SECURITY NOW!

Transcript of Episode #839

Something Went Wrong

Description: This week we, of course, look at the massive global outage that took down all Facebook services for six hours yesterday. But before we get there we look at this week's new pair of zero-day flaws which Google fixed in Chrome. We note the arrival of Windows 11 with a yawn, and also caution about one known flaw that it's already known to have. We look at some potential for global action against ransomware, and some possible movement by the FCC to thwart SIM swapping and number transporting attacks. We also examine a widespread Android trojan which is making its attackers far too much money. And speaking of money, there's a known flaw in Apple Pay when using a Visa card that neither company wants to fix. And finally, after a quick check-in on SpinRite, we're going to examine what exactly did "go wrong" at Facebook yesterday.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-839.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-839-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Boy, do we have a packed show for you. There is a big Android malware going around. Steve will talk about that. Windows 11 is here, but is it ready for primetime? And then Steve breaks down step by step what went wrong at Facebook yesterday and what we can learn from it. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 839, recorded Tuesday, October 5th, 2021: Something Went Wrong.

It's time for Security Now!. Check your BGP routing tables, kids. Here he comes. He knows where he is, thanks to DNS: Mr. Steve Gibson, the man in charge.

Steve Gibson: Yo, Leo.

Leo: Hey, Steve. How are you?

Steve: Good to be with you again for our, what is this, this is the first podcast of October.

Leo: Holy moly.

Steve: And of course our listeners, and you'd have to be under a rock - actually, there was one guy who was playing videogames all day yesterday, and I got a tweet from him at the end of the day.

Leo: What happened?

Steve: Saying "Oh, my god." Because, I mean, he suddenly resynchronized with the rest of the world, and learned what we all knew, was that Facebook, all their properties - Facebook, Instagram, Oculus VR, and WhatsApp, all had a massive, well, comparatively, six-hour outage. So today's podcast gets its name from this wonderful page which was being displayed yesterday. If you insisted on going to www.facebook.com, up came this page titled "Something Went Wrong." And so of course that's the title of today's podcast: "Something Went Wrong."

So we're going to explain what happened. And sort of it was fun because I was already working on today's podcast yesterday in the morning when this all began to happen, so I was able to tweet in real-time and do some digging around and see what was happening. I think it's going to be a fun and interesting podcast for our listeners.

But before we get there, we're going to take a look at this week's new pair of zero-day flaws from you-know-who. Google fixed them in Chrome. In fact, I forgot to try it this morning. I was unable to update to the latest one yesterday, all day. And I thought, maybe this has something to do, like maybe it's weirdly tied in with Facebook because I've never had an update server failure from Google. But I was unable to update Chrome yesterday. Anyway, we were supposed to, we should, because two more zero-days. So we'll talk about those.

We're just going to note the arrival of Windows 11 with a little bit of a yawn, but also caution about one known flaw that it's already known to have, a pretty serious memory leak problem in Explorer. We're also going to look at some potential for global action against ransomware, and some possible movement, but don't hold your breath on this either, by the FCC to thwart SIM swapping and number transporting, or "porting" as it's abbreviated, attacks. We also examine a widespread Android trojan which is making its attackers far too much money, which unfortunately I think means we're going to be seeing more like this.

And speaking of money, I know you mentioned it on MacBreak Weekly. There's a known flaw in Apple Pay when using a Visa card that's set in its transport mode. Interestingly, both companies have been notified. Neither one wants to fix it. So it's still a problem. Anyway, we'll be talking about that. I've got a quick check-in on SpinRite. And then we're going to examine what exactly did go wrong at Facebook yesterday?

Leo: I was counting on that. I said, "Steve will definitely cover this."

Steve: And just sort of apropos of that, I've had a picture sitting in the queue for a long time, a fun Picture of the Week showing Earth's submarine cable network.

Leo: Oh, I've seen that, yeah. Very cool. Yeah. It's amazing.

Steve: And we'll talk about that on the other side.

Leo: Good. Steve, shall we look at the Picture of the Week?

Steve: This diagram really is fascinating.

Leo: You've been looking at it the whole time, haven't you. You're funny.

Steve: I have. It's really interesting. I mean, it's not something you normally think about. But over land we have fibers running around all over the place. But how do you get across the Atlantic or across the Pacific? Or actually, based on this chart, even like down the coast of Africa?

Leo: The Bering Strait, yeah.

Steve: Yeah. And in fact now it's gotten - bandwidth is such a thing that Google and Facebook are laying their own cables, like that they will control. So this isn't just like, you know, originally it was so expensive that you'd put together multiparty ownership in order to share the cost. But with Google and Facebook being as large as they are, there's some ship heading off with a big roll of fiber optic cable on its back, slowly unspooling. And when you think about it there's, like, nothing to contain a cable; right? A submarine cable? It just rolls out the back of the ship and lays on the ground. And, you know, best of luck.

And of course it turns out that maintenance is a problem. There are things that are cutting them. Earthquakes are a problem. Anchors of other ships, you know, you drop anchor, and when you pull it up, you drag a cable up with you and go, whoops, and, you know, slip it off the anchor and hope, and then quickly move your ship so that, you know, you hope that you didn't cut the cable and create problems. So anyway, this chart in the show notes is the earth's submarine cable network which the predominant traffic is from the U.S., both coasts of the U.S. over to their respective other sides on the other side of the Pacific and the Atlantic in order to connect all of this together. So we're able to talk to each other and annoy each other for, like, nothing anymore.

Leo: [Crosstalk] on these cables right now. Right? Yeah.

Steve: Yeah. That is really just something. So it's just sort of a cool thing. In fact, look at that little thing down there, it looks like it's a little blip like from the - it looks like it's linking opposite sides of Texas or something. It's like, what is that down there in the Gulf, in our Gulf? It's like a little blip.

Leo: It's a funny one, isn't it.

Steve: Cheaper to go by sea than by land, apparently.

Leo: Yeah. Houston and New Orleans or something. That's weird, yeah.

Steve: Sort of a little jumper, a little jumper cable.

Leo: It does, doesn't it, yeah.

Steve: Anyway, just very cool. Okay. So another two in-the-wild, true zero-days found and fixed in Chrome. In fact, right now while I'm here, because it'll take no time, I'm going to fire up Chrome. And I just fired up PaintShop Pro by mistake. I pressed the wrong icon.

Leo: So easy to do. I understand.

Steve: Yes, PSP6, still my go-to. Let's see. So Help > About Google Chrome. Oop, I'm still getting an error. "An error occurred while checking for updates: Update check failed to start. Error code 3: 0x80004002 - system level." And then it says "Learn More." And when you click on that, as I did and as I have been recently, it's taking a while to load. Ooh, now, I wonder if a lot of people are clicking on that. Anyway, what it tells you is that errors 3 and 11 are upgrade server is not found. And I'm not even getting that page up now. That's interesting. Everything else is working fine.

Anyway, their most recent update late last week resolved yet another pair of authentic zero-day vulnerabilities which were found being exploited in the wild, thus authentic zero-day vulnerabilities and not let's just call everything a zero-day. For those keeping score at home, this brings us to 14 so far this year, as we finish with month nine and start into month 10. And as I said, when I went to Chrome to ask it to update itself, because I always seem to have to give it a little nudge - and what have you got there, Leo? You're showing...

Leo: 94, it says.

Steve: 94. And I've got it in my notes...

Leo: 94.0.46. This is on the Mac, of course.

Steve: Ah. But at the end is it .71 or .61?

Leo: Dot 71.

Steve: Okay. You've got the latest. I'm stuck on .61, and I don't seem to be able, at two locations, to be able to get Chrome to update itself to .71.

Leo: Maybe it's a Windows thing, yeah. Huh.

Steve: Interesting. Or you just slipped in and got lucky maybe.

Leo: I got lucky, yeah.

Steve: Because, I mean, even now it's like saying, uh, can't get there. Anyway, so issues were CVE-2021-37975 and 976. They were found and, as Google does, immediately put to rest. They were fixed by a total of four patches surrounding a, once again, a use-after-free flaw in the VB JavaScript and WebAssembly engine, as well as an information leak from Chrome's core. The discovery and report of the 975 flaw was credited to an anonymous researcher, so we don't know who, some good person.

But 976 was found by the same researcher within Google's own TAG team, who also discovered the actively exploited use-after-free flaw in Chrome's Portals API that we talked about last week. And eventually, for me and for whoever else might be having a problem, and I'll check Twitter when we do our next sponsor break, Leo, see if other reports are of a problem. But I'm right now still stuck at .61, which is where we were with last week's podcast, trying to get to .71.

So anyway, it's not an emergency. These are going to be probably targeted attacks. It is the case, we know that once they become public, they do tend to get used more quickly because now they have a window of opportunity, which is not for me closing as quickly as it should be. But anyway, Google's on it. And, you know, I was thinking about this. At what point do we start saying this is an awful lot of zero-day vulnerabilities?

I think Google probably takes a bit more heat over these than they deserve, since whereas this does bring the total to 14 zero-day vulnerabilities so far this year, Microsoft fixes scores of such problems every month. And even then only after they decide to stop ignoring them for many months beforehand. So I still think Google is doing the right thing, and I'm glad that we're down essentially to two browsers; right? We've got Firefox, and we have Chromium, you know, Chrome, Chromium, and all of its users. So, good.

Windows 11. Yesterday, thanks to a tweet from Paul Thurrott, we have the official Windows 11 download page. So since I was in a tweeting mode while I was following all of this interesting Facebook adventure, I said: "Anyone seeking additional pain can now obtain the official Windows 11 directly from Microsoft." Then I said, in parentheses: "(I would call it the 'Final Windows 11,' but who are we kidding?)" So, yeah, this notion of a final Windows, well, we've been disabused of that completely, thanks to the Windows 10 adventure. We never did have a final Windows 10. And now we definitively know that they say they never intended Windows 10 to be the last Windows ever.

So every few years I assume Microsoft will find something new to do to the UI that allows them to increment Windows major version number and to enforce some new random and arbitrary restriction on which machines it will run on. And, you know, because I'm a pragmatist, I'm pretty sure that Microsoft will never lose me as a user. For me, the things I need to do, Windows is far more practical than any of the alternatives, though I certainly acknowledge that the alternatives are getting a lot better. But still, Microsoft has and they continue to needlessly discard bits and pieces of my respect for no good reason. I'm just unimpressed with the way they're conducting themselves. The fact that I have to actually remove Candy Crush Soda Saga from my Start menu, are you kidding me? Like, whoa.

Anyway, to make finding Windows 11 easy, I created the shortest practical and easy-to-remember grc.sc shortcut possible. It's just, of course, 11. So grc.sc/11. And that will bounce you to Microsoft's official, it's all there now for everybody who wants it, software download Windows 11 page. And you can download. You can get the ISO. You can download it onto a ready-to-install USB. I mean, they've got that much worked out, at least. So yay for that.

And speaking of Windows 11, we have a known memory leak in Windows Explorer for Windows 11. PC Gamer notes that "Windows 11 will soon be rolling out," and then rhetorically asks: "But will File Explorer keep chomping through your RAM?" At least in what they believe they have, it will. It seems that among the many known new bugs which Windows 11 is introducing is a big new memory leak in Windows File Explorer.

So PC Gamer informs us that: "The Windows 11 File Explorer memory leak bug, which surfaced a couple of months ago, thanks to the keen eyes of one Windows 11 Insider preview user, is outlined in a post by user gyrohan269 on the Windows 11 subreddit. They note that with each instance of Windows File Explorer opened, the RAM usage stacks and doesn't disperse upon exiting. The post was met with thousands of upvotes from those experiencing the same thing, and plenty of comments from users who were able to replicate the issue."

So they wrote: "We're currently running version 22000.194," they say, "which we believe" - and I do, too - "is the release version, on our test bench," they wrote. And this author said: "And I was able to reliably replicate the bug several times. And, no, the RAM usage still hasn't freed up even after about half an hour of waiting."

They wrote: "If you want to check this is a problem for your own Windows 11 version, open Task Manager now and sort your processes by highest memory usage. Then," they wrote, "spam the Win+E," you know, key combination. They said: "You'll notice Explorer rise up the list pretty fast. And once you've closed them all, keep an eye out to see if the memory frees up."

There doesn't seem to be an official acknowledgement of the issue anywhere, let alone news of a coming fix. Which, again, they've known about it for months. It's been not acknowledged by Microsoft, but a lot of noise has been made. Yet they're shipping it anyway because, eh, we'll fix it in post. Thankfully it has been logged in the Feedback Hub; so, if you could replicate it, this PC Gamer site - I have the link in the show notes - suggests to pile on so Microsoft will be made aware that this is a wide-ranging issue.

Leo: There are so many, though. I wouldn't count on getting bumped to the top of the list.

Steve: Oh, I know, Leo. I know.

Leo: Even, as you know, Paul thinks it's way too early. As you said.

Steve: Yes. In fact, it was interesting because I heard him, yes, basically echoing me, yes.

Leo: On the show he said that, yeah, yeah.

Steve: On last Wednesday's Windows Weekly he said: "Come on, really? This is not even beginning to be ready." But the show must go on. Hey, I figure once the corners have been rounded, it's like, well...

Leo: It's done.

Steve: I mean, it's like they've given up; right? They've just like, oh, you know. Yeah. Like we're never going to get this right. We're never going to actually finish this. So let's just push out a stream of these. Now we'll just call it 11 rather than 10.

Leo: It is interesting, though, that we are now accustomed to the notion that every bit of software is constantly updated; right? There's no longer this sense that anything's done. We know it's all going to be updated forever until they stop, until they give up.

Steve: Well, you know, we know the lone, the sole voice in the wilderness.

Leo: Yes. Oh, yeah, you have one program, yeah, okay. All right. No bugs in your software.

Steve: Well, and when it's done, somebody over in the Spinrite.dev newsgroup the other day suggested, pointed me to some fancy automated system for automatically checking for patches and downloading them and integrating them in software that had already been shipped. I mean, like proposing that I use that. And I said...

Leo: Why? Why?

Steve: You know, SpinRite 6 hasn't changed since '04. The DNS Benchmark hasn't changed since I shipped it. Or it's like there was a couple little things shortly after launch, but then it's fine. Never10, well, not since 10. SQRL never had a bug, and I didn't touch it since I finished it. So yeah. And in fact I do have a DNS-based means for all the software moving forward to ping a pseudo server that I run to just check to see if anything might have changed. All is quiet on that front.

Leo: Although that's prudent, to put that in, if you had to do a patch; right.

Steve: Yeah, makes sense.

Leo: Yeah.

Steve: Makes sense to have it. Because of course the moment you don't, well, actually not even then. But still. But Leo, I take your point. Yes. All other software is just a moving target. It's like, well, you know. And really I have the newsgroup gang to thank because I'm getting ready to launch the next release. And they will happily pound on it and will find some obscure stuff where, if you touch your nose when you're standing on your right foot and spin around three times and hit ENTER while coughing, then oop, look, sure enough, it's a different shade of green on the screen or something. But I would rather fix it now than fix it later.

Okay. There is some news, kind of, on the ransomware and cyberwarfare front. And what's sad is even that we have a term "cyberwarfare." Okay. The U.S. announced last Friday that the administration will be conducting a series of online virtual meetings with representatives from 30 countries, because after all they're online and virtual, so let's just all get ourselves a little Zoom window, and we're going to do this. Or maybe we're going to have somebody on that stage that we saw, right, Biden was doing a bunch of, I don't know what they were, but lots of little windows, and everybody was talking at once. Anyway, this will include NATO allies and G7 partners on the topic of cybercrime, with an explicit focus on ransomware and the concomitant abuse of cryptocurrency.

In their press release on the topic, the White House said: "This month, the United States will bring together 30 countries to accelerate our cooperation in combating cybercrime, improving law enforcement collaboration, stemming the illicit use of cryptocurrency, and engaging on these issues diplomatically." And actually I didn't make it into the show notes, but I did see in passing that the U.K. has budgeted a bunch of money. I had to do a double-take. And apparently it involves proactive cyberwarfare. It's like, ooh, the tip of the spear. So okay. Or the tip of the packet, at least. Accordingly to the release, additional topics to be discussed will include 5G technology, supply chain attacks, quantum computing - I'd love to be a fly on the wall for that discussion with 30 countries - and AI.

So I guess, you know, you poke the bear enough, and it rouses. As we know, last May was the attack against Colonial Pipeline, which resulted in fuel shortages across the U.S. East Coast. The next month, in June, the attack on JBS Foods disturbed the supply of meat across the U.S. Don't mess with the bear's meat supply. And then in July the massive series of attacks which leveraged flaws in Kaseya's IT management servers, which as we well know created disruptions at hundreds of companies across the U.S. and more than 1,500 globally. So we're told that President Biden first raised the issue of ransomware attacks carried out from within Russia's borders with Putin during a face-to-face meeting last June, and that he again raised the issue in a phone call in early July. Presumably he's supposed to have a great relationship with Putin. So saying, look, crack down on gangs operating in Russia. You're really pissing us off over here.

And now we know that, if that was done, what actually transpired remains unknown, but it apparently didn't last very long since multiple attacks resumed last month. And this did lead the FBI to formally conclude that they saw "no indication" that Russian officials had actually taken any effort to crack down on these groups. As we talked about, what, I guess when these attacks resumed, it looked like maybe they'd taken a vacation. And they said, you know, everybody needs a little time off. We're back. Wonderful.

So there are a number of big, significant, and very interesting macro trends taking shape. What governments ultimately decide to do about encryption that they cannot crack is another one. So it's going to be very interesting to see how this all plays out. We actually do appear to be entering a period of true inter-nation economic cyberwarfare. And that idea, as I said, still startles and unsettles me. I don't want it to be true. But it seems to be becoming so.

On the topic of thwarting SIM-swapping attacks, as we know, I titled our August 31st podcast "Life: Hanging by a PIN." And I know from the much greater than average level of feedback I received that my painting a clear picture of just how vulnerable we would typically be if our smartphone number were to fall into the hands of an attacker, that a greater awareness of the trouble did hit home with our listeners. So I was interested to see, and I was initially hopeful, and I wanted all of our listeners to know that at least some of the U.S. bureaucracy is awake to this threat and is beginning to move on it. The bad news is that the specific arm of the U.S. bureaucracy is the FCC.

Last Thursday the FCC, of course our Federal Communications Commission, announced its plans to introduce new rules for U.S. mobile carriers to govern any changes made to their subscribers' telephone numbers in an attempt to address the growing problem of SIM swapping and port-out fraud attacks. Okay, that sounds great; right? Of course, as we know, these attacks surround mobile carriers' failure to correctly verify the requesting party's identity when that party requests either that their service be transferred to a new SIM card, or to an account at another mobile operator. That podcast #834 painted a very clear and depressing picture of just how much devastation could result.

The U.S. Justice Department has charged many individuals over the past few years with theft enabled by SIM swapping and port-out fraud. And some of the victims of these thefts have, understandably, brought lawsuits against their mobile carriers in an attempt to recover their monetary losses. Many of those lawsuits are still working their way through the U.S.'s delay-prone legal system, so no conclusions to those yet.

In response to the problems, some U.S. carriers have introduced additional verification measures to attempt to limit this sort of fraud, but the SIM-swapping gangs have also upped their game. Some groups have started bribing carrier employees, or they've used vulnerabilities they've identified in the carriers' networked backend systems to perform the attacks for them, thus skipping the need to have any direct contact, and trick the carriers' frontend support staff.

In its press release Thursday, the FCC announced that in response to having received "numerous complaints from customers" - I'll bet - that it's initiating a "formal rulemaking process" by issuing a "Notice of Proposed Rulemaking." Okay. These are the same people who said they were going to outlaw and prevent telephone spam. So I think this means that it's still going to be up to us to protect ourselves.

An FCC spokesperson told "The Record," who reported on this issue, that: "The FCC's rulemaking process generally starts with a Notice of Proposed Rulemaking that asks questions and makes proposals. We then have a period during which we take public comments, generally made through our Electronic Comment Filing System."

Leo: Which as we know in the past...

Steve: Exactly.

Leo: ...has been somewhat unreliable.

Steve: Just to remind everyone, that's the ineptly designed system that was brutally spammed during their Net Neutrality Request for Comments period, you know, during which more people than exist on Planet Earth weighed in with their opinion about Net Neutrality. Anyway, they concluded: "After that, we review comments before taking any next steps." Because, oh, goodness, we would not want the FCC to act prematurely on our behalf of securing our SIMs and require them to be more secure before allowing them to be moved. So although I wrote about the fact that some of the U.S. bureaucracy is awake to this threat and is beginning to move on it, unfortunately that bureaucracy is the FCC. And I have a friend who's...

Leo: Really wants to talk to you, yes.

Steve: He's having his home repiped, and that's quite an adventure.

Leo: Oh, my.

Steve: So I'm getting pipe-by-pipe updates.

Leo: Well, they've put in the drain pipe now. Wow.

Steve: Yeah. His house, well, his house was built in '84, and he's beginning to have pinhole leaks.

Leo: Oh.

Steve: Apparently as a consequence of Southern California water, which by the way is why I'm on the other side of a quad-filter reverse-osmosis filter. It is undrinkable down here in Southern California.

Leo: Plus you steal it from us.

Steve: Leo, remember when...

Leo: We poison it before we send it to you.

Steve: Yeah. Remember when we were growing up you'd have the hose on, like on a hot Saturday afternoon.

Leo: You'd drink from that.

Steve: You just slurped.

Leo: So good. So good.

Steve: Oh, my god. It was. What happened?

Leo: But see, you were up here in Northern California. The water was better in San Mateo back in those days. San Francisco has very pristine water.

Steve: Even now you can drink it?

Leo: Yeah, it's from the Hetch Hetchy. So it's, yeah, it's good. I'm not sure where L.A. gets its water. But I'm not surprised it's not good.

Steve: Comes out in the sewer system, I think. Right now really you don't want to be getting it out of the ocean, lord knows.

Leo: No.

Steve: Because we've got a big oil spill down here.

Leo: Yeah, right off the coast, right where you are, yeah.

Steve: Don't I remember, are you using well water?

Leo: We have a well. And we, like you, have a significant amount of filtration. We have softening because there's a lot of iron in it, and then it goes from the softening, and the drinking water specifically is reverse-osmosis and charcoal filtered, yeah. It's delicious, though. By the time it's done, man, that's good. It's kind of like getting a hose in your backyard, out of the backyard. All right. We were going to take a break, I think, yes?

Steve: Yes, we are.

Leo: And then we will talk about a new Android trojan.

Steve: Ooh.

Leo: Ooh. And of course the Facebook story coming up.

Steve: And Leo, you probably haven't scrolled ahead, but I just, to be a little more punchy, I wanted to list the number of apps which were found with that trojan. So just prepare yourself.

Leo: Oh, it's a long list. I shall scroll through them. Holy cow. Wow. Malware on the Android again.

Steve: Yeah. I don't talk about Android a lot because, well, there's just a lot to talk about. Zimperium, the research group that we've often referred to, recently revealed their research into one of the best-named Android malware campaigns I've seen in a long time. So it's a trojan. So we have a horse. And it signs its victims up to high-cost services to reap the rewards. So they named it GriftHorse.

Leo: I get it.

Steve: It's just perfect.

Leo: Don't look it in the mouth.

Steve: Don't look that GriftHorse in the mouth. That's right. So but what caught my eye was the tremendous number of apps, the tremendous effort that had gone into the creation of GriftHorse's, count 'em, 139 individually created and infected with a trojan app. Some of them do what they say they're going to do. Some of them don't do anything. But there's just a ton of them. And in fact a total infection base of greater...

Leo: Now, some of these were not created by the virus, like Forza H or TrueCaller. So there must be a library they're using or something.

Steve: Right, right.

Leo: Because some of these are legit apps. That's what's really scary. These are not...

Steve: Exactly.

Leo: ...just throwaway apps.

Steve: Right. And like Handy Translator Pro, there were a million downloads of that.

Leo: Yeah. Oh, this is fascinating.

Steve: So Zimperium explained, they said: "The threat actors have exerted substantial effort to maximize their presence in the Android ecosystem through a large number of applications, developer accounts, and domains. The Zimperium zLabs researchers have noticed the technique of abusing cross-platform development frameworks to stay undetected has been on the rise, making it more difficult for legacy mobile AV providers to detect and protect their customers.

"The timeline of the threat group dates back to November 2020, suggesting that their patience and persistence will probably not come to an end with the closing down of this one campaign. The threat to Android users will always be present, considering the innovative approaches used by malicious actors to infect the victims." They finish: "The numerical stats reveal that more than 10 million Android users fell victim to this campaign globally, suffering financial losses while the threat group grew wealthier and motivated with time. And while the victims struggle to get their money back" - often not possible, as we'll see in a second - "the cybercriminals made off with millions" - actually hundreds of millions - "of euros through this technically novel and effective Trojan campaign."

So just to give people, you know, instead of saying, oh, yeah, there's a lot of them, I grabbed the list and formatted it for the show notes. I mean, it's just like it's a Who's Who. And they estimate as many as 17,345,450 downloads spread across this 390 Android apps. So they summed up their position on this by writing: "These mobile applications pose a threat to all Android devices by functioning as a trojan that subscribes unsuspecting users to paid services, charging a premium amounting to around 36 Euros per month, which is about $42. The campaign has targeted millions of users from over 70" - seven zero - "countries by serving selective malicious pages to users based on the geolocation of their IP address with the local language. This social engineering trick," they write, "is exceptionally successful, considering users might feel more comfortable sharing information to a website in their local language.

"Upon infection, the victim is bombarded with alerts on the screen letting them know they had won a prize and needed to claim it immediately. These pop-ups appear no less than five times per hour, until the application user successfully accepts the offer." Just to shut it up. "Upon accepting the invitation for the prize, the malware redirects the victim to a geo-specific web page where they are asked to submit their phone number for verification. But in reality they are submitting their phone number to a premium SMS service that would start charging their phone bill over 30 euros per month. The victim does not immediately notice the impact of the theft, and the likelihood of it continuing for months before detection is high, with little to no recourse to get one's money back.

"These cybercriminals took great care not to get caught by malware researchers by avoiding hard-coding URLs or reusing the same domains and filtering/serving the malicious payload based on the originating IP address's geolocation. This method allowed the attackers to target different countries in different ways. This check on the server side evades dynamic analysis, checking for network communication and behavior. Overall, GriftHorse Android Trojan takes advantage of small screens, local trust, and misinformation to trick users into downloading and installing these trojans, as well as frustration or curiosity when finally accepting the fake free prize spammed into their notification screens."

So I wanted to put this particular campaign on our listeners' radar because, just as when we first talked about ransomware many years ago, and I commented then that it felt to me as though it was really going to become a problem in the future, this feels the same way. The trouble is that the attackers behind this were known to be netting several million dollars per month. And this creates a lot of motivation. And many of the new facets of this approach have been proven to be surprisingly successful now. So it's not like they tried it, and it didn't work. They tried it, and it did work.

And unlike a single ransomware attack which inherently creates a single point of failure for the attacker if their victim chooses not to comply, even in the face of all the threats and extortion and threats to disclose information and everything, in this model the individual damages in this attack are small, they're incremental, and widely dispersed across 70 countries and 10 million individuals in many languages. So they individually slip under the radar by siphoning a small cash flow from many millions of individuals who are end users. And in the process they create what amounts to a stable cash flow of illicit funds for the bad guys.

This GriftHorse campaign not only managed to fly under the radar and to avoid AV detection for many months, like starting back in November, and it just now came to light, it likely surpassed hundreds of millions of dollars in total amount plundered from its victims until Zimperium responsibly notified Google of their discovery, and Google immediately purged those identified apps from the Play Store. But even so, those apps continue to be available on untrusted third-party app repositories, which again underscores for us the risks associated with side-loading arbitrary apps from untrusted sources. So I'm afraid that we can expect to see more of this style of attack in the future, probably on the Android platform, since it's where it makes the most sense.

Over on the Apple side, there's a problem that's been identified with Apple Pay.

Leo: In case you Apple people feel smug and safe.

Steve: That's right. Before you go, "Huh."

Leo: So there.

Steve: We've got a problem with Apple Pay and Visa. And disturbingly, both companies have been informed, but neither has chosen to fix the problem because they're arguing over whose fault it is. Even though either one of them could mitigate the trouble at their end. So a handful of researchers at the University of Birmingham and the University of Surrey, both in the U.K., have written up their research in a paper which will participate in the 2022 - wow, Leo, that's coming up before we know it, 2022 - IEEE Symposium on Security and Privacy.

Okay, now, their paper's abstract suggests that maybe they've been spending a little too much time in the lab. They should get out more. They explain rather densely: "Relay attackers" - meaning man-in-the-middle, man-in-the-middle relay attackers - "can forward messages between a contactless EMV bank card and a shop reader, making it possible to wirelessly pickpocket money. To protect against this, Apple Pay requires a user's fingerprint or Face ID to authorize payments, while Mastercard and Visa have proposed protocols to stop such relay attacks. We investigate transport payment modes and find that we can build on relaying to bypass the Apple Pay lockscreen, and illicitly pay from a locked iPhone to an EMV reader, for any amount, without user authorization." Gulp.

"We show that Visa's proposed relay countermeasure can be bypassed using rooted smartphones. We analyze Mastercard's relay protection and show that its timing bounds could be more reliably imposed at the lower protocol level, rather than at the EMV protocol level. With these insights, we propose a new relay-resistance protocol" - they call it the L1RP - "for EMV. We use the Tamarin prover" - that's something we talked about back when we were talking about all of this, the idea of doing robust proof using a formal proof technology, that's this Tamarin prover - "to model mobile phone payments with and without user authentication, and in different payment modes. We formally verify solutions to our attack suggested by Apple and Visa and used by Samsung, and we verify that our proposed protocol provides protection" - oh, there we go, proposed protocol provides protection - "from relay attacks."

Okay. So what does this mean practically? Elsewhere, they make this more clear, explaining: "Contactless Europay, Mastercard, and Visa," thus EMV payments, they say, "are a fast and easy way to make payments that are increasingly becoming a standard way to pay. However, if payments can be made with no user input, this increases the attack surface for adversaries and especially for relay attackers, who can ferry messages between cards and readers without the owner's knowledge, thus enabling fraudulent payments. Payments via smartphone apps generally have to be confirmed by a user via a fingerprint, a PIN code, or a Face ID. This makes relay attacks less of a threat.

"However, Apple Pay introduced the Express Transit/Travel feature in May of 2019 which allows Apple Pay to be used at a transport-ticketing barrier station without unlocking the phone, for usability convenience. We show that this feature can be leveraged to bypass the Apple Pay lockscreen and illicitly pay from a locked iPhone, using a Visa card, to any EMV reader, for any amount, without user authorization. Furthermore, Visa has proposed a protocol to stop such relay attacks for cards. We show that Visa's proposed relay countermeasure can be bypassed using a pair of NFC-enabled Android smartphones, one of which is rooted."

Okay. So here's the bottom line. The Apple Pay lockscreen can be bypassed for any iPhone with a Visa card set up in transit mode. The contactless limit can be bypassed allowing unlimited amount EMV contactless transactions originating from that locked iPhone. An attacker only needs remote access, a powered-on iPhone, stolen, in a handbag or in a pocket. It does not need to be unlocked. The transactions can also be relayed from an iPhone inside someone's handbag or pocket, as I said, without their knowledge, and the attacker needs no assistance from the merchant. So no clicks anywhere.

Backend fraud detection checks have not stopped any of their test payments. They did a 1,000 euro transaction, no red flags, no alarms. It went through. So just to be clear, this attack is made possible by a combination of flaws in both Apple Pay and in Visa's system. It does not, for instance, affect Mastercard on Apple Pay or Visa on Samsung. You've got to have Apple Pay and Visa both. Their research provides formal modeling that shows that either Apple or Visa could mitigate this attack on their own. As I noted above, both companies have been informed months ago, yet neither have fixed their system. So the vulnerability remains live and workable today. The researchers recommend that all iPhone users check that they do not have a Visa card set up in transit mode. And if they do, it would need to be disabled to prevent this attack from succeeding against that phone and that card.

So it's not that it's not going to happen, but it could. Right? I mean, so it seems like I don't use my iPhone. I do have a Visa card. Yeah, I do.

Leo: I'm sure you don't have transit mode turned on because there's no subways in Irvine.

Steve: Exactly.

Leo: Or maybe you take the bus. Do you take the bus, Steve?

Steve: No. I don't.

Leo: I don't see you in a bus on your way to Starbucks.

Steve: Hopefully, Apple will get some flak now that this research is public, and they'll fix this.

Leo: They're blaming Visa, though. They say, well, it's not our problem. It's a bug in Visa's stuff.

Steve: I know. And here we have another instance of Apple only fixing something when someone makes them. And even then, it's like saying no, no, pointing the finger at them. The researchers make it clear, Apple could fix this if they chose. But no. And that's disappointing.

Leo: Yeah.

Steve: Just a quick update on "Foundation." You and I chatted about it briefly before we began recording, Leo. In my opinion, no improvement after the third show, third episode. It's just not fun. You know, it's not exciting. It's slow and heavy and boring. And, you know, so I'm hoping that when hostile aliens invade the Earth this Friday on Apple's new series "Invasion," which begins on Friday, that that will provide some much-needed sci-fi excitement. Because there's no excitement coming from "Foundation." Wow.

Leo: Yeah, yeah. That's too bad. I hope "Invasion" will be good, too.

Steve: And I heard you ask Alex what he imagined this cost them to produce.

Leo: Yeah. Did you hear?

Steve: Yeah. Tens of millions of dollars for an episode.

Leo: He said 50. He said 50, which is, like, movie money. $50 million an hour, that's how much a - I can't believe it'd be that much. But maybe it is. 15 million was "The Morning Show," and that didn't have aliens. Not that aliens are necessarily more expensive. Just a little rubber makeup and...

Steve: I was trying - yeah.

Leo: They didn't have spaceships. Now, they're expensive.

Steve: Matt Kopit tweeted to me: "Regarding the sound quality on 'Foundation,' I noticed the same thing with muddled dialogue, and found that disabling Dolby Atmos in my Apple TV's audio settings improved things dramatically. Your mileage may vary, but feel free to give it a try."

Leo: That makes sense. That might make sense, yeah.

Steve: So that's worth considering.

Leo: I didn't notice any problem on my stereo speakers. So maybe that is the case. So it's been Dolby that's the problem, yeah.

Steve: Yeah. Over the weekend I posted an update to the patient gang who's been waiting to test and pound on the next release of what...

Leo: And they responded back, apparently.

Steve: Who will be pounding on the next release of what will become, as we know, SpinRite v6.1. I explained that I had reached the point where, as far as I can tell, everything is working. I tracked down a problem with the non-DMA IDE driver, and I made the BIOS associator a bit more bulletproof. I had previously invested a bunch of time updating SpinRite's original benchmarking code for 16 bits' worth of sectors. That is, it used to be able to do 32 bits, no drives back in 2004 were even close to 32 bits' worth of sectors.

Of course now we know that, what is it, 2.2, 2.3TB, and drives are bigger than that now. So we've added another 32. Actually, we added another 16 bits, so they're 48 bits. But I figured, well, I'm doing 48, let's get ahead of the curve for a change. I'll do 64. But really that'll take care of us for a long time, literally until the aliens have come and gone. But I hate the way the benchmark is working, and I've decided to scrap it and rework it now since I can't live with it.

The trouble is that the only timing reference the original SpinRite had was the PC's counter/timer which is a 16-bit counter/timer that runs at 1.193 MHz, which is one count every 838 nanoseconds. The counter/timer is still there, and that's plenty of resolution. But the counter/timer chip was a ripple-counter which could sometimes be sampled and caught mid-count, or mid-ripple, which would produce a false count result. The chip did have a latching function, but I encountered some that still produced bogus results. So I developed a sanity filter that would take three successive readings from the counter, and SpinRite would only believe the final one if each of the previous two readings showed the same count or one that was increasing. And the filter also had to be smart about the 16-bit counter's wraparound since that happened 18.2 times per second. If the filter thought that it had obtained a mid-count reading, it would just start over.

The point is that, back then, that's all I had. So that's what I had to use. Today's RDTSC, which is the read timestamp counter instruction, did not exist as it does now. My new code, the code I've already written for the ReadSpeed benchmark, uses the RDTSC instruction for various delays it needs when waiting for hardware to settle, like down in the nanoseconds range, for hardware to settle, and for all of its benchmark performance measurements. But SpinRite had not been using it for the benchmark timing since I was trying to minimize the rewriting. And I figured that what SpinRite was already doing was fine. But I had to go back into that old code to make sure it would work, and I saw just how horrendous it was.

So I need to rip it out and replace it with the newer code that I have already developed for the ReadSpeed benchmark. Although it doesn't technically need it, SpinRite will be needing the new system's insane picosecond resolution in the future, and 6.1's code will be surviving for years. So now is the time. I've already worked out all the details for ReadSpeed, and I have all the formatting and everything ready to go. So it's just a matter of removing the old bad code and moving the newly written good code over. Once that's done, I'll be glad that I did. And that will be Release 4 of the pre-release of SpinRite, which the gang will pound on while I'm working on finishing up the rewrites of the data recovery portion. And then it's going to be ready for use. So we're getting there.

Leo: Excellent. Excellent. Well done. I can't wait. And it just shows, you know, you don't settle. You just don't settle.

Steve: I try to, but I...

Leo: You can't. Your name's on it. It's going to reflect you. And given how long - how long ago did System 6 come out? SpinRite 6?

Steve: '04. 17.

Leo: '04. So given that this is - it's going to be like another while. You don't want it to be not perfect. Right?

Steve: I want it to be...

Leo: 17 years, wow.

Steve: Although the truth is, because we are connected now the way we weren't then, you know, back then I was arguing with Egghead about why they were returning boxes of SpinRite that were fine. It's like, whoa, we have a return policy that lets anyone return anything they want at any time. I said, well, you realize that they bought it, they took it home, they fixed their hard drive, and they brought it back to you. And you've sent it back to me. Anyway, the point is I'm much happier with the way the world is today, 17 years later. And Egghead is gone, and sorry.

Leo: How much, by the way, out of curiosity, how much of the retail price did Egghead keep? 50%?

Steve: I think they got 30 because they - but they didn't buy direct. They had to buy through Softsell. And Softsell got 50 plus marketing - oh, excuse me. Did I just say that?

Leo: Let your feelings out, Steve. It's good for you. You don't want to...

Steve: Leo, it was so frustrating working through distribution because we would send the stuff off. Remember we had like bound manuals with spiral binding.

Leo: It's expensive, yeah.

Steve: Both the 5.25 and the 3.5" disk sizes. And a registration card. This whole beautiful box, all color printed. They'd go off in big shipments, and then half of them would come back damaged. They looked like elephants had trodden on them. Of course, so they would deduct that. And then it was supposed to be net 60, and I had two people working full-time just to get us paid through, first it was Softsell and then Ingram Micro. We had a couple of the big distributors that were doing all of that. And so I'd end up waiting half a year to get my money.

Which, you know, meanwhile they had the product. The customers had it. Everyone loved it. But I wasn't getting paid. So I remember just swearing to myself, the instant I can cut these people out of this, I will. And thank god for the Internet because it's allowed me to work directly with my end-user customers, and things are much better.

Leo: The reason I ask is when people complain about Apple's 30%, that's nothing compared to the 50% taken by the distributor, Softsell or Ingram or whoever that was. And then you only get 70% of the 50%, minus marketing, before, you know...

Steve: So I did not have a zero cost, either.

Leo: Right.

Steve: With everything being an electronic download, the publisher really, it's like they're complaining that they're losing 30%, but they have zero cost.

Leo: Right. No download cost. No manual cost.

Steve: Duplicating diskettes and stuffing all this and shrink wrapping everything.

Leo: You maybe, if you were lucky, took 10-20% of the selling price as profit. I mean, 70% is pretty damn good, I mean, compared to what it was back in the day.

Steve: Yeah.

Leo: All right. Take a little break. When we come back, we will look at the Facebook - people just joining us saying, "Did he talk about Facebook yet?" No, he didn't. It's coming up next. Steve Gibson.

Steve: Because you know, Leo, something went wrong.

Leo: I hear. I hear that. That's the name of our show.

Steve: Not good.

Leo: Facebook. Sorry. Something went wrong.

Steve: Something went wrong. I love the screen that came up because it says "We're working on it." Well, first it says "Sorry, something went wrong. We're working on it, and we'll get it fixed as soon as we can." And then there's a link, "Go back." So, like...

Leo: Was the image broken, too? That's the other thing I love.

Steve: Yes.

Leo: Something went wrong. We couldn't even give you an image. Sorry. We don't even have that.

Steve: Don't even know what that - I would love to know what that was of. But you can't see it now. And I love it because then of course they added their copyright. Facebook copyright 2020.

Leo: Oh, whoops.

Steve: Yes.

Leo: They couldn't even update this. They were so out of luck.

Steve: Yeah. Okay. So the Internet's big iron routers are connected to each other by their peering interface links. For example, if you had a piece of paper covered with a bunch of circles representing routers, then drew straight lines between each of the circles close to other circles, those straight lines linking circles would be peering links. Those peering links carry the low-level packet traffic which routers route. And the routers also use those same links to maintain persistent TCP connections over which the BGP protocol flows. BGP, the Border Gateway Protocol, is a TCP-hosted protocol like HTTP, FTP, SMTP.

But in this case BGP is a peering protocol exclusively used by routers to talk to each other, specifically to talk to their immediately adjacent connected neighbors. The conversation they have has the purpose of synchronizing their respective routing tables with each other, with the routers to which they peer. Which is to say to which they are directly connected. When a router receives a change to its own routing table, either introduced by a local admin or received from another router peer, it updates its own table appropriately to reflect the changes, then sends news of those changes which it has just made to its table to its other peers so that they, too, might make any needed adjustments.

Now, a large enterprise which has its own Autonomous System number, AS as it's called - Facebook's AS is 32934 - will use their large backbone routers to connect their public enterprise network to the rest of the public Internet. Their routers will know which IP ranges belong to its enterprise owner, and will contain entries in its routing table to route traffic bound for those IP ranges to the enterprise's router interfaces.

And since every such router shares its routing table with its peers over BGP, all of its peering routers will also know that any traffic they receive for those IPs should be forwarded to that Autonomous System router. And since those routers also share their routing tables with their peers, all of the routers they connect to will also know. And so it goes, over and over, peer by peer, until every big router on the Internet knows where to send traffic that's bound for any of that enterprise's IP ranges.

In the weird parlance of BGP routing, we say that the original router is "advertising" the routes which it alone is able to handle by forwarding any incoming traffic to its Internet-connected enterprise. So it advertises the routes for the traffic it should receive.

So yesterday, at 11:39 a.m. Eastern Time, shortly before noon (15:39 UTC), someone at Facebook updated the routing information for Facebook's networks to something that no longer worked. Even now, following Facebook's official post-recovery blog posting, they're not telling the world exactly what happened. Initial reports suggested that it might only be the routing to Facebook's four authoritative DNS servers, which are a, b, c, and d dot ns, as in name server, a, b, c, and d.ns.facebook.com, that may have been messed up to render those crucial servers unavailable. But some additional evidence suggests that the trouble was much bigger than that.

The reports of a DNS-based failure came from those whose own DNS servers suddenly started returning SERVFAIL errors, indicating that none of those four authoritative Facebook DNS servers responsible for Facebook.com were replying to their queries. Presumably, those four authoritative Facebook DNS servers were still up, but they had just become unreachable due to a routing error. The reason I believed that DNS was only part of a much bigger, probably network-wide all-of-Facebook problem, was that, as shown in the diagram I have in the show notes traffic being served by Facebook dropped like a rock off a cliff.

And in the show notes I have this chart which shows the moment this happened, and traffic just collapsed. It went from near its peak around noon on the East Coast, which because there is sort of a sinusoidal traffic pattern that we see from domestic-based companies. It just over the course of about five minutes just went to zero.

So as we know, DNS caches. And caching is a core feature of DNS. If this were "just," and I have that in air quotes, a major DNS outage, we would expect to see a far more gradual reduction in traffic to Facebook over a span of hours, rather than minutes, as the Internet's massively dispersed and distributed DNS caches which exist at every level, even right down to the individual end-user smartphone and desktop PC. As those individual cache entries independently expired, their own local DNS would only then go in search of an IP address to update. And only when an expired entry could not be updated would the user's local machine report that Facebook had become unavailable; or, as their copyrighted page wonderfully stated, "Something went wrong."

Yesterday, during the outage, just like the rest of the world, I was unable to query Facebook's authoritative DNS servers. So I was unable to obtain the details of how Facebook had set up their DNS. But this morning, with Facebook back on the air, we can more closely examine the way they have their DNS configured. I used NSLOOKUP to pull the SOA, which is the Start Of Authority, DNS record directly from one of Facebook's authoritative DNS servers: a.ns.facebook.com. Here's what I received. In the show notes I have a picture of the output of NSLOOKUP, which I had configured to do a query type of SOA.

And I queried Facebook.com using the server a.ns.facebook.com. And what we get back is a couple of - both an IPv6 and an IPv4 address. And then the Start of Authority record which shows the serial number for that record; the refresh interval in seconds, which is 14,400, which is four hours; the retry interval, 1,800, which is 30 minutes; the expiration, which has 604,800 seconds, in other words, seven days; and the default TTL, the time to live for this record, which is 300 seconds, or five minutes.

The last two items there tell the story, the record expiration time and the default TTL. Facebook's default TTL for their DNS records is only five minutes.

Leo: That's pretty short; right?

Steve: That's very short.

Leo: Yeah.

Steve: This means that, unless it's been overridden by a per-record TTL, and Facebook's A and AAAA records for its IPv4 and IPv6 IPs are not overridden, that means that once every five minutes, every DNS client on the planet will see that its local cached copy of Facebook's IP address has reached its end of life and should be re-fetched from that client's configured DNS server. But if that update query fails, what happens? That's where the SOA's specified record expiration time keeps you from being SOL. Although Facebook uses a short TTL, they also use an expiration of seven days. And that will be seven days from the most recent previous successful update, which would have been within the past five minutes.

This means that even if Facebook's DNS had dropped off the face of the Earth, so long as the rest of Facebook was still present, clients on the planet would have happily continued using Facebook for the next week while their DNS patiently and periodically attempted to check in for a DNS update. I figure that they have such a short TTL as part of their load balancing because the servers are able to decide which IPs they want to give out. Typically they have huge banks of IPs. And by asking clients to come back to the trough for another IP every five minutes, that allows Facebook to dynamically decide which IPs they want to send where. But again, in the event that there's no IP available, all the clients have been told you can keep using what you have for a week.

So in other words, all of the evidence points to this being a massive whole-of-Facebook Internet outage, an Internet routing error. Add to that the fact that Facebook's Instagram service also disappeared from the Internet, despite the fact that Instagram's DNS is hosted by Amazon, not Facebook, and Amazon didn't have a problem. So this wasn't about just DNS. As we know, DNS caches. But what doesn't cache is BGP. If someone at Facebook made the colossal mistake of deleting all of Facebook's routes from the Internet, that change would have propagated at the speed of the Internet. And within a few minutes, just as we saw from that sheer Internet traffic cliff, no Internet backbone routers anywhere in the world would have had any idea what to do with a Facebook IP address, even though all of those users would have still had them cached for the next week.

Since BGP is just a protocol like any other, anyone with access to Autonomous System level routers which peer with each other, or is on the inside of internetwork operations, can monitor BGP traffic and activity directly. And Monday afternoon Cloudflare was doing just that. In the show notes again, I have a strip, a diagram. The chart shows a 30-minute graph taken from 15:30 to 16:00 UTC, of Facebook-related BGP traffic. Normally there's nothing happening since routing tends to be boring, except when it's not. And there was nothing boring about routing yesterday.

The chart shows that starting at exactly 15:39, in the darker blue trace, some new routes were announced, and a bit later there was a similar volume of routes withdrawn. That's in the lighter blue. It may have been that that was deliberate because it looks like things were being moved around. But then, about a minute and a half later, we see a massive flurry of routes being withdrawn in the lighter blue, where the area under the curve dwarfs that of any new replacement announcements. Now, maybe it was supposed to be that way. The goal may have been to consolidate routes, which is much better for routing efficiency. In that case you would expect to see many more withdrawals than new announcements to replace them. Except that we also know that the result of this sudden flurry of activity was a catastrophe. So it really looks like a mistake.

Leo: There's actually a secondary post which I'll - I don't want to interrupt your flow. But at the end, Facebook has posted in much more detail about what happened. I think they now know what happened. But go ahead because I want you to speculate, and then you'll see you're actually pretty right-on.

Steve: Okay. And I wrote here, by now Facebook certainly knows precisely what happened. But based upon their public statement, once this had all been resolved, it doesn't appear as though we're going to get much clarity from them unless it eventually leaks out. But what they did say confirms our hypothesis from the evidence. So what they posted first is, they wrote: "To all the people and businesses around the world who depend upon us, we are sorry for the inconvenience caused by today's outage across our platforms. We've been working as hard as we can to restore access, and our systems are now back up and running. The underlying cause of this outage also impacted many of the internal tools and systems we use in our day-to-day operations, complicating our attempts to quickly diagnose and resolve the problem.

"Our engineering teams have learned that configuration changes on the backbone routers that coordinate network traffic between our data centers caused issues that interrupted this communication. This disruption to network traffic had a cascading effect on the way our data centers communicate, bringing our services to a halt." Now, of course that's a very generic way of saying what we just talked about. They said: "Our services are now back online, and we're actively working to fully return them to regular operations. We want to make clear at this time we believe the root cause of this outage was a faulty configuration change. We also have no evidence that user data was compromised as a result of this downtime," blah blah blah. And we apologize and so forth.

Okay. So what was most curious, and really sort of unbelievable while this was all happening, was that Facebook didn't quickly pop back up on the air. When I learned that they were down, I was thinking, okay, this won't take long. You know? If someone had entered a bad routing update, how difficult could it be to at least roll back the change? As it happens, while this was going on, I was already at work on this podcast, so I was able to participate in the drama with my followers on Twitter.

My first tweet read: "Facebook may have 'deplatformed' itself, along with Instagram and WhatsApp. Hope no one depends upon 'Login with Facebook!' Whoopsie! Somehow, the BGP entries for Facebook's DNS resolvers have been withdrawn from the Internet's routing tables. Insider? Attack? Who knows. Wow."

After a bit more digging I added: "Someone on the Facebook recovery effort has explained that a routine BGP update went wrong, which in turn locked out those with remote access who could reverse the mistake. Those who do have physical access do not have authorization on the servers. Catch-22." So anyway, so we were beginning to get some clarification at that point of the trouble. And apparently the unexpected loss of Internet routing had cut off their own engineers from the routers they needed to access in order to roll back the changes. Again, whoopsie.

Now, I know this is actually a very real concern. I manage GRC's network at Level 3 remotely, and I sometimes need to alter the operation of the equipment that manages GRC's border with the Internet. My own management traffic is crossing that same boundary as the one I'm managing, and I want my own management traffic to be crossing that same boundary since it's a security boundary, and my management traffic needs to be highly secured. But that also means that I need to be very careful not to edit some rule that locks me out. And it has happened, which necessitated a drive over to the physical plant to log onto the machine directly and correct a mistake.

During all this, a New York Times reporter tweeted: "Was just on the phone with someone who works for Facebook who described employees unable to enter buildings this morning to begin to evaluate extent of outage because their badges weren't working to access doors." So apparently everything at Facebook is IoT, baby, on the Internet and dependent upon some super-secure access control server somewhere else which cannot be reached during the routing outage.

After some additional digging, my next tweet was: "Reports are that Facebook employees cannot enter their headquarters because their badges don't work, and those inside are unable to enter various rooms because access is dependent upon obtaining authorization from remote Facebook authentication servers." Those who live by technology, dot dot dot.

And in light of all the recent news about Facebook's internal awareness that an obsession with Instagram postings may not be good for many of the youngsters who have flocked to it, I added a tongue-in-cheek tweet: "Meanwhile, there's been a noted global decrease in reports of teenage depression and poor self-image. Mental health is on the rise. But fear not, BGP is sure to be restored soon."

And finally, to top it all off, five and a half years ago, back in April of 2016, Facebook acquired the domain registrar RegistrarSEC.com. Ever since then, they've been their own domain registrar. Because why not? But what happens when Facebook's own domain registrar goes offline due to a routing outage? Thanks to the automated systems being used by some less clued-in registrars, which are continually searching the Internet for previously registered domains which appear to be expired, abandoned, or recently vacated, yesterday - and I kid you not - several different registrars were listing the domain Facebook.com as being available for sale.

Leo: I'll take it.

Steve: I grabbed a screenshot of one of those several for the show notes. It announced with an exclamation point that "Facebook.com" is for sale!

Leo: Wow.

Steve: Something went wrong, indeed.

Leo: Probably those are automated systems that monitor for this kind of thing, sniping these things.

Steve: Yup. Yup.

Leo: So Facebook today published more details about the outage.

Steve: Good.

Leo: And I think this actually sounds very credible as to what happened. This is from Santosh Janardhan. "Now that our platforms are up and running as usual after yesterday's outage, I thought it would be worth sharing a little more detail on what happened and why and, most importantly, how we're learning for it. The outage was triggered by a system that manages" - and you were right, it's load balancing - "our global backbone network capacity." Now, he does a lot of explanation about how DNS works and stuff, so I'll skip over that. I think we all know that.

"So when you open one of our apps, the app's request for data travels to the nearest data center, which then communicates directly over our backbone network to a larger data center. The data traffic between all these facilities is managed by routers." Of course. "In the extensive day-to-day work of maintaining this infrastructure, our engineers often need to take part of the backbone offline for maintenance perhaps to repair a fiber line, add more capacity, et cetera." Or updating the software on the router itself.

"This was the source of yesterday's outage. During one of these routine maintenance jobs, a command was issued with the intention to assess the availability of global backbone capacity." And you can probably understand better between the lines what kind of command that was. "It unintentionally took down all the connections in our backbone network, effectively disconnecting Facebook data centers globally." They took down their own backbone. "Our systems are designed to audit commands like these to prevent mistakes like this." I would hope so.

Steve: Yes.

Leo: "But a bug in that audit tool" - a bug - "prevented it from properly stopping the command. So this change caused a complete disconnection of our server connections between our data centers and the Internet. And then that caused a second issue" - this is the thing you saw and everybody else saw - "that made things worse. One of the jobs performed by our smaller facilities is to respond to DNS queries."

Oh. This is interesting, too. "To ensure reliable operation, our DNS servers disable BGP advertisements if they themselves cannot speak to the data centers, since this is an indication of an unhealthy network connection. Because the backbone was removed from the operation, these locations declared themselves unhealthy and withdrew the BGP advertisements." So they stopped saying, "We're here, we're routing." "The end result was our DNS servers became unreachable even though they were still operational." And then, well, you saw the consequence.

And of course the reason it took so long: "Our engineers worked to figure out what was happening and why. They faced two large obstacles. First, it was not possible to access our data centers through normal means because the networks were down. And second, the total loss of DNS broke many of the internal tools we'd normally use to investigate and resolve outages like this. So we sent engineers onsite." But obviously these facilities are highly secure, hard to get into. And even once you get into them, the hardware and routers are designed to be difficult to modify, even if you have physical access to them.

Steve: In other words, they're heavily protected.

Leo: As they should be. Right, yeah.

Steve: Yes. Yes.

Leo: So they were even protected from themselves, their own remediation efforts. So, you know, you've got to read between the lines of what exactly...

Steve: Yeah. He's putting a little more weight on DNS outage than I think is proper. As we saw, clients will wait a week before they get upset.

Leo: Yeah. Once the backbone is down, you don't have any connection back to the home office.

Steve: Yes. It was the fact that their routes got pulled. Whatever this thing was, I mean, lord knows what kind of command would have...

Leo: Well, it sounded like a diagnostic.

Steve: We're still not having the detail that would satisfy.

Leo: Yeah, what tool? What's the name of the tool and all that? They may or may not reveal that. Again, for security reasons.

Steve: They have no obligation to. They're like, whoops, we're sorry. Everything's good. Move along.

Leo: So there were two problems, one that they were testing the reliability of the backbone and inadvertently issued a command that brought the backbone down. And then the tool that was to protect that...

Steve: Well, Leo, that made the reliability test really easy.

Leo: It's off.

Steve: Zero.

Leo: Zero.

Steve: Zero.

Leo: But the tool that was to protect from this kind - and this is always my question when you get BGP routing tables screwed up, is wouldn't you test that? Wouldn't you have some software that could - and we even advertise software that will assess the change and see what will happen without actually implementing it. Apparently they had a tool like that, that had a bug and failed. So it's interesting.

Steve: Yeah. It's still, you know, it's murky enough as not to be...

Leo: There's more, more to this story.

Steve: Yes.

Leo: But it kind of makes sense. And you could see how it's a cascade of failures. We knew that must be; right? And it was just, you know, it was a normal maintenance procedure that went wrong. You can imagine, though, the engineer that pushed that test out, and then all of a sudden all routes...

Steve: The world ends.

Leo: All the backbones are down. Oh, my god. The freak-out that must have been.

Steve: Yeah.

Leo: Somebody's saying, well, they had a thing that said, "Are you sure you want to run this command? Y or N?"

Steve: Are you really, really sure? Double all the routes.

Leo: Are you sure? "Our engineers often need to take part of the backbone offline for maintenance." This was the source of yesterday's outage. "A command was issued with the intention to assess the availability of global backbone capacity."

Steve: Yeah.

Leo: What could that be?

Steve: Yeah. I mean, for example, if you were to unroute a network segment, then you might run traffic through it in order to check that network segment. So you wouldn't have traffic going across the segment while you were using it, you know, while it was under test. So that would mean that you would put in some commands to route around that segment that was down for testing. And he put in a *.* when he didn't intend to.

Leo: I do know that there are now three openings on the same post for software engineer routing and data path.

Steve: You know, and the fact, is with Facebook being as big as they are?

Leo: Oh, man.

Steve: They have to be a little obtuse, I mean, for security reasons. They can't really tell us what it was because if they gave us a recipe for taking Facebook off the air, then that's information that nobody should have.

Leo: Yeah. Nobody needs to know those tools. Many of them are probably custom-built inside Facebook.

Steve: Actually, Facebook built their own entire stack.

Leo: That's right.

Steve: All of their software, all of their network software is from scratch.

Leo: Right. As is their hardware. I mean, they created their own little modules and everything. The thing that always blows my mind, whether it's Facebook or Google or Microsoft, 20-30 years ago we would never have contemplated the notion that there may be a network operation that serves three billion people daily.

Steve: I know.

Leo: Mind-boggling. To go from zero to 60 over a decade, it's kind of incomprehensible. So the fact that it works at all and is frankly as reliable as it is normally is kind of impressive. When you go to Facebook, it pops up. Those images load almost instantly. It's amazing.

Steve: And I feel the same way about the human body. Someone says, oh, something's broke and doesn't work, I say, "It's amazing it works at all. Are you kidding me?"

Leo: Well, actually, and I've been meaning to tell you, given you're Mr. Quantified Self, you're I'm sure aware of the fact you can do these full-body MRIs.

Steve: Yup.

Leo: Lisa and I recently did this from a company Kevin Rose had recommended to us, they're out of British Columbia, called Prenoveau.

Steve: Yup.

Leo: Are you familiar with Prenoveau? Have you heard that name?

Steve: Well, I heard you talk about it.

Leo: Oh, you heard us talk about it.

Steve: And I immediately jumped online to see. And unfortunately the closest one is Silicon Valley.

Leo: That's what I was saying. Next time you come up to Redwood City, let us know. It's expensive. And there are two negatives to this. One is 2,500 bucks. And they take - by the way, it's an hour procedure because they take many more pictures than a normal MRI would. They're getting exposures from every angle and in many different ways. So it's expensive, and it's a long procedure.

The other downside to it, which doctors are always bringing up, is the human body is not perfect. We know that on the outside we're not perfect. Well, you know, you're not perfect on the inside, either. And you're going to find out about all sorts of things you didn't know. And probably most of those you'd live your entire life without knowing or caring, and it wouldn't make any difference. And the trick with this is to know, well, that's something, that's actionable. Or that's just, you know, you're not perfect. I have apparently an atrophied kidney. Could have been congenital. I will never know because this is the first time I've done this.

Steve: Right.

Leo: And that's why you have two; right? There's no blood test or anything that's shown any problems. Should I be concerned about it? Well, I am, kind of. I mean, that's not a good thing. But on the other hand, it probably has been there my whole life, and who knows, and who cares? Right?

Steve: Well, what I like about that is it being an MRI rather than a CAT scan.

Leo: It's safe.

Steve: It's not, yes, it's non-radiation, so you're not increasing your radiation load. So it has that going for it. And frankly, if there was one in my neighborhood, I'd seriously consider it. And I would keep Lorrie away from it because oh, my god...

Leo: You mean from your scan. You wouldn't want her to see your scan.

Steve: She worries about whether the sun is going to come up in the morning.

Leo: Oh, I know, I know, I know.

Steve: And it's like, oh, that would not be a good - that would not be a good thing.

Leo: Well, Lisa and I are very glad we did it. I won't go into the details. I've already shared too much. But we're glad we did it. And we will now, now that we have a baseline, I think we will do it every 18 months.

Steve: Yeah.

Leo: Because now you know. I wish I'd done this when I was 25 because then I would know, well, this is where you're starting. And it's the changes you want to look for. So I think it's very interesting. And I know you're Mr. Quantified Self, so you should do it.

Steve: Oh, but I do carotid ultrasounds every couple of years.

Leo: That's right, I remember that, yeah. Same reason.

Steve: Yes, because it's a very good proxy for your cardiac arteries. And so if you see excessive plaque building up, it's a clue to not wait until a piece of it flakes off or a major artery clogs all the way or something.

Leo: I'm happy to say, and I don't know if this would do as well as that, but no heart disease, no cancer. So, I mean, that's important. I'm glad. That's the main reason I did it.

Steve: I think that's very cool.

Leo: Yeah. Somebody I'll show you my pictures. Steve Gibson is at GRC.com. That is the place to get his unique versions of this show. He has a 16Kb audio version which sounds like Thomas Edison recorded it in his lab. But it is small. "Mary had a little lamb." But it's small, and that's the reason we do that, so those of you who have limited bandwidth or metered bandwidth, that's a good way to get the audio. He also has a transcript, which is even smaller. That's really, really great. You can read along as you listen, or use it to search for parts of the show. It's on Google, so it's indexed. So that's really great. Elaine Farris writes those for us. He also has 64Kb audio.

All of this is at GRC.com. And while you're there, by all means this is a good time to pick up a copy of the world's best mass storage recovery and maintenance utility, SpinRite, currently 6.0. 6.1 is, as you heard, in very active development. You'll get to participate in that, and you'll get a copy of 6.1 automatically for free if you buy 6.0 now. So GRC.com. Lots of other great free resources there, including his DNS Benchmark, which I used the other day to figure out who I should use for DNS, that kind of thing.

We have copies of the show at our website, TWiT.tv/sn. There's a YouTube channel. You can go there. We have video there and on our site. And of course you can subscribe. In fact, I encourage you to subscribe in your favorite podcast client. That way you don't even have to think about it. You just have it whenever you're in the mood to listen. If you do, if your client does allow reviews, please leave us a five-star review just so others know about this really phenomenal resource.

Steve will be the guest of an AMA on Friday as part of our Club TWiT. I'm very excited about this. This is brilliant. Ant Pruitt, who is now our community manager, we said, "Ant, we need to - we really want to grow the club." He said, "I'm going to help, and we're going to bring in Steve because I know people want to ask him questions." So I'm surprised, I thought you - I didn't think you'd do that. That's good.

Steve: I'll support the Club.

Leo: Thank you. We do this show every Tuesday right after MacBreak Weekly, usually between 1:30 and 2:00 p.m. Pacific. That's 4:30 and 5:00 p.m. Eastern Time, 20:30 UTC. The livestreams are at TWiT.tv/live. There's audio and video there. The chat room is irc.twit.tv. The Discord is also open for chat during all of our shows. And we even have forums. The TWiT community is at twit.community, and the Mastodon instance, our social kind of Twitter alike, our Fediverse, is at twit.social - twit.community, twit.social. You're welcome to join up there. Those are free, as well. Steve, have a great week. I'll be watching in the "Invasion," too. Cross my fingers.

Steve: Yes, yes.

Leo: We'll see you next time on Security Now!.

Steve: Okay, buddy. Bye.