Cryptographic HMACs

Description: Steve and Leo discuss the role, importance and operation of cryptographically-keyed message digest algorithms and their use to securely authenticate messages: Hashed Messages Authentication Codes.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-185.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-185-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 185 for February 26, 2009: HMAC. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that talks about all this security stuff, which lately has been quite a bit of security stuff. Our host, Steve Gibson, is here. Hi, Steve.

Steve Gibson: Hey, Leo. We have another week crammed with news. This one will be like your original concept of Security Now!, which was less the sort of the security school that I've had for the last - or the security school that I've been holding for the last three and a half years, and more security news because we've got a whole bunch of interesting stuff to talk about that has happened in the last week, actually.

Leo: And I don't mean to downplay security school because that's been really important to us understanding the security news, you know, having the ground of being to understand what's going on.

Steve: Oh, and I love the school. And in fact one of the consistent things that I read in people's feedback - at GRC.com/feedback - is that people say "Every single week I learn something." I mean, even IT professionals, security professionals, everybody says that you'll say something, you and Leo talking, or the content will have something I didn't

know. And so that's really valuable.

Leo: Yeah. I think this is, for me, anyway, it's an education. It's college-level education in security. But because it changes, I mean, there's some stuff that's eternal. But there's also stuff that's changing all the time, so.

Steve: Well, and the show, the title of this one would be HMACs. We're going to talk about Hashed Message Authentication Codes. That's the last bit of technology, sort of fundamental technology that we need in order to understand, essentially to have our toolkit complete so we can then really get a grip on how SSL works. And I was thinking of doing that week after next. But I'm going to push it back another two weeks because of something that has happened this week that I want to give - that we'll talk about briefly, but I want to get more coverage to, I want to give it its own podcast - about the new flexibility that Microsoft has added to the Windows AutoRun, where you can, like, individually, granularly control what is allowed to run and what isn't.

Leo: Oh. All right, Steve. Let's get to the voluminous security news today. You sent me notes. It's two pages of stuff.

Steve: Well, yeah. Mostly just little - some things I wanted to quote exactly from the reference source. And mostly things where I just needed to have the details in front of me. So I figured, you know, mostly I wanted to give you some sense for where we were as we moved through this so that you're not saying are we done yet, are we done yet?

Leo: Well, and it's very handy because I could put it in the wiki, which I'm doing right now. So that if you go to Wiki.TWiT.tv, you'll get a complete list of, you know, detailed list of all the notes. And I'm sure you put those in your show notes at GRC.com, as well.

Steve: So we've got security news. We've got errata. And my always interesting, that I try to find, fun SpinRite testimonial.

Well, first of all, for Windows users we had an interesting, out-of-band update. Microsoft called it a "non-security security update." And in fact in their own FAQ they even asked themselves the question, well, isn't that a contradiction in terms, to have a non-security security update? Anyway, what they've done is, it turns out that some time ago - and I looked around, I've got the prior knowledge base number, but it's been updated since so I can't tell when the original publication date was. But some time in the past Microsoft realized that AutoRun disabling - which is standard practice. We've talked about it many times. It's one of sort of the first things a guru does when they're wanting to bolt down a system, is they turn off AutoRun. That's the feature of Windows where, if you put, for example, most commonly a CD in the CD drive, if in the root of the CD directory there's an autorun.inf file, Windows will look in there for instructions and execute code, typically also contained on the CD.

That's been a problem. For example, this was the way that Sony installed their rootkit into Windows machines, the infamous Sony rootkit that we talked about years ago. So in general, savvy Windows users dislike the idea that their system is going to run something without their explicit intent for that to happen. The flipside is that neophytes,

you know, people who are not all security conscious and are maybe more trusting, it's advantageous for them not to have to go dig around in the contents of a CD in order to find what it is it needs to run. I mean, I'm often opening up the autorun.inf file manually, looking at it, seeing what it is that it wants to run, and then going and running it manually because I've told my system don't do this for me. But that's a lot to ask typical users to do. So again, it's a feature that Windows has. Many people disable it.

Well, it turns out that until this prior update was - security patch was added, it wasn't working right. And so quoting from Microsoft's new announcement, they said, "Microsoft is announcing the availability of an update that corrects a functionality feature that can help customers in keeping their systems protected. The update corrects an issue that prevents the NoDriveTypeAutoRun registry key from functioning as expected." And a little bit later it says, "The updates offered in this article correctly disable the AutoRun features. These features were not correctly disabled if you followed previously published advice. The updates that are offered in this article have been distributed to the following systems through Windows Automatic Update," and blah blah blah.

So here's the deal. This has now been re-released as a Windows Update fix, which it wasn't before. Which meant that, unless users were aware of this non-Automatic Update, sort of published on the side update, their AutoRun had some holes in it. And from poking around, it appears to have been network shares which, that is, attached network drives could have AutoRun functionality and would have, even if you thought you had turned that off.

So I want to talk about this in detail in two weeks because it turns out what they've done is they've really sort of bumped up the granularity of this so that users will be able to automatically disable AutoRun on drives of unknown type, on removable drives, on network drives, on CD-ROMs, on RAM disks, and all drives individually, which I think is very cool. For example, I would definitely want to, in my case, and I think many users' cases, they want to just disable it across the board. But, for example, we've talked about the removable thumb drives, where you stick them in the computer, and they contain some code that runs when you insert them. I really dislike that. And so, for example, it would be possible to disable that while leaving CD AutoRun enabled by using this new granularity feature.

Leo: Does that even count for those U3 drives, as well?

Steve: Yes, exactly.

Leo: That's the ones that are most scary, of course.

Steve: Yeah, now, they look like two different - they have two different profiles. They have a CD profile, which is the way they get the AutoRun to function. And then they also have a regular mass storage profile. So again, you would be able to take control and make this thing work just exactly the way you want to. They've added some new registry keys. And in fact, with this update there's another key that they've created in the registry which is called HonorAutorunSetting, which is set to one by default.

And the problem is, Microsoft realized that maybe people had gotten used to the buggy way that it was working, and there ought to be a way to disable the fixed version and have it fall back to the buggy version so that the automatic update which everyone is

probably going to get now, and in fact received out of cycle. They sent it yesterday, on Tuesday, two days before the date of this podcast, the day before we're recording this today on Wednesday. So Microsoft is concerned, since they moved this thing from you've got to go looking on Microsoft's site to get it, to auto update status, where everyone's going to get it. Now what's going to happen is - and it does require a reset because they changed the shell32.dll, which is a fundamental intrinsic core of Windows, so you'll definitely be needing to restart your machine. So since it's going to be an automatic update and fix the problem, suddenly behavior will change, and people might have been dependent upon the old behavior. So Microsoft has added a feature that allows you to disable the improved, fixed behavior.

Anyway, in two weeks we're going to go over this in detail. All people have to worry or have to think about now is that, when they update themselves, Windows will be functioning correctly, that is, the way they probably intended Windows to be running anyway. And so what we'll be able to do is we're going to explore in two weeks how you can back off from that blanket AutoRun, if you want to allow some specific AutoRun behaviors and essentially, with a great deal of granularity, tune it so that it works exactly the way you want it to.

Leo: That's really nice. That's a nice update, to be able to say this, but not this.

Steve: Yeah, yeah.

Leo: I mean, I know you would turn it all off. But it's handy if you put an audio CD in, and it starts to play. I don't think that's necessarily a security risk.

Steve: Or, for example, if you yourself, like, want to use the U3 drives...

Leo: Yes, right. I mean, some people want them; right.

Steve: Or you have an - you're a little bit more of a guru, and when you plug a USB in that has a program, you can create an autorun.inf file. So you want to say, look, I don't want CDs to run by themselves, but I do want USB devices. So it'll allow people to tune it so that it works just the way they want to.

Leo: Frederique, our office manager, has - and I encouraged her to do this. She has a U3 drive with a password manager on it. She uses Roboform. They make a version for U3 - actually they make even a version, I think, that even if you don't have U3 drive, that will autorun. So she keeps her passwords. They're secure. This is why I encourage her to do this. She plugs it into any machine that she's going to be using. Now she has access to her passwords. And that's a very nice, safe system for her. That's something that's good to have.

Steve: Yup, exactly. And so you would not want to disable that on a system where you would otherwise want to bolt some other parts of it down.

Leo: Precisely, yeah.

Steve: Yeah. So anyway, two weeks from now we'll go into that in painstaking detail. I wanted to mention that some recent changes - this is sort of an obscure one, but it no doubt has an intersection with some of our listeners. Some recent changes to v7 of my favorite UNIX, which is the FreeBSD version of UNIX, caused a new security problem in the telnet daemon. Telnet is the remote console protocol. I would be surprised if any Security Now! listeners had their telnet server wide open and exposed to the Internet. That is, it's a common attack vector. Telnet runs on port 23 by default. And if nothing else you'd want to move it to a different port because port 23 scans have been historically quite common, and a brute-force attack on the login credentials is something that could be happening in the background with you never knowing it.

Anyway, what happened is, even without - as I understand it, even without logging in, there are some telnet-level commands that allow you to configure environment variables such as baud rate and other telnet protocol aspects, which you can imagine you need to do, like, right off the bat. And these changes in v7 of FreeBSD introduced, unfortunately, a remote code execution exploit which would allow somebody who just found your telnet daemon exposed to cause code that they had somehow gotten onto the server through different means to be executed. So, for example, if you also had an FTP file upload capability that allowed you to accept files, but you had been safe about not allowing them to be executed, well, this would allow that to be bypassed.

So the patches are available. I just wanted to notify anybody, any of our listeners using FreeBSD v7, if by any chance they use telnet in a way that has any Internet-facing connectivity, you really want to fix that immediately because that's potentially bad. And scans for telnet are common and easily implemented.

In other news, we talked two weeks ago - which was our podcast after the Patch Tuesday. We've had another instance of what's called "Exploit Wednesday." Patch Tuesday, of course, is Microsoft's patching Tuesday, where they issue all their updates, and we had some big ones. We had, you may remember, a critical update to IE7. Well, less than a week after that was released the patch was reverse-engineered, as now unfortunately often happens, and an exploit has been developed which is in the wild.

The current implementation of this is email containing a Word DOC file which itself contains an embedded ActiveX object. So when you receive the email, if you attempt to open the Word document - and there will be some social engineering thing which will induce a nave person, a trusting person to open the attached document. After all, aren't documents safe? Eh, no, because they can have an embedded ActiveX object.

This one causes IE to visit a malicious site which runs a script - and there's my favorite "S" word. And the vulnerability in IE7 is leveraged by the script to cause malware to be downloaded. It installs a backdoor in the system which is persistent, and also sends a load of confidential information from your machine off to a server in China. So you don't want that to happen. I just want - I know that everyone will have updated their Windows by now. But the problem, of course, is that corporations often deliberately delay patching their systems by policy because they want to vet them because there's been a history of Microsoft patches messing things up.

Leo: Right, screwing everything up, yeah.

Steve: Yeah. So this is bad. And all of that, the Downadup/Conficker worm stuff, I mean, which is becoming a huge problem, this is a problem which was - this was an exploit that's been patched in Windows last October, October of '08. And here we are toward the end of February '09. And that demonstrates that there is a huge gap between the availability of a patch and the time that these systems really get themselves updated. Again, I know it's not our listeners. But it's certainly, somehow, it's people who don't have Windows Update enabled or are not restarting their machines or whatever. One way or another, these problems are persisting. And the fact that this Downadup worm is causing such problems for something that was fixed in October demonstrates that there is this window of opportunity.

So for that reason, patches like the one that was issued on Patch Tuesday for IE7 are being reverse-engineered and exploited. And these guys know they've got a big window of opportunity during which they're going to be able to install this junk on people's machines. Now, the AV guys are on top of it. And the antivirus updates will be catching this, too, even though Windows itself should be patched to avoid the vulnerability.

Next bit of news is rather interesting. There are two distressing bills, one in the Senate, S.436...

Leo: I know where you're going. Oh, boy.

Steve: And the other is a House Resolution, HR 1076. And this is one of those where somebody really struggled to come up with an acronym. They wanted to call it the Internet SAFETY Act, so SAFETY is an acronym for...

Leo: This is the worst acronym ever.

Steve: I know. Stopping Adults Facilitating Exploitation of Today's Youth, S-A-F-E-T-Y. Stopping Adults Facilitating the Exploitation of Today's Youth. Well, okay. And the thing that annoys me is they're pulling the child porn card. That's their whole justification for this is, oh, you know, we've got to protect our children. Well, we all agree we've got to protect our children. But get a load of how this thing is written. Quoting from the bill, and these are identical legislation in both houses because they both have to pass it, then they go to conference, and then the President signs it - if he's asleep. Okay.

"A provider of an electronic communication service or remote computing service shall retain for a period of at least two years all records" - okay, wait. Before I go any further, let me say this applies to home - because I want people to listen to this wording. This applies to everyone hearing this. Anybody with a WiFi access point that uses DHCP that distributes IPs automatically - businesses, homes, hotspots, I mean, it's unbelievably sweeping - this affects. That is, all of us end-users must do the following:

"A provider of an electronic communication service or remote computing service shall retain for a period of at least two years all records or other information pertaining to the identity of a user of a temporarily assigned network address the service assigns to that user. Definition of 'electronic communication service' from the prior sentence is 'any service which provides to users thereof the ability to send or receive wire or electronic communications.' The U.S. Justice Department's position is that any service 'that provides others with means of communicating electronically' qualifies."

Leo: Wow.

Steve: So that - so literally this proposed law gets not just AT&T, Comcast, Verizon, and so forth, wired ISPs. They're all using DHCP to give us our IP addresses at home. But public access points, including password-protected ones. Individuals, small businesses, large corporations, libraries, schools, universities, and government agencies. If this law were to pass in its current form - and it's difficult for me to believe that it could because there's got to be some back push against this.

Leo: Well, and this is not the first time they've tried something like this. This has been going on since 2006.

Steve: True, there was about four years ago it was brought up. And then so now here it is again. It's come back again. And essentially we've talked about the need, for example, for ISPs to log. But the idea of requiring a chain of login, that is, not only the ISP logging that you have been given an IP, but this legislation as written requires for you to log that you have given a NAT router IP to somebody who's using your WiFi connection. I mean, it's just, I mean, interestingly, there's a simple way around this. And that is not to use DHCP, to assign IPs statically within - even within your WiFi network, which we know is able to be done by doing MAC address to IP association. In that case it's not an automatic assignment, and you're no longer violating this bogus, hope this never passes, law. But it's there. And we'll see what happens.

Leo: Yeah. This is something that's been really around in conception since the mid-'90s, believe it or not, when the idea of data retention has been proposed. And it closely mirrors a law that's already in effect in the European Union.

Steve: Yes.

Leo: Now, the EU doesn't require this home stuff. And I think if that was...

Steve: Let's all hook a hard drive onto our $49 router.

Leo: It's crazy. And as you point out, this has nothing to do with child pornography. That's just the easy thing to say because nobody's going to say, well, I'm for child pornography.

Steve: It's the hook, exactly.

Leo: This is all about - and I'm sure this is written by the movie industry and the record industry, who really want to use this to facilitate their lawsuits against people who are pirating.

Steve: Yeah, they just have to, I mean, again, it's conceivable that an ISP would log so that the FBI, under subpoena, could say we have criminals at this IP at this time. We need to know who they are. And so it's like, okay. I mean, I can see on that scale it can make some sense. And we know that to some degree that's being done now. But to ask, I mean, the way this is written, to require anybody with a DHCP server, which is to say anyone with a router, even wired DHCP because that's automatic assignment - this doesn't say "wireless connections," it says "any automatically assigned IP." Well, that's what most people use. So it's just crazy.

Leo: I think that whoever is writing these just is kind of clueless, to be honest with you.

Steve: Yeah, well...

Leo: They're lobbyists...

Steve: Washington?

Leo: Yeah, Washington.

Steve: Washington, are you kidding? Clueless?

Leo: The lobbyists come in, and they say, hey buddy, we gave you 10,000 for your campaign. We really think this is important. Please, you know...

Steve: And besides, you don't want to expose children to pornography, do you?

Leo: Children, yeah. Can't lose in that. It's funny because the Democrat - the member of Congress, I think she was from Colorado, one of the few Democrats who supported this last time around is very active in children's issues. And so I think what happens is sometimes they're fooled. And if you don't have a technical background, you may not really understand what's at stake and the difficulty of doing this and the consequences and the privacy implications.

Steve: Well, and the only thing that's a concern is we would tend - we would like to have faith in the idea that a bad law cannot happen, except we have the DMCA, which demonstrates that bad laws do happen.

Leo: So the solution is, for those of you who are listening who are in the United States, write your member of Congress. Just make sure they understand the technical issues involved here. You know, what they're - I don't think you have to get into DHCP. But I think you might explain to them that, as written, this law would require every home user to keep two years of logs. Do you really think that's a good

idea? Nudge nudge. That's all you have to say. I'm glad you brought this up because it's driving me crazy.

Steve: Just nuts.

Leo: Yeah.

Steve: We have a zero-day exploit in Adobe's Acrobat Reader for which there is no patch. And it's actively being exploited in the wild. This was discovered in the wild, so it's an update - it's a vulnerability that Adobe is now aware of. They've got a link on their site saying, yes, we know about this. We're not happy about it. As a temporary workaround, if you disable - guess what - JavaScript in Acrobat Reader, which you can do, that will be a prevention for instances of the current problem, but it doesn't really fix the underlying cause. They have said that the most recent version of Acrobat Reader, the most recent major version, v9 is where they are, that it will not be until March 11th that they're able to get a patch out for v9 of Acrobat Reader; and another week after that, March 18th, before they're able to get versions 7 and 8 patched for those users who are still on 7 and 8 and haven't moved up to 9.

So there is an unofficial DLL that's been created by an independent researcher that replaces the acrord32.dll, the Acro Reader 32 dot DLL. I don't recommend using non-official DLLs from anybody. I would say disabling JavaScript sounds like a good thing to do in Acrobat Reader if you're a person who uses Acrobat Reader often. Or maybe just be extra cautious until this update. And we'll certainly advise everyone that there is a new version available as soon as it is available. That's not good.

The only cool thing is that our users of Sandboxie could simply make a configuration change in Sandboxie using the forced option. I've got, for example, Eudora and Firefox and Internet Explorer, although I don't use IE very much any longer, all tagged as forced programs. So any attempt to run them makes them run in a sandbox. You could simply do that to Acrobat Reader. When you click on a PDF in your browser, it runs embedded in your browser, so it's automatically sandboxed by the browser being sandboxed, so it's contained within that container. But if you wanted to, you could just add the Acrobat Reader executable as a forced program. And in fact that makes a lot of sense because PDFs are now a constant source of problems. We're hearing about security problems like this all the time. So I would absolutely consider adding Acrobat Reader, just as a general safety measure, if you're a user of Sandboxie on Windows. And then you're safe that way, also.

Leo: Always Sandboxie. It always comes back to that.

Steve: Oh, it's just a great tool.

Leo: NoScript and Sandboxie, and you're safe.

Steve: And, Leo?

Leo: Yes?

Steve: I got my Kindle 2.

Leo: I'm jealous. I know.

Steve: I got my Kindle 2.

Leo: Mine's supposed to come any minute now.

Steve: It really shows every characteristic of being a second-generation device. They've got the battery life extended.

Leo: Now, you haven't had it long enough to really notice that, have you?

Steve: No. Very good point. So that's what they're talking about. The buttons are very clever. The old buttons rocked from a pivot inside toward the outside. And that would cause the big problem of, any time you picked it up, someone would just sort of tend to squeeze the edge, and that would cause a page turn. These buttons, as you'll see when you get yours later today, Leo. they rock toward the inside. So the button pivot is on the edge, and you cannot push it. It will not push if you just push on the edge. So they've flipped that around. And so it takes a little getting used to. I mean, basically I was just so in love with my v1 Kindle that it became instinctual to use it.

So these buttons are very clever. They look a lot nicer. They're lots smaller, but still easy, when it's in your hand, to press the button. And so you get used to it. After a while you can sort of feel the little gap where the button meets the rest of the plastic. And so you sort of press there, rotating the button into the Kindle, in order to actuate it. They got rid of that wacky LCD strip and the little roller, the roller wheel that you push down, and instead you move a cursor around using a four-way, or actually five-way little joystick. I don't like it. It's smooth on the top, and I can't push it to the sides with my thumb. It's too smooth. So I have to kind of, like, get my thumb onto the side and push against one side. I hope - that's unfortunate. Maybe someone will come up with a little sticky cap for it or something. It really needs that.

But the one interesting bit, I didn't expect the text-to-speech to be at all, like, useful or good. And it actually is. Here is my Kindle reading from the book I'm currently reading called "The Chip," which is the introduction, or the invention, of the monolithic integrated circuit.

Leo: I just want to point out, while you're holding it up, that Andy Ihnatko said, and now that I see it...

KINDLE 2: ...Texas Instruments. Texas Instruments today, largely because of Jack Kilby, is a global semiconductor giant, one of the world's leading manufacturers of

microelectronic devices. In 1958, though it was just beginning to make a mark in the electronics business - the company had been born in the mid-'20s as the Geophysical Research Corporation...

Leo: Wow. That's completely listenable.

Steve: I know. I mean, it really is.

Leo: It's completely listenable. Now, I just want to say something. Hold that up again, for those who are watching at home. Turn it around. Andy Ihnatko said, and now that I see it I completely agree, if you flip to the other side, the metal side, he said it looks like it's an iPod designed for Andre the Giant. And he's exactly right. It looks like something Apple - it does look much prettier.

Steve: Well, and even the packaging, when you see it, it looks like, you know, very Jobsian in packaging. They went really over the top in terms of the way they packaged the thing. It's like, okay, this is very familiar looking.

Leo: We're going to turn you into an audiobook listener, though; because you're right, that's pretty listenable.

Steve: Yeah. I mean, you could, if you were reading, and you wanted to continue that experience while you're in the car, it works. And in fact, one thing is really funny. It can't say its own name very well.

Leo: What? What does it say? [Attempting pronunciation]

Steve: It's a, well, let me go there.

Leo: You'd think they would have fixed that.

Steve: Exactly. It reminded me - do you remember the movie "Colossus: The Forbin Project"?

Leo: Yeah, yeah. Loved that movie.

Steve: They gave Colossus a voice. And it was interesting because there was one word that it conspicuously did not pronounce well. And I don't remember if it was, like, human or something. But it was like if, you know, ..."and then all humans will be able to," you know it's like, whoa, you know. Okay, so...

Leo: Now, the downside on this, it is still very expensive. It's $359. But if you travel a lot, or you carry books with you all the time, you are always going out with your books, this is such a boon. Is it too hard to turn the page?

Steve: So here's - no, no. I'll get to that in a second. So here is the Kindle reading its own "Hello to the Kindle" book that it comes with.

KINDLE 2: Thank you for purchasing Amazon Kindle 2. You are reading the Welcome section of the Kindle 2 User's Guide.

Steve: I guess that's not bad.

Leo: That's not bad.

KINDLE 2: ...provides an overview of Kindle 2 and highlights of...

Leo: It's just too - I noticed a lot of words are like that, that they're cut. They're clipped.

Steve: Yes.

Leo: It's almost they go to the next word too fast.

KINDLE 2: ...to turn to the next page, press one of the Next Page buttons. If your Kindle was a gift, you will need to register your device. Please look at the Getting Started instructions that came with your Kindle for information on registering your device.

Leo: It's swallowing the word, that's what it is.

Steve: Yeah, and you know, Leo, what strikes me is that the intonation is really good. I mean, it sort of goes up and down, and it has highs and lows. I mean, they clearly gave this thing much more attention in the text-to-speech aspect of it than I expected them to.

Leo: Yeah, yeah.

Steve: But no, the page turning is great. It is dramatically faster. They said 25 percent faster paging, turning the page. I think it's much faster than that. I mean, it's no longer like - I used to be, and I'm sure you were, toward the end of the last sentence of the last line on the screen I would hit the page turn, knowing that I'd be able to finish reading it before we got to the new page. And that behavior will get washed away. Oh, also, many

things they fixed. You can easily delete something from right there on the screen using - now that we have left and right thanks to this little joystick positioner, you're able to just push it to the left, and it says, you want to delete this? And of course it goes off, if it's a book, it goes back to Amazon land, where your library is archived. And if it's a periodical, it's just gone. And their default seems to be not to keep old back issues of periodicals, which I think is a mixed blessing. You had to use the Content Manager, which was sort of painful, in the first version in order to go in and, like, clean out all the newspapers.

Leo: Oh, it's so hard, yeah.

Steve: It was really annoying. It was slow and took a long time. Now there's a menu option right there on the main menu to save this one. So it knows it's a periodical. And so you can say "Save this edition." So my guess is that - and again, I've only had it for one day, so I haven't seen this work. But by default it will replace the next version of The New York Times with the prior one.

Leo: That's how it should be. You don't want to read days-old newspapers.

Steve: Right. And also that marking feature used to be really frustrating because it would only mark the page you were seeing, not the article. Now it marks the article.

Leo: Oh, good.

Steve: So it knows about that. It knows about article boundaries, too. So anyway, I'm very pleased with mine. I have to say, I mean, I have such an affection for the first version, I kind of picked it up this morning after I'd been using the Kindle all morning at Starbucks, and it was like, awwww, Little Willy, I still kinda like him, you know? He was...

Leo: Well, what are you going to do with that? Are you going to keep it around as a spare, or...

Steve: Oh, no, I've got to - absolutely going to keep it around. I have all the packaging and the original materials. This all goes into the master GRC archive because someday it'll - in 30 years it'll be like, oh, look at that. Remember when we thought that was really cool?

Leo: Yeah, yeah, yeah.

Steve: You know, it's when we have automatic eyeballs that are scrolling text on them or something.

Leo: Well, knowing - see, I always gave away all my old technology. But knowing now that both of us kind of covet the old stuff, and you're spending money on eBay

buying it back, I think it's probably a good idea to keep this stuff, just for...

Steve: Well, and I've got to say, the fact that it is without a cover, that's a conspicuous fault, I think. It is, I mean, it's - you want to take care of it. You don't want to scuff up and scratch up the brushed aluminum back. But the fact that it doesn't have any, even a cheesy cover, where you could buy a nicer one for yourself, it's like, I don't know. I think they made a mistake there. I've got a couple covers coming. The neoprene one, I think, is the one I will probably like because I think - because I like holding it without anything encumbering it. So I was always taking it out of the little book cover that they provided. But that's where I put it back to protect it. So anyway, I think they've made a great jump forward. This is...

Leo: Which case did you buy?

Steve: I got the neoprene one, and I got the top-of-the-line glove leather, the black leather Cole Haan.

Leo: And which are your thoughts right now?

Steve: Oh, neither have come yet.

Leo: Oh, so you haven't seen them, okay.

Steve: So I haven't seen it. I'll give it to you next week.

Leo: Good.

Steve: Yeah. And I wanted to remind our live listeners that I'm going to be on Maxwell's House tomorrow, yeah, tomorrow, Thursday at 2:00 p.m. with Ray Maxwell. We're going to do a sort of really fun walk down nostalgia memory lane, talking about our first experiences computing. I've had a lot of revelations as I've been relearning the PDP-8 instruction set. I've got some fun stuff to share with our listeners for that. Unfortunately, non-live listeners - oh, wait. We decided we're going to make a podcast of it and just slip it in as an extra Security Now! podcast.

Leo: Yes.

Steve: So even our regular listeners will be able to hear it, but not live.

Leo: You'll hear it later this week. We're going to tape it on Thursday. So - tape. We're going to record it digitally on Thursday.

Steve: Speaking of a walk down memory lane.

Leo: And we will post it probably Saturday or Sunday. So this weekend you'll get an extra - just so you know, you'll get an extra Security Now!. This is 185. We'll number it 185A. And just dispose of it if you're not interested. But I have a feeling anybody who listens to this show will be very interested in this trip down memory lane.

Steve: Well, for people who don't know Ray, he's a tremendous wealth of knowledge and experience, and a neat guy. And I think we're really going to have fun.

Leo: I can't wait. I can't wait.

Steve: Meanwhile, "Another Success Story" was the subject line. William J. Burlingame sent this. He said, "I got an SOS from my daughter this past Wednesday. Her system wouldn't boot, and she has yet to get the backup religion. She had the original recovery disk that came with her system; but it indicated it would restore everything to the way it was when the system was first delivered, sans critical data. She had financial data, pictures, my grandkids' homework, et cetera, that needed to be retrieved. I packed up my black bag and headed out to make a house call.

"I ran SpinRite in the Recover mode. It took about an hour to complete. Although there were two unrecoverable errors on the SpinRite map, the system booted just fine, and we were able to back up all her critical data onto an external drive. I don't recall which version of SpinRite I first purchased. But it was before the introduction of Windows. My first hard drive was a 5MB that was an upgrade to my original IBM PC that came without any hard drive. It was a full 5.25" full height drive and was quite heavy. I enjoy the TWiT conversations with Leo." So, William, thanks for sharing your positive SpinRite experience.

Leo: That's great. We just love Spin- everybody here uses SpinRite. Colleen, you know - and by the way, Colleen is now a full-time employee.

Steve: Oh, yay. I heard that you were going to do that. That's great.

Leo: She is - we have converted her into a - she is your best evangelist.

Steve: Well, we hear things like, well, SpinRite did what it did. It fixed the drive, but there were two sectors that were unrecoverable. Well, it's painful, I described last week or the week before why even that is a tremendous benefit because SpinRite is able to give you all but a couple bytes, even of an unrecoverable sector. And in doing that I'm pretty sure it's unique in history.

Leo: Yeah.

Steve: But I just think if, again, I recognize I'm not going to get non-owners to do preemptive use of SpinRite, even though people who once discover SpinRite then run it every, well, about every quarter, maybe four times a year, three times a year. And in doing that, it's able to catch problems that are developing and fix them and/or cause their sectors to be swapped out before you get to this problem of something critical being unrecoverable.

So the one thing I notice that's so redundant about the email that I read, the testimonials, is that time after time it's the system wouldn't boot, the system wouldn't boot, the system wouldn't boot, because that's an all-or-nothing sort of see-your-life-passing-before-your-eyes sort of experience. But well before the system wouldn't boot, had SpinRite been run, it would have solved the problems, I mean, preemptively, before it got to the point that it wouldn't boot. And you'd have never had that problem, which can still cause problems. So anyway, I recognize that we'll convert people when their system finally gets into such bad shape that it will no longer boot. And at that point they'll be able to run SpinRite preemptively and use it as a tool that prevents them from having that problem in the future.

Leo: Yeah. No, Colleen's become the maintenance, SpinRite maintenance queen here. She goes all over the place, SpinRiting things. But it's true, as the drives get bigger, this becomes more and more important. I mean, these drives, I mean, think of all the error correction they're already doing.

Steve: And because they have so much valuable information on them now.

Leo: Right.

Steve: Stuff like his grandkids' homework, in this case.

Leo: Yeah. Geez. All right. HMAC.

Steve: Yes.

Leo: What is it?

Steve: Okay. Well, this is a final piece of sort of core component technology that we need to understand in order to be able to talk about in detail the most used security protocol of all time, which is SSL, that we all use every time we establish a secure connection with our browser to anywhere - to Gmail, to our bank, to PayPal. Oh, speaking of which, I forgot to add to my notes. Remember last week I talked about having found, myself having discovered a glitch in PayPal's login that allowed me to bypass the use of my security token, the football. And I guessed then that what I was doing was more than was the minimal necessary, and I was correct.

If any time you are in eBay, which doesn't require any sort of an authentication token to log into, and you leave eBay to go to PayPal, for example to pay for something through an eBay link, you come to the PayPal login screen. You give them your email address and

password, or your first-stage login credentials. If you own and have registered any security tokens, you will then naturally go to the next stage, which is it's asking you for your tokens. All you have to do is hit back arrow. You go back to the login screen. Now the text says you're already logged in. So the fields are grayed out that you filled in, and the button that used to say "Login" now reads "Continue." So you click that, and you're into PayPal without having to authenticate using your security token. Not good.

Leo: Pardon?

Steve: Not good.

Leo: What happened?

Steve: No, no, I mean it's not good that you don't have to...

Leo: You woke me up, I'm sorry. No, you know what happened? My Kindle came.

Steve: No kidding.

Leo: I just stepped out for a moment to get it. And then I come in, and you say "Not good," and I went, what? So that is not a good thing. And we'll open the Kindle after the show. I won't be distracted by opening it now.

Steve: Okay, well.

Leo: It's tempting.

Steve: Congratulations.

Leo: It's tempting, let me tell you.

Steve: The other thing that they did that I thought sort of was fun was it used to be that the Kindle would deliberately blank its screen when you turned it off. And as we know, the screen technology absolutely doesn't require it. It's literally - in that sense it's like an Etch-A-Sketch. The Etch-A-Sketch, after you've scraped the light gray dust off the back and so that it's black there, it just sits there. The Kindle's the same way. So what I love is that the Kindle ships with a static screen of instructions about how to turn it on.

Leo: Oh, that's clever.

Steve: And but then, from now on, when you turn it off, you get one of those screensaver images that they used to only put on when the Kindle would put itself into sleep mode, if you put it down. Now it puts it up and turns the Kindle off. So it used to be that they would blank the screen to give everyone a sort of a warm, fuzzy feeling that, okay, it's turned off now. But now they deliberately put an image up there. So it's cool.

Leo: Very cool.

Steve: Okay. So anyway, so I did verify the minimal approach for logging into PayPal from eBay that is a security dongle bypass. It works every time. And it's unfortunate because that's why you told PayPal you wanted a security dongle.

Leo: Yeah, yeah.

Steve: So, whoops.

Leo: Whoops. Not good, as you said.

Steve: Not good. Okay. So, message authentication codes. We've talked about basically every other aspect of security except MACs. MACs, Message Authentication Codes, is sort of the complement to security. That is, as we know, you encrypt a message in order to obscure what its contents are. But sometimes you don't want to encrypt it. Or if you do want to encrypt it to hide it, you want to detect any changes. Now, we've often talked about message digests, MD5, which we talked about weeks ago having been weakened to the point where it's really no longer secure, SHA-1, and then other types of message digests. Those are all hashes.

And so the traditional way, for example, of verifying the integrity of a document, whether encrypted or not, is that you would run it through one of these digest functions, which cryptographically digests the content you're feeding through, and you end up with essentially a token. In the case of MD5 it's 128 bits. In the case of SHA-1 it's 160 bits. The newer, stronger, so-called SHA-2 functions, which is now what is being recommended, they're even longer. They're, like, 256, 512, 1024, much longer. The length gives you more security. And their newer modern design has also enhanced their security.

So we've also talked about cryptographic signatures, where the way you sign a document is you hash it into one of these tokens that comes out of the digest function. Then you, for example, if you wanted to sign the document you would use your secret private key to encrypt just that token rather than the entire document because, as we know, public key technology is really too compute-intensive for it to be practical to sign the whole document, or to encrypt the whole document. And in this case you might want to sign it without encrypting it.

So you sign, that is to say, you encrypt with your secret key just the output of the hash, and append that to the document, send it to somebody. They apply the same hash function, and that gives them that token. Then, since they don't have your private key, by definition, your secret key, they're not able to encrypt that. But they did receive the result of your encryption, which they're able to decrypt with your public key.

So now they've got - they've decrypted the signature that you attached. They can compare that to the hash they independently made. And the logic is, the only way those things will compare is if the document hasn't changed. Well, we know because we were talking about MD5 that, if the hash function is weakened, that creates a problem. And the problem is - and it's exactly what I just described. If the attacker has some control over the creation of the documents - and this is still within relatively strict guidelines. But the point is, since we're encrypting the hash's output, if we can make modifications that cause the hash to give the same output, then we're going to get a valid signature even though the documents change. So that's a problem.

Message authentication codes, properly designed message authentication codes, don't have this problem. And in fact, if in the case of security certificates, if they were signed using a properly designed message authentication code, then it turns out that, due to the nature of message authentication codes, they are not as dependent upon the strength of the underlying hash. And that's really important. For example, an HMAC could be based on MD5. And I'll explain what an HMAC is next. But you could base it on MD5, even in its now-weakened state, and you lose none of the integrity of what the message authentication code is authenticating, which is, not surprisingly, the message.

Leo: Right.

Steve: Okay. So cryptographic ciphers have been used in the past. We talked last week, or sorry, week before last, about a so-called CBC MAC, a cipher block chaining MAC, where you take blocks of the text, and you encrypt it, and then you XOR the output with the next block of text and encrypt it, and XOR that with the next block of text and encrypt it, in a never-ending chain until you get all the way done.

The nice thing about that is that it's a strong fingerprint, a strong signature for the text. However, you're using a cipher. And traditionally, hash functions are faster in software than block cipher functions. Also, software implementations of hash functions are freely available. Whereas ciphers historically have been patented. And it's only recently that those patents have expired. So ciphers have tended to be encumbered by intellectual property, whereas hashes have not been.

And ciphers, as we know, have also historically suffered from export restrictions. Hashes never have because hashes can't encrypt. They can only digest. So whereas ciphers can encrypt, and so they were unfortunately qualified as a munition, and exporting them from the U.S. and other countries had been prohibited. So there were some benefits that hashes had over ciphers. That is, there were reasons that you would prefer to use a hash function in order to generate a digest of a message than using a cipher. Otherwise you could definitely use a cipher.

So some of this is sort of historical. But the crypto guys have looked extensively at the way hash functions are used in message authentication codes. And the idea is that you want to incorporate a key into the hash function. Notice that when I talked about the CBC MAC, where you use a cipher block chaining, you use any symmetric cipher to encrypt a block of text and then XOR the output of that with the input of the next block in this chain. Well, implicit there is that symmetric cipher. So the nice thing about a CBC MAC is that it's a keyed digest, meaning that the output that you get is a function, not only of the digested content, but of the key. Whereas, for example, MD5 and SHA-1, any of the standard hash functions, they're not keyed. They're just MD5. So the advantage of that is, if you were just using it as a sort of a simple message authentication, or like that

a file had been modified - we talked about this in the case of websites where websites will post the MD5 and sometimes the SHA-1 of a file that you download. The point is you're able to independently run the same function that they ran prior to posting the site on a server and verify the output.

The reason you can do that is MD5 is MD5 is MD5. Anyone who runs something through MD5, that runs the same thing through MD5, is going to get the same output. That's its benefit. However, there are instances where, for example, for authenticating a message, where you want to have a secret key as part of this, that is, you want to be able to say here's the message, yet I want to prove that I signed this. So you could generate a random number and use the random number to key a keyed hash function. That generates an output. Then you use your private key to encrypt that random number.

And so now you send to somebody the document and the result of the hash function. They have the encrypted random number which was used to sign the document, to run the keyed hash function. They're able to decrypt it using your public key. And then, if they apply the keyed hash function using that as its key, they'll see that it matches. The only way that's possible is if you were the person who encrypted the key using your private key, because in using your public key you're only going to get the proper result if the public and private key match. So there's sort of another way that you can see that all of this fits together.

So the question is, how do we turn a hash function - which is inherently unkeyed, MD5 or SHA-1, they're not keyed - how do we turn them into something that takes a key? Well, the simple-minded way to think of it, the first thing people thought was, oh, well, let's just put the key at the beginning of the message, or at the end of the message, just sort of like add the key to the message. Since we know that any change to the hash function will result in a different output, if we put the key at the front of the message, then every time we change the key we're going to get a different hashed output. It turns out that, unfortunately, the nature of hash functions makes that insecure. The crypto guys in analyzing this said, eh, that's not such a good idea. And then people said, okay - and the reason is, we've seen that one of the main ways of exploiting hash functions is by altering them in a way that changes their length but not their outcome. That's exactly what we saw in this case of the attack on MD5, with the chosen prefix in an MD5 hash.

So then people said, because they understood that length was the problem, okay, let's put the key and the length and then the message, glom them all together, and digest that. It turns out that the crypto guys, in analyzing this to death, said nah, that's not good either because there's ways you can exploit that. And then they said, okay, how about if we put the key at each end? And it's like, well, okay, that's better. But we have something that would really work because in pounding on this the crypto guys saw what it was, like what advantage they had. And the advantage they had is they were able to see the output of the hash function. They were able to tell what it was. And that gave them the leverage that they needed against the hashing function in order to make this work. So it turns out that, if you hash it twice, with a couple tricks, that's the key. So specifically, this thing called an HMAC is now a formal standard because it's passed through all the crypto analysis. No one's found any sort of a problem with it.

So here's what you do. I talked about, when we were talking about hash functions in depth, the idea that a hash function - this was when we were talking about MD5 - a hash function tends to process its input in blocks that are larger than its resultant hash. That is, for example, both MD5 and SHA-1 take 512-bit blocks. They take the digest 512 bits at a time and then run that through their algorithm. And every time through their algorithm it sort of gives them another state. And then they take that state, sort of like how far they've processed so far, and then they take the next 512 bits and that state and

run that through the algorithm and get another state, an intermediate state. So the final result is 128 bits, or 160 bits, however long the hash function digest output is.

So to securely key any hash function, you take the key and pad its length out using a specific pattern of bits. What's been chosen for the first hash is hex 36. So you take - essentially you take hex 36 repeated out 512 bits, and you XOR the key with that. So what that does is it's sort of a - it gives the hash a good start for the first 512-bit block. You then process that, and then continue to process the rest of the message. So essentially what this does is it takes the key you're giving it, and this magic 36 hex that tends to flip some of the bits of the key because we know that that's what XORing does. And when you run the first cycle of the hash function, it initializes it to, after processing this 512-bit block, to a strong state based on the key. Then you just run the regular hash over the rest of the content.

Okay. You take that output, and you do the same thing. You take its output, and you take the same key. This time you XOR it with hex 5 Charlie, 5C. I'll talk about where those two values came from in a second. So you do the same thing. You take 512 bits' worth of 5C in hex. You XOR the key with that, and initialize the hash function using that. Then you simply hash the result from the first hash. So you've essentially nested hashes. You're hashing the output of the inner hash. That's your final MAC, the Message Authentication Code. And it has passed all crypto. I mean, people have pounded on it. They cannot find a weakness. They recognize why. Because of doing such weakness analysis on hash functions, they realize that getting the result of the first hash is the key. Well, by masking that with a second hash, making that first hash be essentially an intermediate value, they have no access then to what is really going on, and nobody's been able to break it. And it is so strong that even a weak hash like MD5, where we've learned all this about it, it doesn't weaken the output of this at all.

Leo: Cool.

Steve: And so that gives us a keyed message authentication code. It is used in SSL. It is very handy for communications. And I'll be using it myself in my forthcoming CryptoLink product because a keyed MAC is a very strong way, much stronger even than, as we saw, signing the output of a non-keyed hash, which is where the vulnerability in the SSL certificates was created because they were just using MD5.

Leo: Now, when I use OpenPGP - and I use it all the time for authenticating messages, verifying messages, not for encryption - it puts a hash string at the bottom. Is that an HMAC?

Steve: I don't know. I've not looked at PGP enough to look at the protocol...

Leo: I would guess it is. I mean, what it's doing is it's doing two things. It's verifying the sender using the sender's public key, but also verifying the message content's integrity. So it must be hashing the message; right?

Steve: Well, yeah. But again, just a standard digest, an MD5 would do that.

Leo: That would be sufficient, yeah.

Steve: My guess is that they're using an HMAC, a keyed MAC, and they're keying it based on your private key. But I don't know.

Leo: Private key, not public key; right, yeah, yeah.

Steve: Right. We will be looking at SSL in detail, and it sounds like looking at PGP would be a good thing, too.

Leo: Yeah. This is - I'm using OpenPGP, but I think it uses the same standards as PGP. And then there's GNU Privacy Guard, which is...

Steve: GPG.

Leo: GPG. And that's what I use to implement OpenPGP, as confusing as that is.

Steve: Well, if anybody's still with us, given where we've just gone, I think that was easy compared to understanding the nature of keyed message digests. But these are super useful, very strong, and they are the final component that we hadn't talked about, that we needed to sort of lay down in order to understand, to look at the SSL protocol, what is it that goes on every day when people hook up their web browser to a server remotely? How are they safe? How do they know they're safe?

Leo: Right.

Steve: We've talked about the certificate side. We haven't talked about the actual communications protocol side. And we're going to do that in a few weeks.

Leo: Very cool. As always, Steve Gibson, fascinating material. And if you put together this with the last few sessions that we've done on crypto, you have a very strong basis in modern crypto techniques, which is great.

Steve: That's what we do here.

Leo: That's what we do here. And a little bit of news thrown in.

Steve: All kinds of interesting...

Leo: Little bit of news thrown in.

Steve: And a little eBook...

Leo: And I've been very good. Lookit, still sealed.

Steve: Ah.

Leo: You know, it's kind of cute. I actually got two boxes, and I didn't know which one was the Kindle. But I figured out this is the Kindle because down the side it says "Once upon a time."

Steve: Oh, I know.

Leo: So you're right. This is the Apple-style packaging inside here, I think.

Steve: And I tried to open the box without breaking the little zipper because I thought, you know, I wanted to keep it in, like, really pristine condition.

Leo: Right.

Steve: But they'd glued it all down tight. So you've got to pull the zipper in order to do that. So it's like, eh, that's okay.

Leo: I did get my Patagonia case. This is the neoprene case.

Steve: Oh, good.

Leo: Yeah, but this, I don't know if this has the hooks. Because, right, the new Kindle has hooks that go in the leather case. But this one looks like it's got a pouch. I can't - well, I'll know...

Steve: That's what I was - actually that's what I was expecting. I think those zipper cases are just - they're just a case you slide it into. The ones that have, like, a foldable lid, a flap, like a wallet, those have...

Leo: Those have hooks here.

Steve: ...elastic corners.

Leo: Well, this has elastic corners. This has elastic corners.

Steve: Oh, it does.

Leo: Yeah. So that's the mechanism? Because I also thought there were now - there was some sort of hooking mechanism now on it.

Steve: Nope. They pulled the buttons away from the edges so that your - so that all the corners are now available for tying it down.

Leo: This is nicely padded and has a little handle. I think this will be fine.

Steve: Neat.

Leo: Yeah. But I will be very interested in your lovely leather Cole Haan case, as well.

Steve: Well, I'll show it to you next week when we talk. We'll do a Q&A next week. And then the week after we're going to talk in detail about - given that nothing else really horrible or significant comes up. We always reserve the right to change horses here if something happens. We're going to talk about tuning Windows AutoRun so that you are able to run just exactly what you want and not anything else.

Leo: That's very useful. That's a great one to do. We should mix the really hard academic math stuff, like this, like today's, with useful pragmatic stuff. That's great to have a little bit of each.

Steve: I think so, too.

Leo: And that's what the Q&A is great for. We cover everything. If you've got a question for Steve you'd like to send along for next week's question-and-answer session, go to GRC.com/feedback. And there's a feedback form there, and you can submit a question. GRC is his site, the Gibson Research Corporation. That's where you find SpinRite. It's where you'll find all those great free tools like ShieldsUP! and Wizmo and Shoot The Messenger and DCOMbobulator, Unplug N' Pray, I can go on and on and on, all the free stuff he's writing. That's where the new crypto product will be. Perfect Paper Passwords is there. Also, oh, don't forget, I kind of remind people, we were talking on the radio show about WPA passwords. I sent people to GRC.com/passwords, and you can get that great 64-character uncrackable WPA password. And a new one every time you visit.

Steve: You have Windows machines running there; right?

Leo: Lots of them.

Steve: Okay. Because I've got to say, Leo, this DNS benchmark utility that I'm working on, it's going to be really significant. It's looking like there are a lot of publicly available free DNS servers that are a lot faster than ISP servers.

Leo: Good.

Steve: And this thing knows about them all. So you just run it from your machine, and it tests all of the known publicly available DNS servers' performance against all the ones you're using from your ISP, shows you it, ranks them, and says, you know, you switch your DNS to this, and you're going to get this much more speed.

Leo: Fantastic.

Steve: It's really going to be very cool.

Leo: Fantastic. We'll give it a big plug. Don't forget, now, if you're listening to this show - I hope you listened as soon as it came out. We're going to put it out a little bit early on Thursday to give you a little heads-up that at 5:00 p.m. Eastern, 2:00 p.m. Pacific, Steve will join Ray Maxwell on our show Maxwell's House at Live.TWiT.tv to talk about the old days of computing, the old days of programming, PDP-8 and more. That should be a lot of fun.

Steve: And Leo, when every bit mattered.

Leo: Those were the days, when you had to really program bare to the metal, and you had to pay attention in memory and things like that. That was when men were men, and programmers were programmers and strode the earth. And anyway, it'll be fun. 2:00 p.m. Pacific, 5:00 p.m. Eastern on Thursday, February 26. If you missed it, don't worry because Episode 185A is the entire audio of that show. So you'll be able to hear that, and we'll put that out on Saturday the 27th. Or 28th, I guess that is.

Steve, great to talk to you. Oh, have I left anything out? Yes, transcripts available, GRC.com. Also 16KB versions of this show. We now have the great wiki. The show notes are in there, too, at Wiki.TWiT.tv. That's another great resource with links back to Steve's site. So we really wanted to make sure there was a lot of text-based support for all this because, you know, it's not enough just to listen. You've got to read, too. Thank you, Steve.

Steve: Thanks, Leo. Always a pleasure.

Leo: See you next time on Security Now!.