Internet Identity Update

Description: After catching up with an eventful week of security news, Steve and Leo step back for an overview and discussion of the slowly evolving state of the art in Internet Identity Authentication.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-369.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-369-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to talk about, of course, the Microsoft Second Tuesday Update, the issues with GoDaddy going down daddy, and we'll talk a little bit about OAUTH, OATH, and other Internet identification protocols. It's all coming up next on Security Now!.

Leo Laporte: It's time for Security Now! with Steve Gibson. This is Episode 369, recorded Tuesday, September 11th, 2012: Internet Identity Update.

It's time for Security Now!, the show that protects you and your loved ones and even the people you hate online. When it comes to security and privacy, there's no one better than this guy, the Explainer in Chief, Steve Gibson, who plays no favorites.

Steve Gibson: We're an equal opportunity protector.

Leo: Hey, Steve.

Steve: That is correct. Hello, Leo. Great to be with you again, as always.

Leo: Steve is the guy in charge at Gibson Research Corporation. They do a great piece of software, we'll talk about it later, called SpinRite, world's best hard drive maintenance and recovery utility.

Steve: Yeah, no one's ever heard of it before, Leo.

Leo: Nobody on this show has ever heard of SpinRite.

Steve: No.

Leo: And then, of course, there's lots of freebies and so forth. But I think at this point you may be best known for Security Now!. I'm just guessing. I don't know.

Steve: I think that's probably the case.

Leo: 369 episodes in.

Steve: I figured out how hard drives work, and then I thought, okay, I'm bored. What else can I...

Leo: That's, you know, I think that that's the blessing in this industry is they get a lot of people who are easily bored. And so we move on to other things all the time. Today we're going to talk about identity.

Steve: Yeah, that's a constant recurring topic, not surprisingly, because in my opinion it's probably the No. 1 most important aspect of security and privacy. We come at it from the crypto tech side. We come at it from the best practices side, and databases being compromised, and what does it mean if Apple's UDIDs all get out. I mean, the whole issue of identity is crucial. And one of the things that I see in the mailbag coming from our listeners is, because there's a bit of an acronym soup, and because so many acronyms are colliding with each other - we talked about OATH and OAUTH last week - I thought, okay, let's just - let's step back a bit and do sort of a "where do we stand and who's winning" because there have been competing standards.

We discussed, for example, actually we've discussed all of these in episodes past, given them their own podcast. But it's clear now that we're seeing a winner. And so I thought, let's talk about the terminology, differentiate these, sort of carefully put them in the context of each other, which is something we've never done, and just sort of do, okay, where are we and where does it look like we're probably going to go?

I did, however, want to make sure everyone knew that we're going to have one of our rare special appearance, guest appearances next week, someone who many of us in the computer industry know by reputation. He's arguably famous, and that's Mark Russinovich, who was the co-founder of Sysinternals that was a go-to website for years when you wanted utilities that Microsoft wasn't producing that really drilled down. Mark has written a bunch of books, I mean, he's a Microsoft operating system internals guy, and we were all a little worried when Microsoft acquired Sysinternals. The good news is those tools have remained available over on Microsoft's site. So they didn't disappear, and Mark has continued to maintain them.

The context for his appearance next week is to chat a bit about his own change of, well, not change of profession because he's still at Microsoft doing that. But he became a novelist. And several months ago he sent me the galleys, I guess you'd call it, of his

second novel, "Trojan Horse," following up on his first novel, which was "Zero Day." And "Trojan Horse" has become available, as I mentioned last week. And so I thought, hey, let's get Mark on and chat with him about...

Leo: That should be awesome.

Steve: ...that stuff and where the world is and what's happening.

Leo: Yeah, that's going to be fantastic.

Steve: Yeah. Okay. So, news. Lots of news. Interesting things have happened this week for you and me to discuss with our listeners, Leo. The happily minor event was - this is, because we're recording this on Tuesday, this is the second Tuesday of September.

Leo: Oh, that's right.

Steve: We're normally always a day late for that. But in this case, oh, actually we ought to take a moment to note also that it's September 11th.

Leo: Should have said something about that. So, yeah. We moved the show because tomorrow's the Apple event, as we always do. And Steve was very generous and kind about that. And that puts the show on a Tuesday.

Steve: Boy, it's such a problem for me, Leo. I just sit in this chair a day early, so...

Leo: I think you're sitting there anyway, probably.

Steve: I pretty much would, Leo.

Leo: But also...

Steve: I answer the phone when you call.

Leo: Yeah, that's basically the difference. But also, yeah, you're right. And in fact maybe it's more appropriate to do Security Now! on September 11th. This is a day, 11th anniversary, of course, of the terrorist attacks on the World Trade Center and the Pentagon.

Steve: Actually one of the main networks, one of the cable networks over, I think it was on Saturday, replayed the two hours, as it happened, of their coverage. And, oh, it was, oh, wow, I mean, it was something.

Leo: I think a lot of us have kind of put - this is what happens when something painful happens. You just kind of lose the memory of it, and you don't want to be refreshed. And boy, when you remember, it was just a horrible day, horrible.

Steve: Yeah, yeah. What were we talking about?

Leo: Oh, Mark Russinovich is going to be on. That's good news. It is the second Tuesday of the month.

Steve: Oh, second Tuesday of the month. And it's a - I read somewhere, and this was not my observation, it was someone else's, but it seems to be holding, and I've shared it before. Microsoft seems to alternate big months and small months. And it's odd that it - or even - that it continues to do that. But last month was a biggie. It was one of those, okay, seriously, you want to update Windows now. And this one is, eh, you could skip a month if you wanted to. Truly, there's nothing critical. The only two patches they have are marked "important." They're privilege elevation vulnerabilities which affect the Microsoft Visual Studio Team Foundation Server 2010 SP1 for developers. And then over on the server side, Microsoft Systems Management Server 2003 SP3 and Microsoft System Center Configuration Manager 2007 SP2. So actually most people who don't have those installed, Windows will go and say, do you have anything for me, and Windows Update will say, uh, no. So I think most people won't even know anything happened on the second Tuesday.

Leo: Wow. It's been a while since that's happened.

Steve: It really has been. Although, if we average things out, last month, as I said, was a "press update now" month, and who knows what we've got in store for us for October. So we'll see. But anyway, if you wonder why your computer just cruises through the second Tuesday of the month with nothing wanting to happen, it's because the very few things that were done you probably don't even have installed on your computer, and so there's nothing to update in your case.

I did want to also mention that it seems that, when we do the podcast, and we talk about Java, a problem gets solved the next day. And this happened again last week. We of course were talking about Adobe, or, I'm sorry, Oracle, fixing the big Java vulnerabilities the day after the prior week's podcast. This time Apple updated Java for their platforms on the Thursday following last week's Wednesday podcast. So I just wanted to note, for those people who are using Java on Apple platforms, that Apple fixed their instance also, a week after Oracle, but in a timely fashion, and that I certainly think Apple's doing the right thing with the measures they've taken to protect their users from Java being enabled all the time, even if it's not necessary and not being used.

Then the most interesting, well, one of the two most interesting events of the week was the relatively massive outage at GoDaddy yesterday.

Leo: Yeah. Boy, did I see a lot of Twitter traffic on that.

Steve: Oh, my goodness, yes. It was crazy. And, now, there was some - it wasn't the Anonymous group, but it was a person claiming to be a member of Anonymous took credit, saying that it was an attack that he had launched. And...

Leo: Yeah, but you know what, after the FBI database thing of last week, I don't credit anything Anonymous says. I mean, they are - it's in their interest to disinform.

Steve: Yes. And in fact that's our next topic. We will come back to the UDID leakage, which you and I were skeptical and, as it turns out, rightly so. But in the case of GoDaddy, when I first saw Twitter begin to say something's wrong, I tweeted that there was apparently an attack of currently unknown origin. And, for example, GoDaddy.com, the DNS would not resolve.

Leo: Oh, really, the main site.

Steve: Yeah.

Leo: Wow, that's not good.

Steve: And the report was that millions of sites that they're in some way responsible for, apparently they run servers with MX records, Mail Exchange records, which were not resolving, so email died for people who are using GoDaddy's MX records as part of their service, and people who are using GoDaddy to provide their DNS. I mean, GoDaddy was dark.

And we've now heard from Scott Wagner, who's the interim CEO. He posted earlier today, I think it was. He said, "The service outage was not caused by external influences. It was not a 'hack' and it was not a denial of service attack. We have determined that the service outage was due to a series of internal network events that corrupted router data tables. Once the issues were identified, we took corrective actions to restore services for our customers and GoDaddy.com. We have implemented measures to prevent this from occurring again. At no time was any customer data at risk or were any of our systems compromised. Throughout our history, we have provided 99.999" - the so-called five nines - "uptime in our DNS infrastructure. This is the level our customers expect from us and the level we expect of ourselves. We have let our customers down, and we know it. We take our business and our customers' businesses very seriously. We apologize to our customers for these events and thank them for their patience."

Leo: So it was not hackers.

Steve: Well, it was not hackers. It must be that this is true because the CEO can't make this up. And because, if it were an attack, there would be lots of other people who knew that. But what's curious is that this was six hours of outage, from 10:00 a.m. Pacific time to 4:00 p.m. Pacific time. That's a long time for some router tables being messed up to, like, be causing all of GoDaddy.com and all of their ancillary services to be - now, I mean, maybe it was just DNS. Because, I mean, of course, as we've talked about, if DNS is gone, everything else is, too, because that's the way we get to everything. So I did not

know what GoDaddy's IP was in order to, like, try to do a connection to them without DNS. And, frankly, I wasn't that interested in pursuing it at the time. But six hours is, ouch, that's a long time to be down. So, yeah.

Leo: Now, TechCrunch is saying that Tipster said that their three DNS servers, CNS1, 2, and 3 .secureserver.net were inaccessible for those six hours.

Steve: Yeah. So that could be...

Leo: I mean, that's, if you really wanted to DDoS them, that's what you'd do; right? You'd hit those servers.

Steve: That's exactly what you would do, yes. And what I tweeted was that apparently someone tripped over a cord.

Leo: For six hours. You know, it took them six hours to find which plug was pulled.

Steve: That's the problem is they've got it. And, okay, now, where does this plug in?

Leo: If you tripped over a cord in our basement, that would actually be the truth.

Steve: There's too many sockets. We got too many sockets and one plug.

Leo: Not the wiring Russell our IT guy did. The wiring we did is a little spaghetti-like.

Steve: Okay. So we did hear from, well, we did - almost certain we believe that the data did not come from the, I'm sorry, the Apple - remember that we discussed it last week, 12 million leaked unique device IDs of 12 million of Apple's devices - iPads, iPods, iPhones. Every device that Apple produces has a 40-character unique ID which Apple uses, I'm sure it's tied to iTunes and accounts and so forth. Unfortunately, the official API up through 5 but fully endorsed up through iOS v4, offered this UDID to any application running on any of those devices. And as developers will do, here was a unique token, ready made, courtesy of the API, which Apple was saying, here, this is a unique ID for the device.

Now, one of the things that I'm taking away from all of this is we're learning how to do this, how to do global devices that are connected. We're not good at it yet because this was a mistake. And we now, as a community of developers and device creators, we're learning, okay, don't do that. I mean, make it not possible. Maybe, for example, have the API which you query produce a unique ID which is different for every app that queries it. And so the apps still get something unique that ties that instance to that device. But it doesn't give a leakable ID that is across all the apps, which is what we have had and what Apple, with the release of iOS 5 they said, we realize this has not been - this is being used in a way which could compromise the privacy of owners of our devices. We would like you to stop using it. And then it was reported that in March they began

declining new apps that wanted to use it.

So anyway, so that's the background. And my sense is, okay, everybody should learn that the idea of a cross-application global ID that's available is not something that in the future we're going to do. Apple's going to fix it. I don't know where Android stands with that. But if they've got something like that, or other phones do, it would be a good thing for everyone to just say, okay, this is not - this is too much opportunity for compromise.

So, as we know, there was the claim, and we shared it on the podcast last week. The claim came with an awful lot of detail about exactly whose laptop and where it was and how this happened and so forth. Apparently all nonsense. The FBI denied that they ever had that data. And what happened was that a couple days ago an independent researcher, David Schuetz - he tweets as @darthnull, and he has a blog at DarthNull.org. And he's a hacker and a security guy. And he grabbed the million and one posted sets of data, which was, you know, the million and one were posted on Pastebin, I think it was, out of all 12 million.

And he was just sort of doing some data crunching, like looking to see - kind of looking at the UDIDs. And he ran it through some filters to see if there were any repetitions, and he noticed a surprising pattern that, I think he was first looking at, like, eight digits of the device ID, and there seemed to be a lot of dupes. So then he looked at the entire device ID, and he found to his surprise that it was a very non-uniform distribution, and that some occurred far more often than most.

And then when he drilled down and looked at the human readable data, email addresses and names and so forth that were tied to those IDs, he saw a pattern and sort of peeled layer after layer of the onion and came up with a theory that there was a very non-uniform instance of devices associated with one company called BlueToad that has a site, BlueToad.com. And they are a company that provides exactly what you and I have commented many times annoys us, Leo, these per-site or per-company iPad apps. So when you go somewhere and you get this window that comes up and says, oh, would you like to get the app for this, for us, for our site? It's like, no, I want to browse on the browser. As you and I have said, the browser is supposed to be the app for the site.

Anyway, these guys are an instance of a company that creates that for companies for tablets and smartphones and so forth. So essentially what happened was, because they were testing their stuff, many, many more of their device IDs appeared across their company than was normal. And so David contacted the CEO of BlueToad and said something's funny here. He was being very responsible, and he certainly wasn't going to point fingers before he knew for sure. And he said, what do you think about maybe you were the source of this massive 12 million-record leak? And it took a couple messages, then the BlueToad guy got back in touch with him and thanked him for being responsible. And he said, "We're looking into this. We'll get back to you."

And then he got another call from the CEO saying, okay, I can talk to you, but I need your agreement to embargo what I'm going to tell you until Monday, that is to say, yesterday, so September 10th, yesterday. And David said, okay, well, the fact that you're asking me to keep this quiet until then tells me that I probably want to say yes. So he said yes. And then the CEO explained that they were sure that they had been the source of the leak. They found the problem. And someone with NBC News wanted to talk to David right now. And so David was on the news and explained how this happened.

And now we know that, I mean, it's not inconceivable that the FBI might have had this as part of a, like a copy of it. They might have already known that it had gotten loose, part of an investigation they were doing. We don't know why this claim was being made. But

it seems that it's entirely bogus.

Leo: Well, I mean, come on. This group is not - they're committing illegal acts, so why you would trust anything they say one way or the other is...

Steve: Right, right. And so I think we need to remember that what was an apparently completely bogus claim was made. And, unfortunately, that hurts their credibility in the future.

Leo: True.

Steve: And for a while they were seeming pretty credible.

Leo: Right.

Steve: Okay. Now, this has been on my list to talk about for a couple weeks, and I've just - it keeps getting pushed down. Many of us who use laptops enjoy the finger swipe, the little - it's UPEK is the manufacturer of the little capacitive finger reader than laptops have. And it's like, oh, look, biometrics, that's got to be more secure than not; right? Well, turns out someone took a look at the UPEK software. And this is software which comes with Acer, Amoi - which is not a laptop I have heard of, A-m-o-i...

Leo: Yeah, they're like a Taiwanese clone, yeah.

Steve: Yeah. ASUS, Clevo, Compal, Dell, Gateway, my favorite the Lenovo, Itronix, MPC, MSI, NEC, Sager, Samsung, Sony, and Toshiba. I think there are 16 of them, if I remember. And their software is in all of those. The message is, if you have ever entered your password into this UPEK software, in order to give it the ability to log you in, then your password has essentially been stored in the clear in the system registry.

Leo: So it's not how they do the fingerprints that's a problem.

Steve: Correct.

Leo: It's just bad software.

Steve: Yes. I mean, it's just like, okay, here again, our listeners that have been educated now...

Leo: In the registry [laughing].

Steve: I know. Apparently it's not quite in plaintext. But it's reversed, or the case is changed or something. It's something so trivial, I haven't bothered to drill down and figure out exactly what it was. But what I have seen from the bit of research I did is that, if you knew a password, you could see what they were doing, and then you would know how to do it to all the other passwords on all of these 16 laptops that were using fingerprint authentication.

So it's a classic instance of what we've often talked about, is that security is a chain of links. And it's the strength of the entire chain which yields your security. And so by definition the weakest link in the chain limits the security of all the other - the effective security of all the other links. So here we've got, oh, high-tech, I've seen this in the movies, fingerprint recognition. Yet the thing stores the password virtually in the clear with no useful security at all. And the problem is that, for example, many people rely on the encrypted file system, EFS, which is very well designed. It is effectively unbreakable unless you use UPEK's biometric fingerprint login to access it, in which case it's pretty much in the clear.

Leo: It just shows you how you can assume that you're being more secure, adding biometrics, and you're not. You're being less secure.

Steve: Right.

Leo: By the way, that's not two-factor if you don't enter the password. It's single-factor still. It's just a different factor. So if you use a fingerprint reader to log in, it's not more secure because it's a fingerprint reader.

Steve: Right. Well, it is - correct. Instead of adding something you have, meaning your pinky, and sort of adding that to something you know, it has exchanged something you have for something you know.

Leo: Right. It's still single-factor.

Steve: So you don't need to know anything anymore.

Leo: Yeah, just single-factor. And there's been all sorts of issues that I've heard of with fingerprints anyway.

Steve: Okay. Now, Leo, you and I need to - we may disagree on this. Which is fine because it's arguably controversial. The cofounder of Apache...

Leo: Oh, I know what you're going to talk about. I know where you're going.

Steve: ...has decided that the most recent committed build and henceforth of the very popular Apache web server will ignore the Do Not Track, the DNT header, if it is part of a query coming from Internet Explorer v10.

Leo: And we knew this would happen, by the way. We talked about this.

Steve: Well, we - okay. I didn't know this was going to happen.

Leo: Oh, no, no, I think we talked about it. In fact - maybe it wasn't on this show. I thought it was on this show because, well, go ahead. I'll let you describe it and...

Steve: Well, so just to rewind for our listeners, Microsoft made the decision that with IE10 they're going to push Do Not Track maybe a little further than, well, certainly further than anyone else has so far. And that is, part of the installation of it will have it defaulting on. And in the express install there is an opportunity for you to say no, but...

Leo: No, it's the other way around. So the way IE10 is going to work - so there's three states for Do Not Track.

Steve: Correct.

Leo: Which is unset, which is no preference expressed, on, or off. There are three things. If you go through the normal configuration of IE, it will not turn it on without telling you. It's only if you use the express configuration. So the express configuration of IE10, which most people pick - remember when you install IE10 it says, would you like to do a custom, or do you want to just accept our express settings? If you accept the express settings, then it turns Do Not Track on.

Steve: Okay. And nowhere do you see that it's doing that.

Leo: Right. That's the express settings. Now, if you go through the normal setting process you will be asked.

Steve: Right. So the...

Leo: And most people use express, including me.

Steve: This guy, the cofounder of Apache, is also on the committee that has worked on standardizing Do Not Track.

Leo: That's the W3C.

Steve: Yes. So he's also got - but, I mean, specifically the DNT subgroup or whatever. So he's been involved in this and is adamant about the idea that there ought to be these three states: no specification has been made, either the user has said they do want

tracking, or they don't. And anyway, so what Apache has taken it on themselves to do, because now they're claiming, and this we have talked about before, that IE10 is not standards-compliant - not that Internet Explorer really ever has been for other things - IE10 is not standards-compliant in complying with the letter of how the Do Not Track should be implemented, meaning that it needs to be, by definition, a clear user choice. He's taking the position that, because Microsoft sets it for users who choose the express install, they have not asserted a choice. Therefore IE10 does not abide by the standard.

And what is the controversial aspect of this is that the web server itself, not the apps running on the server, but the server will remove that on the way in so that it is not available to applications. And so it's a little - that, I mean, I guess that's the problem I have is that what if applications assume that Do Not Track is working, and that they'll at least be given the headers that the client has provided. Apache is proactively filtering the request headers, removing it, I mean, and altering them from what the client has sent. So I don't know if this is a little power struggle between Apache and Microsoft, if they hope to force IE to change their behavior...

Leo: I think it's more nuanced. I would interpret his decision in a more nuanced way. In fact, he says in the patch, the quote is - the patch's title: "Apache does not tolerate deliberate abuse of open standards." So I'm not sure he's necessarily expressing an opinion on DNT. But he's expressing an opinion that there is an open standard that was set in a normal open standard process.

Steve: Process.

Leo: And that Internet Explorer 10 has chosen to ignore an open standard, and so we're going to ignore the setting. And it may be more nuanced. It may really be about open standards and saying Microsoft's got to - there's a standard. Now, you could dispute the standard, and there was an opportunity to do so at the W3C level. But having set the standard, Microsoft either has to adhere to it, or we have to ignore their choice. Now, he may also be saying something else. But I think that that title tells you that what he's saying is you've got to - the standard's set, so you need to adhere to it. It has the net effect, though, of somebody, whoever uses IE, even if they decide to actively turn off tracking, it's going to be ignored.

Steve: Yeah.

Leo: Which I think is not the effect that's - it's very much like dimpled chads.

Steve: [Laughing]

Leo: It's attempting to understand the intent of the user; right?

Steve: Yes. It's too bad that there isn't an intermediate between zero and one for the DNT value, which means the browser turned this on on behalf of the user. I mean, like, too bad there's...

Leo: Yeah, well, the browser should be configured so that you have to make the choice one way or the other. Explicitly. The problem, I think the problem they have is the browser doesn't tell you explicitly that you're making that choice.

Steve: Yeah. And of course the reality is probably users would choose it if they knew it was available. But users don't choose things.

Leo: Right.

Steve: They just, as I...

Leo: It might, believe it or not, have underlying - he might be on the side of the angels, and I'll tell you why. Because if advertisers decide that the Do Not Track is being set without the choice of the user, but just by default...

Steve: Then they're justified in ignoring it, too.

Leo: Right. So I think he may - I'm giving him a very positive spin on this. But I'm just saying, you could say - he said, look, at the World Wide Web Consortium we discussed this. And they did, I'm sure, endlessly, because they changed their mind once. And we've decided that, if this is going to have any merit, advertisers cannot in any way say, no, no, the user didn't have a choice in the matter. It has to be - we have to be able to say to advertisers, no, look, a clear choice has been expressed. Our standard says so. So you must honor it. And I think that he may be defending that. He may be protecting it. But certainly he's protecting open standards.

Steve: I can certainly see that.

Leo: In other words, it's complicated. And whether - and we're taking out our disagreement over tracking cookies completely from this. That's not it at all.

Steve: Yeah, it's not about that. And what this really says is don't use IE10.

Leo: Well, if you want to use Do Not Track, don't use IE10.

Steve: Yeah, until Microsoft - and which is where him putting pressure on Microsoft comes in. So, I mean, and Microsoft has to have thought about this a lot. They keep touching this third rail with advertisers every time. 8 was going to have it, then it got pulled back. 9 was going to have it, then it got pulled back. Actually 7 was going to have it and it got pulled back. But now...

Leo: And really what Microsoft's doing is just a PR move to users. Look, we protect you. But the effect of what they're doing, by ignoring the open standard, is they may in fact be doing the reverse. And, by the way, Apple does this. And this was the whole thing that Google went around on Safari. Apple did turn on Do Not Track by default. So it looks good for a company to say, look, we're...

Steve: No, it was - I think Apple was blocking third-party cookies by default.

Leo: That's it. Same idea, though; right?

Steve: Yes, right.

Leo: That doesn't have a standard one way or the other. DNT is something new. In response to third-party cookies. It's really - and I'll dial out on the, I mean, I would love to see - you know what I'd like to see? Every browser - and, by the way, Chrome doesn't have a setting at all because Google is never going to turn on Do Not Track. But what I would like is everybody - Chrome, IE10, everybody - in the setup process to have a little description of what tracking cookies are, what they're used for, and then say, "And you must choose now." That may violate the standard, too, because the standard allows for three settings: unset, on, or off.

Steve: I really - I think that what I saw Rev. 3 do, I can see that that's going to happen.

Leo: What's that?

Steve: Where Rev3 notices that you've got this blocked, and they say, you know, we're happy to give you all this content, but we need you to take a look at all of the content. In this case it was advertising blocking. And they said, you know, just turn this on. So it's like, okay.

Leo: Yeah. Or I think we may win this one, and advertising companies may just say, fine, we don't need to track you. Because there's not a lot of proof that tracking works.

Steve: That's absolutely right.

Leo: Because - and at least, as used as an ad preference, ad customization preference, ad customization doesn't seem to work at all.

Steve: Yeah, that's so perverse. That's what's so perverse about this is they build all these huge databases that are oftentimes de-anonymized, but it's not clear that offering per-user ads, I mean, for those who even notice it, it creeps them out.

Leo: Right.

Steve: I mean, it freaks out users. It's like, wait a minute. I'm over on this site. How do they know I was just over there looking at that? I don't want anyone to know that.

Leo: We're really in the early days of recommendation engines. Just look at your ads on Facebook, which are absolutely, by the way, every ad on Facebook, you are tracked on Facebook. Whatever the setting that you want to set is, you're on a single site, so there's not third-party cookies. And the ads, personalized ads on Facebook are worthless. They're stupid. I mean, just look at the ads you're being fed on Facebook. And that's the best possible case of tracking cookies.

Steve: I was just going to say, look how much information they have and the result of that, yeah.

Leo: So I think advertisers might just say this is the third rail. I mean...

Steve: Nice while it lasted, but...

Leo: I'm looking here, I've got an ad for T-Mobile, Chevron, giving away money to public schools, San Francisco '49ers store, make money with your photos, sign an anniversary card for Barack Obama, and cut my electric bills by 75 percent, and State Farm Insurance. I don't think that that's particularly tied to anything. But these are all customized heavily. It even says my friends Brett and Jessie and Robert like T-Mobile. That, if I were Jessie, Robert, and Brett, I'd be pissed.

Steve: Yeah, that's leakage.

Leo: That's real leakage.

Steve: Yeah.

Leo: And that's, when you do a "Like" on Facebook, then your name may be used with your friends to say, hey, Leo likes this. Why don't you? That's annoying.

Steve: We ought to just change that from "Like" to "Leak Me."

Leo: Leak Me. Leak. Leak. Anyway, I'm glad you brought that up because I thought that was a - we tried to talk about it on TWiT, and I think you did a better job.

Steve: Well, and you were all up to speed for it, too, so that was really good.

Leo: Yeah, I am up to speed on this one, boy. [Whistling]

Steve: Okay. So Kindle's new Paperwhite screen. I'm excited.

Leo: Yeah. I want to see it.

Steve: It looks great. And for some reason I feel like I have, and of course I haven't because I don't have a time machine. Or, if I did, I couldn't tell you. But the way the screen is illuminated is wonderful. And I don't know why I know that. But that's what - maybe I was dreaming it.

Leo: You have faith. You believe.

Steve: Sometimes I have vivid dreams.

Leo: You saw the ad.

Steve: Maybe I just dreamt it. Okay. But here's my gripe. You got that link in the show notes, Leo, the g-ecx.images-amazon.com link?

Leo: Yes.

Steve: They posted what I think is a very deceptive graph. And that just bugs me because it's like, okay, this is not necessary. They show tiny bars for laptop, for the battery life of laptops and smartphones and obviously competing pads or readers and things, showing, like, nine to ten hours.

Leo: Eight hours, nine hours, ten hours, yeah, yeah.

Steve: And then this monster bar of theirs that, like, you have to scroll your browser in order to see where it ends, that says eight weeks. And then underneath in fine print it says "Based on 30 minutes per day of Kindle use." Oh, at a setting of 10. Well, now, the brightness is 24 levels.

Leo: Oh, so it's half brightness. Ooh.

Steve: Less than half. It's only 42 percent of max brightness. And eight weeks at 30 minutes per [day] is 28 hours. Which is only twice as many hours as the times they show for the competing devices.

Leo: They're mixing apples and oranges.

Steve: The bar goes off to New Jersey.

Leo: Yeah.

Steve: So, okay, Jeff, clean up your act here. This is not necessary.

Leo: Eight weeks sounds good, though, doesn't it. [Laughter] If you only used your tablet 30 minutes a day, it would last how many days? It would last 20 days; right? 10 hours? So that's what they should really be showing. 20 days, eight weeks.

Steve: Yeah. If you use your tablet for the same amount, then it'd be this long compared to that long.

Leo: They're mixing apples and oranges.

Steve: I did get a nice note - because I tweeted this. I was just - this just really infuriated me because, I mean, I'm - lord knows I am a Kindle fanboy. There's no doubt about it. I've got every model and make that there is, and I love the Kindle, and I've already got two of these on or der, one for Jen and one for myself. Anyway, Chad, tweeting as @laurion in Framingham, Massachusetts, he said, "Fortunately, the increased contrast and resolution should mean needing the light less than a typical Kindle. It's not always on." And that's a great point is that - although it's got this wonderful ghost-y glow look to it. And again, I don't know why I know that.

Leo: Let's see. It's only 25 percent more contrast. So I don't think it...

Steve: I'm sure if you make it look white on the...

Leo: Paperwhite.

Steve: I hope they didn't mess that up on the web page, either.

Leo: It's grey-green.

Steve: Yeah, although much higher - I'm excited about more pixels because I'm always - I'm Mr. Pixel. I like...

Leo: And it's, I mean, you know, look, there's going to be incremental improvement in this stuff always. The thing they should probably point out is the reason you're only using it 30 minutes a day is because it doesn't do as much as the tablet, the laptop, and the other stuff. All you can do is read.

Steve: Okay. That's true.

Leo: Yeah. You'll use it less, so it will last longer.

Steve: Okay. Actually, you could just, like, glue it to the back of your tablet so you just flip the thing over, and then you've got a Kindle.

Leo: That's what Kenny did.

Steve: Okay. So, finally, I just wanted - I did want to bring - I tweeted this. Anyone who's curious, with a touch device - a tablet, an Android, an iPhone - something very cool. You and I gripe, our listeners have heard me say how annoying it is when I go to a website, and the site wants you to download their own app. I know they're trying to get mindshare, and you have now their icon on your icon array and home screen or whatever. But it's just, it's like, okay, no, I don't want your app. I want to - give me a good experience with my web browser.

And so in researching the story about the discovery of how the UDIDs really did leak out, that is, where David did this, I went to IntrepidUSGroup.com/insight. So that's IntrepidUSGroup.com/insight; or Twitter.com/SGgrc, look at my feed because the link is there. And what I was presented with was what looked like an iPad app, where you can move between articles, you can see summaries, I mean, a complete sort of fluid touch custom app. And it took me a minute to realize I hadn't been taken to an app. There were still browser tabs at the top of the iPad screen. I'm like, wow. This is JavaScript. They've written an app in JavaScript.

Well, it turns out it's from a company called OnSwipe.com. And if you go to their site with a pad, then it's also a nice experience. And it's free. So I just wanted to give a shout-out to these guys. This is beautiful touch tablet technology that allows companies to pour their content into this framework which is free and not require people to download an app, which there's certainly some resistance to doing. I'm not the only person who feels this way. Certainly you do, Leo. And again, it's like, that makes you do something else. Here I got a very nice browsing experience without having to download another app. So I thought it was just very, very cool.

Leo: HTML5, baby.

Steve: I know.

Leo: You know.

Steve: And I got one nice note also from James Lewis, who said, "SpinRite got me a free TiVo." And he said, "I was skeptical SpinRite even worked. I figured it was worth giving a shot for 89 bucks. I've been using my Series 1 TiVo for years. My friend had one of those fancy Humax Series 2 with DVD burners."

Leo: Those were good, yeah.

Steve: "One day" - yes, yes. He said, "One day it wouldn't reboot for him, so he gave it to me for free. SpinRite spent a couple of hours on the drive, and when it was done I had a Series 2 TiVo. Thank you." So, yeah, SpinRite fixes everything.

Leo: We've heard other people talk about it on the TiVo drives.

Steve: Yeah, yeah.

Leo: Because it doesn't care about file system. It doesn't know or care.

Steve: No.

Leo: I think TiVo uses Linux ext2 filesystem. But it doesn't matter.

Steve: It does. And, in fact, it's saved my own Series 1s many times. And they use a PowerPC chip that is a big endian rather than a little endian chip.

Leo: That's right.

Steve: Yup. So the bytes are even in the wrong order, and SpinRite says, eh, I don't care what's there.

Leo: It just fixes it. Before we get to our topic, which is about online identity and authentication - and I'm glad you're going to do this, which is kind of everything you need to know about OAUTH, OATH, as we talked about last week. Steve Gibson here, a little bit of a different time because we flip-flopped, as we often do. Apple does its events on Wednesday these days, which means Steve gets to move to Tuesday, MacBreak Weekly to Wednesday. We will be doing live coverage of the Apple iPhone 5 event starting 10:00 a.m. Pacific, 1:00 p.m. Eastern tomorrow, 1700 UTC. And we've decided, and this could either be great or it could go very, very wrong, that since Apple doesn't provide a stream for us, we're going to reenact the keynote with puppets.

Steve: You're not.

Leo: Yeah.

Steve: No.

Leo: Yeah. You think I'm joking. No, I commissioned Brian Hogg, puppet master, who did my Leo puppet, which you see behind me, he's doing a - he did Walt Mosspuppet, which was quite a character for a long time. He's doing a Tim Cook puppet for us. We didn't have time to do everybody that could conceivably be onstage because there's a lot of people. Could be Tim Cook, Phil Schiller, Scott Forstall. So Tim will be everybody. And he will be reading the live blog back. Not the whole time, but when there's something - and we have the images from the live blogs. And so we have a little stage. We're going to put him on a little stage. It'll be great. Puppet show tomorrow.

Steve: Why does Apple not just make a stream available?

Leo: We've been discussing that. We talked about it on TWiT.

Steve: I know.

Leo: One network engineer in our audience said, you know, the cost of making something super reliable that would be watched by that many people simultaneously is probably prohibitive.

Steve: It's too big.

Leo: It's too big.

Steve: Yeah.

Leo: And there are other things. Apparently they do have engineers sitting there doing secret sauce during it because, I was told by somebody in the know, because they turn around the keynote almost minutes after it ends and put it online.

Steve: Have the video up, right.

Leo: So they're doing some stuff. And maybe someday it'll be ready for primetime. They've done in the past, like unaccountably, like a year ago, they did one live, and then never again. I would think they'd want to do it live. There's such interest.

Steve: God, yes.

Leo: But anyway, puppets might work.

Steve: If not, Leo, if they see your puppet performance...

Leo: Oh, I gave up on Apple ever...

Steve: That may convince them to go back to a live stream.

Leo: To go live.

Steve: It's like, either Leo's going to do the puppet, or we're going to stream.

Leo: It is our - you know, it's funny, we do a lot of live events. We do a lot of stuff. Obviously we do 40 hours of programming a week. There's nothing like an Apple event for driving traffic. There really is interest. And I was talking to Ryan Block of Gdgt.com. They do a live blog of it, and he said, oh, yeah, easily our biggest. He said AOL burned up a server when they were doing it.

Steve: Oh, that's too cool.

Leo: They actually burned one up. It broke.

Steve: A meltdown.

Leo: Which I guess you can have - a switch, if you have too much traffic going through it, can melt.

Steve: Okay. Maybe it's made out of chocolate.

Leo: It was a chocolate switch.

Steve: Oh, goodness.

Leo: Anyway.

Steve: Okay, now, and why - okay. So you're calling it the iPhone 5 event. I mean, is it just a bigger screen?

Leo: Well, we don't know. But...

Steve: How can it be any better?

Leo: So the invitation has a "12" for the day; right? And then the shadow the 12 casts is "5." Now, it's not the fifth iPhone. There have already been five. The 4S is the fifth iPhone. So it's really the sixth iPhone. But we think they must, maybe, well, some - and then a number of people tweeted me, well, what if instead of being about the iPhone 5, they're saying there are going to be five new products we're going to announce? I don't think so. I think they're going to call it the iPhone 5. And it will be 1136x640, one extra row of icons. There's quite a bit of debate over where it will have LTE or...

Steve: Less scrolling of your home screen.

Leo: Right, yeah, more on your home screen. And it will be 16:9, so it'll look better for movies and games.

Steve: Oh, cool.

Leo: Yeah. I don't, you know, I think that it's gotten pretty competitive now. I don't think Apple's got a free pass anymore. Maybe they do.

Steve: And we do think that there's going to then be, like in October, a mid-size pad?

Leo: "We" being John Gruber and John Paczkowski and Jim Dalrymple, you know, the pundits who seem to have a handle on all this, yes, they believe a mini iPad. Now, that will be interesting, especially after the Kindle Fire HD. I bet you ordered one of those, too.

Steve: I didn't.

Leo: You didn't.

Steve: No. I'm going to wait and see.

Leo: You're not interested.

Steve: The Kindle Fire burned me, if you'll pardon my choice of words, so badly, it's just so horrible, that I thought, uh, you lost me on this one, Amazon.

Leo: Interesting.

Steve: You sold me a piece of crap, and I'm not buying a second one.

Leo: Now, the Nexus 7, same thing, $200, great.

Steve: Oh, my god, I love it. And I have said to everybody, $199, this is a completely workable, worthwhile pad.

Leo: Right. Well, I had to order the Fires. Not that I, you know, I might kind of agree with you. My Kindle Fire, the original's just sitting there, not doing anything. But I looked at it the other day, it's on my dresser, and it hasn't charged in months. And I thought, well, I'll bring it in for the event. And I guess I'll Gazelle it after that. But I am going to get the 7", and we'll get that in two days, I think, the 14th, three days.

Steve: Oh, no kidding. So it is available soon.

Leo: Yeah. That one is the first thing to come out. So we'll have that, I'll have that by next Security Now!. I'll let you know.

Steve: I ordered a pair of the Paperwhites as soon as I saw that it was up, and I have an arrival date of, like, mid-October, I think October 17th.

Leo: Yeah, they're not shipping those right away, yeah.

Steve: But I hope they have a lovely glow.

Leo: They will glow. I'd be very - I'm really - so, see, I didn't order a Paperwhite. So I'll let you tell me how they are.

Steve: Oh, perfect.

Leo: And I'll tell you how the HD Fire is.

Steve: Deal.

Leo: We have to work these things out ahead of time. Our show today...

Steve: You're going to have it for next week's podcast; right?

Leo: I will.

Steve: Okay.

Leo: If all goes well. Will I have an iPhone 5 by then? They're 12th, 19th, no, it comes out on Friday, a week from Friday.

Steve: There is a huge staffing effort, isn't there?

Leo: Yeah, but it won't be till two days after we do the show. So Friday the 21st is when it comes out. Sorry. Meanwhile, you know, I buy everything, and this is the first time I'm kind of thinking, I wish I didn't have to buy everything. I just want to stay with one phone for a few months, please, I beg of you.

Steve: And it's easy to fall behind when this much is happening.

Leo: I have never seen a more fecund fall. I'm going to call this the "fecund fall." Fertile with gadgets. They're blossoming everywhere. And there's Nokia phones, there's Motorola phones, there's tablets, there's Windows 8 stuff is coming out any minute now. Oh, my gosh. And I've got to buy it all.

Steve: Bezos is claiming that one of their things, I guess it's the Kindle HD, is better than the iPad.

Leo: Well, it is in many respects. Its DPI is lower, but very close. It's got two speakers with Dolby sound, which the iPad, the new iPad has terrible sound. It's got one little tinny speaker. I guess they figure - and in fact most games now say, "Please wear headphones for a better experience." I don't know in what other respects it's better. It might be faster. We'll see. I don't think it is. The software is dumbed down, it's dumbed-down Android. It's not a content creation device, it's a content consumption device. It's a door to the store. Exit through the gift shop. All right, Steve, let's talk about authentication. I guess...

Steve: Okay. Yeah. Obviously it's crucial.

Leo: Yeah.

Steve: We have looped around it, coming at it from all different directions, because there are so many different facets to the issue. One of the things that I was thinking about after we were talking last week was remember someone, we did a Q&A, and someone was suggesting maybe the role of the post office as an authenticator of identity.

And that reminded me that there's two very different classes of identity, which is worth noting because it's important. And that is, there's the notion of real-world identity, that is, who you actually are in the physical world, which is entirely separate or is at least entirely separable from your online identity, or you might call it a "virtual" identity. And there you still want to authenticate, but you want to authenticate, at least optionally, anonymously.

And then it's like, okay, well, wait a minute. Why would you care about anonymous authentication? And the reason is reputation. And "reputation" is just a wonderful word in this online context, the notion of you're investing in an online virtual personality. And what you are doing over time is acquiring, earning, building a reputation among the people you're sharing this community with. And so for that reason the strength of your reputation is directly tied to the community's understanding that it's unlikely you could be impersonated.

And so once again, even though in that entire universe we're disconnected from what's your street address, what's your Social Security number, what identity would the post office see if you walked in and showed them your driver's license, it's not about that. It's about the integrity of the system's, the Internet's ability to provide authentication of a virtual person. So it's just this is who I say I am, and nobody else is able to say that they are me, and over time that builds value. So that was something we hadn't really ever explicitly discussed, so I wanted to do that.

But something happened last week that surprised me. And I thought, oh, this is further along than I thought. I brought up IMDB, Internet Movie Database, which I poke around from time to time. And this was the app on my iPad. Same experience under Firefox, for example, in Windows. And it prompted me to log in with IMDB, Amazon, Facebook, or Google. And I thought, Amazon? What? And sure enough, if I click on Amazon, I jump over to Amazon, then LastPass sees that I'm being prompted to log into Amazon, it does that for me, and I'm back to IMDB, having logged - and it knows my name. It says, "Hi there, Steve Gibson." It's like, oh, okay. I didn't know we were here yet.

We've talked about, we've seen and discussed the "Log in using Facebook," "Log in using Twitter." We know that Google is active in this area; and, of course, the Google Authenticator we've talked about. I was unaware that Amazon was offering that service, and I'm delighted because I'm an active Amazon user. I'm not an active Facebook user. Actually I use Amazon more than Facebook, Google, or IMDB or anything else. So that was cool. And of course Amazon is now also offering multifactor authentication. So if I were using that, then I would have been prompted for, if I had set it up, for another factor of authentication.

So what's happening is there are groups working to standardize these protocols. And it comes down to three acronyms that are essentially - two of those are protocols, another one is sort of crypto technology. And that's OATH, which you helped me produce, or pronounce, because I was saying, wait a minute, O-A-T-H and O-A-U-T-H. And of course OATH, obviously, is how you pronounce OATH. And actually it's kind of a good acronym for that.

What OATH is, is just - it's not a communications protocol or an authentication Internet protocol. It's just the crypto standards that, for example, Google Authenticator is based on. And what's significant is that, for example, the VeriSign VIP service that we have talked about extensively in the past, here's my little PayPal football that I got, I think was it free or $5 or something? I think it was very inexpensive from PayPal, which I'm still using. And one of our listeners famously realized that the upper digit of the six-digit code is incrementing linearly, not part of the pseudorandom sequence driven by the

secret key.

Well, Google Authenticator doesn't work that way. Neither does the Windows Phone 7 Authenticator, which by the way is, I have verified since, Google Authenticator compatible. So how is it possible that the PayPal football, which is tied into the VeriSign VIP system, uses a different protocol? Well, and that brings us to another critical or sort of key aspect to this which is the VIP system, VeriSign's, is closed. It is not an open system. You get credit cards from them that are time or event based - in this case they are event based, so they don't have a clock running all the time or the battery wouldn't last long - using the little eInk printing that we talked about a long time ago; or you get the football from them.

But they're not publishing, and their technology is not open. So they know what the key is associated with my token, and they can have any algorithm they want to. We now know that, since the first digit is linearly incrementing, it is a weaker authentication than if it were six fully pseudorandom digits. And we understand that they're doing it because it's time based, and having that most significant digit changing uniformly allows them to better lock onto and deal with time drift.

But the central factor here is the notion of a closed versus an open system. One of the reasons I am so bullish about Google and their effort to push this open, this OATH open standard is that, when I inquired of someone who was using the VeriSign VIP system, I was a little taken aback by how expensive it is. And while I know that they've got lots of customers and their enterprise class and so forth, to me this feels like a problem that ought to have a free solution. Which is what Google is helping to push with the success of the Authenticator, which is based on these open standards, so it's not a closed system. But there's a downfall.

Leo: Can I ask you one question about the Authenticator?

Steve: Yeah.

Leo: Because a listener to the radio show called this week. He said, "I'm nervous about the Authenticator. Can it be reverse engineered?" And it would be the same question about the VeriSign dongle.

Steve: Okay. So the Authenticator, the OATH protocol, I'm sorry, well, the OATH, I don't want to use the word "protocol" because...

Leo: It's not a protocol.

Steve: ...OAUTH and OpenID are protocols.

Leo: Right.

Steve: The OATH standard is a hash-based standard where it takes either an event counter or a time-of-day clock. And in the case of a time-of-day clock, essentially it

creates an event every 30 seconds tied to Internet time. One of the things I loved as I was reading through the standard was they make a point of saying, "And this must be 64-bit Internet time."

Leo: High resolution.

Steve: Well, the reason is the Internet time that has been established, the so-called UNIX time, same sort of thing, it's 32 bits, and it wraps around in 2038.

Leo: Right.

Steve: So it's like, whoops, we go back to the beginning of time.

Leo: This won't wrap around till the beginning of the actual end of time.

Steve: Correct.

Leo: 64 bits, [indiscernible]. So it's taking, okay, so it's taking the Internet time at 64 bits. Is it hashing it with something?

Steve: Yes. It uses a keyed hash. So, and I'm thinking that we ought to talk about that in detail. I don't want to get into the crypto today.

Leo: No. I speculated, though, that it was a one-way hash.

Steve: Yes. It is a secure keyed hash.

Leo: I liken it to a meat grinder where you can put a steak in and get hamburger out, but you can't put hamburger in and get a steak out. It's one way.

Steve: And the other thing is the hash is large. It's like SHA-1 or SHA-256. Now, one of my annoyances with OATH, this standard, is it doesn't specify this tightly enough. That is, it says, oh, well, you could use SHA-1. You could use SHA-256. You could do this; you could do that. It's like, no, no, no, no. Please don't, don't give people the choice because they will take different choices, and we will have nonstandard implementations.

So it's another reason why I'm glad some entity as large as Google has said here's ours. And here it is at code.google.com. It's open. Please, take it and use it. And so it's available for free download for our devices. It is being copied, which is fine, but it's being compatibly copied. And so it behooves people to do this, to implement exactly the same set of choices that Google chose out of the OATH standard. So while the standard is not nearly as rigid as I would like, I mean, for example, back in the early days of TCP there was a lot of things that were not closely enough specified, and people implemented

incompatible TCP stacks. So you had an interoperability problem in the beginning. And in fact TCP is complicated enough that a lot of the way it works has sort of gone into sort of mythology, almost. It's like, oh, just take one that works. Don't go reinvent that because, even if you follow the standard, you may run across some edge cases where you'll have problems.

So knowledge about actual implementation has now been embedded in the code, and it's more a spec than the standard that is written. And so we have the same sort of effect with the Google Authenticator, where it's like, this is Google's implementation. Copy it. Do exactly this, and you will get something right. So but the point is the width of the hash is much larger than the digits we see. So we don't even see all of the hash output. Which is good for security. If we saw all of it, it'd be an insanely long thing that no one could type in in 30 seconds before it changed, and that would be a problem.

But also, if we only see part of it, no cryptographer, even if they wanted to do, like, brute-force one-way functions, because we know that hashes can be brute-forced with rainbow tables and by, like, putting every possible value into the front and see what comes - or into your meat grinder, Leo, and see what comes out. And so you just keep putting enough things in until you get a match. The problem is, or, well, the problem for the cryptographer or the cracker/hacker is that all we're seeing is a tiny piece of the hash output. So to do any kind of meaningful crypto you've got to see it all. And what that - so what that practically means is that we have a hash function which is doing a very good job of producing pseudo, really high-quality pseudorandom numbers. And, for example, it is possible that when the 30 seconds elapses and the next number is shown, that it doesn't change.

Leo: What?

Steve: Yeah. Because...

Leo: That's highly unlikely, though, isn't it?

Steve: Very unlikely, but very possible.

Leo: But possible.

Steve: Because we're only seeing six digits of a much larger hash. And so any of the other high end of a hash would be different except those six digits. So we know that six digits is one out of a million. So there's a one in a million chance that you will get the same code again because it's truly very good random. And that's what you would like.

So here's the danger, and your question is the perfect setup for this. The problem with having your phone full of accounts - remember I was excited that Google Authenticator essentially allowed me just to create as many individual instances for authentication as I wanted. For example, we were talking about Dropbox and how Dropbox now supports multifactor authentication using Google authenticator. And that's cool. Except that Dropbox knows the key for this instance. And as long as I only use the key with Dropbox, well, then I have some security. But if I deliberately gave other authenticators the same key because I wanted to reuse the instance of Google Authenticator, that is, that account

in Google Authenticator, now we start having the same problem again as using a common password across multiple sites.

So the point is that we have a nice system with Google Authenticator and its clones because there are now a bunch of people doing a compatible solution, and I think that's great. But in order for this to be secure, because essentially we're able to manage our identities by creating instances of these authentication accounts, we again bear some responsibility in managing those well.

Now, to make this more clear, I don't have that problem if I were using VeriSign with VIP because theirs is not a distributed system. Theirs is everybody I authenticate with has to go to VeriSign to determine what my code should be, rather than just asking me, and my phone knows. So these are different models. And it's not clear to me, I mean, it's not completely without cost that we remove any middleman from the solution and take responsibility ourselves. And remember that the downside of the VeriSign approach is what we saw with RSA when they had the huge breach because RSA was the same thing. They had all the keys to the kingdom, all of their time-based RSA tokens that corporations the world over and a lot of government were using for, for example, logging in securely to their VPNs. They lost the keys.

So there is a, if you centralize like that, there's a single point of failure. If they were down like, for example, GoDaddy was down yesterday, no one can authenticate during that period of time using their approach. So you would want to have some sort of backup solution. And also there's a single point of vulnerability to them losing the keys to the kingdom as they actually did in the case of that breach.

So where we are today, I think, is we're moving towards a - thanks to probably the strength of Google and the pressure from users who want something better than username and password, where in fact username is email address more often than not, so that doesn't even count, as we were talking about last week, where we have a distressingly loose standard in OATH which has been adopted and is now being copied. And so in the same way as other standards that were not well specified but became well specified in practice, we've got that. And that's beginning to happen. Now, so that's OATH.

Then in terms of protocol there's OAUTH and OpenID. Now, we talked about OpenID years ago when that effort was relatively new. And their site claims, I think it's nine billion websites are now using - no, it has to be million, nine million websites. But it's like it's, I think it's one billion users and nine million websites. Is that possible? That sounds like a...

Leo: Yeah, that's possible. There's a billion Facebook users.

Steve: Okay.

Leo: Yeah.

Steve: And so I don't encounter OpenID that often. I mean, I've seen it. The idea is it's got kind of that little funky sort of 3D, it's like gray and orange sort of thing standing up. It's kind of a funky little logo.

Leo: Oh, the logo, okay, yeah, yeah.

Steve: Because that's the visual cue that you can use your OpenID identity to log into this site. And so, and you'll remember that the concept was you give them a domain name or a URL for a page on the 'Net which you control. And so the idea was, in the same way that an email confirmation loop is useful because you control that email account, and so you've said this is my email account when you set something up, and so you hold onto that, and so the idea is you know how to log onto that email account, you control it. Similarly, the OpenID concept was you control a page that is available. Now, it may be your blog. It may be your own website. Or more often it's going to be a provider, an identity provider where you have established an OpenID account. And then they've given you this string which essentially is your page there.

Leo: Yeah, I used a company called claimID for this purpose for a long time.

Steve: Yes. Yes, exactly. And so the idea is that that's your universal tag. And if you are at any of these sites that says, oh, you can authenticate, you can identify yourself, essentially, with your OpenID, if you've got one, you use it there, and off you go. Now, that's competing with OAUTH. And I think it's lost. I think it...

Leo: [Sighing]

Steve: I think it was a nice idea. But there is so little tolerance for any friction in authentication that the idea that I could go to IMDB and am presented with how would you like to log on, with us, Amazon, Facebook, or Google?

Leo: Yeah, that's the problem, or Twitter. That's the problem. It's so much easier, and everybody's got one of those.

Steve: Yes, yes. And so there isn't a way to compete with that. I mean, there just isn't. Now, the good news is that - so what we have is we've got - it could not be easier than that. I mean, it just - as you said, Leo, because everyone has at least one of those, they don't have to go create an account on IMDB. And so I think we're going to see now OAUTH adoption take off. As people get - they see this more often, they encounter it more often, they see that it worked for them, oh, look, I can just log on using my Facebook.

Leo: So that's what Facebook's using. That's what - I know this is what Twitter's using. Google+.

Steve: Yes, everybody. The way to differentiate it is, if you are not - the OpenID has to get information from you first. It has to ask you for your ID. And then it goes off and authenticates you against that. Anything that says "Log on using Facebook, Twitter, Amazon, Google," whatever, that's OAUTH. And we do have a podcast in the past about it in detail [SN-266]. Essentially the way to think about it is it uses browser page

redirection on the front end. That is, when I click on "Log in using Facebook," my browser receives from the server instructions to go to a particular URL which Facebook publishes for this purpose. And that's where my browser receives a login page from Facebook, asking me to authenticate who I am, that is, to log into Facebook.

Leo: Is that going to leak information back to Facebook? That's, I guess, the privacy concern that people have.

Steve: That's a good, that's a very good point, yes. Facebook knows who has sent me to them. And so that is a potential privacy concern, you're right.

Leo: And sometimes when you click Facebook it'll ask you, besides logging in, to give that app permissions, including "Post on your behalf" and things like that.

Steve: Yes, yes. Now, OAUTH was actually more designed for that, the notion of autonomous background inter-application data exchange, the idea being, for example, you've got some web-based printer app that wants access to your Flickr account.

Leo: Right.

Steve: And so when you're using the web-based access thing, it says we'd like to be able to print your Flickr photos for you. You need to give this website permission to access that website on your behalf. And so OAUTH is more than just bouncing you around and log on using Facebook. That's an aspect of it. And then the other is this notion of persistent permissions. And in fact we can see that, one very easy place to see that is in Twitter because so many people have allowed other Twitter clients to access their Twitter accounts on their behalf, and that's OAUTH. All of this is happening by OAUTH.

If you go to Twitter.com, log in through a web page, if you drill down through security and permissions and things, there's applications that have permission to access Twitter on your behalf. And so there's an enumeration that you'll be shown of all the things you have given at some point in time permission to access your Twitter account on your behalf.

So OpenID was a nice idea. It was competing for a while. But what I'm seeing more and more is that nothing can compete with the simplicity of asking a user, I mean, when you go to a - one of the things that really annoys me is having to create an account on some site that I probably am never going to come back.

Leo: Right, never be back, yup.

Steve: Yup. And so being able to just say, oh, use my Amazon identity, fine. I mean, as I did on IMDB. It's like, wow, this is clearly the solution that's going to work. We probably need to take a look at OAUTH v2 at some point because it's been kind of lumbering along in committee mode. Some of the early founding people have gotten a little bit miffed and disgusted by just the nature of the development. It hasn't gone in the direction that they had wanted. But it really looks like it's the solution that is going to win. And I can just

see, as libraries become available, as authentication continues to strengthen, you would be able to, for example, establish strong authentication with your Google account if you tend to be Google-centric, or strong authentication with your Amazon account if, like me, you are Amazon-centric. Or with Facebook. Is Facebook multifactor yet?

Leo: Oh, yeah.

Steve: Oh, okay.

Leo: You may not know it. I don't think anybody knows it because they describe it in a very bizarre way. I should show you this because...

Steve: But are you able to use Google Authenticator?

Leo: No, you have to use Facebook's authenticator.

Steve: Okay, but they have one.

Leo: Yeah. I don't think people know it. If you go to it...

Steve: An app for smartphones?

Leo: It's the Facebook app has an authenticator built into it. So here's the deal. You have to go to...

Steve: Oh, there's a Facebook app for your phone.

Leo: ...the Facebook app itself. You go into Security, and there's a setting for - and this is why it's not clear what it's asking for. "Login approvals. Approval is required when logging in from an unrecognized device." If you turn that on, you have two choices. And I'm not going to show it because I'd have to show my cell phone. But you can have a security code texted to a phone number. Or you can use a code generator. And it turns out the code generator's built into the Facebook app. So they've got it down, actually. But I don't think they've publicized this.

Steve: Nice. So we have multifactor authentication at Amazon, multifactor authentication to Facebook, or multifactor authentication to Google. Who doesn't that cover? I mean, that's got to be the world. And so as soon as libraries become available that allow webmasters to easily incorporate OAUTH, I mean, and the libraries are there, they'll just be part of the default install, or they'll be enabled or whatever, or there'll be pressure from users, hey, look, let me log on using my Facebook account. Everybody else does. And so I think this is the direction we're going to go. And there were some glitches, there were some security problems early on that have been solved. So I don't see this as a

problematical solution that we're moving toward. To me, I think this is going to be a great step forward.

Leo: I should - I have been showing - I just showed the Facebook code generator, which is built into the app, at least on Android. I think it's on the iPhone app, as well. And then I before that showed my Google Authenticator. And people are starting to freak out, hey, wait a minute, you're showing your codes. But they change every 30 seconds.

Steve: Well, not only do they change every 30 seconds, but a proper implementation refuses the same code again.

Leo: Right.

Steve: So even somebody who was monitoring and, like, if you were logging in with it and showed it, even using the same code within the 30-second window that that code is valid, it would not be honored a second time.

Leo: And they'd have to have my password, as well; right?

Steve: Exactly.

Leo: So I feel fairly secure showing that.

Steve: Yeah. And, oh, no, I mean, I don't have it near me, but, yeah, there's just no reason not to because it is really good, pseudorandom, changes every 30 seconds, and it being used a second time is blocked.

Leo: Rorx has a good point, and I guess this is - he says, you know, it's too bad, though, that I have to be forced to use Facebook to authenticate. Is it possible to run your own OAUTH authenticator like OpenID? And the problem is that, no, because the page you're logging into has to know, oh, you're going to use Leo's authenticator, not Google, Amazon, Facebook, Twitter.

Steve: Yes.

Leo: It has to support those OAUTH implementations.

Steve: And that's a perfect example of what OpenID was giving us...

Leo: Right, yeah.

Steve: ...is that you were providing it with your own URL to represent you, and it would go there, and then you would negotiate with it. So there you had control.

Leo: I have an OpenID token at TWiT.tv. So I could, if I wanted to OpenID, I could go to TWiT, use TWiT.tv as my identifier. But nobody uses it, so.

Steve: Yeah, yeah. And so, yes, we do lose some flexibility. And then there is the issue of, as you said, some privacy concern because anyone you authenticate with knows where you're coming from.

Leo: Right.

Steve: But it's like, eh, okay, well, you know, it's way convenient. And I think it's a great solution. And it looks like it's winning.

Leo: It's winning.

Steve: Yup.

Leo: And, by the way, we allow, not that we use it, but you could use OpenID - Drupal supports OpenID. And it's possible in this case, this is the TWiT.tv site, to log in using OpenID or to sign in with Facebook. There's no point in logging into the TWiT.tv site. We don't do anything with those. Those are for administrators. But still, Drupal has that built in, so it's kind of free. But, and then in order to log in using OpenID, what do you - you provide it with your - I've forgotten now. I guess I just give it my URL.

Steve: Yeah. If, for example, if it was your own domain and web server, you could literally use TWiT.tv to represent you. But oftentimes it'll be a domain and then a specific page at that ID location.

Leo: Yeah, I have claimID.com/leo, I think, as an OpenID.

Steve: Yeah, exactly. So we're step by step, little by little, we're figuring out how to do this. We're fixing the mistakes we make as Apple learned the hard way with making their unique device ID available to developers and even promoting it for that purpose for a while until they realized, oops, that's not what we want to do. And so we're learning these lessons. And I just think that, when I imagine my mom or my sister or somebody going to a website they haven't been to before and being offered the choice of create an account, which no one gets off on creating accounts all day long, versus push this button to log in using your Facebook, they, like, oh, done. It's like, thank you.

Leo: So easy. And of course because the token's on your machine, once you've done

that, it's painless because it just knows you are who you are. Which is another security flaw because now that token could be used by others if they can get access to your machine; right?

Steve: It is over SSL.

Leo: That's what Firesheep showed us.

Steve: Exactly. Yup.

Leo: So Firesheep could be used - could it be used to do an OAUTH authentication?

Steve: Well, remember that Facebook has moved to SSL pervasive, so it's not easy.

Leo: You're okay; right.

Steve: Yup. It was not an easy transition. But, I mean, here during the podcast over the last couple years we've seen one little problem after another fixed and solved and moved. We can see our progress. We really are moving forward and getting this stuff done right.

Leo: Steve Gibson is a major proponent of all of this and probably has had a lot to do with it getting better and better and better. You can hear 369 episodes now of Security Now! on his site, GRC.com. He's got 16Kb audio and transcripts of all those episodes. We've got audio and video at TWiT.tv/sn for Security Now!. GRC is also where you'll find SpinRite, the world's best hard drive and maintenance utility, all the free programs and information Steve offers. There's a ton of great stuff. It's worth browsing around. And while you're there, if you have a question about this or any of the topics we talk about on the show, you can ask at GRC.com/feedback for that feedback form. And that's the one and only way you can ask a question for the show. Next week we'll do a Q&A episode.

Steve: And we'll have Mark Russinovich at the top of the show.

Leo: Oh, that's exciting.

Steve: Yeah.

Leo: Mark Russinovich joins us next week. For those who are watching live and saying what the heck, we flip-flopped MacBreak Weekly and Security Now! because tomorrow's the Apple event, 10:00 a.m. Pacific, 1:00 p.m. Eastern time. We'll

stream that live. But Steve will be back Wednesdays on his usual time, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 UTC for Security Now!. And I hope you'll join us live. Thanks, Steve.

Steve: Thanks, Leo.

Leo: See you next time on Security Now!.