SECURITY NOW!

Transcript of Episode #514

Tor's Astoria Client

Description: After catching up with a lot of interesting security news, Father Robert and I take a look at recent research into improving the privacy delivered to users of the Tor network. Our conclusions are somewhat distressing.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-514.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-514-lq.mp3

SHOW TEASE: Is your coworker spying on you with his lunch? Is your ISP throttling you for giggles? Is Google listening in to your conversation? Plus Adobe's back in the news, Tor gets an upgrade, Samsung pulled a Lenovo, and Steve Gibson has the Rosetta Stone of security. Security Now! is next.

FATHER ROBERT BALLECER: This is Security Now! with Steve Gibson, Episode 514, recorded June 30th, 2015: Tor's Astoria Client.

It's time for Security Now! with Steve Gibson. That's of course Steve Gibson from GRC.com. He is the brain behind ShieldsUP! and SpinRite, very soon SQRL. He's the man you want whispering in your ear to find out all the issues about the world of information that we live in. Steve, how are you doing today?

Steve Gibson: Hey, Padre. Great to be with you. I get to be working the podcast with you for the next three weeks while Leo is off on his world cruise.

PADRE: No, no, no. He's doing research, Steve.

Steve: Actually, I heard it referred to as a "honeymoon" on the TWiT show on Sunday. I thought, wow, okay. Well, I guess this is like the first time he's been away since he and Lisa got married. So I guess it could be a honeymoon. Anyway, that's what Becky Worley was calling it. I just think of it as we don't have him for three weeks, so you and I get to the podcast.

PADRE: And I am absolutely thrilled. I love every time I get to sub for Leo on this show because it means I get to have a heart-to-heart conversation with a man whose opinion I very much respect on issues that are incredibly important for people who live in a world of information technology. And this - I'm sorry?

Steve: I was going to say, and you are landing on a great week because we've had just a bunch of stuff happening in the last week. We've got Adobe issuing an emergency out-of-cycle patch for Flash. I'm going to go a little more deep into it because the information about it is there. And I think it helps to sort of show our listeners, like, how this war back and forth has escalated. Like in this case it was a Chinese hacking group installing advanced persistent threats. And the level that this battle has reached is kind of sobering. And Microsoft also updated its root certificate store and ruffled a few feathers. Google's Chrome browser has unnerved some people who are concerned that it's now listening to them without their permission.

Samsung did the dumbest thing, I mean, we would seem to have Samsung in the news a lot lately. This is something new, an original solution they came up with for avoiding driver collision. We've got an AM radio that's able to steal nearby crypto keys; an amazing privacy page that someone tweeted me that I'm just - I want to recommend without reservation. Some new information that ISPs are not being Net Neutral, an interesting question about NoScript that has surfaced, some miscellaneous notes, and then we're going to talk about the result of some research to see how much Tor's privacy, I'd like to call it a guarantee, but that's, unfortunately, a little too strong for what it turns out Tor is actually able to offer. So lots of stuff to talk about.

PADRE: So a light week. This is not a whole lot's going on.

Steve: Yeah, you know. We ought to finish this in a few minutes.

PADRE: This Adobe emergency patch, anytime I hear "patch" and "Adobe," I just naturally assume, yeah, that's of course going to happen. But this one's particularly bad.

Steve: Well, it's bad. It's a zero day. So the guys at FireEye security found this in the wild being used in a phishing campaign by a group they've been watching, a Chinese hacking group that they call APT3. And they've named this "Operation Clandestine Wolf." So what they found was that aerospace - and that is United States aerospace and defense contractors, construction and engineering companies, high-tech, telecommunications, and transportation agencies were being targeted. So this is a zero-day vulnerability, and it uses phishing emails to send sort of a generic-looking email. I saw the email. In the show notes, if anyone's interested, there's, like, an even deeper dive.

But what I liked about this was that the FireEye guys really took this apart. And I wanted to sort of share some of the, like, what this attack does inside of Adobe's Flash Player in order to be able to execute arbitrary code on victim computers. So the scenario would be you get a phishing email, and you click on a link, which is how phishing works. And the emails, knowing where they're sending these, and maybe even to whom because they have an email address, they can craft them to maximize, like, the benign-ness of the incoming email.

So first of all, what the FireEye guys wrote was that they said: "The China-based threat APT3, also known as UPS, is responsible for this exploit and activity. This group is one of the more sophisticated threat groups that FireEye threat intelligence tracks, and they have a history of introducing new browser-based zero-day exploits. After successfully exploiting a target host, this group will quickly dump credentials, move laterally within that victim's network to additional hosts, and install custom backdoors. This APT3 group's command-and-control infrastructure," says FireEye, "is difficult to track, and there is little overlap across campaigns."

So that tells us these guys have a lot of resources. They're a professional organization. And they're taking the time, trouble, and effort to, for example, not reuse existing infrastructure. It's so much easier to just reuse, for example, an existing command-and-control infrastructure. But it's clearly not as safe to reuse something from a previous campaign. So that sort of sets the tone of this. So in looking at what this does, FireEye writes: "Upon clicking the URLs involved in the phishing emails, targets were redirected to a compromised server hosting JavaScript profiling scripts. Once a target host was profiled, victims downloaded" - that is, the victims' computers downloaded - "a malicious Adobe Flash Player SWF file and an FLV file as detailed below. This ultimately resulted in a custom backdoor known as SHOTPUT being delivered to the victim's system."

And what I really liked was that then they got into some details. They said: "The attack exploits an unpatched vulnerability" - and now, of course, this was also unknown before it was found - "in the way Adobe Flash Player parses Flash Video (FLV) files. The exploit uses common vector corruption techniques to bypass address space layout randomization" - ASLR that we've talked about often, which is used so that software running in a victim's machine doesn't know where things are, the idea being that the OS deliberately randomizes the layout of standard subsystems within the process space so that it's not easy to know where existing code is because existing code is very useful.

Continuing, they said: "...and uses return-oriented programming (ROP) to bypass data execution prevention." Once again, so the idea is that data execution prevention is a technology that attempts to make exactly these exploits more difficult to perpetrate. It explicitly marks which pieces of memory can be executed versus which are data, the idea being that many attacks involve arranging to execute data which the attacker has provided. But if the data is in a data region, it won't be marked as executable, so that thwarts the attack.

But return-oriented programming is a trick that has developed where you use, "you" the bad guys use existing code snippets, sort of like the end of various subroutines or services that already exist in the operating system. You jump to the tail of some routine, and it executes some code that you want executed. But since it's valid operating system code, the data execution prevention flag is marked as executable, so you bypass DEP. And so by chaining together a series of jumps to existing code, the bad guys can pull off exploits. And that's why you want address space layout randomization, so that the bad guys don't know where the code is that they want to jump to in order to bypass the DEP. Well, it turns out none of that actually works.

PADRE: You know, Steve, this kills me because address space layout randomization was supposed to be - that was the silver bullet. I mean, that was the thing that, well, now we can't have buffer overloads because all they could do is crash something.

Steve: Right.

PADRE: Now, are they getting past that by using the parsing? So if they use the parsing of a valid Adobe Flash file, is that how they're getting in on the tail end of a valid executable?

Steve: Well, I'll just say that what these guys wrote was: "A neat trick to their ROP technique" - that is, the return-oriented programming - "makes it simpler to exploit and will evade some ROP detection techniques." So exactly as you said, Padre, they know that the Flash code knows where the OS code is because there's something called - everyone in Windows is familiar with DLLs, dynamic link libraries. So it's these DLLs that are being scattered around. But when the Flash module is loaded, it is linked to the DLLs wherever they are. So the Flash code has to know where these randomized modules are sitting in order for it to be able to call them. And so that's the way these guys get around all of these protections.

PADRE: Of course I'm not into hacking at all, and people who would exploit this for gain, especially since the gain for this seems very, let's just say "state organized." They were going after aerospace and defense. They're going after construction and engineering, high-tech, telecom, transportation. But putting all that aside, that's an incredibly elegant attack. I've seen people try to get away from address space layout randomization just by continuously pounding on it and hoping to get lucky. But this is an entirely new level of attack. This is people who actually understand what's going on, and they've figured out a backdoor.

Steve: Well, we can assume, since we know that - we are accusing China of having been behind that OPM breach, which is now, you know, every time we look there's more millions of records that have been lost to that. This is the way that kind of breach begins. It's somebody at the Office of Personnel Management clicked on email from their mother or somebody that they trusted or an offer for a loan that they just couldn't pass up. And in fact a loan offer is the example of the phishing mail that these guys show in their article. And that is the beginning of it. So what we're seeing here is this being caught in the wild. But this is how OPM likely got breached.

So to continue, just to give our listeners a sense for the amount of effort that now goes into this, FireEye writes: "Shellcode is stored in the packed Adobe Flash Player exploit file alongside a key used for its encryption. The payload is XOR encoded and hidden inside an image. The Adobe Flash Player exploit is packed with a simple RC4 packer." That's an EXE packer in order to make it smaller. "The RC4 key and ciphertext are BinaryData blobs" - which is the data type that Adobe uses - "that the packer uses to decrypt the Layer 2 Adobe Flash Player file. Once decrypted, Layer 2 is executed with the function loader.loadBytes.

"Layer 2 uses a classic" - and I love it that these guys are at the point where specific corruption techniques have become classic to them. So they say: "Layer 2 uses a 'classic' Adobe Flash Player vector corruption technique to develop its heap corruption vulnerability to a full relative read/write available to ActionScript 3. In this technique, the attacker sprays Adobe Flash Player vectors to the heap, and triggers a write vulnerability to change the size of one of the vectors. The attacker can then perform subsequent reads and writes to memory outside the intended boundaries of the corrupted vector object from AS3, ActionScript 3.

"Once the attacker has limited read/write access to memory, they choose to corrupt a second vector to increase their access range up to 3FFF FFFF bytes," which is a lot. "This second vector is used for the remainder of the exploit. Finally, the attackers use an ROP chain" - a return-oriented programming chain, as I said, a series of calls into existing code - "to call kernel32" - which is the deep kernel OS - "VirtualAlloc routine" - which is the way you get access to memory in Windows - "to mark their shellcode as executable before jumping to the shellcode. Instead of writing their ROP chain to the heap along with their shellcode and payload, they used a different technique.

"Usually, exploit developers will corrupt a built-in Adobe Flash Player object, such as a sound object. Instead, the attackers chose to define their own class in ActionScript 3 with a function that takes a lot of arguments." And in this example they called it CustomClass, with looks like 80 different unsigned int arguments. "Then, the attackers can simply overwrite the function pointer with a gadget that adds to the stack pointer and returns to pivot to the return-oriented programming." It's like, my lord.

I mean, so this gives you a sense of this is not simple, but a sense of the amount of industry which is going into now both sides of this war, where Chinese attackers, who clearly are state-sponsored at this level, are going through this in order, I mean, and must be looking at a disassembly of Flash Player, scrutinizing it instruction by instruction, to find something that no one has found before, a way that they can give it some data that it doesn't expect, that it wasn't designed to handle, in a way that they get a tiny foothold. And then they wedge that open further and further and further until they get enough advantage that they're able to run code that they've supplied encrypted in an image file. I mean, that's just amazing.

And then of course on the other side is FireEye, who finds this and then goes through and reverse-engineers what the APT group has engineered in order to understand what they've done and then says, oh, yeah, this was the classic this and the common that. And it's like, oh, yeah, this is the way things are done these days. That just, you know, once upon a time we were just glad computers worked.

PADRE: You know, Steve, that's an incredible mind dump. And I know there's still a lot of people in the chatroom who are kind of reeling from everything you've just explained. But let's go back to the Adobe Flash Player, the classic vector corruption technique. So my understanding is you throw these vectors onto the heap so that they can be used. But then you use that write vulnerability in order to increase the size of that vector that you're playing with, which gives you access to memory you shouldn't have access to. What makes it so difficult? If this is really a classic attack, what makes it so difficult to close off that vulnerability, just not let you rewrite the size of a vector once it's on the heap?

Steve: Well, see, this is the problem is that all of these exploits use features which we've built into the system for the good guys to use. I mean, for example, a buffer overflow, a classic stack-based buffer overflow, the stack is there for convenience. It's very fast to allocate some memory on the stack. It, like, takes no time at all. There's no faster way to briefly borrow some memory from the system than from the stack. You don't have to ask the operating system. You just change a register in the processor, and it says, oh, okay, any additional space will be below that. So that sort of reserves the chunk you've asked for.

The problem is we also store return instructions on the stack. So the stack is like a convenient scratch pad. And it's the abuse of that, the abuse of that convenience, that bad guys have figured out how to leverage into an exploit. And so that's what we see everywhere is fundamental features of the system which are installed to be used for good, that they don't know how they're being used, and so they don't know that they're being abused.

PADRE: Well, Steve, let me dip into some of the voodoo here. How would the FireEye guys be able to backwards engineer this? I mean, okay, so they notice there's some strange behavior going on on a workstation that they're using for research. What would be the typical process to step back and back and back? I mean, of course they're going to look at the regular, the usual suspects. They're going to look at the common corruption techniques. But at some point, to figure out this kind of detail, how do they do that?

Steve: The only way you can really do it is by following it in. And so they probably captured a phishing mail. Something brought this to their attention. Or they found a corrupted system, then they looked at the recent email in that person's inbox and found something suspicious. That gave them the link. And then they were able to follow the link to the website, see what the website was doing, what payload it was downloading.

And then, in a highly instrumented machine, they watched this thing happen. They, like, followed it in and watched what it did as it went about doing its work. After the fact, I mean, many of these things cover up their own trails so you can't see what's going on after the fact. And so the only way to catch it is by literally single-stepping and watching it go in. And they're seeing it do things, and they're having a series of "oh my god" experiences. And at some point they see, like, the entry point mistake that exists in the Flash Player. And then from that point you're into classic techniques. You need some foothold, a little foothold. And then what's happened is that the bad guys have become expert at prying these open.

PADRE: Steve, we've actually got a very good comment from the chatroom from "Synack," who wonders if containerizing your apps, all apps within an OS, would solve problems like this. That actually reminds me of a conversation that we had when VMs started becoming very popular, and people said, well, you just run everything in a VM, and a VM is not aware of any other VMs. It's not aware of the hardware it's running on. But of course even that protection has been broken. There was just recently a very interesting attack that allowed you to exploit an old floppy disk driver in order to...

Steve: Right.

PADRE: ...make your VM break out of its allocation. The same thing's going to happen with containers; right?

Steve: What we have fundamentally is systems that are incredibly complicated. And one of our mantras on the Security Now! podcast is that complexity is the enemy of security. And so we do the best job we can making the systems as bulletproof and secure as we can. Certainly, Adobe doesn't want to have zero-day exploits, any more than any VM maker or an operating system maker wants to have mistakes in their code. But the systems are so complicated. I mean, sometimes we see very clever exploits that use unintended consequences in a way that wasn't expected. There was an exploit a while ago where some services listening to the Internet were storing temporary variables in the environment. And then, when the shell was run, bash would execute commands if any were present in the environment. Well, so this wasn't an exploit. This was something no one had found before. It was poorly conceived, but it wasn't a flaw. It was just a clever reuse of existing mechanisms that were insecure.

Then we have things like, as you mentioned, the floppy disk driver problem, where old code that had just been sort of brought along because, oh, well, why not, you know, floppy drivers have been around forever, they're probably fine. Well, it turns out there was a problem in that which allowed them to break out of the VM containment. So fundamentally it's that our systems just keep getting more complex, not less complex.

PADRE: A large part of that complexity comes from the fact that we are supporting legacy software and legacy devices. When do we finally say the pain is high enough that we say, look, if it's older than X, we're just not going to use it? If this hardware is beyond this spec, we're just not going to use it.

Steve: Well, one of the things that Apple has been good at, and it's sort of regarded as a mixed blessing, Apple's been rather ruthless at terminating old API systems that they just figure are old enough that not enough people are using them anymore. Microsoft is far more reticent to do that. Microsoft is all about making sure that all of your legacy stuff still works. So that, for example, when they removed 16-bit code support from the base Windows 7 OS, they said, oh, yeah, but, you know, we're going to give you a VM where you'll still be able to run that if you really have to. You know, Microsoft just doesn't want to leave people behind. Apple's a little bit more ruthless about it.

And I would argue that at some point the stuff does become so old that you just have to say, look, supporting this old stuff is just too much of a problem. And in fact we really see that with smartphones, where vulnerabilities are found in years-old phones, and the carriers just can't be really bothered with going back and patching old OSes, leaving people with known vulnerabilities that are just never going to get fixed.

PADRE: Well, I think we, as a geek society, we like to hold onto our old things. And someone telling us that, no, you can't use it, that doesn't sit well with us, even if we know it's for our own good. Even if we know, yeah, you know what, I probably shouldn't be using that Windows XP machine because it's host for every kind of vulnerability I can think of right now, I'm still going to want to use it. Steve, we could bog down in this for the rest of the episode. But for the people who are sufficiently scared, what's the workaround? What's the patch? I know it's out already. Is that good enough? Do they just update?

Steve: Oh, yeah. It's been fixed. Adobe broke their, what was it, I think it was quarterly they were going to do - they announced a couple years ago, oh, yeah, we're going to update quarterly. And I don't think they managed to hold to that during even one quarter because Flash has been notoriously problematical. The truly security-conscious among us no longer have Flash installed. iOS has never had Flash installed. That was one of the famous battles between Steve Jobs and Adobe back at the beginning of the iOS platform was, nope, we're not going to have Flash on this. Chrome updates it automatically. IE updates it automatically. Firefox checks to make sure that you're running the current version and warn you if you're not.

And of course NoScript prevents Flash from running and shows you like a broken symbol. And then you have to explicitly say, okay, in this instance I really do want to run it. So those who are vulnerable are those not listening to the podcast who just have Flash because their system came with it, and it's always had it, and here was a problem that wasn't known. So it wasn't that they weren't updating, it's that there was no known problem until there was.

So you do want to make sure that you've got the latest version of Flash. You can go to, if you're not sure, and if you do need and run Flash, Adobe.com/software/flash/about. So again, Adobe.com/software/flash/about. And that will check your version of Flash. You need 18.0.0.194. That's the most recent zero-day fixed version of Flash. Again, Chrome and IE should take care of that for you. But after something like this it's worth making sure. But best is, one way or another, not to have Flash run by itself. That's the key. In fact, you'd like to go to Adobe.com/software/flash/about and have it tell you, oh, Flash isn't here, it's not installed, or it's not running. You'd like your system not to automatically verify that version. That says that you're safe from unknown future Flash exploits.

PADRE: Of course Adobe wasn't the only company that had a little security egg on its face. Our good friend Microsoft may have done something that twerked the people in the security industry just a tiny bit wrong?

Steve: Yeah, you know, I'm of a different opinion about this. There was a lot of controversy generated because Microsoft, the way it was portrayed was Microsoft secretly is sneaking new trusted root certs into your Windows. Well, they've been doing that forever. I mean, that's what they do. Part of running Windows and any operating system is trusting the vendor. And we know that root stores are big, but we don't have a solution for that at the moment.

So what happened was that somebody was monitoring the changes to their root store and "caught," unquote, Microsoft updating the root certificate store. Well, that's what Microsoft does. It's like this was always happening and nobody was paying attention, and so no one cared. Or how, like, for example, Windows Update was controversial when it began, and now we're just like - in fact, we'll have a story here in a second. This is what Samsung did, stumbled over themselves with Windows Update. Now we want it on, and we want to make sure that we get all the updates.

So the problem with these are, I mean, if you look at this list of 17 certs, they don't look like anything anyone actually wants. There was one global signed cert that you probably need. But there was like a cert from Tunisia. And the certificates are supposed to have human intelligible friendly names, and one was like SPK3-QR or something. So it's like, okay, what does that tell me? It actually turns out to be a Cisco cert. But so this is what Microsoft does. I didn't think it was any big deal.

But I did get something from this that I wanted to share with our listeners because I know we've got some people that will think this is very cool. There is a root certificate auditing tool called RCC. It's a very small executable. In fact, I was surprised. I was impressed that it was so small. It was like 40K or something because it takes advantage of Windows power shell commands. Anyway, this audits your certificate store in both your Windows store and Firefox and notifies you of any certs that are suspicious, which is really handy. For example, it would immediately reveal the Superfish certificate that had been stored in the Windows route. And it would show if your employer had surreptitiously added certificates to the root store.

Anyway, it is at trax.x10.mx/apps. And there's a list of goodies there. And you're looking for RCC. Maybe you could google. I didn't try it. I should have. Maybe google "RCC local root certificate auditing tool." It runs on Windows 7 and later, Windows only. But I know our listeners, and there are people who will love to have something that they can run that will immediately tell them if there are certificates in their own local root store that are not part of the normal set. So I recommend it highly.

PADRE: What would the red letters be for you? I mean, what would be, if these certificates showed up during this sweep, what would be the ones about which you would be most concerned?

Steve: I guess I wouldn't know. It's like the judge famously said of pornography, he knows it when he sees it. It's like, well, looking at those, if it said "Superfish," for example, and it was being highlighted as not part of the standard Windows store, I would say, "What the heck is Superfish?" and then go about doing some research. So I would just say you need to use some judgment. But this thing highlights stuff that other people don't have. And so which immediately asks the question, or begs the question, why don't other people have this? Why do I? What makes my computer different? So, for example, if you've got antiviral software which is doing HTTPS interception in order to "protect you," unquote, you will find its certificate sitting there, enumerated as not part of the standard trust store. So anyway, I think our listeners are going to go nuts for this. I wanted everyone to know about that.

PADRE: That's a very cool, yeah, everyone should have that in their toolkit. I'm playing with it right now and finding a few interesting things in my box here. So you've got me paranoid.

Steve: So Google did something that upset people. And, you know, these sorts of things are making me glad that I'm still over on Firefox. They, without permission, and without any announcement, quietly added something to Chrome and the Chromium project, which is ostensibly open source, to allow them to add the "Ok Google" speech recognition. And it's there. It's in mine. So go to, if you open Chrome and put in chrome://voicesearch, in fact, if you put in, like, "voi," that's all you need, and it'll complete it for you, you will find it saying "Microphone: Yes." "Audio Capture Allowed: Yes." In my case I didn't have a microphone on my machine. I'm using a different machine, for example, for the podcast, so it wasn't able to do that.

But this came to light because apparently something about the way the Debian packages install themselves caused this not to be off by default. Google has said that you have to opt-in under Chrome settings, you know, chrome://settings. You need to turn on "Enable Ok Google" in order for this to happen. But Debian users noticed that the light was on, showing that their webcam was active, without them doing anything. So Google's pushing it off on something about the way Debian's packages update. The controversy is that what Google has created is a blob. It's a binary blob which breaks with the whole concept of open source, especially over on the Chromium side. Chrome users, just like, okay, well, whatever Google wants to do is fine.

But the people who are active with, like, building their own Chromium browser, they take some offense at the idea of a blob being secretly downloaded by their Chromium browser. In Chromium's bug tracking, they said the actual fix for this was add a build flag to disable "hotwording," is what they called it. So Ok Google is a hotword. And the description of that so-called bug, which was going to be fixed by adding this optional - oh, and by the way, it's by default not disabled. You have to turn it off if you don't want hotwording to be compiled into your build of Chromium.

They said: "Hotwording downloads a shared module" - this is the binary blob - "from the web store containing a NaCl" - and remember that's Google's native client technology, which essentially is x86 binary code. So it's a precompiled sandbox binary module. They said: "There is a desire to build and distribute Chromium without this happening. This change adds an "enable_hotwording" build flag that is enabled by default, but can be disabled at compile time," if you don't want hotwording enabled.

And then they said: "A build-time flag" - oh, this is just more of the same. Then two comments I thought were interesting in the thread that this triggered. One guy posted: "May I ask why this extension is hidden from the extensions list at chrome://extensions, although the page chrome://voicesearch shows it as an enabled extension? I suggest that sensitive functionality intended to process data from the surroundings -sound input, video input, et cetera - should be presented in an open and transparent way, with easy-to-find controls."

And then someone else said: "This weirds me out, as well. The whole behavior of hotword is pretty conspicuous: Opt-in default, downloading a binary blob without notification, extension being hidden, ability to record audio." He says, "I almost fell out of my chair when I saw that. Great strategy to erode trust of any user who is even slightly concerned with security." And he says, parens, "(Which I assume a lot of Chromium users are)."

PADRE: Steve, this strikes me as this is "kid town frolics." I mean, you wouldn't expect this out of a company like Google, which has had security issues in the past and has had to become very transparent with its users. This just seems like there were a couple of engineers who said, you know what would be really cool, if Ok Google was enabled on every Chrome browser, and let's not ask anybody else. Let's just push this out.

Steve: Exactly. One of our other common phrases here is "the tyranny of the default," which is the way I like to note that, yes, there are settings. But most of them are never changed from their default. So exactly as you say, Google knows that in order to get Ok Google to be used, it just needs to work. Users are not going to be bothered to go into their chrome://settings in order to go find out, enable Ok Google. And the concern is that we know, first of all, that Ok Google is done locally. The speech recognition is done in this - that's what that binary blob is doing. So not everything in every user's system is streaming out to Google. Google couldn't handle that.

But we also know how often these things fire by mistake, how people's Alexa is going off, or, I mean, it's happening to Leo all the time. He's got so many things listening to his environment now that things are happening just as a consequence of things people say. And the idea is that that's supposed to be like the magic phrase that triggers this. The problem is, when triggered, then it does start sending everything that it picks up offsite to Google's cloud, where number crunchers then try to figure out what you're doing. And as I understand it, it's a third party that offers this, this basically speech to text. And the text probably then goes back to Google, and they figure out what to do with it.

So, you know, I can live without this. And but apparently it's not enabled by default today. But we know that Google is able to change their mind. After some settling-in period, maybe with the next update, they'll just flip it on and say, oh, guess what, you can now say "Ok Google" and ask Google something with your voice.

PADRE: You know, Steve, I remember not too long ago when Chrome was so popular because it was a lightweight option. It didn't have any crud in it. It was fast. It loaded quickly. And now it has become the new Firefox. That's what killed Firefox; right? I mean, they just added so many things onto it, it became impossible to manage.

Steve: On this privacy page that we'll be talking about here in a minute, the only browser they recommend is Firefox. And I'm glad we have you today rather than Leo, Padre, because Leo is a dyed-in-the-wool Google Chrome person, and so I'm sensitive to that. But I'm using Firefox. And Chrome is no longer small. When I launch Chrome, I run Task Manager, I see my memory consumption jump up massively. And so I don't run it because it's a huge memory hog. And to Firefox's credit, or Mozilla's credit, they have gone through some serious memory improvement. I have 144 tabs open at the moment, and it's working just fine. That is, Firefox is. And I love it.

And what we need to understand is that Google is walking a very fine edge when it comes to privacy. They're about advertising. They're about tracking. This Google Analytics that is everywhere is tracking technology. And they're leveraging everything they know about us. In return, we get all this stuff for free. But given a choice of Chrome or Firefox, I'd rather go with a company whose operating model is not leveraging everything they know about me in order to give me free stuff.

PADRE: Right. Yeah, I mean, I'm the same way. I'm starting to migrate away from Chrome. I find myself using IE 6.0 more and more, just as my primary. I can't even say that with a straight face.

Steve: I know.

PADRE: Sorry.

Steve: Actually, IE 6.0 has a problem now. You really can't use it anymore.

PADRE: Exactly. It's dead.

Steve: Because it's just - it's too old. Well, we have, with SSL 3.0 and then the TLS 1.0, 1.1, and 1.2, we've left it behind. So, I mean, I was forced to give up, for GRC's server, I was on Windows 2000, and it was working just fine, nothing's broken, except that I couldn't support any of the new security suites. So I had to - I made a jump, I think it was the holiday season before last, over to Windows Server 2008 R2, only because I needed access to, I mean, I needed GRC to be able to terminate useful connections from GRC's visitors. And then of course SSL Labs liked me a lot more then. It wasn't happy with me at all when I was using Windows 2000, even though it worked just fine.

PADRE: Yeah.

Steve: So, okay. Samsung is in the doghouse again. This is just incredible. And, okay. So the best way to explain this is that some guy was having problems, like noticing that his Windows Update, the Windows Update service in Windows, kept getting disabled. And he, like, well, it sort of seemed anomalous the first time. Then he turned it on again, and then it was disabled again. And he rebooted, and it was disabled. Then he turned it on, and it was okay for a while, then it was disabled. And so he began digging in. And what he found was an executable as part of the Samsung OEM goodies package, you know, all of the so-called crapware, the benefits that Samsung brings along with them OEMing Windows, and you can have a Samsung laptop or desktop. This executable was named "disable_windowsupdate.exe."

PADRE: Huh. I wonder what a file like that might do?

Steve: So first he verifies, he absolutely verified, I mean, the guy knows what he's talking about. His blog is BSOD - you know, Blue Screen of Death - analysis.blogspot.com. So he knows what he speaks of. He verifies absolutely that this is the culprit. Then he contacts Samsung's technical support to, like, say hey, your software is preventing Windows Updates from happening. And their response, oh, goodness, it is just classic.

They responded, and I'm quoting: "When you are enable Windows updates, it will install the Default Drivers for all the hardware no laptop which may or may not work. For example, if there is USB 3.0 on laptop, the ports may not work with the installation of updates. So to prevent this, SW Update tool will prevent the Windows updates."

PADRE: It's very poetic in a fortune cookie style way.

Steve: In sort of a haiku fashion, yeah. So what we've got is Microsoft is not happy. Samsung has danced around this and sort of - it's not clear what their official position is, except they say they're working with Windows. And so it sounds like essentially what happened was Windows Update was replacing some Samsung drivers which, for whatever reason, Samsung hadn't registered to prevent replacement or hadn't provided to Microsoft to offer through Windows Update properly. I mean, obviously nobody else has this problem. But Samsung decided, or some engineer somewhere, probably the same guy that thought Superfish should be included with all these OEM bundles, let's just turn this pesky Windows Update service off because it's bothering our own drivers.

PADRE: I know that Patrick Baker, who is a Microsoft MVP, he's the one who first broke the story. He actually discovered this. And his gut feeling is that it's Samsung's implementation of the USB 3.0 drivers. For some reason their hardware doesn't work well if Microsoft updates its kernel. So go figure. Again, this is a lot like the Google situation, and it's very much like the Lenovo situation in that it's some engineers who - there's recurring problem. They can't figure it out. One of them figures out, oh, if we turn off Windows Updates there's no problem. And they don't pass this up along the chain, along with the impact, which could be, oh, yeah, by the way, we're basically opening up every Samsung customer to a world of exploits.

Steve: Well, and you're also, for example, there's a molehill somewhere, so you drop an atom bomb on it. You know, it's like, wait. If you've got a problem with your USB 3.0 driver, fix the problem. Don't completely disable all Windows updating for your customer base for the rest of time.

PADRE: It just, it boggles the mind, Steve.

Steve: Yeah. Yeah. Problem solved, though. Hey, Padre, problem solved.

PADRE: This isn't an engineer.

Steve: No more...

PADRE: This can't be an engineer because an engineer at Samsung would have figured that's probably not a good idea. We probably don't want to do that. Oh, goodness.

Steve: Yeah. So, okay. This generated more tweets this last week than anything else. And I have to say that it was the pita bread that really put the icing on the cake. The fact that you could tuck this thing into a piece of pita bread, that - I don't know. I was going to call it a "tempest in a teapot," because of course Tempest is the famous technology for using radiation from something for spying, so "tempest in a Teapot." But I thought, well, no, "Tempest in a Pita Bread" doesn't quite have the same ring. So this is an early look at a paper that will be presented at the Workshop on Cryptographic Hardware and Embedded Systems this coming September 2015.

Some beautiful engineering on the part of these researchers. And quoting from their page, they said: "We successfully extracted encryption keys from laptops of various models running GnuPG - popular open source encryption software implementing the OpenPGP standard - within a few seconds. The attack sends a few carefully crafted ciphertexts." This is the brilliant part. "The attack sends a few carefully crafted ciphertexts, and when these are decrypted by the target computer, they trigger the occurrence of specially structured values inside the decryption software. These special values cause externally observable fluctuations in the electromagnetic field surrounding the laptop, in a way that depends upon the pattern of key bits - specifically, the key bits window in the exponentiation routine. The secret key can be deduced from these fluctuations, through signal processing and cryptanalysis."

PADRE: Wow.

Steve: It's just a beautiful piece of work. So, oh, and so they have multiple versions of this. In their first one, they use an SDR, a software-defined radio, and a wire going off to about a six-inch diameter loop, which they hold near the laptop. It can work a few feet away, but it still needs relative proximity. And of course if they improve, if they invested time in directionality and signal amplification, they could end up with a gun sort of thing that you point at a laptop. But, I mean, it isn't the case that the key bits are just tumbling out.

The secret to their cleverness is that they exactly understood the open source algorithm. They were able to determine that, by causing it to decrypt a specific ciphertext, the act of decrypting it would generate relatively low-frequency transience. That's the other thing, is even though the processor is cranking away at 3GB, the actual signal that comes out is about in the 1.7 MHz range, way lower.

And in fact, I said that that first attack used a USB SDR connected to a big loop. Then they came up with a battery-powered one - that's where the pita bread comes in - because they demonstrate, in fact, that's the picture of the week on the show notes this week is their loop with a power supply, four AA batteries, the SDR, and a small battery-powered computer to drive it and collect the signal, all sitting on top of a piece of pita bread, which you imagine you might be able to slice open and just tuck it in there. And if you just sort of - if there was a lumpy-looking pita bread sitting next to your laptop at Starbucks, you might want to think twice about using PGP to decrypt incoming mail that you weren't expecting because that could contain the special ciphertext and leak your PGP keys to the pita bread which is nearby.

PADRE: If you're in the office, and one of your coworkers is walking around with a gyro all day, then, no, maybe you want to take a look at it, just a little.

Steve: Well, and I loved - their final refinement of this was the so-called "consumer radio attack," which is the last picture there on the page, showing an AM radio with its earphone jack plugged into a standard smartphone. They said of the consumer radio attack: "Despite its low price and compact size, assembly of the pita device still requires the purchase of an SDR device. As discussed, the leakage signal is modulated around a carrier of around 1.7 MHz, located in the range of the commercial AM radio frequency band. We managed to use a plain consumer" - I don't even know where they found a plain consumer-grade radio receiver. You know, it look like something with a nine-volt, well, it's called a transistor radio battery for that reason, you know, that we listened to in the '70s. But so they used "a plain consumer radio to acquire the desired signal, replacing the magnetic probe and the SDR receiver. We then recorded the signal by connecting the radio to the mic input of an HTC EVO 4G smartphone." And still pulled off the attack with the radio sitting next to the laptop.

PADRE: Oh, beautiful, beautiful. It's a great bit of engineering. Steve, let me ask you this. So this is essentially using the fact that your computer is an antenna, not a great antenna, but it's still an antenna.

Steve: A transmitter, a transmitter.

PADRE: Exactly. How difficult would it be to take the next step and use a device like this to actually induce certain devices in your computer so that I could, say, control your keyboard remotely. I could control the TPM chip remotely. I mean, that's sci-fi, far off in the distance.

Steve: Yeah. You would need something that was deliberately sensitive. That is, the reason we have digital circuits, not analog circuits, is that digital circuits are inherently noise immune. They deal with zeroes and ones which have clear distinct voltages. And specifically, noise riding on or that alters the zero voltage or the one voltage is never enough to cause confusion between the two voltages. So it's the inherent noise insensitivity of digital which analog systems don't have, which digital does. But any radio receiver is inherently analog. It's looking for noise. Basically, it is a noise receiver. And it goes then through a lot of filtering in order to filter out all the stuff it doesn't want.

So, and there have been, in the past, there have been exploits. For example, if you had WiFi on a laptop, there's a radio receiver. There is a very sensitive noise receiver designed to receive noise and try to find a signal in the noise. And so, if there were mistakes in the processing of the signal that it found, then you could imagine taking over a laptop remotely. And of course we know that's, unfortunately, a little distressingly easy to do. But without something designed to receive a signal, it would take more like an EMP in order to alter the state of the digital circuitry in your laptop. And of course that's the good news.

PADRE: You know, it strikes me that it's a whole lot easier just to take someone's security keys, rather than trying to transmit something into your laptop. Once you've got that, you don't really need to break into the machine.

Steve: Yeah.

PADRE: Okay. Well, now that I know I need to keep my laptop covered in tinfoil, what's next?

Steve: So, okay. Absolute number one recommendation of maybe - there have been too many good things we've covered this year, so I can't say "of the year." But I need to give a shout-out to Mike Liljedahl, L-I-L-J-E-D-A-H-L, who tweeted me: "Maybe you have seen this privacy site. Nice aggregation." Anyway, I had not seen it. I thanked him. And I want to tell everybody about it. So it's www.privacytools.io. And oh, my goodness, is it wonderful. It is on our side. It is crowd driven. At the bottom of the page they talk about their areas on reddit where they're saying please talk to us. Tell us your tips and tricks. The headline of the site reads: "You are being watched. Private and state-sponsored organizations are monitoring and recording your online activities. PrivacyTools.io provides knowledge and tools to protect your privacy against global mass surveillance."

So the page is just a cornucopia of things that we have recommended. And it's like a one-stop shopping for among the best sorts of privacy tools in the industry. So right off the top, at the top of the page, they offer a private search, which I played with a bit, and I noted that it was sometimes pulling from Wikipedia, so it was forwarding the search to Wikipedia, while stripping any information in its forwarding about who you were, and also pulls from Google doing the same thing. So it serves as a proxy layer between you and other search systems. It queries them, but blinds them to who you are. So you can just use this private search that they offer to do your searching. And it shows which search solution produced which result. Very nice piece of work.

And then they talk, I mean, the page is also laced with sort of stuff that we talk about on this podcast all the time. For example, why is it not recommended to choose a U.S.-based service? And they said: "Services based in the United States are not recommended because of the country's surveillance programs, use of National Security Letters (NSLs) and accompanying gag orders, which forbid the recipient from talking about the request. This combination allows the government to secretly force companies to grant complete access to customer data and transform the service into a tool of mass surveillance."

So, for example, then they provide a list of VPN providers with extra layers of privacy. The VPN providers have no affiliates, and so this page shows you a table of the jurisdiction, where the offering company resides, whether they maintain logs over time, and I don't think there was a single one in there that said yes. So none of them are doing logging. These are privacy-protecting VPN providers. What kind of encryption encoding they offer; whether they take bitcoin, and I think they all do; how many servers they have; whether they allow you to use peer-to-peer networking a la BitTorrent through, and lots of them do; what the cost is; and whether they support a warrant canary, which then they go on to explain what a warrant canary is.

This is a page, if you've got people who are interested in privacy, but don't know where to start, point them here and just let them spend some time. And, I mean, it is an education on one page. Their browser recommendation, as I mentioned earlier, there's only one, and that's Firefox. And then they note that what Tor bundles as its frontend is a modified version of the Firefox browser because, again, we know it's open source, and we know what Mozilla's intentions are. There's a discussion of browser fingerprints that we've talked about, educating you about what that is and giving you a button that can check to see whether your browser is presenting a unique value to the Internet with every query that it generates or not. And of course the button forwards us to Panopticlick, that we've talked about extensively in the past, the way browser headers contain so much information that you can often generate a secure fingerprint from that.

PADRE: This is what's coming off of my browser. So this is Chrome. Yeah, that's a fingerprint.

Steve: Holy crap, what is that blob in the middle? Oh, my god.

PADRE: That's my system fonts.

Steve: Oh, my god.

PADRE: Why do you need to know that?

Steve: And why are you sending that with every single query?

PADRE: This is Chrome. I don't open this a whole lot.

Steve: Wow. Wow.

PADRE: My goodness. This is horrible. I feel so bad now. Oh.

Steve: Wow.

PADRE: Okay. Steve, you just gave me my evening reading. That's a great site.

Steve: Oh, it's a fabulous page. They talk about Firefox's privacy add-ons because of course they're recommending Firefox. And in there is Disconnect. And actually, I have to say, we've been talking about adblocking and tracking on the podcast recently. I just switched from Adblock Plus to uBlock Origin. uBlock Origin is the one these guys recommend, and that got me off my butt. I know that uBlock Origin blocks more than Adblock Plus. And Adblock Plus has been a little controversial lately because of course they sell non-blocking to companies that don't want these guys to block them. They do have a requirement that the ads not be annoying, so you are blocking annoying ads, or you're getting non-annoying ads if Adblock is there. But uBlock looks very nice. And it also gives you a running total of how much good it's done for you, which is always fun.

There's even a fingerprint spoofer add-on for Firefox that relates right to what we were talking about, about browser fingerprints. This deliberately scrambles some aspects of your outbound queries in order to spoof your fingerprint, so that you aren't trackable that way. They've got a cookie deletion add-on, the HTTPS Everywhere add-on. And of course NoScript. And then there's a list of privacy-related about:config tweaks for Firefox, things that, you know, we've talked about about:config all the time. That's where the Mozilla tracking protection setting is that's not yet in the UI. But so you have to go into about:config and put in "tracking protection" in the search box. And that limits you to about five or six items, and you want to enable tracking protection. But this is a whole bunch of additional tweaks relating to privacy.

Then they talk about email providers. Again, a beautiful page showing where they're located, how much storage, are they free or not. If not, what do they cost? Whether they take bitcoin. What kind of encryption they offer. They go through email clients. They recommend email alternatives, encrypted instant messenger, encrypted voice and video messenger, encrypted cloud storage services, self-hosted cloud server software, secure file sync software, password manager software, file encryption software, decentralized social networks, DNS providers that have privacy in mind, PC operating systems, live CD operating systems, mobile operating systems, and open source router firmware. I mean, this page has it all. I can't recommend it highly enough.

PADRE: Yeah. If someone is really confused, because let's say they've listened to an episode of Security Now!, and they're blown away about everything they're leaking out, this is the site you recommend they go to and say, look, this is your toolkit. This is your toolbox. This is the primer. Look through this and pick one from each column, and that's your starting point. Not bad.

Steve: Yeah. I just think, I mean, our listeners will dig into things like the about:config privacy enhancements for Firefox. And I know people are curious, like what would you recommend for like a really good privacy enforcing VPN? I've asked several VPN suppliers what are their logging policies and never had a response. Here we have a page of them, located in places like Sweden and The Netherlands, where you would like them to be, not in the U.S. And they explicitly say we're here to protect your privacy. You can use BitTorrent, and we do not log your actions. It's like, okay. That's the VPN I want for that kind of privacy-protecting work.

PADRE: Steve, let me ask you a question. There was something very early on in that site, which I understand, they can't recommend service providers in the United States because of the political environment that we've built around security. There can be national security letters that basically bypass all of our freedoms and protections. You've got the federal government telling service providers, big service providers, like Google and Microsoft, that they can't tell us when information is being requested about that. So I get that.

Steve: Yup.

PADRE: But when they say they can't recommend services in the United States, what are the next logical options? Where can you go and say, hey, I know the things that are happening in the U.S. are not happening here?

Steve: Well, we know that the U.K. is off the chart for good guys also. The U.K. is seriously looking at legislation to outlaw cryptography. It'll be insane if they're able to pass that. We hope not. And the same thing goes for the U.S. We also know that various governments have colluded with each other. For example, legal restrictions prevent our law enforcement from monitoring U.S. citizens in the U.S. But they don't prevent the U.K. from doing so, and vice versa. So we know from some of the documents that Snowden leaked that there are agreements between the intelligence services of other countries to essentially perform information exchanges. So I would say we know the U.S. is out. You wouldn't want to be in China or the U.K. I would say not in any repressive regime. You'd want to be in a country which is generally known for freedom and honoring the rights of its citizens.

PADRE: There's not a whole lot to pick from, though. There really aren't, I mean, because everyone's either in collusion with the United States, or they've got their own program running. I mean, China's got a program running. Russia's got a program running. Any of the large developed countries will have something in their infrastructure that allows either them or someone else to take a peek. There was actually a counter example that was offered to me on an episode of TWiET where, in trying to move your vital services - let's say I'm trying to host an app, I'm trying to host a service - to a place where I think my freedoms will be protected, I actually red-flag my service as something that the NSA should really pay attention to.

Steve: Ah, good point.

PADRE: And it's one of these where, well, I mean, you're damned if you do, you're damned if you don't. Where can I put it, if either having it in the United States or not in the United States is going to trigger something that makes my data exposed?

Steve: Yeah, I'm looking at the location of the VPN services. There's one in Italy. There's Hong Kong, which I think would be problematic. But we have Iceland, Malaysia, Gibraltar, Sweden, Panama - Panama sounds kind of good. And then we've got Perfect Privacy in Panama, Switzerland, and New Zealand. And so there are lots of choices that sure look better than someone saying, yeah, we don't take bitcoin, and we'll log your activities in order to respond to law enforcement requests. I'd rather have a VPN that said we do take bitcoin, we don't care who you are, and we're doing no logging.

PADRE: I just - I can't trust anyone anymore, Steve.

Steve: The world really, in the last couple years, has been turned on its head. I mean, and when we're talking about Tor here, toward the end of the podcast, one of the things that these researchers who examined how could Tor's efforts to provide privacy be increased and to what degree and at what cost, they made it very clear that, once upon a time, when Tor began, it was sort of an exercise in anonymity, and we didn't really know if it was necessary. Now we know that there is huge money behind specifically busting Tor's anonymity attempts being made by the NSA in the U.S. and the intelligence services over in the U.K. So it's not just theoretical anymore. It's real. And I find myself when I'm, I mean, it's now an issue in my head that it never used to be, that we're being watched.

PADRE: Steve, I can't take it anymore. Between me not being able to trust any service provider anymore and worrying about my coworker taking up my PGP keys with his lunch, I need some good news. So why don't we switch into something that is good news? Like, for example, what about ISPs slowing down Internet service? I mean, that's a change of pace.

Steve: Yeah, well, that's a change of pace. At least it's not about our security...

PADRE: Exactly.

Steve: ...being violated. So an outfit called MeasurementLab.net posted the results of a recent study which revealed that major ISPs, and I don't mean to single out AT&T, but I have an example using them, across the U.S. are dramatically throttling their customers' bandwidth to specific sources. So the Guardian had an article quoting this study. And, good, Padre, you've got that up. Scroll down and look at some of these charts. Anywhere you see a big downward dip in, like - there is the AT&T example. Those are two different carriers, the one in the foreground, the darker one being AT&T's bitrate. So actually that chart goes with this text.

In Atlanta, for example, I'm quoting the Guardian: "Comcast provided hourly median download speeds over a CDN [content delivery network] named GTT of 21.4Mb per second at 7:00 p.m. throughout the month of May." So during all of this recent month of May, last month, 21.4Mb, if you were a Comcast customer, and you were pulling content from the GTT content delivery network. "AT&T provided speeds over the same network of one-fifth of a megabit per second. When a network sends more than twice the traffic it receives, that network is required by AT&T to pay for the privilege. When quizzed about slow speeds over GTT, AT&T told Ars Technica earlier this year that it would not upgrade capacity to a CDN that saw that much outgoing traffic until it saw some money from that network."

Now, this is something we've talked about a lot in the past. There's this notion in peering relationships, the idea with peering is they're called "settlement free peering." If two big ISPs interconnect their network, and each of them is able to use the other's network to the same degree, this is the way they think about it. And it's hard for us as consumers to kind of get our mind wrapped around this because it just feels like, you know, the bits go where they want to.

But the way ISPs consider it is, if they're peering with somebody, some other network, which is massively one-sided, meaning massively incoming, then they feel that that other company is using their network unfairly. That is, that company is getting more use of their network than they are getting of that company's network. It isn't reciprocal. And so the ratio, it really does matter in the networking world. I remember when I set up my relationship just as a colocation website and services with Level 3, I was asked what's the ratio of my incoming and outgoing traffic. They did not want it to be too lopsided because of course every one of their customers who has that ratio aggregates together, and then they end up having a problem, if everybody is too lopsided.

So this is back, again, this is the Netflix problem. This is the way the Internet has evolved is for the first time ever we're delivering incredible amounts of bandwidth, like Netflix, which is the majority of the traffic on the Internet in the evenings now. And that's not the way it used to be. So that is incredibly one-sided. It's bandwidth, video, coming from Netflix through multiple providers, finally to the subscriber's provider, where it's then delivered. But in any of those links, if the bandwidth is heavily lopsided, the person receiving the traffic doesn't feel they're getting a fair deal. They feel that their network is being used, and they're not getting reciprocal use of the other network.

Well, no one is going to get reciprocal use of a content delivery network, by definition. It's delivering content. It's not receiving very much. So this is the problem. And what's significant here is that AT&T is reacting. They are saying they are deliberately punishing this GTT CDN by throttling its incoming traffic. Comcast is not. And so Comcast customers are pulling 21.4Mb from this CDN for whatever content it happens to be hosting. AT&T customers are being hurt because any content they want from this GTT is trickling in at one-fifth of a megabit, as opposed to 21.4, so one-hundredth the speed, essentially.

PADRE: You know, it's interesting that this - so these are Tier 1 peers.

Steve: Yes.

PADRE: When we talk about CenturyLink, AT&T, Comcast, these are all up in the Tier 1. The fact that they would make an arrangement with a CDN in the same tier, that really doesn't make sense to me. I mean, they would know from the onset that, yeah, a CDN is never going to have as much of your network's traffic flowing over its network as it is vice versa. So if anything, I think now that we're starting to see transparency in these peering deals - and let's demonize AT&T if we want to, but they do have a legitimate complaint in that it's not a true peering arrangement - at least this will force them to say, okay, well, what's a real peering arrangement, and what's not? You know, who can I deal with as an actual peer, and who should be paying as a customer? And we get to figure out, we get to find out exactly what's going on back there. That all used to be backroom deals.

Steve: Right. And...

PADRE: Now we're actually seeing the info.

Steve: And again, remember, I did not mean to pick on AT&T. On that page it shows all of the various major providers at one time or another, for one reason or another, are upset with one other party or another. And I think that the argument is a CDN isn't paying for the bandwidth it's using. That is, it's charging its customers for offering high bandwidth service, but not paying for the bandwidth that it's charging for, essentially. I mean, so they're highly profitable, a CDN is, if it charges people for caching their content and has a free ride to deliver it.

So, I mean, I can see the argument that, like, some rethinking of these fundamental relationships needs to happen. And as you said, Padre, that doesn't happen, I mean, at least we the public can't get involved until it comes out of being backroom deals and there's some light shined on it. But it's a report like this that puts the lie to the claim that, oh, no, we're Net Neutral. We're treating all traffic the same. I mean, this is why they're fighting back against Net Neutrality is they want money for the bandwidth that they feel they're unfairly being asked to provide to another company that is profiting off of their bandwidth without paying for it.

PADRE: But to offer the other side of the argument, you could say this. Why would customers, why would we pay for a network that doesn't connect to anything?

Steve: Right.

PADRE: Why would I give you my monthly subscription if I can't connect to any of the content that I want? Isn't that what I'm paying you for?

Steve: Right. And...

PADRE: That's the other side of the argument, which is, yeah, you can have an ultrafast network that peers with nobody, and I would never contract with you because I can't get the things that I actually want to see.

Steve: Right. And if in this country we had choice, then we would have some leverage. But, for example, in Southern California, we have Cox. There is no AT&T, no Comcast, nobody else. We have absolutely zero choice. And so this is the problem is we're not able to vote for the provider we want because we can't choose who we want. I have nowhere to go if I want a broadband connection. It's Cox, period.

PADRE: Where I live in California, and actually also down in - so San Francisco and in San Jose, I've had choice in that I've had Comcast and AT&T, but that's not a real choice. I mean, even the FCC has recognized AT&T doesn't really provide - DSL, standard DSL is not broadband; 1.5 mbps is not broadband. So, I mean, and I love the fact that in the last six years they've come around to the fact that, oh, by the way, satellite-delivered Internet, that's also not broadband. And, oh, if you have a 56k modem, which used to be considered broadband, that's really not broadband. And I, for one, prefer a system, even if it's messed up as it is right now, where I have access to that information. And as you said, I may not have access, input into the policy. But at least I know who's playing above the board.

Steve: Right, right.

PADRE: Okay, well, that was good news.

Steve: Yeah. Well, okay.

PADRE: So I'm better.

Steve: This is sort of neutral. And that is, the question was raised, should we trust NoScript, my absolute favorite, I mean, can't live without it, Firefox extension? A guy named Matthew Bryant posts to his blog called The Hacker Blog, and we refer to his work from time to time, good guy, knows what he's talking about. And he decided to take a look at what it was that NoScript was actually doing. He's been annoyed, maybe with me, talking about, like, oh, yeah, I'm not being attacked, I'm not having any problems because I run NoScript.

And so he wanted to check the security of NoScript to see whether he could execute any cross-site scripting or any other exploits that should be prevented, thanks to NoScript. So he dumped out a list of the trusted domains in NoScript and then began to look through them and didn't have to get very far before he found one in his lookup. It was called Zendcdn.net. It was on the trust list, and the domain registration had expired. So he thought, what? No NXDOMAIN record for this. Is it available? He registered it for 10 bucks and was then able to host malicious script on it, which NoScript trusted because NoScript came preset to trust that, in addition to a whole bunch of other domains.

PADRE: Oops.

Steve: So upon finding this, Matthew writes in his blog, he said: "I contacted Giorgio Maone" - who we've talked, you know, Leo know Giorgio, he's the guy, the author of NoScript and the maintainer - "about the vulnerability, and the response time," writes Matthew, "was incredibly quick. Within hours Giorgio had a patch out on his site, and less then two days later the patch was pushed to all NoScript users. This is by far some of the fastest response and patch times I've ever seen, so hats off to him for that." And then he says: "Please note: This stray domain" - that we were talking about, the Zendcdn.net - "is no longer in the default whitelist for NoScript users."

But Matthew's blog raised in me the question about - and he shows on his page the list of default trusted domains. Now, looking at them, you can see why they're trusted. They're probably to minimize the pain that a novice NoScript user goes through. So things that people are probably going to end up trusting anyway, like Google.com, Live.com, Live.net, MSN.com, Outlook.com, you know, things you probably would trust, he preloads. I realized, for myself, I've been using NoScript for years. I try to say "temporarily trust," but sometimes if I think I'm going to be using a site a lot, I say "trust," and it goes into this permanent white list. This morning I cleared mine. I clicked one entry in there, hit Ctrl-A to select them all, and I hit Remove. And there's a few that are grayed, like about:config, that is, you know, that aren't really domains, they're domain-esque because they're shortcuts that Firefox uses. Otherwise, I am starting over.

And I think, if you don't mind the hassle of retraining NoScript, I was never really aware, I wasn't paying attention years ago when I first installed it, about the list of pretrusted domains. But I don't want any. I'm willing to have to go back and say, yes, Google.com, I guess, and Google Analytics, maybe, and so on. And I'm finding things are breaking, and I'm saying, oh, yeah, of course. Well, at GRC.com I don't have any scripts on GRC.com, so that I don't really have to. I do have some script over in some of the SQRL test pages, so I turned that on, although it works without scripts. And so forth.

So I just wanted to say, hey, you know, sort of I know that we've got a lot of Firefox users here because Firefox is the privacy-respecting browser. We've got a lot of NoScript users in Firefox because we all understand that, while scripting may not be perfect protection, you can't run a Flash exploit when you go to a website if you've got NoScript there. It will block that. So I definitely want that present. But we do tend to sort of accumulate sites in that whitelist. I don't know what performance overhead there might be as the list grows. But I cleared mine this morning, and I'm going to start training it from scratch.

And if anyone feels similarly, I'm glad that I brought this up, and I thank Matthew for bringing it up, too. It is obviously an "oops" to have a domain that's on the whitelist expire so that any miscreant - not that Matthew is, of course, a miscreant. But anyone else could have snatched it up and hosted some malicious content and arranged to get people to go there; and anyone with NoScript would be reduced to the same level of security as everybody else who runs with scripting on all the time.

PADRE: Yeah, and that's bad, that a trusted domain was allowed to pass into other hands. I mean, we can all agree on that. But to look on the bright side of this, this is about as good of an outcome that you can hope for, which is...

Steve: Yes.

PADRE: ...NoScript responded extremely quickly. I mean, that's what you want to see out of a vendor that's giving you a product that you trust for your security. If it took them a week to respond, then you have to worry. I mean, there was a problem; they addressed it; it was done. And you bring up a very important part that I think just needs to be part of our security hygiene, and that is we will become accustomed to our tools. It's human nature. We will think, this works, and as long as I keep doing it this way, it's going to work.

Every once in a while we have to stop, we have to clear it, and we have to make sure that everything we trust, we still trust. And there is no automation for that. That's just us. That's us, every once in a while, as you said, dumping the whitelist, starting from scratch, and saying, okay, do I really need to give this site permission all the time? We've got "Nerve" in the chatroom who's saying, look, what I don't like about NoScript is that it becomes annoying. And that's true. It becomes incredibly annoying. But that's part of it.

Steve: First of all, we have talked about that, and I'll just say it again, because I also reset all my defaults. And immediately I get this little banner across the bottom saying that it's blocking script. The first thing you do is you turn that off. And I don't mind having sounds, and so I turn on the audible - it's kind of a little "grriit" sound, just a little "zwiit" sort of thing, when it blocks a script. That's a little cue to remind me that the site I'm visiting has had some scripts blocked. So that if it does something wrong, then I allow it either temporarily or permanently, depending. But I absolutely agree. NoScript in its default configuration is unusable because this thing is constantly telling you it's blocking scripts. I mean, I get it. It's like advertising its effectiveness or something. But it's like, okay, turn that off. I know it's there. I know it's effective.

Now, if he's saying that just visiting a site whose scripts are initially disabled is annoying, I have no help for you, my friend, because that's also protecting you. It is keeping anything that wants to run on your machine from having its way with your machine. Today's scripting is not as safe as it should be. That's what this whole Flash thing we were just talking about was about. Browsers are trying to be sandboxed, but the problem is we're putting an artificial constraint around them to keep them from leaking. And again, they're complex, and we don't know how to do that well enough. So I'm a strong proponent of NoScript.

And I've got a little icon sitting there in my toolbar. It makes it easy to allow a site to script while I'm there for the session; or, if it's something I'm going to be using all the time, make it permanent. And frankly, it's sort of an exponential curve. After a while, everything you do all the time has permission, those sites that you choose to trust. But I think all of us in our browsing habits, you know, we're scattered all over the place. We're following links. We're like, oh, there's an interesting link over there. Go follow that. Well, for all of that kind of promiscuous browsing, I think it makes sense to have protection.

PADRE: Yeah, absolutely. And you know what else is really annoying, getting my yearly checkup and making sure that I'm doing the things that I need to do to stay healthy. I think this is just the new norm. If you want to stay secure, if you want to enjoy not having people in your system, in your box, having advanced persistent threats running around in your OS, then this is just what you have to do. And I'll take a little bit of annoyance for a little bit of security. That's just me.

Steve: So another tip for a password app that I had not been aware of it, I'm not sure how long it's been around. The website is - and maybe try googling this, "master password app," Padre, see if you just google "master password app," does Google find that? Because that would be much easier than giving people the URL, if Google will take them to it.

PADRE: Bing definitely doesn't work. Oh, there it is, master password app [ssl.masterpasswordapp.com].

Steve: Okay. So here's what this is. First of all, it's completely cross-platform: iPhone, iPad, Mac, so iOS, Mac. It can run with Java on your desktop. The Android version is in beta and the Web version is in beta. So there's two ways, okay, there's three ways you could get a hopefully good password. You could try to make one up yourself. You could go to some random password maker. Maybe, for example, if you're using LastPass, you'd just have it generate your password. Or many people go to GRC.com/passwords and use the gibberish that that page provides them.

Or you could use an algorithmic generator. And that's what this is. This takes your full name as part of the seed, and one master password so that there's something more than just your full name because other people would know your full name, and a monotonically increasing integer, meaning just an integer, zero, one, two, three, in order to allow you to do different passwords for a given domain. Then you give it a domain, and it uses the fixed information - your full name, one master password, and the integer - as the key to hash the domain name into a high-quality password.

So the point of this is, this would allow you, rather than using truly random passwords, to have deterministic passwords. I would say you would still use a password manager in order to make them easily usable. But the advantage of this is that, if you set all of your passwords through this, you could then - you could deterministically recreate them all, rather than them just all being random and lost. So it's kind of a cool idea. I don't know that I'm going to go to the trouble to use it, but I thought that it would appeal to some of our listeners, so I wanted to share it. It's just, you know, it's a password generator based on some deterministic information that you know, and the domain name which is used. And that way the idea is, when you put all the same stuff in again, and the domain you want a password for, it regenerates the same password that it generated the first time, and as many times in the future as you ask it to.

PADRE: Right. And that actually addresses a weakness of a lot of random password generators that I've used, which is, if you forget a password, it doesn't matter if it's random or not, you're still forgetting the password, and you're still more likely to do something silly like writing it down or having it accessible. If I at least know the sequence that I need to go through to regenerate that random password, and I can recreate it, then I'm not going to be, well, pushed to leaving it on a Post-it note underneath my keyboard.

Steve: Right. Something came up after last week's podcast. In fact, I think it was on one of Leo's Wednesday podcasts. I just had it playing in the background while I was working on SQRL. And Leo loved it, and I wanted to share it with our listeners. I don't know if it will appeal to people. But it's an interesting idea. It's called "Google Contributor." And if you just put "google contributor" into your search bar, you'll get taken there. The idea is that, as we know, Google is a major supplier of ads on the Internet. They purchased, for example, what's the infamous massive early advertiser? I'm blanking on the name.

PADRE: Not DoubleClick?

Steve: DoubleClick, yes. So they own DoubleClick and other ad providers. Google is saying, if you would rather not see ads, but you do want to support the sites that you're visiting that would normally be supported by your seeing those ads, you can provide Google with $2, $5, or $10 a month. And Google will use that money to reimburse the sites that you visit to the degree you visit them on your behalf. So you're supporting the sites that you visit, and you can choose among a couple different, like, blanks that show in place, or even provide your own URL to some artwork of your choice, which appears in the areas where the ads would have been. And I've used it for a couple of days, and I can see - and it also audits. It shows you how much of your money has gone to which sites you have visited and had no ads presented in exchange for you providing them essentially a micropayment out of this monthly fee that you pay.

So anyway, I tweeted it. There was a lot of controversy because some people just feel that this is making explicit the sort of a devil's bargain that Google, being the ad purveyor and having so much control over our privacy and tracking and so forth, is pursuing. Other people liked the idea. Leo went crazy over it, saying, hey, you know, here's a great solution.

PADRE: This almost feels like Google's version of Patreon. This is a way for you to directly support the content creators that you enjoy.

Steve: Right.

PADRE: Now, of course Google's going to be taking a percentage of that, and it's running through their ad system.

Steve: Right, right.

PADRE: But, yeah, I mean, I could see this being, not a huge thing in terms of the business that Google does. But it's a nice way for you to say, hey, you know what, you run this security site, or you run this hardware comparison site that I visit each week. I'm going to throw $5 in there, and it's going to directly support what would have gone, the ads that would have gone on your site. I'll get a better experience, and I get the knowledge of knowing that I'm supporting a content person I enjoy directly. Which is why people do Patreon. That's why Patreon is so popular in some circles.

Steve: Yeah, yeah. So I just wanted to let people know that it's there. The page shows that it's - I guess it's been around for six months, like in a testing mode. It's still an invitation-only. You can ask Google for an invitation, or you can get invitations from anybody else. When you do subscribe, you get five that you're able to give away, although you're only able to email them to Gmail addresses, so it's keeping it within the family. And I guess maybe you have to have a Google account or presence. Actually, I think you do because you need to use Google Wallet in order to generate the payment stream.

So anyway, I just - I thought it was an interesting alternative. We're experimenting with how can we have content and not have ads? Now, others noted, hey, but wait a minute, it's not the ads I care about as much as the tracking. And so this is doing nothing to thwart tracking. And, yeah, I have to agree with that. It's probably the case. You're still going to be tracked. At least you won't be seeing ads.

PADRE: Well, I mean, there would be no way around that because, if it's going to remove the ads just for you, it needs to track to make sure that it's you.

Steve: Yeah.

PADRE: Okay.

Steve: So as I'm sure you know, Padre, we're in Leap Second Day.

PADRE: I am so looking forward to this. I have everything planned that I'm going to be doing with my extra second. It's going to be epic.

Steve: So at 5:00 p.m. Pacific time, 8:00 p.m. Eastern, where the clock - and really this is - it's those weird times because it's midnight UTC. In UTC time, the way they're saying this, there's two ways I've seen it described. At 11:59:59, either the time goes to 11:59:60 instead of wrapping around to 12:00 o'clock, or it stays at 11:59:59 for one more second, and then it wraps around to 12:00. So, or is it 29:00? It might be - I mean 23:00 around to zero. But so this is happening at 5:00 p.m. Pacific, 8:00 p.m. Eastern. And of course this is due to the fact that the Earth is slowing down, which really bums me out whenever the Earth's rotation slows down. But essentially we need to make up for the fact that our otherwise perfect timekeeping system, which is all now cesium beam and GPS satellite and in all kinds of, you know, extremely sensitive, the Earth is not cooperating. It's actually rotating a little differently than 24 hours.

And so twice a year, I think it's at the end of December or the end of June, the opportunity presents itself for a Leap Second to be added or subtracted, whichever we need. And Ars Technica notes that in June of 2012, which is when the last Leap Second was applied, reddit crashed, Gawker went down, and lots of Linux servers fell over. Oh, and the Qantas Australian airline had some computer problems that caused 50 of its flights to be delayed. So it's not like this is nothing. I mean, you do something like this, this uses code in unexplored code paths, and there can be side effects. So not huge cataclysms of any kind.

I guess maybe this is why Google has decided they're not doing it all at once. They've chosen to smear that second across all of today. So all during the day, much tinier fractions of a second in Google's timekeeping system have been added so that they're not going to have a single one-second-long cataclysmic event at 5:00 p.m. Pacific time because Google is here in Silicon Valley on the West Coast. So anyway, that's happening. I thought that was kind of cool.

PADRE: Steve, I'm okay, I'm okay when we get an extra second. But on those days when they take that second away, it just really messes up my internal clock, and I'm just - I'm no good for the rest of the day. It's just - it bothers - no, I will say, seriously, though, this has happened a few times, so I've known about this phenomenon.

Steve: Yup.

PADRE: And I do IT for a different organization, and we do have some very interesting proprietary secure terminals. And we actually have a process for this because we have to shut down those secure connections because the extra second actually corrupts the tunnels.

Steve: And stock market systems are being deliberately shut down during that period of time because people are just not sure what they would do, and so they just don't want the stock market computers to experience something that might confuse them. Better just to turn them off, and then they won't know that that second got doubled. They'll just come back up and say, oh, look, my clock's off, and then they'll adjust their clock.

PADRE: Right. The first time it happened to us, we had no idea why it had gone crazy. I mean, and it's a large network, so we had to restart the network.

Steve: It actually had a consequence on your system.

PADRE: Yes. Well, because the way it works is our system is incredibly sensitive to jitter, and it's to make sure that there's nothing in the middle, there's nothing listening, and there's nothing unencrypting and then adding traffic. And so the addition of the second just drove it crazy. It felt like something had jumped in there and was now intercepting all traffic. And it killed the network. And it actually took me three days to get everything back up and running. So now we do an orderly shutdown, we let it pass, and we bring it back up.

Steve: So one of our fan favorites, our listener fan favorites, is the new series "Mr. Robot," which premiered officially last Wednesday. Second episode will be on tomorrow, next Wednesday. And I just wanted to mention that it is so popular from the premiere, that is, the preview that was offered, because they made it available, because it's on USA Network, USA Network made it widely available for the entire month prior to its maiden voyage. It was renewed for a second season before the first episode officially aired. That's how sure they were they had a hit.

PADRE: Did you watch it?

Steve: Oh, yeah.

PADRE: It is fantastic. It's really, really well done.

Steve: It's great. Really well done.

PADRE: And the funny thing is last weekend I was watching "Battleship." Don't ask, it was just - I just had to watch it. And there's a bridge scene with Liam Neeson.

Steve: That's actually a good - "Battleship" is a good movie.

PADRE: I enjoyed it. I enjoyed it.

Steve: I've seen it twice.

PADRE: If you watch it again, look for the bridge scene with Liam Neeson. And the ensign that he's speaking to, that's the actor from "Mr. Robot."

Steve: No kidding.

PADRE: I was just, "I've seen this guy. Where have I seen him from?"

Steve: Interesting. Because I really like their choice. I think that they chose a - he's got these weird eyes, it's like, whoa, you know, like big eyes.

PADRE: Well, he could do that deadpan face incredibly well. Which I see in programmers all the time. But the portrayal of what it's like to be a programmer, the portrayal of what it's like to be a security person, of course they've made it a big more dramatic. But it was spot-on, and very entertaining.

Steve: Yes. And I did want to mention that "Halt and Catch Fire," which we talked about last season, we had great hopes for it, it was a series about the beginning of - we weren't sure what it was. It sounded like the story maybe of Compaq computer or something. It turns out it was another Texas-based clone manufacturer. It wasn't as good as we hoped. For what it's worth, I just wanted to mention that the second season is now running, we've had two episodes so far, maybe three, and I'm liking it. If you like the characters, it's become character driven. It's really - it's got kind of technology and networking and computers. They use jargon kind of okay, not nearly as properly and correctly as "Mr. Robot." But it's, you know, I'm enjoying it. And so for what it's worth, if you like the characters that were developed in the first season, second season is underway. And I think it's okay.

PADRE: I have to say I saw that on Netflix, and I saw the little thumbnail. And I had, I'm like, oh, what is this, a stoner series? And then I watched one episode, I'm like, oh, I had no idea that this was about IT. This is essentially - it's a darker take on Silicon Valley lifestyle than, say, "Silicon Valley."

Steve: Right.

PADRE: It was interesting. I can't get into a new series right now, especially since I'm probably going to be doing "Mr. Robot," but I like it, yeah.

Steve: Yeah. Well, and where they are is sort of interesting. I don't know if they're going to, I mean, I don't know how well the writers understand the history. But they're, like, on the verge of playing with the idea that maybe it's not multiplayer gaming, and this is way back then when, I mean, you were talking about incredibly low resolution, and it was nothing like we have multiplayer gaming today. But they were experimenting with social networking, that is, the idea that maybe people just wanted to talk to each other. And it's interesting to see how this is evolving because of course we know things like CompuServe and the Source and Delphi, you know, these major services were just all online forums, largely, people just talking to each other, which turned out to be a big revenue generator. So it'd be fun to see if they actually pick that thread up and run with it.

PADRE: Steve, I feel the need for something a bit more classy than the regular security fare. What about something with a name like Astoria?

Steve: Yes. So, okay. We've talked about Tor extensively on the podcast because, of course, it's technology, and it's interesting. It's about privacy. A quick, very quick refresher. Our listeners will remember that Tor used to be an acronym. They've said officially they're no longer an acronym. But it used to be TOR that stood for The Onion Router. And the onion notion was that you would get - your traffic would move from one Tor node to another, where we can think of "node" as sort of like a proxy server, that is, it's you're creating a connection with it. It's creating a connection to another machine, creating a connection to another machine, and so forth, until finally it lets your traffic back out on to the Internet.

And the idea is that the path you take is intended to obscure your location. And in order to make this trustworthy, this onion routing technology, the concept is that you choose the route you're going to take, that is, these nodes that your traffic will move through. And you obtain from each of those nodes that node's public key. Then you take your information, and you encrypt it with the first node's - wait a minute, no, with the last node's public key. Then you encrypt that with the next to the last node's public key. Then you encrypt that with the next closer node's public key, and so forth, until the outer wrapping is the public key of the closest node to you.

So then you send this blob - and that's where the onion comes from. It's like shells. It's encased in multiple layers. You send this to the first node. Well, that was the outer layer was encrypted with that node's public key. So it alone knows how to decrypt it with its private key. So it takes the outer wrapper off. There it finds the next IP to send this to. So while it was going to it, the next IP was not obvious. No one monitoring your traffic could see - they could see where it was going to, but they could not see where it was then going to go to, or then going to go to, or then going to go to. Only as it gets to each layer, each node, is that node able to use its secret private key to decrypt the outer wrapper and obtain the IP address of the next node in the chain. And notice that now it's encrypted with that next node's public key, so this node can't tell anything further about what's in it. It only gets one layer, its own layer. So it's very clever. I loved the technology. We talked about it years ago.

The problem is that this was sort of designed in an era, as I mentioned earlier in the podcast, when we sort of thought, well, traffic might be monitored on the Internet. A server did have - if your system was connecting directly to a service, then they had your IP because their traffic has to get back to you somehow. And so it was like, oh. Well, maybe I want a little more privacy than this direct two-party relationship. I'd like to have it bounce around a little bit first. So that's where the whole Tor concept came from.

In today's context, though, things are very different. We now absolutely know that VPN servers are being monitored because the traffic egresses from the VPN termination server out onto the Internet. And there's just suspicion that sort of naturally surrounds users of VPNs, in the same way that there's suspicion that surrounds people who use encryption. It's like, unfortunately, our government's law enforcement agencies feel the need to be able to see everything that everybody is sending to everyone, and who they're sending, and know who they're sending it to, all the time. And they would like to have that information. And now we know that they have the budget to obtain a lot of it.

So there's been questions raised about how good Tor's privacy attempts - I keep wanting to use the word "guarantee," but guarantee is too strong a word. Guarantee means something. And the problem is the Internet was never designed, the beginning of the 'Net had no concept of privacy. It was that point-to-point relationship. It was client and server. And you both knew each other's IP address in order to send traffic to each other. And their goal was not to have privacy, it was to make the darn thing work. I mean, just the fact that, you know, "Watson, come here," you know, the fact that the message got through was a miracle. They couldn't believe it when this stuff began, like when routers actually routed. That was just incredible to them.

So the whole structure was designed to function, as its first priority, not to give privacy. And the problem is, I mean, this podcast, one of our main themes is the difficulty of having private communication. And unfortunately we're trying to have a private communication across a network that never had privacy as one of its designed goals. That just wasn't what it was aiming at doing.

Now, in the beginning, Tor had about 33 nodes, when it sort of began to come to life. I mean, you know, at a point that it was being measured and looked at, it had 33 nodes. Early. Today, thanks to many volunteers setting up onion routers and volunteering their bandwidth, putting them in the onion router network, we're at more than 6,000 Tor nodes. Now, it's tempting to think that there's safety in numbers, that our traffic would get lost among all those nodes, that there are just too many of them, that there's, like, a lot happening, and we're not going to be found. But the problem is the routing technology of the network does not give us what we want. It does not give us privacy.

So a group of researchers decided to examine what the true privacy deliverable is of Tor today, and look at the problems, and see what they could do to improve it. So they refer to like the standard Tor client, and then they made an experimental one called AsTORia in order to play with making better decisions.

So quoting from the beginning of their paper, they said: "The popularity of Tor as an anonymity system has made it a popular target for a variety of attacks including blocking, denial of service, and traffic correlation attacks. In this paper, we focus on traffic correlation attacks, which are no longer solely in the realm of academic research, thanks to recent revelations about the NSA and GCHQ actively working to implement these attacks in practice."

So the notion that these guys use is that of perimeters of knowledge, perimeters of control. We were talking about Level 1, Tier 1 ISPs. Most of the Tier 1 ISPs, probably all of them, are so-called "autonomous systems." They're ASMs. They have an ASN, an autonomous system number, which uniquely identifies their network for Internet routing purposes. And so when you are someone who acquires a block of IPs, not like renting them from an ISP, but ICANN says, "Here is a block of IPs," you get an ASN, an autonomous system number.

Now, these autonomous systems are huge. Networks like Level 3, like, well, it used to be Global Crossing, but now Level 3 bought them, and now it's a huge Level 3, like AT&T or Sprint, major carriers of bandwidth. And what often happens is, just because these guys are so big, a random choice of entry and exit nodes, the entry node being the first node or the node that your client connects to as the beginning of the link in the chain of onion routing, the exit node being that node where the last wrapper of encryption was removed. And very much like a VPN server releasing its clients' traffic onto the Internet at that point, similarly the exit node releases the user's traffic onto the Internet to go towards its final destination. And the idea being that by bouncing around inside this network of now more than 6,000 nodes, nobody can tell who is behind the traffic that comes out the other end.

So the problem is, because these autonomous systems, these networks are so large, it can very often be the case that your exit node and your entry node are in the same autonomous system. It turns out that an analysis of the entry and exit nodes that the standard Tor client generates has it very often the case that a single network entity can know, can observe both the incoming and outgoing traffic from Tor nodes. And that's the problem. The assumption being that, if your traffic crosses autonomous system boundaries, that's much better because the presumption is their information is getting lost. They're not able to actively coordinate, if they wanted to, monitoring all the traffic coming into their network and exiting their network. That is, they would be seeing it going to another autonomous system, not back out to a destination server or a client on the other end.

So their analysis of the circuits - these multi-hop nodes that Tor generates, they're called "circuits." So when you establish this set of links, that's a circuit. So they analyze the default circuits that the Tor client produces today, and they find that up to 58% of circuits constructed by the current Tor client are vulnerable to network-level attackers. That is, nearly 60%, 58% are not well constructed by the current client. They are not deliberately constructed to thwart traffic pattern analysis.

Up to 43% of all sites in the local Alexa Top 500, so Alexa's Top 500 sites, destination servers, websites, in Brazil, China, Germany, Spain, France, England, Iran, Italy, Russia, and the U.S., that is, 43% of the Alexa Top 500 in all of those countries had main content that was not reached via a safe path, what they call a "safe path," meaning one that is deliberately using different autonomous systems for its various nodes. And that is to say a path that was free from what they describe as "network-level attackers."

They found that connections from China were the most vulnerable to network-level attackers, with up to 85.7% of all Tor circuits and 78% of all main content requests to sites in the local Alexa Top 500 being vulnerable to colluding network-level attackers. And of course part of this is that China just has much less diversity of networks. There are fewer autonomous systems, or they have sibling relationships, which this paper also discusses, feeling that a sibling relationship doesn't provide security because it makes it easy for them to collude for traffic analysis purposes.

PADRE: You know, Steve, I've always heard about the exit nodes being the real vulnerability of Tor because - in fact there was a demonstration that got canceled at Defcon two years ago. There were two engineers who they say they had figured out a way to compromise Tor for less than a thousand dollars. But this idea of having both the entry and the exit nodes in the same autonomous system, that's actually - I don't know why we didn't think of that earlier. I mean, you can notice traffic patterns if you have both ends.

Steve: Exactly.

PADRE: Doesn't matter what you do in the middle, if I know that I've got a clump of packets coming in here and a clump of packets going out there, because I own both spots. Both nodes sit on my network.

Steve: Right.

PADRE: I can make correlations much more easier than trying to compromise one of the exit nodes.

Steve: Right. And having to synchronize the traffic across completely different corporations owning different networks. It's much more difficult to perform the correlation that way. So what Astoria did, these guys building this test system, was they modified a standard Tor client to be smarter. It downloads the 9MB IP-to-ASN mapping database. You're able to download that. It's freely available from APNIC. So that's essentially a 9MB database, sort of like a master routing table, that allows you to determine who is the owner, the ASN, the autonomous system number owner, of any IP in the IPv4 space. So they use that database to intelligently select nodes when they build the Tor circuit.

They also require that nodes earn trust over time. They're suspicious of new nodes that have appeared on the network. It's like, eh, you know, maybe we don't want to route our traffic through you yet. We'll just kind of keep an eye on you. We'll send little, you know, test pulses through every so often to see how you're doing. They also monitor the amount of bandwidth that nodes have available so that nodes that would otherwise be preferentially chosen, but just don't have the bandwidth capacity, don't have circuits made for them. So it's a whole - it's a very complex algorithm that they put together, and with an attempt to increase security.

Now, the bad news is that they never really report in their paper what they were able to do. Reading between the lines, and like I said, wait a minute, where are the results? Why aren't they telling me, like, how much better they got? Because they certainly told us how bad it was early on, and the problem that they were going to try to solve. The sense I got was they were unhappy with the results they got, that is, they were not fundamentally able to dramatically improve the security of the Tor network, even choosing, like doing the best job they could of choosing nodes.

They said, under usability of Astoria, they said: "From our evaluation of Astoria, it is clear that the performance-security tradeoff is favorable only in its higher security configurations." That is to say that to get higher security really created a performance hit. "At high security configurations," they wrote, "Astoria is able to perform good load balancing, achieve reasonable throughput, avoid asymmetric colluding attackers whenever possible, and even handle situations where safe circuits are not available.

"However, at lower security configurations, the performance offered by Tor is clearly better, and its security only slightly worse. Therefore, Astoria is a usable substitute for the vanilla Tor client only in scenarios where security is a high priority, meaning that users would pay a substantial cost in performance. So it was an interesting piece of research, to explore how to make this thing better and more private and what the cost would be."

PADRE: A very interesting piece of research. But, I mean, naturally, if you're going to be deliberately jumping off of a single automated system, you are going to take a performance hit. You're going to be taking it inside of the encrypted tunnel, but you're still going to take the performance hit. The part of the research that actually surprises me more and has me more interested is the addition of the trust metric because they've used the performance metric in Tor. That's the whole idea.

In fact, that was one of the primary defenses against having an exit node that was compromised, which is it just wouldn't give you as good of a performance boost as you would get from a non-compromised node because you had to have someone sniffing the traffic. But I'm wondering if you could use the trust metric in today's Tor network without having to do the deliberate jump off an automated system in order to get more security without taking such a big performance hit.

Steve: Yeah, I don't know, I mean, their focus, at least, was the idea of avoiding staying within a single AS, or at least, I mean, the typical Tor circuits are three nodes. You've got your entry node, your exit node, and one in the middle, the feeling being that you're not getting substantially more security if you add additional interior nodes in the circuit. And I sort of lost my train of thought. You're not getting more security, and, oh, I know what I was going to say. And there is a substantial performance penalty as you make your Tor circuits more lengthy. So you're just losing performance without much security benefit. But their main deal was the entry and the exit nodes deliberately being in different networks.

And so the problem may be, you know, the point is you might have a server that is just like it's being served by Level 3. Level 3, like GRC, GRC is in a Level 3 datacenter. So if your entry node is in Level 3, and then you have a middle node also in Level 3, or maybe somewhere else, but the exit node is in Level 3, and then it goes to GRC in Level 3, that's going to be very efficient because essentially you're exiting in the same network that is hosting the content.

But if you force an exit in Australia, then you're deliberately sending traffic all the way to a different continent, and it has to come all the way back, you know, in a different continent into an exit node in some other ASN there. Then it exits, it has to come all the way back here. And we know from experience that, for example, CDNs are local. They have local presences so that you keep the network distances, the network paths short. So it's clear that what they're saying here is that, by making sort of what is an unnatural choice, you are really hurting your performance in order to gain additional security.

PADRE: Steve, this has been an episode of horror and eventually disappointment. So I'd like to end on an up note. Specifically, I want to talk a little bit about something I found in SpinRite the other night. I just updated my tool. It's been a while. It's been working for me, but I wanted to try out a couple of your new features. And there was this thing that popped up on a drive I was trying to fix where I could do partial recovery. I had not heard of that before.

Steve: Yeah. Actually, that's one of SpinRite's main claims to fame. It turns out that, as we know, SpinRite excels at data recovery. It will go crazy and use all kinds of tricks in order to successfully recover the 4,096 bits, the 512 bytes in a single sector. And in fact, you know, sometimes people, it's like, okay, how long is this going to take because I don't need my data that much. It's like, well, SpinRite's going to work at it until it either decides it absolutely cannot recover it, or it does.

But there's something, there's a trick that I developed way back in the beginning of this that surfaces first of all in the DynaStat system. The DynaStat system SpinRite owners with damaged drives see, where it's a statistical analysis of the data that SpinRite is recovering from unreadable sectors. And that's the key. If you think about it, you know, you're seeing the sector's data in that DynaStat readout. But that sector has never yet been read correctly. If it could be read correctly, our job here is done. We're finished.

So what happens is that allows SpinRite to build a profile of the missing data. And if SpinRite finally determines that it is unable to ever get the drive to agree to give it that sector just one last time, and the green R's that people are always seeing means the drive said no, and SpinRite forced it to finally say yes. But there's one thing SpinRite can do which sometimes is what you need, and that is, it will give you most of the data in a sector as opposed to giving you none of it. The drive will give you none of it, period. I could not read the sector, sorry. SpinRite says, well, you've got 4,096 bits. And maybe 12 of them cannot be read. But the rest are here. We've got the rest. Isn't that better than none?

And it turns out, even though some people may say, well, no, I want it all, well, yeah, we do, too. We'd like to give it all to you. But if it absolutely, if the damage is so great that it is beyond SpinRite's ability to pull off a full recovery, it will approximate. And the reason this can be useful is in a couple ways. There are, for example, database files that will not open if a sector is unreadable at a critical part of the database. The entire database is then offline and inaccessible, even though only one piece of it is a problem. Or the directory system in a large file system, if something in the directory system is broken, the system says, sorry, can't read this sector. And so you lose everything downstream, everything down the file system tree from that point, unless SpinRite gives you most of the sector. Or say you want to image your drive. Drive imaging tools stop the instant they find an unreadable sector. And so you can't image any of your drive because you can't get a perfect image.

One of the ways that we sold this a lot back when Microsoft was moving from the FAT16 to the FAT32, during the introduction of the FAT32, there was a converter program that Microsoft had that would run to convert your older format storage to newer. If it hit a bad sector, it would abort and say, sorry, could not perform the conversion. We sold a lot of SpinRite back then because you ran SpinRite on the drive, it fixed the problems, and then you were able to do the job. So one way or another, SpinRite leaves your drive in perfect condition. It may not have been able to read every last bit. But it gives you every bit that it could read. And when a sector has 4,096, missing a few can be okay, if you get the rest. And that's one of the reasons SpinRite is able to pull off so many miracles.

PADRE: Steve Gibson, of course, is available at GRC.com, where you'll also be able to find the audio version of Security Now!, in case you want to get it into your player, along with the show notes. If you want to find out what's been going on during this show, if you want to follow along, it's a great place to go. Of course, GRC.com is also the place to get SpinRite, the tool of choice. I tell people this all the time. In fact, I had Steve on my show This Week in Enterprise Tech yesterday. And my motto is, if you don't have SpinRite in your toolbox, you don't have a toolbox. Of course you'll also find ShieldsUP!. You'll find his very soon coming SQRL, I believe, the revamping of security authentication as we know it. Steve, do you want to give us a little update on that?

Steve: Actually, I have big news, but we're running at two hours and 22 minutes at this point. So let's defer that because there was a breakthrough actually that I came up with a couple days ago to solve one remaining problem. And it hasn't been implemented yet. But I figured we'd maybe talk about it in two weeks, when I ought to have - maybe next week. We'll see. But, so, yes, you know, I do want to share a little bit about that.

PADRE: Mr. Gibson, he is my personal security guru. I go to him with all of my security questions. Of course, you should, too. Don't forget that you can catch Security Now! every week, Tuesdays, 1:30 p.m. Pacific time. And you can find him on Twitter. Do you push your Twitter address? I always forget.

Steve: Oh, yeah, yeah, @SGgrc. There it is there in the lineup. I've got an @SGpad, an @SGvlc, for "very low carb." Those are not used very much. So @SGgrc is where I hang out on Twitter. And I will note that next week we're going to do, since we do every other podcast is a Q&A, next week is a Q&A episode. So by all means, send me some thoughts and feedback. Go to GRC.com/feedback. There's a web form there. You can drop your question in. I check the mailbag a day or two before, and often that same morning, to pull a bunch of questions which the Padre and I will go through next week.

PADRE: And it will be my absolute pleasure. Until then, I'm Father Robert Ballecer in for Leo Laporte. We'll see you next time on Security Now!.

Steve: Thanks, Padre.