SECURITY NOW!

Transcript of Episode #483

Let's Encrypt

Description: This week Leo and I cover two major stories: the discovery of a frighteningly capable and sophisticated espionage malware known as "Regin," and deeper coverage of the forthcoming "Let's Encrypt" free and automated web server certificate issuing and management system. And, as always, we also cover a bunch of interesting smaller issues.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-483.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-483-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here with really two big topics: A new massive malware threat that looks to be written by some government somewhere in Virginia, it's called "Regin," we'll talk about that and kind of the miraculous construction of it; and then Steve will take a look at the new effort from the Electronic Frontier Foundation to encourage encryption on every website, Let's Encrypt. Security Now! is up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 483, recorded November 25th, 2014: Let's Encrypt.

It's time for Security Now!, the show where we cover your security and privacy online with this cat here, the Explainer in Chief, Mr. Steven "Tiberius" Gibson, who is coming up for New Year's, I hear.

Steve Gibson: I am. And early enough to hang out.

Leo: We're going to have a host dinner.

Steve: Correct, on the 29th.

Leo: Will you be here for that?

Steve: Yes, Jenny and I both.

Leo: I think we've got 20 or more people. And what we've done is we've booked the oldest roadhouse in the area, the Washoe House, which used to be a roadhouse in the 1860s where the stagecoach would stop.

Steve: Huh.

Leo: They haven't really updated it since.

Steve: So we'll be picking splinters out of our butts afterwards?

Leo: Yeah, it's - we thought, you know...

Steve: Don't slide. Don't slide.

Leo: Right, exactly. We thought, oh - Lisa and I went back and forth. We really wanted to have a host dinner. And because a lot of people are coming for New Year's Eve and so forth, and we just thought it'd be really fun to have all the hosts together. And since many of you are coming from out of town, initially we were just going to go to one of our many fine restaurants here, but that could be anywhere. So we thought, what could we get that's really - that says Petaluma?

Steve: And you found it.

Leo: And this says Petaluma. But it's good.

Steve: They have plumbing.

Leo: They have plumbing, but you can hitch your horse right up to the front there. It's actually, it's really a great place. And they have great steaks. I know you like steaks. We're going to get some Cab for you. I don't know what Jenny's going to eat, but we're going to get...

Steve: And what's it called again the place, just The Roadhouse?

Leo: It's the Washoe, W-A-S-H-O-E, Washoe House. And they've had in the past, you know, it's haunted, they say. So every Halloween Eve one of the local radio stations spends the night there, looking for ghosts. It's fun. I think you're going to like it.

Steve: Cool.

Leo: Yeah. And IDoTech says dollars bill on the ceiling. That's right. How did you know that, IDoTech? That's exactly right.

Steve: He does more than tech.

Leo: Apparently he's been to the Washoe House. Or else he's looking at the website. It is really an amazing place.

Steve: So today we've got two topics. We didn't have much news. And frankly, I studied, I read the entire Let's Encrypt spec, front to back, and I've got it. I understand all about it. Then I started doing my homework on one of the things I wanted to talk about, which is the - we think it's called "Regin," and I'll tell you why I think so - new amazing malware which has been defined. And it just sort of absorbed my entire morning. The deeper I got into it, it's like, oh, my goodness.

Leo: The folks at Symantec who discovered it said it was the most sophisticated malware they'd ever seen.

Steve: Well, if any of the people or team who developed it are listening to this podcast, nice going.

Leo: And would that person who is listening to this podcast be anywhere in Northern Virginia?

Steve: Yeah, I think they probably have a large stake of data territory. So we're going to talk about that first because technically it's sort of our show notes. And the main topic of the show is Let's Encrypt. And not much else happened. We've had several crazy busy weeks previously, so that's fine. We're going to do deep dives into these two topics. And there is a little bit of miscellaneous stuff because I did see "Citizenfour." I want to talk to you about that and some other stuff.

Leo: Oh. Oh, good.

Steve: So we've got another great podcast here.

Leo: I'm excited. Let's get down to the tech news, or the security news.

Steve: Yeah. Okay. So the bottom of the first page of the show notes is Symantec's diagram of the way this Regin installs itself, which you might want to put up on the screen just to show people who are watching the video. Just the first few paragraphs of Symantec's analysis gives you a sense for what they feel about this. They said: "In the world of malware threats, only a few rare examples can truly be considered groundbreaking and almost peerless. What we have seen in Regin is just such a class of malware.

"Regin is an extremely complex piece of software that can be customized with a wide range of different capabilities which can be deployed depending on the target. It is built on a framework" - actually, in my notes, after digging into it, it's an OS within an OS. It's that sophisticated. It brings its own encrypted virtual file system along. It's like nothing we've seen before. So anyway, they said: "It is built on a framework that is designed to sustain long-term intelligence-gathering operations by remaining under the radar. It goes to extraordinary lengths to conceal itself and its activities on compromised computers. Its stealth combines many of the most advanced techniques we have ever seen in use.

"The main purpose of Regin is intelligence gathering, and it has been implicated in data collection operations against government organizations, infrastructure operators, businesses, academics, and private individuals. The level of sophistication and complexity of Regin suggests that the development of this threat could have taken well-resourced teams of developers many months or years to develop and maintain.

"Regin is a multi-staged, modular threat, meaning that it has a number of components, each depending upon others, to perform attack operations. This modular approach gives flexibility to the threat operators as they can load custom features tailored to individual targets when required. Some custom payloads" - which we'll discuss in a minute - "are very advanced and exhibit a high degree of expertise in specialist sectors. The modular design also makes analysis of the threat very difficult, as all components must be available in order to fully understand it."

Okay. So here's what we have. We have a fundamentally different class, an entirely different class of malware. I mean, this is what books are written about, like sci-fi espionage books. I mean, it sort of gives me, as a developer, goose bumps because it is clearly the work of somebody, as Symantec wrote, with extensive resources. For example, one of the add-on modules is able to infiltrate and intercept GSM cellular admin traffic, parse it, and exfiltrate it.

Also in their report Symantec shows some pie charts or where this thing has been found. Leo, the PDF, the second link in the second page of the show notes is to a PDF. And I think like about the third page of the PDF, if I remember, has two different pie charts. The first one shows the geographic distribution of where this thing has been located, or I'm sorry, the second one is geographic distribution. More than a quarter of instances have been found in Russia; the second most prevalent location was Saudi Arabia; and then other specific geographical regions, working their way down. So this is not...

Leo: And by the way, not the U.S.. and not the U.K.

Steve: Right. Zero. So...

Leo: Just a coincidence.

Steve: ...this is not opportunistic infection. This is clearly explicitly targeted. I mean, if we think of Stuxnet, Stuxnet is the only sort of similar thing we've discussed where something the size of nation states had focused intent and implemented their focused intent through software and through cyber espionage to achieve that goal. In the case of Stuxnet, though, it wasn't a general-purpose platform. It had characteristics of that; but all of it was about, we now know, arranging to get a specifically capable malware package installed on air-gapped machines that were controlling the refinement centrifuges in Iran.

So this is different in that it is fundamentally structured to be a general-purpose espionage tool. I mean, what's been discovered is what probably the NSA has spent huge resources creating and hiding. One of the ways this thing, I mean, it does many things that have not been seen before. For example, they presume there must be an executable "dropper," as it's typically called. That's the module that is dropped onto the machine. It has to be executable because, without any prior knowledge, the machine just has to run it. But even now, they've never found it. So they don't - they've never found a sample of it because it is designed not to need to persist. The various pieces of this live in both the registry, where they're encrypted binary blobs, encrypted with a custom variant of the RC5 cipher, which again, when you think about it, that's very clever because RC5 is very simple.

And so it's a perfect cipher to choose if you could choose among anything. It uses a 64-bit block, which makes sense also because what they're going to be encrypting and decrypting are virtual sectors of their own encrypted virtual file system, which actually has a very FAT-like structure, a FAT file system internally. And so those sectors will always be multiples of 64 so you don't have block padding problems and so forth. So, and it uses a 64-bit block and 20 iterations of a customized version of RC5. So again, none of this is off the shelf. This was designed for a purpose. So what happens is...

Leo: How did Symantec find out about this? And it's been around, they say, for years, like five or six years.

Steve: Well, yeah. So, and that's another creepy thing. They encountered what they think is Version 2 in 2013. So sometime last year they encountered it. But when they found it, there were some similarities with other patterns they have long been collecting, which allowed them to then go back in time since they've been gathering stuff. And they realized it's been around - and, I mean, I'm getting goose bumps - it's been around, never before detected, as early as 2008.

Leo: Wow.

Steve: And maybe earlier. So the farthest back they found, once they saw and recognized something in what they're calling v2, then pattern analysis was able to roll back in their logs as far as 2008 and say, whoa, we didn't know what this was then, but we now recognize it from similarities, and we'll call that Version 1 just because we have to call it something. And in fact it's that which caused them to bump what they had just found in 2013 to v2. They don't know there weren't intermediate things. But one thing that's interesting is that in 2011 it was removed. It was, in one moment, it disappeared across the entire Internet.

Leo: So it also has a self-destruct capability.

Steve: Yes. And where they found remaining little bits is where something was broken that prevented its deliberate extraction. A machine had been, like, taken off the 'Net and never connected to the 'Net, and they found fragments of it there, where it wasn't able to - it couldn't receive the "remove yourself" instructions. So something happened in 2011 that caused the controllers to suck it back in, essentially, to deliberately extract it...

Leo: Amazing.

Steve: ...from all the little nooks and crannies. Oh, I mean, this, it just is science fiction. This is just wonderful.

Leo: Yeah, wow.

Steve: And then it came back. In 2013 it returned, and those pie charts are where it is now, where Symantec has discovered it. And what's really interesting is the one, the other pie chart, not the geographical distribution, but there is a chart by sort of like application domain. For example, it is very modular, and there's a wide range of modules which have been found. For example, one of them infiltrates hotel reservation records and exfiltrates who is in what hotel when, almost as if to figure out, again, this is - I spent some time with Padre yesterday on This Week in Enterprise Tech, talking about this. And I drew the analogy of how, with cell phones, we've talked about how powerful cell phone metadata is because, even though you may not have the conversations, you may never get access to the conversations, the idea of knitting together who is talking to whom when, if you can build that connectivity graph, that's hugely powerful.

Well, similarly, if you're operating on a nation-state level, you care about what other actors in other countries are moving around the globe, where there's - there's also an airline reservations infiltration package. So where people are flying and where they're staying, especially if you want to then send an ops team in to get the adjoining hotel room and set up listening equipment. I mean, this is what we found, essentially, is like...

Leo: So it sounds like, based on the locations this has been uncovered in, and the industries, that it is a highly targeted attack. It doesn't spread as a virus would normally spread.

Steve: Correct.

Leo: It would have to be infected. In fact, Symantec says they don't even have a copy of the dropper, the thing that starts the process.

Steve: Correct. And in fact they - nor do they have any good sense for what the infection vector is. In one machine, one infected machine's log that they happened to find, they found a log that implied that an unknown vulnerability in Yahoo's Instant Messenger was used as the means for getting this into that particular machine. But that's, like, all they know. There's no, like, readily discoverable means for this thing getting in. So again...

Leo: It seems likely that there's a variety of vectors, though; right? They wouldn't just be Yahoo! Messenger. They would email.

Steve: Correct.

Leo: They'd do spearphishing. There's, you know, USB keys. There's a variety of ways.

Steve: But those examples are all too common, Leo. My guess is that they are using undetected vulnerabilities.

Leo: That would make sense because an email would leave a trace.

Steve: Right.

Leo: A USB key would be detectable and traceable.

Steve: This thing is way too stealthy. Those things that we're normally talking about, that's your common hacker stuff. These guys are operating on an entirely different level where, I mean, you know, we've heard that Microsoft tells the NSA about problems in Windows before they're published. Right? I mean, we know that, that the government receives early notification of vulnerabilities in Windows. And note that this is only Windows. This is entirely Windows-specific. This has nothing to do with Mac or Linux, period. This is a Windows-targeted technology. Okay, so...

Leo: And it makes sense that it would be not merely defensive, for defensive reasons Microsoft would tell the government. There might be an offensive reason they'd like to know this ahead of time.

Steve: They would love...

Leo: By the way, 2008 you'd make it for Windows. Maybe in 2014 you'd cross some other platforms, maybe get it in an Android, where there are far more installations; right? We don't know.

Steve: Right, right. And where Android is inherently a location-based system. You have a mobile device where - and if you can infiltrate somebody's Android phone - it's funny, well, no, I won't go down that rat hole. But I'm aware of an instance where a conference was held where there were Chinese delegations, and Australians had their Android phones infiltrated during that conference, was like, oh, wow. I mean, in Australia, not in China. So they bring their own ops people with them.

Leo: So what's really interesting is that the dropper is the only unencrypted part of this. It then decrypts blobs and adds capabilities.

Steve: What it does, yes, in several stages. So all of the payload material which the dropper probably downloads once it - so the dropper comes in unencrypted. So something gets itself into the machine. It probably fetches pre-encrypted blobs which it tucks into unusual places. The registry is a place where you've got flags and switches and parameters; right? But it is a flexible tree-structured database. Well, I mean, and you can store binary data in the registry. So they take components of this system and stick them in the registry, just store them, not, I mean, like abusing what the registry was meant for. But it's already big, and it's big and complex, and nobody tends to look in there. And besides, this is encrypted. So the dropper tucks things into registry keys.

And in Symantec's document - anyone who's interested in this should take the time to just - you can just google "Regin," R-E-G-I-N. Or it's the second link in my show notes, and I just tweeted the link to the show notes. And of course you can get them from GRC - to take a look. Because there are...

Leo: It's amazing. It's just incredible.

Steve: Yeah. There are details I'm skipping. But, for example, Symantec enumerates the key, the registry keys which, if present, are probably an indication of this infection. And their own software, of course, Symantec's own security scanning stuff is now fully up to date and up to speed, as you would expect on this, and protecting their own customers from...

Leo: But what are they looking for? If they don't have the dropper, they're not looking for the dropper, they're looking for the thing that the dropper puts on the system.

Steve: Correct. And so they know which keys it uses in order to...

Leo: Got it, they're looking for the reg keys, got it. Of course, yeah.

Steve: Right. The other thing it does is NTFS is a much more flexible file system than we normally see. For example, you can use a colon in a filename to create another stream. In the same sense that a stream is the file that is at that name, if you put a :1 after a filename, and you can, it creates an entirely separate stream which you normally never see. Well, there's also something known as "extended attributes" with NTFS. And they can be big. And so this is another place this dropper stores encrypted pieces of Regin, is in NTFS file attributes. And finally, it also stores blobs, encrypted blobs, in the slack at the end of the drive, after the last partition.

Leo: Wow.

Steve: For historical reasons, partitions are always a multiple of a cylinder size. And cylinders, which have now become an abstraction, a cylinder is the number of heads times the number of sectors per head, is the old terminology for a cylinder on a hard drive. Well, heads are now typically 254, and sectors are 63 because that's the maximum number of sectors you can have. So 254 times 63 is the cylinder, sort of the virtual cylinder size that you will have today. And that's a lot of sectors. But no drive cares about being a certain exact multiple of that. The drives are just whatever they happen to be. But the partition has to end on an even multiple of that, which means up from zero to that many sectors may, because that would not be quite a full cylinder, there's that much possible slack at the end of the drive.

SpinRite, for example, knows that, and deliberately goes out and does its job out there. So for what it's worth, SpinRite is helping to recover this, if it ever had a read problem. That is, SpinRite's aware of this slack space and also does its work out there. But nothing is normally used out there because it's extra file system. It's outside the file system. This thing can also put components out there. That's another - it's one of the three places: registry, NTFS attributes, and file system slack at the very, very far extent of the drive. And all of that is encrypted.

So what then happens is the dropper also puts some registry entries, which cause the installation of a kernel driver, which is the Stage 2, if you call dropper - I think they call dropper Stage 0, actually. So that's Stage 1 that gets loaded at boot time. Then it decrypts Stage 2, and all of this content is coming - none of this lives in the file system. None of this is where you would normally scan for it or see it. It's all tucked away, all custom encrypted. And then that Stage 2 loads Stage 3. It decrypts. Then it finally gets going and interesting with Stage 4. Oh, and one of those earlier stages also installs rootkit hooks, which prevents any, as they called it, code attributes - there was a better word that they used. Not vestiges. I have it in my notes somewhere. Anyway...

Leo: Vestigital?

Steve: You know, debris.

Leo: Leftovers.

Steve: There's no way, because this is rootkit technology, there's no way, once Stage 1 has executed, to sense that there's any of this code running in the system.

Leo: Wow. Wow, wow, wow.

Steve: So it does all the low-level rootkit hooks to go to stealth itself. So, finally, it gets into the framework functions. And they call it a framework because it is essentially a series of general-purpose modules that function like operating system functions that are available to be called by other custom modules. So, for example, Regin contains its own compression and decompression, essentially API calls; it has them for encryption and decryption; what they call the EVFS, the Encrypted Virtual File System; its own container manager; log management; loaders; network operations; then command-and-control by TCP, by UDP, and some sort of a C&C oversight processor.

The payloads which this thing brings include the ability to sniff low-level network traffic; to exfiltrate data through a whole range of communication channels, not only TCP and UDP, but also ICMP. So when it looks like it's sending a ping, the ping's payload is a compressed and encrypted payload going out looking like an innocent, hey, I'm an ICMP ping. If you receive this, send back an echo. But it's certainly not that. But neither is it anything you can recognize.

And they even go so far, Leo, as to use HTTP with encrypted cookie values. So you'll see, I mean, so somebody, a network engineer, trained, could be looking at a Wireshark packet capture of a standard web query where he sees in the headers a set cookie header with an innocent-looking name, and it uses common names, and the value looks like gibberish. Well, values already, innocent values are gibberish. You know, they're just nonces. They're often, you know, we're used to sort of just kind of going, okay, yeah, whatever.

But in fact, if your system is infected with this, this is your password keystrokes being exfiltrated after they were compressed and encrypted in the value of an HTTP header cookie, and you don't know it because they all kind of look the same. So this is able to gather information about your computer, running processes, installed applications, all of that stuff. It's got keystroke logging to steal passwords. It's able to crawl through the file system and detect and recover previously deleted files and exfiltrate those, if it wants them.

Leo: Oh, how useful is that? Maybe I'd buy a copy.

Steve: It's also able to...

Leo: You put SpinRite in the thing, you've got something.

Steve: ...capture your screen and to take over your mouse and execute mouse point-and-click activities. And then...

Leo: Execute.

Steve: Yes.

Leo: Not merely monitor, but do it.

Steve: Yes, take over your mouse and move it around and click things that it needs to do, probably when the web cam that's monitoring you sees that your back is turned. So in terms of when they talk about the specific expertise required, the best example is that they found a payload which has probably been inserted into foreign nations' cellular network base stations because they found something that is able to sniff and gather the admin traffic from cellular base station controllers. So, again, to do that you need to have really application-specific knowledge. There's also IIS server web traffic monitoring agent and a parser for Exchange databases. So they get this thing into a machine that has Exchange installed, and then they have their own parser of the database to go through and extract things. So, you know, just amazing.

So we know that, well, we know that it is highly complex, that it is general purpose. It's been around for at least six years, since 2008. It had a period of time there was no activity during 2012. So it was deliberately, not just shut down, not just went to sleep, but it was extracted. It was pulled back across the Internet at some time in 2011 and then reappeared in 2013, altered. It was changed. So Symantec does see changes from the earlier samples that they were then able to detect after they found the newer ones. They knew what to look for, which is how they knew that it existed back then. So it is in v2 now and, I mean, and active. It was found, you know, these charts are from discovering it on the Internet, actively in use, right now.

Leo: And it's credible; right? I mean, Symantec - has anybody confirmed the sighting, or is it all - yeah.

Steve: Yes, actually Kaspersky has it. Samples are now being posted to Pastebin. Our friend Simon Zerafa just this morning was tweeting me links to more and more of these bits and pieces showing up.

Leo: And of course I've seen speculation that it's either the NSA or GCHQ, the British NSA. I mean, it's, I think, telling that neither Britain nor the U.S. have any incidences of infection.

Steve: And Padre had the tidbit that he had found, which I didn't see, but I certainly know that he found it, that it was originally authored in English.

Leo: Ah.

Steve: You can typically tell what the language of authorship is in these things. And, yes, so English was the language that the developer spoke. Yikes.

Leo: Well, I mean, it's not, you know, to be honest, you're not surprised; are you?

Steve: But isn't it cool? I mean, I just think, I mean, it's one thing to say...

Leo: It's straight out of "Homeland." Right? I mean...

Steve: It is. It's one thing to say, yeah, they probably have something.

Leo: You can see Saul pulling the strings.

Steve: We found it.

Leo: Yeah.

Steve: Yeah, actually Saul's not pulling anything. Have you been watching "Homeland"?

Leo: No, don't spoil - you know what? I'm, okay, I'm a little miffed. Not at you. But I'm watching Showtime because I love "Homeland," and I haven't...

Steve: Oh my god, Leo.

Leo: ...started the new season. And, yeah, well, but their promos are showing stuff.

Steve: Oh, you're right.

Leo: And it pisses me off because I didn't know anything. All right. I'm one of them whiney spoiler people now. But...

Steve: Well, no, I...

Leo: They don't have to - their promos have to - they're saying, "You won't believe what's happening." And I guess I'm going to have to start watching from the beginning because clearly everything's going to be revealed before it's over.

Steve: Ooh. It's so good.

Leo: But don't you, I mean, I think at this point you...

Steve: Actually, you have two weeks because they're now in hiatus.

Leo: It's season finale?

Steve: No, no, they're on hiatus. They left it at an amazing cliffhanger on Sunday. So now would be a great time to catch up. You've got the holidays and everything. It's like, oh. But Leo, believe me, it's so good.

Leo: I loved the first two seasons. I think it's a great show.

Steve: Wait. Whoa. So you didn't see the third season?

Leo: Wait a minute. Maybe I did.

Steve: I think we're in 4 now.

Leo: We're in 4, okay, yeah. Yeah, yeah, they dragged out that first plot for three seasons, that's right. Now they have a new story.

Steve: Oh, it's good. It's good.

Leo: Okay. But I'm telling you, if you're following the NSA revelations by Snowden, if you're even - if you're watching shows like "Homeland," this is not - of course the U.S. spy agencies, we've certainly got the people with the brains to write something like this. It's exactly what you'd want to do. If you were going to - if you, Steve Gibson, were going to sit down and write a program, some malware that would be useful in espionage, this is exactly how you would - it'd be hard to track, hard to trace.

Steve: As I was reading through this, I had already planned when I was preparing the show to directly address the creators and say, "Nice going."

Leo: And I don't think these people are...

Steve: I mean, this is what I would create.

Leo: It's not your garden - I mean, obviously it's a state doing this.

Steve: Yes.

Leo: But even the people writing it are not your garden-variety hackers. They're not attackers. They're probably Ph.D. graduate-level computer scientists because they're doing all the - I'm sure it's, I mean, we haven't seen the code, but I'm sure it's nicely coded. It's written well. It's got lots of comments. And I really like the idea of, look, if you're going to make something that's not traceable, you take advantage of exploits. You don't do spearphishing attacks because an exploit can be done silently, without anybody knowing anything. You can get in there. You can leave the - and then the brilliant thing, the dropper...

Steve: Yes, it self-destructs.

Leo: It self-destructs, and these...

Steve: And there's no evidence of how it got in.

Leo: Yeah. I mean, it's exactly what you would do if you were thinking about this. And I'm sure Mark Russinovich probably wrote it in one of his novels. This is exactly what you would do.

Steve: Yeah. And you have something which - the other very telling thing is that it was clearly designed for long-term penetration, that is, this is a huge investment.

Leo: Right.

Steve: So they didn't want it to be discovered. They're not happy right now that Symantec has just, you know, is publishing keys.

Leo: Maybe, maybe not.

Steve: Yeah? Well, they may already be on to, you know, Version 3.

Leo: I'm sure they've moved on. The 2008, my god, we're much better than this.

Steve: Version 3.0.

Leo: That's the old one. That's the Casio Smartwatch. We've gotten much farther than that. I'm sure they have. And, you know, if they've really done their job right, Symantec will never know about what they're using today. But of course the thing that always scares me is that these are powerful weapons that could easily be misused, whether by somebody in government or somebody not in government who just gets a hold of it, and could very easily be misused. And it shows you why really, if you're concerned about privacy, forget about it. I mean, you check into a hotel, they know. You use your phone, they know. You get on an airplane, they know. Right?

Steve: Yeah. Yeah.

Leo: Do you want take a break here? Would this be a good time to do it?

Steve: Good, yeah.

Leo: Yeah. Why not?

Steve: We have a little errata, some miscellaneous stuff, and then we'll get into...

Leo: Crypto.

Steve: ...Let's Encrypt.

Leo: Have good reason to encrypt.

Steve: Amen.

Leo: Although...

Steve: That won't help you.

Leo: Doesn't matter.

Steve: No.

Leo: Once you're owned, you're owned. It doesn't really matter if you encrypt.

Steve: They're happy to have their outgoing ICMP encrypted. Eh, thank you very much.

Leo: They'll encrypt.

Steve: It keeps anybody else from getting it.

Leo: Yeah. But if you think about it, I mean, we're spending all this time using PGP on our email and stuff. If they've got something on your machine, you can do whatever you want. They can read your keystrokes.

Steve: Yeah, well, they have Windows on our machines.

Leo: They're looking at you. Hi. I'm waving back. Hi.

Steve: I was chatting with a reporter from Wired magazine yesterday about TrueCrypt, and I made the comment that, you know, there was some story that went by, and I didn't catch it, that indicated that BitLocker's strength may have been recently reduced. I don't remember what that was. But I made the point that, you know, we just basically accept whatever we're being sent every month from Microsoft, assuming that it's helping us. Well, that's a big assumption.

Leo: Yeah. But again, it doesn't matter because, if you can see it and you can read it on your computer, and they have spearphished you, they can see it and read it, too. You can use SSL and use PGP and use BitLocker, it doesn't matter because, as long as you can read it, they can read it. Oh, well. All these people, you know, because this is - this is what gets me is people are so, oh, you know, I'm so worried, I'm going to be very careful and never use Google. Google? What are you worried about Google for? Why do you care about Google? They don't need to put anything in Google's service. They just put it in your computer. I got an email about a Kickstarter project that you might be interested in.

Steve: Oh.

Leo: We know your love of coffee. Do you like French press coffee?

Steve: I've had it, yeah. It's good, yeah. I mean, basically because it means it's going to be fresh, and it's not sitting around on a burner.

Leo: Yeah. Yeah. This guy, it's called the ESPRO, or Espro Press. He's already raised $83,000 on his Kickstarter. Oh, I guess it's over. I thought he had more time on it. But this looks really good. I'm going to get you one. This will be your holiday gift. The idea is it's like a travel French press.

Steve: Wow.

Leo: Yeah, that you carry with you. And then it has two-stage micro filter and all of this stuff. And then - it makes about four cups of coffee, and then it's in a vacuum insulated container. So it just...

Steve: Very nice, so it keeps it warm.

Leo: Isn't that good? You just - it's like your thermos has a French press built into it.

Steve: Yeah.

Leo: Did you ever get that cup? The thermal cup?

Steve: No. But, boy, they're communicating a lot. Their roof blew off of their work shed, and then they....

Leo: [Whining sounds]

Steve: I know, I know. It's like, I'm still hoping.

Leo: You know you're in trouble when you start getting a lot of messages from a Kickstarter project.

Steve: Yeah, well, and, I mean, I understand. By its nature, these are generally non-professionals.

Leo: They've never made a product, right.

Steve: Yes. And I've made a bunch, and I understand. So it's like, yeah, okay, you know? Yeah, the Temperfect Mug.

Leo: They had almost 5,000 backers. They raised more than a quarter of a million dollars.

Steve: Yeah, we'll hope. I mean, I'm still getting email, so, you know...

Leo: It's been there since January 1st.

Steve: When there's email - yeah.

Leo: You talked me into buying this. I'm holding you...

Steve: Yeah, that was - it was a big, it was a heavy lift to get you to click, yes.

Leo: [Laughing] Steve says, "Temperfect mug," I'll take it.

Steve: Okay. How many phones do you have within the span of your two arms at this moment?

Leo: Oh, one...

Steve: And I didn't talk you into any of those.

Leo: ...two, three, four. But this is an unusual day.

Steve: Oh, uh-huh.

Leo: I actually have five phones with me right now.

Steve: Yeah, okay.

Leo: This one's a Chinese clone of the iPhone...

Steve: Oh, I saw that. You were showing it to someone on a different podcast. I was amazed.

Leo: I have to be careful because there's some Apple engineers in this studio right now. So I have to be careful.

Steve: Uh-oh. Well, maybe they can get some tips.

Leo: I should give it to them to bring back to the mothership.

Steve: Oh, I'm sure they've seen that.

Leo: Yeah. Oh, they have them all. They have a museum of these things.

Steve: Jobs is literally rolling over in his grave.

Leo: No, he's not, no.

Steve: Well, no.

Leo: But that's a nice fake.

Steve: Look, yeah, I know. And you said way cheaper than the thousands of...

Leo: $111. $111. But you know why? It's running Android. So you don't have to feel threatened.

Steve: Yeah.

Leo: They're laughing. They're laughing. They don't care.

Steve: They're not threatened.

Leo: They're not threatened. Anyway, continue on.

Steve: So I got what kind of constitutes a piece of errata from Peter McDonald. I only know that because it's PeterMcDonald.co.uk. So he's enough into this stuff that he's got his own domain name.

Leo: Whoo.

Steve: On his own name, which is very cool. Anyway, he wrote: "I think you made a mistake when describing the Schannel issue." And this is referring to Episode 481. "You mentioned that sites running Apache and Nginx would be safe. However. surely this is not correct if they are running on Windows. Granted, if they are using Apache or Nginx on Linux, they would be safe." And he's absolutely right.

So I just wanted to thank you, Peter, for the clarification. Of course, I was sort of conflating Apache-ness and Nginx with Linux, ignoring the fact that, if those two services or either of those server platforms were running on Windows, then they would be running atop the Windows Schannel and then suffer from the security issue that we all ran around and patched ourselves for very quickly. So good clarification, Peter, thank you.

I saw "Citizenfour" last week.

Leo: Is that a movie? What is that? I don't - that doesn't ring a bell.

Steve: You haven't seen it?

Leo: I didn't even hear about it. What is it?

Steve: Oh, my goodness. Okay. It is...

Leo: Oh, it's the Snowden movie.

Steve: Yes, yes, yes, yes, yes.

Leo: Ah.

Steve: Really worthwhile. So if listeners are interested in the Edward Snowden story, essentially what this tells, it's in documentary format, and it's that the guys that Snowden was contacting had the foresight to have a camera rolling from the beginning. So this tells the entire story, sort of behind the scenes, a lot that we've never seen and heard before. And it was really interesting. I thought it was absolutely worthwhile. So I just wanted to say - I had assumed that you had seen it because...

Leo: No spoilers. Does he get caught? I'm just kidding. I'm just kidding.

Steve: I know. So we sort of see everything behind the scenes, including the interview - we see them setting up for the interview that we did see.

Leo: So this is a documentary. I mean, that's not an actor. That's not Edward Norton playing Edward Snowden. That's Edward Snowden.

Steve: Yeah. Oh, yeah. So this is - you get to see much more of him and some insight into his thinking. And, I mean, it is the interviews by Laura and Glenn Greenwald of Edward, where they're, like, you see them, the dawning of their appreciation for the immensity of what he has done. I mean, and you see Glenn's, like, looking through the documents, and his jaw dropping open, and then him asking Edward if this is really what this appears to be and so forth.

So here's my takeaway, though. As I was coming out of the theater, I realized I had an appreciation that I didn't have before and that I don't think you have. Okay. So one of the things that I have noticed is, sort of as I become more politically aware, is that an inherent characteristic of democracy, by its definition of majority voting, majority rule, is it's easy for a majority to vote away rights of any minority. So, for example, gay marriage is an example. Gay people are a minority. It's easy for a majority that doesn't care to say, eh, yeah, we'd just as soon you couldn't get married. Or a bunch of white guys sitting around deciding that they don't think a woman's right to choose what she does is important, so we're going to arrange for that to be minimized. Or, on the flipside, people who don't care about guns restricting the rights of those who do.

So my point is that this is inherent in a democracy, that the interests of a minority should be, I mean, they deserve respect, I think. And it's easy to forget that. Well, I'm not a person who has a particular need for privacy. I mean, my text messages are basically, you know, little sweet nothings to Jenny and when I'm going to meet up with my friends for various meals. And my email is supremely boring. I really don't care. And I pretty much know you sort of feel the same. But the appreciation that I got somehow from watching this movie was that I need to respect the fact that there are other people not like me; that people can want privacy, have a right to it for its own sake; and that this argument that we hear, oh, well, if you have nothing to hide, then why do you care, if you're not doing anything wrong and you have nothing to hide, why do you care, well, that's wrong. I mean, that's a fundamental mistake.

And certainly we know that there are people who have a legitimate need for communications secrecy. We know that there are, I mean, that we can easily relate to, people who are working for human rights in oppressive, non-democratic governments, for example, or in corporations where there's extreme espionage pressure against their secrets, they have an absolutely legitimate right to keep those secrets protected. So anyway, I felt a much stronger sense of understanding that privacy is absolutely, I mean, I feel more like I understand the EFF's position, which to me sometimes seems over the top. It doesn't so much anymore. So I thought that was - I'm glad I saw it, and I'm glad I have a better sense for that.

Leo: It's also, and I think this is a really important thing, is that when - you may not have anything to hide. But if a government wants to put together a case against you, and they have access to information...

Steve: Retrospectively.

Leo: ...it's not hard, it's not hard to fabricate a case against somebody who is an enemy of the state. Remember that for a long time Martin Luther King was viewed as an enemy of the state, and J. Edgar Hoover collected lots of information about him, and there's a massive dossier. I mean, this is not - this is in our recent history. So you may have nothing to hide, you think, but that doesn't mean that you're invulnerable to this kind of government pursuit. And so that's another big issue.

Steve: And I've been in business all my life. And there have been situations where there's been contractual conflicts, when I've been deposed by an attorney who wanted to, from my perspective, misconstrue the past. And it's a weird feeling facing somebody with a stenographer who's asking questions with a clear intent to bend the truth. And you find yourself, I mean, you know, I know what happened. And yet I'm on the defense for some reason. Like how did that happen, that I'm having to defend no wrongdoing at all? But that's the nature of the world.

Leo: I can see a scenario where, to be honest, the NSA decides that you are a threat to national security because you reveal information about their methodologies and says, you know, we've really got to get that Gibson and anybody else, Brian Krebs, we've got to get these guys out of the public eye. Maybe a short stint at Guantanamo would help with that. I mean, you know, we're lucky because I think we do, we still live in a free state, and we can express ourselves, and we are somewhat protected. But you want to watch that slippery slope.

Steve: Yeah. One other little bit of miscellany, and this was just - this is more whimsical than anything. A Utah legislator has put forth a bill, which is working its way through the Utah legislature, to cut off the supply of the NSA's water...

Leo: Oh, we're definitely putting that guy in Guantanamo.

Steve: ...to their super secret new installation.

Leo: What?

Steve: I'm not kidding you.

Leo: That's one way to go after them.

Steve: Oh, my goodness. Apparently in one month they used - and I had it in the notes, and it must not have made it into the PDF. So if you can pull up that Ars Technica story...

Leo: I will, yeah.

Steve: ...that I linked to because I had pulled more out. So, okay. So earlier this year...

Leo: 6.2 million gallons of water in a month.

Steve: Yes. The Salt Lake City Tribune published data showing that since July 2013 the facility used 6.2 million gallons of water in one month.

Leo: They have a $29,000 a month water bill. Water bill.

Steve: And in October of 2013 The Wall Street Journal reported that they had experienced 10 electrical meltdowns in the past 13 months for unknown reasons.

Leo: Wow. So the water is for cooling; right?

Steve: Yeah. The water - but we couldn't figure out why their water usage varied so much. And it turns out that they're having some sort of horrible electrical problems there. So they're just unable to keep it cool or to keep it up. There have been, like, huge meltdowns and fires of unknown nature.

Leo: The facility draws 65 megawatts of power.

Steve: Enough to run a town of 20,000 people.

Leo: I have a lot of hard drives.

Steve: Oh, baby, yeah.

Leo: The town that they're in, Bluffdale, is 8,000 people.

Steve: Yeah. And so, yeah, so basically all the electricity goes off to the left. It's like...

Leo: So does this guy, like...

Steve: I don't really understand what his deal is. I didn't go...

Leo: There's a bill. He's put a bill in front of the Utah State Legislature.

Steve: Yes. Legislation. The idea is that the bill prevents interstate or operating with federal agencies that are collecting consumer data. And so that sounds innocuous, but that means that providing them with water would constitute interacting with them or supporting federal data collection pursuant to this bill. And so, oops, sorry, we've got to turn off your water because we can't work with you.

Leo: This is the website of the assemblyman who has proposed the bill, Marc Roberts. Right on the front there it says, "Life, liberty, and property do not exist because men have made laws. On the contrary, it was the fact that life, liberty, and property existed beforehand that caused men to make laws in the first place." I think it's a bad French translation. And then he's standing there raising his hands to the clouds. Interesting.

Steve: Hmm, yeah. Anyway, I just thought that was a bit whimsical and fun. I did get a nice note from an Andreas Gogstad - I'm hoping I said that right - in Sandefjord, Norway. Anyway, sorry, Andreas, if I mangled those. Just yesterday he wrote, on the 24th, he said: "A user came in with a totally" - I don't know how, so it came in, he must have some sort of a facility - "with a totally unreadable hard drive on a private PC."

He says: "I took the opportunity to purchase SpinRite as payment for the recovery job after hearing about it many times on your podcast. I ran a Level 2 scan despite warnings from SpinRite about an 'invalid partition for drive size,' since there wasn't much else to do, and an LBA setting wasn't an option in the BIOS. I was then able to mount the drive on a Linux machine, which I did in Read Only mode for safety, and recover the photos this client needed for her building permit application. Could you explain the partition message, and what the best solution in those cases would be?"

And so, first of all, thank you, Andreas, for sharing your successful recovery of - apparently this was very important information on, as he said, a totally unreadable hard drive. SpinRite fixed it, brought it back to life, made it readable again. SpinRite is very cautious in its operation. And so what it does is it looks at what the BIOS is telling it about the drive, compares it to what the partition table is telling it about the drive, and looks at what the drive is telling it about itself. And if those things don't mesh up, it will say, uh, you know, something seems wrong here.

What happened was he had moved the drive from one machine to the other, so it was then being seen by a different BIOS. And so that was where this matchup problem came. The warning is just to be, like, make sure you want to do this. And it's normally safe to push through it. That will completely go away in 6.1 because we're no longer using the BIOS, and we don't care what the BIOS thinks. So that will be one of the nice things happening with the next release of SpinRite. But until then, it's just a function of a mismatch caused by moving the drive to a different motherboard with a different BIOS. SpinRite says, uh, we can proceed, but just want to let you know there's some reason that things are confused. And as we see, it works anyway.

Leo: You're like the Honey Badger of file recovery tools. We don't care what the BIOS thinks. SpinRite don't have to care.

Steve: Exactly.

Leo: Steve Gibson. Leo Laporte. What is the EFF program called? They have a good name for it.

Steve: It's called Let's Encrypt.

Leo: That's a good name. I thought there was something else, but maybe not. All right.

Steve: So the idea is, and the reason the EFF is a principal actor here, along with Mozilla and Cisco and Akamai and IdenTrust and some guys at the University of Michigan, is that we're always talking about the idea that more of the Internet should be encrypted. But there are several barriers, practical barriers to that being done. And it's a matter of tradeoff. I mean, I'm using HTTPS at GRC, and have been forever, originally because that was the way for ShieldsUP! to get the correct IP of the user. When users who wanted to check their ports were behind an ISP with a caching proxy, GRC would see the web queries coming from the proxy. So if I tested those ports, I'd be testing the ports of the proxy server, not the user.

But by establishing an SSL connection, since the proxy can't proxy a secure tunnel, exactly as you were describing with proXPN, I would then see the queries coming from the actual users' IP. So there was really no other purpose I had for security. And then I later, years later, added eCommerce, and so I did need to be able to do secure forms to allow our SpinRite customers to be able to put their credit card information in securely. And then, of course, just because being a hundred percent secure, actually it was shortly after the whole HTTPS Everywhere effort, where it was possible to tell browsers never to not accept - wait, never - to only accept...

Leo: Yeah, never to not accept. That's good. A double negative, but okay.

Steve: To only accept secure connections. I said, okay, fine, let's - I want to do that. So now GRC is completely secure. But you could argue that a blog has no - okay. If it's really free and easy to have security, yeah.

Leo: Why not?

Steve: Why not? But if it, first of all, if there's lots of hoops you have to jump over to configure the server, and if you have to pay hundreds of dollars a year, then that's crazy. Or even if they're cheap certificates, and you can get them for four or five, it's still, like, well, okay, what about when they expire and suddenly people are sending me email saying that they're coming to my server and they're getting warnings? It's like, the point is, if there's really a low need, a low level of need, then just all of the annoyance of doing it keeps it from happening. So the EFF, along with these other companies, are saying, how can we make this dead simple? How can we make this one click?

Leo: There's more friction, too, not merely the cost, but just the annoyance and difficulty of installing it. We have some nice certs from DigiCert, but we haven't installed them yet because it's a nontrivial thing to do. It's work.

Steve: It takes, yes, it takes learning something about your server that is - it's funny because the example that I use is that, with certs dying very two or three years, that's just long enough for me to have forgotten how to do it all. And then I have to go, okay, you know, it's time, scratch my head and figure it out again. So these guys have arranged to automate, the Let's Encrypt initiative automates this process.

Leo: Yay.

Steve: They put - yes. And I've now read the entire specification front to back, 36-page RFC, and they've nailed it. They absolutely got it right. But it is crucial to understand what it is that you get for free. It is absolutely the case that you get for free what many certificate authorities are now charging for. So in that sense it is absolutely true competition for what all certificate authorities are currently making some money selling. It is not at all competitive with what all certificate authorities should be charging money for, meaning that it is - okay.

And so what that is: Remember that there are broadly three categories of certificate. There's the so-called domain validation certificate, a DV, which its only assertion is that the certificate is bound to this domain, that is, the certificate is just a certificate for the domain. It doesn't assert anything about the ownership of the domain. So the certificate will make a claim about the ownership. So even a domain validation certificate, it'll have a common name and an organization associated with it. It's important to understand that this doesn't validate that, that is, a domain validation cert is really, it's just saying - it's like the minimum required to establish an SSL/TLS connection to a remote website.

The next level up fixes, sort of at the weakest level, the fact that the domain validation isn't really making an assertion about the organization. That type of certificate is called an OV, an organization validation. The problem is that there's no visible indication of which is which. And so one of the things that I do wish we had, and I don't know how we could get it in practical terms, is three levels of indication in the web browser where we currently only have two. We currently have either EV, which is the Cadillac, the Extended Validation certificate, or not EV. And in the grouping of not EV is both DV and OV. That is, essentially nothing but the validation of the domain name and really not the organization, or the organization validation, which would be nice to somehow be able to assert because that requires - that's not what this system does.

The whole Let's Encrypt thing is automated, and it is free specifically because it can be, because you're able to cryptographically validate the domain control with full automation. You cannot validate organization, that there's a corporate entity, without involving people. And so that won't and can't be free. That's what you need the CA for. Or, going further, if you want Extended Validation and an indication in the URL, the green coloration or glow or whatever it is, depending upon what browser you're using, then that takes extra work. Okay.

So we need to understand that all this is is that weakest form of validation. And CAs are probably going to lose that business to the degree and as this takes hold. And, frankly, that's a big business because Extended Validation certs are still the minority. But this is sort of inevitable because they actually weren't doing much to get any money for just doing a domain validation. I mean, so little, in fact, that it could be completely automated with the so-called ACME protocol, A-C-M-E...

Leo: That's it. That's the name I was looking - that's the funny name.

Steve: Yes, Automated Certificate something. Oh, Certificate Management Environment. So you could tell that the acronym came first, and they had to reverse-engineer...

Leo: They call those retronyms, yeah, yeah.

Steve: Retronyms, yeah. It's like, okay. We know what you meant. So, okay. So how does this work? There is an agent, some software which will be written by some pro Let's Encrypt people for the various platforms. I'm sure that Linux will have the first one. And the example on the EFF site and their video is of a Linux install where basically two command lines act to get something, to get the package from the repository, and then install. And this thing runs.

The idea is that it contains the knowledge of how to configure the server if it had a certificate. It has the knowledge of how to ask the OpenSSL crypto library to generate a certificate. You have to ask, there's an interactive session where you tell it a few things, like what domain name you want and your organization name and the other things that certs typically have, you know, like geographic location and so forth. Then it has the ability to query the Let's Encrypt CA.

So part of this service is a new certificate authority that will only be issuing, probably, domain validation certs through this automated protocol. Oh, and it has to be preconfigured with the URL of the CA. So that would be bound into this package. And of course the domain name of whatever it is wouldn't be changing often. So that's not a problem. So it generates a Certificate Signing Request, a CSR, which is the normal process we all go through when we're creating a certificate. And that certificate signing request involves the generation of a public key pair. The server holds onto the private key. That's its crown jewels. It never lets that go. But the public key is part of the certificate signing request, which it then is, when appropriate, it sends it to the Let's Encrypt CA.

But first what happens is it needs to prove that it has valid control of the domain that it wants to get the certificate which it has just created that it needs to get signed. That is, we don't want some random person generating a certificate for GRC.com and submitting it through this automated protocol to have it signed because then they would have a cert for GRC.com. I don't want them to have that. They would still have to jump through some hoops to use it because GRC's IP is in DNS, and so people looking for GRC.com come to my IP, not this bogus entity's IP. So more needs to be done, but still we know we don't want them to get a cert for that.

So the protocol has three main purposes. One is to validate that you have control of the domain you are wanting activity for. And that activity would be issuing certificates and revoking certificates and renewing certificates, you know, fundamental certificate operations. So there are two ways that you can prove, through this automated protocol, through the ACME protocol, that you prove you have control of the domain. The first is that you place content which the CA provides on your web server, which is then made publicly available, and the CA obtains it. So there's a well-defined, simple, JSON-based, you know, JavaScript Object Notation, JSON-based protocol where the user, the client of this ACME protocol says I want to work with a new domain name. I want you, CA, to ultimately issue me a certificate for this domain name. So let's do that.

So the protocol, the ACME protocol at the client side generates another completely separate public key pair and, again, holds onto the private key and sends the public key to the certificate authority over the ACME protocol, along with a domain name that it wants to associate, essentially. So the CA has a public key and a domain name that the client says it has a right to have certificates issued for. The certificate authority challenges it then with a session ID and a cryptographic nonce, so that none of this can be repeated, and a list of challenges, a list of ways it can prove control of the domain. Currently there are two.

But the protocol is meant to be open and extensible. And in fact it can technically be used for other things than just managing a domain name, though that's all it's defined for currently. So this challenge comes back to the client running the ACME protocol. And the client can choose which of the things it wants to do. One of the things it could want to do is to accept the challenge to put some content, which is just a bunch of Base64, so ASCII-encoded random gibberish, on a certain path in terms of the directory path of the website on the server. The path always begins with ./well-known/acme-challenge/ and then a path which the CA has provided. And again, that's a hex-encoded bunch of random stuff. Hex so that it is valid for a path name.

And so the CA says, essentially is saying in this challenge, here's a blob that I want you to place at this blob location. Let me know when it's there and ready. So in accepting the challenge, this client establishes, takes the data from the certificate authority, arranges to have it appear in public on that path at .well-known/acme-challenge/ and then a gibberish path, and then says, okay, I accept the challenge. I'm ready to go. The CA then makes a public query at the domain name that they're negotiating over at that well-known/acme-challenge/gibberish page and then obtains what's there, which should be the random gibberish that it gave the client.

And so in completing that loop, involving looking up the domain name, getting the IP address, making a query, essentially what that tells it is, with provable security, that the entity that the CA is in communication with is able to affect and influence web pages at that domain. So it has control of the domain. And so in my example of some bad guy trying to get a certificate issued for GRC.com, they'd have no way of influencing my server, of arranging to get a page to appear on a whim on a certain location down my own public server space.

So what the successful accomplishment does is to bind the public key, which was generated just for this, to the domain name at the CA. So that, once done, establishes that the entity has control of the domain. Subsequently, things like issuing certificates for that domain require that they be signed by the matching private key, which never leaves the client. And all subsequent operations are signed with the private key that matches the public key which has been bound to this domain name forevermore.

One of the cool things that the system does at that point, once there is an association that's been made at the CA, is it ups the ante for all future attempts to make a binding to that domain name, which is, when I read that in the spec, it's like, oh, nicely done. Because that's what you want. The idea is the first person to use this Let's Encrypt system to issue a certificate establishes this relationship. And in doing so, nobody else can establish a relationship for that domain name without being able to prove they're the same entity that originally did it. So that's another application for the matching private key.

But then there's also something known as a recovery code, which the CA sends back as part of this initial binding protocol, which should be, can be stored separately. The idea is to - and it's actually very much like SQRL's rescue code. It's offline, and it's a "get out of jail free" card if you screw something up, if your hard drive crashes, if you lose all of your cryptographic stuff. If the worst happens, how can you - and because we've upped the ante on making these associations, how do you say no, no, no, it's really me? Please, I need you to issue me a replacement certificate because I've lost mine, but it really is me.

Well, this recovery code is, again, long and random. And only the person who originally made that association would be able to do it. And what's cool is all the recovery code does is forgive the binding. That is, essentially it says, okay, you still are going to have to reprove again that you still have control over the domain the way you originally proved it, but we're going to let you do that. So very nice aspect of the protocol.

And one other very neat thing that I saw there was when you vet - so the next aspect is you then issue a certificate. So the client takes the certificate signing request that it made. And it negotiates a transaction with the CA saying, okay, I've got a CSR, Certificate Signing Request. I need you to sign it. And it signs that request with its private key, which it uses for proving that it is the rightful requester for activities in this domain, sends that to the CA. The CA looks through the various fields in the certificate to make sure that they're all valid. For example, it would be possible for you to request things that you shouldn't request, like the right to sign other certificates. It's like, oh, no, no, no, can't do that. So it'll strip out things that you're not able to do. And then, when everything looks right, it will sign that and return the certificate to the client as part of this negotiated transaction.

One of the other things it returns, which is what I was going to say I thought was another really nice aspect of this, is a URL of a simple GET query which the server, the client running in the user's server, can use to get a renewed certificate any time it wants. The idea would be it would still have to fit within whatever time horizon constraints the overall certificate has. But, for example, you could use this to issue short-life certificates. If you wanted to experiment, for example, with a protocol that we've discussed before, where instead of a long-life certificate with the need to revoke, you instead arrange to issue short-life certificates and don't worry about revoking them because they're going to expire in a couple days.

So the system is a platform for experimenting with that. And they've already incorporated one of the components you would need into the protocol, which is a way for the server, daily, to simply say give me a certificate, a refresh of my existing certificate. You can't change any of the parameters. You can't change the keys or features or anything. But you could say give me a certificate that's good for four days. And that's done with a simple GET request. And the reply to that GET request is another signed certificate good for some length of time. And this makes that very simple. So clearly, if the world is going to be switching to short-lived certificates, we need a clean, simple, fast way to reliably get them periodically. And this is already built in.

The way I explained of putting content on a page is one of the two currently defined means of proving you have control of the domain. The other one is a use of SNIs, Server Name Indication. That's the technology where a single server that supports SNI is able to, at a single IP, support different hostnames. And SNI is used to disambiguate which certificate should be returned to the client who is requesting an SSL connection. The idea there is that you could use this system for non-web-based applications. Clearly the first application is a web-based application because you need to put a web page up at somewhere publicly available at a random gibberish page name in order to obtain the content and verify ownership. But you can use SSL and TLS for non-web things.

And so using server name indication, the client chooses that as a means of proof in the protocol and creates a self-signed certificate where a random gibberish domain name is appearing in the certificate, along with the domain that they're asking for control over. And so they respond to the challenge that way. And then, again, the CA makes a request at this random gibberish machine name dot domain name dot whatever, and verifies that it's able to establish a server name indication enhanced TLS connection that way, and there again proving that the client has control of the domain. And at that point then the association is established that allows certificates to be issued and revoked and so forth.

So anyway, it's truly elegant. It's simple, won't be difficult to implement. None of this is hard. The spec is all open. The plan is to bring up a CA that operates this protocol in second quarter of 2015, so in the late spring, early summertime this should happen. And I forgot to mention that once the client obtains the signed certificate, since it is then - it contains sort of the little mini expert system that knows how to configure the various web servers on the server platform. It does that. It puts the certificate where it's supposed to go, and the configuration files to bring up HTTPS on the server, and you're done. So what the user sees is they run this, or install this.

Oh, and I should also mention that clearly it won't be long before this is built in. Why wouldn't Apache build this into the server, so you don't even have to add an add-on module? It's like, yeah, once the system exists, just have it there. So you'd have to be able to get it for installations that wanted to add it. But the idea is that the operator of the domain either starts this process or loads it and starts it, answers some simple questions about the domain name they want to secure, their organization and where they're located and so forth, the standard things that go into the certificate. They press a button, and then hum a short tune, and then up pops a dialogue saying, okay, you're now operating a secure server on that domain. Everything else is done for them. And it can be made transparent because the lowest level of authentication, just asserting that I have control of this domain, that's all automated. And I just think this thing is 100% cool. I'll be very surprised if it doesn't take off. I think it's clearly going to.

Leo: I like your idea of Apache just building it in.

Steve: Yeah.

Leo: Wouldn't that be cool.

Steve: Yeah. I'm sure it'll happen. It's like, hey, just, you know, I mean, it might even end up being defaulted on. It's like instead of you having to go and do it, part of installing is, hey, you know, unless you tell us you don't want to, we're going to bring up security on this newly setup server. So tell us a few things, and we'll go get the cert for you. I mean, why not?

Leo: Why not? Let's just do it. Yeah, I can see a day when everything's encrypted. Why not? Exactly.

Steve: Yeah. Exactly. In fact, that's be a great...

Leo: Why not? Why didn't it? Why didn't it?

Steve: That'd be a great name for the service.

Leo: Why Not?

Steve: Why Not? Yeah.

Leo: I seem to remember, was it Tim Berners-Lee, I'm trying to think, it's one of the early web pioneers. Oh, no, no. It was when we interviewed Vint Cerf, who of course is considered by many the Father of the Internet. And I asked him why built-in end-to-end encryption wasn't part of the original spec, because it could have been. He said, well, actually what I asked him was, "Would you have done anything differently?" And what he said is, "I would have built in end-to-end encryption." But we didn't at the time...

Steve: Any idea.

Leo: Any idea, think it would be needed. No one envisioned what has happened. And I think also there was overhead in those days, those slower machines. There was some overhead.

Steve: Yes. Remember that ping excited them.

Leo: Yeah. They were still...

Steve: That was like, oh, my god, I got an echo from my ping.

Leo: They were still happy with fingering.

Steve: Exactly.

Leo: So we've come a long way, baby.

Steve: Yeah. And, I mean, the crypto world, as you said, there was overhead associated with it. But also remember there was all that export nonsense. I mean, our government...

Leo: That's right. It was illegal. Yeah.

Steve: Crypto was a munition.

Leo: Right.

Steve: It was categorized as a weapon.

Leo: Yeah.

Steve: So, yeah.

Leo: Well, here we are. We're going to retrofit the Internet for the modern world. And this is a big start. This is great.

Steve: Yes, it's a really great piece of work. And if this does, I mean, I will still pay the going rate, happily, because I want, I mean, because a lot of...

Leo: You want EV certs; right? You want the green.

Steve: Well, yes. And you know, when you think about it, the prevalence of free domain validation certs, it almost makes the better certs more valuable because it will be clear, it'll sort of filter out into the ether that, oh, yeah, there's a lot of encryption, but all it's doing is encrypting your data. Because the assertion strength of the organizational association will fall because it'll be understood that all this is doing, I mean, this is just sort of an automated thing. And so, yeah, it's encrypted. But we don't - but what that asserts in terms of the company you're talking to, that ends up getting - that ends up being weakened by this because you're not having a human-to-human contact interaction the way we do today. And so that actually makes the ones you pay for more valuable because they're able to make that stronger assertion.

Leo: Yeah, I agree.

Steve: Yeah. And so...

Leo: It's good all around.

Steve: I'll happily pay for my green EV status because I want it to be known that, yeah, this is actually Gibson Research Corporation, and that a human has verified in order to put that stamp on the certificate. And I'm happy to pay for that. And a lot of people will be.

Leo: Yes, indeed. Well, you know what, a lot of people are willing to pay for Security Now!, but we don't charge them. So there. We do Security Now! every week at this time, Tuesday afternoons, 1:00 p.m. Pacific, 4:00 p.m. Eastern time, 2000, I'm sorry, 2100 UTC. And you could come and watch live. We like that. But you don't have to. You can always get it on-demand after the fact. Steve has the 16Kb versions of the show. I wanted to say megabit, but no, kilobit versions of the show. They're barely audible. It sounds like Steve's coming to us from 1924. But that's...

[Crosstalk in muffled voices]

Leo: But you can also read transcriptions which actually have the highest fidelity of anything we do.

Steve: And Leo, for what it's worth, the low-quality ones are very popular. Thousands a week get downloaded.

Leo: Really.

Steve: Per episode, yes.

Leo: Interesting. Huh. Well, I guess somebody wants them. You know, if you have bandwidth caps or that kind of thing, or you just don't care how crappy it sounds, there you go.

Steve: That's right. There you go.

Leo: Well, more than 70, to be fair, more than 70,000 people download the high-quality audio and video versions.

Steve: Yup, yup.

Leo: So, you know. I mean, it's a percentage of your - more than 1%. You can get also SpinRite at GRC.com. That's Steve's fabulous hard drive maintenance and recovery utility. Everybody should have it. If you have a hard drive, you need SpinRite. And lots of freebies that he just gives away out of the goodness of his heart. And that's where you'll go if you have a question because I guess next week, the Internet willing, we'll have - if the creeks don't rise - we'll have a question-and-answer session. That would be...

Steve: You know, everybody's going to be in a turkey coma, even the bad guys and the attackers. So we'll probably have a very quiet week.

Leo: The tryptophan.

Steve: That's it.

Leo: You know what, Steve, it's a U.S.-only holiday. I think that the Canadian hackers are going to work overtime.

Steve: I heard that Thanksgiving had spread. Is that not the case?

Leo: Thanksgiving, I think, is American. Where else would it go? It's the Pilgrims landing.

Steve: I did hear, I heard that something had...

Leo: And then Squanto brought corn. And they celebrated...

Steve: Quanto?

Leo: Yeah, Squanto. You've heard of him. Squanto.

Steve: Okay.

Leo: See, you don't come from New England. In New England you learn these things. You go, you visit the Rock. They call it Plymouth Rock, where the first toe from the Plymouth Pilgrims set foot.

Steve: Landed.

Leo: Yeah. And then you learn all about Squanto. You can go to Plymouth Plantation, where people dressed up like Pilgrims who pretend they've never heard of matches. It's fun. You try to trick them. And then you have your Squanto. He brought corn. He brought sweet potatoes. And because they were celebrating surviving, you know, their first year in the...

Steve: Yeah, there was a bunch of ground-kissing, probably.

Leo: Probably.

Steve: Yeah.

Leo: But I don't see any other reason - now, they have Thanksgiving in Canada, but they do it in October because they're - I don't know why, because they do it...

Steve: Canadian.

Leo: They don't have Pilgrims. They're Canadian. But I can't see France celebrating Squanto bringing corn to the Pilgrims. It just doesn't seem like...

Steve: Maybe it was Franois.

Leo: Franois.

Steve: Something, could it have been Halloween? I don't think it was Christmas. I just...

Leo: Halloween has spread. Halloween has spread. That's what you're thinking of.

Steve: Oh, that's what I'm thinking of, then.

Leo: Everybody celebrates Halloween now. Absolutely.

Steve: Well, okay.

Leo: Who doesn't like to put on a sexy costume and scare the kids? That's fun.

Steve: They didn't have Squanto.

Leo: They don't have no Squanto.

Steve: Okay.

Leo: They don't know Squanto. I don't know what happened to me. I'm sorry. I apologize. I lost my...

Steve: This is all bonus time that...

Leo: Bonus.

Steve: Yes.

Leo: Because vacation is coming up.

Steve: Most of the audience has already hit stop.

Leo: I hope to god you're right.

Steve: There's no other genius coming. No jewels of wisdom at this point.

Leo: Have a wonderful...

Steve: We're all - we're both punch drunk.

Leo: We're punch drunk. Have a wonderful...

Steve: And Elaine is thinking, why am I still transcribing this nonsense?

Leo: Yes, you can stop typing now, Elaine. It's okay. Especially that last thing. Cut that out. So are you going somewhere? You going to see Mom for Thanksgiving? Or what are you doing?

Steve: Actually did. Jenny and I made a pre-Thanksgiving trip up.

Leo: Oh, that's nice.

Steve: A couple weeks ago. And I'm so happy not to be traveling during the next couple days. And Thanksgiving is Jenny's No. 1 holiday, Squanto notwithstanding. And she has...

Leo: Well, because what's not to love about Thanksgiving? You sit around, you eat a nice meal, you watch some football, it's a great time. No pressure.

Steve: That's like the one she really cares about. So she throws a big party. All of her friends come over.

Leo: Oh, fun.

Steve: And, yeah. So I get to be there for that because we've already done family.

Leo: Good.

Steve: Yeah.

Leo: Wonderful. Well, have a great Thanksgiving. And we'll be back here next Tuesday at our normal time.

Steve: Will do. And yourself, as well...

Leo: Thank you.

Steve: ...and all of our Squanto-loving American listeners.

Leo: I'm going to celebrate Thanksgiving by decapitating the Seattle Seahawks at the '49ers game.

Steve: Oh, you've become Mr. Sport. When did this happen?

Leo: Not really. I couldn't care less. But my girlfriend makes me go.

Steve: Oh, okay. That's - okay.

Leo: Let's be honest here.

Steve: I was going to say, I've known you for years, Leo, and suddenly you're talking about sports like I've never heard...

Leo: I don't know what I'm saying. I don't care. Look at my attractive sweater. That's...

Steve: Well, now, that is - that's very nice.

Leo: The NFL has a whole ugly Christmas sweater shop for every team. And this is actually the Philadelphia Eagles. I'm not - I don't have the San Francisco '49ers one. That makes it even more ugly. So have a wonderful time. We'll be back next Tuesday.

Steve: Will do.

Leo: Everybody stay safe. Don't let the Stuxnet or the Flame or the Regin infect your system. We'll see you next time.

Steve: Bye, Leo.

Leo: Bye.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/