Blue Pill

Description: Steve and Leo continue their ongoing discussion of the security implications and applications of virtualization and virtual machines. This week they examine the "Blue Pill" OS subversion technology made possible by AMD's next generation virtualization hardware support. They debunk the hype surrounding this interesting and worrisome capability, placing it into a larger security and virtualization context.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-054.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-054-lq.mp3

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

This is Security Now! with Steve Gibson, Episode 54 for August 24, 2006: Blue Pill.

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

Once again, I am on a boat. The last time we did this, I got a number of messages saying, wow, how do you Skype from a boat? Well, we don't. We record these ahead of time, that's all. Steve Gibson...

Steve Gibson: Yup, to work around our schedule.

Leo: Yeah. Actually, I think I'm not on a boat. I think I might be in Canada. I don't know. All I know is I'm not here right now. But Steve is, and we are here to talk tech.

Steve: Yup.

Leo: And security in particular. And a lot of great response to last week's episode on virtualization. Do you have another – you said we would do a little more on that.

Steve: Oh, yeah. We have a few more episodes. I do want to mention that I got some great feedback from people who have installed and are using NoScript. You know, we talked about the...

Leo: Oh, I installed it, yes.

Steve: Yeah, we talked about the serious problems of JavaScript. And of course I'm anti-scripting in general because I think it's just fundamentally a bad idea to let websites you visit run code in your machine. And people are like, oh, yeah, there's that Gibson, you know, he's all freaked out about, you know, this stuff that's not a problem. But then these researchers came up with a way of JavaScript-scanning people's internal networks, which caught a lot of people's attention. I mean, it's not like the end of the world, but it further demonstrates that, just as a concept, I mean, philosophically, having scripting enabled is not as secure as not having it enabled. So, you know, there's a way of configuring IE that we've talked about, Internet Explorer, where it automatically switches its security based on whether the site you're going to is in your trusted zone or not. And similarly, NoScript for Firefox is an add-on that provides similar functionality. And I got a bunch of feedback from people who are using it and just feel good to know that their scripting is off unless they choose to turn it on.

Leo: Well, I installed it, but so many sites broke I just turned it back on.

Steve: Yeah.

Leo: It's kind of a tough one. And a number of people pointed out to me, including Randal Schwartz, who is our resident naysayer, but a security expert and certainly somebody I trust, that the exploit isn't too hazardous as long as you've renamed, you know, changed the password in your router. In other words, if you've pursued normal kinds of security recommendations that you make all the time, you're not too vulnerable.

Steve: Right. Right.

Leo: So I guess as always with security it's a balance between...

Steve: Well, but remember, he's responding to this specific instance...

Leo: Right, and there may be more.

Steve: ...which wasn't known a few weeks ago.

Leo: Right.

Steve: So what else is there? I mean, so what I'm saying is just, you know, there are things that are clear policies that regard security. I mean, for example, it used to be that people had their networks wide open, and they closed ports when there were problems. Okay, bad idea. Instead, have your system closed, and open ports that you need. So like, you know, there was that inversion. And similarly, you know, even if you think you have a service which is receiving packets that's safe, you know, it's just – it's not good to have it running if you don't need it. And of course that lesson we learned with the Universal Plug and Play exploit that was just uncovered a few weeks ago. Well, actually it was originally uncovered back in February by the EI guys. Again, another perfect example of where running with a secure policy just closes you

down in general and prevents unknown problems from occurring.

So a similar policy is scripting is bad. I mean, yes, I know. It's virtually necessary these days. But from a security standpoint, as opposed to a convenience and features standpoint, which I understand it provides, you know, it's bad.

Leo: I'm turning it back on. Hey, at least it exists as an option, and it's a really good option, I have to say. It works very well.

Steve: But you are a Mac guy, Leo; and so in general your, you know, the attack target size for the Mac is substantially smaller than it is for a Windows-based, you know, IE system. And of course Firefox is substantially more secure than IE, just based on the past. And with security, that's what we have to go on. You cannot declare – oh, and in fact this is – I've got a perfect example of that in this episode. But you cannot declare that something is secure. History needs to prove its security. So...

Leo: A very good point. It's that positive versus negative. Yeah, you're absolutely right, yeah.

Steve: And one last thing. As we mentioned the sale of Hamachi to LogMeIn, I had some dialogue with Alex Pankratov, the father of Hamachi – who you remember, you know, we contacted and talked to back when we discovered Hamachi – because several people asked about that Hamachi server we referred to. And I was impressed, Leo, that you remembered there was such a thing because I hadn't heard of it or remembered about it for a long time. But the idea was that, when we were originally learning about Hamachi, the idea was that there would be this server that people could run themselves to, like, set up their own Hamachi network and not use Alex's central servers to do all of the liaison between the Hamachi agents. So a couple people wrote in saying, hey, you know, if Hamachi's going to go into unknown limbo condition with the sale of LogMeIn, I mean, you know, like we never know what's going to happen when one of these things happens. It's like how the Sysinternals site you couldn't get to for a few days when the announcement of their purchase by Microsoft was made because everyone was frantically sucking down all the cool utilities that Mark had written before, you know, maybe they would go away.

So anyway, what Alex explained was, first of all, he really vetted the LogMeIn people carefully. I mean, his same concern for security and that the people there understand, like, the rights of users and privacy rights and all that. I mean, he's really comfortable with them as the new parent for Hamachi. So he wanted to make that clear. And he said, however, that the server project never got off the ground. It was...

Leo: Oh, so I was mistaken in pointing people that way.

Steve: Well, no, I mean, I'd forgotten about it completely, so I was impressed that you even remembered. But it was at best in – at sort of an alpha-level development, and it never would have been, he explained, a software-only offering due to concerns about piracy. I mean, it's just, you know, Hamachi's so popular, if he had released a, you know, some service or Hamachi server software, it would be, you know, all over the world in no time. So it would have always been tied to some sort of hardware platform so that, you know, you had to buy a Hamachi appliance from Alex and company in order to set up your own network, as opposed to just being something that you could download.

So anyway, the server, while not dead, it really – it never existed. And he did try to explain the

benefits of it to the LogMeIn folks; and you know, they now have it. So whatever it is that's Hamachi is now the property of LogMeIn. But, you know, there is at this point no alternative for people other than to, you know, follow Hamachi over to LogMeIn with Alex's assurances. And, I mean, he really feels as good about them as he could. Oh, and by the way, and I was saying, wow, Alex, 3 million users? And he says, uh, well, that's old news. Actually we were more like 3.9.

Leo: Wow. Wow.

Steve: And he said...

Leo: He ought to be sending you a gift of some kind.

Steve: Well, he thanked us again for, you know, discovering them early and really putting them on the map. You know, I guess his – I think his wife was giving him a hard time for months, you know, over the title of our podcast, Hamachi Rocks. So...

Leo: Yeah. Well, I – it just – again, and I'm just going to reiterate, and I'll stop, but that just is why I like open source solutions. I wish there were an open source solution as easy to use as Hamachi. That's the thing, is OpenVPN is out there, but it's, as we all know, it's difficult.

Steve: Oh, boy.

Leo: It's a nightmare to get going.

Steve: Yup. And I'll say again that I've, you know, it's funny, my prototype menu, remember we talked last week about the script-free, CSS-only menu that I spent almost two months developing, it's really taken off, Leo. A thousand people a day are going to that prototype menu page. And I've had people saying, well, but it doesn't validate. So it's like, okay, fine. So I went over, and it was – I added my standard GRC footer stuff, which has all kinds of old HTML in it that just, you know, it generated 112 HTML validation errors. And I go, okay, I'm just going to get rid of that. So I took that off, and then I fixed a couple little things, and now it does validate. So, but, I mean, it's – people are really interested in the concept of a script-free, robust, CSS-only menu. And, I mean, a thousand people a day are looking at that page and sucking out the CSS and adapting it to their own purposes.

Leo: Yeah, yeah, yeah. That's really – that's great. Good job. I think you may end up – nobody would expect this of Steve as, you know, a premier web developer. But you may end up being – going down in the hall of fame as a website developer, of all things. All right. So that's catching up on past episodes.

Steve: Yup.

Leo: Any other past business?

Steve: Nope, I think that brings us current.

Leo: So what is our topic for today, my friend?

Steve: Today, Blue Pill.

Leo: Oh, I'm excited. I'm excited. This is very interest- for a number of reasons it's interesting. There's a Red Pill, and there's a Blue Pill.

Steve: Yup.

Leo: And also it's interesting because one of the few women in this field...

Steve: Yup. Joanna Rutkowska is – she's been working with rootkit technology for years and has published a bunch of papers about, you know, rootkitting modern OSes, means of detecting, means of exploit, you know, I mean, she is like – she's a pure security researcher in, you know, in the best sense of the word because she's working to develop the concepts of rootkits in a context of, you know, what we're doing now, and publishing her results.

The thing that made so much noise recently was about two months ago she published a paper explaining Blue Pill. And, I mean, and then on two conferences, SyScan and then of course the recent Black Hat conference, she gave presentations where she demonstrated this. We're talking about it because it is right in our virtual machine series that we're doing. It's about virtual machine technology. And what really raised eyebrows was her claim that it was undetectable. And that's really the thing that sets it off from prior rootkit stuff. You know that we talked about rootkits extensively early in Security Now!, in our series. And in fact I would encourage anybody who hasn't read our early treatment of rootkits to, you know, grab those past episodes and take a listen to them because there was, you know, a lot of great stuff there. And I don't want to, like, completely restate everything that was said. But I do want to talk about, you know, what it is that Blue Pill is and how it's different from prior rootkits because it's fundamentally different. And in fact, Joanna has been, like, responding to people ever since her report because many people don't get what it is that she's done and what it is that she's saying. And so there's been lots of misconception. You know, Slashdot, of course, went insane over this. And, you know, Digg had threads. And as you know, Leo, when I even said the word "Blue Pill," you're just like, oh, yeah.

Leo: Oh, yeah, yeah.

Steve: So, okay. What happened is – and we touched on this already. I talked about how there is a clear evolution in what's happening with virtual machine technology. One of the things that's happening is that our next-generation hardware platforms are incorporating much more explicit support for virtualization, which is something that, for example, we talked about VMware last week. VMware, a lot of their technology is about working around the lack of support, I mean, the lack of, like, robust support for virtualization and making their VMware product offerings work anyway. Well, what AMD has done, and this is the platform on which Joanna's Blue Pill technology operates, AMD has something called their SVM/Pacifica virtualization technology. That's part of the Athlon 64 and the Turion 64 chips. It's essentially an extension of the X64 architecture that AMD – it's like a proprietary extension that AMD has added which adds much deeper support for virtualization. Now, Intel is coming along, they've had their Vanderpool chip as sort of their next generation moving forward. Now it's been

renamed VT. So Intel's next-generation chips will have this so-called "VT" technology which is their equivalent. In both cases, this stuff allows software to do a much – essentially a much better job of virtualizing the environment.

Well, what Joanna realized and then wrote proof-of-concept code for was that an OS running on this next-generation hardware – so first of all, no one with 32-bit platform hardware, nobody with 32-bit AMD or Intel hardware, has anything to worry about. I mean, this is all next-generation hardware platform requirement. But any operating system – Joanna implemented hers on the current beta, I think it was Beta 3 of Vista, you know, Microsoft's next-generation machine that's going to keep us so busy in the future. Any operating system running on top of this hardware has a problem in that it's possible for client software to assume the role of a so-called "hypervisor." Hypervisor is one of the new jargons that has been adopted, sort of to explain this notion of something running even above a supervisory level. So in general the operating system is called the "supervisor," since it supervises the operation of the various client programs that are running within it. So hypervisor is meant to imply something above the level of the operating system. And we've sort of touched on that when we've talked about, for example, VMware's ESX solution, where you run it, and then you install operating systems in it. So it's a hypervisor running above the level – or below the level, depending upon which way you draw your diagram – of the regular supervisory operating systems.

Well, it turns out that the code Joanna wrote – so the first thing that just made me grin was that it completely bypassed, I mean, just didn't even care about Microsoft's Vista new kernel protection stuff. Microsoft is going to great measures to make Vista more secure. We know, because we understand security, that it's going to take them a while to get the bugs out of all that. Well, one of the things that Vista requires by default is that all drivers be signed. This driver-signing requirement of Vista is very controversial because it means that you need to have a digital signature, and digital signatures are not free. So it's, you know, people are complaining that it's open-source hostile, it's free-software hostile and so forth. Microsoft is saying, well, yeah, sorry, but do you want security or not?

Leo: Right.

Steve: But anyway, so...

Leo: If you want open source, write it for an open source operating system.

Steve: Exactly.

Leo: You know, I mean, come on.

Steve: And so what Joanna demonstrated was that, leveraging the hardware platform and this Pacifica virtualization technology, she could, with all of Vista's security up and running, just cut right through it and slip – essentially this is sort of a super rootkit. She could slip her rootkit into the system, and it was undetectable. And undetectableness was the real thesis behind what she was talking about because – and now we'll talk a little bit about the rootkit and the normal rootkit technology and why it's detectable. She makes the point in her paper, and this is the whole point of what she did, that rootkits have traditionally relied on some sort of kludge or hack of some sort. They're hacking the kernel. For example, if a rootkit wants to hide itself so that it just cannot be seen by, you know, anyone looking for malicious code running in the machine – I mean, that's the whole point of a rootkit. Most operating systems, if not all, will somewhere – they'll have a list of running processes. And so – and a list is a programming term meaning – and it's also called a "linked list" – meaning that there'll be a structure that manages

and represents a process which will contain in it a pointer to the next structure for the next process, which will contain a pointer to the next structure for the next process. And so it's a series of these pointers form links that link this list together.

So, for example, when we run Task Manager in Windows, or you run Top in UNIX, what the operating system does is it has a pointer to the first process in the list. And it then – it enumerates these processes basically by going to each one, reading the pointer for the next one, and then using that pointer to essentially follow this linked list, or this chain of processes. So, for example, what a rootkit will do is, as soon as it starts up and is, for that brief moment, it's on this list of processes that are running, the first thing it needs to do is hide itself. So with knowledge of the operating system at the kernel level, which is below the knowledge that you would normally have that the operating system publishes in its API, which is what normal programs use, the whole point is the rootkit has knowledge sort of of the underlying technology of the specific operating system it's running on. It would itself follow this list of processes until it finds itself. Then it would go to the one before it that was linking to it, and instead it would change that link to point to the one after it, essentially unlinking it from this list of processes.

Now, when you run Task Manager or Top, it just doesn't appear because the operating system follows these links, and basically you've sort of created this lost process, the sort of, you know, it's off the reservation. It's no longer in the accounting system of the operating system, which the operating system inherently trusts. I mean, these are the structures that the operating system uses for managing processes. So a rootkit is able to, by having knowledge of the kernel, it's able to play games.

Now, one of the other things that we've talked about earlier in our podcast series, like way in the beginning, was the idea of hiding files. In order to hide files, the rootkit would do something similar. It would essentially – the technology is called "hooking," or "filtering." It would hook or filter the file enumeration functions, or API – Application Programming Interface – which all of the operating system uses in order to, like, make a directory listing. So when a program wants to do a directory listing or find a file or open a file, it says give me the first file in the directory, and then it makes a series of calls to obtain successive files. Well, the rootkit would intercept those functions that the operating system is offering to itself and everyone else. And it would, if it sees itself about to be returned, it would say, whoops, let's not give up our own identity, but instead we'll make a request for the next file after us and return that. So the idea is a directory listing that would have shown the rootkit files no longer does.

So the root – so these are traditional rootkit technologies that are well understood. And in every case – and this was Joanna's point – in every case they are dependent upon specific OS structure knowledge, and they're using a hack or a kludge in order to obscure their presence in the system. And her real point is, once you know what that hack or kludge is, the rootkit is detectable. And that's the big point.

Leo: You detect it by seeing the traces of what it's doing, not by seeing it.

Steve: Well, or for example you would come up – for example, say that it was hiding from having disconnected it from the process list. Well, there are other ways of enumerating processes. And so, you know, you would compare one way of enumerating versus following the list, and you would discover a discrepancy that would reveal the rootkit. Or you would, for example, go and directly access the hard drive to do what the operating system does when it's making a directory listing, and you would compare your result of direct access to the hard drive to the interface the operating system gives. And again, a discrepancy reveals what's going on. In fact, that's exactly...

Leo: That's what Rootkit Revealer does.

Steve: Exactly. That's exactly what Mark's Rootkit Revealer – it's the process it used. So...

Leo: So how does Blue Pill get around this?

Steve: Well, and so her final point is that, when rootkits are open source, and you can see exactly how they work, you're able to come up with a way of detecting them. And her whole point is that Blue Pill can be open source. I mean, she could completely publish what she's done, and it is no help.

Leo: How come? How does that – how do it work?

Steve: And the way it works is, it is a fundamentally different kind of rootkit because it isn't relying on obscurity of any sort or a hack or a kludge. It's simply using the hardware which has now been made available in the 64-bit architectures. And it's one of those things where, you know, the moment Intel and AMD and Microsoft learned of this, they're like, oh, no. I mean, it hadn't occurred to them, unfortunately, which is why Joanna's work is so important. Because, you know, they're all excited about adding new features. Well, again, another fundamental principle of security which never lets us down is, you know, new features really need to be looked at carefully from a security standpoint. So all Blue Pill is really doing is taking advantage of the next-generation hypervisory hardware built into, at this point, AMD's next-generation chips, the SVM/Pacifica technology that is now in those chips that Vista was running on, and she just turned it on. She said, okay, you know, I'm going to be a hypervisor. And the operating system said, yeah, okay, well, we don't really know what that is, but have a nice day.

Leo: Okay.

Steve: And so then people were saying, wait a minute, you know, being undetectable is impossible. And so there were a bunch of, you know, a bunch of fur flew. And Joanna has defended herself admirably because she's raised the point that, you know, once you are in this driver's seat as the hypervisor of the hardware, you know, you can control everything. And so people said, okay, wait a minute. What about the timing of instructions which you're having to filter? Because, for example, there's an instruction RDMSR, which allows you to read the MSR register. That's one of the hardware registers. And Bit 12 of the EFER register in there – I know this makes your eyes cross, but it's just, you know, this stuff doesn't really matter except Bit 12 says whether the processor is in SVM mode, that is, has this hypervisory mode been active. So she, once she slips into this role, basically turning the SVM mode on, and then immediately acquiring ownership of it, what she needs to do is prevent anybody else who wants to check to see if this system has been Blue Pilled, she needs to fake the bit of that register and show that it's off. Which means that she needs to intercept anyone's attempt to read the register. Well, in doing so she's inherently changing the timing of that read.

Leo: Ah. So she may be able to trick us, but we can tell because it took longer than it should have.

Steve: Exactly. In fact, in her studies, for example, there's another instruction, which actually I use all the time, called RDTSC, Read Time Stamp. RDTSC is a super-high-resolution timer. Actually what it is is it's a clock counter. So you literally get this 64-bit or, I guess in the next generation, 128-bit count of the number of clock crystal cycles. I mean, it runs at 3 gigahertz or 2.4 gigahertz of whatever speed your system is running. So, I mean, it's, like, infinite resolution because it's the clock cycle of the processor.

Leo: Right, right.

Steve: There is no higher resolution. So, for example, this RDMSR, the attempt to read this register, normally takes about 90 ticks, 90 clock cycles. But her filtered version, when she's in the system, takes about 2100.

Leo: See? We can tell.

Steve: Exactly.

Leo: So...

Steve: Except, except...

Leo: Oh.

Steve: Except...

Leo: Except.

Steve: ...that there's also something in this SVM mode called the TSC Offset.

Leo: Yes?

Steve: And it is an offset from what the TSC, this time stamp, returns.

Leo: Oh, so you could change the result.

Steve: So, exactly. So she filters the read to this timestamp. She also intercepts that. And she subtracts the difference of the time she knows she's going to take. So she simply sets the offset to negative 2010. And so when her instruction, which takes 2100 ticks, has 2010 subtracted from it, it returns 90. So again, I mean, it's a perfect example of how, once you're in this hypervisory mode, even, I mean, anything software does in the system, even like trying to measure the time of things, even time can be faked.

Now, you know, the first thing I thought when I was reading this is, okay, you need an external time reference. We need, you know, something not in the system, outside of the system's control. You know, so like NTP, the Network Time Protocol on the Internet, go, you know, get a real piece of time data. But that uses – that comes through the network interface and through the operating system and can be faked, as well. And so then she carries on the argument that, well, okay, wait a minute, how about having the user tell how long something takes? Because, for example...

Leo: Use a stopwatch.

Steve: Well, essentially exactly that. Some program would have to say, okay, user, push this button, and let's see how long it takes for me to do a certain amount of work, like do this read, this RDMSR a million times. And, I mean, it literally has to be about that much. So if it's really only taking 90 ticks, a million of those 90s would be 90 million. If it's actually taking 2100 clock ticks, a million of those is going to be, obviously, way longer.

Leo: Right.

Steve: But, I mean, but, you know, we're down to the point that that's the kind of detection fallback that is necessary.

Leo: But you could do that. I mean, there is – that would be – there's no way that they could prevent, you know, a manual detection timing of some kind.

Steve: I can't see how she could prevent that. On the other hand...

Leo: How many people are going to do that, is another...

Steve: And the whole point of Blue Pill is that you – by mistake you run this, and it silently, I mean, what's so very cool about this, you know, Microsoft Research has got something called "subvert," which is a similar sort of, like, you know, boot-time subversion of the OS. The research guys did this. But it's boot-time, and it requires writing things on the drive. So anything that's going to survive a boot needs to be on the hard drive. Blue Pill needs to record nothing on the drive. It's on-the-fly software that, if you touched it by mistake, suddenly, you know, I mean, and that's what's so cool, I mean, I love it, you know, of course all the analogies to the Matrix...

Leo: To the Matrix, right, yeah.

Steve: ...because, you know, you don't, I mean, nothing changes. Your screen doesn't even go [sound], you know. It's like, oh, what did you – what just happened there?

Leo: Right.

Steve: I mean, nothing happens, but now it's underneath your operating system, controlling your hardware, and is undetectable except through extreme measures. So she makes the point that, you know, what are you going to do? Do a test that's going to take 10 minutes to run every hour, to see if in the intervening hour your system has been Blue Pilled? I mean, there's no practical way to detect it on the fly. Now, of course Microsoft is not happy about this because, you know, here they are, oh, Vista's the most secure operating system ever. Well, we already know, listeners of Security Now! know how I feel about statements like that. I mean, they are ludicrous on their face.

Leo: Yeah, yeah.

Steve: You cannot declare something to be secure. And here along comes Joanna and says, uh, well, no.

Leo: Could you run – okay, since we're going to use virtualization to attack the machine, could you then run your machine in a virtual machine as well, and kind of protect it from the outside world?

Steve: Well, she makes the point that...

Leo: Double virtualization, in other words?

Steve: It's probably the case that, you know, again, now that Microsoft is aware of this, and AMD and Intel are like, oh, this is – maybe this is a little too powerful, I mean, obviously there are very wonderful, good purposes to which this can be put, the idea being that, I mean, this is what VMware's a little upset about. You know, a lot of their proprietary technology was working around the lack of really good hardware support for virtualization, which is now coming in all of our 64-bit platforms. So, you know, and it's going to end up being subsumed by the OS. So, I mean, it's sort of a next level of responsibility that the operating system will need to take. Joanna makes the point that, well, maybe you could turn it off in the BIOS. But if you turned it off in the BIOS, what's the point of having it?

Leo: Right. In fact, you can, on many machines that are coming with this kind of technology, turn it off in the BIOS. In fact, by default I think it's turned off. But you're right, then you don't get the benefits of the virtualization.

Steve: Yup. So, you know, again, it's that same sort of security cat-and-mouse game that we always end up playing.

Leo: But it does underscore really, I think – and by the way, I've seen the debates over this, and they go on and on, and they are not over. There are plenty of people will say, no, no, this isn't true. But it does underscore the risk of creating a hypervisor mode in general.

Steve: Yes, it is an extremely powerful mode. And, you know, what will probably happen is that Microsoft will, instead of doing nothing with it, which I guess is what Vista was doing, will, you know, they will turn on the SVM mode on the AMD, and they will themselves preemptively filter those things that are necessary to take over that mode. They just – they hadn't. And again, we can thank Joanna for doing this this year and not two years from now. Because, you know, Microsoft would have blissfully gone on and shipped Vista with this massive Blue Pill hole in its security, and then everyone would have been scampering around.

Leo: The chief critic of this is a guy named Anthony Liguori, who is working on a hypervisor for IBM. And obviously he doesn't like the idea of criticizing hypervisors. But in your opinion, is the real problem that a hypervisor exists at all?

Steve: No. The real problem is that it's new.

Leo: And we don't know what...

Steve: New is bad. New is bad for security. You know, in a year...

Leo: Just as you were talking about Vista's virgin stack...

Steve: Exactly.

Leo: Same issue.

Steve: Exactly. And so along came a, I mean, bottom line, I think this hardware technology is going to be fantastic because it will ultimately allow much more, you know, what it really means is simpler, easier, and absolutely zero performance penalty for doing multiple OS solutions.

Leo: Yeah. And we like that.

Steve: Oh, that's a win, yeah.

Leo: Is it truly zero? I mean, it really is...

Steve: It is as zero as you can get.

Leo: Wow.

Steve: I mean, and that's what they've done is that they've absolutely minimized, you know, all of the overhead associated with any kind of multi-OS virtualization.

Leo: Your discussion of all this has really inspired me to think about using virtualization more thoroughly. You know, right now I use Windows and Mac machines, and I have to use them roughly equally because I do – for instance, we record this show on Windows. I edit it on Windows.

Steve: Because we're using Skype?

Leo: Well, I use Skype on the Mac. I don't want them on the same machine and so forth. Anyway...

Steve: Ah.

Leo: But more and more I'm thinking, especially with VMware coming out with a solution for OS X, that maybe I should buy one of these new Mac Pros, which has so much horsepower, and just run both. Get a big screen and have Windows side by side with the Mac on the same system.

Steve: Sure.

Leo: You know, I mean, just it's really interesting what's happening with virtualization. Not just Windows, by the way, Linux as well. I have three operating systems, and of necessity three machines right now with a KBM switch. Boy, it'd be nice just to combine all that into one box.

Steve: Well, and, you know, I mean, there has always been this big divide between, for example, Mac and Windows. I think I've mentioned before that I've got a friend who's not computer literate. She struggles with do I click the right or left mouse button.

Leo: Right.

Steve: You know, and I always wanted to move her over to a Mac, where I think she'd just be more comfortable, except that she's a realtor who needs to access, you know, one application under Windows.

Leo: All users care about is applications. They don't care about operating systems. That's not – enthusiasts care about operating systems, but we're a minority.

Steve: Yeah.

Leo: Users just want to get the job done, and hopefully not get hacked in the process.

Steve: Yeah, I really think that, I mean, long term this virtualization is going to be a very cool thing. As we talked about at the very beginning of our series on virtual machine technology, you know, back when we were using DOS, and we now had – and the 386 chips came along, there were programs like Software Carousel, QEMM, you know, and TopView, that allowed us with a single keystroke to just click between applications.

Leo: Yeah, it was really slick. It was great.

Steve: It was phenomenal back then.

Leo: Yeah, yeah.

Steve: And so now we're talking about literally clicking between operating systems.

Leo: Hey, I like that.

Steve: I mean, imagine if you had a hotkey where, I mean, and we're talking robust because it's supported by the hardware.

Leo: Well, and even more secure, too, right? Because if anything goes wrong, you throw it out.

Steve: Oh, and see, that, yeah, that's a very good point. And in fact, in next week's episode we're going to be talking about sandboxing and, I mean, and the issues of sandboxing, which is sort of lightweight virtualization. And, I mean, and it's all about security. So when you have truly independent operating systems, and the hardware is supporting the barrier between them rather than software that is always subject to modification and subversion, when you've got hardware doing that, they're hardwired, you know, there isn't any way, if everything else is done correctly, for them to get to each other. But, I mean, but I love the idea. You know, because, I mean, there are solutions that kind of work and are flaky, and it's like, oh, okay, well, I'm – and I think it's probably the thing that has put you off of virtualization until now, Leo, you know, when we've been talking about it...

Leo: It's slow and unreliable, absolutely.

Steve: Exactly. But imagine a really, really robust technology where you could just hit a hotkey, and your screen just goes, click, to Windows.

Leo: I kind of have that now with Parallels on the MacBook. And I feel pretty good about – and the new version coming out will be even better. So I feel like there's good – what I really feel like is we're making really good progress in this area. This is a hot area right now.

Steve: Yup. Yup.

Leo: Now, should we worry about Blue Pill?

Steve: No. And that's really, I mean, Joanna raised the alarm. Microsoft got caught unaware. They've promised they are going to close this hole down.

Leo: Good.

Steve: Which must mean they will turn on SVM technology and take ownership of it and filter the instructions just like Joanna was doing...

Leo: They'll do their own Blue Pill, in effect.

Steve: Basically, yes. Vista will come pre-Blue Pilled.

Leo: That's often the way to fight these low-level Ring 0 hacking tools is to be one of your own.

Steve: Well, to get there first.

Leo: Get there first, yeah.

Steve: It's always about who gets there first.

Leo: Get there first. Interesting. And that's why it's so great that there's this robust, feisty, smart community out there that is challenging Microsoft. People might sometimes take umbrage at the fact that people are constantly attacking Microsoft. But that which does not kill you makes you stronger. I think that this is really what we need. It's kind of analogous to an independent free press...

Steve: Well, I will make a prediction here on Security Now!, Episode No. – what are we, 54?

Leo: 54.

Steve: 54, the first episode of our second year.

Leo: Or the second episode of our...

[Talking simultaneously]

Steve: Wait, did I number from zero or from one? The hypervisory technology will be compromised. Because, you know, for example, remember in the old days with DOS viruses, there were these things called "boot-sector viruses"?

Leo: Yeah, yeah.

Steve: Because the hard...

Leo: Michelangelo.

Steve: Yup. Because the hardware is now so powerful, a boot-sector virus to pre-acquire SVM rights could happen. And so, again, who gets there first. If something gets into the hardware before Vista boots, then it can fool Vista. And you can just see hackers salivating at the idea, I mean, just tackling it because it's there.

Leo: Oh, boy.

Steve: I know.

Leo: Oh boy. Or EFI, or getting into BIOS through the new EFI, even better than a boot-sector virus. You could...

Steve: If you can get control of the hardware before the OS boots, now the hardware is powerful enough that, you know, it really needs to be looked at carefully.

Leo: Fascinating stuff. As always you bring up the good issues. This is one that's very hot right now in the security community thanks to her presentation at Black Hat, and everybody's just buzzing about it. And I do hope Microsoft does something about it. And now you know.

Steve: They'll try.

Leo: Next week.

Steve: Next week, sandboxing.

Leo: Okay. More virtualization.

Steve: Yup.

Leo: It's the hot topic right now, and we cover it very, very – as always with Steve – in great detail, very thoroughly. We know that's what you want. This is one of the things I love about podcasting is that we can cover this stuff in detail. It's not the seven-minute segment that we had to do on Tech TV. We can really cover it in depth and give you the information you want. And by the way, if you want to follow it more closely, don't forget Elaine makes transcripts of every episode. So you can actually read along, which I think is a really helpful way to find out what Steve's talking about. And of course you own it, so you can play it back again and again. Lots of people share it with their friends, too.

Support the podcasts by visiting TWiT.tv and donating. Just a couple of bucks a month not only supports Security Now! but almost a dozen other podcasts, and two more to come by the end of the month. So we're very excited, you know, about this. Of course, as we add podcasts, our costs go up. We have to get new equipment and so forth. But we really feel like it's important to kind of get all of these podcasts on the air for you. So...

Steve: Well, and you're talking about getting a T1 now, too.

Leo: Well, and that's, yeah, I mean, there's always a little more, you know, I want to – people have been good. They've been very generous. The donations are strong enough to support things like, you know, spending a little bit a month to have very high-quality bandwidth. We lost a podcast the other day because the DSL went out. And I'm thinking, I'm basing this whole network on consumer-grade DSL. Wait a minute. Maybe we ought to get a T1 in here. So...

Steve: Well, and people really appreciate, I think, the show quality. I see mail from people who say, you know, that the TWiT family of podcasts sounds better than any other podcasts. And, I mean, it makes a difference just in, you know, in the comfort of listening.

Leo: I, yeah, well, that's of course a real goal of mine. And I'm not happy yet. And there are things we can do, but they cost. Everything costs. So...

Steve: We want you to be happy, Leo.

Leo: Your donations help. And it is only $2 a month. So please, go to TWiT.tv and press that button every month. A couple of bucks. Think of it as buying Steve a latte once a month.

Steve: Like I need more caffeine.

Leo: Just what you need. We do thank our sponsors, who also make it possible. The way it works is the donations go to infrastructure, the sponsorship goes to the hosts so that they don't – currently, you know, they're donating their time. I'd like them to get paid, too. For Steve, he gets a little money every once in a while from Astaro Corporation, makers of the Astaro Security Gateway, very good company that came forward early on and have really been committed to this podcast. And we really thank them. If your small or medium business network needs superior protection from spam, from viruses, from hackers, of course complete VPN capabilities, intrusion protection, content filtering, and an industrial-strength firewall, all in an easy-to-use, very simple high-performance appliance – I have one, the 120, it's just incredible. This thing is a rock. And it makes me feel much more secure. Contact Astaro at Astaro.com or call (877) 4AS-TARO to schedule a free trial of the Astaro Security Gateway appliance in your business.

You know, another group I'd like to thank, and I don't think I thank them enough, I really should thank them every single podcast – we do at the beginning – is AOL because they do such a great job. Without them really this network would not exist. The bandwidth provided, over a terabyte a day – a terabyte a day – from the folks at AOL Radio. AOL.com/podcasting. A tip of the hat to Jeff Graham and the server guys and everybody at AOL Radio. We thank you for your support, too.

Steve Gibson's website is GRC.com. Let's not forget that because that is Steve's day job, and we want to make sure – not only because of SpinRite, which is Steve's great product for file and recovery of your hard drives, I mean, it is just the ultimate hard drive tool which everyone should have in their toolkit. I certainly do.

Steve: You know, Leo, I was thinking about what I said last week about how I've stopped putting the testimonials on that page because that page is getting so long, and we've got so many of them. I thought, wait, maybe people are going to stop sending them. But I wanted to – it just makes me feel so good to hear these stories of success. So by all means...

Leo: I'll vouch for that. I'll vouch for that. Steve will come to me when we get together in Toronto or at other times, and he says, "Look at these," and he's got this list. I think that really gratifies Steve almost more than anything is to hear from you.

Steve: Yeah. Yeah. I love that it's able to help people so much. So by all means, you know,

send them if you want. I love to read your stories of success with SpinRite.

Leo: When you write software, it is, I think for anybody who codes, it's more than just a job. It's very gratifying to know that your code is being used to good benefit by people. SpinRite.info to see what people are doing with SpinRite and maybe get an idea of what you might want to do. GRC.com to get your copy. But of course there are also many very useful free security utilities there, including ShieldsUP!, which I use every time I set up a new router to check my router. I got all greens on my new D-Link, I'm so happy.

Steve: Yay, stealth.

Leo: Out of the box, all green. That's nice, a nice feeling. And that's where you'll also find the 16KB version of the show for the bandwidth impaired, and Elaine's great transcripts if you want to read along, follow along with – sing along with Steve. It's like Mitch Miller, kind of. Follow the bouncing security.

Steve: If anybody else knows who he is except you and me, Leo.

Leo: You know, in 20 years we're going to be doing this, a couple of old coots. And people are just going to say, we don't know – we have no idea what their references are, but it's just they're relics, and we honor them, and we let them do it. I have a feeling. We'll be in a museum somewhere.

Steve Gibson, have a great week, and we will see you next week for more virtualization on Thursday, every Thursday.

Steve: And I hope you're not seasick right now, Leo, wherever you are, out on the ocean.

Leo: [Groaning].