Listener Feedback #149

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-365.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-365-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Yes, there was a big, big, big, big Microsoft Second Tuesday Patch. He'll tell us all about it. Also a new version of Flash. That and your questions and Steve's answers, coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 365, recorded August 15th, 2012: Your questions, Steve's answers, #149. It's time for Security Now! now, ladies and gentlemen.

Steve Gibson: After that wind-up, I have to ask you, are you recording this, Leo?

Leo: Yes, I am.

Steve: Okay.

Leo: Let me put it this way. If you're hearing this, folks, I must have recorded it. And if you're not hearing it, I didn't. But then you wouldn't know. It's a conundrum wrapped in a riddle.

Steve: We deal with these sorts of things every week, Leo.

Leo: Steve Gibson's here. He's Explainer in Chief, the guy in charge of security and

privacy on the TWiT network. And it's really interesting because this is easily the nerdiest, geekiest, the hardest-core show on the network by far, and one of the most popular shows. The numbers continue to go up. And I just have to think that it's a real demonstration that people want hardcore tech. They like the information.

Steve: Yeah.

Leo: So today we are going to do the Q&A that we deferred last week because of Mat Honan's - so now we're once again Mod 1.

Steve: Well, we're at Episode 365.

Leo: If you start listening now, one show a day, you'd finish next August.

Steve: Same time next year. So we have our Q&A this week, and a little bit of security news, not too much, some interesting questions. The topic of last week generated a lot of interest, the whole issue of authentication. And I started to put together some commentary because I realized there'd been sort of a critical mass built up in me, and I was thinking I would do it this week, but I'm going to defer it next week to give it the treatment that I want to. So some commentary about the issue of authentication, the problems that we have, and the next step we have to take. We've trained users now about passwords. Do not use "monkey" as your password.

Leo: As I did, by the way, for a long time.

Steve: It's surprisingly popular for some reason.

Leo: Really, wow.

Steve: Yeah, it's bizarre why random people would all choose "monkey" for some reason. But don't use "monkey" and don't use "password." And try to use different passwords on different sites. Those lessons are sort of sifting out into the ether of understanding. And I think it's time to take the next step, and I'm going to talk about what that is next week. Not as the topic. That's not worthy of a topic. But to deliver a little bit of a commentary. And in the meantime we'll do Q&A this week.

Leo: Well, that sounds great. I've got some good questions, which you have come up with. Leo Laporte here; Steve Gibson there. We are ready, Steve, to get the latest tech news before our Q&A. What's going on out there?

Steve: The big news is rather routine, unfortunately, which is we are on Wednesday after the second Tuesday of the month, which is always the day after Microsoft's Patch Tuesday. These are rather important this month. They're a little frightening. We've got

nine sets of patches. More than half of them have critical ratings. They address at least 27 security holes in Windows and related software, and some of them look kind of frightening.

There's one set of four privately reported vulnerabilities that affect IE, all versions 6 through 9, and that's a remote code execution problem. Microsoft writes that the most severe vulnerabilities could allow remote code execution if a user views a specially crafted web page using IE. Then they go on to explain that an attacker would have no way of forcing you to go to a website, blah blah blah, as their CYA because we now know that that's not the way these things work. People send you links in email. High-reputation websites get compromised with exploits installed on them unbeknownst to their webmasters. So that's one. There's IE, all versions.

Remote Desktop has a problem where, if you have the remote desktop exposed to the Internet, it's possible to send it some packets to execute code remotely on your machine. Windows Print Spooler's got four privately reported vulnerabilities which they fixed. And something we don't often see, the Windows Common Controls, that's the library of all the components that make up a window - dropdown list boxes, the tree list, the text box, the buttons and things. To have a remote code execution vulnerability there is uncommon and very worrisome. And they say of that the same thing. The vulnerability could allow remote code execution if a user visits a website containing specially crafted content designed to exploit the vulnerability.

There's a problem with their JScript, which is their version of JavaScript, and their VBScript. It affects 64-bit versions of Windows, which are generally regarded as more secure than the older 32-bit instances. And even something, a remote code execution problem with Exchange Server. So this will require a restart, but it's not something I would put off for very long because many of these are privately recorded, but several are public. And this is not something you want to leave your system exposed to.. There's too much opportunity for exploit.

And Adobe, just yesterday, on August 14, also Tuesday, they released news of a critical Flash Player update across all platforms - Windows, Mac, Linux, UNIX. And what's odd is I'm not seeing any push from them for this.

Leo: I did. I got a new Flash yesterday.

Steve: Oh, you did. Okay.

Leo: But it was on a Mac, yeah.

Steve: Okay. I've restarted IE. I've clicked on Check My Version. I also fired up Google specifically to see whether it was going to update me on its own. At least on the Windows platform I'm not seeing it yet.

Leo: Maybe you have to go to a site that uses Flash. I bet you that's it.

Steve: I've tried that, too.

Leo: Didn't do it, huh? Hmm.

Steve: And clicked on the player or had the browser see that I'm instancing the player. What I have currently is the most recent vulnerable one. 11.3.300.270 and all earlier players are subject to this vulnerability. The reason I'm bringing this up is it is being exploited actively in the wild through targeted attacks. In this instance, at the moment, it's a malicious Microsoft Word doc which exploits the ActiveX version of Flash Player. Oh, it's IE, in my own notes here. And I tried Firefox, and I tried Chrome, but I didn't fire up IE and use Flash. So it may only be a problem with IE.

But you want to update to the latest one, which is .271 rather than .270. And that is being actively exploited. So you can go to Adobe.com/software/flash/about, and that will run a little animation and show you what your current version is, so you can quickly see whether you are yet at 11.3.300.271, which is where you want to be.

And I noticed something that I thought some of our listeners would find interesting. We've talked about the Khan Academy and what a nice piece of work all of the resources there are. They're just in the process of launching a new computer science site which focuses on what they said was the critical early adolescent years where children broaden, or unfortunately, in some cases, narrow their interests.

Leo: More often narrow, yeah.

Steve: Yes. And their identity before high school. What I thought was interesting about this, TechCrunch carried the story. And they said, "The lessons don't get much more complicated than basic algebra, and how these intuitive mathematical concepts can create powerful, artistic, videogame and website experiences." What they've done is, and this is not the first time this has been done, but I think it's nice to see it again, they're merging computer science education with graphics. So they're using sort of the compelling visuals of graphics in order to teach concepts of computer science. And of course you'll remember the LOGO programming language with its so-called "Turtle Graphics," which was an early effort in the mid- to late '60s to interest young people in computers by making something accessible.

Leo: Yeah. It was very cool, yeah.

Steve: Yeah. And I was thinking of it because it also relates to the page I just did where I had the way magnetic data storage functions shown graphically. And you and others immediately jumped into it, wanting to see how I did the things that I did. So there's definitely a connection between something as visual as a graphic presentation and the code behind it that makes it go. And in this day and age now, with videogames being so phenomenal, we understand that there's a lot of code behind there to make it all happen.

So anyway, the Khan Academy launching a computer science curriculum that is based on graphics. And in the little onscreen sample that I saw, it looked like some simple JavaScript that actually they showed implementing a version of Pac-Man, and showed the code to do it, where you could see the way they were actually creating the graphics to draw the little ghosts that were going around the maze. So I thought that was cool.

And from the Twitterverse I had one tweet that caught my attention, and this relates a little bit to one of the questions that we'll be encountering in a minute. Someone named Paul Morgan tweeted, he said, "Just listened to Mat's story. Those kids are in a bad situation, if caught. There is no 'pressing charges' when it's a felony." Which I thought was a good point. Remember that Mat said that he agreed with the hacker not to press charges. But this was serious stuff that these kids were up to, which we will discuss in one of our upcoming questions today. So I thought that was worth noting.

And I did have a really well-written and really neat note from a Joe Kozak, August 8th, so just recently. The subject was "Another satisfied customer, thank you." He said, "Dear Steve, I have an older HP computer with a partitioned 80GB hard drive, C: for me, and D: for HP programs, if they needed to be reinstalled. For several months I was getting warnings from my hard drive. It would make clicking sounds. I foolishly ignored them and failed to back up the drive. Then one day the computer would not boot. It would get to the Windows screen, go no further, then reboot. This loop continued. I shut it down and tried again, dot dot dot, many times, but with no success. I always got the same result.

"I thought, no big deal, I would remove the hard drive, use a USB cable, and read the drive from my other computer, back it up there. No such luck. My other computer didn't even know it was attached. Sure, it saw the D: partition, which I didn't need. But it did not see the C: drive. This is when I started to panic.

"I started to research my situation and learned about SpinRite. I watched the videos and read many, many, many positive testimonials. I must admit, even after watching the videos and reading the testimonials, I was still skeptical, but decided to purchase SpinRite anyway due to the 30-day absolute satisfaction guarantee," which of course we are. Anybody who is sorry they purchased it, who regrets their purchase, we're happy to put the money back on their credit card. So he says, "After running SpinRite for a straight 23 hours, it seemed like nothing was happening. It remained on one sector for over 13 hours."

Leo: Wow.

Steve: "So I emailed tech support to ask if that was normal. They informed me" - "they" meaning Greg - "informed me it wasn't normal, but it was possible. So I let SpinRite continue working. After an additional five hours" - now we're at 28 - "I wanted to give my PC a rest, so I again emailed tech support, this time to ask how I could stop SpinRite without losing its location. Again, tech support quickly replied with detailed instructions, and I stopped SpinRite.

"At that time I was curious to see what would happen if I turned on my computer without booting SpinRite. This time it did not loop as before. I could tell the computer wanted to boot, so I let it continue trying. After about 10 minutes, it booted, and my desktop appeared. It was like magic. I absolutely could not believe my eyes. Without shutting it down, I spent the next five hours backing everything up. The next day I ran another long session, 20 hours of SpinRite, from the place it left off the first time, to recover additional data from the remaining sectors. Throughout the next few days I ran several more sessions.

"At the end, it ran for over 113 hours, spread out over eight sessions. Additional files were located and backed up as it went. My final analysis is that it recovered 99.999 percent or more. I noticed only a few files in one folder which were not recovered.

Conclusion: Low cost, 30-day absolute satisfaction guarantee, excellent technical support. Purchasing SpinRite should be a no-brainer. Be patient and let it do its magic. It's the best software investment I've ever made. Thank you, Steve, for this great program. Best regards, Joe Kozak."

Leo: Hey.

Steve: So, very nice story. I haven't shared an adventure like that with our listeners for some time. So when I saw that, I thought, ah, that'll be fun.

Leo: All right. I've got questions. You ready, Steve? You got answers?

Steve: Speaking of technology, yeah.

Leo: Speaking of technology, this is Question #1. Seth Jameson, Fort Collins, Colorado. He was reading that - remember we talked about that animation that Steve did a little while back, and he said the JavaScript - view source, you'll see the JavaScript. Steve, thanks for sharing the code you used to produce the amazing magnetic storage animation which you'll be using in a video of a forthcoming SpinRite. I've been studying it, and I've learned some neat tricks. But I have a question. The Scalable Vector Graphics, or SVG drawing system, predates HTML5 and the canvas API. Did you think about using SVG instead? And if not, why not?

Steve: Well, okay. I chose two questions out of many. I was really surprised, and I thought it was neat that our listeners were as interested in what was going on under the hood, behind the scenes, much as you were, Leo, with that little demo. So I thought I would just take a second to talk about it a little bit, since this is a forum for doing that. The scalable vector graphics has been around longer than the canvas API. Apple actually originated the canvas API, and it then became a standard much later than scalable vector graphics.

The SVG objects are themselves objects, in the sense of, for example, if you have a rectangle and a circle, for example, they exist as things in what's called the DOM, the Document Object Model, of the page. So in the same way that in an HTML page you've got divisions and tables and images and so forth that are part of that document structure, the drawing objects, the scalable vector graphics are sort of the same class, the same level.

So what that means is that a circle, for example, will have a defined center and a radius and then some things like color and outlying color and outlying thickness, properties of that circle. And if you were to want to animate that, you could change any of those properties, and the browser would show you the new circle. So the circle exists as something virtual, physical, actual, as part of the document. By comparison, the canvas API is just drawing.

Leo: So you have primitives like moveTo, lineTo, arcTo, drawing primitives.

Steve: Exactly.

Leo: Stroke, which will do the drawing itself.

Steve: And the way to think of it is as a bitmap. You define a so-called "canvas," which is a drawing surface, and then the order in which you draw affects the pixels of this bitmap as you draw across it. So you can set a line width and say that it's going to be this color, and then you say "move to this coordinate" and then "draw a line to that coordinate" and, bang, it exists. But there's no notion that the line itself doesn't exist as an object. So, for example, you couldn't change one of the coordinates and have the line move. You'd have to erase the whole canvas and then redraw everything.

Leo: This is not SVG. This is with...

Steve: The canvas API.

Leo: The canvas, the HTML5 API.

Steve: Yeah. So for what I was wanting to do, the kind of graphics that I was envisioning didn't feel to me like they would fit into sort of an object-ish template. For example, we were talking earlier about the Pac-Man, the idea of recreating Pac-Man. That's a perfect example of where you might use the scalable vector graphics because you might define the shape of one of those little ghosts, and then moving it around the maze is just a matter of changing its location, and the browser will update the entire presentation showing these things in a new position.

Leo: That's what they call "sprites" typically in videogames.

Steve: Precisely. As opposed to what I'm doing, which is more sort of freeform graphics that don't really fit into that same object-oriented way of describing things. Arguably, you could do either with the other. But some things just sort of make more sense in one form than another. And also it feels to me like scalable vector graphics is sort of a transient, I mean, it was here first, but it's...

Leo: I think HTML5 is the future, absolutely.

Steve: Yes, that's exactly right. So I was wanting also to invest because it was going to take me a while to learn either of them. Since I could probably do anything on the canvas that I could do in scalable vector graphics, I thought, well, I'm going to do the one that is part of the W3C standard and moving forward.

Leo: This is also kind of intuitive, if you've ever done any graphics. I'm looking at the amplifier, the code to draw the amplifier which is just basically a simple box. And

you set some parameters. "C" is your canvas. You set the color, the fill style, the shadow blur. And then you say what the stroke style is. Then c.strokeStyle = hex 000000. Then you say what the line width is, c.lineWidth = 2; and then you draw it, c.strokeRect; and you give it the four coordinates, the four corners, and it's done.

Steve: Yeah. We were talking about the Khan Academy earlier. And if somebody wanted to play with JavaScript, it is that easy.

Leo: It's pretty straightforward, I have to say.

Steve: I think graphics, yes, I think graphics is a tangible, immediate feedback, kind of like, ooh, look, I just drew something, and then you start asking yourself questions. This is a little bit like the Portable Dog Killer moral, which I ended that story with saying, look, just do something. Just start. And you'll immediately get sucked into thinking, oh, hey, I've got an idea. How do I do this? Or what about if I do that? And before you know it, it'll be 4:00 a.m., and you'll realize you really should have gone to sleep.

Leo: And that's the fun of programming, kids. A lot of sleepless - so let me ask, though. There's animation - by the way, you can see this at GRC.com/animation.htm. You could see this. Are you erasing, for instance, the read amplifier puts a plus and minus up whenever it reads a bit. Are you erasing those pluses and minuses and redrawing them? How do you that?

Steve: Well, that's a very good question because there are, since it is just a bitmap - if I wanted to change the whole canvas, it would make more sense to erase it all and then repaint it all anew. But because this is a bitmap, I can just erase the part that I'm changing. And so, for example, in the plus and minus case, that's what I'm doing. There are only little areas that are changing. And so I go in and just blank those out and then change the plus to a minus and so forth.

Leo: So you do erase, then redraw.

Steve: Yes.

Leo: And actually you're doing a really cool thing with the plus and minus. They're growing and shrinking. So you're really having fun with that.

Steve: Yeah. Again, it's a nice way of playing with a simple programming language like JavaScript and immediately getting feedback, immediately getting some traction for what you do. But do keep an eye on the clock because otherwise the sun will come up.

Leo: This is pretty cool. If you go there and view source, and most browsers will let you do that, actually I'm looking at it in Safari, which has - you open up the

developer menu, then you actually have a really great interface to the JavaScript. But it's only about 800 lines of code, very clearly commented, really easy to understand. And if you just go back and forth, looking at the animation and looking at the code, I think this is a little lesson in itself. I think you can probably understand exactly what he's doing. It's all fairly straightforward. It's neat.

Steve: Well, it is part of this video that I'm putting together. I've finished all of the work. There's actually a bunch more stuff I haven't made public just because I haven't had any reason to. But I'll let our listeners know as soon as I have the video put together because I've done - I also showed the design, the morphing of the longitudinal recording head into vertical recording, and how that increases the density, but what the consequences are to recording much higher bit density because then those pulses begin interfering with each other. They get very close together, and they start pulling towards each other, which has some unexpected side effects, which is part of what SpinRite knows about.

Leo: Is this all integer math, or is it floating point?

Steve: In JavaScript it's sort of typeless, meaning that JavaScript tries to keep everything floating with a lot of resolution. If you do any logical sort of bit manipulation, then JavaScript says, ooh, he's trying to "and" some bits or "or" some bits.

Leo: So smart. Hmm.

Steve: So, yeah, it'll switch it into integers and allow you to do the right thing. So they've really done a nice job. I'm impressed. As I have spent more time with it, I've come away feeling that, wow, here's the language which looks like it's got a future. If only they hadn't called it ECMAScript.

Leo: Well, everybody calls it JavaScript. We can ignore the fact that it's really ECMAScript.

Steve: Yeah.

Leo: I went out and got the four O'Reilly books. And learning JavaScript is, if you've ever used the O'Reilly learning series, you know that they're aimed at kind of programmers, so they're brusque. But there are also some good - there's "Eloquent JavaScript" which is nice for a first-time programmer. There are some good choices out there. And you used a book on web graphics. What was the name of that? Do you remember it? He's looking for it.

Steve: "JavaScript Graphics."

Leo: There you go. It's another O'Reilly book. It's got a - looks like a bison on the

cover. What is this?

Steve: I don't know. It's got a big beard.

Leo: It's some sort of African wildebeest. Question 2, Peter Wilson, Atherton, California: Steve, you missed a trick. Speaking of meat? I see. Love the show and Leo's others, mostly due to how meaty they are with compared with what else is out there. Got it. Speaking of meat, I was looking at your magnetic recording animation page and noticed you're not using the cool new AnimateFrame feature which was specifically created to facilitate animation. Instead you're using the old, general purpose setInterval timer to drive your animation. That's 'cause Steve's old school. I've been listening to this podcast for many years. I have a feeling you would have done what was best even if it might have been more difficult. Is there a reason you didn't use AnimateFrame? Thanks, Pete.

Steve: Okay. Well, I'll answer this question, and then everybody's on their own.

Leo: Okay.

Steve: So, yes. AnimateFrame is clever, and I tip my hat to the designers of HTML5 and the canvas and this feature. What AnimateFrame does is it tells your code what time it is and uses that, together with your browser, to set the optimal frame rate. The idea is that you might have something really complicated that you're drawing on a slow machine or a slow tablet or something else where there's a huge overhead, and this set interval essentially, we all know that animation is the illusion of motion thanks to showing a series, a rapid series of still frames. So even movies, movies are just a succession of still frames shown one after the other, and our brain fills in the gaps, and we see something that's continuously moving.

So the setInterval is a - I use it because I'm saying that I want my code to be called to draw the next frame every X milliseconds, like 30 times a second, so every 30/1000 ms or 1000/30 ms, rather. And so what that does is it allows me to generate that smooth-looking animation because essentially I'm drawing those frames at 30 times a second, which is fast enough so that we see it, we perceive it as smooth motion, even though it's actually not. But it might be that there's a very complex drawing which can't be done at that rate.

So the idea is - and I would imagine users have seen, for example, maybe in the old gaming days, where if you had a slower computer or even a faster computer, I know there was a problem as PCs began to get faster, Leo, remember, there was a problem where you would have to slow the clock down on the processor because otherwise the videogame was running too fast.

Leo: Because people used loops for timing. It was because they weren't tying the timing to the clock. They were just saying, count to 30 and then do another frame. Count to 30, do another frame. Now, that might have worked on the original computer. But then as computers get faster, they count to 30 faster, and it doesn't

work so well.

Steve: Exactly.

Leo: So that was just bad programming, if you ask me.

Steve: Well, so the reason I used the SetInterval was that I wanted to get a feeling for how this looked at 30 frames per second because remember that my real goal was not to show an animation on the web browser page. I did this whole thing in order to capture those frames which I will then import into a video presentation. So my real target was not the web browser and the ability, I mean, it's cool that it works as well as it does on all browser and iPhones and iPads and things. And it's surprising, I'm very impressed with how efficient this whole thing is because it uses, like, 2 percent or 3 percent of the system on my machine. So it's very efficient.

But anyway, so what AnimateFrame does is different. It says we're going to work with the browser to show frames at a rate that we think fits your system, whatever that is. And we're going to tell your code what time it is, that is, when the frame is being shown. So, for example, imagine you had a ball moving across the screen. In the old days, as we were just talking about, with a system based on delay loops, if you had a faster computer, then the ball would move too quickly. But if instead the programmer asked what time it was, that is, with sufficient resolution, then they would know how much time had passed since the last frame they drew, and they could compute the position of the ball for this frame.

So the idea is, instead of moving the ball a fixed amount per frame, you get the time, like the time of day, with a lot of resolution, when the frame is being shown. And so you compute where the ball should be at that time, literally using simple physics to say, at this time, if the ball is moving at this speed, it should be here. And what that does is it makes the experience uniform across computers of any speed. You could have a really slow web browser, and although the ball might be jumpy because it wasn't being drawn often, it would at least be moving at the speed that you intended it to be moving.

So anyway, because my target was generating frames for a video, rather than outputting them through the browser, what I wanted was to see what 30 frames per second would look like, and ultimately I'll be capturing those frames and then merging them into a video presentation.

Leo: Kewl. Kewl. Question 3, Andrew Constantine. He's asking about something that we talked about a couple of weeks ago and that I immediately turned on, it's not on by default, the new Password Iterations feature: Hello, Mr. Gibson. This may have been answered in an earlier episode of Security Now. If so, feel free to point me in the direction of that show. My question is regarding the new Password Iterations or PBKDF2 option. I've read the online help articles on it several times, but the way LastPass explains it's a little over my head. You're really good at explaining these complicated things in layman's terms. That's why we call him the Explainer in Chief. So what the hell does Password Iterations (PBKDF2) actually do? Love the show, applaud your passion for the security stuff even though it is, for the most part, over my head. Cheers, Andrew.

Steve: We've had a couple of people who've asked the question. I won't spend much time on it because we have covered it before. This PBKDF2 stands for Password Based Key Derivation Function. What it really means is that there is a way to slow hackers down. We've often talked about password hashing. Hashes were designed to be fast. But if a hacker is trying to brute-force by guessing all possible combinations, then we don't want that job to be fast. Yet logging in, like using the password interactively, if it took a second of two for the system to say, ah, okay, that was the right password, well, we wouldn't feel that. We wouldn't mind that.

Leo: But a bad guy might.

Steve: Yes. If it took a second to test every attempted password, then that would dramatically increase our security. So the smart guys at LastPass recognized that, as is the case for all offline password-guessing systems and, for example, WPA that protects our passwords in WiFi, has the same sort of situation where you could capture some traffic in the air and then do an offline brute-force attack, trying to guess the password that was used. And for that reason, the smart people who designed the WPA protocol have the same thing. They've got a 4,096-iteration, PBKDF2 option.

And so the LastPass guys said, hey, they realized they could add that, too, just to further strengthen an already well-designed and strong solution. So it appeared in an upgrade, but they don't turn it on. They recommend, I think, what, 512 iterations, 500. And so...

Leo: Yeah. But you could use more if you wanted to.

Steve: You could. And the only consequence of using more would be a little bit, maybe a delay you could feel on a slower platform.

Leo: You know what's interesting, this ties into your last question in an interesting way because these calculations are done client-side. So if you had a faster computer, you might want to turn those iterations up to slow it down.

Steve: Correct.

Leo: It's kind of like a loop.

Steve: That's a very good point, yeah.

Leo: So, yeah. You do have to go into your LastPass settings to modify that. They won't do it for you.

Steve: Right. Although they are now recommending it.

Leo: Yes, yes, yes. It's recommended.

Steve: So it's a good thing. You turn it on, and it increases your security, and it's painless. I've done it, you've done it...

Leo: Yeah. I even notice.

Steve: Yeah.

Leo: Speaking of which, I finally gave in, and I left the two-step authentication - notice I say "two-step," not "two-factor," because that's what Google calls it - on. I turned it back on. It's kind of a pain because you have to do those one-time passwords on the application, the application-specific passwords. But once you've done that - one thing they did that made it better, they don't expire it every 30 days now.

Steve: Oh, good.

Leo: So that eliminates - because if you expire it every 30 days, that means every day I have to change something.

Steve: Right.

Leo: Something's complaining. Question 4. So do turn that on. I'm not recommending against it. Question 4, James Parsons in Virginia - his Twitter handle is @PolicyEconomy, which is interesting, I wonder if he works for the government - tweeted to SGgrc: Is it possible to implement PIE securely using JavaScript?

Steve: Well, PIE is Pre-Internet Encryption. That's an acronym we coined here on the podcast to stand for this concept of encrypting data before it goes out over the Internet. And it's absolutely possible to implement Pre-Internet Encryption for some applications in JavaScript. And the famous one is LastPass that we were just talking about. It knows what your passwords are in the browser in your system. And it never sends them unencrypted into the cloud over the Internet to LastPass. LastPass keeps only a pre-encrypted blob which, if you then connect up on your Android device or your iPad or a different machine, LastPass will send the blob down to that machine where, again, using JavaScript, it will decrypt the blob there and make them available to that local browser. So that's a perfect, LastPass is a perfect example of Pre-Internet Encryption in JavaScript.

Now, the problem is that JavaScript is deliberately constrained. Because it's running in a browser, and a browser, as we all know, is an untrusted client - I mean, we would like to trust it. Unfortunately, it's going out and reaching over to foreign websites all the time and is a constant source of security problems. So for that reason we did not want a language in every browser that gave it access to our computers. So JavaScript is useful,

for example, in that animation page that I did, for LastPass encrypting things before the browser sends it. But it would not, for example, be able to encrypt files on your system and then send them to the cloud because we don't want JavaScript to have access outside the browser to general system things. That would just be asking for trouble.

Leo: Right. Sam Fineberg, Palo Alto, California, suggests - he's talking about our last episode. We talked about Wired.com writer Mat Honan, senior writer Mat Honan, who was hacked last week. He says Mat might have already been doomed: I was thinking about last week's episode. My conclusion is that Mat's digital life was doomed. He may have been lucky to be hacked since there is in fact a way to get his data back. And now he better understands what was always at risk. What if his hard drive had died beyond SpinRite's ability to repair? Someone stole his MacBook? Or if he'd dropped it? Without a backup, those pictures were already toast. You know, parenthetically, that's what Alex Lindsay says, is one copy of anything is no copies at all.

Another thing that troubles me is that his drive was recoverable. If I remotely wipe a drive, I want it gone. Period. I don't want it to be possible to recover the data with a four-digit PIN. I'll explain what happened. I want it securely wiped, or encrypted with unbreakable encryption. Any device I take outside of the confines of my house I consider vulnerable, especially a laptop I run though airport security or leave in a hotel room. That's why I encrypt my laptop drive, and not with an easily brute-forced pin. However, the flipside of that is that it means that I would rather lose my data than expose it. Period. It seems unfathomable that anyone would only keep a single copy of precious data, and even worse to have the only copy on a laptop or mobile device.

As I said, Mat's data was doomed from the moment he didn't back it up. He should consider himself lucky something worse didn't happen. Sincerely, Sam. I guess that's true.

Steve: Well, okay. I have a couple things, and I know you have a couple things you wanted to comment to.

Leo: I'll explain what the hard drive does.

Steve: One of the things that I have noticed is that, over time, we've become increasingly dependent on our gadgets. And that our appreciation of the "what if something went wrong" somehow for some reason, a quirk of human psychology or maybe there's a sense of, well, it was fine yesterday, so it'll be fine today and probably fine tomorrow. My favorite example is back when we used to talk about online banking, and the family would have their communal family PC. Now everybody's got their own. But at one point they were still expensive, and so everyone was sharing them. And there was no security. I mean, there was no notion of security. The kids would bring their friends over after school, and they'd download strange software on it or get stuff off the Internet,, or they were using filesharing stuff that was a big source of problems.

And then Mom and Dad decide that they want to - their bank is saying, hey, you could do your banking online. And so into this disaster of security they begin to add new functions and features that really do need security. But because this sort of happens a little bit at a time, just sort of drip, drip, drip, at no point is there anything that says to them, wow,

you need to really start taking this seriously because, have you noticed, you've now got your accounting, now Quicken and QuickBooks is on your system, and your banking is on, and your husband or your family's investment portfolio is there, blah blah blah blah.

So one of the things that happens, I think, is that we just sort of - this creeps up on us. You can imagine, for example, that Mat may have transferred the first photos of his newborn from his camera to this laptop. And at that point he didn't have a huge investment in that particular content of his laptop. But it slowly grew over time, and it never occurred to him to say, okay, what if this all disappeared tomorrow? What if it was just gone? And we've talked around this in various ways over the years, like what would happen - again, no one wants to think of what if I didn't wake up in the morning, but how would my family deal with all of the passwords that I have in my head that they don't know, that are vital? So this notion of "what if" planning is important.

But I think it's very easy for us not to appreciate how important it is because of sort of over the time that we can look back and remember, we've become far more dependent upon these things. The security of them has become far more important as more and more of our life has become digital. As Mat put it, his digital life, he suddenly realized how important that was because he had digital photographs that used to be printed out on film and on paper, now they're on a hard disk. And as Sam says, you could turn the computer on, and it says we don't have a hard disk here anymore. So I think I liked this because I understood what Sam meant, and I know that Mat is probably now taking measures to make sure that he's backed up.

Leo: Oh, yes. I talked to Mat. He's definitely - cue the Carbonite ad. He's definitely doing that. And I don't know, I have not talked to him since he got hooked up with DriveSavers. I asked the DriveSavers people to help him out. So I don't know.

So I'm a little curious, and Apple understandably is a little cryptic about what happens in the remote wipe. One would presume that what happens is it erases data, then overwrites it, and then erases it and overwrites it. And that's what I assumed happened. And in fact, when I talked to Mat, he said Apple said the overwrite portion had not completed, so they believed that the data would be recoverable. Now, there is a four-digit passcode you have to enter, but they were able to brute-force that, that's not so difficult, and get in and unlock it for him, and then he has to bring it in.

Steve: Yeah, something...

Leo: Now, it gets more complicated. So the Mac Observer had the same questions. And they did, this week, they did something kind of interesting, a couple of days ago. They wiped a drive, and they wanted to see, can it be recovered? And in fact they were able to recover the data. It took a long time, but they were able to use an application that is designed to recover erased data, called Data Rescue 3, and they were able to recover it.

Now, they point out that, if it's on a solid-state drive, which Mat - it was a MacBook Air, Mat was - that the TRIM feature built into the operating system might in fact make data recovery impossible, or at least spotty, because TRIM overwrites data as part of the "trim" process. And this is probably the takeaway: If FileVault had been turned on, which is the whole-disk encryption built into OS X, then it would have

been unrecoverable. The wipe would have been sufficient because, of course, with the wipe you lose the FileVault encryption key.

Steve: Okay. This is all sounding very sketchy, Leo. I mean, it sounds like they just did a delete, like they deleted files, and then you can undelete them.

Leo: It does sound like that, although I think they do more than a delete. I think they delete the partition tables and whatever the equivalent in HFS is of a file allocation table because they had to use the deep scan, which is a sector-by-sector recovery process.

Steve: So this just sounds like...

Leo: But the data was recovered.

Steve: Well, I'm glad for that. But I would say to our listeners, we know about TrueCrypt. TrueCrypt is real encryption. TrueCrypt is - anybody who's concerned, as Sam indicated he was that anybody else would have access to his data, TrueCrypt is a beautiful piece of work that runs under Windows and on the Mac platform. It has the overhead of you need to enter a password in order to boot your system. But, boy, it solves the problem in the right way. This sounds really hokey. I don't know what Apple's...

Leo: Now, but think about it. Maybe this was - I don't bet it's a conscious decision because what you're doing is you're wiping it figuring the bad guy is going to, even if he can get through the four-digit passkey, is going to say, well, the hard drive's wiped, and just sell it.

Steve: Okay. But the notion of a four-digit PIN is at complete odds with any kind of, like, wiping. What does that word mean? No, we're not getting the whole story. This is all funky-sounding. Well, we're wiping it, but we have a four-digit PIN. What, so you can unwipe it?

Leo: Yeah.

Steve: Okay, then it wasn't wiped.

Leo: It wasn't wiped. It was deleted, but not wiped.

Steve: So, okay, so what's the four-digit PIN do? No, there's something - I don't know.

Leo: Well, the four-digit PIN keeps people from using the computer. So there's two points here.

Steve: Okay.

Leo: The bad guy can't even use the computer. And then, should he get in, he'll find a...

Steve: The PIN does not unwipe the drive.

Leo: Oh, no, no, no. You have to use...

Steve: The PIN locks the computer so it's no longer useful to them.

Leo: There are two options, if you've lost your Mac, on this. One is to put a PIN on it; one is to put a PIN and wipe the drive.

Steve: Got it, got it.

Leo: So that's fine.

Steve: Okay, yes. Now I...

Leo: And I think the message, apparently, is, if you're really concerned that some bad guy will get your drive and be smart enough to do a data rescue on it, then you should use whole-disk encryption, which will provide an additional layer of protection. But Apple, of course, is not going to tell anybody what they're doing because they don't want anybody to know.

Steve: Yeah.

Leo: Quite understandably. I mean, you know, security through obscurity. Wil Agnew, Connecticut, takes issue with our casual attitude regarding the Mat Honan hacking:

First - I'm going to read it in a Connecticut accent - let me say I have immense respect for what you do. I am a fan of you and your work. However, I am disturbed by your casual attitude toward the legality of the actions these young people seemed to ignore. Not being a lawyer - I'm not going to continue this way, obviously. Not being a lawyer...

Steve: I think this would discourage Wil from writing to us again, so...

Leo: Yeah, well, we're just - we're having a little fun with you, sir. Not being a lawyer, I may be wrong, but what was described in this hack seems to be a clear violation of Title 18 of the U.S. code, with penalties of 20 years in prison and millions of dollars in fines. I heard no mention of the seriousness of these hackers' actions under the law, and I seemed to detect an acceptance of this sort of activity, especially regarding the "culture" mentioned near the end of your show.

Being an IT security person myself, it is clear to me that the first shots of the Cyberwar have been fired - oh, please. I'm sorry. Maybe I'll go back to that accent, "...and the U.S. needs young people who have this sort of interest to become the next generation of cyberwarriors rather than waste those skills and interest in technology trying to hack and gain access to a cool Twitter account handle. Perhaps some constructive feedback by - I did, but, well, all right, I'll finish the letter, then I'll respond.

Perhaps some constructive feedback by people these kids respect could help with this. I think we mentioned this. I understand the intent of your podcast was to illustrate an example of all the types of issues you regularly discuss being applied. I'm just disappointed that the tone of the program never really touched upon the severe penalties this activity can bring to people who do this type of activity. As a consequence I worry that the Wired article, and your podcast, may actually be encouraging these people to continue doing these activities, in a sense giving credibility to them. What do you think?

Can I just say, first of all, on the podcast, I said it was a federal offense, a felony. I did attempt to get Kevin Mitnick, who has served time in jail, about three years - nobody gets 20 years, certainly not for a hack like this - but who did serve three years for a much more severe hack, to talk to them, and he quite wisely said, you know, maybe it'd be better not to talk to people actively engaged in criminal activities, considering my history. So he declined. But I did say to these guys, behind the scenes, this is a felony. Nobody needs to press charges. The feds do not take well to this. And I think I even mentioned on the podcast, in this post-9/11 world, these things are - it's dumb to do this.

Steve: Nobody is laughing about this, yes.

Leo: I don't think we mocked it. I don't think we encouraged it. It is not our job to enforce the laws of the United States. There are people duly nominated to do that. But I think we were very clear about - I don't...

Steve: Yeah. I chose this, obviously I didn't have to, because I thought it was worth mentioning that there was no benefit that these young kids were deriving from this, yet they really were taking a substantial risk. I mean, as you said, Leo, in this post-9/11 world, and as we really seem to have this notion of cyberwarfare, and cyber is going to be the next battlefield, mindboggling as that is, still, to me, there's a serious consequence to this sort of hacking.

Leo: Yes. But nobody denies that.

Steve: It's one thing for the Russian Mafia to be doing this, and entirely something else for a bored 19-year-old kid to be sort of wantonly committing felonies. And so I don't want to see anyone locked up in jail for just screwing around. And so I guess I see a point to what Wil was saying, which is my sense is these kids probably, as part of their gang or their group or their culture, don't appreciate, I mean, the fact that he was able to ask Mat to agree not to press charges demonstrates a lack of appreciation for the severity, the legal consequences of what he was doing. And, wow, I mean, he was saying that someone else made the calls to Amazon and was impersonating people. But this - it was a lot to do.

Leo: But I'm pretty sure I read this on Security Now!, our conversation in which he said, "Oh, Mat's not pressing charges." And I said that he doesn't have to. I said, "I'm no lawyer, but I think you can. It's a felony, wire fraud." He said, "Exploiting methods for the better cause, seems unlikely you'd get in trouble." I said, "Well, the feds don't have a sense of humor."

Steve: Yes.

Leo: I said, "You want me to get Kevin Mitnick to tell you what's going on here?" But the point is, look, we talk about also bank hacks on this show frequently. We are not nannies. It is not our job to say parenthetically, oh, and by the way, it is illegal to hack a bank, and you could go to jail, so don't. That's not our job, I'm sorry. I'll leave that to you, Mr. Agnew. You can go around to the teenagers in your neighborhood and remind them that the U.S. needs their hacking skills. But it's not my job. I'm sorry.

Question 7, Dan in South Carolina. He's wondering about achieving security in the face of extreme access limitations: Steve, I'm looking for a better business practice for personal account management. I've wanted to go to a LastPass-like solution for a very long time. I don't think it will work for me. I'm in the military, and I work long hours on systems with tight IT controls. I am often not allowed to have my iPhone with me when I work. I also travel regularly and find myself on completely new systems. These are all well-secured systems, but often by different IT departments inside the DOD, with different IT policies. This guy's a heavy-duty hitter here.

Steve: Yeah.

Leo: The nature of my work necessitates my being able to conduct personal business from work computers during lunch or off-duty hours. I'm surprised he can. Six months away from home is a long time not to check your bank account. Ah, now I understand why.

Steve: Yeah.

Leo: Currently, my password management system is a password-protected Excel spreadsheet - hmmm - mixed case, numbers, and special characters that I email to myself. Oh, god. This way, if I find myself sitting down at a new computer in a new place, I can get oriented and get things done. You bet. Anyway, these are not random computers in a hotel lobby. I worry my Excel spreadsheet method may not be sufficient to protect me. Any advice? Regards, Dan.

Steve: So I thought about this a bit, and I kept coming back to the notion of a pad of paper, essentially.

Leo: Yeah, one-time passes or something, yeah.

Steve: Well, actually, the Off The Grid system that I designed is exactly for this kind of purpose. Now, I need to also mention that I never put the links up publicly after I described it and discussed it. A couple people noted that I hadn't handled the case of repeating characters in the domain name during the initial sort of warm-up phase, and I got pulled off of that to work on something else before I got the pages finally all buffed and made public. But I haven't forgotten it, and I am going to - it's on my short list of things I want to get back to and get finished.

But that kind of system, something that is, I mean, if nothing else, maybe just paper in your wallet or something, I mean, frankly, the Off The Grid system is perfect for this because it's low tech. I'm a little worried, and I could hear in your tone of voice, Leo, that the notion of a password-protected Excel spreadsheet...

Leo: Seems an oxymoron. In the past, it is true that Office password protection has been easily hacked. But they did go to a much more secure system. I think they're using a strong encryption technology now. But I still wouldn't rely on it. And you really are, if you're emailing it, you're basically making it public.

Steve: Yeah. There's a cool little utility we've talked about a couple times called AxCrypt. AxCrypt is just a very simple AES-256, standalone, very solid little encryption utility which is freely available on the Internet. And I would trust it much more than I would trust Microsoft's password on their Office documents. You're right, Leo, that there's all kinds of "remove the password from your Excel spreadsheet" turnkey third-party utilities floating around. Maybe they no longer work. It's not something I've looked at. But running it through my own encryption I would feel - and then you don't have to really use a spreadsheet. You just use any file that's convenient.

Leo: He could probably use - I'm sure LastPass would work on a USB key. I don't know if he's allowed to bring, and probably isn't in a secure environment...

Steve: That's what I was wondering, too, yes.

Leo: And certainly KeePass, which is an open source password safe that's quite

good, does work on USB keys. You can bring the password database around with you. You could even email it because it's fully encrypted. But I bet you, given what he's just described, he's not allowed to bring USB keys in, either.

Steve: Yeah, that's why a paper-based system looked like it was the perfect solution for him.

Leo: Yeah, in your wallet.

Steve: Yeah.

Leo: So you don't have those grids online anymore?

Steve: No, actually, it's all online. And I think Google knows where it is. But I have never linked it.

Leo: It's not on your menus.

Steve: I haven't taken them officially public yet, yes. Off The Grid.

Leo: So Google "GRC Off The Grid." Let me just see. Oh, yeah. No. 1 hit. But are you saying it's imperfect?

Steve: No. It works. Some people noted that I had - there was a case that I hadn't handled yet, and so I just haven't gotten around to updating the documentation. And I just want to read it all again. I feel like I just hadn't quite finished it. And so I didn't take it public yet.

Leo: But he could currently print this grid out.

Steve: Absolutely.

Leo: And create an algorithm for generating a password that only he knows. And then he'd have it. And these are all unique. Each time you load this page you get a unique grid.

Steve: Yup. And then there's also the Grid Generator page a little bit further in.

Leo: Ah, okay. So all the details are here. That's probably the best thing. And that -

they can't stop him from bringing his wallet in.

Steve: No, or just a sheet of paper.

Leo: Or a sheet of paper. Philip Boccia - I bet you it's Boccia - in New Hyde Park, New York, asks about the Internet WHOIS system: Steve, in Episode 364 you and Leo talked about the problem of no privacy with WHOIS lookups. Isn't one of the reasons this information is publicly available is to thwart the creation of rogue websites? If you make the WHOIS information private, wouldn't that essentially be protecting those bad guys who set up illegal and phishing sites? We should explain. Go ahead.

Steve: So WHOIS, yeah, back in the beginning, the dawn of the Internet, there was an interest in creating a sort of a translation between the virtual world and the physical world. And there was actually one physical server that DARPA was running that was the WHOIS server. And anybody could send it a query. You could say, who owns this block of IPs, who owns this domain name, and it was like a big lookup database that was freely accessible to anyone. And it's never gone away. You used to even be able to give it wildcards, like you could say, tell me all the Gibsons you have, and it would just dump them out. Clearly those days are long since past.

Today what we have is still this notion of the WHOIS system being used to map domain names to real-world entities. The problem is that it's up to the registrars to fill out this information. And the registrars are not in the business, especially if you're not asking for SSL certificates, they're not really in the business of verifying people's identities or performing any great amount of follow-up to make sure that what you've told them is valid. They're happy to take your money and essentially give you control of a domain name that they stick into the Internet's DNS system, and they feel that their job is done.

I'm unimpressed by several things. First of all, it is, unfortunately, it's been used as a source of spam. That is, spammers have figured out that there are email addresses of what used to at one point have been important people who were administrative leads for their domains. And so it was one way that spam found its way to people is that they would put real email addresses in their WHOIS records. By law, technically, you're supposed to have real contact information there. I mean, that's what it's for. So what annoys me most, frankly, is that someone like Network Solutions charges money per year, it's like $9 or $10 per year per domain name to obscure your WHOIS lookups.

Now, to answer Philip's question, Network Solutions knows who's behind or theoretically behind the WHOIS information that they manage. So certainly, even if it was obscured or encrypted or protected, they could be served with some government documents compelling them to tell them what information they have about somebody who owns a domain, and I'm sure they would turn it over. So unfortunately it's just become useless, pretty much. It's full of junk. It's not accurate. It only contains whatever information someone chose to give the registrar. What do you think, Leo?

Leo: I think Steve Gibson is not impressed by WHOIS. And by the way, the current WHOIS privacy system, if it were a bad guy, they'd just go to the server, and they'd say, well, who is it? I mean, it's not hidden from the hosting company. It's just from

the casual WHOIS searcher.

Steve: Right.

Leo: So I don't think that that's a problem. And I think privacy's important because otherwise my personal address is posted in there.

Steve: Yeah.

Leo: Colin in Cleveland says, what about best practices for security questions? Which are insecure to begin with.

Steve: Yeah.

Leo: Listener for about a year here, love the show. Last week's show following the details of Mat Honan's hacking has brought a question back to mind: What's your approach for handling security questions? I've done what I can to avoid questions that could be answered by a Google search, but I've never been able to find a reliable method for having secure answers that are easy to remember. I'm already using LastPass for my passwords, so anything I can do to improve these questions will significantly boost my online security. What are your thoughts? Yeah, I wish we could just turn these off. Wouldn't that be nice.

Steve: So the good news is Colin is a LastPass user. LastPass has a Secure Notes option, and security questions are not something that you generally need to have on the tip of your fingers or that you're being bugged about all the time. It's sort of a fallback password recovery option. On the other hand, one would hope, if you're using LastPass, you've got enough technology at your beck and call that you're not needing to use password recovery.

But my point is that you could just generate things that look like passwords, I mean, like nonsense strings, to answer security questions, but you could never remember those. So LastPass has this cool little secure notepad where you're able to create notes and name them. So you would name the notes after the domains that you have answered these questions for and store your answers there so that LastPass will keep them secure. They're encrypted. You don't need to remember them. And if you ever need them again, you can ask LastPass what it was that you used for your first girlfriend's name or your favorite teacher from high school or whatever.

I ran across somebody, I think, of the email I saw, or maybe somebody tweeted - I think it's too long to have been a tweet. Someone was saying, though, that they were in an online conversation with some people that they connected with, who were sort of chatting back and forth. And after a while, one of them sort of casually said, "So, out of curiosity, if you had to, who would you say was your first childhood friend?"

Leo: [Laughing] Just askin'.

Steve: Just, you know, we're just...

Leo: My first childhood friend was Sally. Who was yours?

Steve: Uh-huh.

Leo: Wow. That's just...

Steve: I'm not kidding. And this person, whoever it was, it must have been a follow-up that I saw from our Mat episode last week. And he said, sure enough, I went back to the system that we were using, and that was the security question which this particular service had. And so they were just - they were trying to socially engineer him. Just kinda curious who would you...

Leo: Do you remember your first pet's name?

Steve: You know that six degrees of freedom? Maybe we happen to know the same...

Leo: Who was your favorite schoolteacher? Just off the top of your - you know, the truth is, I just realized I said they should turn them off. It's easy. If you don't want them, don't use them. Just put nonsense in there and don't use it. But understand that if you ever have to - now the onus is on you not to lose your password.

Steve: Correct.

Leo: I guess there are some places like your bank and stuff where you - actually, when I call my credit card, they will ask what's the name of your first teacher. It's so - that's the problem is that they - no one should rely on these. That's the real problem.

Steve: Yeah. The one thing that I've seen a couple times are sites that allow you to provide the security question...

Leo: Yeah, write your own.

Steve: And the matching answer. So at least there, they're not all the same. Apple, I guess, has a set of really dumb things that are like, okay, well...

Leo: The concept is flawed. The whole point of a security question is what's something you can easily remember and easily retrieve from your mind if you forget your password, you nitwit?

Steve: Yeah, and we've talked about the nature of multifactor authentication is multiple factors. One of the things that Mat's story showed us last week was that Apple said, oh, you don't remember your security questions. Okay, well, in that case...

Leo: No problem. You're even stupider than we thought, but that's okay.

Steve: Either honor them or don't.

Leo: Yeah, or don't use them. And as you said last week, then let the onus be on the user. We can't help you. You forgot your password, sorry. Maybe you'd better set up a new account because we don't know. And really that's another problem. I don't know if you got emails about this. But should customer service reps have access to that kind of information?

Steve: Yeah.

Leo: Because that's a hole. That's a flaw. And I think the fact is that stuff's not going away. I don't know what we can do about it. I mean, it's going to happen.

Steve: No. 10.

Leo: This is our last question. Security Now! listener Paul in Ottawa, Ontario, Canada, shares his observation about Mat Honan's Very Bad Weekend as it relates to compromised user databases: Steve, as you know, recently several user information databases at Sony, Yahoo!, LinkedIn and others were compromised. With so little information required to reset someone's password at some companies, wouldn't a compromised database be easy to use to access someone's account, or everyone's account, even if their password is hashed?

A lot of the other information, like email addresses, billing addresses, credit card numbers could not be hashed. It's needed by the company; and that information possibly, and most probably, is not encrypted, or at least could be decrypted. If someone were to reset their password after learning of a compromised database, all the information is still there. And that really was, I think, the lesson of this hack, to simply call and get a password reset. A company's password reset policy could be the weakest link in the chain. What do you think? Paul from Ottawa. I think that was really the point we were making is that you can have LastPass, strong passwords. And in a case like this, if the company doesn't have good security policies, you're screwed.

Steve: Yes. And I've been thinking about it in the week since, and I'm in the process of

putting together sort of a commentary that I will share with our listeners next week.

Leo: Excellent. Well, there you go.

Steve: And I agree with Paul. It is clearly the case that the password reset policy is the weakest link. We're to the point where we need to change something, and I have a suggestion. And we'll share that next week.

Leo: What a tease you are. Steve Gibson, the Explainer in Chief. You'll find him at GRC.com, the Gibson Research Corporation. That's where he hangs his hat and SpinRite, the world's finest hard drive and maintenance utility. You can also find lots of freebies there. In fact, even though Off The Grid's not on the menu, it's off the grid, you can Google it, and it's there, too, and a lot of other very useful stuff. He tweets on the Twitter at @SGgrc. There's also @SGpad and @SGvlc, but don't tell Dr. Mom what that means 'cause she's sitting right over here and giving me the stink-eye right now. But it's okay because I had some chocolate, and she feels better. Isn't that interesting. I had chocolate, but you feel better.

Steve: You have chocolate, and she feels better.

Leo: Yeah, it's interesting. Steve, it's always fun. I really appreciate it. We will talk next week about security, privacy, science fiction, coffee, and anything else that's on your mind.

Steve: Fantastic. I look forward to it.

Leo: Take care, Steve. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC on TWiT.tv. Do watch live. We'd love it if you do. I interact with the chatroom and so forth. And of course after the fact we make on-demand versions available. Steve's got a great 16Kb audio file for the bandwidth-impaired, also the smallest version of this show, which is a text transcription. That's at GRC.com. We have the audio and video, the fat stuff, at TWiT.tv or wherever finer podcasts are offered for download. Thank you, Steve.

Steve: Thanks, Leo.

Leo: See you next time on Security Now!.