SECURITY NOW!

Transcript of Episode #78

DEP in Depth

Description: With Steve's new SecurAble freeware now launched, he and Leo discuss the full impact and importance of hardware DEP technology. Steve explains why he believes that hardware DEP is the single most important Internet-related security technology developed so far.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-078.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-078-lq.mp3

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

Leo Laporte: This is Security Now! with Steve Gibson, Episode 78 for February 8, 2007: Hardware DEP.

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

Security Now! is brought to you by Astaro, makers of the Astaro Security Gateway, on the web at www.astaro.com.

I may be afloat in a boat somewhere in the Caribbean, but that's no reason not to do a Security Now!. Hello, Leo Laporte here with Steve Gibson, the security guru. Hello, Steve.

Steve Gibson: And so are we depending upon your new software gizmo to have automatically posted this?

Leo: If you're hearing it, it worked.

Steve: Ah, very cool.

Leo: No, I can actually get online from the boat. But it's very expensive, and it's something I kind of like to eschew because I'd really prefer not to be posting podcasts from the sea.

Steve: Oh, I love that you have the technology to post this on the fly. That's going to be very cool.

Leo: Well, we'll see. We'll hope it works. The software I use is called Feeder from Reinvented Software. And the new edition, which just came out, has scheduled posting. So if the Mac didn't crash while I was gone, and if the scheduler worked and everything, this should have gone up at noon Pacific time. And if not, I will check it from the boat. So the truth is you could be hearing this and it didn't work, but I just made sure that it did. As much as I say I don't want to get online when I'm on a boat, I always end up doing it.

Steve: Yeah, you're a geek.

Leo: It's slow. It's really – it's satellite, it's very slow, and it's something like 40 cents a minute, I mean, it's very expensive.

Steve: Oh, so lots of latency, too, for the packets.

Leo: Huge, huge. It's not so bad for, like, email. But, boy, web surfing is just painful. Posting podcasts isn't so bad because that's pretty asynchronous. You do that all locally, and then you press "Post," and then it doesn't take that long. Hey, let's talk about – we don't have any errata or things to talk about from previous episodes because we recorded these en masse before I got on the boat.

Steve: Right.

Leo: So we may do that next week, depending on how much we got wrong. Which is none, I'm sure. But we do have...

Steve: We do the best we can.

Leo: We have a great topic today, something you've talked about before, data execution prevention.

Steve: At the risk of beating this thing into the ground, I want to make sure people who could be using it will be using it. The greatest problem is that it isn't enabled by default, not only maybe by people's BIOSes, but certainly by Windows. This was one of the substantially new and important security features which Microsoft introduced with Service Pack 2 of Windows XP. So it's been around now for years. When they first introduced it there were only some AMD processors and the Intel Itanium processors which supported this in hardware.

The idea is, and we've talked about in detail buffer overflows or overruns, and we had a whole podcast about it. So if listeners are feeling a little bit like they're late to the party on this, by all means, I don't know which episode number it was, but it was just titled "Buffer Overruns" [Episode 39]. We talked about in detail how it is that software which doesn't protect the amount of data it allows in can accept by mistake too much data. When that happens, the data flows off or falls off the end of the buffer. And due to the architecture of PCs and computers in general, it's possible that the computer inadvertently executes that data when it's through doing what it was doing, essentially gathering this buffer all together.

Leo: Whoops. Not good.

Steve: Not what you wanted. So the idea is that, because this has been such a huge problem for the industry, I mean, buffer overruns are just extremely difficult to catch a hundred percent of, and hackers love them because they're able to essentially send something to a remote user and essentially inject their code as data into someone's computer. The computer then runs it, and you no longer have control of your machine. I mean, it's epidemic, it's such a problem.

Well, responding to that, the hardware makers have added a capability of flagging pages as non-executable. Intel calls this, let me think, Execute Disable. AMD calls it NX, for no execute. Or maybe I got that backwards. I can't remember who is who.

Leo: No, I think you're right, yeah.

Steve: Unfortunately, AMD, who was early to the game with this, they also call it EVP, Enhanced Virus Protection, which is a really annoying sort of marketing term. It doesn't talk about what it actually does at all because it's really not about viruses, it's about preventing the execution of data. And so the idea is that, when the operating system knows that the process supports this, and when the processor has it enabled, the operating system can load programs which are going to be executable, marking the memory the program occupies as being executable. But when the program allocates memory for use for its communications buffers, or even the so-called "stack" which the program uses for holding data, the operating system will deliberately mark those as non-executable.

The beauty of that is that virtually solves the vulnerability of even, well, non-known exploits. That is, if a hacker finds an exploit which no one knows about and tries to inject their own code into the user's data buffer, the system itself will intercept this code before it has a chance of executing and raise a dialogue on the user's screen, telling you that this program has violated data execution prevention and is being shut down. So what I love about this is it is a preemptive protection against this kind of exploitation happening anywhere in the system. And whereas antivirus has to be updated with AV patterns; and there's always the cat-and-mouse game where the patterns are behind, and you're making sure you've got the latest updates. And the point is that antivirus is only able to protect against known specific problems. This whole DEP, the hardware DEP support, actually sort of cures a systemic vulnerability that all systems and software running in the system has. It's fantastic.

Problem is, it's not turned on by default because there are good programs which, because this has never been enforced in the past, good programs might cause false positives. And there are some that do. There was a version of my past favorite antispyware software, Ad-Aware, which was causing a problem. And in fact, even in some cases, in some settings of hardware DEP operation, the current version of, I think it's pronounced IrfanView...

Leo: IrfanView, yeah.

Steve: IrfanView is a neat little and very popular little image viewing tool. It's currently, as I'm writing this, or as we're talking about this, it's at version 3.99, I believe. And if I set DEP into always-on mode, IrfanView will not open. The operating system shuts it down before it has a chance to execute. So there's something going on there which is causing a problem.

Leo: Irfan is a long-time, old-school hacker. And obviously he's modifying his own stack.

He's putting code in the stack or something.

Steve: And that's a perfect segue into one of the things that people who start experimenting with hardware DEP support will find, is they may find that there are things that cause problems. Now, Microsoft has four modes in which hardware DEP can function. But before any of them have a chance of working, the BIOS has to not be turning it off when you boot. Some people's BIOSes will have some UI, some configuration pages that allow you to enable hardware DEP support if the BIOS sees that the chip offers it.

So one of the things I did with SecurAble, the reason I created SecurAble is it's able to check to see whether it is available in your chip, and whether it has been disabled by something, probably your own BIOS, when you boot up. It turns out that one of my tablet PCs, my favorite little TC 1100, my HP tablet, it has a Pentium M processor that does support hardware DEP, this data execution prevention. But the BIOS is disabling it, and there's no way for me to enable it. Which is very annoying. So that's one of the reasons I'm doing a follow-on utility called DEPuty. It will allow things like that to be fixed in what I hope are some clever fashions.

Leo: I like that name.

Steve: DEPuty's going to be free also.

Leo: I like it.

Steve: So anyway, already in our newsgroups people who are running SecurAble, which is just GRC.com/securable, many people were surprised to find that their chips did support hardware DEP. As I mentioned, at the time this first came out in Service Pack 2, only a few AMD chips and Intel's high-end Itanium processors supported this ability to force an exception being raised to the operating system if the processor tried to execute code in a page that was marked as not being code executable. Now, and for the last two years, virtually all AMD chips and all of Intel's chips, for example my older Pentium M that is now a few years old, do support hardware DEP natively. So you have to have support by your processor. SecurAble tells you whether or not you do.

Then these people who were surprised, for example, in my newsgroup, they went into their BIOS and poked around, and most of them – in fact I don't think I know of anyone except in my own case – they were able to find configuration settings that were defaulted to disabling this execution prevention. So it was really frustrating and annoying, first of all, that for whatever reason – again I think in the name of compatibility, making sure that nothing wrong would happen, when this feature was added to processors the BIOS makers were concerned that, if they left it enabled, that it would cause problems. So they were deliberately disabling it. Most BIOSes, certainly any well-behaved BIOS is going to allow the user to turn that back on.

Microsoft recently got an agreement from all major hardware vendors not to disable hardware DEP on boot, but rather by default to enable it. Because Microsoft has learned the lesson that defaults are generally what hold sway over PC users. And they knew that, if people had hardware DEP disabled, they'd just never go about finding out that they could get this additional protection, which as I'm claiming and as I believe is a substantial win for PC users. So now any machines coming out in the future, Microsoft has an agreement from all major makers, they're going to leave hardware DEP enabled by the time Windows boots, to at least then give Windows the opportunity to do what it wants to with it.

So now that we've got hardware DEP present in the processor and enabled by default, or you can use SecurAble to see if you’ve got it and go digging around in your setup pages in your BIOS to enable it if it’s not, now we come to the operating system. It needs to be enabled in order to have this working most usefully. Hardware DEP has four different modes of operation: always-off, always-on, or opt-in and opt-out. What’s interesting, and I’m going to be doing some further research on this, I can’t find any documentation on Microsoft’s site anywhere, because we’re seeing a difference between always-on and opt-out. That is, you would imagine that always-on mode would be the same as opting out if you weren’t having any opt-out programs. It turns out it’s not the case. For example, that IrfanView program, the IrfanView file viewer I was just talking about, it runs fine in opt-out mode, even if it has not been opted out. But it won’t launch, Windows blocks it from launching, if I’m running in always-on mode. So I don’t know why. I’m going to have to – I’m going to be doing some further research. So but that does make it clear that always-on is providing us with some sort of additional protection beyond using opt-out mode, even if nothing has been opted out of.

So Microsoft’s normal mode is opt-in. What that means is that Microsoft – and this is what they say, we don’t even know the extent of it, is they say on their various pages describing this that some Windows binaries are being protected. The presumption is, okay, not all of Microsoft’s own Windows binaries and programs, drivers and so forth are being protected, but some are. So I guess that’s better than none, obviously.

Leo: And why wouldn’t I trust their software?

Steve: Exactly.

Leo: I mean, unless I guess a malicious program could pose as that program.

Steve: Oh, no. No, no, that’s exactly it, Leo, is that certainly Microsoft has been the vector for many of these buffer overruns in the past because they had the bugs in their code that the hackers...

Leo: Well, I would hope they wouldn’t exempt Internet Explorer. That would be a mistake.

Steve: Well, and in fact it turns out that there are some toolbars for Windows Explorer that do have problems with DEP. So essentially at this point we’re in the frontier mode. What you really want to do, if you can, is run DEP in the so-called always-on mode, which is clearly the strongest of all. It works for me, except IrfanView doesn’t. Now, certainly Irfan, who is the author of IrfanView, he can fix it. So what I’m hoping will happen is, as hardware DEP begins to gain momentum and becomes more important, I mean, no one’s been talking about this for years since Service Pack 2, partly because we haven’t had the hardware that supported it. But even now it’s sort of like it’s not getting much attention. That’s the reason I’m going to be spending some time creating freeware and, as I am, spending time here talking to you and our listeners, Leo, on Security Now! to raise the awareness of this really important and significant benefit. Many of the security vulnerabilities that have occurred in the last year were already stopped cold by hardware DEP. That is, had it been in place in the system and enabled properly, those vulnerabilities were never a danger to those users who had their systems set up that way. This is just a very good thing to do.

Leo: I have two important questions for you. First of all, why would somebody code something – why would somebody need this ability to modify the stack or execute code from the stack? And secondly, if you can walk us through the Windows XP, at least, method of turning hardware DEP on, I would really appreciate that. But before I get those answers, if I might, I'd like to just mention Astaro Corporation. Could I do that? And then we'll get your answers?

Steve: Of course, of course.

Leo: This podcast is, of course, brought to you by Astaro. And I do want to make sure that I don't forget to mention Astaro. They've been such a great support to the show from day one. They've signed up for another year, throughout 2007. And we can't think of really a better partner for Security Now! since Astaro is, in my mind, one of the premier security providers in the world. They do the Astaro Security Gateway. That's a small security appliance that just packs all the power you'd ever want. Superior protection from spam, from viruses, from hackers, always updated. It's got a complete VPN. It's got intrusion protection, content filtering even. So it's great for an office. And of course it's an industrial-strength firewall, absolutely. But it's all in a single, very easy-to-use, high-performance appliance. You could even try it for free, just go to Astaro.com, or you can call Astaro at 877-4AS-TARO, and you can schedule a free trial of the Astaro Security Gateway appliance in your business. If you're a non-business user, you can even download the software and put it on a beige box, any old PC, and try it for yourself. Really a great company with a great product. And we are glad to have them onboard. Astaro.com.

So let's start with the first, which is why would Irfan write his program so that it executes code from the stack?

Steve: I really can't speak to that, Leo.

Leo: What's the advantage? You're a programmer. Why would you do that?

Steve: Oh, yeah, I know all about the stack, and I'm a programmer. There are, for example, he might have several different image display algorithms that he has compressed in his own code. So he allocates some memory to hold it, and then himself decompresses his own code into that memory, which is then marked – because it's presumed by Windows to be data, Windows marks it as no execute, even though Irfan's real intention, his own intention is to execute from there.

Leo: I've seen modifiable code and things like that where you put code in a data area, and then you modify it, depending on what you want the program to do. And it's a hack. It is an efficient hack, but it's a hack.

Steve: Well, and famously, the old Windows bitblit routines, that is, the bitblit is the term for moving a rectangle of pixels around the screen, like when you drag a window, you're actually doing blitting. You're blitting that around the screen. And the original Microsoft code, in order to create sort of self-optimizing blitters that would run as fast as possible, they built them on the stack and ran them on the stack. So there was a real reason once for doing that. Now, if Irfan knew that IrfanView didn't run under Windows always-on mode, he could certainly make a trivial change to his program. He'd find out where the problem was and why, and he's able to allocate the memory as executable.

Leo: Okay, so when you allocate the memory you can say this is data or this is code.

Steve: Yes.

Leo: So it's simple to fix.

Steve: It's so simple to fix. So what I'm hoping is that users will...

Leo: That he'll listen to this podcast.

Steve: Well, or that people will contact him. But more than just IrfanView, that as people start trying to live with hardware DEP turned on, they're going to run into some things that don't work. So they could, first of all, they could back off from always-on, which for whatever reason seems to be the strongest protection possible. They could come back a notch to the opt-out mode where they may have to exclude some programs that they wish were going to be protected, but they exclude them opting them out of DEP protection, in which case Windows will not mark the data they allocate as non-executable, allowing them to run. I'm hoping, though, that this will create pressure from the industry for programmers to clean up those programs which are not currently DEP-friendly programs, so that in a relatively short time people are then able to run in always-on mode, the strongest mode, and that problems become the exception. They are actually now an exception, but a much smaller exception so that it becomes really feasible to run our Windows systems this way. What I'm excited about is everybody with XP can do this. That is, you don't have to move to Vista in order to get this hardware DEP capability. This was added in Service Pack 2. So all XP users will have access to this. Now, your second question was...

Leo: How do I do it?

Steve: ...walking our users through. I'm reluctant to do that for a couple reasons. It is possible, and we have had reports of it, that Windows will not boot if you switch the system into always-on mode, or if you increase the strength. You could have drivers, third-party drivers – all of Microsoft's drivers are just fine. And I've got systems booting fine in always-on mode, many different systems, in fact. But it's possible that you could have a third-party driver that would cause a problem.

Leo: That's kind of a problem...

Steve: That's a problem.

Leo: ...because you can't know if you've got that or not.

Steve: Well, you can't. There are ways around it, but not something I can casually describe in a podcast. For example, this is done by modifying the boot.ini file, which is by default marked as a system hidden file in the root of your boot drive. So I don't want to go through here verbally telling people, okay, go make changes. It's a little freaky, like making changes to your registry. The good news is, if you are a power user, you know how to edit your boot.ini to create multiple boot profiles, which is what I have done. You're able to give yourself a menu when Windows boots, saying how do you want to boot? Always-off, always-on, opt-in, opt-out? So and that's what DEPuty is going to do for people. You'll be able to run DEPuty, my forthcoming freeware. And I don't mean to be pre-announcing this, but this is where I'm headed here, is you'll be able to run it, and it will completely make you safe. You'll be able to essentially have DEPuty change your boot.ini anyway you want to, to give you these boot options, to allow you to safely experiment. Once you find that you're able to get booted in always-on, you could remove that option from your boot.ini if you didn't want to have to choose it, or make it the default rather than being able to, for example, use opt-in mode.

So again, there are – unfortunately this is all sort of black art, black magic. On the SecurAble page, where I do talk about this, I have a link to Microsoft's Knowledge Base article describing DEP because I didn't want to tease people saying, well, you need to use DEPuty to do this. You don't, actually. You can follow Microsoft's instructions, although they don't talk about this nice dual boot solution, which is a little bit more advanced. They talk about how you can edit your boot.ini file with these different settings. I'm going to leave it there so that only experts go there. And if people fear to tread, don't worry, we're going to get you there with DEPuty in a really safe fashion.

Leo: So there is a GUI, Microsoft does offer a GUI in the system properties control panel. Does that not do the same thing as the boot.ini switch?

Steve: Correct, it does not. It allows you to move between opt-out and opt-in, but it does not allow you to activate the much more strong always-on mode.

Leo: And there's good reason for that, if it would keep you from booting. Only an expert should be messing with that.

Steve: Exactly. And so that'll be the – but basically I'll be providing a GUI in the form of DEPuty that allows people to safely make these changes and create a multi-boot mode so they're able to really begin running with this thing on in a safe fashion.

Leo: Although I should point out, if it's just modifying boot.ini, you could always get in a recovery mode, and it's just a text file, fix that text file and get the system booted again. But it's for experts.

Steve: Again, yes. I would hate to recommend something that causes people's machines not to boot. Which again, it's why Microsoft has backed off of this so far. I congratulate them and salute them for putting this in at all. It is a potentially way powerful solution.

Now, one person in our newsgroup posted a question a few days ago, saying, hey, Steve, why are you so hyped up about hardware DEP support when there's a way around it? What about return to libc? Okay. First of all, he's right. There are ways around this. What I did was I did a whole bunch of research on the value of hardware DEP and the strength of workarounds. Return to libc is a very clever but also very limited means for bypassing hardware DEP. And it's not a complete bypass. Again, it's very limited. In fact, I quote one of the authors on the SecurAble page already, I quote one of the authors of one of the sophisticated workaround techniques who himself says this white paper, which talks about ways you might be able to get around this, should in no way be construed as suggesting that [audio glitch] not very powerful things to do. They absolutely are. That is, hardware DEP is a potent solution, and this paper should be seen as demonstrating that the options for executing arbitrary code are rapidly diminishing. Which is really what's happening. He's basically saying, yes, we found some theoretical ways that it's possible maybe for a hacker to still get around this. But they're theoretical more than not.

The way this works is, notice that executable code is obviously not going to be marked as non-executable. That is, it'll be in memory which is executable. So imagine a buffer overflow which doesn't execute code in the buffer which we already know is marked as non-executable, but in fact jumps to some code that already exists in the system at a known location, and that that code ends up being something that the hacker wants to run. Now, the classic example of this is the actual code for implementing opt-out. So imagine that a buffer overrun occurs, and it's possible to jump to the routine already in Windows for disabling DEP-checking for a given application, and that you're able therefore to essentially opt the program out so that when you return from that, then you are able to execute code in your buffer.

Now, I can't even say that that's possible. But it's an example of a very clever trick where, because we've locked the buffer so that it can no longer be executable, instead the hacker jumps to some existing code that sort of helps to achieve their end. Again, you can see that this is a real stretch to imagine that that's very useful, which is why my enthusiasm for this is not daunted at all. The fact that there are some theoretical ways around this kind of very strong buffer overrun and buffer execution prevention in no way diminishes the value of this to be applied system-wide.

Leo: And you'd have to really get on that system and have some real access to it to make that kind of a change, wouldn't you?

Steve: Well, as a matter of fact, that leads us directly into the other feature that Vista has which unfortunately XP does not have. And that's this ASLR, Address Space Layout Randomization. Notice that the absolute crux of that exploit was that it knew where some code would be sitting in the system. If the system now starts randomizing the location of its programs, of its own system programs, then it's not possible to know where in memory the code is. And that's what ASLR, Address Space Layout Randomization, does. It randomizes the layout of the system's address space so that hackers cannot know where these things are going to be in memory. Now, the bad news is, that's not available for XP. The good news is, there is some freeware which adds this to XP. It's been around for a few years, and we'll be talking about it in the future. So address space layout randomization sort of adds the final touch to hardware DEP, making it even more impossible for hackers to exploit using any kind of hardware DEP bypass.

Leo: What is the default for Windows XP? Always-off? Or is it software?

Steve: I think it's opt-in, yes. And in fact, software – really interesting that you mentioned that. Software DEP – and I'm trying to be very good about always saying "hardware" DEP. Everything I've been talking about so far in this podcast is hardware DEP. And everything that's SecurAble talks about is hardware DEP because software DEP really isn't.

What I think must have happened is many years ago some manager guy was saying to his coders at Microsoft, hey, guys, what can we do to fix these buffer overrun problems? And they said, well, the chip manufacturers are going to be adding a feature which pretty soon everybody's going to have, certainly everybody buying new machines, which will allow us to stop in the hardware any kind of buffer overrun. And so the manager said, okay, except that we're doing Service Pack 2 now, not in three years. So what can we do for Service Pack 2? And the programmer guy said, nothing. And the manager said, wrong answer. I don't want to hear nothing. Give me something. And so one of the programmers had a brilliant thought. He said, well, there was a hack once where somebody used Structured Exception Handling, SEH, in order to get control of a system. And we could do a better job of checking that and so that doesn't happen again. And so the manager says, oh, is that software DEP? And the developer said, no. And the guy said, oh. But let's call it that. And so they said, oh, but it's not. And the manager says, okay, well...

Of course this is a synthetic conversation. I don't know that anything like this ever happened at Microsoft. But given the evidence, it would seem that something like that happened. So Microsoft calls it "software DEP," and it's not. It's better than nothing because it does prevent the exploit of one very limited class of problems, nothing like what hardware DEP does. Nothing like really blocking the execution of code in a buffer. So having software DEP is not the same as having hardware DEP. But in Microsoft's defense, it did help to increase the security of Windows a little bit. Not nearly as much as hardware DEP. But on the other hand, it existed before hardware DEP support was available.

Leo: Right. It's the best they could do.

Steve: It's not a bad thing. It's the best they could do given the hardware at the time. But sadly, hardware DEP is – I think it's in opt-in mode, which says that if the BIOS does not have...

Leo: It is. I just checked. It says the default is opt□in.

Steve: Yes. So if the BIOS has not disabled it, as unfortunately some BIOSes have, so if you turn it on in the BIOS, then you'll be in opt-in mode, which gives you some protection for Microsoft's own code, which is certainly better than none. Microsoft of course made sure that none of their own code would be false-triggering hardware DEP. We just need to get the rest of the industry to be similarly responsible, in which case we're going to be in really great shape with any processors that have been made for the last few years that have this stuff enabled.

And so we'll be coming back to this at least one or two more times in the future, Leo, as I have more of my own solutions ready. Anybody who's listening to this could go to the SecurAble page, grab a copy of SecurAble, run it on your system, see if you've got hardware DEP. And if you are a power user and feel comfortable finding and modifying your boot.ini file, there's a link to the Microsoft Knowledge Base page that tells you how to change it from opt-in to opt-out, which is it seems to be really safe to use opt-out. And maybe you could try always-on, which is certainly the strongest of all. But we do see more problems with always-on. I'm just hoping that we'll begin to create some pressure so that the third-party program developers and driver makers will be able to make their software compatible with always-on, which is going to end up giving XP users and Vista users much stronger security protection than they've ever had before.

Leo: Yay. So get SecurAble. I'm downloading it right now because I want to turn on my hardware DEP. And I just got a boing, and it says, yes, hardware DEP is available. Very cool. Not hardware virtualization. I'm running an AMD FX64 chip, so. But that's okay because it's running Windows, and that's all I want it to run. But I'm going to turn on hardware DEP. If you want to get your copy of SecurAble, you just go to GRC.com. You'll also find many other useful tools there, like ShieldsUP where you could test your router. All these are free, by the way. Many, many more. And one that's not free, but very much worth the money, and that is, of course, SpinRite, everybody's favorite, my favorite hard drive recovery utility, great for maintenance, keep an eye on your drive, what it's doing, fix bad sectors, recover data on bad sectors, and a whole lot more. It's from GRC.com. For testimonials, to learn more about SpinRite, SpinRite.info is the site to go to.

Steve: We hear from people all the time. I got a nice piece of mail. Someone named Mark

Kramer in Pasadena down here said: "Mr. Gibson. after listening to you and Leo on Security Now!, and Leo and his other Netcasts, I decided back in November to buy a copy of SpinRite just in case I ever had a problem. Well, after coming home after the holidays, I found my Windows XP/SP2-based PC in a nonbootable condition. I tried several reboots, and it kept getting hung at the Windows XP splash screen. I was going to just reload the drive from an image I had cloned about a month ago. But then I remembered that I had a copy of SpinRite 6. I shut the machine down, put in the floppy, and turned it back on. After answering the easy-to-understand prompts, away it went. It was done in approximately 45 minutes, and I exited the program and rebooted the computer in anticipation. Sure enough, the PC went through the splash screen and on to my Windows log-in prompt. Everything seems to be back to normal. Thanks for your wonderful product and your contributions to the tech community. Hope you have a great new year.

Leo: Isn't that nice? You must...

Steve: So I just love...

Leo: You get those every day, don't you.

Steve: Yeah, we get them all the time. It's funny, as I was sorting through these, I thought, which one am I going to read? It's hard to choose.

Leo: Decisions, decisions.

Steve: I just love the fact that we're able to fix these things for people because these days, with drives being so big, and as we talked about, I guess it was two weeks ago, the shocking – even to claim the low reliability of drives, where Seagate is saying that they'll have 0.34 percent failure per year, it's like, yow. People are storing stuff they really need on their hard drives, and SpinRite is, as you say, the best utility for getting it back. So I'm glad to do it.

Leo: And you are the best guy for explaining security. Steve Gibson, what a great job. I know there are a few people who tuned in thinking that we were going to talk about Johnny Depp today. I'm sorry, this is the wrong podcast. But we do now know about hardware DEP. And I'm going to reboot because I just changed my boot.ini while we were talking. I'm brave. I'm bold. I'm scared.

Steve, a great job. We'll see you next week. Have a great week. I will sail back to you, and we'll get on the line via Skype for another one next week.

Steve: Sounds great.

Leo: Take care.