"ScriptNo" for Chrome

Description: This week, after catching up with a busy and interesting week of security news and events, Steve and Leo take a close look at ScriptNo, a new Chrome extension created by a developer who left Firefox (and NoScript) for Chrome and was pining for NoScript's features.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-339.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-339-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's found a plug-in for Chrome that duplicates the functionality of NoScript - which you know he loves - for Firefox. We'll talk about ScriptNo and a bug in a security camera that makes it possible for anybody to see what you're doing, any time, without a password. It's coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 339, recorded February 8th, 2012: ScriptNo for Chrome.

It's time for Security Now!, the show that protects your privacy, security, everything you need to know about keeping it safe online with this fellow here, the Explainer in Chief, Mr. Steven Gibson.

Steve Gibson: My new moniker.

Leo: Explainer in Chief.

Steve: Explainer in Chief.

Leo: And you told me your middle initial last week, and I forgot it. J?

Steve: That's just as well. M.

Leo: Oh. M. Steven M. Doesn't that sound a little bit more - Steven M. Gibson sounds a little more serious.

Steve: Back when I was at the AI lab at Stanford - when I was in high school I was working at Stanford's artificial intelligence lab, and you used your initials as part of your logon.

Leo: SMG.

Steve: So my initials were SMG. And so they called me "Smog."

Leo: Oh, I love that.

Steve: That's wonderful.

Leo: Smog Gibson is here.

Steve: Yeah, I was a coolie, as they termed it.

Leo: How fun. That was when they thought artificial intelligence was going to change the world.

Steve: Oh, it's going to be easy, yeah.

Leo: No problem.

Steve: We've just got to write some programs, and these things are going to start thinking any moment now.

Leo: Simple, simple, simple.

Steve: Didn't turn out. I mean, it was we who got the education, not the computers.

Leo: True.

Steve: It was people who thought this was not hard, we learned, whoa, there's more going on here than we thought. But that's always the case. I was telling someone just the other day, who had a great idea for an Internet startup and wanted my opinion about it, I told him a little bit about - he's not a Security Now! watcher, so he found me

because he's actually the stepson of an ex-girlfriend, who'd graduated and had an idea for an Internet startup. And I told him a little bit, I sort of paraphrased the Portable Dog Killer episode and what we learned, what I learned from that; and the takeaway message that I shared with our audience was nothing is going to happen if you're just sitting around playing videogames.

It's when you try to do something, even if you think you know how, even if you think, hey, I'm just going to go do this - and this is exactly what happened in the AI world was we all thought we knew how to program a computer to play world-class chess or navigate a little robot cart around the parking lot. That was my first introduction to the lab was as I was turning up into the driveway, there was a sign, a warning sign that said "Caution: Robot Vehicle in Use." And this was in 1972.

Leo: That's impressive.

Steve: I thought I had died and gone to heaven

Leo: Yeah.

Steve: It's like, you're kidding me. So that was the beginning of several years of a lot of fun. And...

Leo: And as you say, learning, because we found out you can't do it, or it's not as easy as you thought it was.

Steve: Yeah. It's like, oh, digitize the video, and then we have a picture, and now let's do some edge enhancement, and then model the area, and have the computer know where we are. And it's like, oh, boy. When you start, oh, it's easy to say. But when you turn it into details, that's when you start thinking, oh, wait a minute, there's something I didn't think of. Oh, I didn't think of that, and I didn't think of that, and I didn't think of that. And that just - that goes recursive.

Leo: You can never think of everything.

Steve: Although I did meet with an avid listener to our podcast, actually, a friend of ours. Maybe you know Evan Katz? Do you know of Evan?

Leo: I sure do, yeah.

Steve: Anyway, Evan is a neat guy. And it turns out he follows, among many things, computer-based chess. And so I got a complete tune-up. We had lunch for a couple hours a few months ago, actually over the holidays. Or I guess it was over Thanksgiving. And he knew all about the state of the art in computer chess, which I hadn't been following for years. And it was fascinating that...

Leo: We've come a long way, baby, let me tell you.

Steve: Oh, my goodness. And it's all just in software now. It's all programmed. No one's building big rooms full of custom chess-playing hardware. It's our standard Intel core whatever processors and the algorithms...

Leo: They're so fast.

Steve: Yeah, I mean, and now the humans are out of it. It's no longer about - it's pitting the software against each other.

Leo: You can buy a pretty damn good program that's a grandmaster quality program that runs on your PC just fine.

Steve: And I don't know how I feel about that, Leo. I mean...

Leo: Well, it's brute force, that's the thing. It's not...

Steve: Yeah, exactly. I grew up playing chess with a grandfather of mine and really enjoyed the game. And chess, coming back to the game is sort of something I'm thinking of I'll do after I unplug myself...

Leo: Hey, you and I should play a game. I didn't know you were a chess man.

Steve: Well, but believe me, it's been a long time.

Leo: Well, it has for me, too. I was pretty serious in high school. I played in tournaments. I played in the U.S. Open the day before my wedding. I was very serious about this thing.

Steve: Yeah, well, it's a fantastic game.

Leo: It's a wonderful game. And it did take some of the steam out of it when we found out that a computer could beat us.

Steve: Yeah. And so now I'm kind of thinking, well, I don't have anyone who's in my neighborhood who's fun to play or does play. But I've got technology coming out my ears. But how does that feel, to just be, like, playing a game you know you can't win. And it wouldn't be that so much. My ego's fine.

Leo: You can play humans you can't beat either, yeah, yeah.

Steve: I don't care if I know I'm going to get beat. And in fact my grandfather was way better than I was, and I spent the first many years really just having the stuffing beaten out of me. But then...

Leo: That's how you learn.

Steve: ...the tide began to turn a little bit. But so the point is, I wouldn't mind losing if I knew that I was losing to another human brain, rather than algorithms that are looking at this thing and going, mate in 26. It's like, oh, okay, what's the fun there?

Leo: Well, yeah. I mean, what we learned, and we didn't know at the time, was that chess turns out to be solvable, a solvable problem, given enough RAM. It's mostly, I think it's mostly about calculation speed and sufficient RAM to store the positions. But it's solvable. Now, interestingly - and by the way, you and I, we're going to start playing chess. You have a smartphone; right? We could play chess with friends. You and I could play head to head over the Internet. I'll play you a game. But now, if you're interested, turns out that the most intransigent game is Go, the Japanese game of Go.

Steve: Which is so, like, visually simple by comparison.

Leo: It's all pattern matching.

Steve: Those little stones.

Leo: And it turns out that it's computationally so intensive. Of course it is also pure calculation, and sufficient brute force will solve it. But they cannot make a Go machine that even plays nearly as well as moderate level professionals.

Steve: In other words, it won't Go.

Leo: It don't Go. I think it's fascinating. It's combinatorially too complex, so they can't...

Steve: Wow.

Leo: It is a great subject. I gave a lecture on one of the MacMania cruises about chess-playing machines. It's a fascinating story.

Steve: And how in the old days there used to be cheaters. There were, like, little midgets hiding underneath the table.

Leo: Yeah, they called it the Mechanical Turk. And there was a guy in it. It was a very good player. He beat some of the crowned heads of Europe. And they thought it was a robot.

Steve: I guess there was enough clockwork obfuscation on the outside that it convinced experts, who stared into it and scratched their heads and thought, no kidding, well, I guess you've just got to get the gears in the right place in order to...

Leo: It was a good magic trick, and that's all.

Steve: Yeah.

Leo: So today we're going to talk about ScriptNo.

Steve: Well, yes. Our main topic is - I did a deep dive into a browser extension, and our listeners know that I don't do that often. I did it with LastPass, and the results were stellar, and I'm really glad I did. That was a great investment of my time and learning. I did it with NoScript for Firefox because we know how I feel about scripting in browsers. Well, one of the things that has sort of put me off of Chrome a little bit is that there wasn't anything that gave Chrome users the same kind of control with Chrome. You can, I believe, I'm sure I knew once that you could turn scripting off completely in Chrome. But if you do that, then nothing works. And there was for a while an extension called NotScript, which didn't do anything really. I mean, it just didn't work very well, had very little functionality. I think it was something someone did and then just sort of abandoned.

Well, ScriptNo is recently created, is being actively developed by a neat recent college graduate in Toronto who knows us and has followed GRC and my work for years, it turns out. I didn't learn that until he responded with thanks to a donation that I sent to him to support his work. And the more I played with it and poked at it, the more I've been impressed and liked it. So today is for everyone who feels as I have - oh, and in fact, this is why Andrew created it. He was a Firefox 3 user. Then he upgraded, as we all did, to Firefox 4 and watched the memory just bleed out of Firefox without end. And he was also on Opera.

And then he's very pro-Google, in fact, sent out an application for employment to Google that has never been - Google Toronto or Google Canada or something - that they've never responded to. So if there's anyone within reach of this podcast at Google Canada, there's a very talented-seeming person who would be interested in getting a word back from you.

Leo: What a good way to get a job.

Steve: Yeah.

Leo: Just write an amazing extension for Chrome.

Steve: Well, anyway, I'm going to talk about that. It's got more features in some useful ways than NoScript. It tells you more about what's going on, which I know our listeners will like, just by hovering your mouse over things, the kind of information I want to have and the kind of controls that it's nice to have. So essentially, we've got really good - and it's not just scripting, it's objects. So it's Java objects, embedded stuff, Flash, images, the whole panoply. So that's our main topic. But we've also got some news. And that might not seem like a lot to talk about, but there's one really intriguing disaster of the week.

Leo: Our disaster - I'll get an echo effect ready for you.

Steve: That would be good. Oh, goodness. With a horrible vulnerability in a widely deployed webcam that, without its owners knowing it, is horribly violating the privacy of everyone who has it, without their knowledge. Because it turns out you don't need to log on. And there's a search engine that finds them all for you.

Leo: So let's talk. What have you got?

Steve: Okay. So we're keeping an eye, a background eye, on the so-called NSTIC. I've talked about this a few times. This is the so-called - it's the acronym for the National Strategy for Trusted Identities in Cyberspace, which is the current administration's determination, which is well meaning, to solve the problem somehow of trust in cyberspace, that is, how do we authenticate people? If we're going to be an Internet-based world, truly, and more business is going to go on the 'Net, and more commerce is going to go on the 'Net, and the world is going to go on the 'Net, we need to come up with a robust authentication system for people. And this is a topic that we have talked about often. I'm passionate about it. It's the reason that Stina and Yubico, with her YubiKey, were so interesting. And we've talked about the VeriSign football and eInk cards and one-time passwords and all this stuff.

In the physical world we know each other. We see each other. We recognize each other. And we can believe that we're talking to something that we know, a website that we know, or that a website knows it's talking to one of its users. But as we also know, there's all kinds of ways, I mean, for that to fail. It's incredibly challenging to do that. So the NIST is the organizing body behind this, the National Institute of Standards and Technology that's been around forever. And I'm on the inside of this stuff because I'm interested in what's going to happen and how it's going to go. And so I received a notification of some progress. And I thought, oh, well, what? This could be great.

Well, then I realized just how slowly turning the wheels of the bureaucracy are. Jeremy Grant is the senior executive advisor of identity management at NIST, so this falls, as you'd expect, under identity management. There is such a division. And he said in his note, "I am pleased to announce the publication of a new NIST report entitled 'Recommendations for Establishing an Identity Ecosystem Governance Structure.'"

Leo: Just the name scares.

Steve: Oh, my goodness. So this is a 51-page PDF with recommendations for how to establish an identity ecosystem governance structure. So, yeah. The upshot is we need to keep working on this ourselves, Leo. This is not going to get solved anytime soon.

Leo: Well, on the other hand, some centralized system, and the NIST would be the right ones to do it, I think, does seem necessary.

Steve: Well, yes.

Leo: You don't want Google or Microsoft doing this, although they have been trying.

Steve: Well, and I don't mind their input. But agreeing with you that taking this slowly, this is an important thing. And we know that there are, I mean, this is a hard problem to solve because the tighter you make the authentication of someone's identity, the harder a problem you have with the flipside, which is anonymity and privacy, both of which people have a right to and covet and cherish, as they should.

Leo: Google went right up against this with their name policy on Google+. And it bit them in the butt.

Steve: Yup. And look what we just went through with SOPA and PIPA, where legislation just appeared in Congress and asked for votes with nothing like this kind of careful, we're going to take this thing through, we're going to make sure what comes out the other end is useful and not something that just immediately enrages the world the way SOPA and PIPA did. So I will keep an eye on it, and I will be informing our listeners, as this thing moves forward, where we stand with it.

Leo: Thank you.

Steve: There's some really encouraging news that Brian Krebs reported on his blog. Since he covered it so thoroughly, I didn't go any further. And so I'll just quote from what he wrote. He wrote that "Adobe has released a public beta version of its Flash Player software for Firefox that forces the program to run in a heightened security mode or sandbox..."

Leo: Yay.

Steve: Yay is right - "designed to block attacks that target vulnerabilities in the software." And Brian goes on to say, "Sandboxing is an established security mechanism that runs the targeted application in a [constrained] confined environment that blocks specific actions by the app, such as installing or deleting files, or modifying system information. The same technology has been built into the latest versions of Adobe Reader X, and it has been enabled for some time in Google Chrome, which contains its own integrated version of Flash. But this is the first time sandboxing has been offered in a public version of Flash for Firefox."

And I will note that it has succeeded in Adobe Reader X. That is, the most recent problems, those exploits which were critical and immediately exploited, Adobe had to scramble around in order to protect the users who had not updated to Reader X or Acrobat. But they were able to relax over on the Reader X side and only update the problem with their standard quarterly patch cycle because the sandbox that they developed held, and it worked. So the idea that we're going to get Flash sandboxed for Firefox, as it already is in Chrome, that's just really good news.

So it's in public beta at this point. And again, we'll keep our eye on it, and I'll certainly let people know. I imagine what'll happen is it'll simply be, as soon as it comes out of beta, it'll be a version update for Flash, the instance of Flash which is used in Firefox. And then all Firefox users will just get the benefit from it. So that's really cool.

Leo: Yay.

Steve: Yay. We had an interesting breakthrough in storage capacity and speed. Some of the headlines were a little bit maybe over the top, saying that a new disk storage technology could store 200 gigabits per second. So...

Leo: Wait a minute. 200 gigabits per second.

Steve: 200 gigabits per second.

Leo: So that's like 20-plus gigabytes a second. That's crazy.

Steve: Yeah.

Leo: Crazy talk.

Steve: That's way fast.

Leo: Yeah.

Steve: Well, so, however, I was curious. So I drilled down, and I thought our listeners who have an interest in disks and storage technology and so forth, however they have identified it, there's a metal called "gadolinium" whose symbol is Gd. It's a silvery white "lanthanoid" metal...

Leo: I think you're making this up now.

Steve: ...with an atomic number of 64. I kid you not, Leo.

Leo: Gadolinium?

Steve: Gadolinium.

Leo: Gadolinium, all right.

Steve: Gadolinium, G-a-d-o-l-i-n-i-u-m. Gadolinium and iron were combined. And in yesterday's issue, just published on February 7, yesterday's issue of Nature Communications carried the very technical report titled "Ultrafast heating as a sufficient stimulus for magnetization reversal in a ferrimagnet." Now, this is an amazing team of scientists in the U.K., Spain, Switzerland, Japan, Ukraine, Russia, and the Netherlands. Conspicuously missing, unfortunately, is the United States. So we need to up our science education, I think, in this country.

Anyway, the summary at the top of this article says: "The question of how, and how fast, magnetization can be reversed is a topic of great practical interest for the manipulation and storage of magnetic information." In fact, stopping for a second, that's one of the problems with storing data fast on a magnetic surface is it takes time to, under a magnetic influence, to collapse the existing field and reverse that direction of magnetization which is what's necessary for changing what's recorded on the disk. So there's a clear sort of inertia almost that you have to - the magnetic field is self-reinforcing, and it fights change.

The reason magnetic fields fight change is the reason you can generate electricity with them. It's that it takes energy to make that change, and you're able to capture that and spit out electricity. So it's all - and in fact that's what an inductor does. You have a ferrous rod of some sort, and you wrap wire around it. And when you run current through it, it builds up a magnetic field, and it resists a change to that flow of current, which is the reason inductors are used in power supplies in order to smooth out the flow of power. They actually store that energy and resist its change.

So anyway, going on, it says: "It is generally accepted that magnetization reversal should be driven by a stimulus represented by time-non-invariant vectors such as a magnetic field, spin-polarized electric current, or cross-product of two oscillating electric fields," something of course we all knew. "However, until now it has been generally assumed that heating alone, not represented as a vector at all, cannot result in a deterministic reversal of magnetization, although it may assist this process. Here we show numerically and demonstrate experimentally" - they've actually done this in the lab - "a novel mechanism of deterministic magnetization reversal in a ferrimagnet driven by an ultrafast" - and we're talking ultrafast, I'll cover that in a second - "heating of the medium resulting from the absorption of a sub-picosecond laser pulse without the presence of a magnetic field."

Okay. So turning that a little bit more into English, what these guys have done is they are using an incredibly short pulse of coherent optical radiation from a laser. The duration is 100 femtoseconds, Leo. Now...

Leo: What's a femto? That's like a billionth?

Steve: Okay. We know that milli is a thousand. Then we've got micro is a million. Nano is a billion. Pico is a trillion. Femto is next.

Leo: That's a quadrillion.

Steve: So that's one-thousandth - a femto is one thousandth of a pico. And so this is 100 femtoseconds. So this is short. And thus the reason for the somewhat overly dramatic recording speeds because they're saying, well, if you can write a pulse in a femtosecond, then how many of those can you fit on the head of a pin? Or...

Leo: Something.

Steve: And so forth. Anyway, one doctor, Dr. Alexey Kimel, from the Institute of Molecules and Materials in Radboud University, was quoted for this, saying, "'For centuries it has been believed that heat can only destroy the magnetic order. Now we have successfully demonstrated that it can, in fact, be a sufficient stimulus for recording information on a magnetic medium.' The technique is still a long way from being able to match conventional hard disk drives in terms of cost and ease of manufacturing, due to the use of the laser and specialized materials." You know, that gadolinium, you've got to come up with a bunch of that. "But the team is now refining the technique and feels it has scope for full production."

Leo: So what do we get?

Steve: So we get - terabytes are no longer a challenge because we will have femtosomethings.

Leo: Something or others.

Steve: And these things will suck in data. So, I mean, you don't even have to bother trying to decide what to record and not. You just record everything all at once, all the time, and it will never fill up, and you can never do it too fast.

Leo: Wow.

Steve: So basically we're...

Leo: We're getting there.

Steve: ...heading down toward molecular-level recording and playback.

Leo: Will it look like a solid-state device, kind of, sort of?

Steve: Apparently, well, no, it is, I mean...

Leo: It spins.

Steve: Yes. We see over and over that a disk is an efficient way of presenting repetitive material to a head. Disks always win out over tapes, for example. Historically, we keep trying to do tapes, and we always come back to disks. Remember the eight-tracks that we had in our cars for a while.

Leo: Yeah, tape is too slow to random access.

Steve: Yeah. And so it would be a disk. They have to figure out how to read it back out again. But it can certainly record a phenomenal amount of data. Basically it hugely increases the storage density on a disk surface and the speed at which it could be laid down. So that's very cool.

Leo: Very interesting. And how far off do you think this is from the...

Steve: Oh, well, this is further off than any of these other new solid-state technologies we've been talking about. And the problem is, of course, cost. And it's not clear what you need this for. I mean, it's like, okay.

Leo: Oh, we'll find something. Believe me. That's not going to be the problem. If you build it, we will use it.

Steve: We will fill it.

Leo: We will fill it. That's it. That's the slogan right there.

Steve: So I have a one-liner here just to note, for any people who don't know, that the Chrome browser is now available for Ice Cream Sandwich Android.

Leo: Yeah. So I've got to find one of them.

Steve: Yup. And is Ice Cream Sandwich just Android v4?

Leo: Yeah. So the only phone right now that uses it is the Galaxy Nexus, which I do

have, and then - somewhere. And then some tablets use it.

Steve: Do you know whether, is it the Droid 4, I think it is, that Verizon's going to be coming out with?

Leo: No. Nothing that is currently announced comes out with Ice Cream Sandwich yet.

Steve: No kidding.

Leo: Except for the Galaxy Nexus, which is a Google phone, so it's got a private place.

Steve: Okay. But will those be...

Leo: But tablets are coming out with it.

Steve: Will they be upgradeable to it?

Leo: Yes. Almost anything you buy today will be upgradeable. Not a lot of what you have from yesterday will be.

Steve: Yeah, I don't have anything yesterday, so...

Leo: Yeah. But I don't know if that's anything to jump up and down about. We've been talking about it for a long time. But it's all WebKit. So I'm not really sure what the...

Steve: Well, what I did read was that, from reviewers who were using it, they were jumping up and down, saying they're not going back to the original browser in Android.

Leo: Oh, good. Well, that would be nice, all right.

Steve: They said that there was a ton of innovation in this, in the tab handling. I mean, a lot of it didn't make any sense to me because I don't have existing experience with the current Android browser. But apparently it's like, why didn't they do this sooner? And so they've done it now, and that seems like a good thing. Especially if you'll be able to run ScriptNo on it, which we'll talk about in a second.

In other Chrome and Google news, one of the Google security guys surprised people recently by saying that they're going to remove all SSL revocation checking from

Chrome. Now, I consider that reasonable. We've talked about, endlessly, SSL certificates and certificate authorities and so forth. And remember that when we've had these disastrous breaches of certificate authorities that we've covered in the recent past, actually, where a bunch of certs "escaped" or were created without their knowledge, what in fact the browser manufacturers did was build knowledge of the now-revoked certificates into the browser itself.

That's how we found out, actually, what was going on was some sharp-eyed researchers noticed by comparing deltas of source code that, whoa, wait a minute, what are all these certificates that have been added to the browser's own "do not believe these certificates" list? And in fact, I can't remember now who, because it was Chrome who had this. Oh, it was Mozilla. Mozilla got upset because they weren't in the loop and weren't informed of this as quickly as IE and Google were.

But anyway, what we have now is in the certificate which is signed is a URL which goes to a server maintained by the signer of the certificate, which allows a browser to check in real-time to verify that the certificate hasn't been revoked by the authority that signed it. Remember that certificates have a couple-year, sometimes one, sometimes two, sometimes three, but no longer than three, they have an expiration. So, I mean, it's an annoyance that certificates expire because it means that those of us who have and maintain certificates are being continually, it feels like, forced to renew our certificates. One of the nice things about that is that, from the standpoint of the people relying on the validity of those certificates, is that we are forced to renew them, meaning we have to go jump through hoops, reverify our identity, show that we're still around, pass some security checks and so forth, in order to get the certificate renewed.

From the browser's standpoint, if the only mechanism that exists is expiration, then clearly the designers of the system realized there's a time horizon of vulnerability, from the time that a certificate was compromised, if that happened, to the time that that certificate expires. So the reason we don't want to issue certificates that last a hundred years is then they would never go away. They would never die. And if that certificate was ever compromised, then we'd be in trouble. Essentially what that would mean is that all technology that relied or might be called upon to rely on that certificate would be forced to know it. That is, it would have to be built into all of our browsers for a hundred years.

So the beauty of the short, multi-year expiration is that, for example, even in the case that I was talking about a second ago, where all those certificates escaped control of the signer of them - that was DigiNotar, you may remember, that whole DigiNotar debacle - they escaped their control, well, all of those certificates, even though they're maliciously created and cannot be trusted, they will absolutely be expiring within a couple years. So that means that the browsers only need to maintain their internal knowledge never to trust a certificate with this serial number until it otherwise expires. Because then they wouldn't trust it anyway, and they don't need to keep specific knowledge of that certificate.

So because the designers of this whole system recognized there was still this vulnerability window, they created the notion of a revocation, where the certificate contains a URL pointing back to a server maintained by the signer. And the browser, if it wants to, in real time, verify the trust, can reach out and query the revocation server to see if this particular certificate that's been signed by that authority and was just received, typically, from a foreign website - that is to say a website somewhere else, not necessarily in a different country - verify that it is not revoked.

Now, there's problems with that. Several, actually. One is that this slows down everything. Any time you use SSL, and the browser's running at maximum security, it's

going to, during the initial establishment of a connection, when it receives the SSL certificate from the remote server, before it does anything else, if it's going to honor this level of integrity checking, it must then create a new connection, which also must be unspoofable, so that's got to be SSL, to the URL contained in that certificate, specifically to check and ask if it has been revoked since it was issued and before it has expired. Well, the problem is revocation servers are not reliable. They're also not fast. And if you ever turn your browser into a "check revocation before doing anything else," you will uncheck it before long.

Leo: Yeah.

Steve: Because it's painfully slow.

Leo: Slow.

Steve: And remember that browsers are not doing a single connection to get a page. They're getting that first page, and that page is littered with other resources that they then have to go and reach out and get. And if those other resources are not all also SSL, then you get the horrible mixed-content warning, warning you that some of the things your page is trying to get are not secure, even though the page is. Well, that freaks out users. And so nobody wants to have their web pages cause that. Which means all the other connections leaping out of your browser to go get pieces of the page have to also be SSL. And if they're coming from different servers, all of them have to have their certificates checked for revocation.

And the bigger concern, or an equal concern to performance, is privacy because, now, notice that your system is making a connection to a revocation server, so they know your IP, and they know what certificate you're asking about. So anyone monitoring those revocation servers knows you by IP, at least, and where you're trying to go. So that creates privacy concerns. And it turns out that in the event that a connection is not received from any revocation server, the browsers all fail in the trusting direction. That is...

Leo: Well, that's wrong.

Steve: Of course it's wrong. It's ridiculous. So it doesn't work anyway.

Leo: Oh, that's the - forget it, then. Okay, never mind.

Steve: They go through all this, and then you still don't get protected because, if they don't answer the phone, it's like, oh, well, it's probably fine.

Leo: Yeah. I'm sure it's okay.

Steve: Give them their page.

Leo: Is that just temporary until this is widely adopted? I mean, is there some reason?

Steve: No, Google's taking it out. No, this has always been there.

Leo: Oh. Well, no wonder.

Steve: And there is a setting in Firefox, which is off by default, where you can say "do not display pages unless we get affirmative verification of non-revocation." And it's like, don't turn that on because, I mean, you'll think, what happened? I mean, it's just - it's ridiculous. And so the browser, I don't even know what would happen if later on it came back as revoked. I mean, maybe a new kind of warning comes up. But we've never seen that before because what happens is, if the browsers are being asked to wait too long, then they just go, oh, well, it's probably fine. So I salute Google. I think this is...

Leo: That must be broken. It's not - couldn't be anything wrong with the certificate or anything.

Steve: Anyway, so what Google said is that they're going to do what they've already done, and they're going to rely on automatic updates. I mean, that's what Chrome does. Chrome is already updating itself all the time. That's the way they handled the DigiNotar problem was they immediately added knowledge of the revoked certificates to the browser, and it just knows not to believe any of those. And since the certificates die after a couple years, it'll be a constant pruning process. After the certs die, they can remove them because then the browser won't trust them because the date stamp will say this has been expired. So I think it's an interesting change. This is an example of a problem that did not have a good solution. You need to issue these for a while, and you hope that nothing happens bad in the meantime.

And in fact I had an experience once where, I don't remember now why, but I wanted to reissue a GRC certificate. This was, like, five or six years ago. And so VeriSign, who I was using, was happy to let me fix something that was wrong with my cert. I don't remember now what it was. And they revoked the one that was only a few days old. And, ooh, boy, I started getting questions from people saying, hey, why is your certificate revoked? And it's like, what? Ooh. I mean, so somebody was noticing that a certificate was revoked. And VeriSign had to add it to the certificate revocation list because it was no longer valid, even though it was still me and it was fine, but they had issued a replacement for it for whatever reason. So it's like, ouch. There are some consequences to that. And so I think what Google's doing is a good thing.

A group of industry leaders, including Google, Microsoft, Facebook, LinkedIn, AOL, PayPal, and Yahoo!, among others, have formed an organization and a new spec called DMARC.

Leo: I like the name.

Steve: Good name. Stands for Domain-based Message Authentication, Reporting, and

Conformance. And the site is DMARC.org. What this is, is an agreement, finally, about email authentication. So this is good news. We've had two standards that we've never covered in the podcast because they've sort of annoyed me, and I've been waiting for something to shake out. And so it finally has. And so we'll do a podcast shortly to explain what this is. We had the SPF, the Sender Policy Framework, and some people sort of went with that. And then the alternative technology was domain keys, which Google among others famously went with, and that was called DKIM, which stood for DomainKeys Identified Mail.

The idea is, behind both Sender Policy Framework and DKIM, are that it's possible for a generator of email, someone who's going to be doing mailings - and of course all these companies do. Facebook wants to send email, and LinkedIn and AOL and PayPal and so forth. And they don't want it to trip over spam filters because one of the consequences of the spam problem we're having to see, and you see this all the time, is people saying, well, we've just sent you email. If you don't get it, please check your spam folder because we really did send it, and you really do need it. So it would be nice if there were a way for valid mail to authenticate itself in a way that spammers could not duplicate.

And that's what these technologies provide. And this DMARC is pulling this together under a single umbrella, essentially. And DNS, this is another example of a benefit of DNS, why we really need DNS not to be spoofable, is that someone who's sending mail will be able to add some records to DNS such that somebody who is receiving the mail from a given domain can ask the DNS system for the data that's necessary to verify that this mail was actually sent by the person controlling that DNS information. Since the owner of the domain is hopefully the only one controlling the information that their DNS servers are providing, this provides essentially the necessary crypto information for verifying the sender of email. So this has been a long time coming. It's great that it's going to be pulled together under a single umbrella. And we'll do a podcast before long to explain how it all works, in detail.

Leo: Good. And you support SPF. Somebody's saying that you've had an SPF record on your server for some time.

Steve: Yeah, like years, yes. And it turns out it's very easy to do, not difficult at all. The domain keys is a cool technology that I want to add. And I think it makes sense to do all of that.

Leo: Great.

Steve: Okay. Now. The webcam nightmare of all time.

Leo: Quick, give me the search engine. I want to find it before people realize this is going on.

Steve: Okay. So a random hacker who is pretty good at his craft owned a TRENDnet webcam. And as are all these things, there's going to be Linux inside somewhere, and that means there's a file system, because this thing's got a little web server in it. And so it's remotely accessible over the web. You can log into it with your browser, and it brings up a little configuration page, much like one of our standard Internet routers does. And

you give it a username and password because of course you don't want everyone able to log into your webcam. And you can put it on the 'Net, or you can map port 80 through in order for you to access this remotely.

And so when you're out traveling around, and you want to check on whatever the camera is aimed at, you browse to your IP address. And when you attempt to connect, you get challenged by a standard, you must provide a username and password authentication challenge. And when you provide that, oh, look, there's a window showing you what your webcam is seeing in real time.

So this guy just opens up the file system, and there are tools now for parsing firmware-based file systems, lots of tools actually, that router hackers use in order to do things like create the Tomato firmware and so forth. They want to pull apart what Cisco has provided, see what's there, understand how it goes together, and then add their own functionality and replace things. And then you can even rebuild these file systems and then reflash them so that the router doesn't know any different.

So this hacker looks at the file system, sees some of the files, and notices in the root of the file system, or I guess it's under a directory called "anony," short for anonymous, anony. He sees a CGI script, mjpg.cgi. And so he, playing with his camera, he just calls it up. He puts in the URL of his IP, 192.168.1.17 in this case, /anony/mjpg.cgi. And not surprisingly, lo and behold, he's looking at, in real time, what his webcam is showing him.

So just sort of for grins, he thinks, I mean, it occurs to him, wait a minute. I just brought this up, and I wasn't challenged for a username. Because normally when you go to a web server, you put in Google.com/. What your browser is doing is asking for the root, the root URL, the root path to the server. So that would be 192.168.1.17. So if you just put that in, you're prompted for your username and password. And that's what anyone trying to get to it, either from inside the network or out on the Internet, would have to do. But when you put in the exact path to this CGI script, it didn't ask him. Turns out it never asks. Even if you're on the Internet. It's no different. The web server in the camera doesn't know where you're coming from and doesn't care.

So it turns out that the Internet is filled with - the manufacturers said, well, probably not more than 50,000 webcams where if, instead of just going to the IP, you go to the IP/anony/mjpg.cgi, you don't have to authenticate. And it shows you right then...

Leo: That's convenient.

Steve: Yeah, it's so handy. It's much easier to just create a shortcut on your browser. So then our illustrious hacker uses a really clever new search engine that's been around for almost two years. They're celebrating their two-year anniversary a little prematurely because the domain name was registered 10 days from today, two years ago, February 18th, 2010. There's a cyberpunk online realtime game called System Shock.

Leo: Yes.

Steve: And System Shock 2.

Leo: Yeah, good game.

Steve: Yes. The entity, the automated entity in System Shock is known as Shodan. That's an acronym for Sentient Hyper-Optimized Data Access Network. There is now a web server, or a website, called ShodanHQ.com. So Leo, go to www.shodanhq.com.

Leo: And everybody else don't. Okay, go ahead. Because I'm sure...

Steve: Oh, they will.

Leo: Oh, too bad. We're going to bring it right down, I'm sure.

Steve: Okay. So now, Leo, you should see a little search box.

Leo: I will as soon as everybody stops going there.

Steve: Oh. You want to put in - just put in something that I...

Leo: It got real slow, real fast.

Steve: Oh. Sorry about that. Put in what - I have something in my own web server headers, GRC/IIS.

Leo: Okay. Give me a moment, it's still coming up. You know what, Steve, you're just going to have to describe it because it's not going to come up.

Steve: Okay. You can do it after a while.

Leo: We broke - and everybody at Shodan is going, what the hell?

Steve: Yeah.

Leo: What's all this traffic?

Steve: Okay. So here's what Shodan is. Shodan is a hacker's dream come true.

Leo: Oh, neat.

Steve: Yes, it is neat. This thing is a search engine for all of the connection headers that servers all over the Internet generate. When you connect to a web server, you get back its "Hi there, this is what I am, this is the type of web server I am, here's what I know." If it wants you to provide authentication, it'll challenge you and ask you for the so-called "realm" with a WWW-Authenticate header. When you connect to an email server, it responds with its identity, often its version number and so forth. Same thing with telnet servers that respond to a connection with a response. Most servers on the Internet, when you connect to them - because once upon a time you would, like, telnet to something, and so you would just give it the IP. And as you know, Leo, it would respond with a couple lines of greeting, saying this is where you have connected to, and often the make, model, version of the server software that you're using.

This ShodanHQ.com site has indexed all of that. As a consequence, it has indexed all of the TRENDnet webcams that are on the Internet. And if you search for simply "netcam," it is a case-insensitive search. So you could put in capital "N" Netcam. There are other netcams that use the basic realm "netcam." The lowercase netcam are the TRENDnet netcams. You will then find 2,734 of these in the U.S., 1,309 in Germany, 961 in France, 717 in Hong Kong, 620 in Mexico. Just shy of 10,000 IP addresses of TRENDnet webcams, any of which you can log onto and see what the webcam is seeing by putting in the IP address/anony/mjpg.cgi. So this is a...

Leo: This is so handy.

Steve: This is a disaster. Oh, goodness. So ShodanHQ is kind of cool. If you put in GRC/IIS as your search, you'll see six IPs of GRC web servers that I run that respond with that as part of their header. And you can put in Cisco stuff or random, I mean, Linksys stuff, I mean, anything. This thing is just - it's really a very cool search engine for anything that responds when you connect. I think you can only do web unless you subscribe. But after you get a subscription, then you can do things like telnet searches to search for telnet servers and get up to all kinds of mischief.

So if anybody happens to have a TRENDnet webcam, or you know anybody who does, you want to make sure, well, actually you just - you can't use it. This is the latest version of firmware, by the way. The company, TRENDnet, is scampering around, as you may imagine. The problem is very few people register their TRENDnet webcams. So they don't have the email addresses of the owners of these, and there's no way to notify them by their IP address, which now the world has since this ShodanHQ.com search system is able to find all of the TRENDnet webcams, since they're just little servers there that say "netcam" in their headers when you attempt to connect to them. Oh, goodness.

Anyway, so there will be firmware updates. If you do have TRENDnet webcams, or you know somebody who does, take them off the 'Net right now until you update them and make sure that they've solved this problem. And, you know, whoops. Very clever, interesting little hack with some serious privacy consequences. Anybody who's aiming their camera at something sensitive and hoping or assuming that their authentication is providing them with enough security, it's not the case. It's not good. But I do have something that's good.

Leo: Yes.

Steve: I got email on the 5th, which was what, three days ago, from a Craig Thompson,

who sent email to my sales email address, and Sue bounced it to me, saying "SpinRite 6 saves client's files." And he said, " again." And he said, "While I've been using SpinRite for several versions now" - which means a long time because we know I don't do versions often - "and have managed to recover files from seemingly dead drives in the past. This weekend's experience was by far the best. I'm a small, one-man computer business, so quality affordable utilities are a must for me, and SpinRite has paid for itself over and over again.

"A client dropped off an external hard drive that had died. She was crazed, as she stored, instead of backed up, numerous important files to this drive. Losing it all would have been devastating. After trying a number of other quick fixes, none of which worked, including a relatively simple Level 2 SpinRite scan, I finally opted, as a last resort, to run it up to an 18-hour Level 3 scan. After running all day and night, I reconnected the drive to my system; and, voila, all of the folders and files suddenly reappeared.

"I'm copying off now to return her files to her. I'm ecstatic that I was able to recover this data for her, and she's ecstatic that she didn't lose everything. In fact, she didn't lose anything, all due to the continually outstanding utility, SpinRite. While the cost of this app may seem a little steep, for small-time people like me it's well worth the investment, and my clients couldn't agree more, given the results. Thanks very much, Steve and everyone else at GRC, for such an outstanding tool. Very sincerely yours, Craig Thompson."

Leo: Happy, happy day.

Steve: So thank you, Craig, yes.

Leo: So I don't have an ad, so if you want we can launch right into ScriptNo.

Steve: I want to tell everybody about ScriptNo, indeed. It is as easy to install as any Chrome extension can be. You just search for ScriptNo. In fact, if you search Google for ScriptNo, you'll find a couple references to it and can locate it quickly. Andrew Young is ScriptNo's author, who is a recent honors graduate with a Bachelor of Commerce in Business and Technology Management. His own site is Andryou, A-N-D-R, instead of E-W he does Y-O-U. So Andryou.com, which is kind of fun because his last name is Young, and so that's the first four letters of his first name, the first three letters of his last name that make Andryou spelled differently. So, and that's his Twitter handle, also.

So as I mentioned before, he was originally using Firefox, Opera, and Chrome. And Firefox 4 drove him nuts, as it drove me nuts, and as many people have reported because of its horrendous memory leaks. And most recent reports are that, even though they're now on Firefox 27 - I'm kidding, of course. It's, I think...

Leo: You know I believed you for a second?

Steve: I know.

Leo: Oh, yeah. Really. 27, wow, that's...

Steve: They're on 10 or something.

Leo: They'll be at 27 soon, don't worry.

Steve: Well, 12 is projected for April, so, yeah. They still - this really seems to be causing a problem. And I love Firefox. I'm still using it as my primary browser. One of the things that has kept me from moving to Chrome is that I just didn't feel like I had the instrumentation that I liked, that I've gotten used to thanks to NoScript running over on Firefox. One of the reasons I don't use Chrome more is I sort of only use it when something doesn't seem to be working right with Firefox, or, I don't know, I want more compatibility. Sometimes I've had problems playing videos in Firefox for whatever reason, and so I'll just jump over to Chrome, and it always works in Chrome.

So I was missing this feeling of knowing what's going on with a web page. If a web page is having problems, I can click on the little icon on my toolbar under Firefox and see an enumeration of all the other domains which are trying to provide stuff to this page, and kind of look at them and go, eh, I don't think so. Or say, yeah, okay. Or trust them briefly, trust them just for now and so forth. Well, there was something called NotScript which I got excited about some time ago, until I tried it, and I immediately removed it. It just - it was low function, didn't do what I wanted. It wasn't a replacement for NoScript.

Well, we have one now called ScriptNo for Chrome. And it's under continuous development. I suggested a couple things in my email conversation just yesterday with Andrew, one being that he blocks, by default, he blocks the <NOSCRIPT> tag, which is the text which shows when scripting is disabled. And it's very handy. It's one of the ways that websites can say, and I'm sure people who use NoScript over on Firefox have seen this often, which is a little bar will show or something will happen saying this site requires scripting for its full function.

And of course as my own work has moved more toward scripting, like with Off The Grid, which is local JavaScript, and the Password Haystacks page, if you try to bring up the Password Haystacks page with scripting disabled, I give you a nice banner explanation of why just GRC in general doesn't need scripting, but this page specifically is a scripted page because of the features that it offers. So it's not like I'm just obviously forcing people into using scripting all the time. I haven't gotten to that level yet. But certain pages that have a use for scripting need it.

So anyway, the point is that he agreed. He says, oh, he says, yeah, in the next release I'll turn that off by default. And that seems like a good thing. But in general the add-on is enabled by default. It's set to block unless you permit. So that's the technology that we learned works with firewalls and works with security is by default you say no, and then you selectively enable. He's got essentially all the features that you need in a script blocker, meaning that you can do selective blocking and unblocking of specific domains. You get the enumeration of all the things that the page is trying to do to you. You can temporarily enable, so that you're not filling up your whitelist with an endless list of junk, like sites you had to enable in order to use it, but you're never going to go back to it, so why keep it around? So that's one of the things, I mean, I'm always, over on Firefox, I'm using the "temporarily allow scripting" on sites that I know I'm just there for some one-off purpose. Why add that to my permanent whitelist? So he has that.

He also does something, for those of us who really care, which Firefox NoScript doesn't, which is in the enumeration of the domains that are involved in the page, the main page you're going to, it'll show you how many things, how many assets of what sort that domain is trying to load. And if you hover your mouse over the domain name, up pops a tool tip showing you the asset type and the URL, which is lots of nice information. Again, it helps you to go a little bit further in understanding, well, okay, what is it that this domain is trying to load? Is it just an image? Or is it JavaScript? You can't tell that over on Firefox with NoScript. With ScriptNo, it's easy to see that.

And he has another very cool thing that I like is he has a reputation button that is also there. You're able to turn it off if you don't want to run with it on. He calls it "ratings." And it takes you over to the social networking Web Of Trust site automatically for the domain that you're asking for a rating on. So if you see something, and I have, just in the time that I've been using it, something-or-other.fmqodt.com, it's like, well, what the heck is that? And so you can right then just click on "Rating," and you'll quickly see, and in the case of the thing that I clicked on that was an acronym like that, it came up red screen, and it said "This is a mistrusted site that is not liked by people." And of course then I clicked on GRC.com because I was curious what it said, and it's like all green and happiness there. So...

Leo: Yay.

Steve: I'm sure the same thing would happen with TWiT. So you can then...

Leo: Well, I don't know about that. I wouldn't go too crazy.

Steve: So you can block scripts, objects, embedded objects, iFrames, frames, applets, audio, video, NoScript tags, and images. Images are allowed by default, and NoScript tags will soon be allowed by default. As I said, the rest by default are not allowed for things that are blocked. But then when you allow them - he also does a nice thing, for example, sites that use lots of subdomains - like Google has, like, subdomain du jour. It's one of the reasons I was forced over on Firefox to stop using Certificate Patrol was that Google's got different certificates and seems to just generate subdomains on the fly. So it'll just be randomstuff.google.com. And Certificate Patrol, it says it had a wildcard system, but I could never get it to work, where it was, like, trust everything that is underneath Google.

Well, ScriptNo does that, so you're able to Allow, which is the option for allow just exactly this domain, or you can say Trust, which says essentially allow everything under the root domain. So anything.grc.com, not just www, but it would also allow then media.grc.com and other, for example, in the case of GRC, other of my subdomains. And again, that means you don't have to have individual whitelist listings for every one, and you're not having to constantly be permitting other instances of subdomains. So that's a very nice feature.

He also automatically pulls, there's an option under Privacy Settings for blocking unwanted content, as he calls it. And he pulls from a bunch of known ad and malware domains gathered from the MVPS HOSTS; hpHOSTS, which are ad and tracking servers; Pete Lowe's HOSTS Project; MalwareDomainList.com; and the DNS-BH, which is a malware domain block list. So his utility is also pulling from a bunch of lists of known

problems. And he absolutely keeps those blocks blocked for you. Also he has an option that I also appreciated - he calls it "Antisocial Mode" - which is disabled by default, but he recommends it, and I like it. It removes all of those social widgets and buttons, even if the site is whitelisted. So things like the Facebook Like and the Twitter and the Google+ buttons which are appearing more and more. I mean, those are just not things that I use, and so you can turn that Antisocial Mode on and be antisocial. And you won't be seeing all of that stuff.

And of course remember that all the things that are blocked are things that never have to be loaded. One of the reasons that it's been difficult for add-ons to be created for Chrome is that Google has been only slowly creeping forward the API that allowed more powerful add-ons, or extensions as Google calls them, to be added to the browser. Part of the sandboxing approach has made it difficult because individual tabs or pages run in their own processes, so you need some sort of cross-process reach that's much more difficult to do with that process per page model than it is in Firefox, where it's just all lumped in a single process from an actual standpoint of getting access to the page contents. And Google also - the Chrome API still has some things which it doesn't allow.

So an app like ScriptNo and through Andrew's research, he's able to block most things, but there are some things that cannot today be blocked. So he has to wait until the page comes in and then remove them dynamically from the page. That's technology over time that's going to evolve. And in fact he mentioned last night that there are two new APIs which he can't wait to have access to which are currently in beta for Chrome. And as soon as they stabilize and go mainstream, he'll be able to use those in order to evolve ScriptNo further.

But bottom line is - oh, he also removes web bugs. That's an option that's enabled by default, and you can turn it off. And he says he just removes invisible third-party elements, things that are there just for tracking. And it's interesting because, as I've poked around the 'Net, and this thing has identified web bugs for me, I've hovered my mouse over, and I can see the URL. It's like, oh, sure enough, look where that's going off to. Somebody on that page deliberately planted something just for the purpose of tracking me.

Oh, and speaking of which, another option is removing the referrer tail from non-trusted sites. So if you click on a link that takes you to a site that ScriptNo is not trusting, I think it replaces it with something like "no referrer," so it does put a little tag in just to say I've removed something. But if it didn't do that, it would be sending that site that you don't trust, as we know, the entire URL appearing up in the browser's title of the page that you're coming from. So that's a nice little privacy feature.

And of course he also allows you to export all of the settings that you have customized over time and import them so that you're able to move them to a different instance of Chrome running on a different machine, and the same thing with your whitelist and blacklist settings. So essentially we have now on Chrome a state-of-the-art, very nice-looking, pleasant to use, script controller on a par with what we've had for NoScript. And I'm really pleased, and I wanted to make sure all of our listeners knew because I know a lot of them are going to enjoy this level of control.

And if you like it, if you get a chance, drop Andrew a couple bucks. He's got a donation button, a little heart icon on his pages. And he says, "Support Andrew's Existence and Sustenance." And in fact that's how we struck up a dialogue was I sent some money to him because I definitely want him to know that this thing matters. And in fact supporting him would be one way of saying yes, yes, yes, please keep working on this, this is important. We need this for Chrome. It really is how I feel.

Leo: Neat. I just downloaded and installed it.

Steve: Cool.

Leo: Yeah.

Steve: Yeah, it's pretty and nice and works. And I'm sure it will be getting features. And he's very responsive to feedback, too. So if you see things or have ideas for it, let him know.

Leo: Yeah. Looks good. Looks really good, yeah. Going to be a nice thing to have.

Steve: I know you're not a big script blocker.

Leo: Well, because it just screws everything up.

Steve: And I understand that. But believe me, I know how many people are going to be delighted that we have this capability for when you want it for Chrome.

Leo: Yeah. I'm just looking at all my sites. Of course there's lots of little things on there because we use a lot of JavaScript and things like that, Flash. Turn it on. You can trust our sites.

Steve: It is neat, this nice little auditing tool, too.

Leo: It is, it is.

Steve: You can really see all the stuff going on.

Leo: It's great, yeah. I'm just looking at, like, for instance, I thought Facebook was bad. Then I went to my site. And it's quite a long list of things. APIs, there's Google, there's Facebook, there's Widgets Plus, there's Twitter, there's QuantServe, there's widgets...

Steve: It adds up, doesn't it, Leo.

Leo: Yeah. Well, these are all the little doohickeys I'm running on the - 15 of them. Oh, boy. There were only two on the Facebook site. Trust, trust, trust. Now, if I click Trust, it trusts it forever; right? It's not just for now?

Steve: Correct.

Leo: Yeah. But if I say Allow, that's just a temporary one.

Steve: No, there's a Temp button, if you look down lower. Allow is just the exact URL, the exact domain that you're at. For example, live.twit.tv is different than TWiT.tv. If you do Allow, it only allows that one explicit domain. So you want to do Trust because you trust everything that is TWiT.tv.

Leo: Maybe I shouldn't. Who knows what they've been doing? Very good stuff. It's free, but of course support the author. As always, it's always a good idea.

Steve: Yeah, really do. I sent him a hundred dollars, which sort of knocked him over.

Leo: Wow, good on you.

Steve: But I was able to do that because our listeners have supported me through SpinRite over time. So I don't expect everyone to do that at all. I thank everyone for their support of me and SpinRite. So I'm happy to pass that forward.

Leo: Pay it on.

Steve: Yup.

Leo: Steve Gibson is at GRC.com. That's where you'll find SpinRite, the world's best hard drive maintenance and recovery utility, and all his freebies, as well, including Perfect Paper Passwords and all that fun stuff. He also puts 16Kb versions of the audio up there for people who really don't have much bandwidth, but want to listen. We have transcripts there, as well. GRC.com. Next week we have a Q&A, so go there to ask questions. Don't email them to Steve. There's a feedback form. It's GRC.com.

Steve: Much better because then it's fun to know where people are, and email doesn't let us know where you are. So if you use the feedback form, you can say this is who I am, and where I am, and what accent you would like Leo to use for reading your question.

Leo: Yes. We are taking requests. GRC.com/feedback for that form. Steve will be back next week with a Q&A. You can watch us do this live, 11:00 a.m. Pacific time, 2:00 p.m. Eastern time, at TWiT.tv. It's 1900 UTC. But if you miss it, we've got audio and video of the show. Watch it in the format you prefer. TWiT.tv has them; GRC.com as well. Thank you, Steve. We'll see you next week.

Steve: Thanks, Leo.

Leo: On Security Now!.