ClickJacking

Description: Steve and Leo discuss yet another challenge to surfing safely in the web world: Known as "ClickJacking," or more formally as "UI Redressing," this class of newly popular threats tricks web users into performing web-based actions they don't intend by leading them to believe they are doing something else entirely.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-168.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-168-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 168 for October 30, 2008: Clickjacking.

It's time for Security Now!, the show where we cover everything you'd ever want to know about security. And we do it with a guy who is the king, really, as far as I'm concerned, the man who discovered spyware, named spyware, wrote some of the most used security utilities out there, including ShieldsUP!, Mr. Steve Gibson of GRC.com. Hi, Steve.

Steve Gibson: Yes, in fact sometimes we're discussing things that you'd rather wish weren't the case. I mean...

Leo: Well, lately it's been kind of bleak because it's like, there's bad stuff, and there doesn't really seem like there's any cure.

Steve: Yes, that's true. I mean, because what we're beginning to see now is that the technologies we rely on fundamentally are really being exploited. I mean, what we're going to talk about today, clickjacking, is very much like that. The good news is there are things now that people can do. Firefox users are essentially completely protected by recent updates to NoScript. But we'll talk about all that, and also about some, well, there's lots of news in the last week since...

Leo: Lots of security.

Steve: Yeah, since 167, so...

Leo: So let's get an update on what's been going on in the world of security, Steve Gibson.

Steve: Well, it's funny. Microsoft put out an out-of-cycle patch.

Leo: Last week.

Steve: Yeah. And I picked up a little blurb in GRC's newsgroups. Some people were saying, do you think Microsoft deliberately does this after Steve and Leo record the podcast, you know, like the moment after it's been recorded and before it actually goes public? Because this dropped out at exactly that time. It is substantially different than, or more of a problem in some ways than [audio dropout] referring to it during the Tech Guy stuff over the weekend when you were talking about it. Because this is not a user goes to a bad website with their browser sort of problem.

Leo: Oh, I misunderstood it, then.

Steve: Yeah, it is like one of the old-style, packet arrives at your computer...

Leo: It's a worm, okay.

Steve: Yes.

Leo: Oh, I didn't know that, okay.

Steve: Yes, yes. It runs - it's an exploit in the RPC service that runs in all versions of Windows. It affects Windows 2000, XP, and Server 2003. The architecture of Vista, the updated security architecture mitigates the problem. But it's...

Leo: Mitigates or eliminates?

Steve: Mitigates. It's reduced from critical to important on those platforms.

Leo: So why? Is that because UAC pops up and says something's trying to access your system, or...

Steve: It's the nature of the way they're running the service. The service that is exploitable, the remote procedure call, RPC service, it's sort of sandboxed so that it's not able to do as much damage as it can under 2000 and XP. But, okay. So this is a - basically it's the service which is always lit, is always running and open and listening for things coming in on ports 139 and 445. That's the traditional Windows filesharing ports. You know, file and printer sharing and other things. So many things have happened to mitigate this being a problem, which is why, you know, I didn't immediately find, track you down and say, Leo, we've got to immediately alert our users to it. Microsoft had to do the responsible thing, and call this critical, and push an out-of-cycle patch.

Leo: And that's because there were websites already taking advantage of this; right? There were sites out there that...

Steve: No, it's not websites. But the way they discovered...

Leo: Oh, there were worms floating around.

Steve: Yes. The way they discovered it was that - and I'm not sure who it was. It wasn't Microsoft. It was somebody else who saw some suspicious traffic on the 'Net, like packets arriving to 139 or 445. And they looked at them thinking, hmm, what is this packet trying to do? This is trying to do something strange. Well, turns out it was attempting to exploit this vulnerability that was not known. So this was a zero-day exploit, meaning that it was discovered in the wild, I mean, for the first time. And so...

Leo: And I saw that a security service within two hours of Microsoft's release was able to reverse engineer it.

Steve: Yes.

Leo: Immunize or something like that?

Steve: Yes. There was immediately a proof of concept. And now there are active worms. And the exploits have been moved into some of the security toolkits that exist. So this thing exists now. Now, setting back - so, okay. Once upon a time, when we had Windows 2000 that did not have a built-in firewall, or we had Windows XP before Service Pack 2, which did have a firewall, but it was not on by default, that was the era where we had the Nimda and the Code Red and Blaster worms.

Leo: Right, and Zotob, too. So is this like those?

Steve: Yes. This is as bad as those. Except that the world has changed so much since then because Service Pack 2, of course, turned the firewall on by default. And so many people are behind routers that block incoming traffic by default. And one of the other things that Microsoft has done is that they've limited, by default, file and printer sharing to your local network. So those services are not being offered out on the whole Internet

itself. If you look in the details of file and printer sharing and the server service, inherently they said, wait a minute, it makes no sense, unless someone's nuts and wants to make their C drive available on the whole Internet, it only makes sense to have that available on the local network. So there is local network topology.

However, the concern is that, if something got into an Intranet that is behind either a home or small office firewall or in a corporate network, then potentially it could - this would allow things to spread between machines within a network. So that was really where the concern was, is essentially, you know, you're behind the router, you're behind the firewalls, you're in a local environment where you might have trust among machines. This potentially abuses that trust because there you would have a scenario where you might, for example, have available file and printer sharing among machines, and this can exploit - essentially it's a classic buffer overflow on the stack in this remote procedure call service running on all Windows machines by default. So it was bad, but it didn't just go wild on the outside because Microsoft has, since the era of those worms, they've done many things like limiting the scope of the network where file and printer sharing was available to the local network. And having a firewall turned on and having those ports blocked and so forth.

Leo: Well, it's really an example of the success, in fact, of Microsoft's improved security policies that what would have been a big deal was nothing.

Steve: Right.

Leo: Yet they still put out the patch. I guess that's because there are still people who are not running firewalls.

Steve: Well, that and there is the scenario of local exploitation.

Leo: Oh, internal, inside the network.

Steve: Right. And so I think that was what really motivated them to do it now rather than wait three weeks for the next second Tuesday of November, when they could have certainly fixed it. It also demonstrates...

Leo: That actually makes sense.

Steve: It also demonstrates how quickly they can act if they really want to. I've criticized them often for, like, having bad problems that they have known about for six months before they do something. And it's like, okay, clearly when they want to do something, they can fix something really fast.

Leo: Yeah. I think I got another one this week. But maybe that was just some of my machines didn't get updated earlier. There wasn't another one this week, was there?

Steve: No.

Leo: Okay. That's probably what the other machine is doing. You know, it really is, in a way, a testament to you, too, because you're the one who's really been promoting this idea of using a firewall. The whole idea of ShieldsUP! is, you know, protect yourself. And so I think that drum beat's really been heard.

Steve: Well, and in fact ShieldsUP!, the original concept of ShieldsUP! was to tell people to absolutely close down those ports. I mean, and the other thing, too, for example, in terms of mitigation, I have a Cox cable modem. And Cox blocks those ports because of...

Leo: It's RPC? So is that 138, 139?

Steve: Yeah, it is the same file and printer sharing ports. Several services all live on those ports. So it's 139 - Microsoft said 139 and 445, which to me says it's a TCP exploit. I don't know for sure that it can't be done over UDP. But 139 is the TCP version of the older file and printer sharing that worked on 137, 138, and 139, when 137 and 138 were UDP protocol, 139 was TCP. 445 uses both UDP and TCP, sort of in the new generation of Microsoft's use of these services. But it's specifically in response to the prevalence of those worms in the past that ISPs began blocking those ports. So there is another example of how things have moved along such that this wouldn't be a problem. Because, you know, even a Cox cable user with a wide-open Windows 2000 or old XP machine or an XP machine where they were deliberately trying to open those ports, Cox wouldn't have allowed that stuff to get in from the outside.

Leo: Well, it's good news. I mean, it sounds like - and it's true that we haven't seen any - I remember when Zotob came out, and CNN was brought to its knees. We haven't seen anything like that.

Steve: No. No.

Leo: Will these worms be endemic to the 'Net in the same way that Sasser and other worms just kind of live forever?

Steve: Yeah. We are seeing traffic for a new trojan which is looking for machines that have this vulnerability.

Leo: The sad thing is that those machines get bit, it's the Windows 95 machine that's running dusty in the closet somewhere doing some crazy little service.

Steve: Yes, and we've seen that before. Strangely enough, there are, like, old Windows 2000 servers...

Leo: Unpatched.

Steve: Exactly. No firewall, on the 'Net, just sort of abandoned. But, you know, they will end up getting found. They'll get exploited, and then they in turn will start searching for other machines in standard worm propagation behavior. And so now those machines that'll have this thing on them will just be spraying this traffic out on the 'Net, as I call it, "Internet background radiation," IBR. It's just stuff that'll live there now and we just put up with.

Opera, just after - I think it was last week that I told people about the update 9.61, which fixed three relatively worrisome vulnerabilities in Opera. Just, like, the day after they released the patch, another related one was found. We don't yet have the update for it. It'll be 9.62. And the Opera folks have said real soon now we'll release it. But I wanted to let people know. Probably by the time a week passes and this next podcast comes out, our next one, I imagine that Opera 9.62 will be out. They're just finishing the final stages of testing for this update. It's a cross-site scripting vulnerability that - there was a demo I saw where you could click a link, and it would pop up your Windows calculator, meaning that it was able to, unfortunately, run...

Leo: Nothing should be able to do that.

Steve: Run code on your machine.

Leo: Yeah, that's not good.

Steve: No.

Leo: That's a very dramatic proof of concept, when a program just starts itself.

Steve: And I've talked several times about my own experiences with credit cards getting away from me. And I received a robocall this morning from my main Visa card, asking me if the following charges had been made that morning, or this morning, literally while I was putting things together for this podcast. It's like, no, those are not me. So I called them back and confirmed that these were not my charges. The first one was a four-cent charge for computer consulting, then a $1 charge on eBay, and then a $400-and-some-odd, and they began going crazy with it. But again, this was Visa. And they're no longer our sponsor, but they did immediately lock the card down. They denied those charges. They went through a bunch with me. We walked back to dinner last night, which was the last time I had used the card. I said, yep, that one I recognize. Nothing since.

Leo: Do you think it got stolen at the restaurant?

Steve: Oh, no, no. This is - no.

Leo: You use it online?

Steve: I use the card, I mean - and again, we've talked about what to do in terms of safe use of card. I am going to again check my various cards to see if any of them offer a native, one-time use service. Last time I checked they did not. I'm hoping that someone will. If not, I'm going to make heavier use of PayPal's plug-in for that purpose, even though I don't really like it in my browser. I may try it in Firefox, now that I've moved from IE to Firefox. But I have been using it more. The problem is, it insists on withdrawing from my checking account. And I really don't like that. Also it confuses my bookkeeper because she would like to balance my checkbook, and there's all these random charges that don't have any checks against them. So I think what I may do is create - either transfer a block of money to PayPal and then have it pull from that, I think I can do that, in the same way that you've got PayPal credit in there, and it can pull from that. I think I may just move a block at a time and let it sit there and then - but it's amazing to me, the rate at which a heavily used credit card is now being compromised, even by someone who's very cautious about where I go and who I give it to. So...

Leo: Why do they do those two small transactions?

Steve: Those were to test whether to - so it didn't hit any limit on the card, to see whether - they've probably received this card in a batch of them that they had purchased from someone. And again, this is generally not - it's not like I used the card yesterday, and it got abused today. It's probably from some use months ago where there was some exploit on someone's server that lost a huge chunk of cards, and mine was among them. And it took a few weeks for it to drift over to wherever it went before this really began getting abused.

Leo: I know there are some Visas, some MasterCards, some American Express cards that allow you to generate a one-time number. It's just the, you know, I think it's the bank, the offer. Although American Express offers its own cards. So I think they offer it, as well, is a one-time number.

Steve: Yeah, it's time to get serious about that.

Leo: Yeah, I have that PayPal one, and I haven't used it all that often. And I probably should start using it more. You know, it's kind of a little bit of a pain because you've got to generate it and everything. But, boy, yeah. Although now with everybody so aware of it, you've got to wonder how often does that happen now. I mean, how often do they get away with it, I guess? You know, that $400 charge, do you think they got the merchandise?

Steve: I'm not sure. They're sending me a printout of, basically, of everything that might be suspect for me to go over it and make sure that it's nothing that I did. And then, for example, there was a $5.99 charge from Amazon that was probably some Kindle subscription of mine that just happened to cycle at that time. So it's like, okay, whoops, there is one of mine that was part of an automatic charging cycle. So I have to just go through, make sure. But of course I'm completely indemnified from charges that I say

I've looked at these, I'm sure they're not mine. Which is, you know, that's what you want. It's just a pain to lose the card. I had it memorized, of course.

Leo: You get to start all over, yeah.

Steve: And I did have a fun SpinRite story to share, with someone who deliberately was anonymous. He used the Security Now! /feedback form to send this to me. And he wasn't anonymous for any particular reason. But he said his location was somewhere in California. And he said - his subject was "SpinRite Makes Coworker a Believer." And he said, "I've been a listener of the Security Now! netcast since Episode 1 and have always enjoyed the SpinRite testimonials that you share on the show." Much like this one. "I'm a systems administrator at a laboratory and have had SpinRite help me both in my professional and personal life. Nothing quite as amazing as many of the stories that I've heard on your podcast. But I thought I would share this one as it shows the power of SpinRite to prove itself to those who doubt its true power.

"I had heard of SpinRite in the past, but never purchased it or used it until I started listening to your podcast. I told my coworker, a fellow sysadmin, about SpinRite, and he did not seem to share my desire to try the product. He seemed to think that SpinRite was just like other snake oil products on the market that look fancy, but do not really seem to fix much. I still decided that I would purchase a copy through work to try it out. The purchase process was a little bit longer for me since I had to go through our purchasing department, but that was the only thing that delayed me from receiving my copy. Once I had it downloaded, I burned the ISO and was eager to try it out on a failing hard drive." But he didn't have one.

He said, "I ran it on several systems around the office that were not having problems and showed my coworkers the various screens in SpinRite that showed what was going on. But since there was nothing wrong with these systems it really did not impress him. Then the day to really put SpinRite to the test finally came. One of the tablet PCs that was hooked up to an instrument in the lab started having trouble booting. And similar to other testimonials I have heard, the system was stuck in a boot loop where the system would start booting, blue screen, reboot, and blue screen again, and so on. My coworker had taken this trouble ticket and was not looking forward to reinstalling the Windows on this system, along with all the other software.

"I handed him our copy of SpinRite, and without much optimism he booted the system off the CD and started SpinRite. This time, right off, SpinRite switched into DynaStat mode and started chugging away, repairing the first few sectors of the drive. He seemed a little more interested when he saw this different behavior and watched as SpinRite worked on the drive. After a while we left SpinRite to its task and decided to come back in a few hours. I don't remember how long exactly, but later that day SpinRite was done, and it was time to try booting the system into Windows. My coworker rebooted the system, and Windows started right up without any problems.

"After this experience, not only did my coworker never mention his doubts about SpinRite, he began using SpinRite on every system he thought might have a hard drive problem, and we purchased three more copies so that we had a site license. To this day SpinRite is still one of our favorite tools, and we use it all the time. Thanks, Steve, for the great podcast and for your work on Security Now!; and thanks, Leo, for not only the Security Now! netcast, my favorite, but also for all the other great netcasts you do. I'm leaving my name out in case you do decide to read this on the podcast, but I'm putting it in my email just in case you want to replay." So thank you, Anonymous Listener. I

certainly appreciate the testimonial.

Leo: Always welcome. Well, let's talk about something that actually is pretty timely. It's pretty newsworthy. I started reading about it just a couple of weeks ago. I'm sure you started getting emails right about then, too. Clickjacking.

Steve: Yes. What's interesting about clickjacking is that it's a problem which is arguably serious, yet it's not a consequence of any bugs. So much of what we talk about on the show are, like, are the result of specific defects in software which the author of the software didn't see. Somebody came along and discovered a bug and said, oh, we can leverage, we can exploit this defect to our benefit.

Now, also there are - there's sort of another class of things, sort of like the cross-site request forgery that we talked about two weeks ago, or cross-site scripting that we've talked about where, for example, bad guys can take advantage of Web 2.0 functionality where a site will accept text from a user, and that text, if it were script, would then run on subsequent visitors' browsers when that text is again shown, like in a blog or an online forum or something, thus requiring that accepted text be sanitized for exploits. So there's an example of the power of the technology sort of being turned to a malicious purpose. So clickjacking is something similar. It's interesting, too, because it's been around for a few years. And it's...

Leo: Well, that's funny because we just started hearing about it. So you're saying it's not new.

Steve: Well, I would relate it a little bit to, like, the Sockstress that we talked about where sort of the concept, sort of a dim awareness of the potential for problems had existed, but no one really sat down until - to sit down and look at it closely. And so a couple of researchers, a security researcher, it's Robert Hansen and Jeremiah Grossman, they were getting ready to do a presentation that was going to be earlier this month, OWASP, the Open Web Application Security Project. And they said, okay, we're going to do it on this sort of an interesting class of exploits.

Well, as they explored this sort of dimly worrisome problem and, like, matured it for their - literally for their presentation, they realized much more could be done that was, like, more worrisome than they had thought. For example, one of the things, we've not yet really talked about Flash cookies. But Flash cookies are another type of cookie, user identity persistence in the same way that web browser cookies are, that are supported by, originally it was Macromedia and now Adobe Flash technology, where...

Leo: We talked a little bit about it in the context of banks using it.

Steve: Yes, in order to maintain a grip on you when...

Leo: Right, when you deliver cookies.

Steve: Exactly, you...

Leo: And in fact our Stickam chat, which, you know, if you watch our live video, it's on Flash, and the chat is in Flash. And those Flash cookies sometimes cause problems. So the Stickam people who use our Stickam chat are very familiar with it because one of the fixes for those problems is to clear your Flash cookies.

Steve: Right. So what happened was some of the things that these guys realized they could do - and I'll explain how in a second. But they could, for example, trick people into enabling their webcam and microphone through abusing some security problems with Flash. And Flash is configurable from using pages on Adobe's site is the way you configure the security settings, or by using URLs that the Flash object will see. But normally you have to push buttons to make this happen. Well, what these guys realized was that it's entirely possible to, due to the complexity of contemporary web browsers, to essentially have layers. It's possible to have something hiding below the surface, literally on, like, a layered page, where the user clicks on what they see, but what they're actually clicking on is content on the page behind. So you can use things like dynamic HTML and CSS. You do not need scripting.

So even disabling scripting doesn't protect you from this. And in fact disabling scripting can in some cases make the problem worse because there are - some of the ways that this is exploited is by using frames and causing other sites' content to appear in a frame, but then obscuring that. So, for example, we've talked about frames in several different contexts in the past. The idea of a frame on a web page is that your browser brings up a web page, and there is a region of the page called an iframe, an inline frame. And the URL for that can be some other site.

So, for example, you could cause Google Mail to come up in a frame. Well, now, because your browser has gone to Google Mail and said, hey, load this URL, please, Google Mail knows this as you. That is, it's you. It's just exactly as if you put the Gmail URL up in the browser's main URL field and loaded the page. In this case it's just contained in a frame. The problem is that this clickjacking exploit - it's also known as, a little more formally, the formal name is "UI Redressing." Some people who don't like "clickjacking" prefer calling this "UI Redressing," which is also...

Leo: "Clickjacking" is catchier.

Steve: I think so. I think "clickjacking" is good. I mean, as you say, it's a fun name. So the idea would be that it would be possible to hide this inline frame literally behind an opaque kind - behind an opaque window where a malicious site has put something else, for example, maybe a game where you try to click on successive buttons in a certain pattern, or anything, I mean, it doesn't even have to be something so extensive. And literally, when you think you're clicking on one thing, you're actually physically clicking buttons on the content that's been loaded in a frame underneath what you can see.

And I have, for people who are interested, I've got a ton of links in this week's show notes, in the show notes for this episode of Security Now!. There are some demos which are benign, which have now surfaced, that are on the 'Net. There are some takeaways. For example, I mean, some actions that people can take that we'll be talking about because this is something which has gotten a lot of news. It's like sort of the current state-of-the-art exploit. And it's become a new popular thing to, you know, basically script kiddies and people who have various sorts of intent are now - they have a new way, a new toy they can use for causing people problems. And as I said, just disabling

scripting doesn't help. Now...

Leo: So your favorite little NoScript plug-in isn't going to fix this one.

Steve: Oh, actually it does.

Leo: Oh.

Steve: Because, well, not because of disabling scripting, but because Giorgio - Giorgio Maone is the guy's name, is the author of NoScript, he picked up on this early. He was clued in by the security researchers about what was going on. And he's enhanced NoScript to specifically add a new feature that he calls "ClearClick." So anyone using Firefox can - actually he's had it for some time. I already had a version just through the regular NoScript updates. Because this is about, now about three or four weeks old, since the news first surfaced. I guess about three weeks old. So he's already updated it. He's done a whole series of updates. I updated myself again when I was coming up to speed because he's had a bunch of updates. But there is now, by default, NoScript, even when you allow scripting, it will block these types of attacks.

Leo: Oh, that's great.

Steve: Yes, it's really great, Leo. So anyone using Firefox and NoScript, even if you disable NoScript, if you say "Globally allow scripting for trusted and untrusted sites," he'll protect it. Basically what he's doing is he's looking at, in detail, at what's going on when a user clicks something and will pop up a dialogue saying, wait a minute, you may not be clicking on what you think you're clicking. So he's actually checking the content on the fly to make sure that there aren't any games being played. Which is very cool. Okay.

Leo: So do you - let's take a break, and then maybe you can explain a little bit about how this works.

Steve: Sure.

Leo: And what it looks like. Are you going to give out the URL for your special test site, too?

Steve: Sure.

Leo: Or do you want to keep it a secret?

Steve: No, no. I'll have the formal URL. I did create, for people who just want to easily see what's going on, snipurl.com/clickjack with no "ing," just c-l-i-c-k-j-a-c-k. Our TWiT Live users can do it right now if they're curious. That will redirect your browser to a much

longer and hairier URL that I do also have on this week's show notes for people who don't want to use SnipURL, or they've got browser redirects disabled or something. It's a nice little page because it sort of clearly demonstrates what's going on. If you were a MySpace user - I'm not. But if you were, this would be - it demonstrates how you could inadvertently give the world access to your MySpace profile, even though you didn't intend that. But you don't have to do that. He shows you, he's got two little links that reduces the opacity of the covering layer, allowing you to see the MySpace page that's lying underneath that you were inadvertently clicking on. Anyway, it's just a simple, very visual demo that demonstrates the power of this kind of attack. So that's snipurl.com/clickjack.

Leo: You didn't do this. Somebody else designed this page; right?

Steve: Yeah. I saw somebody else, or somebody in TWiT Live said, "Holy crap, this is scary." If you look at that page, you really get - it makes such a nice, visceral...

Leo: Demo, good demo.

Steve: ...demo that - so what this means is...

Leo: The MySpace thing is a good choice because I suspect that people are actively using this to hack MySpace accounts.

Steve: They absolutely could. So the idea is that, you know, if you imagine that you are a normal web user, you put in a URL to do something, to go somewhere, to eBay or to PayPal or MySpace, for example, where one way or another you are known by the site. Remember we've talked about the problem with persistent log-in. So persistent log-in is a problem. But notice also, for example, that if you have Firefox, as I often do, remember my password. Then I go to the site. My email address or my username is filled in. Firefox then fills in my password. So notice that I now have one-click log-in. Even if I have logged out when I was last there, just going to that page prompts my browser to log me in.

Well, even though that page waits for a click, if that page were hidden behind some other screen where I would innocently click something, I would be logging into a site with my full credentials, even if I had last left it logged out. So logging out of a site doesn't protect you from this kind of exploit. So now imagine that you are, whatever you're doing, banking or PayPal or something less significant like MySpace, but still something where you can imagine people just having some fun and games. The site has a fixed layout with its various elements of the page at known locations. And so anything that you might do through a series of clicks, you could be potentially induced to doing by someone designing essentially a cover sheet, a page that is over that and induces you to click, for whatever purpose, in some sequence. And you think you're doing one thing. But you as you, that is, you known to your browser and known to that site that has been brought up behind the scenes, you're doing something that you're not aware of. And anyway, so the problem is it is completely cross-platform. IE, Firefox, Safari, Opera...

Leo: Well, it uses web standards. It's using - this is how the web works.

Steve: That's exactly right. And in fact it was of a big enough concern that Robert and Jeremiah aborted their presentation that they were going to give at the Open Web Application Security Project. Specifically, Adobe said please don't make this public. Because several of their examples demonstrated how easy it was to cause Flash to turn on to give anyone remote access to your webcam and microphone. And so what happened is - and this is one of the other takeaways. So the first takeaway action item from this is for Firefox users, just update to the latest version of NoScript, and you've got blanket protection. Giorgio has got NoScript up to date, and has actually for some time. You've got blanket protection. Also the latest version of Flash Player, I think it's like 10.0.12, I updated both my Flash Player...

Leo: Yeah, that just came out, actually.

Steve: Just came out, yes. And this is Adobe's response to this specifically. What I had before was 9.0.124. That was a recent update. That and earlier are vulnerable to this. So Adobe has responded to this threat by updating Flash to v10.0.12.36, which is what's current. And it's available for all the various browsers. You want to update to that. And then that's Adobe taking responsibility for making sure that this will no longer function.

Leo: So it can be done in Flash as well as just plain old HTML.

Steve: Yeah, exactly. There's a way, for example, that if someone - if a site tries to - if a site wants to look at you, a pop-up - Flash will present...

Leo: They turn on the camera, that's right, yeah, yeah.

Steve: Exactly. It'll pop up something saying, hey, this site you're visiting wants to turn on your camera.

Leo: I've seen that, yeah.

Steve: Yeah, exactly. Well, what this clickjacking, or so-called "UI Redressing" allows, is that allows that pop-up to be essentially a pop-under, so that it will not come up on top. It'll come up behind.

Leo: And then when you click "Okay" on some other button, it says "Okay" to that.

Steve: Exactly.

Leo: And they're looking at you, or listening to you.

Steve: Exactly.

Leo: Oh, man.

Steve: And so...

Leo: Oh, man.

Steve: Yeah. It was, you know, Adobe said please don't go public with this. The guys gave a much toned-down presentation. They were already on the schedule. They had to do something. So they kept it quiet. However, as we've seen before, just the fact that they had said something caused other researchers to figure it out. And it took about two days after all this first little flurry that it had been - that people had figured out what it was. There were proof of concepts popping up on the web. And then, after it had been worked out, these guys came back and said, okay, well, since the cat's out of the bag, here's the whole story. And they gave a really complete layout of this.

So, yeah, I mean, the problem is, as you said, Leo, this is not a flaw. This is just the power of the system that has been built. I mean, the capability is so extreme now with frames and layers and transparency and CSS and dynamic HTML, that there are unintended consequences of this kind of power. Now, one of the things that has been done, there is something called "frame breaking." It's possible for a site to put some JavaScript in its pages where the JavaScript itself says do not allow me to be in a frame. That is, I'm a page that wants to not function in a frame because I want to prevent any kind of frame-based exploits. I want to prevent the abuse of my pages being enclosed in a frame.

The problem is that requires JavaScript in order to function. And there is a tag that Internet Explorer offers in an iframe where you crank the security up, saying I don't want - I want to bring up an iframe, that is, an inline frame, I want to pull up some content from some other site in an inline frame. But to protect myself I want to disable any scripting. So in disabling any scripting, you're disabling the frame-breaking intent, which is to help you in a site that you're bringing up. So it ends up being that you've got IE working against the positive security enhancement of the content that you're bringing up in the frame.

Anyway, the point of this is it's a mess. And ultimately browsers will have to adapt to this. There is nothing at the moment that Safari, Opera, and IE users can do. There is no - there's no similar protection that I have discovered in all of my looking around that is similar to what NoScript does for people who are using Firefox.

Leo: Firefox, NoScript. Firefox, NoScript. Firefox, NoScript. Just use it.

Steve: It's another...

Leo: It's free.

Steve: It's one more reason. And I have to say, you know, I'm as late an adopter as there is. And, I mean, I really, as I said, I think it was last week that one of the things I felt uncomfortable about Firefox was I was just afraid of incompatibility. I thought, oh, you know, I don't want to have anything, anywhere I go, not work. Well, I've been using it now for several months. And, I mean, I'm not a wild surfer. But I've never, never had a site that I have problems with. And since its adoption is such a high percentage now, webmasters, you know, it used to be that only IE, sites were only being checked in IE. Now, especially with Apple coming on as strong as they are, sites have to work in Safari. They have to work in Firefox as well as IE. So again, I'm very pleased with Firefox. And this NoScript add-on is a beautiful solution. It gives you complete protection from clickjacking.

Leo: Which is very good news because I don't want anybody turning on a camera on me.

Steve: No.

Leo: Without my knowledge. I have them on all the time, but...

Steve: That's not a good thing.

Leo: When they're not on, I don't want them on.

Steve: Or, I mean, or transferring money out of your bank account.

Leo: Or that, yeah. Or just stealing my MySpace or Facebook page.

Steve: Yes, or exactly, changing configuration of standard high-traffic sites and site experiences, behind your back or without your knowledge.

Leo: Now, we talked last time about logging out on all the sites that you don't want anybody to automatically log-in. That's another thing you could do to be proactive about this; right?

Steve: Yes, still a good thing to do because the idea is that this is like you going to a site and pushing buttons you don't intend to. So if you, like for an eBay user, if you stayed logged into eBay, you could imagine that somebody could - imagine them making you push buttons you didn't intend to. There are things you could do you wouldn't want to do. But if you were not logged into eBay, and you had to first log in...

Leo: You'd know.

Steve: There's not a way they're going to trick you into doing that.

Leo: Right. Well, I mean, it'd ask you to log in.

Steve: Right.

Leo: And you'd say, well, what are you talking about? I just - I wanted to punch the monkey. I'm not...

Steve: Well, unless you had - you, like I do, have your browser configured to remember your password.

Leo: There's still an "Okay" dialogue. And so they'd have to kind of get you to click that; right?

Steve: No, because you'd think you were clicking some monkey paw or something.

Leo: So all they have to do is have that click, click the Submit button. And since it's auto-populating the log-in and password, that'd be enough.

Steve: Exactly. That's the problem because you're not having to provide that explicitly. You know, we've simplified our lives by having our browsers remember how to log us on. Well, in the process we've simplified the attack because all we're doing is clicking Okay. We don't know we're actually logging on. We think we're doing something else.

Leo: There's always a lot of trial and error in getting this to work. So they would have to do some clickjacks that would assume you're already logged in, and some that assume that you're not, but that your password's autofilling.

Steve: Yup.

Leo: But I'm sure they can do that. Hackers are nothing if not persistent.

Steve: So our users were sending email several weeks ago saying, hey, Steve, talk about clickjacking. Now we've got the whole readout on it. I've got lots of links for people who want to pursue this further. But, you know, the takeaways are Firefox with a new version, a recent version of NoScript, and make sure you update your built-in Flash plug-in to Adobe's latest. Just go to Adobe.com. Right on Adobe.com's home page it'll say "Get Adobe Flash Player" for your various browsers. Do that, and you're covered.

Leo: Just updated NoScript on my Firefox. And just it's a good reason to keep using Firefox. And I'm not going to say I told you so. I won't. I won't do it.

Steve: Well, no. Because, you know, I've been Mr. NoScript for a long time.

Leo: You have been Mr. NoScript, that's true. You get credit for that. Steve Gibson is the host at GRC.com. Go visit. Boy, there's lots of good reasons. I mean, of course there's SpinRite, everybody's favorite disk recovery and maintenance utility. But it's also the home of Security Now!. You'll get your show notes there, links to the page he just mentioned, 16KB versions of the show, and transcripts, too, so you can read along as you listen. People often find that helpful to understanding the show. And every episode we've done, all 167 of them - 168 of them now. You can also get some great free software. And don't forget ShieldsUP! when you're installing your new firewall. It's GRC.com.

Steve, thanks so much. I won't talk to you again till after the election. I presume that you - are you going to vote on the Tuesday? Or are you going to mail in your ballot, or vote early?

Steve: I think I'm going to vote early, just because, you know, why not? It makes sense to do that. I know what I want to do. And I think I like the idea of getting that done.

Leo: It's nice, yeah. I already voted. Jennifer and I already mailed our ballots in.

Steve: Neat. Neat.

Leo: Yeah, we do that every year because I never know - it came from when I was going to Canada. I never knew if I'd be here on Election Day. And I didn't want to miss an election. So we just encourage everybody to vote. However, this is a very important election. Not just the federal election, but every state has some important propositions. We do here in California. So get out there and cast your vote, let your voice be heard.

Steve: I did want to mention, just to remind our listeners, I really appreciate the feedback that we get. It helps me, gives me ideas for the show, tells me what's going on, what things people think are important. And that's GRC.com/feedback.

Leo: Great. Everybody go there. And we will have questions and answers next week.

Steve: Next week.

Leo: Next week. Thanks, Steve.

Steve: Hey, Leo, bye.