YubiKey

Description: Steve and Leo delve into the detailed operation of the YubiKey, the coolest new secure authentication device Steve discovered at the recent RSA Security Conference. Their special guest during the episode is Stina Ehrensvrd, CEO and Founder of Yubico, who describes the history and genesis of the YubiKey, and Yubico's plans for this cool new technology.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-143.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-143-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 143 for May 8, 2008: YubiKey. This show and the entire TWiT broadcast network is brought to you by donations from listeners like you. Thanks.

This is Security Now!, Episode 142 [sic], Leo Laporte here, Steve Gibson in Irvine, joining us in our new highly technical studio that isn't really working.

Steve Gibson: With the details still coming together.

Leo: A lot of the details still coming together. Hi, Steve.

Steve: Hey, Leo, great to be back with you for our 142nd [sic] week of Security Now!. Wow.

Leo: Practically consecutive. Have we missed any weeks? No.

Steve: No, we have never missed a week.

Leo: Wow.

Steve: Yeah, you and I used to have to bunch them up when you were running around traveling or when we were together...

Leo: Isn't it nice? I don't have to go to Canada anymore, yeah. I mean, that's really simplified things considerably. Very pleased about that. So we have a guest today.

Steve: We're going to have a guest joining us by phone from Sweden. And that'll be someone I referred to two weeks ago, and maybe even last week, and that is Stina, who is the CEO and one of the founders of Yubico, the makers of the YubiKey, which I just happened to stumble on when I was up at the RSA security conference.

Leo: Boy, that was a lucky thing for both of us.

Steve: Really was. Well, and for our listeners, too. They have received hundreds of emails and inquiries from my mention of the YubiKey when I did the episode two weeks ago on the RSA security conference. And I've been in pretty much constant dialogue with Stina and the technical people that they've got. And the news is virtually 100 percent good. I mean, the more I learn about this, they've been evolving their policies - anyway, so this week's episode is the Yubico YubiKey. As we'll see when we get into it, this is an even better authentication solution than I expected it was going to be when I described it last week, or two weeks ago, as the coolest new thing I had seen at the RSA conference.

So this week's episode is Yubico's YubiKey, and I really think - I'm going to go into all the technical details after we have had a chance to speak with Stina. I asked her to come on because she really has a vision for what she would like to see happen with authentication. I wanted to understand, you know, where this wacky name came from, a little bit about the company, and just sort of get a sense for where they are because I know that when we've talked about and explained issues of authentication there's been a strong interest.

Obviously I'm a person, I mean, I'm on record here on Security Now! believing that authentication, you know, getting this problem solved is an enabling factor for the whole future of the Internet as we go from Web 2.0 to 3.0. More applications are moving onto the web. We hear about now there's, like, all this computing in the cloud where corporations are going to be moving more of their infrastructure onto the Internet as we have people who are able to carry that. So in every situation where we've got a network and you don't have your typical - I think we once described it as like the Andy of Mayberry authentication, where you know Aunt Bee, and you know Opie.

Leo: So you give Opie the drugs when he comes to the drugstore for Aunt Bee.

Steve: Exactly. And so here on the 'Net we need a good way of knowing that the person at the other end is who they say. And we've talked about VeriSign solutions and the eBay and PayPal football and the credit card. We're going to talk about something now which is completely open source, no subscription fee, lifetime free authentication. And, I mean,

it's - I'm excited because this is, as long as you've got a USB port, this is the answer. It doesn't have a display, as we talked about it before two weeks ago. It pretends to be a keyboard. But you just touch the button and it shoots out into your computer, into for example a web form, this long string of random-looking encrypted stuff that then can be authenticated, either by you or by Yubico or whomever. And the advantage is that there's no cost to anyone for all of this.

Leo: Wait a minute. Obviously there's some cost or Yubico wouldn't have a business model.

Steve: No, they want to sell the hardware. That's all they want to do. Anyway, we'll go over this.

Leo: You're giving it all away. We won't have anything left when Stina calls. So hang in there. Now, do you have any news, anything you want to do before we talk with Stina?

Steve: Oh, yeah. We definitely have some news of the week. One little disturbing bit of news was posted on Dave Jevans' blog. Remember, he's one of the main founders and president of IronKey. He posted the news, I guess it was on Friday, that Anonymizer.com was acquired by Abraxas. And the bad news is that Abraxas provides anonymity services for the national security community - NSA, CIA, DIA, and so forth.

Leo: Oh, boy.

Steve: And so, you know, I'd feel much more comfortable if Anonymizer.com had stayed independent and just themselves rather than now being part of a government contractor.

Leo: Wow. Yeah, you've got to really kind of wonder. Did you ever - have you ever heard the rumors that Facebook was partly sponsored by the CIA?

Steve: I've heard something about that.

Leo: It's a persistent rumor which has been consistently denied, as far as I know. But it's kind of credible. You would think, if you were the NSA, if you were the CIA, that kind of a great way to watch people would be to be part of these social networks. What better thing to do than buy Anonymizer?

Steve: Yeah, yeah.

Leo: But as a user you kind of have to say, well, hmm.

Steve: So, yeah. I mean, I don't know...

Leo: TOR is looking better and better, that's what I...

Steve: I don't know that anything untoward is going on, of course. But I just wanted our listeners to know, if any of them are Anonymizer customers, that Anonymizer is no longer independent. It has been acquired, and acquired by a company that does a lot of business with the three-letter-initial intelligence services of the United States.

Leo: When you said "untoward," did you mean that as a pun?

Steve: Uh, okay.

Leo: Yeah, it's a pun. UnTORed.

Steve: UnTORed, oh.

Leo: Get it?

Steve: You're at the top of your game this morning.

Leo: It's that quinti venti latte, man. You're right, those things work. So let's - okay, so that's one big story. What else is out there?

Steve: Also there was a - this is just sort of just to keep our eye on. A disturbing constant theme of the FBI has been their request for ISPs to retain data. There was a recent congressional hearing where FBI director Robert Mueller again called for federal data retention laws to force ISPs to keep records of what their customers do for two years.

Leo: He's been trying to do this for a long time.

Steve: I know. And what's really confusing is he's not saying what he wants kept. Now, the weakest information that would be kept would probably be at least the IP addresses that customers have had over that period of time, which frankly would not be that burdensome, I mean, for the ISPs to retain. But there's talk about it being all the way up to and including a website trail, that is, what websites people are going to.

Leo: You mean the kind of stuff that Google keeps track of with its web history.

Steve: Well, exactly. And he immediately...

Leo: Oh, actually more than that because Google's only tracking your searches. Your ISP knows everywhere you go because of the DNS requests.

Steve: Oh, yes. Well, it's watching your click stream, as it's now becoming called. And of course he immediately marches out the child porn peddlers and online predators, saying oh, we could do a much better job of catching them. Well, of course everyone is sympathetic to that. But it's creepy to think that this whole, I mean, that our ISP that's connecting us to the 'Net has the power that they do to see everything that we're doing and that they would be required by the government to maintain two years of logs of everything every individual does on the Internet.

Leo: That's the old argument...

Steve: I mean, that's really - I'm sorry.

Leo: That's the old argument, if you're not doing anything wrong, what do you have to fear?

Steve: Yeah, and unfortunately our government doesn't have the best track record of dealing with this kind of information aggregation.

Leo: Well, no government does. And anybody should be suspicious of any government. Yeah, I trust our government, but any government that wants to collect this information, that's a bad thing.

Steve: Yeah. And finally, four researchers at Carnegie-Mellon University, UC Berkeley, and University of Pittsburgh, they've come up with an automated, they call it APEG, Automated Patch-Based Exploit Generator. Essentially this thing is able to take a look at Windows Updates, analyze the pre and post patch, and design an exploit for the vulnerability that the patch fixes. And so we've talked about how hackers are looking at Windows Update updates and then manually reverse engineering what it was that was changed. Well, these guys, these computer science researchers have essentially automated that process. And so they are now urging Microsoft - I mean, no. Their point is, if they can do it, so can the bad guys. And we know there's big money behind developing, quickly developing exploits for vulnerabilities. And there's a window of opportunity between the time the vulnerability is known about, the exploit is generated, and everyone gets themselves patched.

So they're urging Microsoft to somehow really work to minimize this vulnerability window. For example, maybe getting the updates all distributed, but having them encrypted so that then a key is provided to just, like, simultaneously decrypt them all in place. Maybe use peer-to-peer networks somehow to push these fixes out much faster. Because right now, I mean, they trickle out of Microsoft. When you consider the number of systems that need to be updated every second Tuesday of the month, I mean, it's often the case that my computer doesn't alert me that it's got some updates for several days after those patches began to get pushed out. So essentially what's happened is there's been a reaction to this constant patching that we're now seeing from Microsoft.

And these researchers are saying, hey, if we can automate it, so can the bad guys. And you've got to believe that there's a huge incentive for them to do so.

Leo: You bet. You bet. Automated Zero Day. I like that. I mean, I don't like that. But it's, I mean, you have to admire their technical prowess, if nothing else. And you know, it's really all taken off since there's been a financial incentive for them to do it. I mean, that's the key thing. As long as they can make money at it, well, we'll throw resources at it.

Steve: That's the change in the last five years. This went from being script kiddies screwing around say, hey, look, Ma, what I can do, to now to organized crime saying, okay, we're going to pay you hackers to do that. And it's big money.

Leo: Amazing. Amazing. Any other news?

Steve: Well, I did have one, since we're waiting for Stina to call...

Leo: Well, she's actually here. She's already here.

Steve: Oh, there she is.

Leo: But let's just finish up, and then we'll get to Stina because I don't...

Steve: Well, I had an interesting piece of email that caught my eye, as I always do. This one the subject was "SpinRite helps kids with cancer." And I thought, okay, how is data recovery going to do that? So this is a letter from Pete Harmon that I got a couple weeks ago. It said, "Dear Steve, I wanted to drop you a line and let you know how much good you're doing in so many ways that you probably never considered possible the day you sat down to develop SpinRite." He said, "I'm a FedEx pilot" - so a Federal Express pilot - "and I have gotten a reputation as a computer hobbyist/geek around flight operations."

Leo: Well, if he's listening to Security Now!, that's true.

Steve: Yeah. He says, "On more than one occasion I've provided tech support to friends and fellow pilots." He says, "I run a website for our pilots called PilotSwap.net." And he said, "Several weeks back I got a phone call from Bill, who told me his computer was dead, and he heard I may be able to help. He described the problem, that his laptop was working fine one day in Honolulu and wouldn't boot at all when he landed in Sydney the next day." Obviously carrying FedEx packages across the globe. And he said, "I asked him if he could hear the hard drive spinning, and he said he thought he could, but stated it was just clicking and clicking, but nothing was happening.

"Not sure I could help, I agreed to take a look at my next opportunity. He left his laptop in my locker, and I brought my SpinRite CD with me the following week. I put it in the laptop, and after two or three dozen attempts to get the laptop to boot from the CD, I

was stumped. His computer was simply not going to boot from either CD or the hard drive. So just for grins, I removed the top two screws holding the hard drive panel on and took his hard drive out, brought it home with me. I hooked it up to my PC at home using an EIDE-to-USB cable I have, and the drive spun right up. But it was mostly unreadable and made lots of noise when I tried to access what few files I could even see in Windows Explorer. I rebooted my machine with my SpinRite CD and was able to quickly see Bill's drive. I set SpinRite to work, and about four hours later the data rescue routines were complete.

"I took Bill's drive to work with me the following week and put it back in his laptop. It booted right up, and Bill was able to recover hundreds of family photos he'd been storing on his laptop for years. Elated and grateful, Bill offered to pay me for my services. Instead, I asked that he make a donation for his gratitude. Here's what I got from him this week. He said, quote, 'Pete. Hopefully by now you received my two voicemails letting you know I dropped off your hardware back on your box. I was able to capture all of my family pics. Many thanks. I made a donation to St. Jude's Children's Hospital in your name for $200. Again, many thanks. Cheers and regards, Bill.'" So then Pete ends saying, "I'm glad I could make a difference, but you and your wonderful product SpinRite made it possible."

Leo: Aw. Isn't that neat.

Steve: So I thought that was really neat.

Leo: Nice story. Nice story. Now would you like to introduce our guest? Because Stina's on the line with us right now.

Steve: Hey, Stina. Welcome, and it's nice to talk to you again. I guess it was, what, about four weeks ago that we bumped into each other at the top of the elevators in San Francisco.

STINA EHRENSVRD: Yeah, that was my lucky day. It's been - and thanks for inviting me. And you've been a relief for me. I mean, it's been literally around hundreds of emails that come from all over the world who's now, you know, ordering our stuff and asking all these kind of questions. And things are taking off.

Steve: Well, that's fantastic. I'm going to talk about the technology and the detailed operations of the YubiKey after we're through talking to you. But I loved your story sort of about where the name came from, where the company came from, and also sort of your vision for authentication. So I wanted you to - I thought our listeners would get a kick out of hearing it directly from you.

Leo: Are you a security researcher, Stina? What's your background?

STINA: No, I'm a product designer. I went to art school. And I married an electronic computer engineer. We've been working as a team for 15 years, and he's learned, you know, he learned with the basics, but they really [indiscernible]. He is, you know, he's the one who helped me. You know, he's made the mod hex that you read about just a few minutes ago, Steve, you know, he's the one who - yeah. So and

then Simon, who is the - another Internet security expert in the team. He's been, you know, putting some efforts into this, too.

Leo: So you're a product designer. And are you a security buff? Or was it all your husband's idea? Or how did this come about?

STINA: We worked very closely. I asked all these stupid questions and, you know. I can tell you, actually, you know, we started together working as a - we actually co-founded a company called Cypak a few years ago, Jacob and I. And this was in the RFID space. And one of the many applications for the technology was the [indiscernible] smart card with a PIN keypad on the card itself. And we called this card the PIN-on-Card. And we were very proud of it because it was so secure. I mean, I think it could have been one of the most secure solutions ever invented. We got the European Innovation Award, 200,000 euros for this. And we were just, you know, the world wants, needs this. And we were just so excited. Until we started talking with customers.

You know, we hadn't even thought about that this card - and it was very secure. But it required a specific [indiscernible] chip built into a specific card with an integrated keypad. And it had to be connected in our [indiscernible] reader. And it needed client software. So when we actually - we were approached by an online bank. And we were planning a pilot with them. But by the end of the day they, you know, one of their bank guys called me up and said we really like the automatic thing. But we don't like the card reader, and we don't like the client software. Our customers, they are, you know, they are from all platforms, all browsers. The Windows versions, the Mac, Firefox. And this client software thing would probably require us to hire 30 new full-time employees only to take care of all the online support.

So this bank guy, he said to us, you know, you're good inventors. But, you know, you can come back when you've removed the client software and the reader. So that was a good challenge. We, you know, we said okay, thanks, we'll make a try. And we started to examine it, you know, looked at the computer, Jacob and I, and I asked a stupid question, you know? I said, there's a keyboard to this computer. You know, and that doesn't require a driver. And he said, hmm, yeah, you're right. So, you know, why couldn't we make a code generator that's simulating the HID driver, you know, acting the same way, with - and we, you know. And, yes, that's where the idea started.

And so we went back to the bank. We got - first we got rid of the client software, and then we made it into a USB fob to get rid of the reader. And we reduced the 12 buttons to one little button. And this was the first version of YubiKey. It was a fat, you know, looked almost like any other USB memory. And that guy who looked at it, he said, hmm, this is an interesting concept. But there's one problem. We prefer to buy security solutions from the big guys. So anyway, I thought it was a good comment.

So this was in May 2007. And with this first prototype version of the YubiKey I decided to start Yubico. I did not have a clear business plan. But I thought it was the only problem was that I was not a big guy. It was [indiscernible]. And the name Yubico came from the word "ubiquitous." I did not want a name that had anything to do with secure, verified, ID, trust, you know, all these boring other names that are

out there.

Leo: Like everyone else at RSA.

STINA: Yes, just without imagination. So I thought I wanted a friendly name. And I like the word "ubiquitous." I envision this to be everywhere, mass market. So I started playing with the word "ubiquitous," and I ended up with Yubico. That's it.

Leo: I like it.

STINA: So we, you know, so the first step, I had the prototype version. And now I realized I needed someone to say that this was a good product. So I asked people, you know, who can write a security report for me? And I came in contact with Simon. And he wrote an independent third-party security report. You know, the one I sent you, Steve?

Steve: Yes, yes, right.

STINA: And the good thing with this was when Simon had written his paper, he was so enthusiastic. So he said - he asked me if he could invest in the company and work for me. Well, but the problem was that I no longer had an independent security review, but I had the perfect inventor. So, yeah, you know.

Leo: It's a good sign when the guy reviewing your security says can I work for you.

STINA: Yeah, and it's on the website tomorrow, so anyone can look at this. It's, you know, it's not third-party, but he was third-party when he wrote it, you know. And Simon, he's a great guy. He is very passionate about open source security. And he recommended me at that time to fly over from Stockholm to an Internet identity workshop in California, and where I could learn more about this OpenID initiative that we think is a great initiative. And, you know, I would learn how we could fit in YubiKey in OpenID so we could enable one YubiKey to go to all Internets.

So I went to California to this workshop. And I met a guy from VeriSign. And he introduced me to another guy at VeriSign. And this guy, he said that the YubiKey could be quite interesting for them, "if." You know, this is the story. It's always been "if." If we could make this device to fit in a wallet and make it very, very cheap and in big volumes. So I thanked him for his feedback. I fly back to Sweden and started, you know, looking for designs. I'm a product designer, so I went on the Internet and said what are - what kind of USB devices are there that are really thin? And the other day a friend of mine gave me a very minimalistic USB key. It was just designed in two parts, a little circuit board and a plastic casing, that's it. So when I saw it I thought, you can't make it smaller, can't make it thinner, can't make it less expensive. And that's, you know, that was the inspiration to the current YubiKey design.

And meanwhile I had met this guy at VeriSign, he had introduced me to another guy

at eBay. And I sent him, actually sent him the first version because we had - the thing wasn't ready yet. But I sent him the first version of the fat YubiKey. And I asked him to look at it because I thought eBay might be a big customer for me. And he said, you know, he wasn't very interested. He even didn't want to look at it. It took him four weeks before he even answered my emails. But then one Sunday in October he came back. And he - yeah. Actually this is what he wrote. So I'm reading from his email: "Dear Stina. I have now tested your product. I'm impressed by its simplicity. I think the YubiKey is the only hardware authentication token that would fulfill the requirements for Web 2.0 services. Looking forward to a further dialogue." You know, that was a good email. So it just took four weeks. And after that he left eBay, and he started working for Yubico in California.

Leo: You're stealing people left and right.

STINA: So now I had an office in California. So I had an office in California, one in Stockholm. So it was Simon, me, and Paul. And, well, in January the little thin YubiKey was ready. And we started shipping the first pilot box. It was to one of Paul's friends who has set up a Chinese IPTV company called Dragon IPTV. We have a, you know, a theme on our website and a film actually show that service. And, you know, we're very enthusiastic because, you know, within a couple of months we had five pilots starting. And we didn't really understand, you know, the customers, they were so happy, they came back, and they said this works so perfect, and the users love it. But the business really didn't - we didn't get any next orders. And eventually we asked them, and they said, you know, there is one problem. There's always one problem. And now they said, you haven't given us a price list, and we don't really understand your business model. Is this open source, or is it not open source? You know, you haven't been perfectly clear on that.

So when we started Yubico, Simon and I, we had envisioned Yubico as an open source company, a web shop where it's free SDKs with a developers community around it and with almost no salespeople, you know, people just sort of sending out things from the web shop, and no flashy offices, eliminating all the expensive layers of distributors and resellers who are now driving up the prices in these, you know, existing Internet security infrastructure. And Simon and I, we were very excited about this idea. We tested it on some Internet security professionals and other safe people we know in this industry. And they all warned us. Actually they warned us. They said you're, you know, it's too risky. We would not recommend you to do that.

So we were sort of standing on one leg. You know, we didn't make [indiscernible]. The customers want to buy from us. We couldn't give them a price list. Because we didn't know, you know, we didn't know who we were. We just knew we had a great product. And then I bumped into you, Steve, so you made us, you forced us to make that decision, you know, emails coming, you know, literally they came in, you know, in my email box, hundreds of them. And I had to call Simon, you know, they're asking for prices. You know, we have to give them some prices. And they're asking for the SDKs. You know, [indiscernible] software [indiscernible]? And then I had another investor who actually joined a little later, a former CEO of Microsoft. So I called - he's the other, you know, we are the only ones taking the big decisions of this company so far. So it was very easy to make an on-the-phone-call decision, okay, now we go, just shift. We know we are taking risks. We know there are big challenges. But this is the way we want to do it, and this is the way that feels right

for us.

Leo: I have to apologize because we probably should have explained what the YubiKey is because I think there are some people who are probably listening, going, all right.

[Talking simultaneously]

Leo: ...synopsize, Steve, give us a...

Steve: Yeah, I'll be covering that after we're through talking to Stina, in detail. But essentially it is what we talked about two weeks ago. It is an amazingly small little, essentially, piece of plastic that is an emulator of a USB keyboard. So it's - we have pictures of it in our show notes from two weeks ago and also this week's show notes, so people can see what it looks like. Or they can just go to Yubico.com and see pictures of it there. It contains cryptographic technology which essentially produces a one-time password which is typed into your computer by this little piece of plastic, by the YubiKey.

Leo: It shows up as a standard USB keyboard, as an HID device.

Steve: Exactly.

Leo: So it can do that. I mean, there's no magic, and it works with everything that supports HID, which is pretty much everything.

Steve: Well, exactly. So it's OS independent. And what Stina was saying before, the problem that people had with the RFID approach was that it needed - there had to be a companion reader, and you had to have client-side software in order to interface it. And what's so cool about this is, I mean, it's funny because when I bumped into Stina at the RSA conference, she was standing there and saw my press credentials and thought, well, maybe I could - I'm sure she was doing this with other press people, too. Maybe this person, who I don't...

STINA: I think I talked to about five people. You were the fifth one.

Steve: Oh, good. Well, I'm sure she was thinking maybe this person will help me get the word out. And being an engineer, when she said this is a one-time password device which is a USB keyboard, my mouth just dropped open because it's brilliant. And that's what I loved about the concept is that it just does what it does beautifully. And we'll go into the technology because the design that underlies this is spectacular.

But what I'm so pleased with, and the reason I wanted to give this a whole Security Now! episode is that what Stina and her colleagues have decided to do is to make the backend authentication services free. No subscription, no license, nothing. They want to just sell the YubiKeys. And unlike a huge company like VeriSign that has a massive infrastructure

that they need to support, and literally all the other companies that I saw on the RSA showroom floor, they were all into locking you in, signing you up, and they were big businesses that were looking for big corporate and offering big corporate solutions. Well, here is something, this YubiKey technology, that is - and I'm looking at the prices that Stina and her group have come up with. Quantity 1, price is $35. Quantity 10 is $25. Quantity 100 is $20. A thousand of them are $16 each. 10,000, $12 each. 100,000, $8 each. So this price drops rapidly.

And their model is to sell the YubiKeys and provide everything else open source. So, for example, they've already posted their source code up on Google's code pages that shows how to decrypt the output from the YubiKey. And they're in the process of putting the technology together and the documentation for how to program the YubiKey. And, I mean, again, it's even better than I thought it was two weeks ago due to the approaches that they're taking and how open they're being about what it is that they've created.

Leo: It's very cool stuff.

STINA: Well, there are some bits and pieces we are, you know, we now rapidly need to put in place because there are huge orders coming in. And it's great.

[Talking simultaneously]

STINA: Our developers community, I vision that to be really something dynamic, growing. But now it's just an embryo. So it's, you know, people are asking where can I find that document, and please, please, give us some weeks. But, you know, we can produce the YubiKeys in large volumes. We set up that production. And anyone can buy these from our website and start downloading whatever they can find there. And eventually there will be more. And I think instead of just fighting against these big guys that we mentioned, we believe we can do this in a different way. I mean, we can - I'm very excited of what this developers community will go. I'm even envisioning that little guys in countries with little IT budgets could develop their own e-democracy and education and payment system based on the YubiKey and open source, and what will happen, you know, that would be really [indiscernible]. You know, there could be systems that sort of are built on this that would require so - will be so less expensive than what our, you know, the current infrastructures are for security and systems and governments and payment data.

Leo: And we should make this clear, that you run a server, so people can use your server. But...

STINA: I mean, we have - it's more for eval. And it's more...

Leo: For evaluation.

STINA: The server is for pilot, for testing, for anyone who wants to use it. We believe that most want to write it themselves.

Leo: Yeah, for security you'd run your own server. So there's a complete SDK, and there's a server, and there's everything you'd need to do that.

STINA: And it makes a free choice. And some small companies or individuals, they don't have a server, or they don't want to invest in servers. So we give them an option. But our focus is the keys. And, you know, downloading software. And we're not focusing on the server. We just have it as an option, too.

Leo: Do you see at some point some sort of unified system so that - see, I don't want to carry 20 different keys. It'd be really nice if something like the YubiKey would become kind of the standard, much like VeriSign's trying to do with their system.

STINA: Yeah. And we have a dialogue with them. We met them at the RSA show. And, you know, we don't see it's either/or. And they support OpenID. We support OpenID. There are other standards coming. And they support OSS, and we will eventually support OSS, too. So...

Leo: Well, standards, that's all you need. If everybody supported OpenID, then I would just use my YubiKey. When a site asked for my password, I'd plug it in. It would have to interface, though, with the one-time password code; right? I mean, it would have to somehow have a server to support that.

STINA: Someone has to be the identity provider.

Leo: Somebody has to be, right.

STINA: And there are a lot of people who want to be that. I mean, Google is already an identity provider. Yahoo! is one. AOL is one.

Leo: So your next big step is to get somebody like that to adopt the YubiKey system.

STINA: Yeah, that would be helpful.

Leo: And then I can go to any OpenID site because that's widely spread. I could choose that provider, whoever uses the YubiKey, as my OpenID provider, use the YubiKey, and I'm done.

Steve: Well, and also, Leo, I mean, there's nothing magic about being an OpenID provider. So, for example, I would imagine before long it would be possible to get some open source server software that is an OpenID authenticator, that knows the YubiKey, and you just run it on your own server.

Leo: Yeah.

STINA: Yeah. I mean, there is no limitation. You don't have to be a big guy to be an open identity provider. You could be, you know, could be a one-man guy that does it, you know.

Leo: And everything you're doing is open source, the server and everything. So that it's very transparent. People know what's going on.

STINA: Yeah, we figured out that that was the way it has to be.

Leo: Oh, yeah, I think it's...

Steve: And it's completely open spec, also. So for example, I mean, the business model of, you know, that we've talked about before when we were excited about the VeriSign VIP system, the football and the card, they're a big company standing behind it. But there's nothing that individuals have to use there. I mean, it's a large corporate solution, you know, an eBay, a PayPal kind of company, not something that universities or, for example, I couldn't use it as the authentication technology for my forthcoming VPN product. I absolutely can use the YubiKey. I mean...

Leo: Yeah, you could write a server, run a server, and use GRC as an OpenID provider. If people trusted you, they have the YubiKey, that'd be it.

Steve: Well, I could except that I'm going to make the VPN server itself, that is, the thing that you're connecting to will be able to authenticate your user YubiKey, I mean, right within itself.

Leo: Wow, cool.

Steve: I mean, really it's a transformational technology because these guys have committed to just opening the spec, opening the software, and selling the hardware at an affordable price.

Leo: Very interesting. Stina, we really want to thank you for joining us, and congratulations on your success. I think that's exciting.

STINA: Thank you very much.

Leo: Yeah, you've created a really interesting product. I haven't seen Steve get this excited in a long time.

STINA: Okay. Take care.

Steve: Thank you.

Leo: Thank you.

STINA: Bye bye.

Steve: Be talking to you in email.

Leo: That's very cool. So I know you're going to talk in more detail about how the functionality works. So we have some questions from the chatroom, but I'll hold off on those until you get, you know, kind of lay it out for us.

Steve: Perfect. Perfect. Okay. So, okay. One of the first things that they realized was since this thing was going to be a keyboard, that is, since you plug it into a USB port, the computer recognizes it as a keyboard, then at the proper time, when you want it to emit its cryptographic string, you just touch a little touch surface on it. It has a really nice sort of green glow. It is literally, it's the thickness of a PC board, a printed circuit board. And remember, Leo, many months ago when I was up in Vancouver, and I showed you something that I had just discovered? You'd already seen it, of course. But it was an SD card that was also a USB...

Leo: Yeah, you flip it open. I think SanDisk makes it, yeah.

Steve: Yes, exactly. It was a SanDisk product. And I thought, this is so cool.

Leo: Because it's as big as an SD card, but it has its own built-in USB interface.

Steve: Yeah, so it's both. It sort of has a funky little hinge. And so you're able to plug it in as an SD card. But then you're able to kind of almost, like, break it in half. And part of it hinges away, leaving the four fingers of the USB interface. Well, that's what Stina talked about seeing and realizing that they could do an authentication device rather than a memory device in the same form factor. And the brilliance of what they did is they said let's - it's going to be a keyboard. Well, so...

Leo: That's what I really think is interesting.

Steve: Yes.

Leo: Because it types, as you say, when you press the button, it types the

password.

Steve: Well, and what they realized was, okay, the football that we've talked about so much, the VeriSign technology, and even the RSA SecurID technology, those are all six-digit tokens because that's, first of all, they're one-time passwords, so that's enough of a string length to be - what's the chance of guessing it? Well, we know that's one in a million.

Leo: Right, right.

Steve: So actually we know it's a little bit less than that...

Leo: And since you only use it once, I mean, it's not like somebody could bang on it for a long time.

Steve: Exactly. But one of the things these guys realized was, wait a minute, since users don't have to type this string, we're not limited to, for example, something easy to type. So we can convey much more information in our one-time password string than you could ever ask a user to type. So, for example, when you touch the contact, it emits 44 characters. It's 44 characters of gobbledygook. It goes zoop, it just kind of comes out.

Leo: Now, of course there's some user intervention involved. You have to click the field that it needs to be in. It's not going to figure that out automatically.

Steve: Yes, that's correct, because all it's doing is just shooting out this...

Leo: It's just typing.

Steve: Yes, exactly. Now, the front 12 characters, the first 12 never change. They are essentially the public ID for your YubiKey. And every single one of them is different. So...

Leo: Ah. It has to send that, though; right? Otherwise it wouldn't be able to figure out if the remaining part of the code was correct.

Steve: Precisely. So that 12, the first 12 characters do not encode the key at all. It's just a serial number, essentially. It's the public identity for the key. Then, okay, because USB keyboards don't send ASCII, they send scan codes, that is, the actual - the USB spec, lord knows why they designed it this way, but it's actually scan codes. Well, that's a problem for internationalization because, when you move the keys around, the scan codes still refer to the same key, but that's a different character. So one of the problems these guys had to solve, and they call it "mod hex" is their name for it, they had to find scan codes which were invariant across all keyboards, so that the language-specific interface on the computer would still convey the same characters. Turns out that was not

a hard problem to solve. There are, among all the keyboards, there are enough keys that are always in the same position that therefore always have the same scan codes on the keyboard that they were able to do this. So the ASCII that you see is always the same, that is, it's...

Leo: It's kind of the standard set that every keyboard has, as opposed to the extended stuff.

Steve: Exactly. Although it's a bizarre set of characters. I mean, I'm looking at LVKCCUTLIBFIVJGUTRJNDJBUK...

Leo: So it sounds like it's alphabetic, not numeric, not punctuation.

Steve: Correct, it's all alpha characters.

Leo: Uppercase and lowercase?

Steve: Well, I'm looking at a capital "I" at the beginning. But then everything else is lower case. So it looks like maybe they just capitalize the first character.

Leo: The reason I ask is sometimes passwords are case sensitive, sometimes not. Sometimes they require you have to do special requirements like a number at the beginning, which really drives me crazy, by the way.

Steve: Well, and see, this would not be used by any normal, like, thing that was asking for a password.

Leo: I guess you're right, huh.

Steve: Yeah, the idea is this would always be going directly to someone who's doing your authentication.

Leo: Got it.

Steve: Okay. So we have - they're encoding four bits, four binary bits in each character, so they have a 16-character alphabet. So those first 12 characters that are invariant, that identify which YubiKey out of all YubiKeys you've just stuck into the keyboard, those 12 characters convert to 48 bits. So there's this 48-bit ID which the YubiKey declares itself as. Then you've got 32 characters which immediately follow. So that's a total of 44 characters, the first 12 followed by 32. Well, those...

Leo: That should be enough.

Steve: Oh, yeah, I mean, it just - and it's got - what I was reading a second ago was the output from my YubiKey, which...

Leo: Notice I interrupted you before you got to all 47 characters.

Steve: I was - I had four to go. But my point is that's what this thing looks like. So it's 32 characters. And of course, again, four bits per character means that it encodes 128 bits. So essentially you are sending a 128-bit blob every time you authenticate. So the YubiKey contains a write-only 128-bit AES secret key. So mine is different than yours is different than everybody else's. Those first 12 characters that the YubiKey sends in the case, for example, of authentication by Yubico, those 12 characters are used to look up in their database the associated 128-bit AES key that is also contained in the YubiKey itself. So the YubiKey encodes some data that I'm about to describe using its secret 128-bit AES key. And we know AES is just Rijndael, my favorite cipher of all time. It encodes the 128 bits of data using its secret AES key, turns it into this mod hex, and spits it out.

It then travels across the Internet or to wherever it's going for authentication. The receiver knows this key's secret 128-bit symmetric key. It simply - it does a Rijndael decode to turn it back into the 128 bits of plaintext that then allows it to proceed with authentication. So the YubiKey itself, you can write the AES key into it, that is, its own secret 128-bit AES key. But there is no provision at all for reading it out. It will never tell you, nothing you can do to it will cause it to relinquish its key. You can only push the button and have it spit out these tokens. But you cannot get it to tell you its key. You can give it a key. It'll never give it back to you. It'll only give you the result. So it's very secure from that standpoint.

Leo: I think this is such a cool technology. I can see why you got jazzed about it.

Steve: Well, and so now we've got 128 bits. And, I mean, they came up with 128 bits, of course, because that's the width of a Rijndael block. So that's 128 bits is 16 bytes. The first six bytes is a unique device ID. And again, every single key ever made has a unique ID. This is part of what, you know, they're planning ahead. They're saying what if this thing really takes off? Well, we want to make sure that all keys are unique so that we can use them for identity purposes without any collision. So you've got six bytes, which is 48 bits, which is a ton. I mean, I don't have a calculator in front of me, but that's a lot of devices. That's more than we're going to need. It's like the same six bytes in a MAC address for an Ethernet controller where you want every Ethernet controller in the world to have a unique ID so that - because that's the way that the MAC address identifies Ethernet devices on an Ethernet LAN, and you don't want them to collide or you can't have those two devices on the LAN. So similarly, this prevents any collision between all the YubiKeys that will ever be made.

Okay. Then we have a two-byte, what we call a "session counter." That is a nonvolatile counter, and it counts the number of times the YubiKey is powered up. So if you plug it in, that session counter increments once when the YubiKey powers up. And that's nonvolatile. So it only increments, and it never resets to zero. Next is three bytes of a timestamp. And that's a three-byte counter, 24 bits, that runs at 8Hz. So eight times per

second this three-byte timestamp is counting up. Well, that means that it will run, before it wraps around, it runs for 24 days. And that always starts at zero when you plug it in. So you plug it in, the session counter, which is two bytes, increments by one. And this timestamp starts running.

Well, this has a number of features. One is it has an anti-phishing feature because it means that they're able to determine when - because essentially you've got time embedded in the YubiKey's output. They're able to determine, that is, the recipient is able to determine for successive outputs during a single session when these were generated by the key. So if anything were to intercept this and impose some interception delay and then try to use it, it contains a timestamp. So by comparing the timestamp received from previous receptions of this YubiKey output, they're able to determine whether these are out of sequence, whether they've been delayed for some reason, because normally the authentication happens in near real-time. You know, you're on a form, you go to the YubiKey field to authenticate, or maybe you've got this all built in, for example, into a, for example, a VPN client. And you press the button, it types the stuff out, then you would submit the form. So there's only, like, a few seconds delay between the time the YubiKey generates its token and the authenticator has it and is able to authenticate. So just, I mean, they had 128 bits to play with. And so from an engineering standpoint they said, well, what cool things can we do with all this face? So they gave us an 8Hz timestamp, so every YubiKey token is timestamped in real-time as it's generated.

Leo: That actually solves a problem that VeriSign has with their football or their little card; right? Because if you're out of sequence, sometimes, occasionally, if you press the key a bunch of times or whatever, you'll have to get back in - they can lose track of the sequence, I guess. Does this solve that?

Steve: Well, actually we've got so many bits. And what we're really doing is encrypting this thing. In fact, all you really want to do is prove that you have the magic 128-bit AES key. So the fact is, just decrypting this and doing a sanity check or...

Leo: Oh, that's all you have to do.

Steve: That's really all you have to...

Leo: You don't have to match it up. You don't have to generate a matching key or anything like that at all.

Steve: Exactly.

Leo: Oh, I see. So it really is a different technology than the football.

Steve: Yeah, well, it's a completely different approach because they're not, again, they're not having to try to say, okay, we've only got six digits. Because six digits does not give you enough specificity. I mean, you know...

Leo: There's more than a million keys, one hopes, out there, so that's not going to do it.

Steve: Exactly. Okay. So the next byte is a session use byte, just one byte which increments every time you use it during that session. So remember we have the session counter that increments once for the whole power-up cycle. And then the session use byte, it starts at zero for at the beginning of every session. And then it increments. And that's just to make every single one unique, even though the timestamp would also do it. But the next two bytes is 16 bits of pseudorandom data. So they have a pseudorandom generator that just generates 16 bits of noise that is added in. The reason they do that is that the one concern that you would have in simply encrypting Rijndael or any symmetric cipher block, we've talked about this before, is that this uses what's called ECB mode, Electronic Code Book mode, meaning you simply take the data, and you encrypt 128 bits into 128 bits.

Well, the problem with ECB mode is the so-called "known plaintext attack," meaning that if you ever are encrypting the same data or potentially similar data, there's a theoretical vulnerability, that is, that you could begin to build up a mapping between the plaintext and the encrypted data. So what they do is they throw in this two bytes of pseudorandom data in addition to three bytes of timestamp, which is running at 8Hz. That's much faster even than you're able to emit these key output. So there's a lot of randomness in those two things. Or at least nondetermination. And then they have the pseudorandom bytes. And, finally, a 16-bit, two-byte CRC, a Cyclic Redundancy Check, which applies to the entire block.

So the idea would be you receive one of these things, which is this funky mod hex code. You translate each of the 16 different characters in the alphabet into four bits. That gives you 128 bits. Then you look up the key's secret 128-bit Rijndael symmetric key. You decrypt that 128 bits into this data that I've just described. So now you have the device's unique ID, six bytes; the session counter; the timestamp; the session use byte; then the two pseudorandom bytes that you ignore. But you do run all that through the CRC just as a sanity check to make sure that you have probably decrypted something that is valid and that there was no data loss or corruption at any point. And then you've got all this information about the YubiKey, that is, how many times it's been used in that session, a sense of the time flow during that session, and you can use that to authenticate and to provide various forms of anti-spoofing protection.

Leo: Very cool. Somebody asked in the chatroom if a keystroke logger could capture these keystrokes.

Steve: Absolutely. And I'd be happy to read mine out to anyone who wants.

Leo: It doesn't matter.

Steve: Precisely.

Leo: It's a one-time key. And that's what makes it so powerful.

Steve: Yes. Exactly. It is a one-time key. And again, oh, I forgot to mention, that session counter that is two bytes, they actually have stolen the top bit from it. So it's only 15 bits, meaning that it runs up to a maximum of 32767. It starts at zero. When it gets to the maximum of 32767, it stops, and the YubiKey dies. So that's one thing worth noting.

Leo: Wait a minute, say that again? It can only generate how many?

Steve: No, no. That's what's cool. It's not about - it's how many sessions it can have. That is, it counts - it's a 15-bit counter. So it counts up to 32767.

Leo: So what's a session?

Steve: A session is when you power this thing up.

Leo: Ah. So you would have to unplug it and plug it in again to start over.

Steve: Exactly. Well, no...

Leo: Big deal. You're not going to use 15,000 sessions.

Steve: No no no. No. Now, remember that the key is - this is a one-time password generator. Therefore that session counter can never be allowed cryptographically to wrap around to zero because that's where it started. And although...

Leo: It would repeat passwords.

Steve: Exactly.

Leo: So are you saying that after 15,000 passwords this stops working?

Steve: No no no. It's very important that people understand this. First of all, it's 32,000. It's 32,000 sessions.

Leo: 635, what it is. No, it's only 15 bits.

Steve: It's 15 bits. So it's not 65,000, it's 32,000 sessions. But you can generate as many passwords in a session as you want to.

Leo: And you're saying a session begins when you power it up. So it sounds like every time you unplug and plug it in, that's a new session?

Steve: That is correct.

Leo: Okay. So you wouldn't want to unplug it and plug it in.

Steve: Well, consider that that's a big number. First of all, that's 10 times a day for nine years.

Leo: Okay. Never mind, then. We won't worry about it.

Steve: Well, and imagine that this thing takes off. For example, you're using it as your OpenID token.

Leo: Which means you'd probably want to leave it plugged in.

Steve: That's my point is you're - or you're using it to authenticate yourself to your bank and your corporation and so forth.

Leo: So before you get to work, you sit down, you plug it in, and you press that button whenever you need it, and you unplug it at the end of the day.

Steve: Precisely.

Leo: You're not going to use - how many per session do you get? Is it...

Steve: No, it's infinite. There's no limit on the number of keys you can generate per session.

Leo: Oh, okay. Then forget it. Then it's not a big deal.

Steve: And the other reason that this is important is, remember, we know about nonvolatile RAM not lasting forever. That is...

Leo: So it's not writing to the RAM, or the EPROM. It's reading from the PROM.

Steve: Well, the nonvolatile portion, that is, this two-byte session counter, that's

changing. So they did need to protect against the standard NV RAM fade, because we've talked about how some nonvolatile RAM you can only write to 10,000 times. Some is 100,000. Well, in this case, from an engineering standpoint they knew that the nonvolatile portion of this would be aging as it's counting sessions. So exactly as you said, Leo, I mean, imagine that the typical use might be you plug it into your laptop, turn your laptop on, a little green light comes up. And then during your use of the laptop over the course of several hours, any time you needed to authenticate to an OpenID site you would just reach down and put your finger on the little touch surface, and it would emit a YubiKey token.

Leo: I love this.

Steve: It is really neat.

Leo: You know, I use - and actually it's interesting, our new office manager, Frederique, said is it okay if I plug in my RoboForm. She has, and I use this, too, RoboForm AI has a USB version. So you plug it in, and your passwords are on there and authenticated. And it's a very nice system. But so it's the same idea. I think people are already used to this. But this is so much slicker and so much secure.

Steve: Well, yeah. I mean, it is absolutely secure. You cannot get the YubiKey to tell you its secret 128-bit AES key. All you can get it to do is to spit out unique tokens which only have meaning if the authentication end already has the key. And what I was so pleased about as Yubico's concept of what they were going to do with this evolved is, I mean, and they even changed the language on the website in the last couple weeks because there was language about, well, you know, the keys you're buying from them now are evaluation only, and they'll expire. All of that's gone. That was, you know, they weren't sure what business model they wanted to have. And they've settled on, okay, we're going to sell these keys.

Leo: They picked the right model, I think; don't you?

Steve: Oh, I mean, it's why I'm so excited about this. Leo, I can't - there's no way that VeriSign will tell me the algorithm that they use in their footballs or their cards. Therefore I cannot...

Leo: You can't trust it, right.

Steve: Well, I cannot authenticate.

Leo: Oh, you can't do it yourself, either. Right.

Steve: Right. You cannot do it offline. There's no way, for example, that this could be used for, like, Windows log-in that is VeriSign stuff. You have to have a network connection in order to get them to do it. And it's like, well, okay. But corporations have a

substantial cost associated with using that kind of big corporate authentication solution. And it's, I mean, VeriSign's model is we're going to be - we have a big network. We're not going to go down. You can trust us to be up all the time. And it's like, well, okay. But it does limit the applications. Here Yubico tells you everything you need to know. I mean, it's why I love it. I mean, I love crypto, and I love authentication. Now I've got these keys that I can use for any purpose I want. I mean, Sue, Greg, and I are going to use this to access...

Leo: So you're going to do it. You're going to implement it.

Steve: Absolutely.

Leo: Oh, isn't that neat. I want to do it. I don't have anything to do it with, but I just want to do it.

Steve: Well, it is immediately an OpenID.

Leo: You know what I thought it would be really good for? Now, we're not probably going to do this. But if you wanted to do a paid, say, paid podcast, a paid show, somebody could subscribe, and you'd mail them, they're cheap enough, you could mail them a YubiKey.

Steve: Yes.

Leo: And they couldn't watch it without the YubiKey. And it's kind of - I don't want to say, I'm not recommending it for DRM. But it could be the ultimate DRM.

Steve: Well, as a matter of fact one of the applications that Stina mentioned is the idea of for online gaming or even for downloadable games. It ends up being a very painless hardware key where you would allow people then to download updates and download the software which won't work until they authenticate with their YubiKey.

Leo: So now it's gone clean out of my head. I thought of some negatives about this. I mean, I guess one negative would be if you lose it there's no way they can give you a replacement; right?

Steve: That is correct. Now, I did want to mention my concern over the idea of this 32,000 sessions, or days, or however you would use it. The comment's been made that if this thing is on your key ring, and you're putting it in, pulling it out, putting it in, pulling it out, it probably mechanically degrades.

Leo: Exactly, it's going to wear out before it numerically wears out.

Steve: Exactly. And so at some point it's looking kind of ragged. And so you would tell your IT department, hey, you know, my YubiKey's chipping off, and my dog chewed on it, and can I have a replacement, please.

Leo: Yeah, and then what do we do?

Steve: Well, oh, that's no problem at all. Or if you lost it. You'd report it lost the way you would a credit card, and they'd just cancel it. They'd just hand you another one for, I mean, I didn't go through her whole price list. But at a million quantity, I mean, I guess a large corporation that wanted to standardize on this, they're $5 each.

Leo: Okay. Which is what the football costs from PayPal. And they're subsidizing it.

Steve: Well, yes. Now, I do want to say that one downside, it's worth mentioning, is that the football and the credit card, that is, the two visual numeric ID solutions, because they don't use any kind of electrical interface, they could be used for authentication over the phone or at a...

Leo: Right. You have to manually enter the number or speak it, but you can do that. You couldn't do that with the YubiKey.

Steve: Or like in some sort of a, like a Windows kiosk or something where you don't have access to the physical machine. So one limitation is it is, being a USB thing, it's for an end-user who has a computer and has access to that computer's USB ports. Someone in the, I think they have an FAQ on their site where they said, well, wait, my USB ports are all on the back of my desktop. I can't get to them. And the answer...

Leo: Get an extension cord.

Steve: Exactly, just get one of those cool little USB extension cords.

Leo: Right. And most PCs now have it right on the front. I mean, that must be an older machine.

Steve: Well, because it's becoming so ubiquitous, to use a term. Anyway, the other thing that I think is interesting is that, I mean, on the positive side, I know that our listeners are thinking about this, and there are ways that this can solve problems beyond just sort of generic OpenID-style authentication. For example, imagine a corporation where they wanted tight control over their corporate portal so that, for example, they don't want spambots coming in, posting things. They even want control over what sections of the site you're able to go to. So it'd be very easy for them to YubiKey-enable their own corporate portal so that, if you want to make the query from a database, it says fine, please authenticate. And all you do is you just touch this little spot that is glowing green on the YubiKey. It spits the string out, and then you've authenticated yourself.

So you can imagine all kinds of applications where, again, because once this thing is installed, it's so simple to, like, reauthenticate, that it really provides, I mean, imagine the pain of being asked to continually read six digits from the football or the credit card. I mean, yes, you could, but it's much easier to just touch the surface, and it authenticates for you. And your entire involvement is just touching the YubiKey.

Leo: And as you point out, by virtue of the number of digits it can spit out, it has much more secure setup. I mean, it's a better way to do it.

Steve: Well, I mean, yes. We would argue that six digits that are changing all the time is secure enough. But it is the case that this is vastly more secure because you're communicating 128 bits which are encrypted with 128-bit Rijndael key. Only the matching key will decrypt it and then give you the data. And as I said, you can really ignore the data. The fact that you decrypted it means you decrypted it for the proper key. So that proper key had to be at the other end of the connection. So it's dramatically more secure than six digits could be.

Leo: Couple of points from our chat. [Loveman ph] says if you have to phone home, doesn't that mean that it wouldn't work with static passwords on a website? What we're saying really is that it's an OpenID device or something like OpenID, where it would establish your identity, and then OpenID - so the website - so say you use it for TWiT.tv, which we support OpenID. We don't have logins at this point, but if we decide to do logins we support OpenID. All you would have to do is use an OpenID provider that supported the YubiKey. Then when you go to TWiT.tv and it says, okay, log in or provide your OpenID identity, you just plug in the - I think, now correct me if I'm wrong, Steve, but you would plug in the YubiKey. You would click on the place where it said provide open - no, it wouldn't work for that, would it. You'd have to enter your OpenID identity, so go to your OpenID provider, then...

Steve: Then you authenticate...

[Talking simultaneously]

Steve: Yes.

Leo: So then you'd put in your key or press the button, and that would spit out the code that your - you wouldn't even need, say, a log-in and a password, or might you?

Steve: Well, yes.

Leo: You're in effect logging in because you have a unique number in that.

Steve: Well, you have the flexibility - that's, again, that's what I love about this is that this is a - it's like a low-level perfect crypto toy that you can do anything with you want. Now, the reason you probably want a passphrase is that you want to protect against, remember, we're talking multifactor...

Leo: Somebody stealing your key, of course.

Steve: Yes, multifactor authentication, meaning more than one factor. So you would have something you know would be your passphrase. Something you have is the YubiKey.

Leo: Got it. So you'd want both.

Steve: Yeah.

Leo: Yeah. That makes sense. So this is a very cool - also one more comment. And this one is more about our broadcast than it is about the show today. Robodog has twittered to me on our Twitter account, by the way, which is TWiT Live, if you want to follow the podcasts, broadcasts, and send us questions. He says, all right, Steve has the pipes to support many cams. Can we see Steve by next week, please? And he also wants a - and I think this is a brilliant idea - a whiteboard for you. Wouldn't that be cool?

Steve: I'm busy enough, Leo.

Leo: No, no, I can do it. I'm not asking you for it. But we will - eventually the set up will be, and it's just we're, you know, in fact after the show today I'm going to open up our TriCaster, which will give us this capability of switching to a camera. So Steve, all you would have to do is send video with your Skype, which you can easily do. And then we'd be able to switch to your video as you're talking.

Steve: Eh, we'll see how that goes.

Leo: You don't want to do that?

Steve: I don't think so.

Leo: You don't want anybody to see you? You're not wearing any pants, are you. Steve doesn't want to have to put on makeup.

Steve: There's nothing to see. It's me leaning forward, talking into this beautiful Heil microphone.

Leo: What do you think it is when they look at me? At least they'd have something else to look at. Now, the whiteboard is kind of an interesting idea, and I think we could do a digital whiteboard. We're going to redesign the homepage for this. In fact,

we have a very nice homepage in mind. But that's an interesting idea, where we would have something that you're on, on your side - you've done PowerPoints for the TV show - where we could actually throw those things up so that people would have some additional information to...

Steve: The problem is that was a TV show, and everyone who was watching it was watching it. This is an audio podcast. And I would always be focused on conveying this information through audio. And I think that's, for me, that's the model of this podcast.

Leo: No, you're right. In fact, I don't want ever the video to supersede or in any way impinge on the audio. Because most, 99 percent of the audience listens to the audio, not watches the video. So you're absolutely...

Steve: .99999.

Leo: Well, it's not that bad. It's not that bad, Steve. There are a thousand people watching the video.

Steve: What?

Leo: Oh, yeah.

Steve: How do they even know about it? No one who is listening to Security Now! has even heard about any of this stuff happening.

Leo: Well, they have now. But literally there are a thousand people watching. So...

Steve: Next week watch out.

Leo: So it's not .999, but it might be 99.9. I don't know what it is. But so we will have things like show notes and stuff in real-time on the page. So we'll at least be able to give you links and stuff if you're watching and you want to have more information right there. I think that's a good idea. But you're right, Steve, and I really want to emphasize this to everybody who listens. You're the audience, so we're not going to do anything to impinge on you. And you're right, if we started doing a whiteboard that would change the dynamic of it. So I agree with you, Steve.

Steve, anything else to say about Yubico? It's Yubico.com. But it's really not selling to end-users. It's selling to people who would implement it as part of their system; right?

Steve: Well, it is selling to end-users. And I know for a fact from the email that I've received that a bunch of end-users have ordered $35 YubiKeys.

Leo: But what would you do with it?

Steve: It's an OpenID. And right now it is useful as an OpenID authentication.

Leo: But who's...

Steve: They provide...

Leo: Oh, they're doing it.

Steve: They are, yes, they are right now an OpenID authenticator.

Leo: Oh, so at the very least you could use it as an OpenID tool right now using Yubico as your OpenID provider.

Steve: Exactly. And they've also published that they're doing backend authentication. They've got the secret AES key for every YubiKey they sell. And they have servers up and running, and a fully published public open source web interface that allows anyone who wants to, for example, well, to finish that thought, anyone who wants to to use their backend authentication right now.

Leo: Right, right, right.

Steve: So, for example, you could use it for access to your own wiki stuff and that kind of thing.

Leo: Perfect. Oh, you're right. So we could use it internally, yeah. All right, Steve. Very interesting stuff. I'm glad Stina could join us. Stina, we never did attempt her last name, but I think it's Ehrensvrd.

Steve: Yes.

Leo: And we should probably have said that, and said to her, is that how you say it? But anyway, of course, as usual, as with everybody I've met from Sweden, she speaks better English than we do.

Steve: Well, I'm really glad we've covered this. We're done with the YubiKey at least for now, unless any other new developments happen. But I think it's - authentication is crucial for the future. And I love the policies that these guys have adopted for making this really cool one-time password hardware authentication token available. It's so useful.

Leo: Next week we're going to answer your questions and suggestions and share them with the world. So you've got to go to Security Now!'s website, which is GRC.com/securitynow, and you can submit suggestions and questions there. You can also find there 16KB versions for the bandwidth impaired, and full transcriptions thanks to Elaine - tip of the hat to Elaine. Cory Doctorow sent us a note saying is Elaine available for other podcasts, other stuff? And we said yes. [Note from Elaine: Thanks!]

Steve: Yeah, she loves it. She's just tremendous. [Note from Elaine: Thanks again!]

Leo: What else? Oh, show notes are there. And of course, don't forget, that GRC.com is the same place you find all of Steve's great free security programs like ShieldsUP!. More than 50 million people have tested their firewalls using ShieldsUP!.

Steve: I think we're at 73 million now.

Leo: What?

Steve: Yeah.

Leo: Holy comoly. That's amazing. Well, we'll add another thousand right now, just like that. And of course that's where SpinRite is, everybody's favorite, my favorite, hard drive maintenance and recovery utility. If you've got a hard drive, you need SpinRite. GRC.com. Thanks, Steve. We'll see you again next week.

Steve: Talk to you next week, Leo.