WPS: A Troubled Protocol

Description: This week, after catching up on an interesting week of Security and Privacy news and legislation, Steve and Leo examine the troubled Wi-Fi Protected Security (WPS) protocol in detail to understand its exact operation, and to examine a series of limitations that cannot be resolved.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-337.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-337-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 337, recorded January 25, 2012 - WPS: A Troubled Protocol.

It's time for Security Now!, the show that covers protecting you online. Here he is, ladies and gentlemen, our Explainer-in-Chief, the man in charge of Security Now!, the man who explains how this stuff works and what to do to keep yourself safe, Mr. Steven Gibson.

Steve Gibson: Hey, Leo. Great to be with you again, as always.

Leo: He's the guy in charge at GRC.com, that's the Gibson Research Corporation, creator of SpinRite. It's kind of nice because you have one thing that you do for a living, which is the software.

Steve: Yup, pays the bills.

Leo: Pays the bills. And that gives you enough free time to not only do this show, which is a lot of work, but to program and work and study. And so you've kind of got a good life, if what you want to do is kind of be in touch with what's going on in technology.

Steve: Yeah, putter around and work with technology and come up with new stuff from time to time.

Leo: He's a putterer.

Steve: I'm a putterer. I putter.

Leo: Putter. I putter. So today what are we puttering around with?

Steve: Okay. As I promised last week, when I learned from a number of experts in the industry that my off-the-cuff idea for solving the WPS problem couldn't work...

Leo: Oh, no.

Steve: ...I thought, okay, why not? So that required me to dig in and really understand exactly how this WPS protocol works. The good news is, it is not tricky. I mean, this is going to be a propeller-head episode. This is one where we're going to need our listeners to focus and concentrate a little bit. But the takeaway will be some cool new understanding that we've never had before of crypto, on at least this podcast, and a really good understanding of what the problems are with WPS. And the bad news is that there is no good solution. They've made some compromises in the implementation of this. Apple got it right because Apple's approach uses a one-time eight-digit PIN, whereas the rest of the industry has it printed on a label. That's the big mistake that they made.

And so this episode we're going to, first of all, as always, catch up on some interesting security news. Lots of interesting legislation has gone down in the last week, and we'll bring ourselves current with that stuff. But then I want to explain exactly how the protocol works and introduce a couple new ideas. I talked about it briefly last week, the notion of how it's possible to prove you know something without telling what it is. Which is a really interesting problem. You want to be able to prove your knowledge, but not divulge...

Leo: Not give it away, yeah.

Steve: ...what that is, yes. And it turns out it's not hard, using the bag of tricks we already have with crypto stuff.

Leo: This would be such a fun assignment for a student, a smart high school student or a college comp sci student. How do you do this? Well, I'll tell you what...

Steve: Everyone's going to know about an hour from now.

Leo: Do a thought experiment right now. Think about it. And in a little while Steve will give you - I love that. It's like cover the page, kids, and think about it.

Steve: Or hit pause.

Leo: Hit pause and think about it. And then Steve will explain a clever way to do this.

Steve: Really, really clever.

Leo: I like it. And it actually will help if you think about it before you hear Steve's solution, so you understand what the issues involved are.

Steve: Oh, yeah. Independently inventing things is the best way...

Leo: I agree.

Steve: ...to figure out how things work. There's something...

Leo: Or at least trying. Even if you can't do it, banging your head against it gives you some idea of the parameters involved, of the challenge involved and so forth.

Steve: But if you don't feel like that, just stay put.

Leo: That's how I do this show. Passive osmosis absorption. Hey, we're going to take a break. We've got some security news, too, and we'll talk about all of that in a bit. All right. Let's get the security news, and then we can talk about WPS, Steve.

Steve: Yeah. Many, many people tweeted me various instances of this troubling judgment that came down from a judge in Peyton, Colorado, who ruled that Fifth Amendment U.S. Constitution protection does not apply to encrypted laptop passwords. Now, of course the Fifth Amendment, famously, is the amendment that allows an individual to not incriminate themselves. So they're saying that the Fifth Amendment doesn't apply in the instance of forcing someone to give up their password, even though doing so would be self-incriminating.

This Judge Robert Blackburn wrote, he said, "I find and conclude that the Fifth Amendment is not implicated by requiring production of the unencrypted contents of the Toshiba Satellite M305 laptop computer." He also said that the All Writs Act, which dates back to 1789 and has been used to require telephone companies to aid in surveillance, could also be invoked in forcing decryption of hard drives. And in doing this he sided with the U.S. Department of Justice, which argued that Americans' Fifth Amendment right to remain silent doesn't apply to their encryption passphrases.

Leo: Oh, I saw this. This is really horrible.

Steve: Yes. Federal prosecutors, who did not immediately respond to requests for comment, did post a brief, and they said, "Public interests will be harmed absent requiring defendants to make available unencrypted contents in circumstances like these."

Leo: You know, public interest is harmed by not being able to execute anybody you want whenever you want. That's demonstrably true. But that's why we have a Constitution.

Steve: And interestingly, Leo, this has been a case that's been flopping around for years.

Leo: Well, and other courts have ruled other ways; right?

Steve: Yes. And we talked about this particular case. This was a woman who is accused of mortgage fraud and, they believe, has a bunch of incriminating records on her laptop. In this case she's not using TrueCrypt, she's using Symantec's PGP encryption, which is the same thing, a really strong whole-drive encryption. And she hasn't said she even remembers the passphrase, so she hasn't said that. But her attorney, who actually has done some work in the past for Phil Zimmermann...

Leo: Right, PGP's creator.

Steve: And so he's up to speed on this stuff.

Leo: She got the right guy.

Steve: Yes, exactly. So the DoJ says, "Failing to compel Ms. Fricosu amounts to a concession to her and potential criminals" - and here we go again, of course, we're marching out child pornography - "(be it in child exploitation, national security, terrorism, financial crimes or drug trafficking cases) that encrypting all inculpatory digital evidence will serve to defeat the efforts of law enforcement officers to obtain such evidence through judicially authorized search warrants, and thus make their prosecution impossible." So basically that's a long-winded way of saying we don't like encryption.

Leo: Yeah, because we can't prosecute bad guys. And you're right, they always bring up child pornography, or they always talk about what this woman's alleged crime was. And that's not germane.

Steve: Right. Now, this is just some random judge. He was, doesn't really matter, he was a Bush appointee, so he's been around for a while. This is not the Supreme Court. The Supreme Court has not confronted this topic, but a bunch of lower courts have. Now, the way the law works - I was married to an attorney for a while so I got to see this in action - is that the way attorneys work is they normally form a theory of a case through analogy. So they'll try to say, well, this is analogous to some other situation which has the outcome that that they want, and try to say these are the same things in different

clothes. So prosecutors tend to view the PGP passphrases as akin to someone possessing a key to a safe.

Leo: In a locker, yeah, or a safe, okay.

Steve: Right, filled with incriminating documents. That person can, in general, be compelled to hand over the key.

Leo: Interesting.

Steve: So there is existing case law where, if someone had a key to something, then they could be forced to give it over. But...

Leo: And so this all comes down to the Fifth Amendment and the right not to incriminate yourself. And so you're saying in the case of a physical safe, the courts have not deemed that protected by the Fifth Amendment.

Steve: Protected, yes. And other examples include the U.S. Supreme Court saying that defendants can be forced to provide fingerprints, blood samples, or voice recordings.

Leo: Potentially incriminating.

Steve: Yeah, and arguably skating on this Fifth Amendment incrimination issue. Now, on the other side, civil libertarians cite other Supreme Court cases that conclude that Americans cannot be forced to give "compelled testimonial communications" and extending the legal shield of the Fifth Amendment to encryption passphrases. Courts already have ruled that such protection extends to the contents of a defendant's mind, so goes the argument.

Leo: Wow.

Steve: So why shouldn't a passphrase similarly be shielded, as well? So the idea is, if it's a physical key, that seems to cut on the other side of this issue, and you could be forced to hand that over. But the argument is, if it's something you know, then the Fifth Amendment and the Supreme Court has said you cannot be compelled, you cannot have compelled testimony of the value of that passphrase. So anyway, this is...

Leo: And this is why people become lawyers, because it's fascinating.

Steve: Yeah, yeah. I mean, it truly is. And what's interesting is these things are not black and white. I mean, there is - the law operates in a big gray zone. And after you've had enough exposure to it, you develop an appreciation for the fact that there are hard problems that don't have an easy and obvious solution.

Leo: It's a living document. I know people want to be constructionist about the Constitution, but I don't think the founders considered encryption.

Steve: Right. No. And in fact, that's one of our problems, of course, is that our beloved Constitution is creaky compared to the challenges that it's being given these days. But there was another thing that we talked about some years ago on the podcast, about the issue of warrantless GPS tracking. And the U.S. Supreme Court has just ruled that GPS tracking of a vehicle requires a warrant.

Leo: That we were happy about. That was - we talked about this case; right?

Steve: Not you and I, but maybe on...

Leo: Oh, another show.

Steve: ...one of the other podcasts, yeah. There's a guy, Antoine Jones. The U.S. Department of Justice had argued that Jones had "no reasonable expectation of privacy." So that was their phraseology for their ability to put a tracking device on his car. He had no reasonable expectation of privacy. But in a unanimous decision - which says, whoa, okay, everybody, when does that happen - in a unanimous decision the U.S. Supreme Court said that U.S. law enforcement agents need to obtain court-approved warrants before tracking suspects using GPS devices.

The decision rejects arguments from the U.S. Department of Justice that a four-week-long warrantless GPS tracking of a suspect's vehicle was within the law. The decision upholds a U.S. Court of Appeals decision that overturned the conviction of this Antoine Jones. Justice Scalia wrote, "We hold that the government's installation of a GPS device on a target's vehicle, and its use of that device to monitor the vehicle's movements, constitutes a 'search.'"

Leo: As I remember - now, maybe it was on This Week in Law. I can't remember where we had this discussion. But as I remember, the question was - because the car was, I think, in his garage. And the question was whether they had the right - there's no question they can go into his...

Steve: To sneak in?

Leo: To sneak in. And whether his car, whether they had the right to access his vehicle.

Steve: Yeah, I think this is broader. Scalia is saying that the use of a device to monitor the vehicle's movements...

Leo: Of any kind.

Steve: ...is a search. So it falls under the warrant requirement for searches.

Leo: Very interesting.

Steve: Which is really good news, yeah.

Leo: And it seems like it's completely the opposite of this other ruling, this PGP ruling.

Steve: I know.

Leo: The problem is that the language - I was just looking at the Fifth Amendment of the Constitution. And the language isn't - it's antiquated. And so it's going to take interpretation.

Steve: Interpretation, yes.

Leo: Let me see if I can find this here.

Steve: Yeah, it's like what did the framers intend, and...

Leo: It says - the essential clause is "No person shall be compelled in any criminal case to be a witness against himself."

Steve: Yes.

Leo: That's it. And so you could interpret that, well, you can't be forced to testify against yourself, and that's of course how it's been interpreted.

Steve: Right, taking the Fifth.

Leo: Taking the Fifth. But is giving somebody a password testifying against yourself? I'm not sure that's the case. Certainly, I think it is more analogous, and I hate to say it, to taking a fingerprint or a DNA sample.

Steve: Well, and ultimately, remember that contempt of court is the charge. That is, this individual, if compelled to divulge the password, I mean, the fact is we now have the

technology that makes it utterly unbreakable. So that's not even a question. So this individual has complete control over whether the contents of the laptop will be released and made available or not. So she's able to say, "I choose no."

Leo: Right.

Steve: And then the court holds her in contempt and does whatever they're going to do with her that way.

Leo: They can keep her in jail indefinitely. That's the thing.

Steve: Yeah, well, that's really a...

Leo: And then of course that's why, and we've talked about this before, TrueCrypt has this plausible deniability feature.

Steve: Right, where you're able to take slack space and set up another, sort of another containment that no force on earth can prove is data. The way TrueCrypt works, it randomizes it and just looks like noise. So you're able to say, here's my password, look, there's nothing bad here. And then there is, however, like a trapdoor, there's a backdoor, there's another whole drive which is there. Now, the problem is everybody knows about that. It's not like it's a secret, so...

Leo: Right. And there would be other ways for law enforce- you know, I once asked the Secret Service about this. We talked to the Secret Service, Patrick Norton and I flew out and spent some time with them. And they said the truth is these cases don't come up much. This was, by the way, 10 years ago. These cases don't come up much because most of the time criminals just give us the password. It's not - in most cases you get a confession of some kind. People tell you.

Steve: Yes. And in researching this, there was another case where this was the same issue, and the individual gave the password, the drive was decrypted, and it provided evidence that led to his being found guilty, and he was incarcerated. So as you said...

Leo: Most of the time people just confess.

Steve: Right. And from my standpoint as a technologist, I won't take an ethical or moral position. I'll just say this is math, this is capability, this is technology. How it's used is not the technology's fault. It's the fault of the people. So encrypting stuff is something that there are lots of valid uses for. I want to be able to have my little 32GB thumb drive on my keychain and have it immune from poking by the valet or the car service station because we know that that's been a problem in the past. So that's an entirely valid reason for me having really strong encryption and having access to it. And unfortunately criminals are going to be able to shield themselves using the same technology. But it's not the technology's fault.

Leo: Right. Any more than it's the phone's fault that people can use the telephone to...

Steve: To upset somebody.

Leo: ...to upset somebody. That's technology. It's neutral.

Steve: Right. So DNS poisoning, it turns out, we haven't talked about for many years, since Dan Kaminsky warned the world that DNS servers were more vulnerable than we thought. But it apparently bit CBS. The hacker group Anonymous is claiming credit for changing the main DNS for CBS.com.

Leo: Wow.

Steve: It was 92.122.127.27, and it got changed. It was completely repointed to 198.99.118.36 and .37. And it only lasted 20 minutes because, again, this is not something hard to fix. The problem, of course, is that if the changed entries had a long caching time, then anybody who retrieved that DNS record during that 20-minute window would have caused that CBS - the bad record to get cached for the duration of, I mean, which could be days or weeks. And so what happened was the site went completely black. It went completely dark for people who, when they went to CBS.com, required querying the DNS system in order to get the IP. That is, if they had gone through an ISP, and the ISP's entry for CBS.com had expired, then the ISP would go and fetch it from the second-level servers and then cache it. So it's a problem. And we don't know any more details. CBS is not saying anything except that, well, it only lasted 20 minutes. They're trying to minimize it. The problems actually had to be longer lived than that. But DNS poisoning still is happening. Not often, but it's still a problem.

Leo: It's really intriguing, yeah.

Steve: Yeah, it is. Well, and it completely took the site. It aimed everybody who wanted to go to CBS and all of its subdomains, I mean, everything got pointed. There wasn't even anything really there. There was a blank page with one file. So whoever did this wasn't getting much benefit from it. So it was probably not planned. They probably saw a tiny window of opportunity to perform some sort of an attack and slipped in and did it, but really weren't set up to present people with some - the danger, of course, is not to take them to a blank page, but to a fake CBS page with, like, all kinds of fake news and things.

Leo: Right, or a page with a login for your Google account or something.

Steve: Ah. Yup, exactly.

Leo: And that 20 minutes would be enough to capture quite a few.

Steve: Oh, goodness, yes. Well, and again, as I said, if the caching record said this entry is valid for a week, then it wouldn't get flushed. They'd still be serving the bad IP until it finally did expire, which is why DNS takes a while, why DNS changes take a while to propagate across the Internet because the power of DNS is that it is caching. However, it's worth noting that this could not happen with DNSSEC. So this kind of spoof, there's no way that you could get a record that was fraudulent once DNSSEC is signing records. And so this is a variant, of course, on what SOPA and PIPA were trying to do to us in that legislation, where sites would get redirected. We talk about that a little bit here in a second, so we'll be coming back to that.

I did want to mention the DreamHost, a very big hosting provider that's been around forever, they're based here in Southern California, they detected some sort of unauthorized activity on one of their databases is the only thing they said. But as a consequence of that they did the right thing. They shut down access. They obsoleted everyone's FTP and shell passwords as a precautionary measure and then sent out notices to all DreamHost customers that were affected, saying we're really sorry for the inconvenience. Something is going on that looks suspicious, so we're going to be better safe than not, and you're going to have to come in and update your passwords. So I was pleased to see them handling that nicely.

Leo: Yeah, that's neat, yeah.

Steve: Now, many people, and I'm sure you heard about this, noted that Megaupload got taken down.

Leo: Yes. Oh, yeah.

Steve: Yeah. And it's interesting because the first thing I saw about it was someone I thought made a very good point, which is, well, that happened without SOPA and PIPA. So the system works now, if we just use the laws we have rather than advancing laws that are somewhat questionable.

Leo: Oh, but it's so hard. You have to go through due process. You have to enlist the support of other law enforcement agencies all over the world. It'd just be so much easier just to flip a switch, Steve.

Steve: Yeah, wouldn't that be. Just push a button.

Leo: Yeah. Why make it so hard?

Steve: Yup. The SANS Institute, in reporting this, said that U.S. federal law enforcement agents have shuttered the Megaupload.com website, seized 18 domains connected to the site, and indicted seven executives and two companies. The company is based in Hong

Kong, which means, even in Hong Kong, probably nearby the Post Office...

Leo: Yeah, I was just going to say...

Steve: It's possible to extend our reach and make that happen. The executives face a number of charges, including criminal copyright infringement and conspiracy to commit money laundering. The government says that Megaupload allowed users to access movies before they were released in theaters, as well as music, television shows, eBooks, and software, most in violation of copyright law.

Leo: Oh, where can I get that? That sounds great.

Steve: Yeah, I like that pre-release part. Megaupload has reportedly earned more than $175 million U.S.

Leo: See, that's the issue is did they know and did they intentionally profit from this.

Steve: Yeah. The FBI...

Leo: And looking, I mean, I don't want to prosecute these guys before they get their day in court. But if you kind of look at all the evidence, they picked one that was a - this is a good prosecution in the sense that they have a case, I think.

Steve: Yes. SANS goes on saying the FBI also seized company assets. While some have used the closure of Megaupload to point to the need for stronger antipiracy laws like SOPA and PIPA, others have pointed out that, if the U.S. government can shut down such a large operation, perhaps such laws are not necessary, as they already have the power they need.

Leo: Were their servers in Hong Kong? Were they offshore? Because that was - the contention of SOPA is, yeah, oh, no, no problem, we can handle U.S. piracy sites. But what about offsite, offshore Internet?

Steve: Oh, yeah. They had to reach outside of the U.S. jurisdiction.

Leo: Yeah, they went to New Zealand to get these guys.

Steve: Yup. And in fact one of the guys legally changed his name to something...

Leo: Yeah, Kim Dotcom. His real name is Kim Schmitz, I think. Yeah. Look, I'm not going to go through all the reasons why they seem like they might be guilty. And

you can also - there's all sorts of conspiracy theories. This company had also announced a new music label based on their upload service that was about to launch. So apparently their servers were in the U.S. They were in Virginia. So that's why this worked, and that's the whole contention of PIPA and SOPA, is yeah, well, if you're in Virginia, we've got you. But if you're outside, if your servers are in Hong Kong, then we've got a problem. Or let's say they're in China, where we...

Steve: Because it's so much more difficult.

Leo: There's no copyright law in China at all. So that's the whole idea. And so the idea of SOPA and PIPA was to...

Steve: Is there no copyright law in China?

Leo: It's not enforced. You can go - it's so weird. I mean, I don't know. There may not be. I don't know if it's on the books, so who knows what the law is. You can go, in Beijing, quite openly into a department store, go down to the basement, and every movie ever made is available on DVD, high-quality copies. I mean, it's...

Steve: For a dollar.

Leo: For a dollar. So it's wide open there. I can presume that, if these guys had been running their servers out of China, that we wouldn't have gotten much cooperation from China. So that's the point is what - and of course that's why it's a broken system, because it doesn't take the servers down, it just kind of disconnects them from U.S. users by modifying DNS and search. And that's the problem.

Steve: Right, right. Now, what was interesting, too, was a U.K.-based technology news site, V3.co.uk, reported that, quoting from them, "Panic is spreading among filesharing websites with a U.S. user base."

Leo: Oh, yeah. A lot of them went offline or changed their policies.

Steve: Yes. They said "...FileSonic, FileServe, and Uploaded.to all restricting their services following the shutdown of Megaupload last week. "'All sharing functionality on FileSonic is now disabled,' says a message on the FileSonic homepage. 'Our service can only be used to upload and retrieve files that you have uploaded personally.' Uploaded.to and FileServe carry no messages on their home pages, but FileServe is said to be taking similar action to FileSonic, while Uploaded.to reportedly suspended service to U.S. customers." So that's really interesting, too. They looked at what happened and said, oh, wow, we'd better clean up our act because we don't want to have ourselves rounded up and carted off.

Leo: Right.

Steve: Now, the EFF has a nice little piece that I wanted to share with our listeners. It's pretty short. It's the EFF's, of course, our Electronic Freedom Foundation.

Leo: Frontier Foundation. Electronic Frontier...

Steve: Yeah, I knew that wasn't right, Frontier, Electronic Frontier Foundation. They said, "Post-SOPA and PIPA, What's Next? No Legislation, More Innovation." They said, "Last week's historic protests made clear just what the tech community and Internet users are capable of accomplishing when they act together - not only have the Protect IP Act (PIPA) and its House counterpart, the Stop Online Piracy Act (SOPA), been tabled for now, but in a welcome change, the public debate has increasingly considered the interests of Internet users and the opinions of those who actually understand how the technology works." Imagine that, Leo. "Despite this, we keep hearing people ask: what's next? And where do we go from here? Our answer: We don't need legislation. And let's keep moving innovation forward.

"The answer to maintaining an open, thriving Internet does not lay in legislation, but rather in fostering innovative (and oftentimes disruptive) business models that allow content creators to get paid and consumers to have easy and efficient access to content. We've seen time and again that consumers are willing to pay at a price point that makes sense for them - this is Economics 101. When new business models emerge, artists and fans win. It's only the traditional distributers and gatekeepers (we're looking at you, MPAA and RIAA) who lose, so it's no wonder that those parties desperately tried to ram through dangerous legislation to stop disruptive new business models, with no regard for the attendant serious potential collateral damage. Remember, these are the lobbies that have a history of attacking nascent technologies as far back as the player piano.

"A modern day case in point: last week's public takedown of Megaupload. We've only heard one side of the story so far, so let's set aside the many outstanding legal questions. But it's clear that many artists were using the site to connect with their fans."

Leo: That's right.

Steve: "Given the legacy media companies' reluctance to innovate internally, it's especially unfortunate that the dramatic takedown of Megaupload could chill future innovators who would otherwise experiment with new business models.

"To be sure, there are plenty of exciting new content services emerging. For example, take the Humble Indie Bundle, video game developers who have realized substantial success devising a pay-what-you-want scheme for distributing video games. Or artists like Jonathan Coulton, who has, with much success, produced and distributed his own music (and who recently provided this pithy advice to creators who reject new business models but complain about piracy: 'Make good stuff, then make it easy for people to buy it. There's your anti-piracy plan.')."

Leo: I agree. I agree. But, you know me, I'm very anti-SOPA and PIPA. We went black-and-white last week and everything. I do have to point out that, if you're a rights holder, you have the right to protect your rights Whether it's good business is another question. But it is not - it is completely your right to protect your rights, even though - because the point EFF's making is, well, don't protect your rights, look for a better business model. But as it stands they have a right to protect themselves. So it is not completely easy.

And remember, I am on the "kill Hollywood" side. I mean, that's what we're doing here. That's what TWiT is all about. Y Combinator - somebody in the chatroom just passed this along to me - is what they call a start-up school. It's one of the best known for training entrepreneurs. They give them seed money. They teach them how to do a startup. They just issued a request for submissions for businesses that kill Hollywood. We want to fund startups that will compete with movies and TV, not to protect the world from more SOPAs, but because SOPA brought it to our attention that Hollywood's dying. They must be dying if they're resorting to such tactics. If movies and TV were growing rapidly, their growth would take up all their attention. So they're saying, let's come up with some business models that speed the process. I'm all for that. That's what we're doing here. That's what this is all about.

Steve: Yup. And it works, Leo.

Leo: But at the same time I'm not going to gainsay the right of content creators to protect themselves. They do have that right, and they ought to have that right. Let's just not let - let's not have the pendulum swing so far in the direction that it takes away the rights of the rest of us to have a free and open Internet and a right to compete.

Steve: And we're back to gray area, unfortunately. I mean, many problems do not have good solutions.

Leo: I agree.

Steve: And so it is necessary to strike some compromise, to say - well, I mean, I think we're there. It's interesting that valid use of Megaupload has unfortunately been damaged by the wholesale takedown of the entire site. It does sound like this was a little blunt, and...

Leo: There are a lot of people who have legitimate files on Megaupload, business files and more, who are out of luck. That stuff's gone. So that's a very good point. We've got to find a way that is not all one-sided.

Steve: And we will. I mean...

Leo: I think we are.

Steve: This is all new.

Leo: Exactly. And anytime you have a disruption like this, you're going to have unhappiness and trouble. Look at the Industrial Revolution. If you were a weaver in 16th, 17-century Britain, bad times ahead.

Steve: Yeah, sorry about that.

Leo: Sorry. What can I do?

Steve: So I don't have a testimonial to share because I wanted to tell our listeners that I saw a clear effect on SpinRite sales from what I did last week.

Leo: Oh, really.

Steve: Remember that last week, when I was going through the mailbag for our Q&A, I ran across a neat letter from a relatively new listener who said, you know, Steve, love the podcast, you and Leo are great, et cetera, et cetera. And "I listen to the SpinRite testimonials, but I can't figure out what it does. And I went over to your website..."

Leo: He apparently was not alone.

Steve: "...and I watched the video" - exactly, that's my point. He said, "I watched the video, and it's very entertaining, but I still don't really know what it does." So I launched into a sort of extemporaneous "Here's how SpinRite works." And that's now on the record in Episode 336. So just in case anybody who feels similarly, who doesn't really have an appreciation for the nature of the way defects can grow on a disk, how sectors can go bad that were once good, how and why preventative maintenance actually works, and how it's possible to recover data from a sector that is unreadable, I explained all that last week. So I just wanted to give another pointer back to last week, Episode 336's explanation of what SpinRite does. And I'm going to come up with some way of sticking that up on the website statically.

Leo: Yeah, you could just extract it, yeah.

Steve: Yeah, because it's just...

Leo: One way to do it, just as a thought, we put this on YouTube, youtube.com/twit. And there is a way to do a YouTube URL that jumps to a particular time marker.

Steve: Ah, yes, an offset.

Leo: Yeah. So you could use the YouTube embed code. I know you don't like Flash. Actually there's HTML YouTube, as well. So you go to look at that.

Steve: Yeah, YouTube no longer needs Flash.

Leo: That's right. And so you could have a link. I don't know, though, if the non-Flash version will jump. Anyway, it's worth investigating that you could just - this is what we're going to do on the Tech Guy site. The new Tech Guy site will have - because every show of the radio show is a question-and-answer, question-and-answer. So we'll just have it jump right to that question, and you'll be able to watch the question and the answer.

Steve: Oh, very cool.

Leo: Yeah, I'm excited about it. Anyway, thank you for that explanation. So now we all know the magic that is SpinRite.

Steve: Well, and it had an effect. It was very clear that this guy wasn't alone in thinking, okay, what does it do? And so I'm really glad he asked the question. I'm glad I responded. I'm glad I saw it. And so thank you, everybody, for supporting me by buying SpinRite. I really appreciate it.

Leo: Yes, thank you, yeah.

Steve: So WPS, WiFi Protected Security. We've been talking about it for several weeks. It's not going to go away, unfortunately. We know that we have an industry full of WiFi routers that can be configured not only using a secure passphrase, using WPA and WPA2 with a really long, unguessable passphrase. But in an attempt to make it easier for people to pair a device with a WiFi access point, whether a router or a standalone gateway or whatever, the Wi-Fi Alliance decided that they wanted to come up with something a little more like Bluetooth. We've talked about Bluetooth pairing in the past where you just sort of tell each end that you want to pair, and then they find each other.

Well, the problem is that routers are incredibly inexpensive. No manufacturer wants to spend any money they don't have to. I mean, even some of them that have a button, that's considered an expense because you take the cost of the button, even though it's mere pennies, and you need to multiply it by markup and stocking and shipping and all that, and it adds up. So they typically don't have displays. They have a PIN printed on the outside, an eight-digit PIN.

Now, as we know, it came to light a few weeks ago at a security conference, some very smart person realized that there was, for some reason, four digits were being verified at a time, and that there aren't many combination of four digits, 0000 through 9999. So that's - that sounded a little bit like Herman Cain's tax plan. But...

Leo: [Laughing].

Steve: 999. So that's 10,000 poss...

Leo: That's going to be a trivia question. I don't think that's going to be widely known in a few months, but that's a good one. I like it. Look it up, kids.

Steve: So what was discovered was that this eight-digit PIN was being sent half at a time. And so the result was a massive drop in effective security, from an eight-digit PIN - and, arguably, even that's not very secure. I mean, none of us would use an eight-character password. We just wouldn't. But there's a static eight-character, eight-digit PIN on the router. So what was discovered was, as a consequence of this protocol which is being used, it's possible to try four digits separately from the entire eight. So my immediate reaction was, okay, well, why doesn't the access point just fake that, yes, you got the first four, go for the second four. And instead of denying when the first four were wrong, hold that back until all eight have been guessed, and then the bad guy doesn't know that just the first four digits are incorrect. It doesn't allow him to tackle, essentially, the eight-digit PIN, two pieces at a time.

Okay. So in order to understand why this was done, I plowed into the protocol. And what I learned was interesting, which is the reason we're talking about it, because I think all of our listeners who get a kick out of, as you said, Leo, thought experiments, how would I solve this problem, this is explainable in one of our typical Security Now!, everyone's going to understand this when we're done ways that I think people are going to get a big kick out of. So first of all, the reason that the protocol was designed as it was, was it requires mutual authentication. That is to say, if you think about it, it's not just the client that needs to prove that it knows the router's PIN because this is radio, and we don't really know who we're talking to. It could be someone next door or upstairs or downstairs. It could be somebody within WiFi range that we're connected to.

So think about it, that if there was a third party, if there was a malicious access point that was trying to get you to connect to it, I mean, it would want to. It would be able to have all of your unencrypted traffic and get up to all kinds of mischief that way. So there's a tremendous incentive for an evil access point to pretend to be the access point you want to connect to and you think you're connecting to, instead you're connecting to it. And because remember that our encrypted traffic is only encrypted in the air, and unless we're also over SSL for endpoint-to-endpoint encryption, from the client out to a remote server, then the access point decrypts it in the clear. So malicious access points are a problem.

The point is that the malicious access point also does not know the actual target access point's PIN. So what we want is we want to require the parties at each end of this connection, this sort of nascent, we're in the process of building some trust between Party A and Party B, or the client and the access point, the server, we want both ends to have to prove that they know this PIN. That way we know that we're connecting to the access point that we intend because it knows the PIN that's printed on its own label. But a bad guy can't produce the PIN. So this isn't a unidirectional, client proves it knows the PIN. This has to be bidirectional. It has to be mutual authentication.

Now, here's the problem. Again, it's radio. So it's in the air. And we have no security. There's no starting security. We have a client that the access point has never

encountered before. They don't know anything about each other. They don't know make, manufacturer, model number, I mean, they know nothing. So we don't have something like we have with SSL, where we've got the whole Certificate Authority system where, for example, the access point has a certificate that was issued by VeriSign or DigiCert, and the access point has a known label which, for example, in the case of SSL is a domain name. And we're connecting to it with that domain name, and we know that nobody else has that domain name, et cetera, et cetera, et cetera. There's a whole infrastructure in place that we've discussed many times which SSL uses for connecting to remote servers and authenticating. That is, it's not just security, it's they're proving who they are, thanks to this whole Certificate Authority system.

We don't have that here with a WiFi client and some random access point. They're starting from ground zero. And a bad guy by definition can eavesdrop, that is, listen to all of our communications; can inject their own traffic; can modify the traffic in a man-in-the-middle sort of way; can intercept, delete, or drop any traffic. We have to be secure against all of that. Which is a really interesting problem. If you think about, like, there's three people here, three actors. There's two who are trying to develop a trusted connection, and a third that we're trying to exclude from that, yet we've given it full attack power. It can do anything it wants to to our communications.

So this is a cool problem. So the question is, how does each end of the good guy connection prove that it knows this one PIN? And we'll just assume it's an eight-digit PIN, an eight-digit PIN for the moment. We'll talk about why it's necessary to chop it in half in a second. But how do they prove they know it without disclosing it to somebody who's listening, who can see every single packet that goes back and forth between these endpoints?

Well, the way this is done is very clever. It uses a technology, a crypto technique that we understand, we've talked about many, many times: a hash. Just a hash. We know that a hash function is a so-called one-way function. You hash something of any size, and it produces a fixed-size signature, essentially, that has a relationship to what you gave it, so that any time you put the same thing in, you get the same thing out. Yet the cryptographic definition is that you cannot go backwards.

And in fact this hashing process is a lossy function, meaning you could pour the whole dictionary into it, and you'd get the same-size token. Obviously, the whole dictionary's information content cannot be represented by a little small little thing. It doesn't have enough bits to represent all the information in the dictionary. So it's a lossy, an information lossy function. But the point is, every time you were to feed the same dictionary into the same hash, you'd get the same token. And if you change one character anywhere in that dictionary, you get a radically different result. And the whole point of this is, it is not feasible, computationally feasible, even with all the computers we have now, to predict or predetermine what that output will be for a given input. So it's a one-way function. It's very cool.

Okay. So how do we use that? This is just so neat. The client takes the PIN that it knows, and it adds a random blob. Now, in cryptography we call that a "nonce," as in it's only going to be used once. So it takes 128-bit random or pseudorandom thing, and it appends it to the PIN and hashes that. So it hashes the PIN plus this nonce. Now, we know what that nonce is called. It's called "salt." It's called "salting the hash." It's a way of essentially creating hash functions which cannot be mapped out in advance. Like if it was just the PIN by itself, with the hash, then it would be possible for someone to precompute all the outputs for all the possible PINs. There aren't that many PINs, after all, it's only eight digits. So that would not be hard to do. And that way a bad guy could see, when the output of the hash was sent over through the air, they could say, oh, look

that up in our table, and they'd instantly know what the eight-character PIN was and then themselves be able to authenticate to this router and get on the individual's network and so forth. But appending this 128-bit randomness blows that completely.

So we have this 128-bit randomness plus the PIN, which is hashed. And the client sends that over to the access point. So it's just this blob of debris, essentially. And notice that it tells the attacker, or anyone listening, nothing. All they get is noise. 128 bits, or actually I should say whatever the hash size is. This protocol uses SHA-256, but it only uses 128 bits of the 256-bit output. So it is 128 bits. So this hash can go through the air; and, because it's a one-way function, nothing can be determined about what went into the hash. So the access point has that.

Okay, the next step is the access point does the same thing. It invents, it comes up with its own nonce, its own 128-bit random blob, concatenates that to the PIN, hashes it, and sends it to the client. So now each end is holding the result of the other end's hash of the PIN that they both, maybe they both share, and random things they just invented. So now what happens is the client sends its nonce, the random number it made up, to the access point. And the access point does the same thing, sends its nonce to the client. And if you think about what's in the air, what went by in the air was a hash of the PIN and randomness that tells an attacker nothing. And then in the air, through the air, is that random thing that was concatenated to the PIN. And that's also just randomness.

So what this allows is, without ever exposing the PIN, now what happens is the access point is in receipt of the hash and the client's nonce. So it can concatenate the PIN, which they have in common, presumably that they share, to the client's randomness, hash that, and it should get the same hash value that the client first sent. Which is to say, if those match, if the access point hashes the same data that the client did, it'll get the same result. That's what hashes do. And the only way it's going to get the same result is if the PIN was the same, both that the client hashed and that the access point hashed, and vice versa. The client, I mean the access point, has sent its hash and, secondly, its random number over to the client. And the client is able to hash those and verify that the result of that, it's basically doing the same thing that the access point did, and the access point sent the result to the client. Only if those matched, the only way those can match, is if the access point knew and had the same PIN as the client.

So it's cool and very elegant. It's this interlock, the sending of the hash to each side is called a "commitment," that is, each end makes this commitment of data, taking the PIN and this random value, and sends it to the other. Then each end sends the random value, which allows each end to recreate the hash and verify that it matches what they expect. Only if the PINs are the same will that work. So it's very cool.

So here's the problem. The people who developed this wanted to have a short PIN. And somebody listening will see the hash go across, and they will see the output from the hash go across, and then the next phase is the random number that is being - the salt, essentially, for the hash goes across. An attacker can now perform an offline attack. That is, they know what the hashing function is because that's part of the spec. So eight digits is just not enough protection. Eight digits is 10^8, obviously, is 100 million combinations. Well, that's 2^26.575. It's the equivalent of 26 bits. We know 26 bits is just not enough strength.

So what any eavesdropper could do is simply listen to this dialogue, capture the information in the air, and then take it home, crank it through a forward process, meaning that they know what the random number was that's added to the PIN, so they go PIN, 000, add the random thing to it, hash it, see if it matches the hash that was first sent. If not, 0001. Add the random hash, see if it matches. Then 0002, 0003, and so on.

Fact is, this is just not secure.

Now, the protocol was cut in half, as we talked about it, in order to attempt to provide some protection against an active attacker. You'll notice that, if we were talking to a bad guy, instead of a bad guy listening - we just described the problem of a bad guy listening. But if we were actually talking to a bad guy, we have sent the bad guy our hash. The bad guy sent us, we assume, his hash of the PIN that he's pretending to know, but we don't know if it's valid or not. Then we send our random nonce. Well, now the attacker, again, has everything they need to brute force this protocol. The attacker can't actually follow through with a fourth phase and give us its nonce because it doesn't have the PIN. But because it captured those two pieces of information, the hash result and the nonce from the client, it can now brute force.

So this troubled the designers, and they said, okay, how can we strengthen this? Unfortunately, the way they did it was let's not give them everything at once. Let's make the - see, the problem is, with just those first two messages from the client, all the information has gone over that is needed to brute force this eight-digit PIN. So instead they chopped it in half. They said we're going to hash only the first four digits and send that over, and make the other side verify that first half before we ever put the second half on the line. So, frankly, it's weak. There is fundamentally - I'm sure you could demonstrate this mathematically. There is no way to do what the Wi-Fi Alliance wants to do and have it be secure.

Leo: Wow.

Steve: They chopped this up into two - yeah, I know. They chopped it up into two halves so that the other guy would have to show the first half of the PIN, prove that it also knew those four digits, before the client would give the second half, which would then provide enough information for brute forcing. The problem is, as we've just seen, if the access point is evil, and it's unable to provide the information, then the protocol shuts down.

Well, I mean, that's good, but it turns out that you're only having to guess four digits for that first phase. So now we're down to 2^13. That is, half of that 2^26, 2^13 bits. So it's easy to do some math and say, if I have a one in 10,000 chance of guessing, and I'm able to guess once every two minutes, for example, which is like what these access points do, is they go dark. You get three guesses, then it's dark for some number of minutes, then it lights up again. So you look at how often you're able to make a guess. And it turns out that even in the worst case, in a number of hours or maybe a day, you're able to get the first four. And then you simply do the second four.

The problem is the PIN is static. When I was researching this, I went back and looked at the original source documents, the papers that were written by the security researchers who went into great detail and depth on this whole concept of simple pairing, the idea of two entities that want to build trust. And every single instance of this discussion, this academic discussion with crypto, specifically says the PIN can only be used once. And think about it. If...

Leo: See, if they just did that, if they'd adhered to the standard, we'd be all right.

Steve: Well, no, see, that's just it. The academics, the cryptographers know you can only use the PIN once. The Wi-Fi Alliance said, well, that would be too expensive.

Leo: Oh. Because we couldn't print it on the side of the router.

Steve: Yes. They said we want to print it, and so it's going to be a static PIN. And so they made a compromise...

Leo: That's too bad.

Steve: ...which is really a problem. So standing back from this now, I mean, I know I just dipped everybody's head in some serious protocol. But the takeaway from this is that anybody who ever listens to WPS pairing can take it home, can take the traffic home and crack it offline. They do not need...

Leo: Well, that's important, too.

Steve: Yes. They do not - it doesn't matter if you shut down WPS, when you make a mistake, for a week. If a successful pairing is observed, that's the problem.

Leo: Oh, wow. That's a bigger hole than I knew. I mean, that is a - yeah.

Steve: Yes. As I said, there was some new information in this podcast, Leo.

Leo: That's not good.

Steve: No. If a successful pairing is observed, then just the way I explained it, even with chopping it in fourths, or into four pieces, in half so that each side gives half at a time, all the information is there that allows a bad guy to take what they observed in the air, go home, and do a forward brute force attack on the hash. And they only have to do this eight digits, and they can come back in a short time with a WPS pairing that will succeed the first time. So blocking this, if it fails on the first four digits, isn't secure.

The takeaway from this is WPS can never be secure because they did not use a dynamic PIN. Apple does. Apple, the way you use simple pairing with an AirPort is you turn on their AirPort agent. It generates a PIN, which you then authenticate using. And it's thank you very much, you're now paired. Works beautifully. That's not a problem because an attacker will never be able to attack with the same PIN. If they captured that traffic, well, they'll know what the PIN was that time, but they won't be able - next time they try to pair, the AirPort will come up with a different PIN, and they're in the weeds again because these nonces, these random tokens, guarantee that every one of these transactions is completely unique, using different data.

But the really troublesome takeaway is a passive eavesdropper, whoever sees a successful WPS pairing, can take that traffic home, brute force it, come back, and pair the first time. No shutout will be effective. So everybody within distance of this podcast needs to sooner or later arrange to shut down WPS. It was a bad idea because they took a shortcut. All the academicians know you have to use a dynamic PIN. They said, well,

these access points aren't going to have displays. You can't bring up, well, actually you could bring up a web...

Leo: Well, that's what Apple does. See, that's what's interesting. So Apple uses its client software. There is no display on a router from Apple.

Steve: Right, right.

Leo: But there's a conversation that goes on.

Steve: Right, right. Over the LAN side; right? Over the wired LAN.

Leo: Or wireless LAN, but you still are setting - I can't remember how it works.

Steve: A PIN which is changing every time, yes.

Leo: It's a new PIN every time. It's a one-time use.

Steve: Yup. And that is robust. I mean, it'd be nice if it were longer. But it's like, okay, fine, the chances are one in a hundred million of guessing it, if it changes every time. And they stay one in a hundred million because it's going to be different. So you can't run through 00000000 all the way to 99999999. It's going to be changing all the time. So we have a troubled protocol with Wireless Protected Security. And the only thing people can do is turn it off.

Leo: And if you have a Linksys, you can't turn it off.

Steve: Currently it's going to - I have not looked recently and updated where vendors are. Vendors tend to move rather slowly. They're scurrying around. I contacted the PR firm two weeks ago, when we first talked about this, for the Wi-Fi Alliance and was given a statement that sounded like it came from a PR firm, that basically said nothing. There was no technological component at all. And it's like, okay, well, that didn't tell me anything. But it did say there are three things that we have identified, and we are in communication with all Wi-Fi Alliance partners and working with them towards a solution for this. And it's like, okay. Good luck with that. I mean, certainly Linksys needs to be able to disable this. Netgear we know you can disable it. Everybody needs to disable it. Just this was a bad idea. And at least our listeners know.

Leo: Similar to WEP. Although easier to crack.

Steve: Well, and, I mean, the developers knew this. This is not a surprise. There are charts showing that 2^-13.36 or something, which is the number of guesses, the probability of guessing a four-digit PIN, they understood that there was a brute forcing -

there was a guessing problem or there was an offline brute force attack, which is why you can't use the same PIN. But if it's printed on the back of the router, you're stuck using the same PIN. That's the router's PIN. In fact, someone tweeted me that somewhere in Europe there was an ISP that was widely deployed, hundreds of thousands of routers. Everyone had the same PIN. They didn't even change them. It's like, oh, god.

Leo: Oh, come on.

Steve: I mean, that's a disaster.

Leo: You don't need...

Steve: Now you've got all of the routers with the same PIN...

Leo: You don't need...

Steve: So a bad guy can pair the first try.

Leo: Eh, no big deal.

Steve: Although I think maybe it was disabled by default. I think WPS was disabled in that case, so it was the same PIN, but it was not on. And of course the majority of WPS is enabled because the Wi-Fi Alliance says, oh, we want it to be easy. Ugh. Wow.

Leo: Wow is right. Steve Gibson, always an eye-opener. This one's a really bad one. And as you said on the radio show, and thank you for being on last weekend, if you have a Linksys, all you can do is keep checking to see if there's a firmware update. And for the rest of us, just turn off WPS. If you can is the point. Now, Linksys will make you think you can because there's a checkbox.

Steve: Which it ignores.

Leo: But it ignores it.

Steve: And the other thing that would be nice would be if it were possible for the user to change the PIN. That's the other problem, is if you ever give the PIN to a friend who comes over, then they have access to your router as long as WPS is enabled. So, I mean, it's going to cause a lot of problems and confusion because the PIN written on the back of the router would no longer be the PIN that the router honors...

Leo: That's a good point.

Steve: ...if you're able to go in and change it.

Leo: They may never fix this.

Steve: But you really should, yeah. Just disabling it is the best thing. Ugh. What a disaster.

Leo: Or put DD-WRT on if you have a router that supports it, or Tomato.

Steve: Yes, or Tomato. Yup.

Leo: It's better anyway, frankly.

Steve: Yup.

Leo: Steve is at GRC.com, that's his website, the Gibson Research Corporation. That's where you'll find SpinRite, the world's finest hard drive and maintenance utility. You'll also find all his freebies. You'll also find the feedback form because next week we're going to do Q&A, if something...

Steve: An hour early, by the way, for you live listeners.

Leo: Oh, 10:00 a.m. Pacific, 1:00 p.m. Eastern at TWiT.tv because of Live With Kelly.

Steve: Yup, you're going to go back East...

Leo: I'm flying to New York right after the show.

Steve: ...to be a star.

Leo: Well, whatever.

Steve: So, yes, GRC.com/feedback. Fill up my mailbag with thoughts and questions, and we'll do them next week.

Leo: Excellent. Excellent. Steve, by the way, if you're there, Steve always makes 16Kb audio versions available for the people who are really bandwidth constrained, maybe on a smartphone or something. And he also has transcripts, which is really

the ultimate in squeezed-down size, all at GRC.com. At TWiT.tv we have the video. We have audio and video. And so either way. And of course you can watch live, next week a little early, 10:00 a.m. Pacific time. Thanks, Steve.

Steve: Thanks, Leo.

Leo: See you next time on Security Now!.