SECURITY NOW!

Transcript of Episode #847

Bogons Begone!

Description: This week we'll note that the new Edge browser's Super Duper Secure Mode has been deployed and can be enabled by security-conscious users. We also have more than one third 37% of the world's smartphones vulnerable to audio monitoring and recording flaws in their MediaTek firmware. We have an important reminder about clicking links in email and wonder how that can still be a problem, and the entirely predictable evolution of a Windows zero-day vulnerability which is latent no longer. We have some interesting closing-the-loop feedback from our terrific listeners, and a sci-fi book update. Then we take another and much broader look at the recent efforts to clean up IPv4, but this time from the perspective of those working to do so.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-847.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-847-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. The zero-day Windows exploit that we just found out about. Schrodinger's cat pays a visit. We'll also find out why 37% of the world's smartphones are vulnerable and how many of those are going to get fixed. And then Steve is going to explain something we talked about last week, and I think he changes his mind by the end of the show. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 847, recorded Tuesday, November 30th, 2021: Bogons Begone!

It's time for Security Now!. Yay, you've been waiting all week. You're very patient. But here we are, Tuesday again. And here he is, fresh and ready to go, straight from the keyboard where he's been typing, typing, typing, Mr. Steven "Tiberius" Gibson. Hello, Steve.

Steve Gibson: I actually have been. Hello, Leo.

Leo: I know you have.

Steve: I worked all through the Thanksgiving holiday and weekend until something happened that really almost never happens. I literally hit the wall on Sunday around 2:00 or 3:00 p.m., and I posted a note to the SpinRite group, and I said, "Okay, I can't work anymore."

Leo: Wow.

Steve: And so, yeah.

Leo: It's unheard of.

Steve: But I've been so interested in what's going on. I talk about it a little bit later in the podcast. So we have a really fun podcast, I think, 847 for this last day, the final day of November, Bogons Begone! Yes, and everyone will understand by the time we're done what bogons are to be gone.

Leo: Oh, good.

Steve: Maybe. But we've got a lot to talk about. We're going to note that the new Edge browser's unbelievably named Super Duper Secure Mode...

Leo: Yes.

Steve: ...because Microsoft certainly wouldn't want to saddle us with anything less than super duper security, has been deployed kind of quietly. But we can all enable it. It's not on by default, so our listeners are going to want to know all about that. We also have more than one third, 37%, of the world's smartphones all vulnerable to audio monitoring and recording flaws courtesy of their MediaTek firmware in their MediaTek audio processing DSP. We're going to talk about that. We've got an important reminder about clicking links in email and wonder how that can still be a problem - it still is - and the entirely predictable evolution of a Windows zero-day vulnerability which is latent no longer, unfortunately. We have some interesting closing-the-loop feedback from our terrific listeners. I've got a sci-fi book update.

Then we're going to take another and much broader look at the recent efforts to clean up IPv4, but this time from the perspective of those who are working to do so. And I understand their position. So we talked last week about this crazy idea as a consequence of the IETF's proposal to claw back, essentially, most of the localnet 127 network. It turns out that's just the tip of the iceberg. So I think a really fun podcast for our listeners.

Leo: I can't wait. I'm excited. As always, I look forward to this all week long.

Steve: Oh, and a picture. We've got a picture, Leo.

Leo: A picture, we've got a picture. And now, the kitty-cat of the week. I never thought you'd be doing cat pictures, Steve.

Steve: Only these particular cat pictures when they involve the nature of the universe. I can't explain this. It looks like a lost cat flyer, you know, Xeroxed, that used some masking tape and stuck up on a telephone pole somewhere. The confusing thing is - and we have of course the picture of the cat, and it explains: "Please return dead and alive to Erwin Schrodinger. Contact shrodingercatman@gmail.com."

Leo: Which is probably a real address. That's hysterical.

Steve: That is pretty good. So anyway, as you said, maybe this was posted in a college town.

Leo: Has to be.

Steve: Over by the physics department.

Leo: Right.

Steve: Yes. So we don't want the cat dead or alive.

Leo: Both. Both.

Steve: We want it in both states simultaneously, unknown until it is observed whether the cat is in fact alive or dead. Either way, or actually both ways, we want it returned.

Okay. So Super Duper Secure Mode. We previously discussed the experiment Microsoft was conducting with their Edge branch of the Chromium browser, in recognition that a disproportionate percentage of security troubles arise from the most extreme measures being used to push browser performance to the limits, and the recognition that underlying system processor performance has advanced so far recently that pushing the browser so hard that it breaks may be producing diminishing and even negative returns in terms of security. So Microsoft began experimenting with [fanfare] Super Duper Secure Mode for Edge which deliberately pulls back on the most historically troublesome performance optimizations in favor of improved security.

Of course we're talking about it today because, without any ballyhoo, Microsoft recently quietly added Super Duper Secure Mode to Edge. We all have it already. It appeared in 96.0.1054.29. And I noticed I'm already at .34 or something. It's currently disabled by default. To enable it, as I did, you'll need to open Edge and then put "edge://settings/privacy" into your URL. That'll take you to the proper page. Then you've got to scroll way down to the Security section.

At the bottom of that section on the right is a switch which you need to flip to the on position because mine was off. And then you can choose between "Balanced" or "Strict," where Balanced is the default. I of course switched to Strict. Under Balanced it says "Adds security mitigations for sites you don't visit frequently. Most sites work as expected. Blocks security threats." If, however, you choose Strict, as I did, it "Adds security mitigations for all sites. Parts of sites might not work." And I really don't know why that is, but okay. "Blocks security threats."

Leo: Because this isn't disabling something like NoScript would do, disabling JavaScript. That would break sites.

Steve: Oh, that's, like, goodnight. That's why we all finally had to give up on NoScript was it, like, I had to turn it on so often that it's like, what's the point any longer; you know?

Leo: Right, right. But this does disable, am I correct in saying Just In Time JavaScript compilation? Is that...

Steve: Exactly.

Leo: That's the thing that they said was a problem.

Steve: Yes. Just In time compilation, that's the JIT, J-I-T, from the V8 processing pipeline and, for example, it also enables Intel's Control-flow Enhancement Technology (CET) which is a hardware-based exploit mitigation that provides enhanced security. Now, based upon evidence from historical exploits, it's believed that this will significantly reduce the browser's attack surface. Microsoft describes Super Duper Secure Mode as "a browsing mode in Microsoft Edge where the security of your browser takes priority, providing you an extra layer of protection when browsing the web." Oh, and the one last thing, the other option there is "exceptions." You can, if you are running in Strict, as I am now, and something did not work, you could add that as an exception.

But really, in fairness, Balanced is probably the right thing for most people. I'm hoping that once they gain confidence in this, which is presumably the reason that little switch that you had to first turn on to enable any of this, once they gain confidence, maybe they'll flip it on by default. Because it does make sense, if in fact Strict mode actually breaks some things - and again, why would it if it's just turning off optimizations; okay?

But the idea of watching where you go and, for example, obviously, you know, after a while Google is going to go into open mode, and the things you do most often will be automatically added to the, okay, we trust this, and it makes sense for a site you've never, that Edge has never before seen you visit to be automatically in Strict mode. Maybe you'll never come back. But if it's somewhere you've never been before, let's keep the shields up until we have some reason to trust the site. So I really like the logic behind Balanced, but I imagine our listeners want to be in Strict mode.

And, you know, I'm just not seeing a performance problem on any pages I visit. Our machines are so blazing fast now that I like this concept. If in fact half, I think it is, no, it's 45%, 45% of all security vulnerabilities that were found in the V8 JavaScript and WebAssembly engine were related to JIT, to the Just In Time compiler. Half of them, nearly. So turn it off. Unless you have something where you absolutely had that performance, turn it off. And if you do have sites like, I don't know, web-based gaming or something where you notice not having JIT creates a lag, put that site in as an exception, and it will be fully trusted and run with full optimization. But otherwise...

Leo: The most difference it would make would just be performance; right? I mean, it's...

Steve: Yes.

Leo: Yeah, it probably shouldn't break anything.

Steve: Yes. And it'll be interesting to see...

Leo: Just putting that in.

Steve: ...and gain more experience, yes, if it actually - I would ask our listeners who often send me notes, if you run Strict and something breaks, shoot me a tweet. I would love to know what this broke.

Leo: Right.

Steve: Because this idea of half of the bad problems being shut down by backing off on crazy performance optimization, I would say it's time for that.

Leo: Yes.

Steve: So bravo. Moving forward, they plan to include support for Arbitrary Code Guard in this Super Duper Secure Mode. ACG, they explain, is another security mitigation that would block attackers from loading malicious code into memory. Hmm. That sounds like a good thing. I'll take two, please.

The Android and macOS editions of Edge will soon also be obtaining these new vulnerability mitigations. And you put the screen shot onscreen, Leo, that shows that I immediately flipped the switch, set Strict mode, and I didn't even both putting GRC as an exception because I don't have any JavaScript on my site. So there's nothing there to optimize.

Leo: Just In Time, it's never a time.

Steve: And I don't think, like, we need the performance at the expense of security. And so I think this was really a smart thing. This is probably the best thing Microsoft has given back to Chromium, and it would be nice to see if it ends up getting adopted by the other browsers.

Meanwhile, we learned that 37% of the world's smartphones are vulnerable. Chips by MediaTek are installed in roughly 37% of the world's smartphones, and Checkpoint Research recently went to all the trouble of reverse engineering the firmware of those proprietary chips. And this is another, like, we've talked about this before. I take my hat off to these companies that are willing to go to such extreme, to invest so much time and trouble in reverse engineering proprietary stuff that a huge percentage of the world is using. And it feels wrong to me that the bar has been set so high, the idea that some MediaTek company can create a proprietary DSP and say, oh, yeah, don't worry, it's secure, trust us, and 37% of the smartphones in the world adopt it, and then it turns out to be a vector that is seriously putting all of those devices at risk.

And the only way we know about this is that a Robin Hood security company comes along and says, okay, well, shoot. This is going to be hard, but we're going to do it. They published a highly detailed technical report which showed that malicious apps installed on a device would be able to interact with the MediaTek-based audio driver. Okay. Apps do interact with audio drivers. But such apps could send maliciously crafted messages via that audio driver that all apps have access to, to the MediaTek firmware to gain control over the driver and steal any audio flow going through the device, turning it into a spy device for more than one-third of the world's smartphones. And since the MediaTek subsystem is deep in the system, exploitation of the vulnerability also allows audio from phone calls, WhatsApp calls, browser videos, and video players to be recorded. Basically the audio in the device.

The fact that MediaTek chips are installed on roughly 37% of the world's smartphones means that this creates a massive attack surface for any malicious app and malware creator. Devices from Xiaomi, Oppo, Realme, and Vivo are all known to use MediaTek chipsets. And that's probably a fraction of the total. Three issues were patched last month in October, and a fourth issue will be patched next month.

Checkpoint explained that the MediaTek chips contain a special AI processing unit - doesn't everything now - the APU, and audio digital signal processor (DSP) to improve media performance and reduce CPU usage. But both the APU and the DSP use custom microprocessor architectures, which makes the MediaTek DSP a unique and challenging target, as I've said, for security researchers; right? I mean, it's not just - it's not an ARM processor where they can dump the firmware and run it through a decompiler and check it out. No. They had to, like, reverse engineer the processor. But they did. They grew curious about the degree to which the MediaTek DSP could be used as an attack vector for threat actors. So they managed to reverse engineer the audio processor and discovered a handful of security flaws.

The flaws can be updated, this is the good news, with firmware. So keeping Android devices current continues to be as important as ever. And while the chips remained a proprietary mystery, the likelihood of their exploitation remained low. Unfortunately, the flipside of us finding out that there's a big problem with them is that Checkpoint's write-up is necessarily extremely detailed. I mean, you know, if they're going to go to all this work to do all this, then they want some credit for it.

Unfortunately, this means that there is now a readily available roadmap to any technically competent miscreant who might want it, and it's got topics like "Classic heap overflow in the AUDIO_DSP_TASK_MSGA2DSHAREMEM message handler," don't you know. We also have "Classic heap overflow in the init_share_mem_core function," and an "Improper validation of an array index in the audio_dsp_hw_open_op function." In other words, Checkpoint has documented the problems that nobody knew, not only about, but how to exploit, and they've exploited them, and that's all out in the public now.

Leo: [Grumbling]

Steve: I know, yes. And the problem is that off-brand or unmaintained smartphones are far less likely to ever obtain updates to their MediaTek firmware. They just won't be available. Their manufacturers made the phone, sold the phone, and moved on to something else. So those original vendors won't ever bother, even if their users wanted to update. So on top of the already overwhelming number of known and never-to-be-patched vulnerabilities that are still accumulating from the past, Checkpoint has just carefully uncovered and documented another handful. Nice that we know, but really it puts 37% of phones from non-mainstream manufacturers at further risk. And in this case, turning them into spy devices. And you've got to know that there are major state-level actors that are looking at this thinking, oh, let's add this to our war chest. There may be some dissident somewhere that's got an old Xiaomi phone that they're careful not to load anything into. But it turns out they've got a bug that they can't get rid of. Boy. You know?

And again, Leo, as you and I have been saying now, stick with the mainstream - Google Pixel, Samsung, Apple iOS devices. Or obsolete your phone and stay current as often as you can. Checkpoint, I mean, MediaTek has responded. They were good. They've patched their firmware. They'll be providing the firmware to the OEMs that are using their chips to be incorporated into new devices. But we don't have a channel now for moving backward in time at this point and reliably fixing the devices which, as we know, are handheld computers. So it's a little worrisome.

Okay. We have the RAT dispenser. It's probably worth taking just a moment to reinforce the need to never, and I mean really never, click to open an attachment received in an email, any email, even if it's from your mom. Okay. Now, "Cybersecurity experts from HP" said they discovered a new strain of JavaScript malware that criminals are using as a way to infect systems and then deploy dangerous remote access trojans. In other words, the remote access trojan, RAT, R-A-T, thus the RATDispenser. But I put, Leo, "Cybersecurity experts from HP" in quotes.

Leo: Why?

Steve: Because, well, try going to, and I'm serious, threatresearch.ext.hp.com.

Leo: Okay. The HP Wolf Security Blog.

Steve: And you got there somehow. They must have just fixed this.

Leo: Oh, it was broken?

Steve: Oh, yes. Look in the show notes. I've got the TLS certificate that both Firefox and Google were displaying this morning. I had to fight my way through in order to get there.

Leo: Keep your sites up to date, kids.

Steve: Wow. Let me see. Why is it working for you?

Leo: I mean, they may have just fixed their certificate. Let me look at their certificate here.

Steve: Yeah, it came right up now.

Leo: Yeah, they fixed it.

Steve: That's annoying. Someone must have told them, probably because they were in the news. But this morning I got both Firefox and Chrome refused. And what I couldn't understand was that the certificate that they were serving expired on March 18th of 2021.

Leo: Yeah, so this is from March 15th, 2021. So they got a new one, but maybe they didn't apply it. And it goes through 2022. Maybe they forgot or, I don't know.

Steve: Right. So they got that certificate.

Leo: They probably didn't install it.

Steve: On the same cycle. But Leo.

Leo: But the site should have been down since last March.

Steve: Correct. It could not have been offline for 10 months.

Leo: No.

Steve: So something weird...

Leo: It was a, what do they call that, a regression?

Steve: Yes, we do call it a regression. It was definitely that. Okay. Anyway, HP explains, because they do have good researchers, they said...

Leo: It's embarrassing if that is.

Steve: Oh, god, yeah. I mean, and it's why I took a screen shot of it. It's like, what?

Leo: This is embarrassing.

Steve: And then I thought maybe Firefox. So I went over to Chrome, and Chrome wouldn't show it either.

Leo: Whoopsie.

Steve: Yeah. Yeah. Anyway, "Threat actors," they said, "are always looking for stealthy ways of delivering malware without being detected. In this article we describe how attackers are using an evasive JavaScript loader that we call RATDispenser to distribute remote access Trojans and information stealers. With only" - and this is what's really amazing - "an 11% detection rate" - meaning one out of 10 gets flagged, one out of 10 instances - "an 11% detection rate, RATDispenser appears," they wrote, "to be effective at evading security controls and delivering malware. In total, we identified eight malware families distributed using this malware during 2021. All the payloads were RATs (Remote Access Trojans) designed to steal information and give attackers control over victim devices.

"As with most attacks involving JavaScript malware, RATDispenser is used to gain an initial foothold on a system before launching secondary malware that establishes control over the compromised device. Interestingly," they said, "our investigation found that RATDispenser is predominantly being used as a dropper in 94% of samples analyzed, meaning the malware doesn't communicate over the network to deliver a malicious payload." In other words, it incorporates it. "The variety in malware families, many of which can be purchased or downloaded freely from underground marketplaces, and the preference of malware operators to drop their payloads, suggest that the authors of RATDispenser may be operating under a malware-as-a-service business model." Wonderful.

Okay. So the infection chain begins with a user receiving an email containing - and I read this, and I just, like, really? This is still happening? - a malicious attachment. It's the classic double file extension, something like "OrderInformation.txt.js" which we've all known about for how long? Yet it still works. It still isn't being displayed properly. It's still being handled by our email agents. So the unwitting user simply needs to double-click the file to run the malware. What I want to know is how is it that any of today's email clients will run such a file with a simple double-click? This morning I just had to promise my first-born child to view this report from HP because their once-legitimate TLS certificate had expired. I had this, like, "WARNING! WARNING! Do not trust this site!"

Leo: Oh, god.

Steve: "Go back now." And it's like, I look at the URL, no, it's right. And it's like, you know, and I pressed Advanced, and it said, "Are you sure you're advanced?" Yes. Okay. Do you have any children? Will you? Are you planning to?

Leo: Can we have them?

Steve: Right, oh, my god. But no, .txt.js, oh, wonderful, run the code.

Leo: Oh, gosh.

Steve: You know, what is happening?

Leo: Yeah.

Steve: We clearly have our priorities backwards.

Leo: Yeah. Oh, my god.

Steve: So HP notes that network defenders can prevent infection at the enterprise level by blocking executable file attachment file types from passing through their email gateways, for example, JavaScript or VBScript. What a concept. Defenders, they said, can also interrupt the execution of the malware by changing the default file handler for JavaScript files, only allowing digitally signed scripts to run, or disabling Windows Script Host. Okay. But I ask why any of that is even necessary. It won't protect anyone outside of those enterprise boundaries; right? The standard default is you click on something .js, we don't care where it came from, we don't care how it got into your system, you want it, you got it. It's crazy.

When the malware runs, the JavaScript decodes itself at runtime because of course it's all heavily obfuscated. It's just a bunch of /hex codes behind some evals. And that writes a VBScript file to the %TEMP% folder using cmd.exe. To do this, the cmd.exe process is passed a long, chained argument, parts of which are written to the new file using the echo function. Then the VBScript file runs to download - oh, it is downloading the malware payload from somewhere. If it was downloaded successfully, it's executed, and the VBScript file is deleted. So it's like, oh, we got what we wanted. Now delete this. Again, how can it be that you're not allowed to visit a site whose certificate has expired? Yet click on a link in email, run some code, no problem.

The initial JavaScript downloader is obfuscated and contains several eval functions, as I mentioned. One of the eval calls is a function that returns a long string, which is decoded by another function. And it's clearly effective since only, as I said, about one out of every 10 instances, well, 11%, that's one out of nine, are now being detected after many months of successful exploitation.

Over the past three months, HP said the malware had been used to drop at least eight different RAT strains, such as STTRAT; WSH, we know what that stands for, RAT; AdWind; Formbook; Remcos; Panda Stealer; GuLoader; and Ratty.

Leo: Love the names, wow.

Steve: Yeah, the Ratty RAT. As I started out saying, it's worth just refreshing the strength of the prohibition against ever clicking on anything that is received in email. Really. I mean, these guys are clever. They're going out of their way to avoid the protections built into our systems. It's dispiriting that it is still possible to do this today. But it is.

Okay. So we have an entirely predictable zero-day Windows exploit. Right on...

Leo: Of course. I'm sorry. I shouldn't laugh. I shouldn't laugh.

Steve: I know. I know. It's just so sad because, again, our listeners will be right there with us. Right on schedule, Cisco's Talos group discovered the active exploitation of a zero-day elevation of privilege vulnerability in Microsoft Windows Installer. This vulnerability allows an attacker with a limited user account to elevate their privileges to become an admin. The vulnerability affects every version of Microsoft Windows, including fully patched Windows 11 and Server 2022. Talos detected malware in the wild taking advantage of this vulnerability.

What was entirely predictable about this? We've been tracking this one for some time. Microsoft was first informed of the fundamental underlying problem by security researcher Abdelhamid Naceri, who discovered this elevation of privilege vulnerability and worked with Microsoft to address it. However, when Naceri examined Microsoft's supposed patch for this, following this month's Patch Tuesday, he discovered, oh, what do you know, Microsoft had merely patched against the proof of concept that he had provided, rather than addressing and repairing the underlying flaw.

To demonstrate that, Naceri then published proof-of-concept exploit code on GitHub on November 22nd, eight days ago, which works despite the fixes implemented by Microsoft because Microsoft didn't fix the problem. They just broke his proof of concept. The code Naceri released leverages the Discretionary Access Control List, the so-called DACL, for Microsoft Edge Elevation Service to replace any executable file on the system with an MSI file, which is a Microsoft installer, a setup installer file, allowing an attacker to run code as an administrator to gain full control over the compromised system, including the ability to download additional software, modify, delete, or exfiltrate sensitive information stored in the machine.

Independent security researcher Kevin Beaumont, who tweets as Gossi the Dog, tweeted: "Can confirm this works, local priv esc. Tested on Windows 10 21H2 and Windows 11. The prior patch MS issued didn't fix the issue properly." Naceri noted that the latest variant of this 2021-41379, which was the CVE assigned to this, is more powerful than the original one, and that the best course of action would be to wait for Microsoft to release a security patch for the problem due to the complexity of the vulnerability. Apparently this is not something that even the 0patch guys can offer a quick fix for because he says there's a great likelihood of breaking something.

So what do we have? A security researcher responsibly and privately reports a serious problem to Microsoft, including a proof-of-concept demonstration. Microsoft responds, not by fixing the problem, but by breaking the security researcher's proof of concept, claiming that the problem has been fixed. Annoyed, the security researcher then publicly posts another proof of concept to demonstrate that Microsoft actually fixed nothing. Somewhere, this is seen as good news as malware authors jump on this now public and well-documented unpatched vulnerability in Windows, using it to obtain admin rights on Windows machines.

We're all now living with the consequences of Microsoft's deliberate de-emphasis of Windows' pre-release testing, which has been much talked about here and over on Windows Weekly. Unfortunately, it appears that post-release vulnerability patching has also been de-emphasized. Wow. And Cisco found this being used. Maybe we'll get an emergency fix. Maybe we wait till December's Patch Tuesday. Unfortunately, this being the 30th and tomorrow being the 1st, December is one of those, of 2021, one of those months where the Patch Tuesday is halfway through the month, right, it's on the 14th. So we'll be waiting as long as we possibly could be.

Okay. Not much else of interest happened. And as I said, I'm going to talk a lot about the begoneness of bogons shortly. I did want to provide those listeners who may not have been tracking the Frontiers saga, as I certainly have been, with an update. The first of 15 next books in Ryk Brown's third 15-book story arc launched on Thanksgiving. Actually it came out on Wednesday of last week, the day before Thanksgiving, titled "Fringe Worlds," that is, this sequence will be "Fringe Worlds."

Ryk, as I mentioned before, spells his name R-Y-K, is no fan of Kindle Unlimited, feeling that it doesn't fairly reward authors for their work. In his announcement email he said that the book would not be on Kindle Unlimited, but would be for sale for $3 for at least a few months, after which he would then put it on Kindle Unlimited. And I would have happily paid $3 for all the hours of pleasant relaxation I obtain from his writing. But I just checked, and there it was on Kindle Unlimited. It's kind of hard to find. So I put a link to it in the show notes for anybody who has been following along and reading the first 30 books, as I have, of the Frontiers saga.

I'm at this moment finishing up my complete re-read of Ryk's first 30 books. I'm on the last one of the first 30. But then I plan to finally see what the Bobiverse series is all about. Our listeners are huge fans of this Bobiverse. So that'll be what I do next. However, I have to confess that my recent reading progress has been really retarded because I have truly become fully engaged in the work on SpinRite. I go to sleep thinking about it, I awaken thinking about it, and I'm thinking about it right now.

Leo: Good man. Everybody's very happy to hear that.

Steve: I know.

Leo: That's excellent.

Steve: Yeah, that's all I want to do because I'm becoming very excited about what it is becoming. I won't go into detail that I did go into detail in my posting to the group because in this last week I ended up developing, and I'm going to try not to go into detail, a heuristic system for statistically determining which interrupt request lines specific controllers and adapters were using. After we learned that they were lying and that I couldn't rely on what they were saying, it was necessary to get, you know, everybody else would say AI. Well, it's not AI. But it works, apparently.

So anyway, it's a big challenge. I'm having a ball. But I guess I'm becoming excited about what it's becoming. For one thing, it's guaranteeing me highly engaging full employment for the next several years as I publish successively more capable versions. When I watch its new benchmarks run, the non-uniformity of performance that is very clear as the numbers flash on the screen across both spinning and solid-state memory surfaces is, well, it's bracing. I mean, most people probably won't appreciate what that means. But if you're reading successive 64, wait, no, 32MB blocks from a spinning hard disk, and they are end to end, they ought to have the same timing or be slowly slowing down as you move gradually toward the center, like toward the center of the disk. That's not what I see. I see clear timing changes which mean that there's a problem there that the drive had to pause and work on for a while.

And what really surprised us was when we did the ReadSpeed Benchmark, and we saw that our solid-state RAM was not behaving like we would expect solid-state RAM. It's behaving like a bunch of leaky buckets. And in fact, as we know, that's exactly what it is. So I cannot wait to get to the point where SpinRite will have the ability to zero in on spots that it has discovered are reluctant to be read. In the case of solid-state memory, what's happening is the electrostatic charge of individual bits are leaking, losing their certainty, thus requiring more work and time to be read. SpinRite will be able to detect that and selectively rewrite just those leaky trouble spots to recharge them before any data is lost. And of course if there's any actual media damage, SpinRite will demonstrate that to the media's controller, enabling it to relocate the data to keep it safe.

And there's a lot of work yet to be done before we'll be there. I love the work. I need to get version 6.1 out to satisfy the immediate needs of everyone who has it while I keep working. Then SpinRite needs to be moved away from DOS over to the OnTime RTOS-32 kernel for operation on either BIOS or UEFI. Then native drivers for USB and NVMe need to be added to that environment. But all of the work I've been doing basically rewriting SpinRite is in preparation for that. Those will just be drop-in now, thanks to the fact that I've got this object-oriented I/O system which is now up and running and is proven. And at that point, when that's done, I'll be at a place where we can finally start to develop an entirely new media surface analysis system. So, yeah. If I sound excited, it's true.

Leo: Good. We're excited, too. That's great.

Steve: Yeah. And with any luck there'll still be a podcast here. So I'm going as fast as I can.

Leo: Take your time. It's okay. Just don't get distracted. Focus, focus, focus.

Steve: Well, I did, literally, I saw, I induced a problem on Sunday where I forced ATA drives to be recognized as IDE in order to verify a code path that wouldn't otherwise be used, but could be. And the tricky part of SpinRite which figures out which BIOS identifiers are connected to which hardware drives, it failed to - it's called the associator because it associates them. We need to know which items have already been found for direct SpinRite access and which ones will still have to go through the BIOS. If I don't associate them, then they'll show up as both a BIOS-accessible drive and as something that SpinRite can access, and that'll just be confusing to users.

So SpinRite is maintaining the same ease of use it's always had, and I'm taking responsibility for making that, for holding onto that ease of use. Anyway, when I forced ATA drives to be recognized as IDE, the associator didn't function. And I looked at it, and I just thought, okay. I'm out. I'm done. I've hit the wall. I went home early, at like 3:00. And Lorrie said, "Honey, you're home early." I said, "Yeah, I ran out of steam."

Leo: Burned out. That's fine.

Steve: But yesterday morning I sat down, looked at the problem, and solved it.

Leo: See? Just one good night's sleep.

Steve: Before I started working on the podcast.

Leo: I'm curious because I don't do any real programming, but I like these programming problems.

Steve: Well, you're about to do the whole programming...

Leo: The Advent of Code, yeah, yeah.

Steve: The Advent of Code; right.

Leo: So what I've noticed - and as a warm-up I was just doing a previous year's for the last few days.

Steve: Cool.

Leo: And what I noticed is if I can't - I'm looking at something and I don't know what's wrong, or I can't think of how to solve it, if I go to bed, I will think about it, it processes, and I'll wake up and go, okay. So you're saying that's what happened, kind of?

Steve: Yeah.

Leo: Or just getting some rest.

Steve: No, but I do know that phenomenon. A few days ago in the shower I realized there was a way to further improve the code that I was working on.

Leo: I have a theory about this because I always come up with the best stuff in the shower. I think the heat of the shower gets the blood vessels going or something, and relaxed, and you think better in the shower because you're right, always where the best ideas.

Steve: Yeah.

Leo: Don't know why. It's strange.

Steve: Thus, yes, our water company in Southern California is saying, you know, Steve, there is a drought on. And we realize you're looking for some good ideas. But really, stop. Is there nowhere else you...

Leo: Tavis Ormandy came up with a solution in the shower, I seem to remember, right, way back when from Google's Project Zero.

Steve: Yes. Indeed he did.

Leo: So it is a real phenomenon. I don't understand it.

Steve: And is well known to programmers the world over.

Leo: Oh, yeah. I love it, though, going to sleep. Because as I'm drifting off, the problem will be going around in my head.

Steve: Yup.

Leo: And just somehow my brain just keeps working on it, I guess all night. And in the morning, bing. Pops right out.

Steve: There it is.

Leo: Yup.

Steve: So we have some fun feedback from our listeners. David Wright, he tweeted: "Another month, another Microsoft balls-up. Windows Server updates this month kick Exchange in the teeth. Some part of the update doesn't have the correct privileges, so a certificate doesn't get updated, and you can't access Exchange control panel or OWA, et cetera, afterwards. That was a great start to Monday morning, after updating over the weekend. Everything seemed to go through smoothly," he said, "(mails going in and out after the update, but only through automated SMTP). Once the users started coming into work, things went downhill." So yes, David, there's a tweet from the field.

Bob Thomas said: "@SGgrc Any word on Microsoft plans on fixing network printing? I can't print any other way. I got notice to update, and it was horrible. No printing, and I could not tell which update of three was responsible. Until all uninstalled, and then the printer printed. PO'd at Microsoft!"

And I assume Bob meant that he had no choice other than to print over a network. There's news that the promised fix for printing has been released into broader testing; and assuming that all goes well, it should be December 14th. So fingers crossed. I think probably by the year end we're going to get printing as a Christmas present from Microsoft.

Rick Nyman tweeted: "In Microsoft's defense regarding JavaScript in Excel, they are playing catch-up to Google, who has had JavaScript-based Google Apps Script in Sheets for a long time. Microsoft needs to replace VBA with something current. But yes, I hope they can secure it."

Donn Edwards tweeted: "Hi, Steve. Happy Thanksgiving. Is the IETF going to allocate the 0.0.0.0 address space as well?" He says: "Windows uses 0.0.0.0 in the hosts file in the same way it uses 127.0.0.1. What could possibly go wrong?" And I think maybe Donn meant the routing table because, if you do a print route, you will see 0.0.0.0 is also locked down.

Someone whose Twitter name currently, he tends to change it, it's 418: Tea Ready. And of course those with a long memory will remember that there are some bizarrely named HTTP errors. There's the 404, which is Page Not Found. Anyway, 418, Gary reminds us, is Tea Ready. Anyway, he was replying to @nixcraft. And Gary said: "It appears you have a copy of SpinRite by @SGgrc relabeled as Norton Disk Doctor. How did that happen?"

Leo: Well, we know the story, don't we.

Steve: Yes, we do. I mean, it was - they stole it. And it was such a clone that it even looks the same, so much so that Gary, who owns SpinRite and is one of our testers, looked at it and said, "How do you have SpinRite labeled Norton Disk Doctor?"

Leo: Well, now we know.

Steve: I wonder. That's right.

Leo: They don't still sell that, I think. They haven't...

Steve: Oh, no. No, in fact, they abandoned it like maybe after a year because their own tech support, since they hadn't written it, their own tech support couldn't support it.

Leo: Oh.

Steve: And so we were getting support calls from people with Norton Disk Doctor saying, yeah, Norton told us to call you because, you know, we got Norton Utilities, and now it has SpinRite built in. I said, "No, it doesn't."

Leo: They approached you; right? They wanted to buy it.

Steve: Yes. And it was not Peter's fault. For the record, I want everyone to know Peter Norton is a nice guy. It was Ron Posner who was the heart of darkness.

Leo: Oh, legendary heart of darkness, yes.

Steve: Yes. And so Peter and I went to lunch, with Peter having the instructions to get SpinRite from Steve. And he paid me a great compliment, Peter did. He said, "You know, Steve, when I heard you were low-level reformatting hard drives on the fly," he said, because he was in Santa Monica, he said, "I thought I was going to look to the south, down there toward Irvine, and see a mushroom cloud because you can't, you know, really? You're nuts. But apparently you've managed to do it somehow, and it's the number one most requested feature at Norton Utilities is everybody wants SpinRite, and they don't want to have to buy it again, don't have to buy it from you and buy Norton Utilities from me. So we just want to buy SpinRite from you, and everybody will be happy."

I said, "Except me, Peter." I said, "Unless you're willing to pay a ridiculous amount of money, I'm not selling." And he said, "How ridiculous?" I said, "Really ridiculous. It's like we're not even going to talk about it." And he said, "Really? You're sure?" I said, "I'm sure." So we went back and met with Ron Posner, and we walked into Ron's office, Peter and I, and Ron said, "So, we got a deal?" And Peter looked like the dog that had been bad. I mean, he was like he was - he didn't do it. And Peter said no. Then Ron said, "Why not?" Because, I mean, they figured they were on top of the world.

Leo: Oh, yeah, yeah, yeah.

Steve: And how could this little weenie named Steve Gibson...

Leo: Yeah, why would you turn that down, yeah.

Steve: Yeah, this little guy who was trying to scratch together some coin. And anyway, saying no was the best decision I ever made. So, you know, again, they ended up abandoning it. And in fact today is Greg, my support guy's 31-year anniversary.

Leo: No. With you?

Steve: Working with me.

Leo: Holy cow. That's kind of a legend.

Steve: And Sue is several years before that. So both of them...

Leo: Oh, man. You keep the good ones. That's the point.

Steve: More than, yeah, I had two.

Leo: At your peak, how big was SpinRite?

Steve: 23.

Leo: Yeah.

Steve: I remember we had 23 people because on Black Monday, as they called it, I reduced us from 23 to 12 by lunch.

Leo: Wow, that's not a fun time.

Steve: It was not good.

Leo: No.

Steve: No.

Leo: Anybody who's ever owned a business knows that's the absolute worst part of any business.

Steve: Yeah.

Leo: But I'm sure a relief afterwards, you know.

Steve: It had gotten out of control. And they didn't think I was in touch with what was going on. I remember one of them at the back in our group, we met at lunch, and actually his name was Richard, he said, "Who's next?" And I said, "I heard that, Richard." And I said, "Now..."

Leo: Guess who?

Steve: They called themselves "the survivors." And I said, "Look around." I said, "I know you guys all think I haven't been paying attention to what's going on here, that I've been busy coding." I said, "And you probably haven't stopped to think about this. But look around at who's not here listening to this now, and ask yourself if I made a single mistake."

Leo: Oh, wow, yeah.

Steve: And they hadn't thought about that. They were too concerned about themselves.

Leo: Understandably, yeah.

Steve: Yeah, of course. And they started thinking, oh, he's gone, and she's gone, and that nightmare gossip is no longer with us, and so forth. And they understood. And that was the last time that happened. Attrition just...

Leo: The rest was attrition, yeah.

Steve: Yeah, the rest was attrition. And then the Internet happened, and I didn't need them because I need Greg to talk to customers and Sue to do the books and keep the money.

Leo: You still had people designing box covers and writing manuals and all that stuff. You don't need that anymore.

Steve: Yeah. So Kevin Mix, my final tweet. He said: "Hello, Steve. As I listened to SN-846" - of course that was last week - "I had to pause and pull up the IETF draft for the 127/8 unicast proposal. As a frequent reader of RFCs for my day job as a network engineer, I often find myself checking the author's section at the bottom to try to glean some insight into their motivations by seeing which companies or organizations they're associated with. In the draft proposal you referenced" - and he has the link - "all of the authors reference the 'IPv4 Unicast Extensions Project.' Google led me to this GitHub page." And I have a link in the show notes, and he provided it. "So it looks like this is a longer term effort these gentlemen are working toward.

"Professionally speaking, I would not want to be assigned an address out of this pool if it were released to the RIRs for unicast use" - that's the Regional Internet Registries - he says, "as I imagine there would be some hosts or even entire ISPs that would never implement this standard, ensuring that any services hosted on those IPs would have flakey connectivity, at best. One whopper of an understatement in the 'Compatibility and Interoperability' section of the draft gave me a bit of a chuckle." And he said, he then quoted it: "Since deployed implementations' willingness to accept 127/8 addresses as a valid unicast address varies, a host to which an address from this range has been assigned may also have a varying ability to communicate with other hosts."

So an understatement indeed. Which leads us into today's discussion of the broader goals of the IPv4 Cleanup Project.

Leo: Let's go bonkers about bogons.

Steve: Bonkers about bogons. Okay. And bogons is a real thing. I appreciate that you did not put it into the Google machine. Had you done so, you'd have found out what they were.

Leo: No spoilers, no.

Steve: Before we begin, since we're necessarily going to be talking about network addressing, I want to make sure everyone's on the same page about the nomenclature for describing IPv4 networks.

Okay. One of the many brilliant innovations made by the designers of the Internet's routing architecture was this idea of dividing the 32 bits of IPv4 address space into a network number, and the number of a specific machine within that network. So when, for example, we talk about the 10-dot network, we mean any IP whose first, leftmost 8 bits, or also called an "octet," of network address is 10. And that means that all the machines within that network, their IPs begin with 10, and then the other 24 bits differ. The clearer and more formal way of describing it is to say 10/8, where the 8 refers to the number of bits, counting from the left, which will be used to designate the network number, and the rest of the bits to the right will be used to designate a specific machine within that network.

So last week I prefaced our discussion of the IETF's stated intention to redefine the entire currently unroutable 127/8 network into two pieces, to cut it in half. We'd have 127.0/16, which would remain unroutable and be used as a localnet which contains the default localhost IP of 127.0.0.1 plus 65,535 other 127.0 IPs. Then the rest of the 127/8, which would run from 127.1/16 through 127.255/16, would be newly made available to the IANA and the regional Internet registries as blocks of newly routable IPv4 addresses. And the feeling is that's a useful amount. That's more than 16 million addresses.

Okay. But it turns out that this IETF draft proposal is only the tip of the iceberg. And since our discussion of this bit of an iceberg drew so much interest and response last week, it was kind of overwhelming, I decided that while we were on the topic, I ought to share the rest since there's more. So we need to talk about bogons. By definition, "bogons" are IP packets having unroutable source or destination IPs which should therefore never appear on the public Internet.

We have a formal and fun definition of bogons over on Wikipedia, which explains: "The term 'bogon' stems from hacker jargon, with the earliest appearance in the Jargon File in version 1.5.0 dated 1983. It's defined as the quantum of bogosity, or the property of being bogus."

Leo: Yeah, yeah, that makes sense.

Steve: Yeah. "A bogon packet is frequently bogus both in the conventional sense of being forged for illegitimate purposes, and in the hackish sense," writes Wikipedia, "of being incorrect, absurd, and useless." You know, it can't be on the public Internet. "These unused IP addresses are collectively known as a bogon, a contraction of 'bogus logon', or a logon from a place you know no one can actually log on from."

Leo: That makes sense.

Steve: So a bogon.

Leo: A bogus logon.

Steve: Yeah, a bogus logon, a bogon. And just for the record, despite the similarity in sound, bogons should never be confused with Vogons, Leo.

Leo: Very important.

Steve: Vogons are, of course, the distasteful alien race created by Douglas Adams for his "Hitchhiker's Guide to the Galaxy." As we learned in his first novel, Vogons take on very large construction projects and specialize in demolition. And as you reminded us at the beginning of the podcast, they're also very, very bad poets.

Okay. Oh, and just for the record, I was thinking about this, and dare I say in the shower, they are also - these bogons should also not be confused with Bunnons.

Leo: What's that?

Steve: Bunnons were the aliens in the Star Trek movie that my junior high best friend and I and our group created on Super 8mm.

Leo: Not widely seen, but okay.

Steve: No. His sister, Scott's sister, had a huge collection of stuffed bunny rabbits. And so of course we used stop-frame animation to bring the bunny rabbits to life. So of course they were the Bunnons were the aliens.

Leo: Sure. Absolutely.

Steve: And our long-time listeners will remember that at one point in this, because there was an audio track, we needed the very threatening Bunnons to address the Captain of the Enterprise, saying, "We are the Bunnons. Surrender your ship or be destroyed." And so I came up with the idea of recording that on a reel-to-reel tape deck and then reversing the tape so that it would play backwards. And if you play that backwards, you get sna-na ba-na-ni, yo-sha ba-di-dro, pa-shor-yor-nar-ros. So we then recorded yo-sha ba-di-dro, sna-na ba-na-ni, pa-shor-yor-nar-ros. And we reversed that in order to get a very wonderfully alien-sounding "We are the Bunnons. Surrender your ship or be destroyed."

Leo: Oh, that's clever. Very clever.

Steve: It worked.

Leo: Yeah.

Steve: It worked. So yes, we were clever little pre-hackers.

Leo: What grade was this?

Steve: This was eighth grade, before high school.

Leo: Wow, that's cute, that's really cute.

Steve: Well, and of course we had to have a battle between the - I'm sure we had a Klingon battle cruiser and the Enterprise. So Scott put up some black construction paper which he poked a whole bunch of little tiny holes into and then backlit it with spotlights so that we had stars. And the two ships were hung from black thread. And every so often he would like move one, and it would like shake, and then the other one would shake, and then the first one would shake. And then so that was what was recorded on this 8mm film. Then we took a pin and carefully scratched the emulsion on the back in order to create phaser strikes from one ship to the other in order to add our special effects. So, yeah, it was quite a project.

Leo: What fun.

Steve: We always wondered what happened to it. We sort of lost track of the film. Anyway, where was I? Last week, before introducing the localhost 127/8 network, we talked about the concept of non-routable networks by reminding everyone of the most common non-routable private networks that most of us probably have. Remember, 192.168.0 dot something, so that would be 192.168.0/24 since all of the leftmost 24 bits are the network; and then you have one byte, the last number, to specify the machine on the network. But Netgear and some other routers default to 192.168.1/24 for whatever reason.

And of course these are both small pieces of the larger 192.168/16 network, all of which was set aside by that original RFC 1918. And as we know, that RFC also defined the 10/8 and the 172.16/12 groups to be similar set-aside, non-routable networks to be used to number the machines inside of private LANs. So all of the IPs within those ranges are by definition bogons.

But there are also other longstanding set-asides. And they, too, have come under the scrutiny of a group which has been founded and calls themselves the "IPv4 Cleanup Project." And they're quite serious. Before we consider what they have to say, let's take a look at these other Bogon addresses. The original RFC 3330, which was dated September 2002, was later obsoleted by RFC 5735 in 2010 to reflect a few changes, and both RFCs were titled "Special Use IPv4 Addresses."

In the abstract of this thing it says: "This document describes" - again, remember, originally in '02, and then updated in 2010, so still 11 years back. "This document describes the global and other specialized IPv4 address blocks that have been assigned by the IANA, the Internet Assigned Numbers Authority. It does not address IPv4 address space assigned to operators and users through the Regional Internet Registries, nor does it address IPv4 address space assigned directly by IANA prior to the creation of the Regional Internet Registries. It also does not address allocations or assignments of IPv6 addresses or autonomous system numbers."

So here's how they start in describing the whole intent here. And it's brief. "Throughout its history," they said, "the Internet has employed a central Internet Assigned Numbers Authority (IANA) responsible for the allocation and assignment of various identifiers needed for the operation of the Internet." That was specified in RFC 1174. "In the case of the IPv4 address space, the IANA allocates parts of the address space to Regional Internet Registries according to their established needs. These RIRs are responsible for the registration of IPv4 addresses to operators and users of the Internet within their regions.

"On an ongoing basis, the IANA has been designated by the IETF to make assignments in support of the Internet Standards Process." And the Internet Standards Process was documented in RFC 2860. "Section 4 of that document describes that assignment process. Small portions of the IPv4 address space have been allocated or assigned directly by the IANA for global or other specialized purposes. These allocations and assignments have been documented in a variety of RFCs and other documents. This document is intended" - this, the one that we're talking about now, 3330 - "this document is intended to collect these scattered references and provide a current list of special use IPv4 addresses.

"This document is a revision of RFC 3330, which it obsoletes. Its primary purpose is to reflect the changes to the list of special IPv4 assignments since the publication of RFC 3330. It is a companion to RFC 5156, which describes special IPv6 addresses." In other words, this document, this RFC, is the - you could call it the "Bogon Bible."

Okay. So what are the bogon address blocks? First one, 0/8, which is to say, you know, 0.0.0/8, any IP whose first octet is zero. Addresses in this block, the RFC says, refer to source hosts on "this," in other words, the current network. So it's a self-referential network. And they said address 0.0.0.0/32, meaning exactly that IP, may be used as a source address for this host on this network. Other addresses within the 0/8 may be used to refer to specified hosts on this network.

So that's an interesting set-aside; right? That says that an IP whose first octet is zero is a self-reference, that is, it's kind of like a wildcard always referring to the network on which this host resides. And 0.0.0.0, that IP is also referring to this device. And sure enough, if you put in a route print, like under Windows, you'll see 0.0.0.0 matching exactly to the network interface of that machine. So there is another, what is it, 24 bits. So that's 16 million IPs tied up in the 0/8 network.

Then we have the 10-dot that we've talked about, you know, the 10/8 for local networks. And documented in this RFC we have 127/8, that entire block set aside for loopback. Then we have another one that's interesting. And people who've been paying attention may have noted or may have seen this IP and thought, what the heck? Where did that come from? And that's 169.254/16. That's the so-called "link local" block that's described in RFC 3927. And it's allocated for communication between hosts on a single link. Hosts obtain these addresses by auto-configuration, such as when a DHCP server cannot be found.

And if anyone has ever turned on a Windows machine that has its LAN adapter enabled, yet no WiFi set up and no cable plugged in, you may notice that that LAN adapter will be given an IP 169.254 dot something dot something. That is sort of an auto-configuration. If Windows doesn't see anything else, has no other way of getting an IP, hasn't been configured as a static IP, but it's set for obtain IP address automatically, that's what gets used, something from that block.

We also have, as we've mentioned, another one of the RFC 1918s, the 172.16/12. And also 192/24 is a block reserved for IETF protocol assignments, 192/24. So that's 192.0.0 where the last byte is then the machine on the network. And the RFC says at the time of this writing this document there's no current assignments. Allocation policy for future assignments is given - and they talk about the assignment policy. But that is to say 192.0.0 dot anything, or in other words 192.0.0/24, well, that's not a big network; right? That's got 256 machines maximum, never been allocated. And if you had some clever use for it, it's probably safe to do that.

Then there's 192.0.2/24. They define that as TEST-NET-1. They say: "This block is assigned as TEST-NET-1 for use in documentation and example code. It's often used in conjunction with domain names example.com or example.net in vendor and protocol documentation." So they're not legitimately on the Internet, and they've never been assigned. But it's a /24; right? So only 8 bits' worth of machine, so it's only 256 IPs.

There is 192.88.99/24, which is to say 192.88.99 dot anything. We don't see that often. This block is allocated for use as an IP6 to IP4 relay anycast for those relay anycast addresses which are described in RFC 3068. And they said: "In contrast with previously described blocks, packets destined to addresses from this block do appear in the public Internet." And then RFC 3068, Section 7, describes operational practices. Okay. But again, not a big block, only 8 bits of machine.

We also have the other private LAN, 192.168/16. We were just talking about that. That typically is what we have on our routers, although we use it with a 0 dot something and a 1 dot something. There is a 198.18/15. So they say this block has been allocated for use in benchmark tests of network interconnected devices. RFC 2544 explains that this range was assigned to minimize the chance of conflict in case a testing device were to be accidentally connected to part of the Internet.

Packets with source addresses from this range are not meant to be forwarded across the Internet. So again, a bogon range of IPs. Let's see, it's 15 bits, so that's going to be 32,000 IPs. Not a tiny level, but not up at the 16 million level. 198.51.100/24, that's another block of 256 IPs designated as TEST-NET-2. And 203.0.113/24 is TEST-NET-3. So those are small blocks, not of particular interest, but they are reserved.

Now we come to the two biggies. 224.0.0.0, or I should say 224/4, okay, now think about that, 224/4. That means that 28 bits, because 4 plus 28 is 32, 28 bits in the 224 network are machine names. So that's, okay, so 24 bits is 16 million. 25 bits is 32 million. 26 bits is 64 million. So that's 256 million IPs sitting underneath 224. And the RFC says: "This block, formerly known as the Class D address space, is allocated for use in IPv4 multicast address assignments. The IANA guidelines for assignments from this space are described in RFC 3171." So 224/4 is sitting on a huge number, 256 million IPs.

And 240 is the last one. 240/4, also a /4, formerly known as the Class E address space, is reserved for future use. And they say the one exception to this is the "limited broadcast" destination of, and any network engineers know about 255.255.255.255. The very last IP in the 32-bit IPv4 space is a broadcast address, meaning that all devices on the network receive things sent there.

Okay. There's no way we can mess with the three private and the widely used RFC 1918 networks. They're clearly safe. 169.254/16, that one is safe, too. That's the one that is used for auto-configuration. And as I mentioned, anybody who's ever turned on a network adapter that's attached to nothing, Windows chooses an address there. Presumably it will emit an ARP packet asking if anybody else listening has that IP. And if someone responds "Yeah, I do," then Windows will choose a different IP and re-issue an ARP until it finds one which nobody responds to, and then it'll settle down and use that IP. That is kind of a cool way that a bunch of machines without any kind of a central arbiter, no DHCP server, for example, would be able to all obtain unique IPs on a large network. And again, that's a /16, so it's 64,000, 64K, 65,536 IPs.

The zero net is interesting. And it's a /8, so it's taken 16.77 million IPv4s, remember, 1/256th of the entire IPv4 space, out of service. The three TEST-NETs are all /24s. They're not worth bothering with. The 198.18/15 does set aside and tie up 32K IPs, ostensibly for use in benchmark tests of network interconnected devices, as this thing says. I've never run across it, but that one's sort of difficult to appraise. So, aside from the zero net, this leaves us with the two monster allocations that we have not yet talked about in detail. We did talk about reassigning most of 127. So we have 224/4 which has been set aside for use in IPv4 multicast, and 240/4 which the RFC simply states has been reserved for future use. Well, that's sitting on a huge allocation.

Which brings us to the IPv4 Cleanup Project. And this is a page over on GitHub where they're organizing and running things. Under About, they describe themselves briefly in a sentence: "The IPv4 unicast extensions project - making class-e (240/4), 0/8, 127/8, 225/8-232/8 generally usable - adding 419 million new IPs to the world, and fixing various other slightly broken pieces of the IPv4 world."

Okay. So NANOG is N-A-N-O-G, the North American Network Operators Group. And their NANOG listserv is pretty much where most of the Internet evolves. Thursday before last, on November 18th, a well-known guy by the name of John Gilmore responded at some length to a NANOG posting. John, one of the people in the IPv4 Cleanup Project, is one of the founders of the EFF, the Electronic Frontier Foundation.

Leo: And was also, just a few weeks ago, kicked off the board there for reasons we do not know.

Steve: Oh, that's right, we talked about that, right.

Leo: Yeah.

Steve: He created the Cypherpunks mailing list and co-founded Cygnus Solutions. He also created somehow the alt.* hierarchy in Usenet. And, boy, what a sewer that place was.

Leo: Thank you, John.

Steve: Back in the day. Oh.

Leo: He's kind of a libertarian, I think. I get the feeling, you know.

Steve: Yes. He describes himself as a civil libertarian. And remember, Leo, you'll remember this, he once famously quipped that: "The Internet interprets censorship as damage and routes around it."

Leo: Mm-hmm.

Steve: Uh-huh. So I guess he's something of an idealist, as well. Anyway, Steven Bakker (B-A-K-K-E-R), a network engineer of some repute himself, posted, to which John responded. Steven posted: "The ask is to update every IP stack in the world, including validation, equipment retirement, reconfiguration, et cetera." And John took this in good spirit and appeared unruffled. He makes a number of valid points while presenting and stating a well-thought-out case for hugely reducing the Internet's current bogon bloat. So I felt it was worth hearing John out. The purpose of the preamble was to create a foundation and context for understanding John's reply.

Here's what he wrote. He said in response to Steven Bakker's "The ask is to update every stack in the world," he says: "This raises a great question. Is it even doable? What's the risk? What will it cost to upgrade every node on the Internet? And how long might it take?"

He said: "We succeeded in upgrading every end-node in every router in the Internet in the late '90s and early 2000s when we deployed CIDR." Meaning that networks would no longer be strictly Class A, Class B, Class C, meaning /8s, /16s, and /24s, but rather that boundary could be variable. He said: "It was doable." Of course let's also remember that was back in the late 1990s when a lot was doable that you could argue, you know, we didn't all have little IP stacks in all of our plugs and light bulbs and widgets around our homes back then. He said: "It was doable. We know that because we did it. And if we hadn't done it, the Internet would not have scaled to world scale." And that's certainly the case, that you couldn't arbitrarily divide address spaces down if you had to allocate in huge chunks.

He said: "So today, if we decide that unicast use of the 268 million addresses in 240/4 is worth doing, we can upgrade every node." Maybe. But that's what he said. He said: "If we do, we might as well support unicast on the other 16 million addresses in 0/8, and the 16 million in 127/8, and the other about 16 million reserved for 4.2 BSD's pre-standardized subnet broadcast address that nobody has used since 1985. And take a hard look at another hundred million addresses in the vast empty multicast space that have never been assigned by IANA for anybody or anything. Adding the address blocks around the edges makes sense. You only have to upgrade everything once, but the 268 million addresses becomes closer to 400 million formerly wasted addresses. That would be worth half again as much to end users, compared to just doing 240/4."

So the point he's making there is there is a lot of bogon bloat. And if we're going to make a change, let's make a comprehensive change. He says: "It may not be worth it to you, or to your friends. But it would be useful to a lot of people, hundreds of millions of people who you may not even know. People who didn't get IP addresses when they were free; people outside the U.S. and Europe who will be able to buy and use them in five or 10 years, rather than leaving them unused and rotting on the vine forever.

"We already know that making these one-time patches is almost risk-free. 240/4 unicast support is in billions of nodes already, without trouble. Linux, Android, macOS, iOS, and Solaris all started supporting unicast use of 240/4 in 2008. Most people, even most people in NANOG, didn't even notice. 0/8 unicast has been in Linux and Android kernels for multiple years, again with no problems. Unicast use of the lowest address in each subnet is now in Linux and NetBSD recently. See the drafts for specifics."

He says: "If anyone knows of security issues that we haven't addressed in the drafts, please tell us the details." He said: "There's been some arm-waving about the need to update firewalls, but most of these addresses have been usable as unicast on LANs and private networks for more than a decade, and nobody's reported any firewall vulnerabilities to CERT.

"Given the low risk, the natural way for these unicast extensions to roll out is to simply include them in new releases of the various operating systems and router OSes that implement the Internet protocols. It's already happening. We're just asking that the process be adopted universally, which is why we wrote Internet-Drafts for IETF. Microsoft Windows is the biggest laggard. They drop any packet whose destination or source address is in 240/4. When standards said 240/4 was reserved for what might become future arcane, for example, variable-length, anycast, 6to4, et cetera, addressing modes, that made sense. It doesn't make sense in 2021. IPv4 is stable and won't be inventing any new addressing modes. The future is here, and all it wants out of 240/4 is more unicast addresses.

"By following the normal OS upgrade path, the cost of upgrading is almost zero. People naturally upgrade their OSes every few years. They replace their server or laptop with a more capable one that has the latest OS. Laggards might take five or 10 years. Peoples' home WiFi routers break, or are upgraded to faster models, or they change ISPs and throw the old one out every three to five years. A huge proportion of end-users get automatic over-the-net upgrades via an infrastructure that had not yet been built for consumers during the CIDR transition. Patch Tuesday could put some or all of these extensions into billions of systems at scale, for a one-time fixed engineering and testing cost.

"We've tested major routers, and none so far require software updates to enable most of these addresses, except on the lowest address per subnet. At worst, the ISP would have to turn off or reconfigure a bogon filter with a config setting. Also, many 'Martian addresses'" - those are also bogons - "bogon lists are centrally maintained and can easily be updated. We have found no ASIC IP implementations that hardwire in assumptions about specific IP address ranges. If you know of any, please let us know; otherwise, let's let that straw man rest.

"Our drafts don't propose to choose between public and private use of the newly usable unicast addresses, so the prior subject line that said 'unicast public' was incorrect," he said. "Since the kernel and router implementation is the same in either case, we're trying to get those fixed first." He says: "There will be plenty of years and plenty of forums - NANOG, IETF, ICANN, IANA, and the RIRs - in which to wrestle the public-versus-private questions to the ground and make community decisions on actual allocations. But if we don't fix the kernels and routers first, none of those decisions would be implementable.

"Finally, as suggested by David Conrad, there is a well understood process for 'de-bogonizing' an address range on the global Internet, once support for it exists in OSes. Cloudflare used it on 1.1.1.1; RIPE used it on 128/16 and on 2a10::/12," an IPv6. "You introduce a global BGP route for some part of the range, stand up a server on it, and use various distributed measurement test beds to see who can reach that server. When chunks of the Internet can't, an engineer figures out where the blockage is and communicates with that ISP or vendor to resolve the issue. Lather, rinse, and repeat for a year or more until reachability is 'high enough.'

"Addresses that later end up allocated to private address blocks would never need 100% global reachability, but global testing would still help to locate low-volume OS implementations that might need to be updated. Addresses purchased to number retail cell phones need not be as reachable as ones listed on public-facing servers, et cetera. The beauty of a market for IP addresses, rather than one-size-fits-all allocation models, is that ones with different reachability can sell for different prices, at different times, into different niches where they can be put to use." Signed, John.

And I like his argument. The gist of it is that we have nothing to lose by immediately changing some of today's long-established IPv4 standards. Essentially they're just set-asides. IPv4 has clearly outgrown its original design, which deliberately incorporated a lot of waste that once seemed insignificant. That's the only reason you'd give a university or a project a /8. It's like, here you go. Have some IPs. But that waste is not insignificant today. IP stack vendors will be formally notified that the IETF is changing a bunch of definitions which affect a handful of mostly unused IPv4 space. All that's needed are some tweaks to their OS's default routing table.

Engineers simply need clear guidance and specification to do whatever they need to. Last week we noted that F5 Networks BIG-IP systems were currently using some of the space under 127 that's now slated to become routable. But the entire 127.0/16 network remains local and non-routable. So all F5, for example, needs to do is migrate those arbitrarily scattered blocks they're currently using, which are outside the lower 64K IPs, down into that range. Once that's done, they'll know that if the routability of 127/8 should ever change, they'll already be compatible. And there's no reason for them not to change that today since it will be completely compatible with our current system.

The other thing I appreciate about John's post is his sense of time. Everyone is always in a hurry, and 10 years sounds like forever. But it'll be here before we know it. And when we get there, it would be nice to find that 419 million more IPv4 addresses have been widely usable for some time. I think the point is it's entirely possible that it might never happen. But it can never happen if we never make it possible. And making it possible is not difficult or expensive.

So a different look at the same issue. And I think it's one that makes sense. Microsoft will change their routing table. Linux, Mac, iOS, a whole bunch of other OSes largely already have. And once this is tested and is widespread enough, those new blocks of IPs can be made available. So anyway, bogons begone. I think really it just says that we were giving them away like crazy and setting them aside. And just there's been no need to set them aside for quite a long time.

Leo: So you amend what you said last week about this being a potential disaster? You think you agree with his point of view that doing this, I mean, isn't it going to be painful for 10 years?

Steve: So last week I wasn't thinking of it, obviously, the way he is. Last week I was imagining that in a year, for example, or six months, like the IETF was going to make the declaration and say everybody needs to change, and we're going to start handing out those 127.* IPs in 2022, or 2023.

Leo: But if we wait 10 years...

Steve: Right.

Leo: There'll be a minimum amount of pain because by then we'll all be used to the idea.

Steve: Correct.

Leo: We'll have fixed all the systems that might be...

Steve: Well, yeah. I mean, so the natural turnover in equipment will have rendered these things accessible. And the other point he made is it's different to be a server on an IP than to be, for example, a cell phone on an IP. The server, because it's offering services to the entire Internet, needs to be reachable by the entire Internet. An IP that an ISP has given to a cell phone, it needs to be reachable to the IP. So it's a different bar of accessibility. And so the point is, he says, you don't - it's like, yes, certainly the 240/4 IPs, there's no reason not to use them except Windows needs to get changed because right now it's stomping on that entire network in its routing table. If you do a route print in Windows, you'll see 240.0.0.0 is like it's got four entries. It just won't go anywhere. But none of the other OSes do. And we know that Microsoft could change that in two weeks, in December's Patch Tuesday.

So I guess I can see his point. If you're not in a hurry, then it makes sense to lift the barrier, make it official, and see what happens. But if you don't do that today, it will never happen. And we will then forever be wasting 400 and however many it was IPv4s that could be useful to some people. And as he says, it's easy to say, oh, go use IPv6. But obviously a lot of people don't want to for whatever reason. So anyway, I wanted to show and share the flipside of this, which is that, okay, maybe it looks like something worth doing. Again, no one's being forced to do it. But if we don't allow it to happen, it can never happen.

Leo: Okay. I like it. So there. We've just changed the timeline. That's really - we're looking at a larger scale.

Steve: We've said, okay, it's not practical to do it tomorrow.

Leo: Obviously.

Steve: Maybe it will happen by itself in 10 years.

Leo: Right. Although somebody pointed out at the rate we're going to IPv6, we should be there in about 500 years. Maybe we should wait till then.

Steve: Yeah. Exactly.

Leo: Okay. This is what I love about this show. I mean, I don't, ladies and gentlemen, I don't think there's any other show in the world that you would hear this discussion and hear it explained, I think so clearly. And by the way, kudos to Steve for looking further into this after talking about it last week. I love it. There's a real benefit to this show. Thank you for doing it. I appreciate it, Steve. I hope more people listen to it.

There's a couple ways you can consume it. You can of course watch us do it live. I think a lot of people do. I hope you would still download it because we don't really count the live views. But if you want to watch us do it live, kind of like behind the scenes, go to live.twit.tv of a Tuesday afternoon, right after MacBreak Weekly. It's probably around 1:30 p.m. to 2:00 p.m. Pacific. That'd be about 4:30 p.m. Eastern time. That would be 21:30 UTC. You could watch us do it live, chat with us live at irc.twit.tv or in the Club TWiT Discord. Then of course after the fact you can get it from Steve. Steve's got copies at his site, GRC.com, the edited versions, and some unique formats, too, the 16Kb version. He commissions Elaine Farris to write transcripts, and that takes about five days, a week for her to do?

Steve: She really does hurry them out. So normally Thursday afternoon is our typical turnaround.

Leo: Check those out, plus 64Kb audio. That's at GRC.com, Steve's website, the Gibson Research Corporation, now pared down to a mere three people. But they've been there for decades. That's really awesome. That's really awesome. It is three people; right? I mean, basically.

Steve: It is three people. Three people, yeah.

Leo: That's awesome. For 30 years you got the right three people, 30-plus. You can also find, while you're there, so many great free things, information about SQRL, his I think brilliant login system that, I don't know, I don't know if anybody's going to adopt it. Well, some of us, we have. But I don't know if the world's going to adopt it, but it's there if you need it.

You can also get the one thing that makes Steve money, which is SpinRite, his bread and butter, the world's best mass storage maintenance and recovery utility. You can get it right now. 6.0's the current version. But if you buy 6.0 now, you'll get a free upgrade to 6.1, and you get to participate in the development of it, which has been ongoing, but I think we're seeing the light at the end of the tunnel.

Steve: Yeah. I'm very excited.

Leo: No promises. But soon. There's also, just like I said, a bunch of free stuff there, including a chance, if you want to leave him feedback, leave him feedback there at GRC.com/feedback. Probably better to DM him on Twitter. Those are open. His Twitter handle: @SGgrc. Steve is of course also very kindly letting us put it on our website, TWiT.tv/sn. You can get audio and video there.

There's a YouTube channel. You can watch the show there. I think it's YouTube.com/securitynowshow. Maybe. Anyway, you know, if you go to YouTube.com/twit, that links to all the shows. Individual show channels are there. Easiest way to get it may be subscribe in a podcast client. That way you'll get it automatically every week. Even if you watch it live, do that please for us because we do count those downloads, and that helps us.

All you have to do is find a podcast client you like, Pocket Casts, Overcast, Apple Podcasts, Google's Casts, there's tons of them, and subscribe. And if your client allows reviews, please, you heard this show, you know how good this is. Give it a five-star review. Let other people find it. Help them find it because everybody ought to be listening to Security Now!, now in its 18th year. I don't know how many years it is. 17?

Steve: We're in 17, yeah. We're in year 17.

Leo: Feels like it. Yeah. We're a teen, soon to be an adult, yeah. We're going to be an adult. So thank you, Steve. Have a wonderful week. "Wheel of Time," good show on Amazon Prime, by the way.

Steve: Ah, heard you talking about it. You said that the first couple episodes were not quite there, but you really liked what they've done.

Leo: It's often the case, especially for a big saga, the first few episodes, takes a little while to get into it. But Episode 4, which they just aired on Friday, was amazing.

Steve: Ah, cool.

Leo: Even if you haven't read the book, I think it's very good. So I love the books. People grew up reading the "Wheel of Time" series, love the books. I don't think they'll be disappointed by the show. And a lot of people who haven't read the books think the show's good. It's on Amazon Prime, "Wheel of Time." Thank you, Steve.

Steve: Okay, buddy.

Leo: On with the debogonization. See you in December.

Steve: See you in December. Bye.