. This is the result of a global group of serious security researchers who have been working for some time to figure out and deal with a very subtle flaw which was discovered in the TLS protocol, you know, SSL, the protocol that we all use. The upshot is that there is a means at the time that they discovered this, and remediation is already underway, but there was a means for a man-in-the-middle attack to intercept about 8.4% percent of Alexa's top million, I think it is, websites. So a serious breach in the privacy guarantee that TLS gives us.

But anyway, this thing, it's really interesting, and I thought it would make just such a great podcast. So I said, okay, when these things happen, we rearrange the schedule. So we'll do a Q&A next week, and I'm sure there'll be lots of people wondering about the Passive Keyless Entry and Start topic that we covered last week because it generated a great deal of interest. So I'm sure my mailbag will be full of that, and we'll get to that next week.

This week I want to talk about Let's Encrypt's Terms of Service, the draft of which was posted. We have another worrisome consumer router flaw. The debate about government back - what?

Leo: Just about that.

Steve: Yeah.

Leo: It's very, at this point, because I talk about this on The Tech Guy, the radio show, which is the normal people show.

Steve: Right.

Leo: And it's...

Steve: The which end do I plug in.

Leo: These are people who buy their router and keep using it for years, like six, seven, eight, nine, 10 years.

Steve: Yeah.

Leo: And all I can say is just you have to buy a new router. I can't explain why. Just get a new router. And you'll have to probably get one in a few years again because they just - it's sad.

Steve: Yup. The older they are, the worse. So we have a new router flaw. The government backdoor debate is still continuing. A whole bunch of security people wrote a letter to the Office of the President, and James Comey replied. I want to talk about that. A little bit more on Chris Roberts, the wacky airline maybe hacker. A comment about blocking scripts on Chrome, and some miscellaneous stuff, and then some really, I think our listeners are going to find it very fascinating, explanation of exactly what Logjam, this latest vulnerability is and how it works. And on the first page of the show notes I have a diagram of the TLS protocol showing in red the things that are modified by the man in the middle. But we'll come back to that here toward the second half of the show.

So Let's Encrypt is this really intriguing effort. For some reason I hang the EFF on this, although they call themselves ISRG, the Internet Security Research Group. And so the EFF is involved. I'm not really clear on where this came from or what the group was. I probably covered it when we first talked about Let's Encrypt a few months ago. And at the beginning of the year, when this first surfaced, we were saying around summertime this year. And we appear to be still on track for that.

What became available on the 21st was the draft of what they call their subscriber agreement, which is the agreement that somebody obtaining certificates from this service would implicitly and explicitly agree to, however they arranged that. And so remember that the whole idea of Let's Encrypt is we're going to change the model of certificate issuance, administration, and management for that class of certificates that require moderately weaker authentication.

For example, it won't be issuing extended validation certs, where you really are making a powerful assertion about the identity of the company. These are more like the DV, the Domain Validation certs, because the way this works is you drop an agent in your server, and the whole certificate issuance system is automated so that the agent in the server, in a web server, contacts the Let's Encrypt facility and says, "Hi there, I want a cert. This is my domain." That facility, essentially the Let's Encrypt CA, sends something back for that agent to post on the website of the domain it's claiming that it has control over. Then it says, okay, I posted it.

Then that external facility checks to see if the thing they sent back has appeared on the site. If it has, that says that this agent that's previously unknown to the CA has control over the domain. So the Let's Encrypt CA issuing facility says, okay, you're obviously the man for this domain. Here's a cert. And it electronically sends the cert. This agent knows how its own server works and installs the cert, configures it, and you're up for full SSL, TLS, HTTPS, private domain-authenticated security with, like, no cost. This is all free. And revocation is handled through the facility. As certs begin to approach expiration, they notify Let's Encrypt and say, "Hey, you know, I've got only a month to go here, let's update me." And so that all happens. So basically, for the class of certificates where all you need to assert is your ownership of a domain, rather than the actual corporate identity that owns the domain, this system looks like it's got a great chance to go.

So I read the six-page legal agreement carefully, just because I wanted to see what it looked like, you know, were there any gotchas, was there anything strange. And the answer is no. It's short. It's legible, which is nice for anything that came from attorneys, and I'm sure there were attorneys involved. And so, you know, it's exactly what you'd expect. There's a section of the warrantees that the user of the certificates is implicitly making under the agreement, and things like you warrant to the ISRG, which is the issuing authority, and the public at large, that you are the legitimate registrant of the Internet domain name that is or is going to be the subject of your certificate, and/or that you are the duly authorized agent of such registrant.

You warrant, oh, that you have not participated in the seizure of a domain name that had ongoing lawful uses. I thought that was interesting. So if a domain name got seized, then the Let's Encrypt facility would allow the seizer of a domain name to immediately obtain a certificate because all you have to have is the domain name.

Leo: Right.

Steve: And so that little clause says that, if someone does that, and Let's Encrypt finds out about it, they can immediately cancel the certificate under this agreement.

Leo: But, see, that's the question, because isn't certificate revocation broken?

Steve: Not under this facility. And that's one of the nice things is that they're providing revocation for the certs and are able to proactively...

Leo: Does the browser pay attention?

Steve: ...proactively revoke. I've not yet looked at the protocol. It's called ACME, A-C-M-E is the protocol, and it's on my list of things I want to get to because it's going to need a podcast to explain exactly how this works. But I remember, in our initial coverage of it, being very impressed with the fact that they'd figured this out.

They say that you're also warranting that your domain name is accurate, reliable, complete, not misleading; that the information you've provided is the same; and basically that. And, oh, there is one sort of an interesting caveat. They said, under "Use of Your Certificate," they said: "The purpose of your certificate is to encrypt Internet communications. You are responsible for all legal and other consequences associated with the use of your certificate. You agree that you will not use your certificate for failsafe performance, such as the operation of utilities or power facilities, air traffic control or navigation..."

Leo: Good.

Steve: "...weapons systems..."

Leo: No, shouldn't, no, bad idea.

Steve: "...or any other system the failure of which would reasonably be expected to lead to injury or property damage."

Leo: In other words, this isn't going to be a 100% reliable solution?

Steve: This is, well, this, you know, you and I have worked with attorneys, Leo.

Leo: Right.

Steve: This is the kind of thing that they put in there.

Leo: Okay. It's called CYA.

Steve: Exactly. So that the Let's Encrypt guys, they're a nonprofit organization. They're not making tons of money. They're not doing any human oversight at all. This is fully automated. So I think it makes sense for them to put in here, you know, don't use this for these things because, if you do, you're in breach of the agreement, and then we're not liable, you're liable for anything that happens if you, like, misuse it in a way that could cause injury or property damage. So it's, you know...

Leo: I kind of hope, though, that there's not an unintended consequence to this. We've talked before about the issues with cert revocation.

Steve: Yeah.

Leo: I'm very curious how they plan to support that. The unintended consequence could be that you've got - that it breaks the system, that you have untrusted certs floating around. And it breaks the - sometimes you push too hard to make everybody use certs, and there's an unintended consequence. I fear that. We, by the way, thank you, DigiCert, the new site launches June 1st, all SSL, all the way, with a real cert, a man-sized cert. None of these cheap certs.

Steve: Nice. Green. Your bar will be fully...

Leo: No, we didn't - no, no, no, no. Wait a minute. Let's not go crazy. We didn't do extended.

Steve: Oh, that's right, because you have...

Leo: We have wildcard.

Steve: I think you needed a wildcard, yes.

Leo: We need wildcard. You can do extended, but it's limited to a certain number of domains, and then you have to buy another one, and it's very expensive. So we didn't feel like extended was that - we're not doing ecommerce here. So we just did a wildcard, yeah.

Steve: Right, I agree. Nice. Anyway, so for what it's worth, we're moving forward with Let's Encrypt, and it looks like it's going to be a good thing. The six-page agreement has no nasty surprises in it, nothing that I can see as, I mean, as a showstopper at all. I think even Stallman would think this was okay.

Leo: That's saying something. I don't think so. I think you're crazy.

Steve: Maybe not. Is it all open source? Is it owned by the people?

Leo: Yeah, no.

Steve: So, okay. This new router problem. If your router has a USB port on it, you might have a problem.

Leo: Yeah. This is the one I talked about on the radio show this weekend, yeah.

Steve: Yeah. This is the NetUSB bug. NetUSB is the protocol that a company called KCodes developed which is USB over IP - meaning that, if you have a router with a USB port, you can plug, like, a big USB drive into your router and turn the router then into a file server, essentially, because clients running in Windows and Mac PCs are then able to use their Internet connection to the router to access the USB device that's plugged in. The bad news is there's a problem with the kernel driver. This affects Linux-based embedded systems, which most of these routers now use in their core. OEMs who have licensed this driver from KCodes include D-Link, Netgear, TP-Link, ZyXEL, and TRENDnet. And one of the components, apparently, lists another 26 companies. So rather than looking at the company names, the right way to say this is, if you have a USB port on your router, then you need to take notice.

Now, the good news is it's almost exclusively a local problem, meaning a LAN side, such that something would need to be already in your network that abused this protocol in order to create a buffer overrun in the kernel that would let it run some code in your router. So people are worried, and this is worth taking action. I think that means, as you said, Leo, for people who aren't firmware updaters, maybe it'd be a good time to just buy another router, and get one without a USB plug or connection. And note, you don't need to have anything plugged into it for this to be a problem, either. Just the presence of the socket on the back of the router causes the vulnerability because it means that that code is in the kernel.

Where the problem exists is some routers - and I'm closing my eyes as I say this, it's hard to believe - expose this protocol to the Internet. It's on the WAN side, as well. So if it's there, this protocol runs over port 20005. And Michael Horowitz, who does the Defensive Computing blog at Computerworld, he tweeted, he said "Test your router for public (WAN side) exposure to the NetUSB flaw with..." and then a link to the GRC port probe because you can easily check your router. It's not very well known, but the GRC port probe that I wrote years ago accepts a port number in the URL. So if you went, Leo, for example, right now, if you went www.GRC.com/x/portprobe=20005 and hit Enter, GRC server will check your connection to see whether that port is open and accepting any traffic. And so anyone who's interested can - and I've got the link in the show notes, again, GRC.com/x/portprobe=20005, and just like that you can check to see whether your port is opened or not.

Leo: NetUSB-wise. And we are stealth, baby. Of course we are. We've got an Astaro

Security Gateway protecting us from you, outside world.

Steve: That's what you want.

Leo: Yeah.

Steve: So anyway, just as a heads-up, there is a problem with the USB connection. It would be good to update your firmware, maybe wait a few weeks, or check to see whether your manufacturer has addressed this. We know the problem is that a lot of manufacturers sell the routers and then abandon them, which I think is probably why, Leo, your advice was just go get a new one.

Leo: Buy a new one. Or even if they fix it, as in the case of, well, I don't want to name names, but a number of these they don't fix it properly.

Steve: Correct.

Leo: Right.

Steve: Yes. They say, oh, yeah. And then it's like they still leave the backdoor open, but obscure it a little bit more.

Leo: Right, right.

Steve: Yeah, they use some glue. Okay. So...

Leo: If I were talking to sophisticated users, like our fine audience here today, I'd say get a router that you can put your own firmware on, Tomato or DDWRT or something like that.

Steve: Exactly.

Leo: Because then you do the work. You do the updating.

Steve: Right. And I don't know whether that protocol is supported by those firmwares. I mean, it's possible that KCode is licensed to those guys or, you know, whether they support it at all.

Leo: Highly unlikely. Yeah, I would be so surprised.

Steve: Yeah, because the whole KCode thing is going to be a commercial, licensed solution. And the open source guys aren't going to go that way.

Leo: Right.

Steve: So, okay. Government encryption backdoor debate. What's neat is that nearly 150 tech companies and crypto experts signed a letter that was sent to the Office of the President. I mean, Google, Apple, Cisco, Microsoft, Twitter, Facebook on the corporate side, and many more. Phil Zimmermann, of course, of PGP fame, who has sort of been through the whole crypto problem in his past. Whitfield Diffie, as in Diffie-Hellman, the DH key agreement protocol, you know, Diffie invented public key crypto. Ron Rivest, who's the R of RSA. Our friend Bruce Schneier, and Matt Blaze. Even Richard Clark, the longtime counterterrorism guy in the White House. So, I mean, this is a Who's Who of Silicon Valley established companies. And even Cisco, you know, old school. And also everybody in crypto. And I just named a few, I mean, as I scroll down the list, there's like, okay, everybody's here. And basically they said, look, this is a problem without a solution. There is no way to give the government a backdoor that does not fundamentally break the security of what we're trying to offer. It cannot be done.

And then a day later the FBI director, James Comey, claims that the world's most knowledgeable cybersecurity experts are "not fair-minded," was the term he used, about encryption backdoors. And he produced a letter that said: "A group of tech companies and some prominent folks" - okay, yeah, "prominent folks," the people who invented cryptography - "wrote a letter to the President yesterday that I frankly found depressing. Because their letter contains no acknowledgment that there are societal costs to universal encryption.

"Look, I recognize the challenges facing our tech companies. Competitive challenges, regulatory challenges overseas, all kinds of challenges. I recognize the benefits of encryption, but I think fair-minded people also have to recognize the costs associated with that. And I read this letter and I think, 'Either these folks don't see what I see or they're not fair-minded.' And either one of those things is depressing to me.

"So I've just got to continue to have the conversation. We've got to have a conversation long before the logic of strong encryption takes us to that place. And smart people, reasonable people will disagree mightily. Technical people will say it's too hard. My reaction to that is: Really? Too hard? Too hard for the people we have in this country to figure something out? I'm not that pessimistic. I think we ought to have a conversation." And the story that I ran across this in was in Techdirt, so of course they've got a bit of 'tude.

Leo: Mike Masnick always has fun with this stuff, yeah. I love him, yeah.

Steve: So he says, he editorializes, saying: "Hey, Comey! No one is saying it's 'too hard.' They're saying it's" - and he has in all caps - "IMPOSSIBLE to do this without weakening everyone's security. Impossible. It's not a 'hard' problem, it's an impossible problem. Because if you weaken security to let the FBI in, by definition you are weakening security to let others in, as well. That's the point that was being made."

Leo: Good. I like it.

Steve: Yeah. To me, this is fascinating to see this. And I've got to say I have an explicit note in the show notes, second to the end of this, after we talk about Logjam and what security experts are now beginning to think the NSA may have. And I just - I'm struck by sort of how sad it is that we're now really taking an adversarial posture with respect to our own government's law enforcement. I mean, it's openly adversarial. It's, like, what is the NSA doing? What can they do to crack the privacy that we feel we have a right to? You know, and this is our government. And in the U.S., we're its citizens. And this has really become, I mean, you know, frankly adversarial.

Leo: Yeah. Sad.

Steve: Yeah. So I don't know, again, this is - I can't predict what's going to happen. If you look through this letter, I mean, it's beautifully written, signed by everybody. And, I mean, I really understand the "going dark" problem that law enforcement faces. I mean, from their perspective, I can put myself in their place. But this is binary.

Leo: Well, it would also be easier for law enforcement to do their jobs if they could walk into each and every home in the United States of America and look around, just to make sure you weren't doing anything illegal.

Steve: Yes.

Leo: That would make their job easier. That is not the point of the Constitution.

Steve: Right.

Leo: Not to make your job easier.

Steve: You have to have a reasonable expectation in order to get a warrant and then do a search. You just can't bulk collect. And in fact, at this podcast, here we are on Tuesday, and the end of the month is approaching when the Patriot Act expires. And of course...

Leo: That will be interesting to watch, when they have three hours to renew it.

Steve: Yes, because Congress tripped all over their feet. Rand Paul, of course, held forth for 10.5 hours and essentially threw what was already a rather precarious schedule into the hopper, essentially, with the three-day weekend. And as you said, Leo, I think they get back in, and they have, like, the late afternoon of the last day of the month. And our government has said, even the threat of the bulk surveillance - this is the whole Section 215 mass collection of phone records which is about to expire - our government has said they have had to start winding it down in anticipation of nonrenewal.

Leo: It was also deemed illegal by the courts. So they would be doing that anyway, I think.

Steve: Right. Well, unconstitutional.

Leo: Unconstitutional. So they would be doing that anyway, even pending an appeal. They have to shut it down.

Steve: Do they? Because what I remembered was that...

Leo: Well, don't they?

Steve: The appellate court ruled it unconstitutional, but they didn't go...

Leo: Oh, they did, they sent it, you know what, they sent it back to the previous court.

Steve: Right.

Leo: So, but I still think they're making plans for the wind-down, probably as much precipitated by that.

Steve: Well, and, of course, the weakness of the program is that no one has been able to demonstrate that that bulk surveillance produced any results. Yes, as you said, of course they want it. But they haven't been able to even retrospectively point at somewhere where they were able to use it in order to substantially help with the case. And, frankly, we have a history of overreacting after horrible things happen. And the events of 9/11 may have caused this to get a little carried away, in this regard and others.

Leo: Yeah. I think the will in Congress at this point, with everybody except perhaps Mitch McConnell, is, yeah, I think it's time to reassess and maybe retrench a little bit on the Patriot Act.

Steve: The other thing, too, is you hear the people screaming about that the instant it expires, we're now vulnerable to the terrorists.

Leo: No.

Steve: And it's like, uh, no.

Leo: That's people who don't understand what it is. The mass collection of phone data is prophylactic. The point of it is, if you do anything wrong, well, they can now look back in time, and they can gather stuff. I think there is some analytics going on with it. But I think all those analytics are, you know, what have they produced?

Steve: Yes. And they're unable to demonstrate, even in closed-door committees, they have not produced evidence that this was useful. So it's like, well, you had a chance. And I guess, it's interesting, I could tolerate that, more than I could tolerate having my connections broken into. It's like, okay. I mean, if there was a halfway point, metadata doesn't seem so bad to me. But it needs to be constitutional, or we need to create laws that say this is what's going to be done explicitly.

Okay. So Chris Roberts, our wacky airline hacker. I've seen, since we talked about it, so much on both sides. I received a very nice tweet from a pilot who knows his, I think he said he flies an A360 out of Long Beach, which is not far from me. And he said, "Steve, come up. I'll show you the manuals. I'll show you that..."

Leo: You should.

Steve: "...the IFE, the inflight entertainment system, is not connected to the avionics system." And so, at the same time, there have been other people claiming to absolutely know the reverse. So maybe some planes are isolated and others are not? I don't know. I'm not going to invest any time untangling this. It's sort of just been an interesting drama. I sort of wanted to thank everyone and acknowledge all of the feedback. And it's like, on both sides, people saying "Chris Roberts is the real deal, he's done this for years and blah blah blah," and other people saying it's bull, it's just, you know, in fact, NYC - I have a link, "Anatomy of a Story: Why the Airliner Hacking Claim Is Bull," and that's NYCAviation.com, that just works to debunk this.

And at this point it's like, I have no idea. I know that it absolutely - and we know, the listeners of this podcast, the entertainment system which passenger compartment has access to must not be connected. But remember that, in the car hacking, you would think that the inside of the car and its entertainment system would not be connected. But we know it is. You can put a CD in that causes a buffer overflow and take over the car. So these lessons are difficult to adhere to because it's cheaper if you just make it all one system and just say, okay, we'll put some filters in here to prevent anything bad from happening. And it just doesn't work. It's like the dinosaurs on "Jurassic Park" that had the lysine deficiency, and they could not reproduce without lysine. Yet we found some eggs.

Leo: That just might be the nerdiest reference I've ever heard. But nice, nice. Well done.

Steve: So I wanted to bring to people's attention an add-on for Chrome called ScriptBlock. A lot of people have said, yeah, you know, you keep talking about NoScript and how good that is. There was something called NotScript which was discontinued, like a couple years ago. But I learned of ScriptBlock for Chrome. I'm not using it because I'm still not on Chrome. I'm staying with Firefox for now. But it's well thought of. A number of people have said they have been using it, and it does what they want. So I'm sure that

I've had success with avoiding infiltration of my system because I've got scripts blocked by default. You know, life works for me that way. It's pleasant and safer. Leo, you've got the advantage of being on a Mac. And, you know, what are the percentage of stories we talk about that are Windows problems versus Mac? It's like 99.9999 to one.

Leo: Right. So I don't care.

Steve: Yeah. Well, yeah. I mean, you don't have the problem.

Leo: Run your scripts.

Steve: You don't have the problem.

Leo: Right.

Steve: And it may well be that Chrome itself provides enough protection because we know that Google has put a lot of time into the security of Chrome. But one of the stories that did not make it onto my list that I'm going to cover next week talked about the performance improvement from doing some management of all the assets which web pages pull. And in this case it was tracking. The story was about Firefox's tracking blocker. I don't think that's the name of it. Something like that. We've talked about it once before. It's not yet on the UI, but you can get to it in the about:config.

Anyway, the point is that 80% of what is loaded from third-party sites is scripts. Not even ads. Not anything visual. Not something you see. Little code snippets that have the sole purpose of tracking and monitoring what you do, or providing analytics of one sort or another. But 80% of what, you know, on like major news sites, for example, that stuff you're loading is scripts that are running from third-party websites. So I'm happy that they're not running on my browser.

Some quick follow-ups on last week's Passive Keyless Entry and Start. Many people asked, and I don't see where I have it, I had it right around here somewhere, that cute little RF blocking, I called it a "baguette," that I found on Amazon. It's called the Pu Leather Cell Phone Anti-Tracking Anti-Spying GPS RFID Signal Blocker Pouch Case Bag. Anyway, P-U, Pu, leather cell phone...

Leo: There's a million of these because...

Steve: Yeah.

Leo: And it all started when the United States government started putting RFID tags in passports.

Steve: In passports.

Leo: And so there's a million different ways. And I think anything that would hold a passport probably would hold your key fob. Just an update, I did take the battery out of my key fob, and it was totally...

Steve: Yep, I wanted to ask you.

Leo: It was totally impractical.

Steve: Uh-huh.

Leo: So, you know, it has a key. So you can pull out a physical key, and that'll get you in the door. But I could not find an ignition key anywhere. And...

Steve: So no way to start the car.

Leo: I didn't think so. Well, I didn't spend a long time because I really wanted to go home. But I spent a good 15 minutes tapping the key to different surfaces because I thought there was an RFID maybe point that would validate it. Because what you need to do is, before you can start the car, validate that you have this fob.

Steve: Right.

Leo: And the key, I couldn't - maybe there was a hidden key port. I don't know. I know on some cars there is, for an actual physical ignition key. But I couldn't. So I just said, oh, screw it, and I put the battery back in, and it's been in ever since. Now, I want to clarify this because I think I might have said it wrong on the radio show. I thought you buy this $17 eBay device, and you stand next to the fob.

Steve: No. It's the car.

Leo: It's the other way around. You stand next to the car that you want to rob.

Steve: Right. It's the car's transmission that is the limited distance.

Leo: Got it.

Steve: So that's what you need to amplify to reach the key in the house.

Leo: But you don't need the key at all.

Steve: No.

Leo: To be anywhere near the key.

Steve: Right.

Leo: That makes it more scary.

Steve: And, well, and part of the reason is that I confused you a little bit because I did talk about both scenarios. You could have one, which seems to be what's being done in practice, where someone just at the car is able to amplify the car's signal to allow them to open the car. The alternative would be the two-person mode, where you have an agent that you're working with who has a briefcase that, like, walks over...

Leo: Yeah, but that's crazy. If the other way works, why, I mean...

Steve: Right.

Leo: So that's why somebody could go down the street and just open doors. All they have is that transmitter in their back pocket.

Steve: And they're reaching inside the house to ping the key, and the car hears the key respond.

Leo: So that's why, when you're home, if your car is out of the garage and not otherwise protected, you should put it in a bag, what did you call it, baguette.

Steve: A baguette. Now, okay. So there was lots of different feedback about this. There is something called Fob Guard specifically for this.

Leo: Okay, Fob Guard.

Steve: F-O-B-G-U-A-R-D dotcom, made for this purpose, Fob Guard. And I think it's a little family business. I got a tweet from the guy saying, hey, Steve, you know, so glad you've covered this. Anyway, so Fob Guard. They're a little more pricey.

Leo: It's just a baguette.

Steve: Yeah, it's a baguette.

Leo: A $30 baguette.

Steve: I think it's the same thing as you get from Amazon with the Pu leather cell phone.

Leo: Right.

Steve: Andy Ferguson tweeted, "An empty Altoids tin also works nicely."

Leo: There you go. Oh, sure.

Steve: So you just want something closed and metal all around.

Leo: And the way you test it would be walk up to your car and see if you can get in while it's in that thing.

Steve: Precisely. That's all you have to do.

Leo: That's much easier than taking out the battery.

Steve: That's all you have to do to test it is see, you know, if you put your fob in whatever the protection is and then walk up to your car, and it won't let you in, you're good to go. And Kyle Boroff tweeted, "My Ford gets broken into once a week in front of my own house."

Leo: Oh, my gosh.

Steve: "I always locked, didn't know how they got in. Now I know. Faraday bag on order."

Leo: Jiminy.

Steve: So, and some people said that just wrapping it in tinfoil works. And I heard someone said the Altoids can doesn't work. So you just have to try it and see. Now...

Leo: I want something I can take it in and out of easily.

Steve: Yeah, oh, exactly. Yeah, you don't want to be wrapping and unwrapping tinfoil all the time. But so the Altoids can is kind of cool.

Leo: Yeah.

Steve: One interesting thing, and I don't now remember the make and model, but one of the cars is very smart. If you take your second key and lock it in the glove box, then its presence is noticed by the car, and the car disables all of the long-range radio technology so that...

Leo: Oh, that's smart.

Steve: Yes, it's very smart.

Leo: But then how do you get in your car?

Steve: No, well, and then you have to hold it right up at the handle or, I'm sorry, or press the button. You then press the button in order to manually unlock and lock.

Leo: Oh, I'll have to try that.

Steve: Yeah, you might...

Leo: I've always wanted to do that, but I thought, well, then if I leave it in the car, somebody just gets in the car and drives off.

Steve: And then there are some other cars, through their UI, that allow you to manually disable the whole feature of Passive Keyless Entry and Start. That is, you can say, "I don't want the at-distance features. I want to manually press a button to lock and unlock my car." And if you do that, it shuts down the car's pinging, and again you're safe.

Leo: Yeah. Somebody emailed me and said that their key fob, I think it was on a Mustang, had a rolling, like a garage door number, had a rolling number thing.

Steve: Yeah, that doesn't help because what that would do is that would prevent a replay attack. That's why garage doors do that is that the same code won't work twice in a row. It's got to be, like, it's got to be a rolling code. But this will defeat anti-replay because you're not replaying.

Leo: No.

Steve: You're just range extending. This is just a simple range extension, is all it really amounts to.

Leo: It's amazing.

Steve: So, okay. Now miscellany. This is completely random. I just wanted to chat with you a little bit about why you think 3D, like TV, hasn't ever obtained much traction.

Leo: Oh, I can give you a dozen reasons.

Steve: And I was thinking, I was reminded of the quadraphonic phenomenon in the '80s, where it's like, oh, stereo's two speakers, but quadraphonic, we want four. And it just sort of never - it limped along, but never really happened. Of course now we have home theaters with seven, you know, they're, like, all around us, and we sort of take that for granted. And so I'm just wondering if it's just that it isn't worth the hassle.

Leo: There are a number of reasons. You have to wear glasses. That's, I guess, under the "hassle" category.

Steve: Right.

Leo: The glasses cut the brightness of the screen in half.

Steve: Ah, true, true.

Leo: That's in the "quality" issue. But there is a fabulous article that Roger Ebert republished in his blog many years ago that quotes Walter Murch, the great film editor. And he talks about...

Steve: Of course, good old Walter Murch.

Leo: Are you laughing, or you agree?

Steve: I am, I've never heard of Walter Murch.

Leo: Oh, yes, you have. Walter Murch has...

Steve: I may know his work.

Leo: Oh, you know his work. One of the great editors of all time. He wrote a letter to Roger Ebert which Ebert republished in which he said the fundamental issue with 3D is a brain is a biological issue, which is that you have, normally, in normal life, in every other respect, your focus point and your convergence point match. So if I'm looking at something, I'm focusing at it.

Steve: Right.

Leo: And your brain understands that. But in 3D, your focal point is the screen, but the 3D tricks you. Your convergence point is the screen, but the 3D tricks you to focus somewhere near field.

Steve: Interesting.

Leo: In other words, a divergence between your convergence point and your focus point. Your eyes converge at the screen, but you're focusing on something nearer. And that makes your brain - that's why people get headaches. It's why it's disorienting. And that's biology. That ain't going to be rewired any time soon.

Steve: So it's its failure to, well, not to be an actual holotank. What we need is, you know, a holotank...

Leo: Yes. You need your convergence of your eye. You should focus on the thing, and your eyes could converge on the thing. In other words, that would reproduce real life. You would feel like the thing is right here. But instead your eyes are telling you, no, the thing's there, and your focal point is here, and it's very confusing.

Steve: Yeah. So when the soccer ball comes out of the screen...

Leo: It makes you want to throw up.

Steve: ...at you, yeah, because the screen is still over there where it is, and the images of the soccer ball are still actually at a distance.

Leo: They're there, right.

Steve: Even though it's pretending to be close. I think it's...

Leo: Isn't it interesting?

Steve: Very nice.

Leo: Why do you bring this up?

Steve: I'm glad I asked you. Glad I asked.

Leo: Why did you bring this up, though?

Steve: Just because I've heard you, I've been hearing you talk about 3D on other podcasts to just sort of, you know...

Leo: Yeah, Scott and I - finally Scott, I think last week, said you were right. Look, I'm all for - the whole goal of everything is to make movies, but also games, more immersive. Like you're there. You want to be there. And 3D, that was the point. But the problem is it's un-immersive if there's this biological issue, not to mention the fact that I wear spectacles, and the spectacles on the spectacles is not good.

Steve: Yeah. It's a variation of riding the Segway. It's, like, difficult...

Leo: Now, don't you knock my Segway.

Steve: It's just difficult to...

Leo: I like the Segway. There's no problem with the Segway.

Steve: Next topic.

Leo: Segue into a new topic, yes.

Steve: I found another puzzle. And unfortunately this is iOS only. But I've had so much positive feedback from people who have been following my puzzle recommendations. I'm finicky about puzzles. It's easy to create bad puzzles. It's difficult to create good ones. And so I wish that these guys, the Kieffer Brothers, had done the good ones for Android, but they've only done a bad one for Android.

Anyway, it's called Blockwick for iOS, B-L-O-C-K-W-I-C-K. And for those who want to put their toes in, it's not free, but Blockwick 101 is their free sample. So for anyone with an iPhone or an iPad, Blockwick 101. I wholeheartedly recommend it. It's a sliding block puzzle, which I enjoy. But again, it was beautifully designed, in the same way that Hook was beautifully designed. There's no timer. There's no hurry. There's no rules. It's extremely simple. And what these guys did was, unfortunately, they did Blockwick 2.

Leo: Oh, you don't like Blockwick 2.

Steve: No, because there they went off the deep end. Blockwick 2 has, like, snake blocks and sticky blocks and warp hole blocks and all this gimmickry which destroys it. What you want is very clean, simple rules, and the original Blockwick has it. Yup, and there you've got it on the screen, Leo. What's nice is that those different-shaped pieces can move freely around. And the goal is to bring the same colored blocks to touch each other so that, like, all three of those blue blocks just need to have some part of them in contact with each other. But anyway, for what it's worth, if people have been enjoying my recommendations, I found another winner. And people are now tweeting me lots of their suggestions. Those that you don't hear about didn't make it because, I mean, as I said, I'm picky. Oop, and there they are. All three are now in contact with each other. Anyway, it is, it's pleasant.

Leo: All right, I'm getting this. It looks good.

Steve: It's relaxing. I think you'll find - oh, and huge number of puzzles. They've got, I don't know, 20 levels with 20 in each one. Anyway, I'm really enjoying it.

Leo: Thank you. This looks great.

Steve: So I wish it were available.

Leo: You've played Sokoban; right? You know about Sokoban.

Steve: Oh, my god. Sokoban, yes. I have it in the refrigerator.

Leo: This reminds me of that a little bit.

Steve: Yes. Sokoban is Japanese for "warehouse worker." And in there you have a little guy you run around, and he can only push blocks. Sokoban is, like, classic. It's available on Linux and UNIX and text mode and, like, everywhere. Yeah. And so this is like that. I think, Leo, if you start playing with it, you're going to like it because it's just the right balance.

Leo: Downloading it now, Mr. Gibson. Downloading it now. If I seem a little distracted later in the show, you'll know why.

Steve: Well, we're about to get into the spinning propeller phase, so...

Leo: Oh, good. I'll play Blockwick while I listen.

Steve: It's the perfect time for you to do that.

Leo: I don't know if I'm alone, but I find it nice, doing something mindless like that while I'm listening to Steve. It somehow soothes the mind. And then I can listen and understand.

Steve: Keep the blood pressure down.

Leo: Yeah, exactly.

Steve: Speaking of which, the last thing I want to say is I got a tweet from "antd" who said, "Hi, please consider sharing more health information. I haven't been sick in two years since Vitamin D3, due to you. Used to have very bad health." And I will find a way somehow. I can't do it now because I just don't have time, and I set a high bar for myself for when I want to produce one of these things. For example, I know amazing stuff about magnesium and Vitamin C, fun stories about, like, the chromosome that's broken the codes for an enzyme, L-gulonolactone oxidase, which is broken in our liver, so we can't synthesize Vitamin C, although all the other animals in the kingdom do, and we really should be getting more than we are, and why. But anyway, the point is I just need to study up on it before I put it all together in a podcast. Somehow, at some point in the future, I'll figure out a way because I'm an enthusiast, as everybody knows. And I have so much more that I want to be able to talk about.

Leo: And when we do it, we wouldn't do it as a Security Now!. Just let me know, we'll do a special. We'll put it in a feed so people can decide whether or not they want to listen to it. But I think...

Steve: Or just talk about it during the podcast and point...

Leo: Well, no, that's what I'm saying is let's...

Steve: Oh.

Leo: Oh, no, you're agreeing with me, say that we've recorded this, but have it as a separate recording, yeah. No, we're in a agreement. That way people who want Security Now! but don't want health information don't have to hear it.

Steve: Which is completely understandable.

Leo: But those who want it can do it. No, I'm all for it. In fact, you've teased this a few times, and I'm dying to know what you think because I want - should I take a magnesium pill? What should I do? What do I need to do? Tell me what I need to do. So we'll hear that.

Steve: Okay.

Leo: Do it soon.

Steve: I will.

Leo: Our lives are in your hands.

Steve: Well, yes. The clock is ticking, as they say.

Leo: I ain't getting any younger, Gibson.

Steve: So two tweets, just two tweets to remind people this week about SpinRite, one from av440studios. I got a kick out of his twitter handle, ArtVandelay. That, of course, is the made-up name that George Costanza called himself on some Seinfeld episode.

Leo: Because he couldn't think fast enough. They said, "What's your name," and he needed a pseudonym, so Art Vandelay of Vandelay Industries, yes.

Steve: Anyway, so he tweeted, "@Sggrc We just recovered over 1TB of potentially lost family pictures and home movies thanks to SpinRite. Best $89 spent of my life." So whatever your name is, thank you.

Leo: Mr. Vandelay, if that's your real name.

Steve: And Simmo3D tweeted, "I recommended a friend purchase SpinRite to recover their drive, and it fixed their issues with bad sectors." And then he said "#youralegend." And I would say, no, SpinRite is a legend, and I'm glad for it.

Leo: No, I'm merely its creator. Security Now!. Steve Gibson, Security Now! guru. Time to look at the Logjam.

Steve: So, okay. One thing I don't know, and I haven't seen anywhere, because everyone's only talking about the results of this, and I'm curious, is what the foothold was, how this research got started. I may see if I can find out because I'm just sort of curious. And you'll see why, I think, because it's sort of amazing that this problem was found. The nature of the problem is, as I mentioned near the top of the show, is about 8.4 - that's the number I couldn't remember, .4 - of the top one million domains were vulnerable at the time this was discovered.

And I'm going to take this apart so people understand what was found and what it means because it's fascinating from a protocol and security and cryptographic standpoint. Part of this has to do with the size of the public key data which is being processed. A 512-bit prime is used for the weakened, the deliberately weakened export of ephemeral Diffie-Hellman key agreement.

And it turns out that the researchers were able to do something which had only been theoretical before now, which was they came up with a precomputation attack where the bulk of the work only needs to be done once, and then that can be leveraged any number of times to crack in very short order, in a matter of minutes, specific instances of the use of the protocol, if it has a 512-bit prime, which is smaller than anyone's using now. We're at 1024 and 2048. But for academic purposes, this smaller prime has essentially been shown not to be secure. And they also demonstrate that, as a consequence of a hack on the TLS protocol, 80% of current secure TLS servers are vulnerable.

Then they look at a 768-bit prime, which is midway between 512 and 1024. And they feel that an academic team could break that, given the attack, this precomputation technology they worked out. And further, that a nation-state actor, of course like our NSA, can break a 1024-bit prime. So that's a bit sobering because then they did a survey across the IPv4 space of existing servers, both HTTPS, but also VPNs and secure shell servers. And they found that 66% of VPN servers that are using this 1024-bit prime would be vulnerable, and a little more than a quarter, 26% of secure shell servers, also using this 1024-bit prime could be vulnerable.

So then, after seeing what they had, the way they put it, "A close reading of published NSA leaks shows that the agency's attacks on VPNs are consistent with having achieved such a break."

Okay. So now let's back up a little bit. What happened? We've talked about the way the TLS protocol, previously SSL, operates. For anyone who's interested, if you just search - actually I'm sure I've used the term SSL way too many times in these podcasts. It'd be hard to find the one.

Leo: In every single show.

Steve: I meant to look it up ahead of time. But we did one just on the way the SSL protocol functions, where we took the whole handshake apart. I won't go over the whole thing again. But suffice to say, and this comes up from time to time, your browser presents to the server in what's called the "ClientHello" exchange. It's the first...

Leo: You actually, let me just interrupt real quickly, you have a whole page on Security Now! devoted to it, and it's Episode 195.

Steve: Good. Perfect.

Leo: 195, if anybody wants to find out more. Here's the transcript, everything.

Steve: Good, 195.

Leo: Thank you, sir, 195.

Steve: Yeah. So the client initially sends a so-called "ClientHello" message to the server.

And among other things it contains the list of the cipher suites that the client understands. And the cipher suite is a conglomeration of the method that'll be used for agreeing upon the symmetric key, which will then be used to encrypt the bulk data. So the method used to arrive at the key, the so-called "key agreement," then which cipher will be used. RC4, hopefully not. AES, hopefully. And then, like, the bit length of the cipher, the key length that'll be used with the cipher, and also what message authentication code or message authentication code algorithm will be used to authenticate the messages.

And so you have sort of a mix-and-match deal where, for whatever reason, you're able to combine different combinations of these in different ways, sort of creating a big menu of the way they operate. So the client supports some set of those and sends them to the server. The server, as we've often said, looks at its preferred list and hopefully chooses the first one in its - from most preferred to least preferred that it understands, which the client also offers. So that's where they agree on the cipher suite.

Now, it turns out that, due to the 1990s-era export restrictions on cryptography, not only were the ciphers themselves weakened - remember, what was it, 40 bits was the maximum bit length that you could use for RC4 back then. It turns out that a similar weakening was imposed on the key agreement. So so-called "export qualifying" key agreement, the Diffie-Hellman key agreement, is available in export grade, which is where this 512 bits comes from. 512 bits was regarded as weak enough that it could go outside of the U.S., not 1024 or 2048. But no clients, no one's browser offers the DHE, the Diffie-Hellman Ephemeral export cipher suite, that is, none of the cipher suites offer the DHE export, the weak grade key agreement. Yet many servers do.

And that's one of the discoveries is that, it has to be for reasons of default configuration and neglect, the servers - and this is this 8.4 number, 8.4% of all the servers supporting security on the Internet, at the time this research was done - and it's only two months ago, this all occurred in March and April - supported the export-grade Diffie-Hellman key agreement, the Ephemeral Diffie-Hellman key agreement. Now, again, since the clients don't support it, it didn't seem to be a big problem since the server had to choose from its list what the client supported.

But a flaw was found in TLS. What happens is the client sends its list, which does not include this export-grade weakened Diffie-Hellman key agreement. But a man in the middle can intercept this ClientHello packet and essentially remove the client's list and put in DHE-EXPORT cipher suites, basically change the list so that the server believes that the client is supporting DHE_EXPORT grade. And if that's the case, the server receives that and sort of shrugs and says, uh, okay, if that's the best you can do, that's what I'll generate. So the server generates the keying material for this weakened 512-bit, essentially public key crypto and sends it back to the client.

Now, the client - and this is a problem in the TLS protocol is that that packet of parameters for the Diffie-Hellman key agreement does not specify the cipher suite. So the man in the middle has tweaked the cipher suite on the way to the client, in the ClientHello, in order to only have export-grade Diffie-Hellman key agreement. In the server's initial ServerHello message, it sent back an export-grade, essentially an export-grade agreement saying, okay, export is what you want. Again, the man in the middle intercepts it and returns it to what the client originally sent, so the client can't tell that the server is seeing export grade. The client just sees regular full-strength, as far as we know it's uncrackable grade.

But the flaw in the protocol is that the subsequent message from the server containing all of the parameters that are going to be used that the server chooses, the server will have chosen a 512-bit prime number, which is the so-called "modulus" used for this key agreement. And as I said, a flaw in the protocol does not include the cipher suite. This is expected to be fixed in the next version of TLS. TLS 1.3 will probably tweak the spec so that the specific cipher suite is part of this signed package, signed by the server's certificate, which the client is able to verify. If that were there now, this problem wouldn't occur.

But what does happen is the client receives the set of parameters. It turns out that it will accept a smaller prime. It would accept a big one, you know, the full-strength 1024 or 2048-bit prime. But the server sends back a 512-bit prime. It turns out there isn't any reason for it not to accept it. It just decides that the server chose to use that for some reason. So at that point the negotiation finishes. And what the attacker has managed to do is downgrade - this is classic security downgrade attack - has managed to downgrade the exchange of material for them both generating a shared key from what it would normally be, 1024 or 2048 bits down to 512.

Okay. So the one glitch here is that - and our listeners who remember the way SSL works will remember that the very end of this back-and-forth, both sides essentially authenticate their entire communication. They know what they sent. They know what they received. Each side knows what it sent and what it's received. So each side is able to essentially hash the entire conversation and verify to each other that they have the same agreed-upon dialogue that the other side believes. And this was thought to be complete protection from this kind of tweaking. That is, any kind of a man in the middle that changed anything back and forth would get caught out by the so-called "finished" messages that each end sends to the other.

Except it turns out that what these researchers figured out is that the best-known attack on what's known as the discrete logarithm problem, that's the problem that protects the Diffie-Hellman key exchange, in the same way that RSA, where you multiply two primes, the prime factorization problem, that is, that we know no fast way to factor a huge number into the two primes that once were multiplied to get it. We similarly were able to exponentiate very quickly. We can take something to some power fast. But once we have that, in what's called a finite field, where you then take that modulus something, and that throws away a lot of information, so all you essentially have is the remainder from the modulus, but it's a huge remainder. It's 512-bit remainder.

The point is we have no way of performing the logarithm function, a discrete logarithm, which is what we would need in order to reverse that exponentiation function. So that's the hard thing. So of course mathematicians have pounded on this. And there's something known as the "number field sieve," or sieve, I guess you could pronounce it both ways. How do you say it, Leo? Is it sieve or sieve?

Leo: Sieve, yeah, sieve.

Steve: Sieve, okay, sieve. Number field sieve, abbreviated NFS. That's an approach that uses a technique known as "index calculus," which involves four stages. The first three of those depend only on the prime. This prime that I've mentioned is the modulus factor. So in the Diffie-Hellman key exchange, another publicly known value known as the generator, "G," is raised to a secret value. And then that's taken modulus this prime number "p," which is also known publicly. So the generator and the number "p" are publicly known, and they both come from the server. Those are part of the parameters that the server sends the client during the early stages of this agreement. It turns out that, for this number field sieve - sieve or sieve?

Leo: Sieve, sieve.

Steve: Sieve.

Leo: You know, the Sieve of Eratosthenes.

Steve: That's right, or the sieve.

Leo: Remember that was the benchmark we all used back in the Byte days?

Steve: See, I don't know if it's GIF or GIF.

Leo: Well, that one we can debate. But I think sieve is accepted. I'm not sure.

Steve: Sieve, okay. So it turns out that, for this index calculus to be performed, of the four stages, the first three depend only upon the value of this prime number "p."

MACHINE: Sieve, sieve, sieve, sieve.

Leo: I don't see an alternative. Sieve.

Steve: Okay, sieve. So the point is that could be done in advance. It takes vast resources, but it can also be done in parallel. Each stage requires the output from the next, but each stage can be done massively parallel. So if you have huge computing resources, it is possible to precompute for a single given prime "p" the bulk of the computation required to actually break this discrete logarithm for a specific instance of this prime such that the fourth stage can be done in only a couple minutes.

Now, in a back-and-forth TLS handshake, that's within the timeout of both ends. And it's possible to extend that by generating some benign retry errors, essentially, in the exchange in order to keep both ends patient if it was going to take longer. The point is that the problem of the MACs, the Message Authentication Codes, which are made from the hashes of both sides' conversation, before the finished messages are exchanged, it's possible for this man in the middle who has faked out the server to cause the server to issue a 512-bit prime. The man in the middle can compute fast enough for the protocol to be happy, that fourth stage of the number field sieve, or sieve, whatever the hell it's called, and solve the problem, essentially crack the handshake and obtain the key. And from that they're able then to synthesize the proper MAC that they know the client saw in order to satisfy it. So we have a break-in of TLS.

But notice, all of this work, huge amount of work has to be done ahead of time for each prime "p" that's used. And any 512-bit prime can be used. Then they did the research, and it turns out that, of the 8.4% of Alexa's top one million sites which do support the DHE_EXPORT key agreement, of those, that 8.4%, 92% of those use one of two never-changing primes. Apache, that has 82% of that share, has one prime built in. It doesn't need to have one, but it does have one. Which means every Apache server uses the same prime in its Diffie-Hellman exchange.

Leo: Is that like the fallback prime?

Steve: No, it's just, see, the problem is that the implementers did not know what the cryptographers knew.

Leo: Right.

Steve: They figured, since the prime and the generator are public, they can be public, they can be reused. And the problem is, if the prime is widely used among many sites, then you have this offline attack which is possible. And it's funny because many sites have switched to Diffie-Hellman - now, I should mention this is not Elliptic Curve Diffie-Hellman. That's the good one. ECDHE is what we want. There's no weakness there that anyone knows about. This is the discrete logarithm Diffie-Hellman, which was believed to be better than good old-fashioned RSA, which is actually still stronger than Diffie-Hellman in this situation. So I was trying to find - I knew that I had a quote from the paper.

MACHINE: Sieve. Sieve. Sieve.

Leo: When I play this music, does it make you a little nervous? That's the music for that game, I'm sorry.

Steve: Yeah. Actually, they brag about their soundtrack being available somewhere. You're able to download the soundtrack.

Leo: You can buy that and play it, and it would soothe you.

Steve: Yes.

Leo: It's fun.

Steve: Yeah, it's a great puzzle. Yeah, I really like it. So anyway, so they ask in the paper, why are so many servers - oh, and I forgot to mention, this is the same problem with VPNs and SSH in IPSEC. They are using static nonchanging primes.

Leo: Crazy.

Steve: And the problem with VPNs and secure shell is even worse. So they say, "Why are [they] using a single fixed prime modulus?" The answer is it's easier than coming up with a new prime, and it wasn't believed to be a problem. Quoting from the paper, they said: "The NFS algorithm for discrete logarithms allows an attacker to perform a single precomputation, after which computing individual logs in that group has a much lower marginal cost. Although the cheaper cost of individual discrete logs was known to cryptographers, it appears to not have been as widely understood by implementers."

Leo: Yes. You stupid programmers.

Steve: Yeah. So they said: "Indeed, many implementations believed RSA key exchange to be inferior to Diffie-Hellman, which offered forward secrecy. Ironically, the opposite appears to be true. For a medium-value target, a fresh, well-generated 1024-bit RSA key would be significantly more expensive to factor than a 1024-bit discrete log in a group for which precomputation has already been done." Then they finish, saying: "A key lesson from this state of affairs is that cryptographers and creators of practical systems need to communicate better. Systems builders should be aware of the difficulty of cryptographic attacks and tradeoffs, and cryptographers should be aware of how systems are actually being implemented and used in practice." So how do we solve...

Leo: You know that's not what they wanted to write. They wanted to write: "Cryptographers should know programmers are idiots."

Steve: Yes. So, okay. So it is absolutely the case that having a server generate its own prime is trivial. Leo, you set up PGP once, and you'll remember that it initially generated your key.

Leo: Right.

Steve: That's all it takes.

Leo: Right.

Steve: It just, yeah, sure, it has to crank for a while. But you want entropy, and you need to do primality testing to verify that the key you've got is prime. But once you've done that, you're done. And that's the point is this doesn't have to be done per connection. This needs to be done per server. Then every server on the Internet would be using a different "p" modulus, a different prime "p" modulus. So every server would need this massive precomputation in order for its connections to be broken. The reason this is feasible now is so many servers are all using the same single value of "p" because it was in the config for Apache. And mod SSL is the other 10%. Apache's 82; mod SSL is 10%. And it also always uses the same unchanging prime.

So one fix is for every server. And, I mean, this, really, this needs to change downstream. But the problem for today is this attack, which is a downgrade attack. And so right now, if you go to - and I've got the link at the beginning of this. There's a site which you can go to, and it is WeakDH.org, W-E-A-K-D-H dot org. And look at that red banner, Leo, because Apple has not yet done it. Oh, wait. Is that Chrome or Safari?

Leo: I'm on Chrome. I'm on Chrome.

Steve: And what does that say?

Leo: It says, "Warning: Your web browser is vulnerable to Logjam and can be tricked into using weak encryption. You should update your browser." Now, there's probably an update to Chrome that I don't have.

Steve: Probably is. Safari is still showing that, but IE and Firefox - anyway, I think IE may not. Anyway, the point is the first fix is that, irrespective...

Leo: No, I just updated Chrome, and it's still broken.

Steve: That's odd.

Leo: [Woody Woodpecker laugh]

Steve: Yeah.

Leo: Maybe because I'm on a Mac? No. It's the browser; right?

Steve: It's the browser, yeah. It ought to - I'm surprised it's giving you red. I get that in Safari, but in IE and in Firefox I'm getting green in both cases.

Leo: Version 43, whoops, of Chrome. It's up to date, as far as I can tell. It says its up to date. Wow.

Steve: Well, they'll be fixing it soon.

Leo: Yes, they will.

Steve: So here's the deal. Even though the protocol allows the server to send back a shorter "p" prime value, the clients are now saying no. The fixed clients will reject a 512-bit prime, even though technically the protocol doesn't have a problem with that. So that's - and we already have that. I mean, Firefox has it. IE has it. I'm sure Chrome has it somewhere, and I don't - I think Apple will be rolling this out shortly. It turns out it breaks a few sites because it is a break of the protocol, where essentially valid parameters have been returned by the server, and the client is saying, nope, I'm not accepting primes of 512 bits any longer. You need to give me a 1024-bit prime.

Leo: Fortunately, the TWiT site is fixed. Oops.

Steve: Yes. And I wanted to also mention, on the server side...

Leo: I have a cert that says Tech Guy Labs is safe.

Steve: Yes. On the server side, the vulnerability is from the server still offering export-grade ciphers. And our friend Alex Neihaus, on the 21st of May, as soon as this thing came to light, he said, "Running IIS on Windows Server? Logjam is yet another reason to use the @SGgrc cipher suite order." And then he tweeted the link. Remember it's bit.ly/grcciphers. And that's the cipher suite list that my server uses. And believe me, the first thing I did was remove all the export grade crypto of any kind, both the encryption, the symmetric encryption and, you know, for example, there's no RC4. I think there's no RC4. Or if it is, it's way down at the bottom of the list. I think I removed it completely. And also the DHE_EXPORT stuff is all gone. So it's that 8.4% of servers that were just, you know, no one has reconfigured their cipher suites in quite a while, or felt that there might be some reason to allow them to still accept export-grade handshake. And there really is no reason. Really interesting attack, and a simple fix for the browsers.

And in the longer term, we need servers at set-up time to take the time to generate their own prime modulus for the Diffie-Hellman key agreement, rather than just using a globally universal fixed prime because now we know that there is a practical precomputation attack. These guys did it for the 512-bit prime. And they now believe that it is very likely that the NSA has cracked 1024-bit. And it's just as bad. There's no downgrade attack for that. But many systems are relying, the VPNs and secure shell are relying on 1024-bit, all using the same prime. So if the NSA has invested in building this precomputation, it is feasible for them to intercept and perform man-in-the-middle attacks on 1024-bit encryption that's protected with non-Elliptic Curve Diffie-Hellman. And again, I hate that I'm talking about our own national law enforcement as the enemy, you know, the adversary. I really - I just - I don't like that. But it's what happened.

Leo: Well, unintended consequences again. I don't, you know, they were trying to protect the - keep the foreign, I mean, it was a bad idea, but they were trying to keep foreign nations away from strong encryption. And in that time, they've stopped doing that, but in that mis-advised time, they've opened the door to this.

Steve: Well, no, but I mean, even now, even now the idea that the NSA, we're saying, whoo, you know, the NSA may have cracked 1024-bit Diffie-Hellman in the same way that these guys cracked 512, by doing this precomputation attack. The NSA, I mean, the slides that Snowden leaked talked about their ability to get into most VPNs.

Leo: Oh, I see, yeah, yeah.

Steve: And they can now get into 66% of VPNs.

Leo: With very little computational power.

Steve: Yeah.

Leo: They're all using the same prime.

Steve: Yes. For no reason except it was easier. And so they only had to do the massive precomputation once.

Leo: Wow.

Steve: So for what it's worth, I mean, it's beautiful this stuff is coming to light because we're finding little...

Leo: Yeah, we'll fix it.

Steve: ...niches in our security and, one by one, eliminating them.

Leo: I should point out this doesn't make - this only is for servers that support SSL. If I go to Leoville, my website, which doesn't have SSL, the server test just says "connected." It doesn't do anything about - it doesn't report back there's an issue because it's not SSL. Right?

Steve: Right. Well, yeah, and so you have no privacy on those connections.

Leo: Yeah. You never did.

Steve: They're all in - exactly, and no expectation of it.

Leo: Right. But for some reason, I guess maybe we've put the certs in already, TWiT.tv, which isn't in fact an SSL site yet, but it does report that it's secure.

Steve: Wow, very cool.

Leo: Yeah. Now, I don't know about the Chrome problem, but, I mean...

Steve: Yeah, because I absolutely was reading that Firefox, Chrome, and IE had this.

Leo: Maybe the beta version of Chrome, or the Canary? I don't know.

Steve: Oh, it could be in the works. Now, remember, this does break some things. In fact, it turns out, I think it was University of Michigan, where some of the cryptographers were who were working on this, it broke their connectivity because their server was still supporting that. And I'm going to go to - I just fired up my own Chrome, and I'll put in WeakDH.org. And I got a nice green padlock. Oop, red, "Warning: Your browser is vulnerable to Logjam." Yeah. So Chrome is not yet up to speed. And I'm pretty sure Firefox is. Let's go over there.

Leo: Steve Gibson, as always, does it again.

Steve: Yup, good news, your browser is safe against the Logjam attack. So says Firefox. So I'm sure Chrome will get themselves updated shortly.

Leo: It would kind of require a man in the middle to successfully execute this.

Steve: Oh, it absolutely does, yeah, yeah.

Leo: So it's not like it's just going and floating around.

Steve: This is not the end of the world. This requires, I mean, basically there's no evidence this has ever been done. But very smart researchers were able to sort of create a multiphase exploit where they came up with a way of tricking the server to use its export-grade crypto, even though the client didn't offer it export-grade crypto. It used it anyway. And then from that they were able to leverage, understanding that there was a way to crack this discrete logarithm problem using precomputation on a single given prime, they were able to do that. And then they noticed that, oh, my god, everybody in the world's using the same prime. Which means you only need to do the precomputation once, and now you can crack all of the world's connections. So really good stuff. From this we learn every server, when it's set up, should generate its own prime, not use a single static one. And that dramatically makes this attack impractical.

Leo: Steve Gibson's at GRC.com. That's where you should go, get your free copy of something, anything, except SpinRite. That's not free. But it's worth 89 bucks. SpinRite is the world's best hard drive recovery and maintenance utility. Everyone needs it, if you've got a hard drive.

Steve: And the reason you can see me, because the lights are on?

Leo: Yeah?

Steve: SpinRite keeps the lights on.

Leo: Keeps the lights on. I like that. SpinRite keeps the lights on. You'll find a lot of other free stuff there, including all of the stuff we just talked about, shows, he keeps 16 and 64 - I didn't know you keep 64-bit versions there, as well. He also keeps the transcripts there, which is very nice, the show notes, it's all at GRC.com. We have video at our site, TWiT.tv/sn. And of course you're welcome to subscribe anywhere podcasts are available, iTunes and Xbox and the podcast app on your platform, and even great TWiT apps, thanks to our wonderful community. They're available on all platforms. Just seek out TWiT, and ye shall find Security Now!.

Steve: And for what it's worth, what I have is the links to the 64KB versions. They look like my links, but they bounce through Podtrac and then go to your CDN. So basically they're your audio.

Leo: Yeah, but I've got to talk to you about that because they currently don't bounce through Podtrac. But we'll talk off the air. This show is changing its time. So I want to be clear. Somebody tweeted me, oh, Leo, but I...

Steve: What, what, what, what?

Leo: No, Steve, don't listen to this. This is not for your purpose.

Steve: Oh, okay. Oh, yeah.

Leo: Somebody just tweeted me, said but I listen, I like to listen to the show. You can listen on our live stream. We're going to still have a live stream. "Live" may be in quotes. We'll have a stream of this live that you can watch of the shows. But it won't be of the production of the show because for various reasons we want to keep the behind-the-scenes stuff out of the stream. So you will get a stream of the produced version, just like you would get if you downloaded it. And because of that we have to allow for a little bit of time to pass between the finish of the show, currently 3:25 Pacific time on a Tuesday afternoon. I would say we're probably going to allow three hours. So - we've got a new one. I'm getting new schedules as we speak. This is the latest one. And I think this is probably going to be doable, which would mean 6:30 p.m. Pacific, 9:30 p.m. Eastern time, and about 1:30 in the morning UTC, if you wanted to watch the live stream. But then it gets repeated, as we continue to do, throughout the day.

So what we'll have is more like a TV show with an actual schedule. And we're trying to make the schedule so we can hit those times every time, which means allowing for a hiccup here or there. But it usually takes us, if it's a two-hour show, it takes about two hours to get out. So we just want to make sure we have enough time to get it out for you. And it will appear on the stream about the same time as you download it. In fact, if you download it, then nothing's going to change. Your download times will be exactly the same. So that starts effective June 1st, next Monday, for all shows. Roughly speaking, you can just add a few hours to the time of the show's completion, and that's when it'll air on the stream. If that makes any sense at all. I hope it does.

Thank you, Steve. Thank you, everybody, for joining us. And we'll see you next time...

Steve: Thanks, Leo.

Leo: ...on Security Now!. Bye-bye.