WiFi Protected (In)Security

Description: After catching up with only a small bit of the week's security news, Steve and Leo discuss the recent revelation of a fundamental security flaw in the functioning of the WiFi WPA standard. WiFi Access Points, following the certification-mandated default configuration, allow an attacker to obtain network access within just a few hours.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-335.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-335-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 335, recorded January 9, 2012: WiFi Protected (In)Security.

It's time for Security Now!, the show that protects you online, your privacy, your security, and also teaches, I think, a lot about how things work. And that's because this guy, Steve Gibson, is a great explainer. He's the Great Explainer, the Explainer in Chief, from GRC.com. Hi, Steve.

Steve Gibson: I like "Explainer in Chief."

Leo: I know. I'm going to call you that.

Steve: That's better than "the man, the myth, and the legend."

Leo: I'm going to say it that way from now on. I've got a new title for you. Steve and I are up early - it's 9:00 a.m. roughly, Pacific time - because I'm headed off to CES this week. So just a word of warning. For those of you listening to the recording, you may have noticed you got it early. And we recorded this on Monday, so anything that happened on Tuesday and Wednesday, which would normally be in the show, will not, and we'll just cover it next week.

Steve: Such as the second Tuesday of the month.

Leo: Which is tomorrow.

Steve: Patch Tuesday, the first patch round of 2012, is tomorrow. And we don't have full details from Microsoft about what they're doing. As they do, they preannounced sort of the number and the criticality of the patches. So that we know about, and we'll talk about that briefly. But we don't know in super detail what it exactly is that they've done. So, yes, as you say, we'll deal with that retrospectively next week, since this is coming out, we're doing this on Monday before Patch Tuesday for a change.

Leo: All in good time, as the Wicked Witch of the West...

Steve: So, yes, as Explainer in Chief today I'm going to have some fun. We're going to, as we promised last week, plow into the details of a new and very worrisome revelation which it turns out has been suspected and even exploited for a long time before it came to light, as it recently did. And that is the fact that it's possible, using the simple security setup protocol which any Wi-Fi Alliance, WiFi certified device must support, it's possible typically in only a few hours to hack into the security of a WiFi access point and obtain the credentials, get on the network, and do all the bad things.

Leo: You'd have to be sitting on the curb, though; yes?

Steve: Yeah. You would need to be within range. And in fact it turns out you need to have a pretty strong connection. There's a lot of associating and disassociating, a lot of failed attempts. And so what the people experimenting with this are finding is that they have to have a pretty strong signal from the access point. But if you - in a typical apartment these days you look at all the networks which are "within range," and more and more of them are now of course saying that they are secured. The problem is, not only does this technology have to exist, but in order to be certified - get this, Leo - it must be enabled by default.

Leo: What? Certified by what? By whom?

Steve: By the Wi-Fi Alliance. In order to be able to have on your access point that you are WiFi certified, to be able to use...

Leo: Not WPS certified. WiFi certified.

Steve: Yes. To have WiFi certification you have to support this easy setup protocol, which we now know is vulnerable, and it has to be enabled by default.

Leo: Now, their hearts were in the right place because the whole idea of this is, oh, my god, nobody knows how to set up WiFi securely. If we put a button on the access point that you press that sets it up automatically for you, everybody's going to be

more secure. So, I mean, they weren't - this was - the intention was good.

Steve: Well, actually I have the 154-page specification in front of me.

Leo: Oh, boy. Oh, boy.

Steve: I spent the weekend reading it.

Leo: Oh, man.

Steve: And I understand how this happened. That is, I understand the nature and reason for the mistake that they made and the way this slipped by.

Leo: Aha, okay.

Steve: Which we're going to get to in the podcast.

Leo: So it is one of those things like WEP where the best intentions went awry.

Steve: Yeah. Yeah.

Leo: All right. Well, we don't have any commercials today. So you might as well just launch right into it.

Steve: Okay. So I briefly mentioned that we have a Patch Tuesday coming up relative to this week's recording. It's tomorrow, on Tuesday. And there are seven security bulletins which address eight vulnerabilities in Windows and some dev tools. So it's a relatively small one. One of those is critical. We don't yet know from Microsoft whether they've addressed the SSL/TLS vulnerability. They were going to do it last month but pulled it at the last minute because of some compatibility problems that they discovered. So maybe they'll do that, and maybe not. But we'll tell everybody next week what it was that they got. But do be aware that our systems are going to be saying, oh, there's new patches available, and probably a good thing to do.

There wasn't much other security news. I did get a kick out of a Forbes article that talked about Apple having a patent on - get this, Leo - a password recovery charger.

Leo: Charger?

Steve: Like, charger.

Leo: Like that, like, plug-in thing?

Steve: Yeah. The idea is - and Apple patented this.

Leo: Now, we should say they patent things all the time. Doesn't mean they're going to make them.

Steve: Yes, they do. We may see it or not. The idea being that somebody who is out and about with their laptop, who forgets their laptop's password, can come home, plug their laptop into their charger, and their charger will enable password recovery. So the idea would be that something about the charger is able to authenticate itself. For example, maybe the charger would have a serial number, and all of those are unique, and so your password is saved in the laptop encrypted under the charger's identity. So only when your charger is remated with that laptop do you have the ability - so basically the charger would be another factor of authentication.

Leo: Oh, that's not a bad idea.

Steve: It would be - we've talked about multifactor authentication. And so this patent that has been seen would allow a specific charger to be another factor in multifactor authentication for the sake of allowing lost password recovery.

Leo: Of course, everybody leaves their chargers in hotel rooms, and I can just see the problems this is going to cause, but...

Steve: Exactly. So it's like...

Leo: ...it's an interesting idea.

Steve: ...someone had an idea. Someone said, oh, patent that.

Leo: Yeah. And that's what happens. They've got a whole building full of people to do that.

Steve: Yeah, got to keep those patent attorneys busy.

Leo: Yeah, exactly.

Steve: Now, I didn't want to go into much detail on this mass SQL injection attack which is underway. But it's now up to about 200,000 URLs have suffered yet another SQL injection. And we've talked about this ad nauseam in podcasts past, so we won't go into

it in any more detail. But the problem being, of course, that many websites are driven by an SQL database backend, and their scripting technology pulls pages, dynamically assembles pages and so forth.

For example, many online forums have a database containing all the forum postings, all of the chained comments and everything that displays these pages. And the problem is that the act of displaying the page interprets valid SQL queries which are sent to the database and return page content. And unless the designers of the websites are very careful, the content being returned is interpreted also. And that allows bad guys to do what's called "injection," that is, basically, the bad guys figure out the names of your tables, the column names, the record names and so forth, and they're able to post to a forum their own SQL commands to display their own content, which the website owners don't want and, for example, spread infections or redirect people to foreign websites, basically get up to all kinds of high jinks that way.

So this is a - it's a convenience in the design of many websites which has been widely adopted, and the bad guys keep finding new ways around the filters which these technologies try to put up. Essentially, the whole idea is prone to security problems. So as we know, the best security is that where you only allow things to happen that you intend, rather than needing to block the things that you don't intend. And unfortunately, the architecture of this SQL-derived backend requires that, by default, everything is allowed, and the misbehavior is blocked. Which always means there's an opportunity for new ways for the bad guys to find ways around the blockage. So that's what we're seeing again. So there's another wave of this going on.

I had a note to myself from last week. I mentioned that - we were talking about hard drives versus SSDs. Remember, one of our Q&A questions was, Steve, what do you use, hard drive versus SSDs. And I mentioned that I only use SSDs on laptops because they're prone to being bounced mechanically, and even that I'm setting up a new server at GRC that'll be SSD-based. And of course the problem with SSDs that makes me nervous is that we know that they get fatigued with writing. That is, the act of writing to them fatigues them. And I made a comment that it's because we're essentially deliberately breaking down the insulation on a little floating island on which we have stored an electric charge.

And so that little - the island is insulated from everything else. And the nature of MOS - Metal Oxide Semiconductor - technology is such that we can electrically sense the charge on it. So in order to set that to a one or a zero, we use a high voltage which is generated inside of these little flash chips in order to overcome the insulation resistance and pull electrons off or put electrons onto this little floating island. Thus, after that's been done enough, we begin to weaken that insulation layer, and that's the way flash memory dies over time. Consequently, flash memory has, in the same way that hard drives have spare sectors which are being swapped in as defective ones are swapped out over time, flash memory has the same sort of architecture.

The point was that there are two other alternative, nonvolatile, solid-state, high-speed technologies on the horizon, and it's not one of those, oh, yeah, by 2050 we ought to have that. I mean, HP has patented something that's called a "memristor," which sort of sounds like "resistor," but this is a resistor which changes its resistance as a function of its history. And they're talking about having production levels of memory, like, next year. So, and this doesn't have the problems that NAND Flash memory does that I was just talking about.

Leo: You know what cracks me up is for years we said, oh hard - I remember saying this in 2000, maybe even earlier - hard drives would be dead by the year 2010. Everybody will have holographic storage or some other form of random access storage. It took us a long time. So when you say it'll be here this year, it's like, well, yeah, but we've been waiting 10 years for something like this.

Steve: Right. And there's another technology called phase-change memory that - I had a link to it, and I tried to find it again, and I couldn't find it again. But so my point was that these are - I'm surprised when I see the sober-sounding, not only is this working in the lab, but we're in preproduction, and we should have it soon. Which is not to say, as you commented, Leo, that hard drives are going to disappear because certainly the developers of these new advanced memories are going to want to get a premium for them. And just in the same way that flash is still far more expensive than a hard drive, I mean, my god, you don't want to have a terabyte of flash memory; whereas a terabyte hard drive is 70 bucks now.

Leo: Well, and of course the other thing that I think nobody counted on - I'd be curious what you thought in 1995 or 1999 - but is how much density, how much data density that hard drive manufacturers were able to jam on those disks. I mean...

Steve: Yes.

Leo: ...they really over-achieved.

Steve: It's been amazing. And Leo, it's also the way the cost has come down. I mean, we were hearing about vertical recording technology for a long time. And it's like, instead of recording horizontally, where you lay the magnetic field down flat on the disk, they were saying, yeah, we're going to - instead of having horizontal magnetic domains, we're going to stand them up on end. It's like, what? What? And it did, it took a decade from the time they had that, IBM had that happening in the lab, until you could go over to Fry's and buy one. But it's here now. And that's the way the density has gone up as high as it has. But not only that, but the cost has come down just amazingly.

I looked at the underside of a drive that I have had around, but purchased in the last few months, and even the PC board on the bottom of it, it used to be that they were, like, they took the whole bottom surface of the drive, and maybe was a couple layers. Now it's like a postage stamp, just big enough to mount the connectors for the drive. Everything has gotten so integrated that it's like, there's not even any parts left anymore. Just incredible.

So, okay, so that's all I wanted to talk about. I did want to remind our listeners that SpinRite is the reason I'm able to do the podcast, frankly. I had a nice note from two guys, Carl and Joe, in Melbourne, Australia, who sent me email saying "You Saved an Artist's Life." And then he said "/SpinRite Success." He said: "Hi, Steve. I'm an avid listener of Security Now!. So when my younger brother Joe called saying that his hard drive, as he put it, had carked it, I knew exactly where to send him. He graduated last year and has been really motivated to get stuck into his artwork." They talk a little funny in Melbourne, I guess.

Leo: Yes, they do. Yes, they do.

Steve: "...to get stuck into his artwork to kick off his career." No offense, Carl. Anyway, he said, "His dead drive had potentially lost him years of work in progress which would be his ticket to getting his career underway. But now all that was threatened. He purchased, downloaded, and ran SpinRite, but he had trouble getting it to his computer, which wouldn't boot from a USB and didn't have a floppy drive. And the computer he downloaded it to also didn't have a CD burner. So he borrowed a USB from a friend, copied SpinRite to it, took it to his friend's place to burn a boot CD. I actually wasn't surprised when he told me it worked. He's now backing up all his precious artwork to DVD" - ah, he learned a lesson - "and getting on with his new career. Thank you for saving an artist's life work. Thank you for all your work on this wonderful product and for actually promoting it on your podcast. Cheers, Carl and Joe."

Leo: Yay. Yay.

Steve: So thank you, guys.

Leo: A nice story.

Steve: Okay. So here's the deal with what's going on. A couple weeks ago a security researcher by the name of Stefan Viehbock released what he thought was his discovery of a new way to brute-force an aspect of all of our WiFi networks that had not been done before. Well, it turns out it had been done before. It had been done for, like, more than a year, but never publicized, by a company called Tactical Network Solutions. Now, this is one of the nice things about security researchers who discover things and go public with them, is we always wonder, well, does the NSA know about that? Have they been doing this all along and only now, now they're, like, going, darn it, now we're not going to be able to do that anymore because everybody else knows about it.

Well, this is an instance like that. This Tactical Network Solutions company - it's TacNetSol.com - they admitted when Stefan went public that they had very mature attack code that they'd been "using," whatever that means, for quite a while. And so they were able to almost instantly put their code up on Google, where it is sitting, their tool they call "Reaver." They're apparently fans of the "Firefly" series.

Leo: Yeah, those were kind of the zombies of "Firefly."

Steve: Exactly. They'd sort of - they were out on the fringe of human space and had sort of reverted to animalistic sort of tendencies. There were, like, legends of them, about the things that Reavers did to you if you got caught by them, and none of those were good.

Leo: No, no.

Steve: So you didn't want to let the Reavers get you. And along that vein, not

surprisingly, there was already another tool called "Walsh." Walsh was the wise-cracking pilot of Serenity, which was the name of the "Firefly" spaceship. Walsh is a scanner. So as we would have predicted, first there's the attack tool, and now there is a vulnerability scanner which you can use to scan all the access points within range to see if they have this WPS enabled.

Okay. So I'm sure we'll be talking about this to some degree over the next few months of the podcast because this is a, well, I guess we would call this, Leo, a target-rich environment that has been revealed. The idea is that all access points and WiFi devices in general want to have themselves certified by the Wi-Fi Alliance. The Wi-Fi Alliance is the standards body which was created to unify the WiFi market. Now, that's all good because we don't want interoperability problems. And these protocols are incredibly complex. So if there wasn't a single weighty body that everyone had to check in with and had to, like, verify their own drivers against, their own access points against, we'd have a problem like we used to have in the old days where it's like, well, this device works with this one, but it won't interoperate with this brand because they didn't quite implement the protocols the same. We've had those nightmares in the past.

And so I'm very glad that this Wi-Fi Alliance exists as a single point of reference that tests devices to certify them. And only if these devices pass interoperability tests do the manufacturers get to use the copyrighted and trademarked logo that says this is WiFi Certified. One of the things that the Wi-Fi Alliance wanted was ease of use. And in their spec for the operation of their WiFi devices they have in Section 1.7, "User Experience" - I'm going to be quoting from the spec here because it's worth understanding the mindset that they had. They said, "The most important characteristic of any initial setup solution is the user experience." Okay, now, listeners of the podcast might feel that security would be an important characteristic of any initial setup solution.

Leo: Yeah.

Steve: But not so much here. User experience is the most important characteristic.

Leo: The experience of security, not the actual actuality.

Steve: We'd like to give you the feeling, oh, look...

Leo: It feels secure.

Steve: I just press a button. And somehow this has all been done for me. So they said, "This section introduces two scenarios to illustrate the WiFi Simple Configuration" - that's WSC - "user experience." Now, this is the good experience, not that where you're getting hacked because of the WiFi Simple Configuration, we should explain.

Leo: Yeah, oh, yeah.

Steve: This is the happy side. So they said, "Sections 5, 6, 10, and 11 contain a more detailed specification of these and other scenarios." So here we have the first, they call -

the so-called "in-band setup." "Setup steps: One, user turns on the access point."

Leo: Okay.

Steve: Okay, that's easy.

Leo: Seems sensible.

Steve: "Two, software on the cell phone automatically detects the access point and asks the user if he wants to configure the access point. Three, the phone prompts the user for the access point's PIN, found on a label attached to the device. The user keys in the PIN, accepts the default settings and clicks okay a few times, and receives confirmation that the access point has been successfully configured on their device.

"Now, the user brings home a wireless printer and turns it on. The phone detects the new wireless device and prompts the user to add it to the network. The user reads the printer's PIN number from its display and enters it into the cell phone. Both the cell phone and printer provide visual confirmation when the printer joins the network." So this is the spec giving everyone this warm, fuzzy feeling about how easy this can be to operate.

And here we have a second example. "Context 2: The user has a portable game console that he wants to connect to the existing wireless LAN for online gaming. This user prioritizes convenience over security." I'm reading this from the spec. "So he decides to use the pushbutton configuration method for setting up the portable game console. Setup steps: One, user presses the PBC button on the game console. User presses the PBC button on the access point. The game console and access point display the progress of the PBC method on their respective user interfaces. Upon completion of the protocol, both indicate connection success."

So these are the scenarios which have been designed into the standard. So the standard, and that is to say, in order to get this WiFi compliance logo that everybody wants - in fact, arguably, at least on the protocol level, needs - this WSC, WiFi Simple Configuration, must be part of the device. So access points must have it; all contemporary access points do have it. And that was renamed WPS, WiFi Protected Setup. So you may see WSC, WiFi Simple Configuration, or WiFi Protected Setup. They're sort of synonymous, essentially. So, and there's a - at the WiFi Org, which is the central repository for all this, are a bunch of specs and papers and so forth.

Now, there are four different modes for configuring this WiFi Protected Setup. We just heard in this example about buttons being available. And some late-model routers will have a "press this button." Sometimes it's on the back panel. Sometimes it's on the front. But that's a button which sort of tells the router to look for a device within its range that has also had its button pressed.

Now, this could be a virtual button. For example, Windows 7 supports this protocol so that, if you press the button on the router and then you go to configure your wireless networking under Windows 7, it will see that the router is broadcasting its support of WiFi Protected Setup. And so there is, in Windows 7, there'll be the standard "fill in your password," or it'll say, "Or just press the button, and we'll do this all for you."

So the specification indicates that the way this works is that the user presses the button on each end of the link that's going to be configured, and both ends from the time the button is pressed enable a pairing, essentially, for two minutes, which is - so there's 120 seconds they refer to as the "walk time," presumably for you to walk from one device to the other if they're not sitting at the same desk, or if they're not temporarily near each other.

So you'd press - you have a system upstairs, for example. It's all set up and established, wired in with its keyboard and display and everything, but your access point is downstairs. So you could start the process at either end, actually, start by pressing - on the UI screen under Windows 7, you click on "Yes, I want the simple setup." Then you have two minutes to go downstairs, press the essentially autoconfig, or one-button setup, whatever it's called on your particular access point. And as long as you do it within that time, your computer upstairs is still enabled for this, and the devices will pair, essentially, and you're connected with relatively good security.

Now, they recognize that there's a security concern here because a bad guy could enable pairing at the same time. And this is all sort of like "headless," is the term they use a lot. There's nothing that says on the access point, "This is the device I just paired with." It just sort of all...

Leo: That's kind of the problem, huh.

Steve: Yeah. So the way that's supposed to be handled is like - and again, a lot of security focus was given to this. One of the things that's sort of sad and funny about our whole situation we find ourselves in here is that there is, like, major crypto happening. I mean, we've got private keys and public keys and Diffie-Hellman key exchanges and secure hashing algorithms, I mean, this thing is just dripping in 128-bit crypto flying through the air. Yet they missed something, as we'll discuss in a second. So it's like, here this little processor in the access point is limping along, barely keeping up, trying to do the crypto which this specification requires. Yet despite the fact that this thing is just dripping in state-of-the-art crypto, the protocol that uses the crypto has a little mistake in it that we'll get to in a second.

Leo: Just a little bug.

Steve: Just a little problem.

Leo: Just a little problem.

Steve: So the way they handle this problem of somebody else waiting for you to click your access configuration button, whatever it's called on your particular access point, is there's supposed to be a light there. And if the access point or the computer, because both are supposed to do this, see more than one other pairing opportunity, they go into a lockdown, like flashing alert mode, and just say, no no no, we're not going to do any pairing...

Leo: [Alarm sounds]

Steve: Exactly, because there's more than one opportunity. So it's like, okay, I mean, listeners of this podcast probably don't get a real comfortable feeling from this. And here's the problem. How many times - we're at podcast 335 now. How many times have we seen situations where "simple" is the enemy of "secure"?

Leo: Yes.

Steve: I mean, yes, we would like to have a simple password, wouldn't we, that we could easily remember. We'd like to be able to use it on every website because then we wouldn't have to remember separate ones, blah blah blah blah blah. I mean, over and over and over, this whole - we could rename the podcast "Simple Is Never Secure." So here again they're trying to come up with a way to prevent people from having to enter a long passphrase. We know it has to be long because the one known vulnerability in WPA and WPA2, the state-of-the-art WiFi technology, is an offline attack that, if encrypted data is captured over the air, it's possible for a bad guy to take that, knowing the SSID of the WiFi network, and that's broadcast, so any bad guy knows where that is, that's the name of the network that you see when all of the available networks are listed, those are the SSIDs of those devices.

So you know that, and you know the protocol. You can perform an offline attack where you use as much processing power as possible to guess what the passwords are until you decrypt the data that you captured over the air. Then you go back to the network knowing what the password is. So thus we really need long, complex passwords on our WPA WiFi access points and networks. And that we all understand now. So the problem is, it's uncomfortable or awkward, I mean, look at the scenarios we just went through about using WPS and how simple it is. Oh, look, I brought a printer home...

Leo: Just push a button.

Steve: Plugged it in; and, oh, it's magically on my network. Uh-huh, right. So the push-the-button approach is one. But they're not happy with that, and it's not exactly clear why, except that they wanted another alternative. So they said in the specification: "The WiFi Simple Configuration in-band registration protocol is designed to provide strong protection against passive eavesdropping attacks and also to detect and to protect the system from an attempt to perform an active brute-force attack." Under device password they said, "All devices...." Okay, so here it is. This is the specification.

"All devices supporting WiFi Simple Configuration" - and remember, WiFi Simple Configuration is required for Wi-Fi Alliance Certification. So, "All devices supporting WiFi Simple Configuration must provide at least one numeric device password PIN for initial setup that is unique and randomly generated per device. Although it is possible and permitted for two devices to have the same device password" - that is, the same PIN - "a group of devices should not intentionally be assigned the same device password. And the device password MUST" - all caps - "not be based on other characteristics of the device such as MAC address or serial number."

Okay. So what they're saying is that, since the PIN is going to be short for convenience -

and again, this is why convenience is important. They're so worried about that that they don't even think users can type in eight digits without making a mistake. So the PIN, the eight-digit version of this spec has the last digit being a check digit so that, if the user makes a mistake entering eight digits, it doesn't even try to use the protocol to authenticate. Instead, the device you've entered those into is able to perform a checksum locally and go, oh, that can't be right, you must have made a mistake. So it's like, okay. So we have seen in the past situations where password-like things, for example, used a piece of the MAC address. I know that we've talked about this in the past.

Leo: Yes.

Steve: It's been done. And it's like, oh, how dumb is that because...

Leo: That's public.

Steve: ...the MAC address is in the air, exactly. So you just take a chunk of that and use that as the PIN. So they're being careful to say this PIN must completely be disassociated. But because it's so short, they know that they may - Linksys may, at some point in its history, reuse a PIN because they kind of ran out of them. So it's like, okay, fine, it doesn't have to be unique. But the idea is it's supposed to be unguessable and not tied to anything that is available on the device or in the air. So then they go on.

"Headless devices: Headless devices (those without a display) are required by WiFi Simple Configuration to include an eight-digit device password called a PIN. A PIN on a headless device is typically printed on a sticker or otherwise physically inscribed on the device. The PIN value of a headless device must also be configured into the device itself. This would typically be done during the manufacturing process. PIN-based device passwords are the basic security level for WiFi Simple Configuration. Since one of the digits in the eight-digit PIN is used as a checksum, the PIN contains approximately 23 bits of entropy.

This in itself is not the biggest limitation, however. The biggest limitation is that this PIN may be a fixed value when it is on a label. Because a fixed PIN value is very likely to be reused, it is susceptible to active attack. The protocol permits the user to override the default device password with a new value, which can help security-conscious users reduce this vulnerability."

Now, I read that to mean that that eight-digit PIN can be reprogrammed through the user interface. That isn't a requirement, but apparently it's an optional capability, probably not supported by most devices because that's going to create a tech support problem if the printed PIN on the outside no longer matches the reconfigured PIN on the inside. But that doesn't help us from a security standpoint anyway, as we'll see.

They say, "Probably the most significant class of headless devices in a WAN," that is, a wireless local area network, is the access point itself. If possible, an access point should generate and display a fresh PIN for establishing external registrars each time the registration protocol is run in the access point setup mode. So what they're saying is, it would be nice if the access point had a display which could randomly generate a new PIN. The problem is, that's expensive. That would be a little LCD that would have to be on the access point. And while, yes, it would give us more security, it would cost more. And

these things don't cost anything. We've talked about how incredibly inexpensive they are. So access points typically don't use that option. They use the printed-on-a-label static PIN, and that's all there is to it.

They said, "However, if a static PIN is used, the access point must track multiple failed attempts" - now, here, this is important - "must track multiple failed attempts to authenticate an external registrar and then enter a lockdown state. This state is signaled by setting the attribute access point locked to 'True.' After three failed PIN authentication attempts within 60 seconds, an access point must stay in the lockdown state for 60 seconds."

So they are saying that they recognize there's a brute-force attack problem with this whole technology. And so if three failures occur within a window of 60 seconds, the access point takes itself offline, essentially, for this kind of configuration for another minute and then reenables itself. Because, again, someone could type in the wrong thing three times, and we don't want to, like, lock them out forever. We're just going to make them wait a minute and then reenable ourselves.

Well, it turns out that, well-meaning as this was, many devices don't do this. But as we're going to see, even a device that follows this specification, due to a mistake that was made in the protocol of this negotiation, it still takes on the order of maybe a day. Many people, however, who are experimenting with this, are able to get onto WPA protected access points that have this enabled. And as I think I said before the show, Leo, I don't think I've said it yet in this recording, that enabling this is part of the specification. That is, this must be enabled, everything I've just read must be enabled by default in order for the device to be certified. They allow it to be disabled. But by default, to get certification, they're so concerned about ease of use that all of this has to be on by default in order to make it easy to use.

So they acknowledge that three failed attempts within 60 seconds will cause the access point to lock itself for a minute. Then it will unlock and allow people to continue. So here's part of where the problem comes in. Under "Devices With Displays" it says, "If an enrollee advertises support for the display configuration method, it is required to generate a fresh four- or eight-digit PIN each time it runs the registration protocol and show this PIN on a display."

And going on, it says, "This has two significant advantages. First, because the password is single-use, it is not susceptible to the brute-force attack described above." That is, if the PIN is printed on a label, obviously it's not changeable dynamically. So they require it to be eight digits in order for it to be long enough for brute-force attacks they assume to be impossible. And in fact, arguably, I did the math, I think it takes on average one point something, I think 1.1 years - oh, yeah, there it is. Oh, no. If all of this were done right, it would take a maximum of 6.338 years to try all seven-digit PINs, remember, because the eighth one you don't have to worry about. That's a checksum, which we can always provide correctly.

So it would take, given this - if you tried it three times in a row quickly and failed, then the access point would lock you out for a minute. So if we just, quick back-of-the-envelope calculation, that gives us three attempts essentially per minute. And if you - seven characters is 10 million possibilities, and we do the math, that comes out to 6.338 years for all, or an average of half that, as we know, if we're just guessing at random, there's a 50 percent chance that we're going to get it within half that time. So an average of 3.169 years to brute force. So they were thinking correctly.

What they're also saying, though, is that a device that has a display, for example, maybe

your printer is a more expensive device, and it's got an LCD display for all the other configuration and your paper out and so forth. So there they could afford to use a dynamic PIN. When you use a dynamic PIN, you only need to have four digits per this specification because it's going to be different every time, so you're not going to have the problem of it being static. And if three guesses fail, then you restart the entire process with another four-digit PIN. So you can't brute-force the same four-digit PIN over and over and over.

So they say, under "Guidelines and Requirements for PIN Values: The PIN requirements for the two main classes of devices are, A, headless devices, devices without a display, must use an eight-digit PIN, a PIN printed on a label attached to the device. The last digit of this eight-digit PIN is used as the checksum of the first seven digits." And then it says, "Section 7.4.1 specifies how the checksum is generated. Or, B, devices that use a display to show the PIN and can generate a new PIN must use either a four-digit or eight-digit PIN. The last digit of an eight-digit PIN is used as the checksum of the first seven digits as above," same section, blah blah blah. "A four-digit PIN does not include a checksum digit."

Okay. So we have a number of different ways which, under this WiFi Protected Security easy-to-use configuration, we're able to set ourselves up. We can press buttons at each end. Or the access point has an eight-digit PIN printed on it. Or if it's, for example, a device that's more expensive, that already has an LCD display, it can generate a dynamic PIN which only needs to be four digits. Unfortunately, it is as a consequence of the protocol's ability to work with a four-digit PIN that this whole house of cards tumbles. That's what happened. It's that this four-digit option allowed them to have a protocol which is not secure.

Now, looking at the protocol, it's a - we've talked about the way security protocols work in the past. The best example is SSL, where the client sends its initial packet containing all the protocols it knows about. Then the server selects from among those, hopefully the most secure of those available that it also knows about. Thereby the two ends are able to arrive at the most secure protocol that they both support, and then they go from there. So these protocols are like a back-and-forth negotiation between two endpoints that arrive at a result.

In this case, the protocol was very nicely designed. What I'm guessing, and it would be necessary to go back in time and read through endless committee meeting minutes, but it feels to me like this whole system originally used a four-digit PIN. That is, four digits was what they thought was going to be enough. And they would have told everybody you have to have an LCD because four digits isn't strong enough to prevent brute forcing. So four digits is what you're going to need. And then some guy raised their hand and said, "Uh, we don't want to have to have an LCD."

Leo: Yeah, this adds a lot of expense.

Steve: Yes. It's going to make these things too expensive. So we need to have a fixed PIN. And so then someone said, oh, hmm. And what happened was the protocol had already been designed around a four-digit PIN. And so at a later committee meeting it's like, wait a minute, this is a problem. Four digits, if we don't have an LCD, four digits isn't really enough for security for a static PIN. So we want to make it an option that you only use four digits if it's going to be a dynamic PIN, and double the length to eight digits if it's going to be static. So what they did was they extended the existing protocol. The protocol that was there had this handshaking go on.

And at one point, after a few negotiations of this is my version, here's a random number, here's my public key, the other side says here's my random number, and they throw in 128-bit random numbers from each end in order to prevent a replay attack, that is, to prevent someone who's eavesdropping from being able to reuse the same packets in order to reauthenticate. So each end always generates a new random - we know it's probably pseudorandom - 128-bit value which they provide to each other. And then their negotiation is based upon the combined 256 bits which will - it's never ever likely to appear again. That prevents a bad guy from just sending, sort of reinjecting the same packets that were used last time again in a so-called replay attack.

So each end sends these packets back and forth to negotiate. And at some point, and here's the brain-dead part, at no point is the eight digits sent. Instead, the end that is proving that it knows the PIN sends four digits.

Leo: Oh.

Steve: Then the other side says, yeah, you got those first four right.

Leo: Oh.

Steve: And then - and then...

Leo: Okay.

Steve: ...[indiscernible] thanks.

Leo: Okay.

Steve: And then that guy sends the second four. And then the other end, who is authenticating, says, yeah, you got those right.

Leo: So there's a significant difference in the difficulty of solving an eight-digit PIN versus solving two four-digit PINs.

Steve: Well, this is how it's broken, is that, yes, because what I believe is the original protocol was a four-digit protocol, they didn't revise it so that the eight-digit PIN was sent instead of the four-digit PIN. Instead, they said, oh, well, we want to allow four or eight. So if it's four, we'll only send four and then get confirmation. If it's eight, we'll send the first four, get confirmation, send the second four, get confirmation. Somehow they just, I mean, the phrase of course is "brain fart," where they just, like, didn't occur to them that the endpoint would confirm the first four separately from the whole eight. That's all this whole thing boils down to. So that means the attacker is able to - only needs to guess the first four.

And so, okay. So what that does is it's, obviously, how many combinations are there of four digits? Well, there's 10,000. And we know they're going to guess right in probably half, so that's 5,000 guesses. And if you do the math, it turns out that it doesn't take long to guess a four-digit - we know it doesn't take long to guess a four-digit number. You never need to guess the - it's not ever really eight, as we know, it's only really seven because the eighth one is a given based on all the checksum of the prior seven.

So you first guess - the way this happens is, as soon as that first exchange of four digits fails, the bad guy knows they didn't get the first four. So the brute-forcer can independently guess the first four. Once they succeed, they then continue to use those first four and guess the next three to get the total of seven in order to make up the eight-digit PIN. So all that would have had to happen would have been that the authenticating side held its denial, that is, didn't fail the first four, but flagged that, well, that was not right, but I'm not telling him that right now, so that the side trying to authenticate wouldn't get information that the first four were wrong separately until all eight had been given. Then the authenticating side could say, uh, that's not right.

But instead, apparently, every router on the planet fails when given the first four. So the good news is, all we have to do, all the manufacturers have to do is fix their negotiation not to fail the first four, no matter what they are, if it's an eight-digit PIN that is being expected. Do not fail the first four.

Leo: Don't give the hacker a checkpoint, says Stillatwork.

Steve: Yes, exactly.

Leo: Like a game. You know you got that far. You're safe. Now keep going.

Steve: Yup. Wait until he's given all of them and then fail. So that simple change fixes this problem. Now, the bad news is routers or access points are all over the map. First of all, and I should have said this earlier in case anyone only listened to the beginning part of the podcast, but everybody listen now. None of Cisco's Netlink or Cisco's own access points can have this disabled, Leo. It's in the user interface.

Leo: So you could turn it off, but it doesn't disable it.

Steve: You can turn it off, but it does nothing. So...

Leo: [Laughing] That's another thing they could fix, of course.

Steve: That would be good, yes. I imagine we'll see that fixed.

Leo: So you're saying in the configuration on the web, I can go there and check the box that says "Disable WPS," but it doesn't.

Steve: Correct.

Leo: Why?

Steve: There is - I tweeted a bunch of links for all of our listeners. So because they're long, we can put them in the show notes for this podcast at your domain, Leo...

Leo: Yes, we'll do that [indiscernible], yeah.

Steve: But also Twitter.com/SGgrc will - you will find a tweet of mine with three shortened links in a single posting. There is a spreadsheet which is being maintained on Google showing the results of all the tests that users are doing on their own routers, that it is growing rapidly. It turns out that even Stefan's original disclosure showed all kinds of strange behavior. For example, the D-Link routers, and in this case his was a DIR-655, which apparently is one of the more popular ones, it doesn't lock down. That is, it just - they just didn't implement the lockdown. So there is none of that "wait a minute if you guess wrong." You can guess as fast as you can.

Leo: Oh, that's bad.

Steve: It's really bad. There's also a brand called TP-Link, TP-Link routers.

Leo: I don't think I'd buy a toilet paper router. That doesn't seem like a good idea on the face of it, yeah.

Steve: Yeah. Could we come up with a different name.

Leo: Mm, TP, uh-huh.

Steve: And it also lacks lockdown. Now, Linksys, the WRT-320 has some strange behavior. He notes that somewhere between two and 150 failed authentication attempts, this particular router just died. And it never came back.

Leo: It killed it.

Steve: It just - it killed it. It did a DoS on the router. So you can just shut down the network, take the router offline...

Leo: That's almost even worse.

Steve: I know. It's unbelievable. So, okay. So the good news is the DDWRT alternative

firmware does not support WPS at all.

Leo: I'm told Tomato doesn't either, which is another alternative fail approach.

Steve: True, the Tomato router also does not.

Leo: I love both of those. In fact, if you've got a router that you can flash with those, it's a good idea to do that anyway.

Steve: Yes. And so you can...

Leo: But not all routers are compatible. That's the only problem.

Steve: Correct. So you want to verify that. So everybody but Linksys can and absolutely should disable this WPS configuration immediately. You probably did this last week when we first notified everyone of the vulnerability. So that needs to be done. But anyone with a Linksys and Cisco router, at the moment there's nothing you can do, unfortunately; except, if it's reflashable with one of these alternative firmwares, either DDWRT or Tomato, then you can do that. I have to imagine that Cisco is scurrying to fix this problem. I mean, it's...

Leo: Cisco owns Linksys, so really they're the same thing.

Steve: Correct. Right, right. So right now every Linksys router that users have tried is vulnerable, can be cracked, and cannot have its WPS functionality disabled, amazingly enough. And that's where we stand with this, Leo.

Leo: Wow, wow. Really.

Steve: It's a - yeah.

Leo: A comedy of errors, really.

Steve: Yeah. And it's just - it's crazy that, I mean, the only reason I can excuse or explain this weird four-digit shortcut is that the protocol was extended. Rather than, I mean, it's not like they're trying to economize on bits or something. I mean, they've got 128-bit pseudorandom numbers flying back and forth, and public and private keys and all that. I don't know why they didn't change the definition of the packet. It's the M4 packet. M1 is the enroller sending its public key and the pseudorandom number. M2 is the other side coming back. M3 is providing some other information, configuration and so forth. And M4 is the packet that first contains these four digits.

Why they didn't change the definition so that that packet could be either carrying four

digits or eight, I don't know. But that's not the way they did it. They simply added an additional two packets or two exchanges to allow the first four digits and then the second four digits.

Now, the good news is, again, I want to make sure, if any router manufacturers or firmware coders listen to this, all you have to do is not fail that first four-digit mismatch. Set a flag, respond as if the bad guy got it right even when it's wrong, and wait till you get the next four digits, then fail the whole transaction. That jumps us back up to three-point-something years, on average, for a brute-force attack to happen. And I don't know if you could - you want to make sure that your lockdown works, by the way. D-Link doesn't lock down. TP, the toilet paper link people, don't lock down. Linksys, as we know, just does a DoS. It just dies in some cases.

Leo: Lockdown is how many times you can try before it stops or slows you down.

Steve: But then again, remember, it comes back online after 60 seconds. So it would be nice, gee, maybe give them five times to fail, but then stay off for, like, five minutes or something. That would really slow the process down for brute-forcing.

Leo: The thing people are wondering in the chatroom, does my router do it? If your router doesn't have a WPS button, it doesn't do it. Right?

Steve: No, no, no. No, it does not have to have a button. The button is optional. The only way to know - and, see, that's a very good point. And one of the reasons I'm sure we're going to be following this in the future is already there are several attack tools.

Leo: Like Reaver, yeah.

Steve: As I mentioned, there's a vulnerability scanner. There's Reaver. There's WPS Crack. There's what's-his-name from Serenity...

Leo: The Fire - what?

Steve: The pilot from Serenity. I knew it a second ago, and I'm just blanking because I'm all...

Leo: The actor's name? There's something named after him? Oh, Wash.

Steve: Wash. Walsh. Walsh. So Walsh is the - is it Wash or Walsh?

Leo: It's Wash.

Steve: Okay.

Leo: His name is Wash. I don't know what Walsh is. Walsh might be the tool. I don't know.

Steve: Maybe there's typos at different ends. But anyway, so there is a scanner. I'm sure what we will shortly see, I mean, the hackers are going to have so much fun with this. There will be something for the Mac and something for Windows.

Leo: Right, because it sounds like it's easy to write, to be honest.

Steve: It's easy to write. We already have - these tools are open source. They're on Google Code. I'm sure people are going to do some easy, like, am I vulnerable, I mean, I would be doing it if I didn't know that everybody else was going to be doing it, too, and if I didn't figure that this is going to be fixed in the short-term and not be a problem before long. So it doesn't make sense for me to go off on a tangent right now when I've got so many other things I need to do. But the only way to know, unfortunately - oh, there is one way. And that is, you can see whether your computer will autoconfig. That is, if you've got Windows 7, and I'm sure the Mac must support this...

Leo: Or your phone.

Steve: Or your, well, if you've got an Android phone, we know that yours does, Leo.

Leo: Yeah, mine does that. It's built into this Sprint Epic Touch 4G.

Steve: And that is the way people have been easily testing to see if they're vulnerable is to...

Leo: So even if you don't have a button, if you can autoconfig on the other end, then obviously it's working.

Steve: Yeah. And what I - the good news is, me being the Luddite that I am, all my routers are old. So none of mine have...

Leo: Yeah, this is a new security feature.

Steve: Exactly. So I looked on my Netgear router, and there is no sign of a PIN. But that would be your first clue is to..

Leo: A sticker.

Steve: Exactly. There would be a sticker on the back, probably along with its serial

number and its version number and its MAC address and those things that are per router. And it would say "WPS PIN," and then it would be eight characters. So if you find a sticker anywhere on your router, that will tell you that it does support WPS. And then what you want to do, assuming that it's not a Linksys or a Cisco, is you want to log into it through the LAN side web administration and just turn off, disable WPS. Then, having done that, you want to try to use WPS, use the simple configuration that's available in your operating system to see if it will pair with the router, asking you for the router's eight-digit PIN, in which case you would know if it worked that it was still vulnerable because, unfortunately, WPS is vulnerable in all instances. Or, if it did not work, you would confirm that you had succeeded in disabling it. So, wow.

Leo: And there is this WPS spreadsheet on Google Docs. Steve's got a link in his Twitter feed to it. So you can check here. Apple routers are not vulnerable.

Steve: Yay.

Leo: Yeah.

Steve: Wait. Not vulnerable. What does that mean?

Leo: Well, they don't support WPS.

Steve: Oh, no kidding. Yay.

Leo: As far as I know. They've never implemented that on an Apple router.

Steve: Okay. Oh, in that case the Mac probably does not allow for simple configuration, either.

Leo: That's right, exactly, yeah.

Steve: So you have to be Windows 7. I don't think that Vista did. I think it got added in Windows 7, although I'm not sure of that. And what I'm hoping is that quickly, like the next week or the week after, I'll be able to start pointing people to tools. And so by the - slow down, Steve. Take a deep breath. If anyone encounters WPS-related tools, please, if you're a Twitter user, send me an @SGgrc mention. I'll be keeping an eye on my feed so that I can aggregate that information and share it back out with our listeners. And I'll probably tweet it, also, to everyone so that, as tools for checking vulnerability are created or finding access points that are vulnerable, we'll let people know.

Leo: And I'll just run through, according to this spreadsheet, the list of routers that definitely are vulnerable.

Steve: Good.

Leo: The Alice/Hansanet, ASUS, Belkin, Cisco, D-Link, Huawei, Linksys, Netgear, T-Online, Thomson, and TP-Link - routers by those manufacturers, at least some are vulnerable. So you should pay attention and act appropriately.

Steve: And so what it really means, clearly, is that a neighbor who was wanting to play could set things up. Also in the tweet is a link to Lifehacker. Lifehacker has gone to the trouble of creating an easy-to-use, step-by-step.

Leo: To do it. Not to fix it, to do it.

Steve: To do it, yes. You download a - right now the tool runs only on Linux. But you can download a Linux DVD that already, the latest version of this Linux DVD already has Reaver that has been added to it. And so you're able just to boot your machine from this Linux DVD and start hacking, if you've got a compatible WiFi adapter. Most apparently are compatible and will do this. So...

Leo: Crazy. Crazy.

Steve: Right now it's a little on the tech-y side. We know how these things are going to go. There will be a Mac tool. There will be a Windows tool. There will be plenty of hacker tools and testing tools and so forth before long because this is, I mean, the problem is, most users, obviously, in the world are not listening to this podcast.

Leo: Right.

Steve: Everybody, you know, how long did it take us to get people to put, like, good passwords on WPA or even to turn on security in their routers? I remember that when I used to look at the list of routers that were available, none of them were secured. Now they've all got padlocks on them. But the problem is they've all probably got WPS enabled also. Which means that right now they're all vulnerable. And this is just going to be too tasty an opportunity for hackers to play with. So unfortunately what this means is that this is enabled by default in order to get certified. So all routers have it enabled by default, even those that can have it manually turned off. Which means today they're all on, unless they've been turned off in the last week or two. Yeah, it's not good.

Leo: Now, again, just to reiterate, the bad guy has to be sitting on your curb, close to your WiFi access spot. It's not a remote hack. But it means...

Steve: Like a neighbor.

Leo: But it means you're - that's, I think, that's why it's kind of a shame that

Lifehacker published this article. It's not that this information's not out there. But it means that any idiot like your neighbor can use this technique to get access to your...

Steve: Oh, Leo, it's going to be worse in a week. I mean, they did it because why not? I mean, there will be single push-button, get on someone's network tools here in a couple weeks. This is just too tasty.

Leo: It's too easy, yeah. Well, that's good for people who miss the good old days of Linksys as your Internet Service Provider. Those days are here again.

Steve: Yeah. So I imagine what we'll see is, we will see every router manufacturer updating their firmware, probably to implement the simple fix that I propose, which is do not fail the first four-digit exchange.

Leo: Seems like that would be an easy thing to fix.

Steve: Yes. It's a trivial improvement to the protocol. The protocol stays the same. Nothing has to change. Just don't fail the first four-digit guess. Wait until all eight digits have been provided, then fail it. And maybe increase the lockout time since we know this is a problem now. If that gets fixed, then we're in much better shape than we are now.

Leo: Steve, as always, a pleasure. You fill us in on all this important stuff; teach us, too, which is great. Next week a Q&A episode. You can go to Steve's website, GRC.com/feedback, to feed him a question for next week. We'll do 10 or so Q&As with Steve's answers. You can also go there to find the transcripts for this show. Steve makes those available, as well as 16Kb versions of the audio, at GRC.com. And of course SpinRite's there, the world's best hard drive maintenance and recovery utility. Lots of other good, free tools, as well. GRC.com. Follow Steve on Twitter. In fact, this is a good time to do that, if you haven't already, because he's got links to all the background stuff there on his Twitter feed: @SGgrc. @SGgrc.

Steve: And I will ask, again, anybody who's out there who is a Twitter user who runs across something I haven't seen or talked about, by all means, shoot me either in GRC.com/feedback, put something in the subject line about WPS so that I'm sure to see it, or mention SGgrc in your own Twitter posting so that I'll encounter it and see it. And I'm sure for the next few weeks we're going to be tracking this story because this is a biggie.

Leo: Let's see. What else? Oh, apparently, well, wonder if there is WPS support on AirPort networks because I see "How to connect a wireless-capable printer to your AirPort network." First the printer must be configured to join the network. Add the printer. If your printer - sounds like you - well, it's not clear. They don't, in this spreadsheet, mention any Apple products. So I don't know if this is WPS or not. But you know what, it would behoove everyone to check.

Steve: Yeah.

Leo: SGgrc on Twitter, GRC.com on the web. We'll be back next week at our usual time, that's Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 or 1900, I guess, UTC on TWiT.tv. You can watch live, but we make audio and video available always after the fact at TWiT.tv. In fact, best thing to do so you don't miss an episode is subscribe in your favorite podcatcher, whether that's iTunes; the Zune Marketplace; I use Downcast on iOS, that's a great little podcast catcher; Listen on Android. There are lots of ways to do it. But do subscribe. That way you get every episode, and you can collect them.

88 - what am I saying? I'm going to give you the phone number for the radio show for some reason. I don't think I need to do that. I need to merely say thank you, Steve. TWiG is coming up next. I'm a little - I'm not used to being up this early. You know what threw me? You know what threw me? I just - I'm never here at this hour. "The Price Is Right" is on TV, and Drew Carey seems to have lost a hundred pounds. That's what threw me. I'm looking at that going, what the hell happened there?

Steve: Well, good for him.

Leo: Good for him. Thanks, Steve. And we will see you next week on Security Now!.

Steve: Thanks, Leo.

Leo: I looked over, just went, whoa.