SECURITY NOW!

Transcript of Episode #225

Same Origin Troubles

Description: This week Steve and Leo plow into the little understood and even less known problems which arise when user-provided content - postings, photos, videos, etc. - are uploaded to trusted web sites from which they are then subsequently served to other web users.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-225.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-225-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 225 for December 3, 2009: Same Origin Troubles.

It's time for Security Now!, the show that protects you as you surf the world on the World Wide Web. Mr. Steve Gibson is here. He is the man behind the Gibson Research Corporation, creator of SpinRite. You know, I was looking at your - hi, Steve.

Steve Gibson: Hey, Leo. It's great to be with you again. You know, we don't have a jingle, and I guess that's a good thing because...

Leo: No, you don't need a jingle.

Steve: ...that would sort of be distracting, like you do with Dick DeBartolo, to be, like, talking over jingles that were going in the background.

Leo: [Singing] You've got viruses, and Steve's got the cure. He's the man - no, that doesn't fit. It doesn't...

Steve: No.

Leo: I was looking at your Wikipedia entry. And it's funny, I mean, it's so funny to watch the discussion behind the scenes because there are people, I think it's one guy, but there's somebody who doesn't like you.

Steve: Yeah.

Leo: And somebody questioned calling you an engineer. And I just wanted to, you know, I wanted to, like, get on there and say, what do you mean, is he an engineer? The guy, for crying out loud, he built the light pen for an Apple II. I mean...

Steve: Yup, that blew Wozniak away. I remember that Steve Wozniak was at the Applefest, the Boston Applefest. And he stood there - I loved the pronoun he used. He looked up as I was demonstrating the light pen. And he said, "You always blow me away with what you're able to get my machine to do."

Leo: Oh, isn't that nice.

Steve: He was possessive about it. I thought, that's really neat.

Leo: Well, that's also, from Woz, very high praise.

Steve: And the real trick of engineering was that, whereas all light pens of that era were big cigar tube-size things...

Leo: I remember, yeah.

Steve: ...mine was the size of a regular little pen. And I had figured out how to mount the amplifier at the other end, back in the Apple. And I used something called a transconductance amplifier because the photo diode, you had to use a photodiode, not a photo transistor, because of the response time. A transistor's amplification makes a photo transistor way too slow to respond to the beam moving past in front of the screen. The photo diode can do that. But the problem is then you have to count electrons, literally almost like a photo multiplier. And to do that at the end of a four-foot cord, and to put the amplifier in the Apple, which is a noisy environment - anyway, it was a, well, I guess it was a feat of engineering. So, yeah, I think I qualify as an engineer, so.

Leo: Probable some people listening have no idea what a light pen is. In the days before styluses and touch screens you used light pens; right?

Steve: Yeah. Well, it was - well, actually I would say the day before the mouse because...

Leo: It predated the mouse, too. You're right, yeah.

Steve: Yes, exactly. The idea was I think it dated from maybe like NORAD or, you know, high-end display systems. And you'd - because I remember like in early movies that were showing, like, military complexes, there would be people in uniform, like, touching the screen with a pen. And the idea was - and those were old XY displays, where the beam jumped around all over the screen like painting air traffic control diagrams and things. And so you could, if you touched the pen to where something was on the screen, the system would notice where the beam was when the pen said, ooh, I just saw the beam go by. And that was a way of feeding back coordinates into the computer. So, yeah. And, I mean, it was my first real independent corporate success was the light pen for the Apple II. And actually there were some tremendous articles that were very flattering about how well it worked. And, in fact, Atari bought it from me because they wanted it for their Atari.

Leo: I didn't know that was yours. I remember the Atari light pen.

Steve: Yeah. I wrote all the software for it, too.

Leo: I'll be danged.

Steve: Yeah.

Leo: And part of the discussion was interesting, which is, well, what constitutes an engineer? How do you - do you have to have a degree in engineering to be an engineer? I mean, that doesn't make sense. An engineer is somebody who designs and builds stuff.

Steve: Yeah, I think of myself, I mean, sort of both scientist and technologist. I think of a technologist as an applied scientist. I like creating things.

Leo: Right. Well, that's what engineering is. It's applied science.

Steve: Yeah.

Leo: And that's what, you know, we often talk about that on the network. And it's one of the reasons we're doing more science stuff with Ray and with Kiki, because I think people forget the strong bind between science and technology.

Steve: It was probably the Windows XP raw sockets thing that upset...

Leo: That's when you made enemies.

Steve: That's when I made enemies.

Leo: And you were right.

Steve: I was right. But that doesn't really count.

Leo: I mean, that's the funny thing. Even Microsoft acknowledged, finally, yeah, it was a problem, and took it out.

Steve: And fixed it, yeah.

Leo: But, you know, damage done. But it's still - it's funny, I mean, when you see somebody saying, well, Steve's not an engineer, it's just you have to laugh at the person [indiscernible].

Steve: Yeah, I love where people who have said, yeah, he's only ever written one program, it's like, oh, have you been to GRC lately?

Leo: People often say things without any knowledge. You hear that all the time.

Steve: Well, and the other thing, too, is the anonymity. We don't know who these people are. I mean, everybody who matters knows who I am and what I've done. But the 'Net lets people with no pedigree whatsoever comment as if they had some. It's like, oh, okay. I mean, I just - I do what you do, Leo. I just ignore it, say okay.

Leo: Brush it off. What are you going to do?

Steve: Yup.

Leo: What are you going to do? Somebody's pointing out in the chatroom that "Jeopardy!" uses a light pen and has since 1984. That's how the contestants sign, you know, write their answers, Final Jeopardy answers.

Steve: Ah, and sportscasters. Weren't they doing things now with, like...

Leo: Telestrator is essentially a light pen, yeah.

Steve: Right.

Leo: Although now I'm sure it's touch, not light, yeah.

Steve: I would think so, yeah.

Leo: Before we get to our subject - what is our subject, by the way?

Steve: Our subject is - you wouldn't know from the title. Normally you do. The title is Same Origin Troubles. There's been a, I would say it qualifies as a kerfuffle in the security community which was launched about the middle of November, so about two weeks ago, by a security outfit who claimed that they had discovered some flaws in Shockwave Flash which made all websites that weren't specifically designed to protect against it vulnerable to user-uploaded content. And Adobe responded; and, I mean, there were articles in a number of major industry newspapers. SC Magazine had a comment; Computerworld had a comment. Of course the Register weighed in. And it brought to my attention the fact that we had never explicitly discussed same origin. Because it's a fundamental security aspect of the Internet that unfortunately most sites actually are not good about handling.

So because it really is a problem, even though I would argue that, sort of on Adobe's side, that it's not their problem, even though unfortunately Shockwave Flash is more permissive in a number of ways than it ought to be. I thought that would be a great thing to talk about is the idea of the danger associated with user-uploaded content and how that danger can be mitigated by making sure that when you serve back user-uploaded content, you don't do it from the same origin as the content you trust because inherently you should not trust user-uploaded content.

Leo: All right.

Steve: So wind up your propellers.

Leo: Fascinating. Yeah, no, this is very relevant.

Steve: It way is, yes.

Leo: Really great to talk about. Before we do that, I would like to - and we will have, of course, as always, our weekly dose of security news, errata, updates, things like that.

Steve: And I have a nice Thanksgiving story.

Leo: Aw. So will we start - you want to start with errata, security news, where do you want to start here?

Steve: Yeah, we've got a bunch of security news. Nothing, well, okay, yeah.

Leo: [Laughing] Something. A little news. A little.

Steve: I wanted to make sure that everyone using Safari had updated themselves. Apple recently fixed multiple critical vulnerabilities which affect both Mac OS X and Windows. There were multiple vulnerabilities in Safari's handling of just pretty much everything, a variety of web page scripting constructions and malicious images. There was an integer overflow error caused by improper handling of images, containing an embedded color profile. It's interesting, color profiles seem to be a problem for Apple. We've had a lot of, like, mentions of color profile problems in the past. I don't know what's going on. Someone ought to really look at that code closely and say, okay, let's just fix these things once and for all. Or, who knows, maybe they keep tweaking with it and messing it up, as our old code/new code dilemma.

Safari can be made to crash while parsing specially crafted XML content. And of course the concern was that a crash could be evolved into an attack, which is generally the way those things go. There was an error in Safari's handling of navigation which could cause a specially crafted HTML file to load a local file and lead to information disclosure, which is not good, depending upon what local file you load. They've discovered that the way cross origin, which is actually the topic of our content today, our podcast, the way cross origin resource sharing was implemented in WebKit could result in cross-site request forgeries, which we've discussed in a separate podcast before. So that got fixed. And the way WebKit handles FTP directory listings could lead to arbitrary code execution, information disclosure, or at least application termination. So that all got wrapped up and fixed in the latest version of Safari. Now, I wrote 4.0.0, but I think I meant 4.0.4.

Leo: Let me check and see what I, you know, because I'm sure I'm up to date. I am running 4.0.4.

Steve: Ah, good. Then my memory was correct, and what I wrote was wrong. So everyone wants to be at 4.0.4.

Not surprisingly, Internet Explorer, another one of our favorite troubled browsers, has a pretty bad zero-day remote code execution vulnerability. The bulk of the market share of IE is still at version 6 and 7. Even though 8's been out for a while, 6 and 7 are the ones that are vulnerable. Interestingly, in Microsoft's extensive note - they've acknowledged the problem. They know they've got it. Hopefully we'll get this fixed in December, on December 8th, which is next - which will be the second Tuesday of December and will be the Patch Tuesday for December. I'm hoping this gives them time to fix it. In their extensive write-up they said, well, but IE5 isn't vulnerable. In fact...

Leo: 5? Who gives a - 5?

Steve: I know. And in fact they even had to say that 6 running on Windows 98 was vulnerable.

Leo: Oh, dear.

Steve: Of course that won't get fixed anytime soon. But it turns out that there's a problem with the way CSS-style objects are invoked which surprised Microsoft. Exploit details and proof of concept of this exploit are on the 'Net. So that makes it a zero-day exploit. And we've often talked about the Metasploit framework, which is this framework that allows hacks to be created and deployed very quickly. The Metasploit framework exploit module 37085.RB will demonstrate this and allow people to do bad things to IE6 and 7.

And I think I remember reading that 6 and 7 still have, like, 80-plus percent of the IE market. So lots of people have not moved to 8. I mean, I was reluctant to do so. I waited a while. And pretty much now when I'm on one of my machines that's still on 7, it's like, okay, yeah, fine, I'll go to 8 now. I'm sort of getting used to it and the way it looks and so forth. And it's arguably more secure. Certainly in this case it is. So if anybody right now is still on 7, this might be a little good reason to kick yourself up to 8, which has become, I think, stable enough to use. In a little twist on Microsoft's second Tuesday of the month updates, remember that last update was November's, and that it was a massive update. A huge number...

Leo: Biggest in history; right?

Steve: Yes. A huge number of things fixed. And it turns out...

Leo: Broken. No.

Steve: Yes. There is now the Black Screen of Death.

Leo: Oh, yeah, I've been seeing a lot about this, yeah.

Steve: Yes. Apparently Microsoft tweaked in this massive update some ACLs. Those are Access Control Lists which govern what processes are able to read and write files and even registry keys in the registry. And Microsoft has acknowledged the problem. They're trying to track it down.

Leo: It's Windows 7 only; right?

Steve: Good question. It might very well be, although...

Leo: That was my sense, but I haven't done a lot - I haven't...

Steve: Yeah. It might very well be Windows 7 only.

Leo: And by the way, I don't think that widespread.

Steve: No. And that's what Microsoft is saying. It's like, okay, we're looking into it. But it's not, I mean, I updated everything. Although I'm not on Windows 7 yet, so it wouldn't have hit me. But, yes. If this had been a huge, huge problem, first of all, it would not have gotten through Microsoft's own pre-release testing. And the world as we know it would have come to an end if everybody running Windows 7 had this happen. But certainly, apparently, maybe it relates to one particular graphics driver. I think I remember seeing, it's like some ATI Radeon something or other, where you could say, okay, well, I could see how that could slip through Microsoft's testing. But it's never good when the security updates that we're being increasingly forced to install just out of our own instincts for self-preservation, when they go bad, that's not good. So...

Leo: It just underscores the difficulty, though, that Microsoft faces with this huge variety of hardware that they run on.

Steve: Yes.

Leo: Testing is a nightmare.

Steve: It's phenomenally difficult, yes. And in a weird story that I just sort of picked up on...

Leo: Just briefly, the chatroom is saying that on Slashdot now the story is being retracted.

Steve: Really.

Leo: Yeah.

Steve: Okay. So Microsoft said they were investigating reports.

Leo: And this might be the Microsoft result. This was a Neowin report. And Microsoft said they hadn't seen issues - okay. Prevx, the guy who found it, has issued an apology. They blame malware for making changes to the registry which causes this behavior. So they had malware on their system, making it an even more oddball environment.

Steve: Interesting. So, okay, so there was something bad there to start with, and then Microsoft's security updates come along, change some ACLs and piss off the malware, and so now other things don't work.

Leo: Yeah. Thank you to Avillafane and Mike and others who notified us of that.

Steve: Yup.

Leo: That just broke, so...

Steve: Cool. A pub in the U.K., a bar, which is offering free WiFi, free open WiFi, was fined $13,000 by the copyright holder of some content that the copyright holder claims was illegally downloaded. And that's caused a lot of news in the U.K. because, I mean, the idea of somebody offering free open WiFi being sued due to the conduct of somebody taking advantage of their open WiFi service, I mean, that's something that obviously would be huge if this were setting a precedent for the way the world's going to be moving forward.

Now, there's pending legislation in the U.K. which they call the Digital Economy Bill, which would provide protection because the business would be classified as a public communications service provider, which would make it exempt from litigation. That is, essentially they would be considered a common carrier and not responsible for the actions of people using their service. And then some legal opinion was obtained which - and the legal opinion came down and said that WiFi hotspots in public and enterprise environments that provide access to the Internet to members of the public, free or paid, are public communications services, which then would exempt them.

But at the moment there's not this legislation in place. And as far as I can tell, this fine is still in place. And I imagine the pub is not - has asked to remain anonymous and isn't discussing, isn't talking about the specifics. But it caught my attention because this does sort of, I mean, we're all massive users of WiFi hotspots as we travel around with our laptops, hopefully keeping cognizant of security. But if this was setting a precedent, obviously, such that people were using the hotspots to download illegal content, and the hotspot providers were then coming under legal attack as a consequence of the actions of the people that they were offering the service to, well, that potentially really throws a monkey wrench in this whole notion of go to Starbucks and get free WiFi, or wherever.

Leo: Yeah, no kidding.

Steve: So it'll be interesting to see how this gets handled. And then it does create, though, another problem, of course. If it's clearly made policy that there's a complete hold harmless and somebody offering such delivery system like open WiFi isn't liable, then where is the liability? I mean, then you would go to an open WiFi hotspot specifically to do your illegal things, and...

Leo: Well, but it's similar to the safe harbor provided to Internet service providers. I mean, if you're a pub owner you're...

Steve: So you're just extending that; right.

Leo: You're passing along Internet service. You're becoming an Internet service provider. And I think that in the U.S. you would be protected. I think it's interesting the British tried to make a law against it. I don't know if that would hold up here. That's very interesting. It's a terrible precedent.

Steve: Yeah. And again, it looks like this Digital Economy Bill, which is slated to pass, will fix the problem in the U.K., which will be a good thing because - and I hope these people don't pay their $13,000 fine until that happens, or they get some good counsel.

And the last little bit of news on the reminding people not to click on links in email, especially if they appear to be IRS refund letter notices...

Leo: You've got money.

Steve: Which I guess is attracting a lot of people. There's the so-called ZeuS Zbot trojan is now spreading very successfully, unfortunately, by so-called drive-by downloads. Email spam pretending to be an IRS refund letter is downloading the trojan if recipients click on the link in the email, without any additional user interaction. So they're leveraging some known vulnerabilities in whatever platform, presumably Outlook, that uses the IE viewer in order to download these things. So it's always worth reminding people, no matter how good you've been in the past, you must still be good in the future because these things, these problems are not going away. And the IRS, actually this is enough of a problem that the IRS has issued a formal statement on their own site saying we don't send email like that. So don't blame us. Don't click on the links. Just delete these things.

Leo: Yeah. Wow.

Steve: I just wanted to mention in the Windows versus Mac never-ending...

Leo: War.

Steve: ...debate, war, that I've been using a MacBook Pro for about the last month for many hours in the morning because I've been doing a bunch of PDP-8 programming.

Leo: Really. On a MacBook Pro?

Steve: Well, because - yes, because that's where this really nice PDP-8 emulator is available. And so...

Leo: Ah. Not on Windows.

Steve: There are some, but not nearly as nice.

Leo: Interesting, huh.

Steve: And so as a consequence of this strange sort of drag-me-along-kicking-and-screaming out of Windows over to the Mac, I've been using the Mac a lot, really for the first time in my life. And I just wanted to acknowledge that it's pleasant. I mean, there are things about it...

Leo: What did you think, it was like oh, my god, it's going to be a nightmare?

Steve: No, well, I just thought it was kind of different and kind of Romper Room-y.

Leo: It's not that different. That's really the real message I think in this war is they're getting more alike all the time.

Steve: Yeah, except there are, for example, the fact that you can't grab any window edge, that you can only resize from the lower right corner. It's like, okay, why?

Leo: Right.

Steve: But I like the way their, what, user preferences is sort of like the equivalent of the Windows Control Panel. I just like the way that's integrated. The fact that the apps have, like, they all share the menu bar at the top rather than each app having its own menu bar, which is a big...

Leo: You like that. Because that's a big complaint people have.

Steve: As a Windows user, it's a little unnerving because I'm not used to that changing. Once you get used to the idea that the window with focus changes that, then it's like, oh, okay, I get it. I don't quite understand - I've always been wanting to ask you, Leo. When I close something, like I'll close Firefox. Well, so the Firefox logo over in the whatever that thing is you call, like - oh, the dock...

Leo: The dock, yeah...

Steve: It's got a white dot next to it.

Leo: It's still open.

Steve: Yeah. Which makes me feel like, okay, it's not really gone.

Leo: See, this bothers me about Windows. Windows does something to me that seems a little odd, which is you don't explicitly close an app, it closes itself after the last window is closed. And on the Macintosh there's a quit command for every - and by the way, one of the things that's great about the Macintosh is the uniformity of menus. There's really a kind of a standard for how menus have worked. And you can always say that on the menu there's the application's name. At the very bottom of that menu will always be a quit command. And it's always Control, or rather Command, Q. So that's how you have to quit it. Closing the last window does not quit an application as it does in Windows. And that's confusing for Windows users, I understand.

Steve: And, okay. So when I - yeah. I mean, I'm confused.

Leo: But it makes sense to me because the window is just - is not the application. The window is just a display window.

Steve: So, okay. So if other apps, if I were to open other applications that needed the RAM, is the one that...

Leo: No, it's still running.

Steve: No kidding. Okay. Well, I mean [laughing].

Leo: It always seemed odd to me that you could - I never felt like the application was closed out in Windows because I'd close the last window, and now it is, I guess it's closed out. But I never explicitly said I'm done with this application. The Mac makes you say explicitly, no, I'm done with the application. Just because I closed the last window doesn't mean I'm not done.

Steve: Well, now, in Windows, unless you're talking about...

Leo: There is an explicit close command, I understand, which is unaccountably Alt F4.

Steve: Oh, Alt F4, I'm there. I'm there, baby.

Leo: I think Command Q's a little more mnemonic. But okay, Alt F4. And you probably use Alt F4. But Windows also will close an application when you close its last window. And that's the behavior the Mac is not doing.

Steve: Okay. But the way you're saying that, applications are windows in Windows. That is...

Leo: No, no, no. You can have an application, you can have a TSR-style application that's running without a window.

Steve: Well, you can have...

Leo: Sure, an index, things do that all the time. There is indexing going on in the background. There's always something going on there.

Steve: Well, services, for example, are applications that don't have windows. And you can have an application present multiple top-level windows. But normally a top-level window is an instance of an application. So for example, if I launch Internet Explorer, and I look in the Task Manager, and I see IExplorer.exe, if I launch, if I start another copy of Internet Explorer, I get another instance of IExplorer.exe. That is, it's actually multiple copies of the application are running. So when I'm closing windows, I'm closing applications on a one-for-one basis in Windows. Instead of thinking of in terms of, like, closing the last window closing the application. Each one is actually an application.

Leo: Is an instance, yeah.

Steve: Is an instance of an application.

Leo: It's not that way on the Mac.

Steve: I get that now.

Leo: An application is instantiated when you run it and remains instantiated until you explicitly close it. The window is a view into the application, but it's not required for an application to be instantiated.

Steve: Right. Interesting. Now I get it.

Leo: I didn't realize, to be honest, I did not realize that Windows was so closely tied, the instantiation was so closely tied to a view.

Steve: Right, I think you...

Leo: So all the views on Windows, the applications always have a view.

Steve: Yes. You can make an application have multiple windows. But normally the top-level window is an application. And if you have multiples of those open, you've got actually multiple copies of applications.

Leo: Now, all of this gets a little moot with threading. And that was the thing that Chrome did that was interesting was that each tab was an instantiation of the application.

Steve: So they could die separately.

Leo: Right.

Steve: Right, or hang and get killed and so forth.

Leo: Now, remember, you're a sophisticated user. And so an operating system is designed not for you. You understand what's going on under the hood. To the person who don't know what's going on under the hood, all they know is...

Steve: They don't even ask these questions.

Leo: Well, they do because you go, well, that is one - that's one very subtle problem that switchers have. The more obvious problems is the buttons, the windows zooming, you know, growing a window and moving a window. But that's a subtle one that kind of bugs people, I think, is that, well, wait a minute, on Windows I didn't close that application, I just closed a window. And now the application is gone? And I don't feel like, I'm like, are you sure it's gone? I don't know. There's no sign that it's there or not.

Steve: And I knew that it was around still because it would come back if I clicked on the icon, bing. It's like, oh. But I was thinking maybe it was like Apple was being clever and, like, leaving it in RAM; but preferentially, it's like other stuff came along and needed its memory since I had closed it technically. And it was like just sort of being around quickly if I needed it.

Leo: Well, I mean, some stuff's probably locked, but most of it's purgeable. So, I mean, the memory manager is pretty sophisticated in OS X. Remember that the heritage of the Macintosh, which just might go back to pre-OS X days, the memory management was godawful. And you really had a hard time reclaiming memory, and there was all sorts of issues. But they have a pretty good - they have a system very comparable to Windows now.

Steve: And when I think about it, like on the Finder icon, if you right-click on it, it says open another Finder window. Now, the Explorer, the Windows Explorer is that way. That is, there's one copy of Explorer running. And then you could have, like, multiple views of your file system and your computer and so forth. But that's sort of an exception. Most of the other things in Windows, the so-called, the top-level window, like the browser window, multiple tabs will live within that one application. But as you launch multiple apps, multiple instances, those are actually separate applications, separate application instances that start up and run. So, yeah.

Leo: To me the Mac way, I mean, I don't want to belabor this, makes more sense because it just reminds you that there are applications running without windows. I have many applications running that don't have a - for instance, here's a little

FriendFeed notifier that's running in the background. It pops up a window when there's something new on FriendFeed. But it doesn't have a window when there's, you know, it pops up a window and lets it go. I mean, Windows has many applications like that, as well.

Steve: Sure.

Leo: So I just think it's - I like to explicitly close an application because then I know it's closed. But that, you know, this is minor. And my real point, and the thing I tell people who want to engage in a war, is this is like human and chimp DNA. There's 99.99 percent similarity between Macs and Windows.

Steve: Yeah. I like the power. Maybe it's just the comfort. Maybe it's the familiarity. To me Windows feels like more of a power user's experience. But I have to say, and I'm noticing, and I'm looking around, I'm seeing more and more Macs. I mean, like, Macs really seem to be coming on strong. And, I mean, and why wouldn't they? I mean, if you want to surf the web and do email and actually get work done and not be all obsessive about the OS itself, but just use a computer as a means of getting your work done rather than it being the end itself, then I could easily recommend the Mac.

Leo: It's great for programmers because it comes with a very powerful programming environment.

Steve: In fact, yes, mine just updated Xcode. There was a major...

Leo: Xcode is amazing, yeah.

Steve: Yup.

Leo: And then you get Ruby, you get Python, you get Perl all built in. You know, comes with that. So I think it's a lot of geeks who want a terminal window. You know, they want the bash, UNIX bash shell.

Steve: Well, and the idea that it's got actual real UNIX underneath it...

Leo: Exactly.

Steve: ...is a compelling feature for many people.

Leo: But to each his own. They're all running on Intel chips. It's all, you know, it's all more similar than different, I guess maybe.

Steve: Yeah. And but it's very pretty. I like it.

Leo: Yeah.

Steve: So I just wanted to say I've been using it for quite a while. It took a while to get used to it. You've got to customize it a little bit here and there. I fixed the caps-lock so it's a control and overrode that because why is there a caps-lock, that's so...

Leo: But by the way, it's easy to do in Windows. You have to install...

Steve: You mean...

Leo: I mean in the Mac. You have to install a program on Windows to do that.

Steve: Yes.

Leo: Oh, I hate that caps-lock key. Hate it.

Steve: I had an interesting little quick fun Thanksgiving story that I wanted to share. Literally on Thursday of Thanksgiving an email was written by someone named Mark Schoonover. He said, "Happy Thanksgiving. As luck would have it, I'm out with the in-laws, and they asked me if this clicking they are hearing with their drive is anything to worry about. Well, it is. I can't read everything from the drive. I do own a copy of SpinRite, but it's at home, 365 miles away. I was hoping there was a way for me to download a copy, but I don't know my transaction ID/serial number to my licensed copy of SpinRite. Is it possible to look that information up? Here is my billing information." And then he gave his name and his street address and so forth. And he says, "I know it's a holiday, but hopefully you're available."

And then I've noticed that in the little dialogue it said "Sent from my mobile device." So Sue, who runs GRC's operations side - taxes, bookkeeping and so forth - happened to check in on Thursday since she, like the rest of us, operates from home so it's easy for her to. She wrote back and said, "Hi, Mark. How about an email address that you may have had when purchasing? Nothing is coming up under the address you wrote from." And he wrote back, "Well, it's possible I used...." And then he gave a different address. "That email address was before I used this personal email address. Thanks for the quick response."

And then Sue wrote back, "Mark, your original receipt with download instructions has just been sent via email. For your information, your transaction code in your email receipt is basically the keys to your account. It will allow you to obtain replacement copies as well as edit your contact email should it need ever be changed. Sincerely, GRC Sales Department." And then his final reply was, "Thanks again for handling this during a major holiday. Got to love a company with great customer support. The great news is, SpinRite saved the day. It managed to repair the problem in a single Level 2 pass. I was able to get the system to restart, get the data off, then remove the drive from service. I'm now in good graces with my mother-in-law. Have a great Christmas."

So, yup, we have that system. As long as you know your transaction ID, you can get it from our server anytime, wherever you happen to be. And if you don't, we can typically find it for you and send it to you. And then you can get it again.

Leo: So once again, SpinRite saves Thanksgiving. That's a nice story. And, yeah, I think there are probably more than a few people who went home for the holidays who were called to duty. How was your Thanksgiving?

Steve: It was great. I just had a quiet meal with some friends. And it was uneventful, which is, you know, kind of what you want. Just good food and great conversation.

Leo: That's the best kind, absolutely. Hey, let's get to our story of the week. We're going to talk about - I don't even know what it means, even after you described it. We're going to talk about same origin problems. All right, Steve. I've got my propeller hat, my virtual propeller hat on.

Steve: So we've never discussed this issue of what's called "same origin policy." Wikipedia defines it very succinctly. They say the same origin policy is an important security concept for a number of browser-side programming languages such as JavaScript. The policy permits scripts running on pages originating from the same site to access each other's methods and properties with no specific restrictions, but prevents access to most methods and properties across pages on different sites. And what that really means is different origins. This notion of an origin, the origin we're talking about is essentially the web domain - Amazon.com, eBay, PayPal, whatever. And the issue is the safe handling of content that's being served by sites.

And increasingly, I guess the term is "mashup," is like this notion of a given site that's now providing content from many different places at once. So this is potentially dangerous, if it weren't for strict enforcement of the same origin policy, meaning that your browser is - it receives a web page from the main site that you're visiting. And that web page then requests that other stuff be pulled in to finish out the page from other domains. So the browser goes out and retrieves other stuff from maybe many different domains.

Leo: Well, you see this all the time. I mean, even now almost every page pulls ads from another domain; right?

Steve: Right, right.

Leo: Is that an example of it?

Steve: That's a perfect example. And what's critical is that the content, whatever it is, if it's a GIF, if it's a JPG, or a Shockwave Flash ad, I mean, we're seeing Shockwave Flash ads all the time. And they can be very powerful. They can be running - there is, you know, ActionScript is the scripting language in Shockwave Flash. And there's a way for ActionScript to invoke JavaScript. So you know how I feel about all this scripting going on.

But it's crucial that the various components that are coming from different origins not be able to touch each other, that is, that there be individual isolation. Otherwise there's possibility for something malicious in one of these things sourced from one origin. There would be the possibility for it reaching into and modifying content in a different origin. So this is generally accepted practice. Well, what this one security firm, Foreground Security, a couple weeks ago made a blog posting where they explained that Shockwave Flash's excess permissivity...

Leo: [Laughing]

Steve: ...almost promiscuity, but not quite, the excess permissivity could be abused to essentially render any user-uploaded content dangerous. So SC Magazine, the security magazine, ran the story: "Researcher finds 'frighteningly bad' Adobe Flash flaw." The Register said: "Adobe Flash attack vector exploits insecure web design." And then the subtitle was "User-supplied malware upload peril." And even Computerworld said: "Flash flaw puts most sites, users at risk, say researchers." And the Computerworld story went on to say: "Hackers can exploit a flaw in Adobe's Flash to compromise nearly every website that allows users to upload content, including Google's Gmail, then launch silent attacks on visitors to those sites.... Adobe did not dispute the researchers' claims, but said that web designers and administrators have a responsibility to craft their applications and sites to prevent such attacks."

And the CTO, the chief tech officer of Foreground Security was quoted in that story saying, "The magnitude of this is huge." And these guys are based in Orlando, Florida. He said, "Any site that allows user-uploadable content is vulnerable, and most are not configured to prevent this."

And then: "He used the example of a company that lets users upload content to a message forum to explain the process." He said, "'If the user forum lets people upload an image for their avatar, someone could upload a malicious Flash file that looks like an avatar image. Anyone who then views that avatar would be vulnerable to attack."

And in their rebuttal, and this is like some back-and-forth going on, Adobe cited Microsoft's 10 Immutable Laws of Security, which Microsoft published back in 2000. Law No. 4 pertains to this, and I love how succinct it is. This is from Microsoft saying this. "If you allow a bad guy to upload programs to your website, it's not your website anymore."

Leo: Good point.

Steve: So, okay. So what all this boils down to is, and it raises an important point that is really crucial, which is why I wanted to talk about it this week, and that is most sites which do display any content which was uploaded by users are not being safe, and serving that from a different origin domain. That is, most applications, web form applications, online email, all of these things that we refer to as Web 2.0 stuff, which has really jazzed up the 'Net and is creating all of this interactivity, anytime a web server is, as part of its business, is accepting stuff from users, it is truly crucial and really unappreciated that it must then serve that back from an entirely different domain. And most don't.

Leo: Oh, okay. That's interesting.

Steve: I mean, and it's critical. Most don't.

Leo: So the safeguard is in there because the domain is different. Right? It's kind of a name space issue; right?

Steve: It is a name, yes, it is exactly a name space issue. And what this researcher was pointing out, on one hand, is something that Adobe says, well, we all know that. We've always known that. And it's like, but yes, but no one is taking - no one is paying attention to this. And specifically, where Shockwave Flash is a little flaky, having looked at this, in my opinion, is that it is - where it's permissive is that you can get Shockwave Flash the plug-in to parse a file that doesn't have a .swf, a Shockwave Flash file extension. You can stick a malicious Shockwave Flash content on the front of an image that ends with JPG or GIF. And if you embed this in a web page, Shockwave Flash will run it, even though it doesn't have the right file extension.

Shockwave looks at the content, the beginning of the content. And if it starts off as a valid Shockwave Flash file, it executes it. And so that allows people to bypass upload restrictions. For example, a site might say, oh, well, we're only going to allow uploads of JPG images. Well, you can upload a Shockwave Flash object ending with a JPG extension. And if the browser displays it, the Shockwave Flash plug-in can be invoked and run script where the webmaster never expected script to be run. And so Adobe has said that they cannot make Shockwave Flash behave better. That is, I mean, you could argue this is really broken, that it is this permissive.

The other thing that it ignores is the so-called "content-type" header. A content-type header is, it tells the browser what type of content the object is. So, for example, it'll say content-type:text/html, which tells the browser this is a text content of html. And browsers are smart. And if it the file doesn't end in html, the browser won't render it because the content it received and the content type don't match. Similarly, you might have a zip file with the content-type application/zip, or application/gzip and so forth.

Anyway, again, Shockwave Flash ignores that. And you could argue, wait a minute, I mean, it really should not ignore that. So I would argue that Adobe has some responsibility in the degree to which Shockwave Flash can be abused. They're now saying, because they've sort of been pushed up against the wall, well, we can't change it because it would break everything.

Leo: Can I get a clarification? You've been saying Shockwave Flash. Do you mean Shockwave and Flash? Or are you just using Shockwave Flash as a longer term for Flash?

Steve: I'm meaning Flash. You're right.

Leo: Flash, okay.

Steve: Yes, yes.

Leo: I mean, I think it's technically Shockwave Flash. But you mean Flash.

Steve: I just, yes, Flash content.

Leo: Yeah, okay.

Steve: Yeah. So one of the problems is that Flash is - and this was the focus of this posting that got everybody stirred up is that Flash is too permissive in allowing itself to run Flash files, even when they have a different extension, and even when they've got the wrong content header. It's that which allows users, bad guys, to upload malicious content which will pass the upload filters and then could be served by the server. So that's a problem.

But the bigger problem - but this is sort of a subset or an exploit of a bigger problem, which is that if all user-provided content were being served from a completely different domain, and that's the way it should be, then even this wouldn't be a problem. So Adobe is saying, wait a minute, this is really not our problem. This is a problem that people have ignored the same origin issue. They are serving user content, user-provided content, from the same origin as other stuff, the website's content. And inherently that allows the user-provided content to interact with scripting that's provided by the site that the user is visiting. And that's really dangerous.

So, I mean, I agree with everybody, the real lesson here is - and I'm hoping that webmasters are hearing this. Because the problem is, it's not easy to fix. I mean, we're talking about having to - talking about having to host all of the stuff that could come from the outside, blog postings, pictures, avatars, content, I mean, there's all of these sites now that are allowing users to upload things. I mean, even Gmail, for example.

Leo: And it's just kind of the way the web works. I mean, you wouldn't want to turn that off.

Steve: Well, no. And you can't. I mean...

Leo: Right.

Steve: You can't turn it off. And the problem is getting to safety from here because, if it had always been done correctly, then we wouldn't have a problem. But it hasn't mostly been done correctly. There are some sites - and, you know, I scanned past them, and I meant to write them down so I could recite them. But there are sites like professional sites, like Hotmail and Yahoo!, for example, I don't know that those are two, and I tried to track them down again, and I couldn't find the references. Because in this research I ran across major sites that are aware of this, and they definitely serve anything coming from - anything that ever originated from other than their webmasters come from a completely different domain. So they did it right. The problem is, that requires every webmaster who's going to be hosting content with any of these state-of-the-art applications to have two domain names.

Leo: Right.

Steve: And all of the overhead that goes along with it. I mean, it's a big deal. That's not a simple thing to ask someone to do. It's certainly not simple to ask them to change it now.

Leo: Well, you know, it's interesting. You for years have served your images from one server and your text from another; haven't you? Have you kind of done this?

Steve: Yes. For a number of different reasons I've done that. And I do have a number of different domains.

Leo: Is that potentially the same problem? I mean, I know it's your content, so obviously it's not.

Steve: Right.

Leo: But it is mixed content.

Steve: Exactly, yes. And so, for example - and traditionally, for example, I have had images.grc.com and then GRC.com. Now, I've folded that back into GRC.com/images. And so there were reasons I was doing it once upon a time. Literally I was serving images from a physically different server, just for bandwidth reasons. I realized that, wait, this is crazy for me to be serving images over my T1s, back when GRC used to be behind a pair of T1s. And so Mark Thompson, for example, was for a while hosting images.grc.com just so I could use bandwidth where it made much more sense. So there were a number of things I was doing.

But so there I was, splitting my content between multiple domains. However, in that case it was a subdomain. And apparently there are scripting ways to get around subdomain variations, so that that's not enough. You can't just say user-suppliedcontent.mydomain.com versus mydomain.com. You've got to have something like user-suppliedmydomain.com, that is, a fully different top-level domain, and have the content coming from there. Otherwise you can get scripts to agree on subdomain sharing having a - being treated like the same origin. And that's not safe.

And so this is big news for many webmasters who never appreciated the danger. And in fact the guys that made a big deal about this were reticent to do so because they were able to demonstrate exploits of Flash on many popular sites. And, I mean, they were saying, well, what do we do? We really want to force Adobe to fix this. And their focus really was on Flash. Adobe has said we're not fixing nothin' because we can't change Flash or it'll make it more restrictive, because its permissivity is one of the - that is to say, it's lax security operation, which is really what it is. It should have been much more standards based, in which case webmasters would have had to make sure that content headers were correct. And many of these avenues for exploit of this same origin problem would have been cut off. But those are wide open right now.

So, I mean, by my talking about it and shining additional light on it, I mean, again, this is a problem which is potentially big. But it's, you know, we know the problems don't go away if you ignore them and if you hide them. And maybe somehow Adobe could come up with a compromise where they do a version of Flash which can be asked to be more secure, and then webmasters can incrementally - webmasters who care and want to protect themselves could incrementally require that more secure Flash plug-in, which would then propagate over time, and they could protect their site using a Flash player that was able to be more restrictive, able to be asked to be more restrictive. I mean, I don't know what the solution is. But this is a problem. And I recognize that it's potentially big. And the problem is, it's not easy to fix. It's...

Leo: Well, let me put you on hold. I mean, you must have some prescription, or something you'd like to see happen.

Steve: Okay.

Leo: I hope you do. I have some thoughts about it, too.

Steve: Okay.

Leo: We're clearly not going to ever get rid of this multi-homed web page problem. So clearly this is a problem. And as I said at the beginning, it's a name space problem; isn't it? It's the problem is that web browsers and JavaScript and other web languages kind of don't isolate their code very well.

Steve: Right. Okay. So to put it simply, and then we'll talk about solutions, there's a clear problem which is created when a page is composed of objects which could be smart, could be scripted, coming from different places. Because traditionally, if scripting came from the same place, that is, different instances of scripting or different parts of scripting, they're inherently able to have visibility into each other. They're able to...

Leo: Yeah. That's what I was getting at, yeah.

Steve: Yeah. So because of the recognition of the problem, the wizards of the web, always understanding that this was a potential problem, created this notion of same origin, specifying that only content that was served by the same origin, the same web domain, could interact with each other. Otherwise, stuff coming from somewhere else, from a different domain, would inherently be isolated.

So, I mean, that was an understood but dramatically underappreciated issue. And the problem is that there's no enforcement. Nobody makes content from users come from somewhere else. And so the de facto, the default, is oh, well, I have mydomain.com, and I'm going to - and I have an SQL database. And when stuff gets submitted to my site, it goes into my SQL database, which is a backend on my server, and then we want to - we're going to have a forum, and so we're going to want to show people what other people have posted.

Leo: Right.

Steve: And so it's, I mean, it's sort of the natural thing to do.

Leo: And the presumption is, and this is the mistake, right, if it comes from me it's safe. If it comes from my domain it's safe, even though I know it doesn't come from me.

Steve: Correct. And so then we say - so the webmaster says, oh, but I don't want people who might be bad people to submit scripts because then the scripts would go into my database. And then when people viewed a page containing scripts that had been submitted by maybe a bad guy, those scripts coming from my database, through my domain, my server, would have equal access to everything else on the site. And that's really bad.

Leo: Right.

Steve: So the webmaster who's trying to do the right thing says therefore I'm only going to allow - I'm a photo sharing site, and I'm only going to allow JPG uploads, or GIF uploads. The problem is that due to the funky way that Flash works, it doesn't care if its script is named .JPG or .GIF. It runs anyway. And so this sort of creates a backdoor that allows scripting to run even though it's identified as an image. And that's really bad.

Leo: Yeah.

Steve: So that's a problem. But the bigger problem, the more generic problem is that, as Microsoft's Law No. 4 of Internet security says, if you allow a bad guy to upload programs to your website, it's not your website anymore. Which is their way of saying you really need containment. You really should be sure that inbound content from users goes to a database which is on a different domain. And, I mean, it's a pain in the butt to do that because you have to now have main domain and other domain and make all the links work and make everything work. And then...

Leo: But would that solve it?

Steve: That really does solve it, yes, because then all of that content comes from a different origin. And the browser will then enforce this different origin and enforce containment. And then malicious scripting that could have snuck in through some other means won't have access to the main domain's scripting. I mean, you still have the problem that it's scripting that's running on the user's browser. But it's toned down at least some.

Leo: It's a little ironic because even Google doesn't do this. And certainly they would have the technology to. Is it difficult to do? I mean...

Steve: Yeah. Well, it's not difficult if you did it from day one.

Leo: Yeah. Retroactively it's hard to do.

Steve: Oh, retroactively it would just be a huge nightmare. I mean, all the links in all your pages that refer to content and, I mean, yes, everything would have to get changed around. And then you have other sort of little side effects, like browsers might be saying, wait a minute, we don't want cookies. I mean, suddenly now that provided content is a third party. So now you have cookie access problems because you're not - because one of the things that you're wanting to restrict is you're wanting to keep bad scripting from having access to the user's session cookies because we now know that session cookies are used for creating persistent log-ons.

And so you definitely want the bad scripts, you want to keep them from having access to that. So you want them to be in a separate domain. So that's an example of what maliciously can be done if you're in the same origin. If you're in the same origin, you've got access to the user's cookies. And all kinds of mischief can be made that way. So it would be nice if, in the case of Flash, it were much less permissive.

Leo: Right.

Steve: I can understand them saying we can't change it now. Unfortunately it's too late. It would break too many things. Maybe some pressure can be brought on them to allow this to get fixed incrementally for webmasters who are willing to tweak their sites as necessary to run the Flash in a more tighter, less permissive way. But the thing I mostly wanted to shine a light on is this whole notion of same origin. Because we've never talked about it. I mean, which also sort of demonstrates how obscure it is, the fact that we've gone to Episode 225, and this is a big deal that hasn't really come on the radar until now.

Leo: Well, it came up because of Flash, but it's more than just Flash. It's been a problem in JavaScript. In fact, I'm sure John Graham-Cumming refers to this, you know, this name space issue. JavaScript can be confused - I think we even talked about it - or tricked by using common variable names.

Steve: Yes. It is one of the things - and this is where we were talking about it was with John a few weeks ago. And that is the way Adobe defended themselves.

Leo: It's not just us. You're all screwed up.

Steve: Well, and saying that this is a well-known problem, and websites shouldn't be doing this. And in fact Howard Schultz, who's an editor for SANS, in the SANS newsletter they talked about this, and Howard was the chief security guy at eBay for many years. He was quoted in an editorial that SANS ran saying who is this Brad Arkin - Brad Arkin was the spokesman at Adobe. He said, "Who is this Brad Arkin kidding? Saying that 'sites should not allow user uploads to a trusted domain' is completely unrealistic." And so that's Adobe saying, yeah, sites shouldn't do that. It's like, well, okay, but they all do. And so basically Adobe is saying, it's not our fault.

Leo: Even YouTube does it. I mean, everybody does it.

Steve: Yup.

Leo: So you're saying if YouTube had a separate server that was FlashYouTube.com, and all the Flash came from there, and all the other content came from YouTube.com, that would - it wouldn't solve the problem, but it would mitigate it a little bit.

Steve: It's not even clear that using a subdomain solves it.

Leo: No, no, yeah. It would have to be completely, like, FlashYouTube.

Steve: FlashTube.com would be where the actual Flash objects lived, within the YouTube domain. So people would go to YouTube. But the things they looked at would be coming from FlashTube.com. And there the completely different domain name would, in the browser, would keep them from ever having any interaction. And today we don't really have that. I mean, it's all possible. The really high-end professional sites understand this, and they do it. But most sites don't. And I'll bet you we'll be doing a podcast here before long, talking about some bad consequence of this.

Leo: Right. And is there anything that the end user can do? I mean, can you - are there any settings you can set in your browser that make it - I guess not because it can't tell. It all comes from the same place.

Steve: There were, among all of this skirmish, people saying, well...

Leo: The browser should solve it.

Steve: You could turn off Flash. It's like, oh, well, thanks a lot. Yeah, I mean, talk about dead websites. And in fact there's a toggle Flash add-on for IE, and NoScript offers Flash suppression. Or, like, Flash on demand, where it just shows you sort of an F, and then you click on it if you want to run it. So that's something I've seen NoScript doing. But again, we've become so dependent upon Flash for so much. And again, it happens to be one too readily exploitable vector of this problem. But it's not really the problem. The real problem is that, as Brad Arkin at Adobe unrealistically suggests, sites should not allow user uploads to a trusted domain. Well, it's like, well, okay, good luck with that. Sites are going to.

Leo: You know, somebody's pointing out in the chatroom, and I was just playing with YouTube a little bit, YouTube does do some of that. They have a server called YTimg.com, where thumbnails come from, anyway. I don't know if the Flash comes from there or not. That's very clever. So maybe they are doing something like that.

Steve: Really, I mean, I wouldn't be at all surprised if sites that are really on their game, I mean, where there's a lot at stake, they've said, okay, well, their webmasters really understand. And here's the problem is that so many sites these days are canned. You buy some blogging app or some foreign app, and you just stick it in. And you run it on yourdomain.com, and there's no concept of this whole same origin policy issue. It just isn't there. Someday this problem will get solved, like so many other ones we talk about.

Leo: And blocking Flash alone, although I do that, there's a Flashblock script, and there's a Safari plug-in which I use called ClickToFlash. So I do that. But I do it more because I don't want Flash animations to start playing while we're doing the show when I go to a site.

Steve: Yeah.

Leo: But there are ways to do that. But that's not enough because it's just - that's just Flash, and now the same problem occurs with JavaScript and other...

Steve: Well, and I use a very nice add-on for Firefox, Adblock Plus.

Leo: Right. That's very extensive, yeah.

Steve: Oh, my goodness, it's so nice. You just don't have fish jumping around your page when you go somewhere. Or just really annoying things flashing at you and trying to get your attention. It just quiets all that down. So that also would protect you in a nice way.

Leo: Right. Steve, I'm glad we did talk about this. It's funny that this has been around forever, but this is the first time we've talked about it. Same origin problems. Steve Gibson is at GRC.com. Now, that's his website. You can go there to find this podcast, of course, both 16 and 64KB versions, depending on your bandwidth capabilities. There's also transcription so you can read along as you listen, and show notes, too. It's all at GRC.com.

We have a feedback episode next week. So if you want to leave a question about this or any other security topic, go to GRC.com/feedback. And while you're there just pick up a copy of SpinRite. You never know when you'll need some hard drive recovery or maintenance. GRC.com. Steve, I can't wait to talk to you next week.

Steve: Thanks, Leo. We'll do it in seven days.