PIP & More Perfect Passwords

Description: Steve and Leo discuss two topics this week: The availability and operation of VeriSign Labs' OpenID PIP (Personal Identity Provider) beta, offering many useful features for online identity authentication; and Steve's recent redesign of the algorithms behind his popular Perfect Passwords page.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-107.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-107-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 107 for August 30, 2007: VeriSign's OpenID and Toward More Perfect Passwords.

It's time for Security Now!, everybody's favorite security podcast and, thanks to your votes, the #1 technology and science podcast of the year, podcast award winner. Steve will be collecting his award at the Podcast Expo at the end of the month.

Steve Gibson: Yes, I will be...

Leo: That's pretty exciting.

Steve: Well, it's my one chance really to annually hook up with Elaine, our transcriber. I met her for the first time - even though we'd had about, I guess, a year and a half worth of relationship electronically, doing our transcripts, you and I both met her for the first time during the last Podcast Expo, which was really fun.

Leo: That was fun, yeah. I will not be there, but I wish you all the best.

Steve: You'll be there in spirit, Leo.

Leo: I will. I will.

Steve: And none of this would be happening were it not for you and your spirit, so it's true.

Leo: Well, it's my goal actually to win every year with one podcast or the other.

Steve: It's all good.

Leo: Next year it may be MacBreak Weekly because they were nominated this year, and I feel bad. But, you know, you predate them, so...

Steve: Well, and what's annoying is that your mainline TWiT podcast and MacBreak Weekly and Security Now! are all in the technology science category.

Leo: Right. And we can no longer win the People's Choice Award because we've won that. So we're basically running out of ways to win. But I'll just keep making podcasts, find a way to win.

Steve: It's not like that's a problem. I was over on your page the other day; and it's like, my goodness, look at all these things you're doing.

Leo: Yeah, I've got to stop.

Steve: Busy boy.

Leo: I've got to stop soon. So today what are we going to talk about, Mr. Gibson?

Steve: Well, a couple things. I want to talk about, in detail, the VIP program, which really first came to light publicly when our guest from PayPal was talking about how they're using VIP, which is VeriSign Identity Protection, as their backend system behind PayPal and eBay's authentication fob, which was, you know, the so-called PayPal Security Key that I thought was such a cool thing, and which I really do continue to think so. What I did was I've plowed into that and found a bunch of cool stuff, told you about it, and now you know about it. So now we've got to tell our listeners. So...

Leo: I've signed up. I'm ready. I'm using it. I love it.

Steve: So I want to talk about that. But also, because I didn't think that would take a full episode worth of time, and because there was something else going on, I thought I would talk about something we've never mentioned before, a way of protecting the use of symmetric encryption, something called CBC, which is Cipher Block Chaining. It becomes up because I implemented it recently for the Perfect Passwords, GRC's Perfect - or our Popular Perfect Passwords Page.

Leo: PPP.

Steve: Yes. More than 3,000 sets of passwords are generated a day. And I looked the other day, and more than three million sets of passwords have been generated total. Just it's very popular with people. The Popular Perfect Passwords Page is very Popular with People. And so now what we have is even more Perfect Passwords, which are the same number of passwords, but they're even more perfect than they were before. So we'll talk about that.

Leo: So, let's see. I know we do a mailbag episode now, every other episode; but do you have any quick emails you want to address or talk about?

Steve: I've got two things. One is an errata, and one is - I keep trying to look for interesting SpinRite stories from our listeners that are not repetitious. And I found one where a guy bought SpinRite by mistake, which I'll explain in a second. But I did want to mention that a couple weeks ago, on the second Tuesday of this month, you may remember that the week before last I reminded people to make sure they were up to speed with their Windows Update because Microsoft was pushing out nine different updates, so a little larger than usual. What was significant is that - and this has actually created a lot of buzz in the security industry - is that one of those nine was a fix for three problems in the new Vista Gadgets.

Leo: Really.

Steve: Yes. And so, Leo, you have continued to talk, as we've talked about security issues, you've been focused on whether Vista would have been safe, whether this is different in Vista and so forth. What's significant is that Vista's Gadgets, being Vista-only, represent a new problem that prior versions of Windows such as XP and 2000 and so forth would not have had because they didn't have gadgets. And the reason there was some buzz in the security industry is that one of these was a buffer overflow, i.e., you know, critically rated, take-your-computer-over remote control, blah blah blah, in the RSS feed reader gadget. Which means that, had any bad guy been able to push a properly malformed content into any feed that was popular and that people had subscribed to, their Vista machines would have been commandeered.

Leo: Wow.

Steve: So, you know, you can see why there was some buzz. It was like, whoa, you know, here's an example of something, a new feature. And, frankly, this is what we were talking about here well before Vista's release was new features are going to have problems. And people go, oh, well, but we're checking them, our whole new security profile, and we're scrutinizing our code. It's like, yes, I know. New features are going to have problems. And that's exactly what happened here. So the good news is, Microsoft found it, someone reported it, Microsoft patched it and fixed it. But this, again, it's an example of standard sort of security philosophy that you just can't get away from.

Leo: Yeah, interesting. And, you know, those are written in - I thought they were written in JavaScript, like most widgets and gadgets are. But there must be some code in there, as well.

Steve: Well, you're always dependent upon lower level stuff. So it might have been in the

JavaScript. It might have been in a library that was called to parse some of the data. You'd have to be really deep in this to know where the problem was. But Microsoft confirmed that, yes, had you subscribed to a malicious feed, or a good feed that had mistakenly provided some malicious content, your machine could have been taken over remotely.

Leo: Fascinating. All right.

Steve: Anyway, we have a Security Now! listener, Rob McCall in Oakridge, New Jersey, who says - and I kind of liked this one because, again, it was a different kind of salute to SpinRite. He said, "I'm a long-time listener of Security Now! and have learned much from your great podcast. I've also been a long-time fan of your site, which has educated me to no end. I've been debating whether or not to purchase SpinRite for quite some time. I finally decided to buy it when I started having problems with my HP desktop computer. Whenever I started the computer, it would rarely make it to the log-on screen. And even if it did, I could not log on because the screen froze, and I could not do anything about it. So I purchased SpinRite thinking maybe it was just..." he says. Now hard drive problems are just a hard drive problem because we have SpinRite. And he says, "...thinking maybe it was just a hard drive problem." We can take care of that. He said, "Sadly, it was a failure with another part of the system and not my hard drive." So SpinRite was unable to help him. He says, "Since then I have used SpinRite several times on my collection of old hard drives. Much to my amazement, SpinRite was able to recover data from these drives that every other program and company had deemed unrecoverable. This is one purchase I know I will be happy with for a very, very long time. SpinRite is a cheap solution..." - well, I would rather call it an "inexpensive" solution...

Leo: Inexpensive solution, yeah.

Steve: "...to services that many companies will charge several hundred, if not thousands of dollars to do. I cannot thank you enough, Steve." Signed, Rob McCall.

Leo: Wow, that's great.

Steve: So, Rob, thank you for listening to Security Now!, for visiting GRC, and for purchasing SpinRite. Took a while, but you're glad you got it, so I'm glad for that.

Leo: That's really neat. All right. Let's get to our topic of the day. Two of them to talk about today: Practically Perfect Passwords...

Steve: You have to have much coffee in your system, Leo, as I do in order to handle Even More Perfect Popular Passwords.

Leo: Whatever you say.

Steve: Or whatever that was.

Leo: Whatever you said.

Steve: The Popular Perfect Passwords Page. But first...

Leo: Go ahead.

Steve: But first we're going to talk about...

Leo: We're going to do this...

Steve: First we're going to talk about - we're going to step all over each other.

Leo: So it's PIP and PPP is what it is today.

Steve: There you go. Yes. Our PayPal security guy who we had on a couple weeks ago talked about how VIP was the back-end that PayPal was using, the VeriSign Identity Protection system. So I did some digging around, and I discovered that there's a program in beta which is called PIP, and that stands for Personal Identity Provider. And that's being hosted at verisignlabs.com, not VeriSign, VeriSign Labs. So if our listeners go to - and in the show notes I've got links to all this. So people who are just listening to this while they're out jogging or driving around, if they go to the show notes they can pick up the links for this. It's pip.verisignlabs.com. That's the doorway to all this fun stuff.

And what's so neat is that VeriSign labs, that is, this PIP, the Personal Identity Provider, which is what PIP stands for, is in beta, but it's up and running and working. And these guys are an OpenID provider, that is, they are an OpenID-compatible authenticator. Now, you'll remember that we talked about the whole OpenID system a couple months ago, the idea being that it would provide you with a single point of authentication, so-called "single sign-on" technology, where you would go to any other site that is OpenID-friendly, and you would be able to, instead of needing to log on and create individual log-on credentials for each such site, they would be able to ask a third-party server, in this case VeriSign Labs, to authenticate you.

Well, what's so cool is that VeriSign Labs, that is, this pip.versignlabs.com, you're able to create OpenID accounts. And actually you can have as many of them as you want. For example, I have stevegibson.pip.verisignlabs.com. I got Gibson. I got GRC. I got SteveG. I got a bunch. And I would encourage our listeners who are interested to run over there and get their names in various flavors that they want signed up because this may well end up being something that you're glad to have been able to get your name on. Which if, you know, somebody else comes along and has the same name you do, would preempt you subsequently from doing so.

So, but the idea is you create these OpenID accounts. Then you can associate them with multiple second-factor authentication, starting with the PayPal Security Key. So you can give the serial number of your PayPal Security Key to this site as part of your account creation, and they will sell you more. In fact, you can choose on the purchase form whether you want one, two, or three. Now, these are not as inexpensive as the keys from PayPal because, as we know, PayPal is essentially covering the cost for this because they want to promote the use of their Security Key. VeriSign doesn't have the same motivation. Their dongles - oops, I'm sorry, fobs - their fobs are $30 apiece.

Leo: Ah. But I can use my $5 PayPal fob with this.

Steve: You can use, yes, you can use the $5 PayPal fob that you already have. But remember that we were also bemoaning the fact that PayPal would only allow you to register a single fob at a time, which prevented us from leaving one at the office and leaving one at home so that we didn't have to worry about filling up our keychain full of these. Well, first of all, the fact that VeriSign Labs has this program which is in beta, which is an OpenID provider, it's beginning to solve the problem of needing individual second-factor authentication fobs or dongles or doohickeys or thingamajigs for every different provider because OpenID is clearly gaining strength, and they're able to function as an OpenID authenticator. So, and the fact that you're able with VeriSign, not with PayPal and eBay - I want to make clear that that's still a single-fob system. But with VeriSign you can register as many of these as you want. I've got four at this point just because, you know, just because I could.

Leo: Why not? Well, so I want to make this clear, though, you can use your PayPal fob with VeriSign, but you can't use those extra VeriSign fobs with PayPal. It's a one-way street.

Steve: Exactly. Or at least not at the same time. You could probably deregister your PayPal fob and register a VeriSign fob because the fact is they're all from VeriSign. They look identical. The VeriSign ones are a nice VeriSign purple color, whereas the PayPal uses that PayPal sort of, I don't know, gray, yeah.

Leo: Gray, yeah.

Steve: But otherwise they...

Leo: So I just did this. I signed up at pip.verisignlabs.com, created an OpenID account, and then I added it. Now, they call it a certificate, though.

Steve: A credential.

Leo: A credential. So I added my credential, my little key, and now it's working with that. So that's really cool. So now one key works with OpenID now, too.

Steve: Yes. So now you have very strong, not only something you know but something you have, two-factor authentication. And so when you go to a site which is using OpenID - and in fact VeriSign Labs has a list of all the sites that are currently supporting OpenID. And these are not sites that support VeriSign. They just support OpenID. So, for example, when Microsoft is, with Vista, supporting OpenID through their UI, you'll be able to use that. It's also worth mentioning that there's a nice Firefox extension - I know what a fan of Firefox you are, Leo - called "SeatBelt." And SeatBelt allows you to automatically authenticate yourself with OpenID sites. And in fact I learned about this from the PIP VeriSign Labs website, where they give you a link to go get the SeatBelt Firefox extension because this is all interoperable.

Leo: Yeah, so I installed that. And what's nice is, once you log in to SeatBelt, you've got a little button on your Firefox toolbar that says PIP, and says your ID name. And now whenever I go to an OpenID site I just - it's one button, and it verifies. Oh, yeah, okay. You're who you say you are. We know you are. And boom, that's done.

Steve: Yes. It is very cool.

Leo: I just wish more people would support this. That's the only...

Steve: Well, you know, we're on the leading edge of this stuff. And I think it's very clear that as, you know, again, we have a chicken-and-egg problem. We need people to be asking for it in order for sites to go, okay, yeah, I guess. At some point I think it's easily foreseeable that a couple years from now OpenID will be the technology that won because it was completely open, it was well-designed, nobody's trying to make any money from it. It's just like, yeah, we want to be an OpenID provider. Or in the other case, we want to allow OpenID because we recognize it makes our site more easy for people to use.

Leo: Well, I'll make a little promise to you. I'm upgrading my Drupal, and Drupal will support OpenID. So at some point TWiT.tv will. I'd like to get my forums to do that. I'll find a plug-in to do that. And I know that - I do one WordPress blog for Munchcast. We'll turn on the OpenID plug-in for WordPress there. So I'll do my part to make my sites as OpenID-compatible as I can. Because [indiscernible] idea, I mean, this is great.

Steve: Yeah, it's the right thing to do. Also there is an information card feature now. I mean, I'm watching these features evolve on PIP just over the last few weeks. For example, you just registered your cell phone as a secondary authentication, where they will send you an SMS message in order to verify that - again, something you have is the second factor of authentication - that you're in possession of your cell phone. So here again, that feature wasn't available just three or four weeks ago when I was doing this research. And now it's online and up to speed. So this stuff is happening very quickly. But, for example, if I were to, after getting everything set up with VeriSign Labs, if I were to go to one of these sites that supports OpenID, I identify myself, for example, in this case as stevegibson.pip.verisignlabs.com.

Leo: Now, that's okay for you to give that out.

Steve: Yes, because there's no way that anybody else can authenticate under my name because this will come back to VeriSign, and VeriSign will say, okay, what's the six-digit code currently showing on any one of the fobs that you have registered?

Leo: That's really neat. I really like - this makes, to me, OpenID really secure. And I really like that. And so you get the convenience of that one-time sign-on. I mean, now with Firefox I'm signed on. So it just does it automatically. But it's only while I'm here. And as soon as I close Firefox, it's gone. And unless they have my fob, won't do it.

Steve: Now, what I'm hoping is that, while this is in beta, we've got this somewhat cumbersome URL, you know, stevegibson.pip.verisignlabs.com. What I'm hoping is that they've already nailed down some very cool top-level domain, you know, like VIP or maybe just PIP or MyID or something cool so that, when this comes out of beta, all of our existing accounts will also be valid at a much shorter URL, so it's stevegibson.vip.com or MyID or something like that. That would be very cool.

Leo: Because it's still a beta program, it's still at verisignlabs.com. But I presume they've got PIP or VIP or something.

Steve: Now, harsh as we've been about U3 in the past, it's also worth mentioning that that is another option that VeriSign offers. That is, you can install - I couldn't quite understand whether it was something you had to install or was already present. But they do also support the SanDisk USB Flash Drive in U3 mode. So apparently there is something that they feel is secure enough that you can use U3, and there is a VIP credential software that is tied to the USB Flash Drive hardware that makes it secure enough that that's another means of authentication, if you didn't want to use the fob.

Leo: So you would somehow put something on your SanDisk U3 key, which I have one in front of me right now. That would be in lieu of logging in. It would be your physical identification.

Steve: Yes. And now of course the problem...

Leo: Now it's a dongle, by the way.

Steve: Oop, you're right, it's not a fob, it's a dongle.

Leo: So if you want to use a dongle instead of a fob, you can do it that way.

Steve: The problem, of course, is that you would need to stick that into whatever computer you were visiting. And so that's a little more invasive and intrusive than just having this fob on your keychain for your authentication.

Leo: Right, right. I'm just intrigued by this whole system. By the way, once you sign up for your PIP account, you can add fields. You can create an information card with information in it that you can then use. For instance, I always use as a nickname "Chief TWiT" when I create a new account. So it can actually populate that automatically. You could say, yeah, use Chief TWiT as my nickname.

Steve: Yes, exactly. And in fact the information card is essentially what we were talking about over on CardSpace under Vista, where you're able to create information cards to contain different sorts of information. The cool thing is that this information is not stored on your machine. Rather it's stored, in this case, on VeriSign's servers. So you authenticate yourself to VeriSign. Then you log in with your VeriSign OpenID. And then VeriSign provides the information directly to that third-party site that you have previously given to VeriSign with the instructions to provide that along with your OpenID.

The point is that this begins to make it substantially more secure to use computers you don't have control over. You could go, for example, to the library and do something that you would normally never do on a library computer, for example, because none of your information ever goes into that local computer. It's being handed off between VeriSign's server and the site you're visiting, rather than to that local machine.

Leo: I'll give you some other advantages. They're really doing this OpenID right. They have a list on the VeriSign Labs site of your trusted sites, sites you've used this OpenID to establish a relationship with, and your activities. So, for instance, I tried to log into the site with the wrong password. It failed. I have a record of that and when it happened. So I

think this is really a nice implementation of OpenID. They also allow you to do your own SiteKey, to upload an image that only you know what it is, and presumably only you have access to, and add additional security by doing that.

Steve: Well, as a matter of fact, Leo, I'm looking at my page, and I've got that smug-looking picture of me...

Leo: No one else has access to that.

Steve: That cartoon that you had done for Security Now!. And I snipped it out and made it the right size. And so the point is that, you know, when I log in, this photo of me is showing. And that's additional verification that I'm actually at VeriSign Labs and that there's been no sort of spoofing going on because nobody else would have access to that picture. Oh, and I will mention also that I smiled. When you try to go there over a nonencrypted connection, that is, over just HTTP, it won't let you. It immediately moves you into an SSL connection so all of this is protected and is locked up by their certificates.

Leo: Excellent. Excellent. It's well implemented, I have to say. And I've been critical of VeriSign in the past, at least as a registrar. But they've done a nice job here. Done a nice job.

Steve: Yeah, I really agree. I think that this is - anyway, I wanted to bring it to our listeners' attention because, you know, here is a functioning, working, strong third-party verification system that is, I mean, it's only running OpenID. That's the whole point of this is to be an OpenID identity provider and authentication service. And it uses our existing PayPal Security Key and additional ones, and has other means for doing multifactor authentication than just the fobs.

Leo: Right, right.

Steve: So it's a win.

Leo: I like it. You started this page a while ago, the Perfect Password page.

Steve: Perfect Passwords. And it's turned out to be popular Perfect Passwords.

Leo: Yes, it has.

Steve: About 3,000 sets of them are pulled every day. And I hear from people all the time who just love the idea. I went to extreme measures to make this safe to use. Similar to VeriSign, you cannot display that page over a non-encrypted, non-SSL connection. If you try, it switches you into that mode before it goes any further. I also have a bunch of headers and code on the page that prevent the caching by any servers, even if the server could intercept it, which it probably can't because it's over SSL and so it's not going to be running through a proxy. So, I mean, it's really, really safe.

Now, I was of the opinion that using RSA's pseudorandom number generator would be all I ever needed. Then, after I implemented that, I wrote a bunch of code to take it to the next level. Well, I had thought that I put that code in place. But it turns out that I had written it and assembled it, but never took it online. So by just some bizarre coincidence, someone in the security field thought, I'm just wondering, you know, how random Steve's passwords are. So he wrote a bot to pound on the password page, that is, to pound on the Perfect Passwords page.

Leo: Good for him.

Steve: And I knew something was going on because I saw this happening. And it was over - there was, like, an initial little burst of insanity when, like, somebody was pounding on the page. And then a couple days later it went on for 24 hours. So this guy collected a ton of the passwords that were being presented by the page and then ran them through a statistical randomness tester. And he came to the conclusion that these were not as - there wasn't as much entropy in the resulting passwords as possible.

Leo: Oh, no.

Steve: I know. I mean, okay, now, this doesn't matter.

Leo: They're entropy enough.

Steve: Well, and as we know, it doesn't matter that, if you got 100,000 of these and did an analysis of statistical entropy, whether I'm at 100 percent or 99.997 because...

Leo: Or even 75 percent.

Steve: I know. Because what you want is a password that you cannot guess. And certainly, if you've ever seen any of these, they're just insane passwords. So anyway, so the trouble is he created a blog posting somewhere where he said - he talked about his analysis. I mean, he's a nice guy. He and I have corresponded several times since. His first attempt to process the passwords had some of his own bugs in it that made it look like it was really bad. And that, you know, but then he fixed his own bugs. And then he said, okay, this is not so bad. But still it's not perfect. So it's like, okay. Maybe it's not Perfect with a capital P. But again, you have to have, like, 100,000 of these in order to detect any variation from them not being absolutely random.

Leo: It's Practically Perfect Passwords.

Steve: Well, I mean, it's not even potentially perfect. They're essentially perfect for the...

Leo: Good enough.

Steve: For the reason that we're generating them. But I thought, okay, fine, I'm going to fix this. So what I did was I finally implemented the algorithm that I had written but not

implemented, which I think our listeners would find interesting because it brings to light something that we haven't discussed before. First of all, I promised people that we would never be issuing the same password twice. Well, given the fact that this thing is a - it ends up with a 256-bit result, it's very certain that we haven't done that. But...

Leo: You haven't wrapped around yet, in other words.

Steve: Well, you know, my god, I mean, 256 bits, I don't even, you know, it's...

Leo: That's a ways off.

Steve: ...a ridiculously big number. But what I decided to do was to implement a system where I would have a monotonic counter, meaning a counter that only goes up, never goes down. Every time I increment the counter, I store that value. And I'm actually storing it in a registry key on the GRC server. And many different processes in my server use this. For example, it's the way my ecommerce system generates the thing I call a "cryptoken," which is a nonrepeating token that is used in order to hand our individual ecommerce users a token that's only good once. And so anyway, I'm using the same thing with the Perfect Passwords page. And I'm using the state-of-the-art AES, that is, the Rijndael encryption, which is 128-bit block cipher, meaning that it takes 128 bits in. And as we'll remember from symmetric block ciphers, it turns that, using a secret key, it turns that 128 bits into a different 128 bits. And it does so reversibly, meaning that, when you encrypt it, it goes one way, but you're able to give the same key to the matching decrypter, and it will reverse the process.

Well, and this is the problem with using a symmetric block cipher is, okay, so I've got a counter. So imagine that the counter starts at one. And I encrypt that. Oh, and this is a 128-bit counter. The counter is actually 256 bits long because, you know, why not. But I'm using the least significant 128 bits of that 256-bit counter because Rijndael takes in its 128-bit block symmetric cipher. So imagine that my count is one, and I encrypt it. Now, it's going to turn it into basically 128 random bits. I mean, that's a good - good crypto does that. It turns something not random into something random. It's not compressible. It's highly random. Then I increment my counter to two, and I encrypt that. Well, that's going to turn it into a completely different-looking 128 bits. So together we've got these - now we have 256 completely random sets of bits.

The problem is that an attack on this would be, if you were crazy enough to try to brute-force the key, if you knew that it was a counter, that is, a simple monotonic increasing counter, that was generating these, what you could do is take the output and start trying to decrypt it with every possible key. And if by some chance you hit on the key - and again, this is a 256-bit key that is driving the mapping between the input 128 bits and the output 128 bits. But, you know, okay. Technically, if you knew that this was a counter on the input, then all you would have to do, you would know when you found the right key because successive decryptions of successive 128-bit outputs would suddenly result in two numbers that were one off from each other because the input was just a counter.

Now, in my case I'm using just a counter. But in many other instances where you're encrypting something you've got, for example, an IP packet where you've got a header that isn't changing, or a header with known fields. And we've talked about this earlier in our crypto episodes. So there's a problem with using symmetric crypto like this any time you either have repetitious data, as you would, for example, in the similar header of every IP packet that was being encrypted; or if you've got predictable input data, as is the case in my use, where I have a large counter which is feeding data into the crypto.

The way you solve this problem is what I wanted to talk about, which is called CBC, Cipher

Block Chaining. And this is, again, something super clever that the crypto guys came up with that prevents the reverse engineering through a brute-force attack like I've been talking about. And it's actually very simple. The idea would be, say that we take my first value, when the counter is one, and we encrypt it, and that gives us our first 128 bits of random result. That's our first output. But now we use that, and we XOR that using the exclusive OR operation, remember, and we've talked about this, where basically it's conditional bit inversion. We XOR the first output with the next counter value. That is, the counter counts to two, we XOR that two, and that's what we encrypt.

Well, now what we've encrypted is no longer predictable. It is not predictable from a standpoint of being able to use brute force approaches to determine what our input was, no matter how many keys we try. And so the idea is you just keep this process going. You take the second output. You XOR that with the third input to create your third output. And what's very cool about this is, if you start at the beginning, this is a reversible process. But it's only by having the proper starting point that you're then able to move this forward in this iterative fashion.

Anyway, I've got a diagram of this algorithm on the new Perfect Passwords page. I don't have it up yet, Leo, at the time that we're recording this. But I will by the time that we hear this because I wanted to completely explain to people what this new algorithm is. And rather than asking our intrepid security researcher to suck hundreds of thousands of new, different algorithm as perfect passwords off the page, he's asked for, I think it was 16 meg of output. So I'm providing him with a file of 16 meg of the new random data for him to analyze so that he will revise his page that he posted on the blog saying that GRC's perfect passwords were not a perfectly random as, you know, given 100,000 of them and an analysis of entropy, as they possibly could be. And I ought to also mention that he has continued using them even after his blog posting. Because he says he has tools in UNIX that would generate similarly random stuff, but it's just easier to go to GRC.com.

Leo: It's easier. So now he trusts you.

Steve: Yeah. Well, he always did. And actually he got - it was really his, as I understand it, it was the early coding mistake that made him believe they were horribly nonrandom.

Leo: And I should emphasize "his" coding mistake. Not yours.

Steve: Exactly. His coding mistake. And I thought, okay, while I'm at it, I mean, I want to remove this blogging post from the 'Net because, I mean, we have had people write to us and say, hey, what about this blog, this posting this guy made? Are your passwords safe? And it's like, yeah, look, he's still using them, they're so safe.

Leo: So they're all right.

Steve: But it's like, okay, fine, maybe I guess RSA's random number generator isn't as good as it's possible to make one. So now mine will be.

Leo: Yes. What a relief. I now feel safe and secure. You know, anytime I see a random, well, even nonrandom, but random-looking hash of 64 letters and numbers and stuff, I figure that's good enough.

Steve: There is no way anybody is able to guess it. It's in no dictionary. And now that I'm

using a counter, a monotonically increasing, never repeating counter, there's no chance that we'll issue the same thing twice because that count will never be the same again. And in fact, I have a - oh, I ought to mention. I forgot to talk about the initialization vector. Oh, horrors.

Leo: Oh, well. How could you not mention the initialization vector?

Steve: I know.

Leo: Shocking.

Steve: It's no good for even the starting point to be known because in my example I start the counter at one, and I encrypt it. So what if somebody knew that, and they got the very first 128 bits that came out and then just started looking for keys in this huge 256-bit key space, and they found the key that would decrypt that first thing back to the number one. Then they have, again, they have found the key. Well, the fact is nobody knows what the counter is because I've been testing it a lot, and it's up in some number somewhere, so there's no way to know if you got it right. But you could say, wait a minute, the count is probably still mostly zeroes in the most significant bits. Right? So you get a security key - again, you brute-force, reverse-decrypt the first 128 bits until you get a number for the counter which is small, and you could then use that to test whether your assumption that that's correct is in fact valid. Anyway, the initialization vector prevents that. It is an absolutely random 128 bits that you first XOR with the first counter value. Then you encrypt that, and that's what you use for XORing from then on.

Leo: So you don't start with one. You start with some random number, basically.

Steve: Exactly. And it's actually - and that's what the initialization vector is. So I've got 128-bit initialization vector. I've got an unknown 128-bit counter value and an unknown, that is to say secret, 256-bit encryption key for the Rijndael state-of-the-art AES encryption. I think we're safe.

Leo: Whew. Okay.

Steve: And people are using it to protect their routers from neighbors who are trying to hack into them by mistake. So I think this is plenty of security.

Leo: More than enough. Given its application, it's probably overkill. But there you go, you have it.

Steve: Why not. And 3,000 people a day like to get passwords there. And now I've decided I'm going to show the algorithm I'm using and make everybody happy about having obtained the highest level of entropy ever known.

Leo: Yeah. It's cool. That's really cool.

Steve: Yeah, it's doing it right.

Leo: Ah. Well, Steve, once again you have - you noticed I got very quiet during that description. I was thinking. Yeah. No, you've made it perfectly clear. And it's actually a fascinating area of computer science is this whole issue of pseudorandom number generation.

Steve: And crypto. I love that stuff. And for people who have had their eyes cross a little bit by listening to this, do check out the passwords page. It's just GRC.com/passwords. Down at the bottom, under a new section called "The Techie Details," I have a block diagram of the algorithm and describe it further, just because I thought people would get a kick out of knowing.

Leo: Yeah, it's fun. And I have to say that you don't have to use these, but the whole idea of playing with it is just great. It's just really a neat thing to have. Who knows where you might find an application for a 64-character random string? Just you never know.

All right. We're going to wrap this up. Also, if you want to follow this a little bit more closely, you can also read the transcription, which might help a little bit. Go to GRC.com, and full transcriptions of all of our shows are available there, as well as 16KB versions for those of you who want little tiny episodes, maybe because your bandwidth isn't sufficient or you just want to share it with friends. GRC.com, that's also where you'll find SpinRite, Steve's great disk recovery and maintenance utility, the best ever, still #1, no challengers. SpinRite.info, if you want to see some testimonials to prove it. And also his free programs, like ShieldsUP! to test the effectiveness of your router. Whenever I set up a new router, first place I go is ShieldsUP!. You can also download great programs like UnPlug n' Pray, Shoot The Messenger, DCOMbobulator, and the ever popular, although now we know useless, LeakTest. Just for old times' sake, download it.

Steve: I actually received a piece of mail, when I was reading the mail last week for the mailbag episode, some guy saying, well, if LeakTest is so useless, like you told us all it was the week before, why are you still offering it? And the problem is it's well known, it's linked to all over the Internet, and some 7 or 800 people a day download it just because they want to make sure that their firewall is doing the right thing, even to...

Leo: It's harmless. It's not like it's dangerous.

Steve: Right, doesn't hurt you at all. And it's additional verification that just changing the name of a program won't allow your firewall to be fooled. So it's there because it's still useful.

Leo: Excellent. Excellent. Steve Gibson, wonderful job. Good luck with the fumigation. We hope the tent comes off, and the house doesn't fall down.

Steve: And I'll talk to you next week, Leo, for episode 109 [sic], which will be another of our Q&A episodes, which I actually get a lot of email from people saying they love the Q&A. So we'll plow into that next week.

Leo: Excellent. We'll see you next time on Security Now!.