SECURITY NOW!

Transcript of Episode #111

OpenID Precautions

Description: Having several times addressed the value and potential of the open source, open spec., and popular OpenID system, which is rapidly gaining traction as a convenient means for providing "single sign-on" identification on the Internet, this week Steve and Leo examine problems and concerns, both with OpenID and inherent in any centralized identity management solution.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-111.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-111-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 111 for September 27, 2007: OpenID Precautions.

This is Security Now! Episode 1-1-1. That seems like either an ominous or an auspicious number. I can't quite figure it out. Maybe Steve Gibson, our security guru, can [indiscernible] with the numbers.

Steve Gibson: Well, we know me with all of the Pretty Perfect Passwords and all that stuff. I do like alliteration. So I suppose 1-1-1 is alliterative.

Leo: And it's binary, what, 16?

Steve: And 7. And 7. Yeah. If we ever get to binary 16...

Leo: That's 65,635 podcasts, ladies and gentlemen.

Steve: It'd be podcast 1,111.

Leo: Oh, that's right, it would be, wouldn't it, yeah.

Steve: It could happen.

Leo: It could happen, if we keep up this torrid pace.

Steve: I'll tell you, our listeners want it, so...

Leo: There you go.

Steve: ...what the heck.

Leo: So today we're going to talk about what we've talked about before, OpenID, but maybe some caveats.

Steve: Yes. I thought long and hard what to call this episode. And after putting it all together I thought, okay, "precautions" is probably the right word. So we'll do that.

But I've got a couple errata, little bits I want to get to first. As I said last week when we sort of covered the issue prematurely, the day after people are listening to this, that is, on Thursday, the day after on Friday I'm receiving, thanks to our listeners, the People's Choice Best Science & Technology Podcast Award.

Leo: Woohoo! Woohoo! Yeah! All right!

Steve: I want to thank everybody again.

Leo: It's a party, you're a party, yes - okay. I'm sorry I'm not going to be there. I'm celebrating now.

Steve: That's good. And...

Leo: That's really great. I'll be thinking of you on Friday when you get that award.

Steve: Speaking of alliteration, we have John J. Jost, I guess that's how I pronounce his name, JJJ. He posted in the GRC newsgroup just a little snippet that I thought was kind of fun. He mentioned that - he said actually ultraviolet, UV tattoos, which was what we discussed last week in our wacky number 13, are visible in ordinary light. "My son and I were just at a tattoo shop, and he asked about getting one. The shop owner recommended against getting it in any place that was visible anyway for ‘job application' reasons. He said that the tattoo would be dull but still visible." And so John says, "So I wouldn't recommend getting a tattoo for security purposes." And then...

Leo: We need to find an invisible ink tattoo technology.

Steve: There you go, exactly. And then a great friend of mine and past collaborator, Jon Lundell wrote to me. He and I correspond by email a lot. Sort of with his tongue in cheek he said, "Regarding tattoos, password revocation could be a literal pain in the butt."

Leo: And that's all. That's all you need to say.

Steve: Exactly. Get your butt cheek tattooed, and you do not want password revocation for that.

Leo: I love it. Is that it?

Steve: Yup.

Leo: Boy, that was easy.

Steve: Well, you and I are on a tight, tight schedule today because we're doing two, and then you're off to Vancouver, so...

Leo: Yeah, Amber and I are reunited tomorrow.

Steve: Oh, how cool.

Leo: As this airs, we'll be doing a kind of a mock debate. I don't know how you can debate the subject of LCD versus plasma, but she's going to take the LCD side, and I'm going to take the plasma side. And I'm betting Amber, knowing her, was probably a college debate champion. She's probably going to run me into the ground here, but it'll be fun. It'll be interesting.

Steve: She's probably doing lots of research.

Leo: Yeah, knowing her. Anyway, that'll be - if you hear this before 11:30 a.m. Pacific time, and I'll make sure the podcast goes up before then so you have a chance to, tune in at FutureShop.ca. They're going to webcast it. And I think a thousand people, they can handle a thousand people. Past that I don't know if they'll...

Steve: Uh-oh.

Leo: But that'll be fun. So I'm just going up for the day. And then I'm coming back on Friday, and then I'm going back up on Monday. Fortunately it's easy to get to Vancouver, as you know.

Steve: So we're squeezing two recordings of podcasts in before your plane leaves in about an hour and 50 minutes. So our listeners don't have to worry about another 90-minute podcast.

[Talking simultaneously]

Leo: Let's see. You want to talk about SpinRite a little bit here?

Steve: Well, I don't want to slow things down because we're on limited time. Certainly all of our listeners have heard lots of testimonials. And I'm getting lots of them, which I really appreciate. Let's skip it this week and get right into our content.

Leo: Well, we'll just say: Buy SpinRite. That's all I need to say. If you, you know - it's the ultimate disk recovery and maintenance utility. It's really good. Maybe next time then I'll talk a little bit about - I had a caller call in the radio show with a question that I think SpinRite was perfect for. But anyway, we'll talk about that in a bit.

Steve: Okay, cool.

Leo: Today OpenID is our topic. Now, we have talked about OpenID a couple of times so far.

Steve: Well, we've talked about, yes, about OpenID specifically, about various OpenID services, about VeriSign, about their PIP program which is in beta as an OpenID authentication or so-called identity providing service for OpenID. And mostly I've been jazzed about the idea that we're beginning to see some sorts of solutions, good solutions, for the problem of authentication, the idea being that the existing model is people are having to create usernames and passwords to identify themselves uniquely from all other users on the Internet for this rapidly proliferating number of websites where people want to be participating in social networks, they want to be adding comments to blogs, they just want to be interactive. So...

[Talking simultaneously]

Leo: ...called a single sign-on. And that maybe is a little bit easier to understand in that context than OpenID, the idea of a single sign-on that applies to all of these sites.

Steve: Right. Well, what happened was that a recent article that was put up, a blog posting essentially was put up by what is arguably a competitor of OpenID. There is a commercial service called Credentica that is an industrial-strength, not free, patented eight ways from Sunday, these guys have eight patents that just lock up their technology. And one of the guys that works there did a blog posting where, I mean, the guy must have gone a lot of Google searching because this is a long posting, full of all the horrors of OpenID. And OpenID is becoming popular enough that something like this that tends to throw a lot of water on the concept came to a lot of people's attention. Digg picked it up, and it got well dug. I've been bombarded with people saying, hey, Steve, did you see this, did you see this? Yes, I've seen this.

Well, I've since researched every single issue carefully, and I want to address this because so many of our listeners have been specifically concerned about this blog posting. But moreover, I mean, it certainly does bring up some points that are worthy of discussion, which is why I wanted to title this OpenID Precautions, as opposed to, for example, I don't know, death, end of life, nightmare, I mean, things more extreme.

Leo: Well, I think it's great that you have a lot of review of these kinds of things from a lot of experts. I mean, if it's a competitor, especially if it's a commercial competitor, well, you have to take it with a grain of salt. But hey, let's address these issues. I mean, I think that is perfectly appropriate.

Steve: Exactly. And there's certainly more than a grain of salt. I'm going to read from the top of Stefan Brands' posting just to give our listeners some quick sense for his take. He says, "OpenID was designed as a lightweight solution for ‛trivial' use cases in identity management. Its primary goal is to enable Internet surfers to replace self-generated usernames and passwords by a single log-in credential, without needing more than their browser. Concretely, OpenID aims to enable individuals to post blog comments and log into social networking sites without having to remember multiple passwords. (Of course local password store utilities already do that. More on this later.)"

And, he says, "Beyond this, OpenID is pretty much useless. The reasons for this are many. OpenID is highly vulnerable to phishing and other attacks, creates insurmountable privacy problems, is not a trust system, suffers from usability problems, and makes it unappealing to become an OpenID consumer. Many smart people have already elaborated on these problems in various forums. In the rest of this post I will be quoting from and pointing to their critiques." Which is what Stefan does. And he does so effectively because I don't argue with any of the things he said except that they've really been sort of taken out of context.

So let's step through some of these issues and precautions because certainly our listeners ought to understand what's going on with OpenID, what is the downside of using it, what are the dangers and precautions. And it's also worth noting that there is some fundamental trouble because not everyone on the Internet is listening to this podcast. And as we'll see here in a second, it is easy to be, unfortunately, lulled into a false sense of security to believe you're getting more than you really are from OpenID. And that's really the only problem I have with it.

So first of all we know, just to recap briefly, how the OpenID system works. In wanting to identify yourself to a website, you give that website your OpenID URL, which is what they literally look like is a dot-separated set of tokens like a URL. So you say, you know, this is who I am. You submit that, and the site you're at then sends your browser through this process of redirection that we've talked about before, sort of in the same way that buying something through PayPal redirects your browser over to PayPal, where you authenticate yourself to PayPal, and then you go back to where you were, taking the information that you have paid along with you, and then the site is able to verify and so forth. Similarly, OpenID, the way this works is the site you want to authenticate to bounces you and your browser over to the place where you have said your OpenID credentials live. Essentially the idea is you authenticate yourself with this third-party site. And then, having done so, it bounces you back to the site where you wanted to log in, and everything proceeds from there.

Well, it is probably the case, given that here we are on Episode 111 of Security Now!, that about half of our episodes have been about problems with browser security and scripting and cross-site scripting and all these problems. It's probably the case that the phrase "web browser security" is an oxymoron.

Leo: Yeah. I mean, that's really becoming the number one vector for infection is the web browser.

Steve: Because it's so popular, it's the way people view the Internet. And unfortunately, many of the things we want to use are inherently hard to secure, difficult to secure. And we know that security is not black and white. It's all just kind of goo. And so things that are harder to secure tend to be less secure than things that are easy. So the browser is an inherent problem.

And unfortunately the power of OpenID is that it uses the browser, which we all have, and the browser experience, which we all have, to perform what it would like to have be a lot of security. But it's using the browser. So there we are sort of with this conundrum that it's just difficult to have a browser be secure.

Now, the perhaps number one complaint people have with OpenID - and that's not just Stefan, but many of the links he provides are people who for whatever reason are upset about OpenID. I'm sort of glad they're upset because it's certainly not a problem to raise these flags and to work to get the OpenID system bolted down tighter and to spread the news that people need to be concerned about the security of this. But, for example, everyone talks about phishing attacks, how OpenID is, like, made for phishing attacks. The reason is this whole browser redirect thing. We know that phishing attacks are, for example, you get a link in spam email saying, "Click here because Bank of America needs to update your log-on information." When you click the link, you don't go to Bank of America, you go to a Bank of America clone site which is hoping to trick you into giving them your Bank of America log-on information.

Similarly, there's clearly an inherent vulnerability with this OpenID approach where you are bounced to a third-party site. If you are bounced to something that looks like your OpenID authenticator, how do you know it really is? Well, one simple thing, and that is, SSL, secure socket, HTTPS protocol. We've talked about this over and over and over about how you can avoid the phishing problem by making sure you've got a secure connection, the page is secure, and the credentials for the site match the certificate. There is no effective way for that to be subverted. And that one fact is never mentioned in all of these examples and attacks on OpenID from a phishing standpoint because they all know, if we prevent phishing, then there's no way a site can pretend to be something it's not.

Leo: Furthermore, if SSL is vulnerable, then we're really in big trouble. I mean, OpenID is the least of the problems.

Steve: Well, and it's not vulnerable. And that's the point. But here's the problem. The onus is on the user because we know how many people get phishing attacks. And so, I mean, this is really worth making sure we understand, and that is that, because we're browser based, that's convenient; but it's also a problem because obviously the whole browser experience is not very secure. So if users did not verify that they were actually giving their credentials, that is, logging in, giving their OpenID verification to their real provider, then it absolutely could be a phishing attack where they went to FreeFlowersRUs.com that said, oh, give us your OpenID. Well, that could be a malicious site. And but here's what...

Leo: Let me ask how the mechanics would work because the way I use OpenID is - maybe it's more secure, but I have, as we mentioned, an OpenID token on Leoville.com. So I go to the flowers place, and all I do is give them Leoville.com.

Steve: Right. Except, okay, so you give them Leoville.com. They go to Leoville.com.

Leo: Oh, they have to spoof my OpenID provider.

Steve: Exactly.

Leo: So they'd send me back a page not SSL encrypted that said, I am VeriSign. Give me your PIP key.

Steve: Exactly.

Leo: This is why, by the way, the little key thing is great. Because even if I do get phished, it doesn't - it's only temporary.

Steve: Well, that is exactly right, Leo. And that's a point I was going to bring up later.

Leo: I'm sorry. Okay.

Steve: No no no, because one-time passwords solve this problem except for that one log-on. That is, so imagine like a super phishing server that you get sent to. It knows where you were really going to be sent. It turns around and sucks down the page from your real OpenID provider...

Leo: Spoofs it.

Steve: ...on the fly, and then, exactly, and then presents it to you so that, I mean, so literally it's able to pretend to be any OpenID provider because it just grabs the page just like your browser would, then turns around and gives it to your browser. So...

Leo: So let's say you don't have a token. So now you've given it your password. But then you say the onus is on the user to notice that you're not on an HTTPS page.

Steve: Correct. Correct.

Leo: Well, that is the flaw. I mean, a lot of users won't notice that.

Steve: Yes, it really is a problem. I mean, and so it is really a precaution that comes along with this is people need to understand that - and I think this is people looking properly into the future as OpenID gets more traction, as it becomes more valuable, it's protecting not just blogging posting but more and more assets, as will inevitably happen unless the whole OpenID movement collapses. And at this point there's more than 5,000 web services now using OpenID, giving users the option of using OpenID for their authentication.

So this thing, I mean, the reason there's so much fur flying here is that it is gaining traction. It's taking off. It's a valuable thing. The problem is we need to understand what the limitations are to essentially use it responsibly. So we've got the fundamental problem of using a web browser and the problem with phishing, which is truly a concern.

Now, there's sort of an overall problem which is not OpenID's fault at all. It's just it comes along with the problem of identity centralization. And that is, you're inherently putting a lot of trust in this, well, in the hands of your identity provider. Essentially, you're concentrating trust in a single point of failure where, if that point failed, it would be much more devastating than creating individual credentials for every site you visit, which is inherently a distributed trust model where, if one password and username got compromised, well, as long as you didn't use it anywhere else, you're not vulnerable beyond that single location.

So there is a problem that's inherent in the benefit. I mean, we want centralized identity management because that's the power of this approach. But inherent in that is what happens if that breaks, what happens if it fails, what happens if it gets compromised. So you want somebody you can trust with strong technology. Again, I think one-time passwords is really the way to go for this approach. And I've got a fun surprise in two weeks for people because I've come up with a free one-time password system that we're going to be talking about in two weeks.

But the one-time password solution which you briefly mentioned, it solves the problem because your credential cannot be reused. That is, if you were intercepted, and you logged on to a spoofed phishing OpenID page, well, it would get your log-on credential once, but it would never be valid again. So some damage could be done based on where you were logging on. But it wouldn't be an epidemic of you completely having lost control of your identity, which frankly, if your password and username for logging onto an OpenID server were not really protected, and it did get away from you, then users wouldn't be looking at having their identity lost. And we know what a problem that's been for people who have been subject, for example, to real world identity theft that, you know, it takes them years to recover.

Leo: Isn't it - I guess part of it is right now we're not using it for anything really serious. I guess it becomes much more serious when we start using OpenID for banking and stuff like that. But nobody's doing that yet.

Steve: Well, and you might argue, too, that maybe we never should. That is, maybe OpenID needs to sort of be given the proper place in a hierarchy of authentication. That is, well, okay, it's probably the case that we have a pyramid, that there's a vast number of low-value sites that we authenticate ourselves to, like all the social networking sites and blog postings and all that, and many fewer high-value sites. I would argue that, if nothing else, it's premature at this point to trust OpenID with logging in there. And frankly, I'd be very surprised if you see Bank of America with an OpenID log-on unless there's backup verification of some sort to lock this down and prevent abuse. So I'd be surprised if very high-value solutions were using or even offering OpenID. I mean, it certainly makes sense as a convenience. But there is this downside.

Now, another aspect of identity centralization trouble is the problem of identity server becoming unavailable. If you are trusting an OpenID identity provider that is a constant victim of denial of service attacks, or they've got their own server problems, or their server is overloaded or whatever, suddenly you're unable to log into any of the sites using that identity provider. Now, the good news is, because the whole system is open and free, nothing says you can't have three or four identity providers, that is, three or four different URLs referring to different providers, and have credentials with each of them. So that, again, it's easy to have a backup to work around this problem. This problem is posed as, oh my god, this is a reason why OpenID you can't rely on. It's like, okay, well, fine, but it's easy to work around that one, at least.

Leo: All right.

Steve: Now, another problem, and this is serious enough that we're going to do a podcast, actually I've got it set up for No. 115, and the title is Third Parties. I mean, just the whole problem of third-party involvement. And this is a problem from a privacy standpoint because, if you think about it, that OpenID server knows everywhere you go and log in because you are bounced to it from the site you're logging into. If it cared to do any sort of aggregation of that information, it's certainly able to do so because your browser's coming from that site and is being sent back to that site. So it's inherent in the OpenID model that the server you use for providing your identify credentials knows every site that you use it with, wherever you go on the Internet. Again, maybe that's a concern, maybe not. I'm sure this varies depending upon the sites that you're going to log into and just sort of individual users' feelings about that kind of privacy issue. But again, it's something that people need to be aware of is that, again, it's a matter of trusting this identity provider. You really do want to have trust there.

Now, one interesting problem that had never occurred to me, but I'm glad that I looked through this downside of OpenID posting, and that's the idea of OpenID URL recycling. Apparently major OpenID providers will make old URLs available after they've expired, after they haven't been used for some length of time, just so that, for example, John Jones, who once created an ID there, if he goes away or changes providers, the site may not want John Jones' log-in to forever be unavailable. So after some length of time, a site can recycle John Jones' name, making it available to someone else.

The problem is, that's a URL, johnjones.myopenid.org or whatever, which is used to identify not only the new John Jones, but the old one. So if new John Jones went to a site where old John Jones had a history, maybe all of his photos uploaded or what else, since that's the token which is being used to identify the user, there is a real re-use problem because that site, unless there was some sort of one-time number or an incrementing token or some other means for identifying the account at the OpenID provider other than just that OpenID URL, you'd have a serious problem of confusing the John Joneses, and old John Jones could find new John Jones having access to his Internet assets, where he doesn't want that to be the case.

Leo: Clarify for me, this is because the OpenID provider maintained this account even after it was closed or moribund?

Steve: Well, yes. Say, for example, that we had myopenid.org was an OpenID provider. And John Jones comes along and creates an account there. Well, then his identification is johnjones.myopenid.org. That's the URL that he uses to identify himself to all these different sites on the 'Net. So he then changes OpenID providers and creates an account somewhere else. Then the original OpenID provider, myopenid.org, sees that this John Jones hasn't logged in or used his identity for a year, say. I'm just making that up, but for some length of time. And they decide, oh, let's just cancel that account. We don't want to store any more information about this. And in the process, that becomes available again. Now a new John Jones creates an account; but his URL, johnjones.myopenid.org, is the same as the other guy.

Leo: So how do OpenID providers handle this right now? Do they deprecate the use of that name forever afterwards, or...

Steve: Well, you would like no one to ever be able to use the same account again.

Leo: I think they need to do that.

Steve: Yes, exactly. I mean, it really does represent a problem. And the final real issue, well, it's sort of another one of these, is that OpenID users would like sites to require nothing but their OpenID. That is, you know, you go to a site that says, oh, you can log in with OpenID. So you put in your OpenID, you'd like it to just be bang, you're done. People are complaining that they're still having to give an email address, and then have an email address loop where they're having to deal with CAPTCHAs and the traditional things.

Well, the problem is, OpenID does not by itself solve the bot problem. So OpenID solves the I'm asserting this is who I am when I'm here and when I come back. But you still have to deal with the problem that bots can certainly have OpenIDs. Nothing prevents them from doing that. So the same things that are in place now using email loops where you give them an email address, they send you a link that you have to click on in order to verify that you own that email address and authenticate yourself, or CAPTCHAs that we've talked about so much. These things are still necessary, at least initially, to establish a new OpenID at a given site where you visited, which annoys people because they thought that's - they thought incorrectly that that's what OpenID was going to solve. They were going to eliminate all of that. In fact, it doesn't. It eliminates the notion of needing individual, decentralized authentication with individual usernames and passwords. It does not eliminate the problem of verifying that you're human as opposed to being a bot.

Leo: Right, right. Well, these are, I mean, on the face of it these are obvious flaws. There's nothing surprising here. He's not saying the underlying technology is somehow flawed. These are just potential, I guess almost implementation issues.

Steve: Well, it's why I think the proper name for this episode is OpenID Precautions. You know, they're issues that OpenID users should be aware of. I regard this whole technology as still immature. It's got a ways to go. The OpenID spec has been moving through versions. There's a 2.0 on the way. It's been through several 1.x versions. I think there's clearly a place for it. But like everything else in security and the technology of the Internet, we want to make sure that it's understood, that is, that the things it offers are understood, and the limitations and vulnerabilities are understood. And certainly it's the case that unaware users could be bitten by this simply because, due to the fact that it is web-centric, there are too many ways to fool people who are using browsers on the web. And OpenID, again, because it uses this technology, it's potentially prone to be a victim.

Leo: I'm sure you'll put a link to the Credentica blog posting on your show notes.

Steve: Yes. In fact, I've got a whole series of links of this stuff for people who are curious to perform some further research.

Leo: So GRC.com is the place to go for Steve's website, and that's where you'll find the show notes. That's where you'll find links to a 16KB version, all the links that we talked about, and transcripts, too, so you can kind of read along. And a lot of times I think we get so meaty on these that it's useful to have a transcript to get the - to tease the details out.

Steve: Oh, yeah, a lot of people really rely on the transcript in order to kind of go over it at their own speed.

Leo: Absolutely. So all of that is available at Security Now!'s website, GRC.com/securitynow. By the way, Steve also at that site has, I don't know, more than a dozen free security utilities, including the world-famous ShieldsUP! to test your firewall. It's just a really useful site that you ought to know about. GRC.com. And of course that's where you'll find SpinRite. He doesn't want to talk about it, but I'm going to do it anyway. It's the world's finest disk recovery and maintenance utility. If you're not using SpinRite, you're missing out. Nerds, by the way, also get a SpinRite license, which is kind of nice. GRC.com. So next week, Steve, we've got a Q&A section and lots of great questions. If people want to add their questions to the mix, how do they do that?

Steve: They can just go to GRC.com/feedback, and that gives them a form where they can submit questions that come directly to me.

Leo: Good. Well, thanks for this update on OpenID. I'm still going to use it, but I'm going to make sure that I pay attention that, you know, SSL is turned on when I enter my password. But you should do that anywhere you enter your password, I guess.

Steve: Absolutely, that's a habit you want. And really, if you do that, all of this other stuff is - it just falls away. That really is the only real problem is, I mean, the major problem is phishing. And the rest are just sort of, well, that we talked about are inherent aspects of the benefit of concentrating your authentication with a single provider. I would just say you want to choose that provider well.

Leo: Well, and that's the balance. Again, convenience versus security. Sure, the best thing would be to have a 64-byte, completely random numbers, letters, and punctuation password for every site you visit. I'm sorry, ain't gonna do that. And for most people the alternative is far less secure. They use one password for all sites, one easy-to-remember password for all sites. And I don't think anybody would deny that that is a bad idea. All right. We're going to wrap this thing up. Thank you, Steve. We'll see you next week on Security Now!.