SECURITY NOW!

Transcript of Episode #521

# Security Is Difficult

**Description:** Leo and I catch up on another in a series of very busy weeks of security news. Then we discuss several recently written commentaries about the distressing state of online web advertising.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-521.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-521-lq.mp3

SHOW TEASE: Hey there. It's time for Security Now!. Steve Gibson is here. And I know we promised you a Q&A episode; but, man, there is a lot of security news. We've got the latest, including StageFright. It's back. Stay tuned. Security Now! is next.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 521, recorded Tuesday, August 18th, 2015: Security Is Difficult.

It's time for Security Now!, the show that protects you and your loved ones and your privacy and everything about it online with this guy right here, the Securer in Chief, Mr. Steven Gibson, as we begin our 11th year.

**Steve Gibson:** Yes.

**Leo:** No, actually, is it - wasn't it...

**Steve:** I actually think it is because it was last - it was August 19th of 2005.

**Leo:** Right, and this is August 18th. So really we're exactly 10 years old, like, tomorrow. And we'll begin our 11th year - this could be reasonably considered to begin our 11th year.

**Steve:** I think this is it, especially since we're at 521.

**Leo:** Right.

**Steve:** And if it were tomorrow, then we would have finished our 10th year, and this would be the first episode of our 11th year. But it's not tomorrow, it's today, so...

**Leo:** You realize that this is the only show that worries about this in any way.

**Steve:** This is as close as we could get to exactly.

**Leo:** Good enough. Good enough.

**Steve:** Yes. Plus or minus one day. Or actually half a day, and so we landed on it.

**Leo:** Now, we were going to do a Q&A; but, once again, current affairs intervene.

**Steve:** As I was saying to you, sometimes the show notes are three pages. Today we have 11.

**Leo:** Yikes. Yikes.

**Steve:** So a lot to talk about. And really, this is another one of these where I'm going to encourage people to follow up if they're interested in the topic because some of these are - for example, the coverage The New York Times gave to AT&T's abundance of cooperation to the NSA, we'll summarize it, just so we've covered the topic. But their story, The New York Times story, goes on and on and on. I mean, with interesting stuff, individual program names and all that kind of stuff.

But anyway, so there are many - the show notes are what they're intended to be, which is something that you can use to follow along. But also, when I refer to something, and I say, okay, you know, there's a link here, so anyone who wants to go deeper - because otherwise it would be a 24-hour podcast.

**Leo:** Yeah. Summarize. That's what we ask for anyway.

**Steve:** Yeah. So we're going to talk about StageFright, which has had a regression, unfortunately, a really interesting one. And I'm going to take a little bit of time to talk about the nature of this bug and how it involved a mistake in the declaration of two variables, an unsigned 64-bit integer and an unsigned 32-bit integer, and what happened there. We're continuing to get to know Windows 10. As I expected, I mean, any new operating system, it takes a while to get to know it. And so there's some additional privacy flags that we want to cover. Some very high-profile malvertising has surfaced, and with enough depth to them that I thought, oh, you know, we've never really talked - we've talked about they're bad, but we haven't really looked at, like, what the mechanisms are. So we have enough, now, material from two different campaigns of malvertising to talk about that.

Then, of course, Kaspersky, Lenovo, HTC, and AT&T, each in their own houses, we'll talk about those problems. We have some tidbits. And then I want to wrap up, because three pieces have been written recently on the issue of advertising, one by our own wonderful Rene Ritchie, and the whole iMore advertising sort of debacle that you guys may have talked about. I don't know whether you have or not. But there was a great piece that was written after the Worldwide Developers Conference, like, what, six weeks ago? A well-known blogger who writes the Daring Fireball blog, Gruber...

**Leo:** Mr. John Gruber, yeah.

**Steve:** John Gruber. He talked about the experience a friend of his had playing with the adblocking that's forthcoming in iOS 9. And John wrote a very succinct little piece that I am going to share. But then there was an interesting post from someone who's been in the business forever, who's a podcaster and uses ads to support himself, who reminded me that we already went through something similar in the era of pop-ups. And which sort of wasn't even on my radar. I forgot about those because they were just so awful.

But anyway, so I want to wrap up talking about that. And also Rene's position because, of course, we know how great iMore is. And it sort of helped me to better appreciate the dilemma that they're in. And it also shows sort of how different TWiT is by virtue of the fact that you have so much control, we have total control over the advertising that you do during the podcasts, which no website can practically manifest, for lack of a better word. So anyway, lots of, I think, really great podcast for everyone today.

So the picture of the week I got a kick out of.

**Leo:** I love this.

**Steve:** My sister sent this to me yesterday.

**Leo:** This is so funny.

**Steve:** It is just wonderful. And I thought, my god, there might actually be such a corner. Because, for those who are not seeing the video and haven't seen the show notes, this shows two street signs, apparently at the corner of Live Long and Prosper. However, the hand in the Don't Walk sign gave it away.

**Leo:** I think somebody might have - that might actually be a real picture, like somebody might have put a little tape in there on the hand or something.

**Steve:** Yeah, it looks like they, I mean, that would be neat, if true, although it looks like the dots were probably lassoed and, like, pulled in order to give us the Vulcan hand sign. But anyway, I just got a big kick out of Live Long and Prosper and an intersection of two streets.

**Leo:** Love it.

**Steve:** Great little picture for the show. So StageFright. The good news is all throughout last week I was getting tweets from people confirming that their various carriers were pushing out patches to their phone. Ernest Koch, he tweeted from @nullconmedia and said: "@SGgrc After yesterday's patch, my Nexus 6 is showing not vulnerable." Our friend of the show Simon Zerafa confirmed that both his Nexus 6 and 7 were just updated. It doesn't mention who his carrier was, but says "gets six greens on the Zimperium StageFright test." And Leo, you...

**Leo:** Now, I've never been able to get six greens on Zimperium, by the way.

**Steve:** I was going to tell you that you're going to have to update Zimperium.

**Leo:** Ah.

**Steve:** Because there's now a seventh, as a consequence of our next piece of news, yes.

**Leo:** Oh, come on. So the Lookout one, which I just used, does say "not vulnerable, congratulations." But Zimperium on this, this is a Note 5, just came out; right?

**Steve:** Yeah.

**Leo:** So this should be the most, I mean, up to date.

**Steve:** Well, it's probably been sitting on a shelf somewhere, though. Don't you think they probably have inventory for a month or two?

**Leo:** Yeah, I mean, I was very pleased at the report. But now I've got to get the new Zimperium; right?

**Steve:** Yeah, you ought to. What does it show at the moment? Does it show - is it, like, three out of six?

**Leo:** You know what, I can't find it. I must have erased it. It showed five out of six, I think.

**Steve:** Okay.

**Leo:** But the first one, it was not - the first of the exploits was not passing, was red.

**Steve:** Okay. I think what they do is actually send test text messages. So I am feeling, although they had a bit of a rough start - they weren't coloring them black, or red and green, they were all white initially - they fixed it. So anyway, so basically, other people essentially throughout the week were saying...

**Leo:** See? First one's still red. So it's vulnerable. All green except for 2015-3864 is the problem one.

**Steve:** Ah. Now the question is - eh, yes. And that's the new one.

**Leo:** Oh, crap.

**Steve:** That is why we're talking about one step backwards.

**Leo:** Oh, crap.

**Steve:** Yup. So, now, that is exactly what I would expect most of our listeners to currently see because Google, I mean, because this just happened. So, and that's where I was talking, at the beginning of the show, about these two integer sizing problems where Google, well, Zimperium provided Google with the patches. Well, I'm getting ahead of myself. So I'll just say that one other person, Eric Throndson, noted that his Nexus 6 is receiving MMS messages from random numbers.

**Leo:** Oh.

**Steve:** "That I assume have StageFright," he wrote. "I'm patched, but nervous and annoyed." And this is what - I've had several reports of this exploit, the StageFright exploit now in the wild, from - and obviously this is people receiving MMS messages where three weeks ago, before this came to light, they weren't. And so these are bad guys sending out text messages to maybe random people, maybe targeted attacks. We're not really sure what the profile is yet. But of course this is a problem. And now we know that most of the phones are using, as you and I have discussed, address space layout randomization, which makes attacks more difficult, but not impossible. So the right solution is for these things to get patched.

**Leo:** Now, I have turned off "auto receive MMS" in the Hangouts app, which is my default.

**Steve:** Good.

**Leo:** So that's the best remediation for now, until we get a patch; right?

**Steve:** And so then what that means is - does it queue them so that you...

**Leo:** I don't know. And that's why I was curious about this guy who's getting them. Is that what he did, or what's going on?

**Steve:** That's a nice-looking phone, Leo.

**Leo:** Yeah, the Note 5. It's purty. Purty.

**Steve:** And that's stylus-based? It's got a stylus?

**Leo:** Yes, sir. You just pop out that stylus, right out of the bottom there, and then you can do all sorts of things like - you know what's a really cool one, I should show this, is if you pop out the - I'll put the stylus back in. If you pop out the stylus, because it's an all-lit screen, it allows you to write on a darkened, on the turned-off screen.

**Steve:** Oh, nice.

**Leo:** So you could still take notes. And, by the way, they've really reduced the latency. You can't quite see it because it's so - just kind of the lights in here.

**Steve:** Nice, nice.

**Leo:** But I'm writing; you know?

**Steve:** So super low latency. That's the thing I care about most with a stylus.

**Leo:** Perfect. It's perfect.

**Steve:** If it keeps up, that's nice.

**Leo:** You would feel like you're drawing right on the screen. I mean, it's instant. And the fact that it'll do this with the thing off is awesome.

**Steve:** Yeah, and it's a non-powered stylus? There's no battery or anything in the stylus?

**Leo:** No battery. It does have a button that you can click.

**Steve:** Okay. Okay, so it is active.

**Leo:** It's active in that sense. But there's no battery.

**Steve:** Oh, okay. Oh, good.

**Leo:** I don't know what the button does.

**Steve:** So in that case it's like the WACOM technology. It probably is - it changes - there's an inductor...

**Leo:** Right.

**Steve:** ...which is oscillating, and the button changes...

**Leo:** So they're creating a little bit of - yeah.

**Steve:** It changes the frequency of oscillation. Very nice.

**Leo:** Yeah. They've done a nice job. I have to say, you know, they've gotten better and better. I know you're an iPhone guy. But, oh, fingerprint reader works really well. You don't have to swipe anymore, you just touch it, and I like it.

**Steve:** Nice. So the bad news is that a different group than Zimperium, Exodus Intelligence, took a look at the patches which Zimperium designed and gave to Google, which Google accepted. And I don't know if they were staring at them, or if it just jumped out at them, but the patch for one of the six critical vulnerabilities, which was 3824 - so it was CVE-2015, for this year, dash 3824. That one was three lines added to the source code of the StageFright library for processing - the specific problem was called the "tx3g MP4 atom integer overflow." And they added three lines, basically checking to make sure - their intention was to check to make sure that the sum of two sizes, a size integer and something called the "chunk size," were less than a maximum size.

And the maximum size turns out to be the maximum size of a 32-bit integer. That is to say, in hex, eight F's - FFFFFFFF, all ones. And so if you have unsigned math, meaning that the quantity that you're storing in this four-byte variable, it's considered to be non-signed, which is what you'd want that makes sense for the size of something because you can't have a negative size. So in this case the size max constant is all ones, 32-bit, all ones. The problem is that the chunk size is derived from a 64-bit piece of metadata at the front of the MP4 file. So it's defined in the source code as being a 64-bit unsigned value.

But it's being compared with - the way they did this "if" statement is odd. They said that the max size minus the chunk size is less than or equal to the size. Now, it's like, okay. I mean, that seems weird because - but it's arithmetically correct because, as we remember from manipulating equations, if you were to add chunk size to both sides of the inequality, then you'd get - that essentially moves chunk size to the other side and switches the sign from minus to plus. So that says size max is less than or equal to size plus chunk size, meaning that there will be an error if size plus chunk size is greater than or equal to size max and generate an error.

Well, it turns out that, due to the odd way that this was phrased, and the way the C compiler handles different type sizes, this doesn't do what the fixers of this patch believed it does in boundary cases. And of course these are always tricky. So if chunk size is 1FFFFFFFFF, that is, one additional bit, which it can be because it's a 64-bit quantity, and you subtract that from the constant size max, which is a maximum 32-bit value, which is all F's, then that ends up not correctly comparing against the value of size. Meaning that this error, which should absolutely fire in that case because you've got chunk size is essentially 33-bits long, rather than just 32 - so it's like it's, what, would that be 8.6 gig, essentially, which you can easily induce just by changing this bit of metadata in the MP4, then this test absolutely never fires, although it absolutely always should.

So I wanted to go into some detail to give our listeners a sense for how hard this is to get these things right, and how multiple people can look at it and go, okay, yeah, that looks good. But where a bad guy or somebody explicitly trying to audit this for security can say, okay, now, wait a minute, that's a 64-bit quantity, which you're subtracting from a constant, so that'll be treated as a 64-bit quantity by the compiler, which will cast it to that size so that you get the same size in the math. But you're comparing it with a 32-bit quantity in a way such that it's not going to do what you intended it to do. So anyway, this was just ineptly written, unfortunately, by the Zimperium guys, who were trying to fix a problem that they found. They didn't fix it.

**Leo:** Isn't that ironic that...

**Steve:** Yeah.

**Leo:** I mean, there's a lot of irony there. But it also points out how easy this is to do. You forget these side effects. And that's part of the problem with side effects.

**Steve:** Yeah. Correct. Well, and - yes. And it's part of the problem of maybe, I mean, who knows what the history of this is, where, you know, how many cooks had their hands in this. You'd have to go back and check the source to see what the sizing was. Sometimes, when they're used, they're cast, because there's something in C called "casting," where you can explicitly say, okay, compiler, you know, I know what I'm doing. Treat this specifically as a 32-bit quantity. And in fact, there is a casting later on that causes the chunk size plus the size - what ends up happening here is that chunk size is added to size, and that's cast to a 32-bit quantity to form the size of an allocation.

With this cleverly - with the metadata cleverly tweaked, that ends up allocating an undersized buffer, and then you get a buffer overrun. Because you can end up creating a very small buffer. Then, when other logic assumes that the buffer is the proper size, because after all it just allocated it, you know, knowing what the sizing metadata said, when other logic reads the same sizing metadata using a different, subtly different logic, it overflows the small buffer. You get a buffer overrun, and that allows you to put your data from the MP4 onto the heap. And then, when that subroutine returns, it pops registers and return addresses off of the heap, or the stack, depending upon how it's organized. And now you're able to, if you've designed it properly, execute your own code.

So as you said, Leo, it demonstrates - and that's why, you know, it was this that caused me to name this podcast "Security Is Difficult," because the problem is we have to get everything exactly right. And, you know, these are smart people writing this code. I mean, to do any of this you have to really know your programming. But at the same time, this is one of the arguments against C moving forward in the future. It was designed to give programmers power, not to get in their way. It assumed that, you know, a certain mindset, a lot of experience, it assumed that even though programmers were programming in C, a higher level language, that they absolutely understood things like the word size of the system that they were writing to, that they understood the subtleties of signed versus unsigned values and integer overflows.

They were, even though they were, I mean, and this is all stuff, as an assembly language programmer, you live there. I mean, that's, you know, you're right there. There's no abstraction allowing you to do, oh, add these two things together and give me the answer. You're responsible totally for the sum fitting into the destination register in assembly language. C, the problem is it seduces you by saying, oh, look, you can do algebra. But none of the responsibility of having the answer come out right has actually been removed from you. You're still there.

So the argument is, it's dangerous to write in C. And here, you know, here some other code was then taking the sum of these and allocating memory, which is another dangerous thing to explicitly do. The benefit is C doesn't get in your way. But with it comes tremendous responsibility. And so it really takes somebody who's fully amped up on whatever your source of caffeine is all the time. And you just - you can't afford to blink. Now...

**Leo:** This is why, by the way, there's kind of this move back towards functional programming, because it doesn't have side effects.

**Steve:** Right. Now, what's a little controversial is that these guys went public with no notice. They just said, hey, this one is wrong. Now, they're arguing that it was Google's fault for missing this, that they already had used up 120 days since they were informed of the bugs, and therefore they were not due any grace period over the fact that they didn't catch the fact that Zimperium miswrote the patch for what was 3824.

So it's like, eh, you know, it is the case that Google has been missing their own draconian, 90-day, we're going to reveal the bugs whether it's fixed or not after 90 days. And we have a little bit of coverage of that here in a second because this has been a - and we've talked, we were talking about, it was like last year I think sometime where we'd spent some time talking about the fact that Google just says, sorry, you didn't fix it in 90 days, we're going public with it whether you like it or not. Which has, you know, really holding people's feet to the fire.

Well, now Google's feet are being similarly held to the fire. So, and again, here's the problem is that Zimperium provided Google with patches. Very likely, if they'd said these are the problems, a Google engineer would have done a different patch...

**Leo:** Yeah. Maybe a better patch.

**Steve:** Right. Exactly. Google would have - a Google engineer would have done it correctly. But in assisting Google, helping them...

**Leo:** Too much help.

**Steve:** Yeah, exactly.

**Leo:** But wouldn't a Google engineer looking at the patch see what the problem was, I mean...

**Steve:** No, see, that's what's so funky about coding. And we've talked about the phenomenon in debugging where you look at your source. And, you know, you wrote it. You're proud of it. Or with the source code comes sort of this - it brings its own assumption of correctness. And what's so interesting about debugging is you look at the code. And you've stared at it before you finally gave up and fired up the debugger. You step the debugger through. It's like, that's good, that's good, okay, here. And you look at what it's about to do. And then it steps on it, and it comes up with the wrong answer.

**Leo:** But you would think they're getting a massive hint from the code merge file or whatever that Zimperium gave them.

**Steve:** Well, no, no. I mean, no doubt the Google guy looked at this and thought, eh, that's not the way I would have phrased the solution. But, you know, okay.

**Leo:** It works.

**Steve:** And who knows if they were - if it was after 5:00, and the guy's wife was, like, sending him text messages. I mean, we just don't know the back story behind how this was processed. It did take them quite a while to get this, I mean, they were notified four months ago and didn't get this thing fixed until, well, I guess they did fix it in their own code. But then there was the delay in pushing it out to the various carriers such that no carriers were pushing this before the alarm bells started going off after, you know, thanks to the Black Hat reveal that occurred.

So anyway, security is difficult. It's just, it's sad, but it is. And this is a problem that no one wants to have. There are other problems which are, by policy, that was my old complaint about XP and raw sockets was that Microsoft was saying, yeah, we intended to do this. It's like, oh. No one intended to have a buffer overflow in MP4 file processing. And so the upshot of all this is there is a seventh problem, 3864. And our listeners should update their Zimperium test. I did mine on my little Galaxy S something. And sure enough, it added another line, and it's red, too.

**Leo:** [Exasperated sound]

**Steve:** So, and in your case, Leo...

**Leo:** I'm all green except for the red.

**Steve:** Yes. And the problem is this is exploitable. This is a remotely exploitable buffer overrun, which we still have, even after all the patches have been applied, because this one there was no notice for, and I'm sure Google will get on it and fix it as soon as they can because now we've got tests in everybody's hands saying, hey, look, I've got six green lines, but that seventh red one, that's the one that's going to bite me.

**Leo:** And by the way, I should point out there is not an updated Lookout tester. So if you've used the Lookout tester - my Lookout tester says I'm fine. It's only the updated Zimperium one that shows this extra exploit.

**Steve:** Right, yeah. And I intended to say earlier, but I think I didn't finish my thought, which was that they had a bit of rough start in that their first tester wasn't catching - it wasn't - I don't remember. It explained it a couple weeks ago. I think it was that, first of all, it wasn't giving you individual itemization. They were all showing white. And it would just say at the bottom, "You're vulnerable." Then they fixed it to show the individual vulnerabilities. And to me, this now looks like they've got it nailed. I'm very pleased with their tester. I just wish that they fixed that first problem correctly the first time. Then we wouldn't have now a number seven, even after we went through all this. Because apparently it's a huge problem for the various carriers to push these fixes out.

**Leo:** And there are exploits in the wild; right?

**Steve:** Yes. I mean, this is why that guy was reporting he's receiving MMS messages. Something is trying to take over his phone. And he's like, oh, I'm fully patched. Well, now, he's not fully patched. Nobody is currently full patched.

**Leo:** I'm not getting any MMS messages, so I don't know. On any of my phones or my numbers.

**Steve:** As we know, there's 950 million cell phones, Android phones out there. So they just probably haven't gotten around to you yet, Leo.

**Leo:** Okay. Good. That's a relief.

**Steve:** So there's something called Google Admin, which Google describes it as - it's by Google for Android - lets you manage your Google for Work account on the go. Add and manage users and groups, contact support, and view audit logs for your organization.

And Google says: "For whom? This app is only for super administrators of Google for Work products, including Google Apps for Work, Education, Government, Google Coordinate, and Chromebooks." And Google explains it provides the following features: For user management, you can add or edit a user, suspend a user, restore a user, delete a user, and reset a password. And then for groups you can add and edit groups, add members, delete a group, view group members.

So the good news is this is not for everyone. The bad news is it is apparently widely used in enterprise settings where you would have an administrator for Google for Work products, and a bypass of its sandbox, of the application's sandbox, has been found. So, and this is another one where, for reasons unknown, Google has dragged its feet. And finally the people who found the problem have fully published the information, even though there's no fix out. Yet there's also, as yet, no known attack in the wild.

Essentially what this allows is any malicious application that arranges to get onto the phone of probably a power user, who does have the Google Admin application installed, which again is not typically on people's phones. It's something you need to go and get if you are one of the people. The malicious app has a way of tricking Android and, as a consequence of the way of some configuration of this Google Admin application, to read any of the files in the Google admin application sandbox. Which is not good because this is typically really confidential power admin enterprise-class stuff.

So this was found back in March. Google was notified and acknowledged the problem. In May, these guys at MWR, who found the problem, request an update from Google, who acknowledge, Google acknowledges they have exceeded their own 90-day disclosure deadline and ask for an extension until July. Note that, even though Google doesn't give extensions, they're asking for them on their own policy. July comes and goes. These guys give Google another month after the disclosure delay, which was given, and this month in August, after five months with no results, informs Google of their intent to disclose, and do so. So again, this isn't a huge vulnerability. But it's there. It's out there now.

So if you are an Android user with Google Admin, you no doubt know you are. I wanted those people who are listening to know that this problem has been released. No doubt people will start - and the problem is this is the kind of thing that would hit you with targeted attacks. If there is a way to get some malware onto your Android device, and you are this kind of user, then this looks like it's a problem. If you don't need this on your phone, you might just want to remove it until Google has it fixed. Or, if nothing else, keep a sharp eye out for an update to this. I imagine, now that it's public, Google will change their priorities around. We don't really understand, again, how they're prioritizing things except we are seeing now sort of a series of examples of them not meeting their own 90-day disclosure policy.

Ars Technica, as I mentioned, getting to know Windows 10. This is something that inherently takes a while. Two weeks ago, Leo, we got a tremendous amount of positive feedback from our listeners, saying that they really liked our walkthrough of Windows 10 privacy settings. I mean, it wasn't rocket science, but it really did seem to, like, it's what people wanted to hear and to understand. Now that we've lived with Windows 10, and are living with it for a few weeks, people have been taking a closer look at it. For example, if we absolutely turn everything that we can find off, that is, to its most privacy-enforcing settings, how does Windows 10 perform? And unfortunately, it cannot resist talking to the Internet.

Now, I'm not taking a position on this. I'm just reporting. And the good news is I'm not worried about this because you'll never get me to use this flying turd. But Ars Technica has taken a very close look at it. And they said: "With Cortana and searching the web from the Start Menu disabled" - and everything else basically completely shut down - they observed that "opening Start and typing will send a request to www.bing.com to request a file called threshold.appcache, which appears" - and I think they were actually using - shoot. I'm blanking on the name of the sniffing tool that everyone's - I have a sniffing tool that I use. There's a very popular one, begins with "F," and I can't think of it. It's not Fisher. It's not Finger. Anyway, I looked at the screenshots of the information they were caching.

**Leo:** Not Wireshark, was it?

**Steve:** Not Wireshark. Boy, I even have it installed. I installed it the other day for a different reason. I probably can see it here. I'm not seeing it. I might have already removed it. Anyway.

**Leo:** Fiddler.

**Steve:** Fiddler. Thank you.

**Leo:** Thanks to Digimax in the chatroom. Fiddler.

**Steve:** Fiddler it is, yes. Fiddler is a tremendous local proxy which is interesting because it is able to crack HTTPS connections also by asking your permission. And it makes it very clear that it's going to do this, but it puts its own certificate in your local store, just like the various AV tools that we've discussed, and Superfish, unfortunately, also did, in order to have visibility into your SSL connections, or TLS now, and see what's going on.

So Ars Technica did this, and they discovered that this threshold.appcache "appears to contain Cortana information, even though Cortana is disabled." And, a little worrisomely, they noted that the file appears to contain a persistent machine identifier which persists across reboots. So it's not an instance token for this use, but it appears to uniquely identify that particular machine. And the point being no controls that Microsoft has given us turns that off. And before I get into a little bit of commentary about that, I wanted to cover the other two things that Ars found. They considered this not a big deal. But they said some of the traffic that Windows 10 continues to exchange looks harmless, but feels like it shouldn't be happening.

For example, even with no Live tiles pinned to the Start Menu - so they, like, they really shut this thing down. They removed the tiles from the Start Menu. And they said, "and hence no obvious need to poll for new tile data," Windows 10 nevertheless downloads new tile info from MSN's network from time to time, using an unencrypted HTTP connection to do so. While the outgoing requests for the information contain no identifying information, it's not clear why Windows 10 is doing this, given that they have no tiles to update. And, okay, and I want to finish before I get into this. And finally they said...

**Leo:** And before Paul Thurrott calls you an idiot. Go ahead.

**Steve:** Yes. Although Paul is liking Firefox. So he's moved back to Firefox, which I appreciated.

**Leo:** Yes.

**Steve:** "Other traffic looks a little more troublesome," wrote Ars. "Windows 10 periodically sends data to a Microsoft server named ssw.live.com. This server seems to be used for OneDrive and some other Microsoft services. Windows 10 transmits information to the server even when OneDrive is disabled and logins are using a local account that isn't in any way connected to a Microsoft Account." So, I mean, that would really - that really seems odd because a local account is just you on the machine.

And by the way, I had to create one the other day for SQRL testing, and it is difficult to, I mean, Microsoft wants your username, and then email address. And then you say, no, I'm not giving you that. And it says, what? You don't have email? Oh, let's give you an account because everything is better with a Windows account. And it's like, okay, just get me out of here. And I didn't even see a way to do it until someone in the newsgroup said, okay, now, if you go all the way out to that screen, then you say you want a non-Windows or non-Microsoft account, or you don't have email, you don't want to create email, like three screens in you can finally say, okay, just give me an account.

Anyway, the point is, if you go through all that, explicitly refusing to hook into the Microsoft online world, OneDrive or something is still sending data out. Ars wrote: "The exact nature of the information being sent isn't clear. It appears to be referencing telemetry settings." And again, it's not clear why any data is being sent at all since they even went so far as to disable all telemetry in the group policies, which is like the deepest level of enterprise OS configuration.

So my take is that maybe these are oops things. Maybe not. Again, if you're using Windows 10, you are using what is arguably a deeply cloud-connected operating system. That's just what it is. And again, I think maybe, if Microsoft feels that these are just things they missed, then in the future they may update it. If there are people who absolutely want a non-communicating version of Windows 10 to be possible, then maybe there'll be, like, the equivalent of airplane mode that we have on our phones and communicating tablets, which makes it very clear, okay, I want to be in airplane mode with my operating system. We don't have that yet. Despite Ars Technica's best efforts, Windows 10 is just reaching out and touching the Internet a little bit.

**Leo:** To be fair, I think they said that we don't see any harm in this; right? In other words, it wasn't...

**Steve:** Oh, yeah. Yeah, yeah, yeah. They're just sort of observing. None of this, you know, they did...

**Leo:** Well, for instance, when you start typing in the Cortana window, Cortana is doing, in many cases, doing a Bing search. If it can't find it locally, that's a Bing search window; right?

**Steve:** Unless you've got Cortana completely disabled. And they had it completely shut down. So there should be no use of Cortana at all.

**Leo:** Well, then it's the built-in Windows search, which still uses an Internet search on that. So you'd have to...

**Steve:** Okay.

**Leo:** I mean, yeah. I mean, anyway, was any identifiable information passed out over that network?

**Steve:** No, just a unique ID.

**Leo:** Yeah.

**Steve:** You know, just a personal identifier.

**Leo:** But a unique ID is you.

**Steve:** Yeah, exactly. Well, it's your machine, or it's you. You know, we don't know.

**Leo:** Although I should point out your MAC address, doesn't that - I guess your MAC address is not visible on the public Internet.

**Steve:** Correct, because MAC is only for local ethernet.

**Leo:** Your IP address is, but not your MAC address, yeah.

**Steve:** Yeah. Again, I'm not saying this is a problem. I just wanted to acknowledge that, you know, I'm not surprised, which is why I don't think Paul will call me crazy.

**Leo:** No, no.

**Steve:** I'm just, you know, this is - Windows 10 is a connected operating system.

**Leo:** Well, and I'll be honest, I think the number one reason for this is antipiracy. It's activation. Windows 10, all versions of Windows since Windows 7 periodically call home to verify that this is a licensed version of Windows.

**Steve:** Good point.

**Leo:** That number might well be your activation code.

**Steve:** Yeah.

**Leo:** They're seeing, is this a legitimate copy of Windows? And it may well just be antipiracy measures. Fact, I bet you it is.

**Steve:** Yeah. And to demonstrate my approach, the next thing I have on my notes here I call the "Win10 Privacy Reality Check." And our friend Simon Zerafa retweeted something that a Fenrir tweeted, which I got a kick out of. Quote: "'Windows 10 is a privacy nightmare!' they said on Twitter, while using Google and updating Facebook."

**Leo:** Yeah. Kind of my point.

**Steve:** Exactly.

**Leo:** I mean, if you really want to use it in airplane mode, disconnect it from the Internet, and I guarantee you you're not passing anything to the network.

**Steve:** Yeah. Turn off WiFi and, yeah.

**Leo:** Just don't use it, I mean, honestly, if that's what people are worried about, why are you connected to the Internet on that machine? Just disconnect.

**Steve:** Or why are you using Windows 10? I mean, that's my position is...

**Leo:** Okay. So let's say I use Linux, and I go out, and I surf the 'Net. Are you saying that information leakage about who I am is not leaking out? Not from Linux, but from my browser, from supercookies.

**Steve:** Yeah. It's different. It's different, Leo. The operating system, you know, is for maintaining a file system and running applications, not for itself...

**Leo:** Well, then, disconnect the Internet.

**Steve:** No, just don't use Windows 10. Now we know what Windows 10 is. Windows 10 is Microsoft going into social networking and live tiles and, you know...

**Leo:** No, but you can use it as you wish to use it. If you unplug that ethernet cable, it's an operating system. It gets you access to files. It runs programs. It does everything you've just described. You're the one connecting it to the Internet.

**Steve:** That's another impractical solution. That's not practical.

**Leo:** But the minute you go on the Internet, whether you're using Windows 10 or anything else, you're announcing yourself to the Internet as an individual; aren't you?

**Steve:** Yeah. I mean, I get it. But the point is you can also just not use Windows 10. There's no need to use Windows 10. It doesn't do anything that 7 doesn't do.

**Leo:** I would disagree with you, but that's - I think Windows 10 is significantly improved. But...

**Steve:** Okay.

**Leo:** If you're worried about privacy, all you have to do is use whatever operating system you want and not go online, period. That's how you solve that.

**Steve:** Right.

**Leo:** And if you say, well, no, I want to go online, well, then, what are you worried about privacy for? You're going online. Right? What, do you go through Tor? No, that doesn't work. Well, let me think.

**Steve:** But you don't want me to argue with you, do you?

**Leo:** You can argue with me, yeah.

**Steve:** I mean, you're making no sense at all. It's nonsense.

**Leo:** Okay.

**Steve:** So, no, I just - all I'm doing, I'm just saying that what Ars reported...

**Leo:** But what I'm saying to you is, you should be arguing against using any Internet-connected device.

**Steve:** I'm arguing against the idea that an operating system has become a data-gathering, social networking platform. I mean, Linux is looking increasingly attractive to me, as is the Mac OS, neither of which are doing that yet at the OS level.

**Leo:** Really? I don't know.

**Steve:** Yeah.

**Leo:** I'm sure Linux isn't. But really, you think OS X's not doing that?

**Steve:** No, I've seen no indication of it.

**Leo:** Okay. Yeah, I mean, to be fair, I guess Apple doesn't need to because it has no antipiracy measures. It doesn't have to phone home.

**Steve:** Right. Well, and it's not, you know - yeah.

**Leo:** To each his own.

**Steve:** To each his own. I will never use Windows 10. So I think it's great if people want to use Windows 10. That's just - it's not for me. Which is not surprising. I'm coding everything in assembly language, so.

Okay. So the problem with malvertising is growing. In June and July we've seen record-setting attacks that have been launched through malicious ads. Yahoo! was notified by Malwarebytes - and of course Yahoo!, we know, is super popular, about 6.9 billion visits per month. They were serving, the Yahoo! pages themselves were serving malware. Yahoo! was notified and responded immediately. And in this case it was a chain of JavaScript which successively loaded. With each retrieval of JavaScript, that contained another retrieval, so it created sort of a chain of scripting. In every case, the original invocation was ads.yahoo.com. And then that fetch ran JavaScript, which then fetched some script from adslides.rotator.hadj1.adjuggler.net.

And so clearly that is script, which is given away by the name "rotator." So it's apparently doing some, like, choosing what it's going to show you. Then that pulled some content from azurewebsites.net. And all the details here are in the notes. I won't go into the whole URL. And then that pulled from basestyle.org, which finally pulled from the malicious domain, which was some crazy domain name. I would say that it looks random, except it has a forum on it.

And the interesting thing about all of the malvertising that this particular campaign has been using is that it looks like ultimately the malicious content is hosted on a forum posting, which is obviously not well moderated. And so it links to that forum posting, back up through this chain that end up where that script finally - so this forum posting contains script, which finally runs on the visitor who receives this ad, in this case when they were going to Yahoo!. And what it ultimately leads to is an exploit kit called Angler, which has been found to be dropping, in some cases, ad fraud, which I guess is just like, you know, ad clicks generating revenue, but also the CryptoWall ransomware. So, and we've heard that ads are delivering CryptoWall without the details.

And the Malwarebytes guys finally said, okay, look, this is what - we've been tracking these guys for months. This is what we've been seeing. And in their blog they write that "Malvertising is a silent killer because malicious ads do not require any type of user interaction in order to execute their payload. The mere fact of browsing to a website that has advertisements, and most sites, if not all, do," they write, "is enough to start the infection chain. The complexity of the online advertising economy" - which is what we'll be talking about a little bit at the end of this podcast - "makes it easy for malicious actors to abuse the system and get away with it. It's one of the reasons why we need to work very closely with different industry partners to detect suspicious patterns and react very quickly to halt rogue campaigns."

So they were very pleased with Yahoo!'s response. And then they just posted a couple days ago that that campaign that they've been tracking had moved, had changed advertisers, was no longer on Yahoo!, but went to AOL, that has the advertiser.com domain. But it's also still hosting on Azure. And now this uses, similarly, a chain of JavaScript-driven - although in this case they're SSL-tunneled fetches, making it more difficult to passively scan traffic and see what's going on.

The original URL was on eBay and was hosted by advertising.com, which is the AOL domain. It first redirected, through scripting, to azurewebsites.net, then to mbiscotti.com, and finally to two different scripts on two other random domains. So the Malwarebytes telemetry that they had running captured this on eBay.com. And again, it takes them to, because it's the same campaign, takes them to the Angler exploit, which is dropping ransomware like CryptoWall and ad fraud malware. And then sort of the tail of this is that this was also just found for the fourth time on the Huffington Post. The Huffington Post has had malvertising found on its site four times since December of 2014. First was December 2014; then February 3rd of 2015, two months later; then just last month on the 16th of July. And now just four days ago, on August 13th, a different group, Cyphort Labs, identified similar malvertising infection on visitors there.

And I won't go through the details. In the show notes I've got it, if anyone's curious. But it starts at HuffingtonPost.com and goes through one, two, three, four, five, six redirects. Which, interestingly, is, like, within the AOL system. So I'll just say that it goes to the AOL CDN, their content delivery network; then atwola.com, then tacoda.net; then back to advertising.com three more times, which is, again, the AOL system; then to Adtech; azurewebsites.net - oh, and this also goes to mbiscotti.com. So this looks like the same group that the Malwarebytes guys were tracking.

And so they wrote, as has been the case in the past, Advertising.com, the AOL platform, was the culprit. And they editorialize a little bit, just saying, "The cybercriminals are always looking for mass distribution of their payloads, and they get their wish list with malvertising. It's much easier to infect a popular site via its ads provider and reach millions of people than to try to put malware on individual victim computers." And they say, "We expect high-profile malvertising cases to continue."

And so this is a little bit sort of like the perfect storm. We have the collision of CryptoLocker, which is malware we've been tracking on the podcast and talking about for some time because unfortunately it is really effective and produces real revenue, if people can get it on their machines. And a problem with this growing advertising ecosystem, where many people are involved, and from what Rene Ritchie said in his posting that we'll talk about at the end of the podcast, a lot of lower quality players are getting involved. And lower quality means that there are more opportunities for exploiting this growing ecosystem.

Okay. And I know you talked about this on TWiT, Leo, this next note, which was that - or one of the shows. And that was that Kaspersky has been accused by two ex-employees of themselves creating fake malware for over a decade. This is rather credible, to my mind. First of all, Reuters covered this, so this isn't like TheRegister.co.uk saying something that nobody else picked up. And of course this was picked up by other people who follow what Reuters News posts. And the problem with this is that the whole thing seems credible. Of course, Kaspersky vehemently denied it. Both corporate and Eugene himself tweeted that the report itself was a false positive, which I thought was sort of a little humor there.

But so Reuters reported that, according to two former employees, starting more than a decade ago, Moscow-based Kaspersky Lab was trying to, was actively trying to damage its rivals in the marketplace by tricking their antivirus, the rivals' antivirus software programs into classifying benign files as malicious. They said that the secret campaign targeted Microsoft, AVG, Avast, and some other rivals, fooling them into deleting, or fooling their AV products into deleting or disabling important files on their customers' PCs.

And these two ex-employees - and I got the sense from the reporting that this was Reuters trying to be as responsible as possible, speaking to each of these ex-employees separately, as if, you know, so it wasn't like they together reported this, but that Reuters pursued this themselves. Because in the story, Reuters says that Microsoft, AVG, and Avast had previously reported to Reuters that unknown parties had tried to induce false positives in recent years. And then when Reuters apparently talked to these two ex-employees who independently confirmed that this is something that Kaspersky - that they had firsthand knowledge of Kaspersky doing.

Anyway, Reuters then followed up with Microsoft, AVG, and Avast, and those companies had no comment. So anyway, the back story here is that Kaspersky believed, Kaspersky organization believed that other companies were ripping off their work; that, you know, rather that independently investing in doing the research that we know Kaspersky does, I mean, we talk about them all the time. They're finding important things on the Internet. There's no doubt they're doing really impressive reverse-engineering work. The problem is that's difficult to do. And they became concerned that other AV companies were simply reaping the fruits of their labor and benefiting from it.

So what is publicly known is that, in an effort to prove that other companies were ripping off its work, Kaspersky has said that it ran an experiment. It created 10 harmless files and told VirusTotal that it regarded them as malicious. So this is very different than taking important files, like core Windows OS files, and basically creating malicious versions that other companies will then pick up on. And, like, for example, Kaspersky would create malicious versions of real files, would not put those patterns into their own product, but would let it be publicly known that there were malicious versions of these files around so that these other companies would start deleting them from their customers' machines.

So this is different than that. This was a probe to verify that this was going on. So they created 10 harmless files, but told VirusTotal that they, Kaspersky, regarded them as malicious. And of course, as we know, VirusTotal aggregates information on suspicious files and shares them, shares that with security companies. Within a week and a half, all 10 files were declared dangerous by as many as 14 security companies that had blindly followed Kaspersky's lead.

Now, okay. I guess I don't understand enough about the way this inter-AV company ecosystem works because, if you are volunteering files to VirusTotal, which is an aggregator and shares this with everyone, then it seems to me you would expect these to propagate back out. Maybe the idea is that this is supposed to be a starting point, that is, the information from VirusTotal is meant to be a starting point, and then the companies should conduct their own research, verify what's going on, and do some of their own due diligence, rather than just blindly adding them to their AV signatures and saying, oh, don't use these. Which is clearly what was happening in this case.

So anyway, Kaspersky says, "We never did it." It's - you know how sometimes things smell true; unfortunately, this sort of has the smell of truth. Of course they're going to deny it. They have to. But, boy, you can see that this is the kind of thing that a company would do if they felt that their competitors were really benefiting from the amazing amount and the effort to reverse-engineer these viruses. I just shake my head. And I'm so delighted that we're able to reap the benefit of Kaspersky giving us the level of detail that they do so we can peer inside these and see what's going on.

And, boy, it is also - it seems counterproductive that that work would have to be independently repeated by all the AV companies in the industry. On the other hand, if it's not, then Kaspersky's work is not being - they're not getting credit for it. So it seems to me that's just sort of a tough problem. I don't have an answer for it.

**Leo:** Yeah, I don't know what really happened, either. I don't think anybody does.

**Steve:** Exactly. Exactly. Yeah. It was one of these things where it was like, okay, you know, these guys said this, and Kaspersky says no.

**Leo:** Right.

**Steve:** And it's like, well, okay. Maybe. Who knows?

**Leo:** It's like that rumor everybody passed around for years that the antivirus companies were creating viruses. Which, you know, would be good for business. But I never saw any credible evidence that they were.

**Steve:** Right. And again, it's exactly as you say, Leo, the kind of thing where it's like, oh, I mean, the moment, the first time you hear that it's like, yeah, why wouldn't they? It's like, well, okay.

**Leo:** Because it would be wrong, maybe. I don't know.

**Steve:** Yeah. And mostly I think they don't need to.

**Leo:** They don't need to.

**Steve:** Yeah. I think they're, like, their hands are full. So, okay. So Lenovo has been found once again doing something really worrisome. Of course, famously, they were the people who were found distributing this Superfish that we talked about just now, as a matter of fact, when we were talking about Fiddler. Superfish, of course, was supposed to be doing good for you. Lenovo has got your back, and we're installing this to enhance your experience and provide a benefit.

What we now learn is that the Superfish technology was not only installing an SSL certificate in the root of all of the machines that Lenovo was shipping that had this what you'd really have to call malware, but it was the same certificate, and it had a well-known, it had a ridiculous password that was easy to find, which meant that all of the laptops were inherently trusting any other certificate signed by this one certificate that the Superfish software was planting in people's Windows root stores. Really bad.

So that's behind us; right? Well, okay. Now it turns out that Lenovo has been doing something else, and is doing something else, until maybe just recently. And there is no clarity on exactly what has just happened, if anything has, except that this affects a huge number of Lenovo laptops. And I'm sad because I've got two right behind me; you know? My instance of Windows 10 is on one, that I'm obviously not using actively. But Lenovo, the ThinkPad was what I was buying before. And of course, you know, that became Lenovo. And, yikes.

Now what they're doing has really got people upset. It turns out that the Lenovo BIOS is reaching up into the Windows file system prior to turning control over to Windows and replacing one of the core Windows files. In the Windows\system32 directory is a file called autochk.exe. And when you do a boot of Windows, of course we all know that the BIOS runs and sets up all of its hardware and does all of its work. One of the things this large range of Lenovo laptop BIOSes does is reach up into the file system and replace this autochk.exe file. It renames the one that's there. If you have an NTFS file system, NTFS has a little-used feature known as streams, where you're able to - it actually uses the colon character after the filename. So you could have, just to use something everyone's dealing with, autoexec.bat, for example.

An autoexec.bat:1 would actually be another stream of bytes, much as autoexec.bat is, which is in the directory and not visible to any typical file viewers. But it still is there. They call it a filestream. It's a feature that NTFS supports. And so the Lenovo BIOS will rename the existing autochk.exe to a :bak to give it to, like, have it - to, like, to essentially disappear it while not moving it. And then, out of its own BIOS image, write its own custom autochk.exe into the file system. Then it boots Windows.

And one of the things Windows does when it starts up is it always runs autochk. Well, autochk is now Lenovo's autochk. And when it's run, it creates - it from itself spawns, from its own code, creates LenovoUpdate.exe and Lenovochk.exe, which are set up as services and run themselves at boot time. So what all this means is you can change the hard drive. You can wipe the hard drive, you can format it and reinstall Windows, and it doesn't matter. The first time...

**Leo:** Sounds like a rootkit.

**Steve:** It is a rootkit, yes. The first time you run it, this stuff magically appears. Even though you never installed any Lenovo software at all, it is essentially percolating up from the BIOS and installing it itself.

**Leo:** Wow.

**Steve:** Now, there's been some confusion in the coverage of this because it's technical. But there was, in last year's Black Hat, so June of 2014, last year, one of the presentations was on something called CompuTrace. CompuTrace is a lost laptop recovery or monitoring tool which many laptop providers, maybe for a while, I know, I remember maybe five years ago it was very popular. I think I actually have it disabled in the BIOS of one of my Lenovos because Lenovo was one of the laptops that was offering it. I think even back in the ThinkPad days it was offered. It was being offered as a service. It was for theft prevention or tracking and recovery. Sometimes it was called LoJack for your laptop.

Anyway, so the company is called Absolute. And these guys in the Black Hat conference found that this CompuTrace was very much almost identical, in fact, in fact, I would imagine that whoever at Lenovo implemented their version, copied what CompuTrace was doing. Because CompuTrace also installs using this autoexec, I'm sorry, this autochk.exe file swap, they rename it differently and move it differently.

**Leo:** Yeah, but that's because they don't want a crook to remove it, to format the drive.

**Steve:** Correct, correct.

**Leo:** And then steal your laptop; right?

**Steve:** Oh, yeah. Oh, yeah, you mean the...

**Leo:** Makes sense on CompuTrace because...

**Steve:** Yes.

**Leo:** ...otherwise the bad guy would just format the drive.

**Steve:** What is a little disconcerting, and this of course is where the Black Hat guys came in, is that what it ends up doing through a very complex system of jumbling things around - and I have the link to the PDF, if anyone's interested - it ends up creating a connection to a remote server, giving it full remote access trojan access to your laptop.

**Leo:** Right. But that's how LoJack for laptops worked, because it would take pictures of the bad guy with your camera. It would send his location back. It has to do that; right?

**Steve:** Yeah. And it would have to have the IP address and so forth.

**Leo:** I was never fond of that product.

**Steve:** Yeah.

**Leo:** But people wanted it. People liked the idea.

**Steve:** Yeah. Well, I mean, now a security-conscious audience would be uncomfortable with their computer...

**Leo:** Horrified. You hate Windows 10, you'd really hate LoJack.

**Steve:** An unremovable remote access trojan, a RAT, which no amount of reformatting and so forth could get rid of. And then here's the other kicker. Now, as of Windows 8 - speaking, Leo, of Windows 10. This was just updated, as a matter of fact, on July 2015 to support additional features of Windows 10. Microsoft has something that, again, the press confused. This is called the Windows Platform Binary Table, WPBT. Again, show notes here have a link to Microsoft's documentation, where they explain what this is. And they said in their little abstract a platform, meaning whatever computer you're running this on, "A platform can be provisioned with the Windows operating system by entities including an enterprise, a system reseller, or an end-user." Meaning, you know, these are the different types of people who could set up a computer.

"If the platform has drivers, system services, or executable files that are integral to the platform, the platform binaries must either be distributed as part of the Windows image, or they must be injected into the Windows image by each of the possible provisioning entities." And, Microsoft says, "A rich set of tools exist to aid Windows provisioning, ranging from driver injection and offline registry management to sysprep imaging tools. However, there is a small set of software where the tools are not enough. The software is absolutely critical for the execution of Windows. But for one reason or another, the vendor is unable to distribute the software to every provisioning entity."

Okay, now, that's Microsoft speak for the hardware has some reason to inject software itself, independent of the so-called "provisioning entity." So Microsoft, this abstract ends saying, "This paper describes a mechanism for a platform, via the boot firmware, to publish a binary to Windows for execution. The mechanism leverages a boot firmware component to publish a binary," meaning an executable program, "in physical memory described to Windows using a fixed ACPI table." And it notes that the - and so what Windows does, so basically this is the same thing. This is a Microsoft-sanctioned official way for a BIOS to put a Windows executable in memory. And it could either map BIOS memory into physical memory, making that image visible, or it's able to copy it into physical memory.

And then, using this Windows platform binary table, essentially it's a way of saying to Windows, and Windows looks there and says, oh, the BIOS has something that I need to execute. And so Windows then copies it to, under the same directory, Windows\system32, it's called wpbbin.exe, and executes it. So what we have, essentially, is three different ways of BIOSes now arranging to run their own code in the operating system, all explicitly to make sure that nothing that the end-user does, reformatting the drive, wiping out the preinstalled Windows because they want a clean start, this stuff, the BIOS has ways of getting around all that. So just a happy heads-up.

**Leo:** Which Lenovos were - were these all Lenovos? Or just - because the Superfish thing was the consumer Lenovos; right?

**Steve:** Yeah. The Yodas are - is the name that jumps out at me.

**Leo:** The Yogas.

**Steve:** Click that, under "Lenovo's Dirty Tricks," click the Ars Technica link, and they show, down toward the end of their coverage, a long list. I mean, I didn't even put it in the show notes because it was like, wow, okay.

**Leo:** I somehow got in the comments section here. Oh, it is the comments section. You linked to the comments section.

**Steve:** Oh, I did? Oh, shoot, that's the wrong link. You're right. I had many links in my notes.

**Leo:** Don't worry about it. People can do this as their own. I guess right now, by this point, everybody should just not buy Lenovo anymore.

**Steve:** It's sad. Like I said, I love those...

**Leo:** Great stuff.

**Steve:** Oh, great hardware. Really. Although I guess, you know, has it been getting a little tinnier? A little, you know...

**Leo:** The ThinkPads were always great.

**Steve:** Right. They were like the enterprise class.

**Leo:** Because that's what they bought from IBM. And for a long time they kind of preserved it as kind of the pure thing that it was. I'm just - the chatroom seems - I don't know what they're saying.

**Steve:** I have the X61, and I think the X1.

**Leo:** Yeah, the X is...

**Steve:** The X61S, yeah.

**Leo:** So I don't know, but I just feel like...

**Steve:** Yeah, I agree.

**Leo:** I mean, after Superfish it was like, oh, well, that was only the consumer ones. And now this, it's like, clearly they're like Sony. They don't care.

**Steve:** Right.

**Leo:** Reminds people of Sony.

**Steve:** Right. They're not asking the user. They're simply saying we know best. We're going to stick our stuff into your - if a user reinstalls Windows and doesn't go and reinstall the Lenovo package - because, you know, Lenovo has, it's like, here's all the stuff that you need to make function keys work and to make all the additional hardware to hook it into Windows.

**Leo:** Yeah, it's very Windows-y, yeah.

**Steve:** Yeah. And according to...

**Leo:** Although Cory Doctorow always bought a ThinkPad and would wipe Windows and put Linux on it. And for all I know that's what he continues to do. And people in the chatroom do say that the ThinkPads were not affected. But I don't know.

**Steve:** Well, yeah.

**Leo:** If you'd do that to one, what's to stop you from doing it to others. It sounds like...

**Steve:** Hey, this is really working. This is working well over here. Let's just do it.

**Leo:** I can see the Superfish. Oh, we're selling these so cheap, we've got to make some additional money. Here, we're going to do this. But this one is ostensibly to protect the user, I guess.

**Steve:** Well, they did get caught. This caused an uproar. They have a patch of some sort.

**Leo:** Oh, yeah, yeah.

**Steve:** They have something you can run, but it's not being pushed out to all users. So it's not an auto-update thing. It's, you know, if you know enough to be upset about this, okay, fine, then we'll, you know...

**Leo:** Oh, if you insist.

**Steve:** I did want to mention that one of the presentations at Black Hat that I didn't pick up on until recently was that our friends at FireEye analyzed, they proactively analyzed the current state of Android fingerprint security and found it wanting.

**Leo:** Oy.

**Steve:** Yeah. In maybe the most glaring case was the HTC One Max, is it 10 or X? Anyway, it's maybe Roman numeral X.

**Leo:** No, One X, it's One X, yeah.

**Steve:** Oh, okay, the One X. They had a Max in there somewhere in their documentation.

**Leo:** It might be the - oh, no, yeah, there was an HTC One Max.

**Steve:** Okay.

**Leo:** I think there's a One X Max. There was a One Max.

**Steve:** Anyway...

**Leo:** That's a pretty - both of those are very old, more than a couple years old.

**Steve:** Okay. Good. Well, but it had a fingerprint reader, which is surprising.

**Leo:** Yeah.

**Steve:** And I thought that was a relatively new innovation in the Android smartphone world.

**Leo:** Well, didn't they mention the Galaxy S5, as well? I think they did.

**Steve:** They do, not in a good way. So, yeah. So anyway, it's the most glaring because, believe it or not, it saved the users' raw fingerprint bitmap...

**Leo:** Oh, dear.

**Steve:** ...in the data directory, /data, called dbgraw.bmp, with - and Linux and UNIX users will know what this means - with a file permission of 0666.

**Leo:** Oh, boy.

**Steve:** Which makes it world readable.

**Leo:** Yeah.

**Steve:** So any application was able to get pictures of the user's fingerprint. This was a mistake. They didn't intend to have the file permissions set that way. So it was easy and quick to correct. But as you said, it's been out there for some length of time. Maybe it only got the fingerprint reader later, or maybe that's a successive version that had that. But they did fix it quickly, which is pretty much all you can ask, except it'd be nice if they hadn't made the mistake in the first place. But going further, what FireEye found was that - okay. So in the ARM model there is this concept of a TrustZone. And there is hardware-enforced protection. In fact, that's called the TrustZone Protection Controller.

Unfortunately, what they found was that - what they said was, even if the protection of fingerprint data in the TrustZone, which is sort of like a secure region of memory, is trustworthy, it only means that the fingerprints previously registered on the device are secured, meaning that they're not easily accessible. FireEye found that the fingerprint sensor itself in many devices is still exposed to the attackers. Although the ARM architecture enables isolating critical peripherals from being accessed outside the TrustZone, by the programming of, as I was saying, this TrustZone Protection Controller, most vendors - not some or a few - most vendors fail to utilize this feature to protect the fingerprint scanners.

And as of the time of the writing, which was just this last Black Hat, "We have confirmed," writes FireEye, "this vulnerability on the HTC One Max, the Samsung Galaxy S5, and others. All vendors have provided patches after FireEye's notification." So FireEye notified them. The vendors provided patches that started then moving the fingerprint scanner into the TrustZone.

So essentially what we have is, I mean, the only way you can explain this is maybe the nature of the ecosystem, the way the software was made available, I mean, it's hard to understand how a manufacturer like Samsung could have fingerprint-reading hardware, understand this is about security, have hardware support for making access secure, and choose not to implement it. I don't know enough about the way Android software development chain works. But, you know, disappointing. But apparently now fixed. So, and FireEye notes that, as far as they know, they're the only people who ever looked. So what we need, clearly, is people to be looking at these things.

I mentioned already at the top of the show, and there's not a lot more to say about this, and that is that AT&T, The New York Times reported, was spying - AT&T was assisting the NSA to acquire data after 9/11, so since 2003, across the scale, I mean, across the Internet on a vast scale. The New York Times said that: "Newly disclosed [by Edward Snowden] NSA documents show that the relationship with AT&T has been considered unique and especially productive. One document described it as 'highly collaborative,' while another lauded the company's 'extreme willingness to help.'" And this went on for more than a decade.

These documents, of course, end at the time of Snowden's departure from the NSA. But so from 2003 up to 2013, that's been the characterization of their relationship, permitting wiretapping just across the entire Internet. The New York Times said that AT&T installed surveillance equipment in at least 17 of its Internet hubs on American soil, far more than its similarly sized competitor, Verizon; and that AT&T's engineers were the first to try out new surveillance technologies invented by the NSA for eavesdropping. In fact, one document reminds - and these of course are NSA documents that Snowden had. One document reminds NSA officials to be polite when visiting AT&T facilities to check on the eavesdropping equipment, noting that, "This is a partnership, not a contractual relationship."

So, you know, I didn't want to spend too much time on this because it's like, yeah, okay, you know, we already pretty much knew that this was what was going on and the way it was going on. And you'll remember when the news first hit, this was what I guessed even, absent any specific information at the time, had to be going on. And of course AT&T has always been, you know, not just the consumer-facing side, but they were the massive Internet backbone provider with much less competition once than they have today. You know, I mean, Verizon was far smaller once upon a time, whereas AT&T, they pretty much were the long-haul, long-distance communications provider, which meant they had the cabling and then the fiber optics to do this. So naturally, that's where you would want your nodes set up, if your goal was to monitor the whole Internet.

I imagine anyone using - do you pronounce it Wuala? Wuala?

**Leo:** Wuala.

**Steve:** Wuala. Wuala.

**Leo:** Like, well, there's always people say "voila" like "wuala." I think that's where it came from.

**Steve:** Yeah, there's no "V." Anyway, they're shutting down. They were a strongly encrypted, peer-to-peer, cloud storage provider.

**Leo:** With no financial plan whatsoever.

**Steve:** Exactly.

**Leo:** No monetization strategy at all.

**Steve:** And it wasn't - they got bought by LaCie, and it wasn't clear that, at the time of the purchase a few years ago, that it was considered even a merger, rather than an acquisition. But as you said, Leo, like, okay, how are you going to make this happen? How do you monetize this? And of course this is a problem that, like, for example, Twitter has. So it's not just constrained to smaller actors.

**Leo:** It's a useful lesson that, if you're going to trust your data to the cloud, it should be trusted to somebody that has a reasonable plan for monetization. Because this stuff's not free.

**Steve:** Right.

**Leo:** And if they ain't charging you, then, hmm.

**Steve:** Yeah. It's why my model is use your own encryption, your own client-side encryption, and then any of the major players. It doesn't matter if it's Google or if it's OneDrive or it's Amazon, you know, and you want to tuck it away in Glacier so that it's lower cost. Then you're in control. And as you said, Leo, then you're using a major player that is not going to go away. So yesterday they announced no further renewals or purchase of storage. And then - and hopefully everyone's getting the word.

So I just wanted to make sure that our listeners knew because, at the end of September, they will no longer accept writes. The system will transition to read-only. And that's only for two weeks. I'm sorry, no. For six weeks, through October and the first half of November, they will be read-only. And in the middle of November, November 15th, they terminate and then may well delete all stored data from the cloud.

Now what they are doing is suggesting that, if you really want to stay, like sort of with something like them, Tresorit, T-R-E-S-O-R-I-T dot com. That's a Swiss-based, a Swiss-Hungarian startup that we've talked about briefly. There's a link, I imagine, on the, well, it's on the Wuala shutdown notice page, which is at support.wuala.com, to the Tresorit.com/business/wuala-alternative. And they're making some sort of an ease of transition to help their users get over to Tresorit as the alternative, if they want. So I just wanted to make sure everyone knew about that.

Many people have, as I asked, shared via Twitter, and probably via the mailbag, although obviously we had no chance to get to it today, their thoughts for my quest for surfing safety, or surfing safely. And I just wanted to say that, in thinking about it further, as I have, the problem, the benefit of a VM, of a full virtual machine encapsulation, is that it's really a belt-and-suspenders approach. The browser is already doing everything it can not to allow anything bad to happen. You know, they all have sandboxing of various kinds and barriers and so forth. And then of course we can add add-ons to improve that. And then you've got that happening inside of a VM. So the browser thinks it's on an operating system where nothing else of any value is happening. So you really have good containment.

The problem is, what about links? I want to be able to click on a link and have the browser accept the link. But it won't in a VM. I would have to copy and paste every link into the browser URL. And, yeah, okay, I can do that. But oftentimes links, I mean, that just means right-clicking on it somewhere. Maybe the links I'm clicking on are already in the browser, so that's not such a problem. But then the second part, of course, is exporting. I would like, when things that I want to download are safe, to be able to relatively easily get them out.

So anyway, I'm sort of at a loss. I've been operating with NoScript disabled. And, oh, it's so nice that everything just runs. I mean, it just - it really is a pleasure to use the Internet like everybody else does. But Sandboxie is kind of okay, but it's got its own kind of quirks. I think I just need to spend some more time configuring it. Or maybe go back to NoScript. I don't know what I'm going to do.

So I guess I just wanted to say thank you, everybody, for your thoughts. And I just - I don't think there's a good solution for this. As our discussion just now about the growing problem of malvertising made clear, bad guys are exploiting ads in order to take advantages of known and unknown, which is to say zero-day problems in browsers, in order to get mass numbers of people infected. And so, boy, you really want to have some kind of protection from that. I know the listeners to this podcast absolutely will insist, they'll need to have some kind of protection from malicious ads, so wrapping in a VM or blocking script and only selectively lowering your guard.

On the other hand, you know, if it's on the Huffington Post that you would tend to trust, then if you have to get scripting enabled in order for the Huffington Post site to work, which has been our experience increasingly, as sites tend to be more and more dependent upon scripting, then, wow, what's the solution? There isn't a good solution. We just really need strong...

**Leo:** MS-DOS.

**Steve:** Yes. We really - yeah. It's just, you know, a clear tradeoff between convenience and security, unfortunately. Now, anything that you do to be secure is going to be a burden.

**Leo:** Why isn't JavaScript secure? It seems like that should be secure. Isn't it sandboxed somehow? I mean...

**Steve:** Oh, yeah. It's sandboxed. And we're not seeing vulnerabilities now, or really any longer in JavaScript. But it's the way, for example, that Flash can be invoked. So if you've got a Flash player in your browser, that's a problem. And all of these malicious chains are chains of JavaScript. So if JavaScript were disabled, none of this malvertising would work because the malvertising is using JavaScript in order to then run the exploit kit. So it is leveraging the fact that your browser is running scripting, unfortunately.

**Leo:** But the vector is usually Flash or something else. Just don't have Flash on your system.

**Steve:** I have to find out how the CryptoWall is getting in because we've seen some zero-days, I've talked about them with Padre, that were Flash-based. And unfortunately, most people do have Flash, also. So you're right, getting rid of Flash would work, except then we couldn't play your videos on TWiT.tv.

**Leo:** No, we have HTML5.

**Steve:** Oh.

**Leo:** Yeah.

**Steve:** Only recently?

**Leo:** No, we've had it for a while. Otherwise we wouldn't work on iOS.

**Steve:** But, eh, okay. It's funny because I saw a whole lot of traffic about not being able to...

**Leo:** People don't really understand what's going on in life.

**Steve:** Okay.

**Leo:** They seem to be quite clueless sometimes.

**Steve:** I did get a tweet that I got a kick out of from a Jason Fiore, who said: "I've used Chrome for seven years. I've never," he has in all caps, "seen this message until this today." And then he said: "Google doesn't like the EFF." And what it was, was a pop-up, complaining about Privacy Badger. And it said: "Warning: This extension is slowing down Google Chrome. You should disable it to restore Google Chrome's performance." Anyway, I just got a kick out of that because of course Google runs on advertising and tracking. And what Privacy Badger is doing, from the EFF, is working to thwart tracking. And it's like, okay.

**Leo:** That warning, though, to be fair, comes on Chrome whenever any extension doesn't act responsively.

**Steve:** Responsively?

**Leo:** Yeah.

**Steve:** Okay.

**Leo:** So, look, extensions, as you well know, are a big problem on browsers. So one of the things Chrome does is, if an extension is hanging the browser, it'll tell you.

**Steve:** Okay. Yeah. I've got...

**Leo:** I mean, I suppose Chrome could have put special code in there to do that for Privacy Badger, but I doubt that very much.

**Steve:** Yeah, I just thought it was interesting.

**Leo:** I think Firefox does the same thing, by the way.

**Steve:** I got a nice note from Chris in Atlanta, who just wanted to comment on SpinRite and drive quality over the years. He says: "Hey, Steve and Leo. Love the show, blah, blah. A quick SpinRite story for you. I was recently moving and getting ready to clean out the backlog of old drives I had in drawers and boxes. I started running SpinRite and DBAN" - which of course is Darik's Boot And Nuke - "on the drives, repairing and wiping each. Going through the box, I found one of the first drives I ever owned, a 40MB" - that's megabyte - "IDE drive off my first 486DX. This is a drive that has sat in the bottom of a dusty box for almost 20-plus years without a static bag, going through five-plus moves across the country, and years in houses without air conditioning. I wondered whether it would even spin. It was immaculate.

"You often talk about the degree to which drive manufacturers have stretched and crammed every last bit they could onto each platter, and stretched error correction to, or sometimes past, the limit. I have seen no better example of the tangible implications of this than seeing such a reminder of how solid old drives actually were. Just thought I'd share the story and the love for SpinRite." So, Chris, thanks. And I've commented that, with the amazing technology we have today, we could make, manufacturers could make a drive that was storing much less data, that was incredibly reliable. But obviously they couldn't sell it because everyone, in terms of dollars per bit, that's just got to be rock bottom and competitive.

**Leo:** You ready for 16TB drives?

**Steve:** I know.

**Leo:** Did you see that?

**Steve:** And it was an SSD.

**Leo:** It's an SSD.

**Steve:** Oh, my lord.

**Leo:** Although I guess you could argue that's probably better because you don't have to increase density and stuff. It's just more chips.

**Steve:** Well, that was using a new technology.

**Leo:** Oh, it was.

**Steve:** It was using a vertical...

**Leo:** That's right.

**Steve:** ...NVRAM that essentially didn't have the bits horizontal, it had the bits...

**Leo:** [Crosstalk], yeah.

**Steve:** Yeah, exactly. So it was physically denser. And they just, I mean, they did it as a demo. I think it was, what was it, someone was saying $5,000 or something.

**Leo:** It's not that bad.

**Steve:** So some crazy price. But still, you know...

**Leo:** In fact, my first 5MB hard drive was $5,000. That's not bad.

**Steve:** You're right. Was it five or 10? But, right, it was $5,000.

**Leo:** That's very expensive for 5MB.

**Steve:** Yeah, baby. No, but they were fast. Oh, compared to floppies? Whoo.

**Leo:** Oh, it was so great. Oh.

**Steve:** Yeah.

**Leo:** C base 2 ran so much better on it.

**Steve:** So, okay. So John Gruber, Daring Fireball, is a well-known Apple-oriented columnist who loves our friends at iMore. And last week he wrote about an experiment that Dean Murphy conducted and that the guys at iMore are well aware of. He said: "With no content blocked" - oh, and so what happened was, as I was mentioning at the top of the show, after the Worldwide Developers Conference where Apple announced that iOS 9 would make an adblocking API, or a content-blocking, sorry, content-blocking API available, this guy, Dean Murphy, just hacked up a quick Safari content blocker, which he made available to John Gruber, and tested it against the iMore website.

"With no content blocked, there are 38 third-party scripts," and he says "(scripts not hosted on the host domain)," for those who don't know what third-party means, "running when the homepage" - this is iMore's home page - "was opened, which takes a total of 11 seconds. Some of these scripts are hosted by companies I know," writes John, "Google, Amazon, Twitter, and lots from companies I don't know, most of which I assume are used to display adverts or track my activity, as the network activity was still active after a minute of leaving the page dormant. I decided to turn off all third-party scripts and see what would happen. After turning off all third-party scripts, the homepage took two seconds to load," so that's down from 11. "And also, the network activity stopped as soon as the page loaded, so it should be less strain on the battery."

John writes: "I love iMore. I think they're the best staff covering Apple today, and their content is great. But count me in with Nick Heer. Their website has problems. Rene Ritchie's response acknowledges the problem, but a web page like that Rene's 537-word all-text response should not weigh 14MB. It's not just the download size, long initial load time, and the ads that cover valuable screen real estate as fixed elements. The fact that these JavaScript trackers hit the network for a full minute after the page has completely loaded," John writes, "is downright criminal. Advertising should have minimal effect on page load times and device battery life. Advertising should be respectful of the user's time, attention, and battery life. The industry has gluttonously gone the other way.

"iMore is not the exception, they're the norm. Ten-plus megabyte page sizes, minute-long network access, third-party networks tracking you across unrelated websites those things are all par for the course today, even when serving pages to mobile devices. Even on a site like iMore, staffed by great people who truly have deep respect for their readers. With Safari Content Blockers, Apple is poised to allow users to fight back. Apple has zeroed in on what we need: not a way to block ads per se, but a way to block obnoxious JavaScript code. A reckoning is coming."

So that was July 8th. On July 9th, Rene replied with a really long - and I guess this is the 537-word response that John's referring to - with a great response. Which really I read front to back. I would urge our users to, our listeners to. We don't have time to read it or drag everyone through it. But Rene made a huge number of really good, I mean, exact - because we know him. I mean, we know iMore. You know, they're good guys. But he drove home the sort of the Catch-22 that they're in, which is that they've known that this has sort of been a growing problem for some time. They've already, before this experiment, you know, done everything they could. They've looked for some solution.

But the nature of today's ecosystem is such that they don't control the appearance of individual ads that their site is hosting. What happens is, to have an ad, you drop a script tag on the page, and a third party fills that when the user displays your page. And Rene did say something - I was really tempted to get him on the podcast, but I knew that we weren't just going to talk about this, so I didn't want to have him sitting around for two hours. But he said that the quality of the ads drops when the number of impressions that the good advertisers are willing to pay for is used up. Which is something I hadn't appreciated before. I guess there's, like, there's a wide range of ads. And I don't know, like, what the refresh period is. But apparently, you know, his site can vary over time, so that when the number of impressions that a quality advertiser is willing to pay for have been seen, then those ads are no longer selected, and you start scraping the bottom of the barrel. So we already know that not all people get the same ads because, if profiling works at all, or to the degree that it works, we're hoping that we're getting more relevant ads.

**Leo:** It might be more complicated than that.

**Steve:** Yes, I'm sure it is.

**Leo:** So I'm an advertiser. I come along. I say, okay, Rene, I'm going to buy - I'm going to pay you a certain amount of money, $200,000, for 1.8 million impressions. When that 1.8 million impressions is reached, that ad's gone.

**Steve:** But remember, it's not the advertiser. Rene has no relationship with the advertiser.

**Leo:** You're misunderstanding.

**Steve:** He has a relationship with the network.

**Leo:** You're misunderstanding it. So, right. Okay. So Google, I'm going to pay you for two million impressions. When those impressions hit, the ad goes away. So the ad network doesn't have an ad from that advertiser anymore.

**Steve:** No, that's not the way it works.

**Leo:** Okay.

**Steve:** No, because, no. No, let's talk about it because there's a third party interposed. So Google is the third party. Advertisers contract with Google. And then the websites contract with Google; right?

**Leo:** No. You don't understand it. So I'll tell you one more time. You don't understand it. So I'm Rene Ritchie. I'm iMore. It's not Rene, actually, it's iMore. It's Mobile Nations. I'm Mobile Nations.

**Steve:** Right.

**Leo:** I'm going to put an ad banner right here. That ad banner is sold, you know, basically I put a Google ad banner there. That ad banner is sold by Google to Ford. Ford says, "I will buy two million impressions." Google says great. Two million impressions come and go. Google still has that hole on the page. It then puts another advertiser in there.

**Steve:** Right.

**Leo:** So the problem Rene's going through is that, because their traffic has gone up - I think this is what he's talking about. Because their traffic's gone up exponentially in the last three months, their site's taking off, you're seeing a disproportionate number of kind of crappy ads.

**Steve:** Ah. Okay. And so, but the problem I had...

**Leo:** That will change.

**Steve:** And the problem you and I had was that...

**Leo:** That's just an interim thing.

**Steve:** The problem you and I had was just one of terminology because you were saying Google is the advertiser. I was saying Ford was the advertiser because...

**Leo:** But Google owns that space. So I'm not sure how Rene does that. But Google owns that space.

**Steve:** But they're an advertising network.

**Leo:** Yeah.

**Steve:** And so I wanted just to differentiate them. And that's the point.

**Leo:** But we even do the same thing. We don't sell to Google. We sell to individual advertisers. But we use Google as the engine behind it; right? And maybe we have set parameters with Google. Those ads are in rotation. If you go to our site and repeat, you'll see different ads; right? There are probably, I don't know, but there are parameters in there. After a certain number of impressions, that ad goes away. That's still us selling directly to advertisers using Google as an engine that displays the ads. So it works in both cases.

**Steve:** And it's just like the ads on the podcast. They go away, too.

**Leo:** Right.

**Steve:** Like toward the end of a quarter, when you've used up the number of impressions.

**Leo:** Right. The difference is we don't usually replace them with some - actually, I take that back. Whenever we can, we do. So you have two ads today. But that - and I don't know where we are in the quarter. We're kind of in the middle of it. But that, yeah, that's how it works.

**Steve:** Yeah. Although...

**Leo:** I mean, that doesn't really change much of anything, to be honest with you. I'm not sure why Rene even brought that up.

**Steve:** Well, I guess - so the point is that there's this sense from what Rene wrote that they recognize there is no alternative for them. You know, I read - there was a ton of comments also, a lot of people saying, oh, you know, do a Patreon approach. Do micropayments. Pay for access. And Rene very patiently responded to those. And he says, you know, we would love to do anything else that would work. But nothing else works. Unfortunately, we've got a big staff. We have a high-traffic site. The ad model is what works.

And what I was reminded of was that we did go through this, and I'd completely forgotten about it, like 10 years ago, the era of pop-ups, where that's the way web ads used to work is, at the dawn of JavaScript, it was scripting that allowed this, is that script that ran on sites was able to create another window. And of course it got abused. You start having windows popping up all over the place, and it drove people crazy. And there was so much pushback that first there was pop-up blocking software, and then the browsers themselves said, okay, you know. They internalized how bad this had become and incorporated pop-up blocking into their own operation, so they would no longer allow script to launch another window.

And what I think we're seeing with this - and this is what we've been talking about, and I think we're done. We've beaten this thing to death, and I'm going to let it lie now, I promise. But we're seeing this same level of unrestrained abuse by the advertisers, where their interest is not the user's interest. The louder they are, the more obnoxious their ads, the more they see results. So they're incentivized to go crazy, I mean, to be as distracting as they can. And due to the way the system works, we don't choose the individual ads. You know, as you said, Google owns the space. We contract with the advertising network for space on the page, and who knows what is going to fill that space.

**Leo:** Yeah. I don't know what Rene or what Mobile Nations does. But often, most of these ads, you don't choose the advertiser.

**Steve:** Right.

**Leo:** You choose an ad network.

**Steve:** And Rene did say they don't. They have no control.

**Leo:** That's, by the way, that's the lower rent way to do it. It's the easier way to do it. But bigger networks like The Verge have an ad sales team.

**Steve:** Right. And Rene did say we're just, you know, we're not to the point where we have that much...

**Leo:** The Verge, if you see an ad on The Verge, it's because The Verge sold that ad to that advertiser. Same thing with us. If you see an ad on our stuff, it's because we sold that ad to that advertiser. But if you just put - but, you know, lots of blogs use Google ads. I used to do that, use AdSense. You put a little AdSense thing on your site. Well, you don't control what ads are there. I mean, you have some control. There's some settings. You could say don't put guns and porn there or whatever. But they pick the advertiser.

**Steve:** Yeah. And this is one thing I wanted to do was to highlight the difference in your advertising from the type that we're increasingly being exposed to on the Internet. Yours is under control.

**Leo:** Yeah, I was at an Android site the other day, I think it might be a Mobile Nations site. I couldn't read it on mobile because there was so much stuff on it. And I had to give up on the content. And so at some point these models break down.

**Steve:** Yeah. And I don't know how we get out of here. One thing I was thinking was, you know, because, like, you know, the problem is the adblockers are a bit of a blunt weapon. We want to support the sites we visit.

**Leo:** No, we don't. You're impugning much higher motivations, because you have them, and I have them, than 99 - most people are not thinking about the relationship of the ad to revenue. They just want the content.

**Steve:** Yeah.

**Leo:** And they don't want to see an ad, and that's that. And you know what, if it puts them out of business, I don't think they even care. That's the problem, really.

**Steve:** Yeah.

**Leo:** And that's why I always bring this up, because I think people need to understand, as you do, too, that there's a reason for ads. There's a consideration going on.

**Steve:** That's why I loved that example of Ars Technica saying, look, you know, because we pull a much more technically sophisticated audience than the Chronicle, more of the visitors are taking advantage of blocking technology and hurting our revenue. If you want us to stay, you need to turn that off. And so maybe we will start seeing selective blocking, where, for example, I wouldn't have a problem with opening myself to no blocking, and only as I have a horrible experience I turn a blocker on. Now, you had said don't go to that site. The problem is you don't know it's bad until you do go.

**Leo:** Well, then just leave and don't return.

**Steve:** Yeah, but how do you - how is that practical? How do you not return? I'm not going to remember all the sites with bad ads.

**Leo:** I think you have a moral obligation not to return because you don't want to pay that site for the content. It's you're stealing. It's like going into a store and taking a candy bar. How am I supposed to know? Well, because you have a moral obligation to pay in the way the merchant asks you to pay. Don't you?

**Steve:** Yeah. The first blog here in my notes, by Marco Arment, his was titled "Adblocking Ethics." And he argues a strong case for it not being on us, that it is on the site not to present the user with something obnoxious. I mean, because...

**Leo:** Yeah, because then users won't ever go to that site again. No.

**Steve:** And this is why I'm arguing for a tool to make that feasible, some way, like, say, okay, this site is intolerable. Somehow block it.

**Leo:** Yeah. I mean, I don't know. I don't know what the answer is. I confess, I skip through ads on my DVR. Same exact thing.

**Steve:** Well, and there is a notion of quality. People watch the Super Bowl, I watch the Super Bowl, of all things, because the ads are amazing. And every so often on TV, there's one, there's an ad now for Fiber One. Have you see the two lions on the African savannah? Oh, my god, it's just, it's - or the old ladies driving around in a VW, the VW commercials. Anyway, there are commercials...

**Leo:** I'm skipping those ads.

**Steve:** There are commercials that are just wonderful.

**Leo:** Yeah.

**Steve:** But most of them aren't. And you're right, they're just...

**Leo:** I'll tell you, that's very, very expensive. Those are million-dollar commercials.

**Steve:** Yeah.

**Leo:** And most business - remember, an advertiser is a business.

**Steve:** Yeah.

**Leo:** That wants your business.

**Steve:** Yeah.

**Leo:** And maybe your buddy down the street who's got an ice cream shop, I mean, I think probably everybody listening to the show knows small businesspeople who are in business. They can't afford those great million-dollar ads. I'm sorry, they can't. They also need to advertise because people won't discover them otherwise. People advertise because it works. It's not that the advertising doesn't work. I agree people shouldn't be rude and mean in advertising; and absolutely I think that, because of malware, everybody has the right to block malware. I don't know what the answer is, I really don't. Again, I fear that we're going to create an environment - in fact we already have - where just content sites don't work. And they're going away. They're dying, rapidly.

**Steve:** The system's going to collapse, yeah.

**Leo:** It is collapsing. We're in the middle of a collapsing ecosystem of content system. And by the way, that makes it worse than exactly what you hate worse. Because what that does is create BuzzFeeds and The Verge and sites that create content that is about getting you to click, so you'll see more ads, instead of sites creating good, high-quality content.

**Steve:** For example...

**Leo:** So you're getting an untended consequence of all this, which is more of what you hate, not less.

**Steve:** Right. And, for example, I'm seeing, when I go to a site that has some long content, they now deliberately chop it up into 10 pages...

**Leo:** Slides.

**Steve:** And make you click through it.

**Leo:** Freaking hate that.

**Steve:** In order, exactly, it's like I get five inches, then I've got to go to the next page, which of course gives me another set of ad impressions.

**Leo:** In the best of all possible worlds, all content, you would have to pay for it. Period.

**Steve:** Yes.

**Leo:** It would all be HBO. And it's very undemocratic because not everybody can afford that.

**Steve:** Right.

**Leo:** And I hate to say it, but people are cheapskates. They don't pay for content.

**Steve:** Yeah. I mean, you and I subscribe to HBO. I've got a neighbor who there's so much stuff there I know he would love, but he doesn't have it, and he's not going to...

**Leo:** Most people are cheapskates, yeah.

**Steve:** Not going to do it.

**Leo:** HBO is, I think, doing fine. So obviously you can do that for some kinds of content. Wall Street Journal's doing fine.

**Steve:** Yeah.

**Leo:** They charge for content. But...

**Steve:** But there have been, there have been paywalls that have failed.

**Leo:** You know, Lisa keeps talking about we're going to do a paywall and charge for content. And it's like, I don't know. I guess - I think we'll just disappear. We'll dry up and go away. I wouldn't do that unless there were no other choice. We did it for the - you were there. For the first three years that's what we did. There were no ads.

**Steve:** Yup.

**Leo:** I don't think I paid you in those three years. I didn't pay anybody.

**Steve:** We made it.

**Leo:** Thank god. Thank you. Thank you for doing it for free.

**Steve:** For what it's worth, I've had a lot of feedback from our listeners who really enjoy this discussion, the point and counterpoint between us about this.

**Leo:** Well, it's germane to security because, alas, as you pointed out, now we're getting through, these ads are so - the system, the way the system works, malware can go through them. It's terrible.

**Steve:** Right. And I guess there are - is there a pecking order of ad supplier? That is, Google is some...

**Leo:** No.

**Steve:** Oh. But I got the sense that there were, like, low-quality networks.

**Leo:** Huffington Post is an AOL product. Yahoo! is serving these ads. These are not low rent.

**Steve:** Yeah.

**Leo:** The problem is they're automated systems.

**Steve:** Right.

**Leo:** And they use Flash, which they should never, ever, ever do.

**Steve:** Right. And they use script.

**Leo:** And they use scripting. Yeah, that'll, I think, truthfully you could make a strong case that an ad should simply be a picture. But you absolutely have to have tracking because people pay for the number of people, the impressions. And we already know from Google that half of all the ads Google sells are never seen.

**Steve:** Well, counting is different than tracking. So, you know, counting is counting impressions so they get paid. Tracking is, oh, look, this user is now on this site, and they were over on that site.

**Leo:** Yeah, well, the problem is, to really count, I have to know how many times you saw it. Not just how many people in general saw it. Because it doesn't - if I show it to you a thousand times, I don't want to pay for a thousand impressions on Steve Gibson. I only want to pay for the first seven.

**Steve:** Even if I go to a different site?

**Leo:** Yeah, right.

**Steve:** If I go to a different website? Really.

**Leo:** I only want to pay for the first seven impressions or whatever. Seven is considered the sweet spot. So a thousand impressions to the same IP address is not as valuable as a thousand unique IPs. Of course. Obviously not.

**Steve:** Yeah.

**Leo:** So it does kind of become tracking, ultimately, to be effective.

**Steve:** Yeah. To do correct counting.

**Leo:** Yeah.

**Steve:** In the same way that you correctly count podcast downloads.

**Leo:** That's how we do it. We have Podtrac, which is the agency that does that, has a list of IP addresses. And it counts each one uniquely. That has some downsides. Microsoft counts as one listener. No matter how many people from Microsoft listen, one person.

**Steve:** Yeah. And that PodCall group, they do redownload, but they unfortunately redownload from the same IP.

**Leo:** Right. They download once.

**Steve:** Yup.

**Leo:** Or even if they download a thousand times, it counts as one.

**Steve:** Yeah. They do redownload every time.

**Leo:** Every time they redownload?

**Steve:** And I said, yeah, sorry. Yeah, they...

**Leo:** I didn't know that. Why would they do that?

**Steve:** I don't know, but...

**Leo:** Just to save space.

**Steve:** Yeah. Yeah, so it'll be interesting to see what happens. I think you're right, Leo. It's a problem, and users, I mean, what's the - we saw the metric, 40% of web surfers are now using adblockers.

**Leo:** No, thank god, no. Oh, my god, no.

**Steve:** Oh, it's not.

**Leo:** Adblocking was up 41%.

**Steve:** Oh, okay. Whew.

**Leo:** It's still only, like, in the U.S., like 10 or 11 percent.

**Steve:** Okay.

**Leo:** Oh, god, we'd be dead if it were. I don't know what we would do. Actually, we wouldn't.

**Steve:** No, you wouldn't, no.

**Leo:** No, TWiT wouldn't, no, because we don't care.

**Steve:** No, no.

**Leo:** Our banner ads wouldn't - but we only charge for one of those two banner ads, by the way. The other one's - the other one we give away. I don't know what to do. Fortunately, it's not going to happen. I'm going to be long gone before anything changes.

**Steve:** Yeah, I don't think you're in danger because again, you know, you're delivering ads from advertisers who we select and know and approve of. That's a whole different model than...

**Leo:** That's why, by the way, I'm doing it this way. I mean, I was conscious of this 10 years ago. I didn't want to have advertising 10 years ago. We finally gave in. But even then it was very constrained because I knew our audience doesn't want this stuff.

**Steve:** Well, and Leo, look at the enterprise you've built. What would power it, if it weren't for advertising? Our listeners wouldn't have all this content.

**Leo:** Well, that's kind of my point, is that not all ads are bad. Adblockers don't discriminate.

**Steve:** No, and that's my point. See, adblockers don't discriminate, but neither do the advertisers. It's that when you're hosting an ad on your site the way Rene and iMore are, they're opening a hole, and they have no idea what will fill it. And that's the problem.

**Leo:** They're not big enough to do what John Gruber does. John Gruber is one person, so he doesn't have to make as much money. But he has ads on his site. But he sells them himself, I think.

**Steve:** Right.

**Leo:** Anyway, I don't know what to say. But this has been a fun show.

**Steve:** There it is.

**Leo:** There you have it. There you have it. Ladies and gentlemen, Steve Gibson. You'll find him at his website, GRC.com. That's where you get SpinRite, the world's best hard drive maintenance and recovery utility; free versions of lots of other things because he gives away everything, everything else for free, including this podcast, 16Kb versions of the audio, 64Kb versions of the audio, fully humanly written transcriptions by Elaine, lots of good stuff. GRC.com. Someday we'll answer questions. That's where you'll leave them, GRC.com/feedback.

**Steve:** Let's hope it's next week. Let's hope.

**Leo:** Maybe next week. You can also tweet him - he's @SGgrc - another good way to ask questions and have a dialogue with Mr. G during the week. We have the show as well, on our website, TWiT.tv/sn for Security Now!, audio plus video. Yes, we have video. I don't know why. That's another thing we're looking at is eliminating video of some shows, like this show, just make it audio. There's nothing to see.

**Steve:** Makes sense.

**Leo:** Nothing to see here. And you can do that at TWiT.tv/sn, or subscribe wherever you get your finer podcasts because they're all there. We had a live person in the studio briefly, but he collapsed in a puddle.

**Steve:** I'd be shaving less often if we didn't have video.

**Leo:** Yeah, you see? I wouldn't have to spend millions on wardrobe and haircuts. If you want to be in studio, we do have a limited space in my studio, but there is room for about five or six people. You can email tickets@twit.tv. We do Security Now! every Tuesday, 1:30 p.m. Pacific, 4:30 Eastern, 20:30 UTC. And please stop by and say hi. Love to see you. Thanks for being here, Steve. Thank you for all you do.

**Steve:** Thanks, buddy.

**Leo:** Here's to Year 11...

**Steve:** Starting in.

**Leo:** ...of Security Now!. Bye-bye.

**Steve:** Bye.